Intelligent Learning Automata-based Strategies Applied to Personalized Service Provisioning in Pervasive Environments

Anis Yazidi

Intelligent Learning Automata-based Strategies Applied to Personalized Service Provisioning in Pervasive Environments

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* in Information and Communication Technology

University of Agder

Faculty of Engineering and Science

2011

Doctoral Dissertation by the University of Agder 43
ISBN: 978-82-7117-706-5
ISSN: 1504-9272

Printed in the Printing Office, University of Agder
Kristiansand

Acknowledgements

I would like to acknowledge the financial support from Ericsson Research, Aachen, Germany who funded this project. I am very thankful to my supervisors: Prof. Ole-Christoffer Granmo (University of Agder), Prof. John Oommen (Carleton University), and Prof. Frank Reichert (University of Agder) for all their help and support. I would like also to thank Dr. Andreas Fasbender and Martin Gerdes of Ericsson Eurolab in Aachen, Germany, for discussions during the course of this research. From a more personal perspective, I am deeply indebted to my family for their lives of sacrifice that made this possible.

Anis Yazidi
October 2011
Grimstad, Norway

Contents

Summary

The vision of pervasive environments is being realized more than ever with the proliferation of services and computing resources located in our surrounding environments. Identifying those services that deserve the attention of the user is becoming an increasingly challenging task due to the fact that:

- The increasing number of services can overwhelm the attention of even the most educated user. It is, rather, plausible that an arbitrary user is not even aware of the services at his disposal.

- The changes of a user's preferences and needs, over time, renders the task of predicting his current services/interests extremely difficult.

- The result of the interaction of the user with any specific service is usually uncertain. It surely depends on the performance of the latter. As low performance services can provoke his dissatisfaction, it is mandatory that an expedient system must be capable of identifying reputable (and disreputable) services.

The complexity of understanding what services could be interesting and important enough to justify disturbing the user is one of the main challenges of our research.

One major aim of this thesis is to demonstrate that the fields of ubiquitous, unobtrusive and pervasive Computing can substantially benefit from the advances in the area Learning Automata (LA). In fact, LA can provide adaptive learning capabilities to a wide range of applications in Pervasive Computing. The success of the integration of LA and Pervasive Computing in the thesis would pave the way towards more research interest in this direction.

From this perspective, in this thesis, we present an adaptive multi-criteria decision making mechanism for recommending relevant services to the mobile user. In this context, "Relevance" is determined based on a user-centric approach that combines both the reputation of the service, the user's current context, the user's profile, as well as a record of the history of recommendations. Our decision making

mechanism is adaptive in the sense that it is able to cope with users' contexts that are changing and drifts in the users' interests, while it simultaneously can track the reputations of services, and suppress repetitive notifications based on the history of the recommendations. In accordance with the multiple dimensions that affect the decision making process, we have identified a set of enablers that constitute the core modules of our hybrid service provisioning architecture. Each of these modules, in itself, is a contribution in its own right.

In brief, we have devised a Reputation Manager that identifies reputable services in the presence of a significant ratio of deceptive referrals. While most of the legacy approaches are vulnerable to the ratio of inaccurate recommenders, the Reputation System (RS) that we have proposed has been shown to be robust to the undermining effect of inaccurate recommenders. In addition, it utilizes the power of Word of Mouth (WoM) communications in an optimal way in the absence of direct experience.

Designing a Novelty Checker for suppressing redundant notifications involves solving a fascinating problem of on-line discovery and tracking of noisy Spatio-Temporal Patterns. In this regard, we have presented a novel solution to this problem using the principles of LA. The solution is based on a new family of RWs with interleaving jumps. Beside the application domain, in a separate study, we have also examined the properties of this RW with interleaving jumps. The results that we have obtained constitute a significant contribution to the field of Random Walks (RWs).

With regard to users profiling, we have proposed a method by which we can use a family of Stochastic Learning Weak Estimators (SLWEs) for learning and tracking a user's time varying interests. Since increasing the learning speed of the SLWE is a problem in itself, we have tackled this issue and reported the first discretized version of the SLWE. This discretized weak estimator has the property that it can provide significant benefits to the Learning Preferences Manager.

Apart from all the above, we have presented an instantiation of our architecture for a real-life, day-to-day scenario involving a proactive location-based application which provides an ensemble of services. In order to evaluate the efficiency of our design, we have devised an overall simulation framework that simulates the functioning of the whole system when all the components are working together. The obtained simulation results confirm that our hybrid architecture avoids flooding the user with irrelevant information.

List of Abbreviations

ACA	:	Adaptive Clustering Algorithm
AI	:	Artificial Intelligence
$CUSUM$	:	Page's Cumulative Sum
EPP	:	Equi-Partitioning Problem
$FPAM$	:	Frequent and Periodic Activity Miner
FSD	:	First Story Detection
$FSSA$	:	Fixed Structure Stochastic Automata
GF	:	Gradual Forgetting
IP	:	Inaction-Penalty
LA	:	Learning Automata
LBS	:	Location Base Services
L_{IP}	:	Linear Inaction-Penalty
L_{RP}	:	Linear Reward-Penalty
$L_{R-\varepsilon P}$	:	Linear Reward-ε-Penalty
MC	:	Markov Chain
ML	:	Machine Learning
MLE	:	Maximum Likelihood Estimation
OMA	:	Object Migrating Automaton
PR	:	Pattern Recognition
QoA	:	Quantity of Affiliation
QoS	:	Quantity of Service
RE	:	Random Environment
RF	:	Relevance Feedback
RI	:	Reward-Inaction
RP	:	Reward-Penalty

RS	:	Reputation System
RW	:	Random Walk
$SDWE$	:	Stochastic Discretized Weak Estimator
$SLWE$	:	Stochastic Learning Weak Estimator
$STPLA$	:	Spatio-Temporal Pattern LA
TDT	:	Topic Detection and Tracking
$VSSA$	:	Variable Structure Stochastic Automata
WoM	:	Word of Mouth
$WWRF$	:	Wireless World Research Forum

Chapter 1

Introduction

1.1 Introduction

The growing number of ambient services in Pervasive Environments threatens to overwhelm the human's attention, raising new challenges for the task of service selection. Garlan writes: "*Increasingly, the bottleneck in computing is not its disk capacity, processor speed, or communication bandwidth, but rather the limited resource of human attention*" [1]. Filtering out irrelevant information has been a focal concern in a number of studies. The main issue of the research undertaken in this doctoral thesis is to reduce the cognitive load on the user when it comes to selecting services.

The Pervasive Computing paradigm envisages an explosion of resources, information and services that fall outside the boundary of the capacity of the user's attention. It is rational to say that today's computers distract the user, and that they can easily be an overwhelming source of interruption and distraction. For example, how can a user find an "interesting" restaurant from a large list of candidate restaurants? The increasing number of services can overwhelm the attention of even the most educated user. It is, rather, plausible that an arbitrary user is *not even aware* of the services at his disposal.

We submit that the user needs to be supported in order to deal with the almost-infinite amount of information and services latent in his ambient Pervasive Environment. We thus foresee that the overload of information can be avoided by means of the design of *unobtrusive* applications that are capable of operating in pervasive environments. The reader must observe that avoiding overload puts stringent constraints on the system to present to the user only the skeletal pieces of information that are needed and useful in his current context. Pervasive systems should also work towards reducing the interruptions to the human attention in the case of proactive applications. The user does not have an unlimited attention capacity, and so should not have to give it to things that do not matter! Simon [2], expresses it along the line of Garlan: "*What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the overabundance of information sources that might consume it.*"

According to the user-centric paradigm proposed by the Wireless World Research Forum (WWRF), the service provision should be tailored to the actual needs of the user [3]. The I-centric vision promotes personalization, ambient awareness and adaptability as the core requirements of future services.

In the context of service provisioning, since September 2001, the WWRF has mobilized a significant effort in order to carry out the vision of I-centric communication. According to the WWRF's vision, ambient services should be able to adapt to the needs of the user, so that the user is the center, instead of having rigid and inflexible services that "*are unaware of actual customer needs or situations*." From this perspective, the WWRF presents visions for building services based on analyzing the user's needs. The system acts on behalf of the user, and autonomously reasons about his needs and interprets his intentions. Pervasive Computing environments have to recognize opportunities and take the initiative of notifying the user if there is a belief that an ambient service is relevant in the user's current situation. Environment monitoring and event notification are therefore essential elements. For example, instead of the user having to manually figure out which services are available when he moves across different geographical domains, the system can make autonomous decisions, thus relieving him of the burden of "manual" service selection. Another example of this is the multimodal adaption of content while driving, such as the automatic reading of received SMS messages.

A number of research studies have focused providing context-aware service recommendations for mobile users. Context awareness permits the system to reason about the user's current tasks, and to infer his needs in a personalized fashion. While these studies are enlightening, one of the shortcoming of these studies is that they merely rely on the user's context, combined with a *statically*-defined user's interest, in order to personalize the service recommendation. Examples of these techniques are various nomadic context-aware applications such as tourist guide applications [4] and mobile marketing and advertising applications [5] as explained in Chapter 2.

The user-centric vision promoted by the WWRF has become the center of growing interest in the research arena. A pioneering recent work was reported by Hossain *et al*. [6] in which the authors proposed a gain-based media selection mechanism. In this regard, the gains obtained by ambient media services were estimated by combining the media's reputation, the user's context and the user's profile. As a result of such a modeling process, the service selection problem was formulated as a gain maximization problem. Thereafter, a combination of a dynamic and a greedy approach was used to solve the problem.

A pertinent study that falls in the same class as our current work is the *Dynamos* project [7]. The *Dynamos* approach is an example of a context-aware mobile application that can be used for recommending relevant services to the user. In [7], the authors designed a hybrid recommender system for notifying users about relevant services in a context-aware manner. The model is based on a peer-to-peer social functionality model, where the users can generate contextual notes and ratings, and attach them to services, or to the environments. They are also permitted to share these with their peers. The attached notes to the environment are delivered to other users whenever they are in the spatial vicinity of the entities associated with the notes.

A comprehensive study for providing personalized services has been performed by Naudet *et al*. from Bell Labs [8]. In [8], Naudet *et al*. designed an application for filtering the TV content provided to users' mobiles based on their learned profiles. The application is based on the use of ontologies to capture content descriptions as well as the users' interests. The latter interests are, in turn, mined using a dedicated profiling engine presented in [9], which leveraged Machine Learning (ML) techniques for user profiling. The work reported in [10] presented a system that recommends vendors' web pages by measuring the similarity between the user's profile and the vendor's web page when the user is in the vicinity of the vendor (seller). The user's profile is constructed through mining the history of his web log. Another example of a mobility-aware application is the *PLIGRIM* system that makes use of the user's location to recommend relevant web links [11]. In the same vein, the *SMMART* framework dynamically locates products that match the shopping preferences of the mobile user [5]. Another example is the *Mycampus* [12] product, which is a system developed and implemented by Carnegie Mellon University. *Mycampus* offers several different types of services including context-aware recommender services, context-aware message filtering services, context-aware reminder services, applications collaboration applications, and community applications. For example, apart from the system being able to recommend nearby services such as restaurants or movies, it can remind users about things they need to purchase when they are close to a store. It can also send messages to the user when he is not busy. The *Daidalos* project [13] is an example of Hybrid Service Recommendation System. The platform is composed of two layers which are tightly connected namely, the Service and Identity management layer, and the user-experience management layer. The service and identity management is responsible for service discovery and composition as well as for the privacy-aware service access via the use of virtual identities, so as to protect the user's identity from being revealed to the service provider. The layer also performs a ranking of the discovered services as per the main parameters which are the context and the user's preferences. The choice of the service can be made by the user from the ranked list, or, in turn, the *Daidalos* system can itself choose, for the user, the service possessing the highest ranking score.

1.2 Motivations and Objectives

1.2.1 Motivations

The thesis focuses on designing an unobtrusive architecture for service provisioning in pervasive environments. In order to motivate our thesis, we list the following:

1. From a high level design perspective, we would like to identify a set of enablers that constitute the core components of our hybrid service provisioning architecture. The latter task involves identifying the multiple criteria that the system should consider in order to autonomously decide, on behalf of the user, if an ambient service in the user's environment deserves his attention. We intend to separate the architecture into a set of core logical components that correspond to the identified multi-decision based criteria.

2. The identified components of the architecture should be able to seamlessly inter-operate in order to guarantee that the service provisioning is "unobtrusive" and that it minimizes the distraction of the user's attention. In this sense, through the thesis, we will study how we can realize the synergy between the devised architecture components.

3. Most of the reported context-aware recommendation systems do not consider the reputation of the services when issuing recommendations. In order to ensure that our hybrid recommender system is unobtrusive, we need to locate reputable services. The success of reputation systems (such as *Ebay*) suggests that there are significant latent benefits in the convergence of these ideas in pervasive environments.

4. In order to ensure minimal user distraction, the system should be able to track the changes in a user's interests, over time. In fact, static approaches, where the user manually defines his interest's domains, are usually not expedient as the user's needs and interests change over time. Therefore, appropriate ML techniques are needed for adapting to changing interests by *inconspicuously* monitoring service usage.

5. Repetitively reissuing the same notification regarding the same service or event is usually regarded as a nuisance to the user's attention. With regard to recommender systems, to the best of our knowledge, the question of suppressing repetitive notifications has not been addressed before in the literature. We would like to probe into this issue.

6. Most of the reported work in the approaches relevant to service provisioning

and RSs rely on uni-dimensional filtering paradigms i.e., on single-criterion based methods. In this sense, the latter approaches usually utilize a unique feature or dimension, (such as the reputation of the service, the similarity between the service description and the user's interests, or the context of the user such as geographical proximity to the service as in case of LBS), to assist it in recommending services to the user from the set of available services. As opposed to this, only few studies have been reported designing hybrid Service Provisioning Systems which combine multiple aspects and criteria in order to recommend relevant services to the user. In this thesis our intention is to fill the gap in this area by designing a comprehensive multi-dimensional decision maker for recommending services to mobile users – one that is not only multi-dimensional but also adaptive by virtue of it utilizing the powerful AI tool, namely LA.

1.2.2 Objectives

The objective of the thesis are the following:

1. The result of the interaction of the user with any specific service is usually uncertain. It surely depends on the performance of the latter. As low performance services can provoke his dissatisfaction, it is mandatory that an expedient system must be capable of identifying reputable (and disreputable) services. We would like to investigate how the principles of Learning Automata (LA) can be used to design robust Reputation Systems (RSs). The legacy RSs suffer from many disadvantages that limit their efficiency, including their sensitivity to inaccurate information that might mislead the RS, as well as an excessive computational complexity. Integrating LA would lead to novel contributions in the study of RSs. To the best of our knowledge, there is no existing work in the field of RSs for which the basis is fixed or variable structure LA. Indeed, our aim is to find a strategy by which we can merge the rich fields of LA and RS. In Appendix A, we have addressed the objective discussed here in greater detail.

2. In an Event Notification System, one can appreciate that repeatedly reissuing

the same notification over time could have the effect of overwhelming the user's attention with alerts that do not convey new information. As a motivating scenario, we focus on an application called the "Friend Notification Service", where the design requirement imposes the constraint of alerting the user of novel events, and of suppressing notifications for regular meetings. The problem can be modeled as one which involves the on-line discovery and tracking of noisy Spatio-Temporal Patterns. To tackle this problem we intend to devise an approach based on a new family of Random Walks (RWs). The state space of the latter RW includes some interleaving jumps that pose key challenges when it concerns the analysis of its properties. The theoretical analysis of this RW has been shown to be a contribution in its own right to the field of Markov Chains. Therefore, we will attempt, in the thesis, to derive the properties of one such novel RW, and investigate if it admits some possible applications. In Appendices B and E, we have addressed the objective discussed here in greater detail.

3. The changes of a user's preferences and needs over time, renders the task of predicting his current services/interests extremely difficult. Most existing approaches resort to sliding windows in order to track the user's interests. The Stochastic Learning Weak Estimation (SLWE) [14] seems a promising approach to adapt to changes in the distribution of the user's preferences over time. We would like to investigate the design of a Learning Preferences Manager based on the SLWE. As far as we know, there is no existing work in the field of learning user's preferences for which the main computational tool is LA. This objective has been addressed in Appendix D.

4. From a theoretical perspective, we would like to attempt to hasten the speed of convergence of the SLWE and its ability to "unlearn" what it has learned so far. Such a contribution, would be simultaneously beneficial for the design of even faster techniques for profiling which are able to track the time-varying distribution of the user's interests. Following the discretization concept, many of the continuous Variable Structure Stochastic Automata (VSSA) have been discretized; indeed, discretized versions of almost all continuous automata

have been reported [15]. In the same vein, we would like to study the design of a counterpart discretized version of the SLWE which is able to attain a faster convergence rate. This objective has been addressed in Appendix F.

5. A major aim of the thesis is to demonstrate that the fields of ubiquitous, unobtrusive and pervasive Computing can substantially benefit from the advances in the area LA. In fact, LA can provide adaptive learning capabilities to a wide range of applications in Pervasive Computing. The success of the integration of LA and Pervasive Computing in the thesis would pave the way towards more research interest in this direction.

6. At the final stages of the Thesis, we would like to integrate the different components of the architecture into a single functioning system. The assessment of the efficiency of the proposed design will be done using some well defined simulation scenarios. In Appendix C, we have addressed this objective.

1.3 Research Approach

Stemming from these objectives and motivations, we shall construct a hybrid recommender system that minimizes the distraction to a user's attention while, simultaneously, maximizing the hit ratio of the service notifications. In this section, we present the applied research approach used in the thesis.

In accordance with the multiple dimensions that affect the decision making process, we have defined a set of core components following a bottom-up approach [16]. As per the bottom-up approach, the system is divided into a set of fundamental components, which are first specified in great detail. Thereafter, these initially separate and distinct components are linked together to form the overall holistic system. Following this research approach, we shall first commence by detailing each fundamental or primitive component of our architecture in a separate manner. These main components that are defined separately are: the Reputation Manager, the Learning Preferences Manager, and the Novelty Checker. At a subsequent stage, we shall demonstrate how all the components are merged together into a final complete system.

In order to evaluate the efficiency of our design including each core component as well as of the whole architecture, we shall make use of the art and science of simulation. Simulation represents a discipline in its own rights. We shall conduct simulations at two levels, namely at a macroscopic level, and at a microscopic level. At the microscopic level, the functionalities of each core component will be rigorously tested on synthetic data sets. The results obtained will then be compared with the results obtained by standard benchmark methods. The *quantitative* analysis of the simulation data will permit us to highlight some *qualitative* properties of each core component. On the other hand, at a macroscopic level, we have devised an overall simulation framework that simulates the functioning of the whole system when all components are working together. Such a framework, permits the reader to understand the dynamics of the general architecture, to evaluate our design, and to verify the synergy between the different components.

Considering now the thesis from the aspect of the various "disciplines" that are involved, we can state the following:

- From an overall perspective, we are dealing with problems in the fields of ubiquitous, unobtrusive and pervasive computing. The problems and techniques used also provide fundamental solutions to pertinent issues in social networking and mobile computing.

- The basic tools which we use fall within the family of AI and ML, and in particular, the algorithms that are involved in stochastic learning and LA.

- The primary mathematical science that we shall employ involves the design and analysis of Markov chains.

- Finally, from the perspective of experimental computing, we shall resort to the principles of discrete event simulation to both justify and verify the models and solutions that we have proposed.

This concludes our discussion on the research approach that we have resorted to.

1.4 Organization of the Thesis

In this section, we present the overall organization of this doctoral dissertation:

- **Chapter 2: Background:**
  This chapter presents the background material needed for the thesis. We first introduce the concept of RSs and then proceed to describe a representative sampling of the state-of-the-art approaches. We do this by providing insight into the concept of I-centricity vision as promoted by Pervasive Computing. At this juncture, we shall endeavor to place emphasis on the importance of the user's attention as a "precious resource" fundamental to the field of Pervasive Computing. Additionally, we shall also summarize the research studies that relate information novelty to the design of unobtrusive applications. Thereafter, we shall review the different state-of-the-art approaches that deal with tracking and estimating the user's preferences. Finally, we conclude this chapter by submitting a fairly comprehensive overview of the field of LA - which constitutes the main computational tool used in this thesis.

- **Chapter 3: Contributions:**
  This chapter summarizes the main contributions of the Thesis which can be categorized into six complementary areas that fall under the same umbrella of designing an unobtrusive intelligent architecture for service provisioning. These areas are:

  – Enhancing RSs using LA

  – On-line Discovery and Tracking of Spatio-Temporal Event Patterns

  – Designing User-centric Architectures for Personalized Service Provisioning in Pervasive Environments

  – Learning the Preferences of Users

  – Analysing Random Walk-Jump Process with their relevant applications in this architecture, and

  – Investigating and utilizing Stochastic Discretized Weak Estimators in various time-varying learning domains.

In each of theses cases, we have catalogued our contributions and mentioned the salient aspects of the solutions proposed.

- **Chapter 4: Conclusion and Future Research:**
  This chapter summarizes the work done in the thesis, gives the final conclusions, and presents suggestions for future research that can be pursued.

- **Appendix A:**
  In Appendix A, we present a paper which explains how we have designed a Reputation Service Manager for selecting high quality service providers, which, in turn, is a cornerstone component of our architecture. Finding ways to solve the Agent-Type Partitioning Problem and thus counter the detrimental influence of unfair ratings on a RS, has been a focal concern of a number of studies. Our LA-based adaptive approach gradually learns the identity and characteristics of the users which provide fair ratings, and of those who provide unfair ratings, even when these are a consequence of them making unintentional mistakes.

- **Appendix B:**
  Appendix B summarizes the work done in recognizing and signaling Spatio-Temporal Patterns. In these works, we introduce a new scheme for discovering and tracking noisy Spatio-Temporal Event Patterns, with the purpose of suppressing reoccurring patterns, while discerning novel events. To the best of our knowledge, the work reports the first *on-line* approach for discovering and tracking of Spatio-Temporal Patterns in noisy sequences of events.

- **Appendix C:**
  Appendix C summarizes the work done to integrate the overall architecture of the system. In this paper, we present an instantiation of our architecture for a real-life, day-to-day scenario involving a proactive location-based application which provides an ensemble of services. More specifically, we report that we have succeeded in building a personalized context-aware decision maker that delivers narrowly-targeted notifications to the user about relevant services in his environment. It is worth mentioning that by the expression “proactive”

architecture, we imply an architecture that pushes notifications to the user [1]. This is in contrast to the pull-based service selection approaches where the user controls when the services should be retrieved from the repository of services. In this case, the reputation and relevance of the services to the user's profile, and the relevance to the user's context could serve as criteria to limit the list of the services that are returned to the user.

- **Appendix D:**
  Appendix D summarizes the work done in Learning Preferences. In the paper included in this Appendix, we propose a method by which we can use a family of stochastic-learning based *Weak* estimators for learning and tracking a user's time varying interests. The objective of the work in this paper is to present a personalized *Learning Preferences Manager*, i.e, a *modus operandus* for capturing user's preferences. The latter will be able to cope with changes brought about by variations in the distribution of the user's interests, which will be where the SLWE plays a prominent part.

- **Appendix E:**
  Appendix E summarizes the work done in RWs with interleaving jumps. In this paper, we have considered the analysis of one such fascinating RW, where every step is paired with its counterpart random jump. Apart from this RW being conceptually interesting, it also has applications in the testing of entities (components or personnel), where the entity is never allowed to make more than a pre-specified number of *consecutive* failures. The paper in this appendix contains the analysis of the chain and some fascinating limiting properties.

- **Appendix F:**
  Appendix F summarizes the work done in designing and implementing Discretized Weak Estimators. In Appendix F, we report a novel estimator, referred to as the Stochastic Discretized Weak Estimator (SDWE), that is based on the principles of LA and which builds on the theory of [14]. The first estimator introduced is able to estimate the parameters of a time-varying bi-

[1] The Friend notification system developed in Appendix B is one such proactive system.

nomial distribution using only finite memory. The estimator tracks changes in the distribution by operating a controlled RW on a discretized space. The steps of the estimator are discretized so that the updates are done in jumps, and thus the convergence speed is increased. The analogous results for the binomial distribution have also been extended for the multinomial case.

Chapter 2

Background

In the introductory chapter, Chapter 1, we stressed the multi-faceted nature of the thesis. Considering the various fields of research involved in this study, it is prudent for us to initiate discussions about the relevant state-of-the-art discussed in this chapter by pictorially presenting the overall landscape in which the topics of research reside. This is best described by Figure 2.1 which can help guide the reader through this chapter, and assist in the understanding of the inter-dependencies between its different sections.

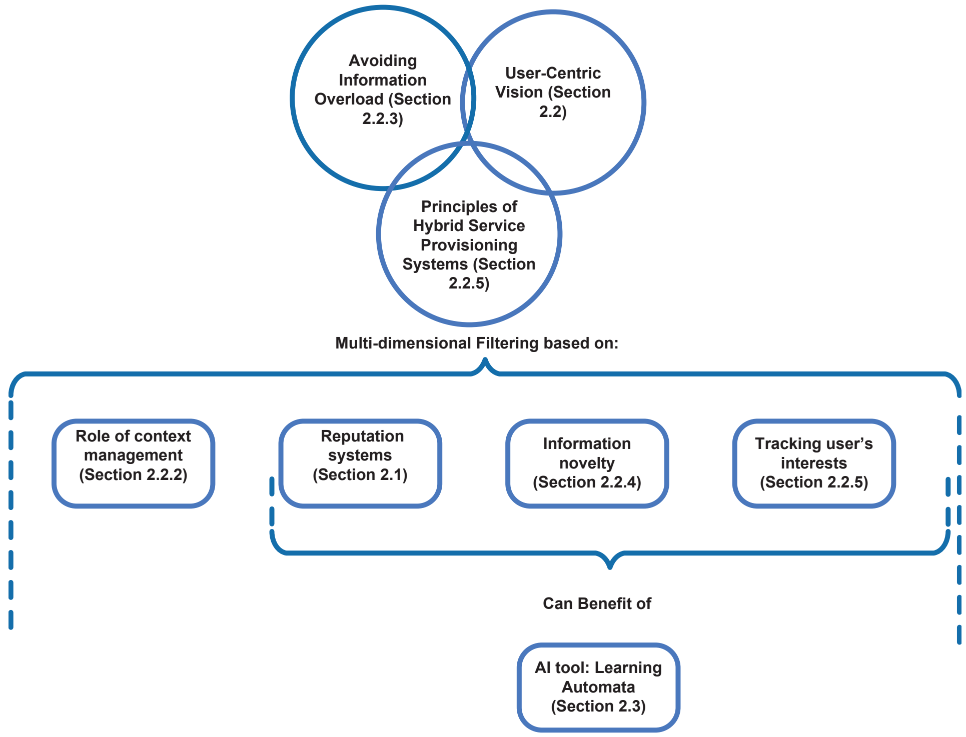


Figure 2.1: A pictorial representation of the different state-of-the-art areas included in this chapter.

2.1 Reputation Systems

In a competitive service provisioning environment, *reputation* is a distinctive feature that helps end users to identify high quality services. While traditional security mechanisms, such as access control, protect resources from illegitimate access by unauthorized users, interestingly, the role of a Reputation System (RS) is totally the opposite. In fact, the role of reputation systems is to protect the user from the services themselves, namely, those that present the risk of low performance. Low performance services result in the user's dissatisfaction, and therefore a reliable RS shall be able to single them out. In his seminal survey paper of trust and reputation systems in online service provisioning, Jøsang writes [17]: "*Online service provision commonly takes place between parties who have never transacted with each other before, in an environment where the service consumer often has insufficient information about the service provider, and about the goods and services offered*".

In this section, we will review a representative sampling of the state-of-the-art approaches for service selection that leverage the power of the Word of Mouth (WoM) strategies. Filtering out low quality services based on their reputation is a cornerstone dimension in realizing our distraction free service provisioning architecture, which is the objective of the thesis.

The reputation of a service is usually represented by numeric value in the unit interval – an estimate of the service quality. The reputation can possess either a discrete or continuous value which allows the system to differentiate between low and high quality service providers.

A viable way to acquire knowledge about a service provider's quality is through direct interactions. Such direct experience is often referred to as *first hand information*. However, in many cases, the user lacks direct experience to judge the service provider. The need to resort to the power of WoM is thus mandatory. The dissemination of information obtained from other users regarding the reputation of a service is referred as "*second hand information sharing*". Relying on second hand information is a remedy to the lack of direct experience. The above premise is true if the second hand information communicated by other users in the system is up-to-date and fair. Unless a person is naive, he must accept the fact that every user may not communicate his experiences truthfully. In fact, the social network and the system itself might contain misinformed/deceptive users who provide either unfair positive ratings about a subject or service, or who unfairly submit negative ratings. Such "deceptive" users, who may even submit their inaccurate ratings innocently, have the effect that they mislead a reputation system that is based on blindly aggregating users' experiences.

From this perspective, we can state a fundamental weakness about using RSs by virtue of the fact that they are prone to "ballot stuffing" and "badmouthing" in a competitive marketplace. Users who want to promote a particular product or service can flood the domain (i.e., the social network) with sympathetic votes, while those who want to get a competitive edge over a specific product or service can "badmouth" it unfairly, thus lowering its reputation.

In the absence of appropriate countermeasures, the RS becomes unreliable. Therefore, it is paramount to filter out and mitigate the effect of unfair reports.

2.1.1 Centralized and Decentralized Reputation Systems

Reputation systems fall into two categories: centralized RSs and decentralized RSs [17]. In a centralized RS, each service provider has a unique global reputation value known to all other participants. A reputation center aggregates all the results of interactions between users and services and computes a resulting global trust value which it thereafter makes available to all the agents[1] in the system. After an interaction with a service provider, the user submits his rating to the central authority, which in this case, is the reputation center. Therefore, untrustworthy service providers are identified by their low reputation value.

In the case of decentralized RSs, the computation of the reputation values are performed by the agent instead of the reputation center. In fact, each agent performs the task of aggregating ratings from other participants and then forms a subjective reputation value based on the combination of these ratings, possibly combining this with his own experience. The reputation is then subjective, instead of global, and will vary from one agent to another. The distributed nature of the environment makes it sometimes impossible for an agent to aggregate ratings from all agents. Consequently, the agent creates a local view of the reputation of the service based on the partial information he collected from reliable parties or his neighbors. In other words, the agent consults a subset of the ratings in order to form his opinion.

2.1.2 Trust Models

In this subsection, we describe a representative sample of computational trust models that represent the state-of-the-art in RSs.

2.1.2.1 Eigentrust

Eigentrust [18] is a well known RS that was devised for peer-to-peer networks. Eigentrust runs on the top of peer networks in order to limit the propagation of inauthentic, corrupted or malicious files by identifying their sources. Each peer is assigned a global trust value that reflects its trustworthiness as perceived from

[1] In this survey chapter, we use the generic terminology of an “agent” to denote an intelligent entity that can represent either a human user or an “artificially intelligent” machine that acts on behalf of the user.

the rest of the peer network. This global value is made publicly available to the whole peer network. Whenever a peer i interacts with a peer j, a local trust value s_{ij} is updated. s_{ij} reflects the opinion of peer i about peer j based on the history of the interactions, namely the number of satisfactory responses and the number of unsatisfactory responses as below:

$$s_{ij} = sat(i, j) - unsat(i, j),$$

where $sat(i, j)$ is the number of satisfactory responses when peer i interacted with j, and $unsat(i, j)$ is the number of unsatisfactory responses.

s_{ij} is normalized using the following formula:

$$c_{ij} = \frac{max(s_{ij}, 0)}{\sum_j max(s_{ij}, 0)}.$$

Using the principle of transitive trust, peer i asks its acquaintances for their opinions about other peers and weighs the opinion by the trust it places in his friends as:

$$t_{ij} = \sum_k c_{ik} c_{kj} \tag{1}$$

Thus, t_{ij} represents the trust that peer i puts in peer j based on the opinion of his set of acquaintances which are the peers that he has directly interacted with in the past.

Let C denote the matrix defined by $C = [c_{ik}]$. In that case, the latter equation can be written in its matrix form as:

$$\vec{T_i} = C^T \vec{c_i} \tag{2}$$

$T^2 = (C^T)^2 c_i$ is used to signify that peer i is soliciting the opinion of the friends of his friends, and $T^3 = (C^T)^3 c_i$ for the case when opinions of their friends is also considered.

A friend is reached through repetitive multiplication and aggregation of trust values until all the peers attain to the same stable value. Therefore, after n iterations, where n is the rank of the matrix one obtains the transitive trust. Observe that, T

should converge to the same vector for every peer i. Obviously, the computation of a peer's reputation relies on solving the stationary distribution of a Markov Chain.

Interestingly enough, the Eigentrust model bears similarity with the PageRank algorithm [19]. The latter algorithm is used by the famous web search engine Google to rank web pages according to their importance based on their connectivity. The intuitive idea of PageRank is simple: ingoing links pointing to a given web page increase its Pagerank, while outgoing links decrease the rank.

Both PageRank and Eigentrust possess the probabilistic interpretation of the Random Surfer model [19]. Without loss of generality, the Random Surfer model is a random walk on the graph where the stationary probability of staying at a given node is assimilated into the reputation of a peer in case of Eigentrust, and to the rank of the page in the case of the PageRank algorithm.

2.1.2.2 Tidal Trust Algorithm

The Tidal Trust algorithm is due to Golbeck [20]. It should be mentioned that it was applied in Film Trust, an online social network for rating movies. The Tidal algorithm belongs to the class of webtrust algorithms and its philosophy is based on the existence of a chain of trust between the user and the movie. In Tidal Trust, each user assesses the confidence that he has in each of his acquaintances using a score between 1 and 10. The participants of the RS and their relationships are modeled in terms of a graph, with each node being a participant (such as T_1 in Figure 2.2) or an object being rated (such as the movie m in the same graph). An edge describes a relationship between two nodes, in turn signifying the level of trust or rating, depending on the kind of nodes being involved. The algorithm works as follows:

- If a node s has rated the movie m directly, it returns its rating for m.
- Else, the node asks its neighbors in the graph for their recommendations.

Assume that a node s is related to a set of neighbor nodes S in the graph. Then the rating r_{sm} inferred by s for the movie m is defined using a simple recursive formula:

$$r_{sm} = \frac{\sum_{i \in S} t_{si} r_{im}}{\sum_{i \in S} t_{si}}, \tag{3}$$

where the neighbor nodes are indexed by i, t_{si} describes the trust of s in i, and r_{im} is the rating of the movie m assigned by i. The formula is applied recursively starting from the source, until the sink (movie) is reached.

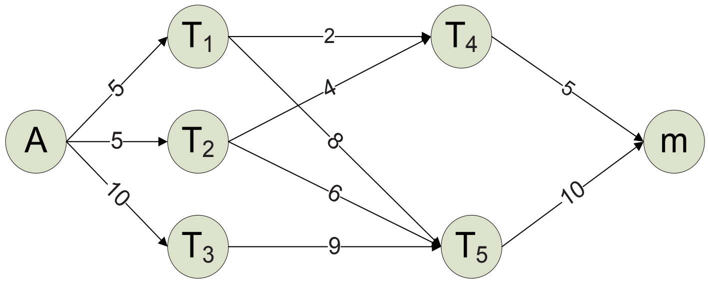


Figure 2.2: An example of Tidal Webtrust graph.

Later, Golbeck [20] introduced two improvements to the original Tidal Trust algorithm as follows:

- The first enhancement was to consider only short paths from the source to the sink by specifying a maximum length of the trust chain. This improvement was motivated by the observation that trust values inferred through shorter paths tended to be more accurate.

- The second enhancement was to discard the paths that involved untrustworthy agents, whose trust falls below a user-specified threshold. In this manner, the system maintains only the paths with strong trust values.

2.1.2.3 Online Websites

Ebay is an example of a well known online e-commerce website. Buyers are allowed to rate sellers with either a positive rating, a negative or a neutral rating. A positive rating reflects the user's satisfaction about the transaction, while the negative reflects his dissatisfaction. Ebay belongs to the class of centralized RSs. The reputation of a seller is expressed as the difference between the number of positive and negative ratings. Ebay also provides the ability to track the reputation of the seller over time by offering the possibility to view the seller's most recent behavior as for three time windows: past 6 months, past month, and past 7 days.

2.1.2.4 Sporas

The idea motivating the Sporas [21] trust model is that the users start with a minimal value of trust, and thereafter they build up their reputation over time via interactions with the system. A fundamental principle in Sporas is that as a user's reputation increases, the effect of ratings *diminishes*. In other terms, a high reputable user will not witness a significant change as a result of an interaction with another entity in the RS. Sporas borrowed ideas from the method used in the Elo and the Glicko system [22], which are the pairwise ratings methods devised to measure the skills of players in pairwise games such as Chess. Each user has a single reputation value, which is updated using the following formula:

$$R_{t+1}(n+1) = 1/\theta \sum_{i=1}^{t} \Phi(R_i) R_{i+1}^{other} (W_{i+1} - E(R_{i+1}))$$
$$\Phi(R) = 1 - \frac{1}{1+\exp(-\frac{R-D}{\sigma})}$$
$$E(R_{t+1}) = R_t/D$$

where,

- t is the number of ratings that the user has received so far,
- θ is a constant integer greater than 1,
- W_i represents the rating given by the user i,
- R_i^{other} is the reputation value of the user giving the rating,
- D is the range of the reputation values,
- σ is the acceleration factor of the dumping function, Φ.

A new user starts with reputation equal to 0, and can reach a maximum value of 3.000, while the ratings themselves vary from 0.1 to 1.

2.1.2.5 Subjective Logic

Jøsang devised a belief theory called subjective logic that handles the concept of uncertainty [23]. In traditional probability theory, the sum of of probabilities over all possible outcomes sums to unity. In belief theory, the sum is less than unity and the rest of the probability space is considered as uncertainty.

Jøsang proposed a belief/trust metric called *opinion* denoted by $w_x^A = (b, d, u)$ which denotes the belief of A in the truth of statement x. b denotes the belief, d denotes the disbelief and u denotes the uncertainty, where $b + d + u = 1$. Subjective logic framework is doted with many operators for combining beliefs; however, the most notable of them are the Discounting operator for transitive trust, and the Consensus operator for the cumulative fusioning of trust.

2.1.2.6 Abdul Rahman and Hailes's Approach

In their seminal paper, Abdul-Rahman and Hailes [24] considered trust ratings as discrete values. According to [24], humans can better express their trust in discrete values rather than by using a continuous number. The discrete values correspond to: Very Trustworthy, Trustworthy, Untrustworthy and Very Untrustworthy. Whenever a participant interacts with a service provider (or any other entity), he will update his trust level in the referrals. This referral possesses a reputation that reflects the trustworthiness of his respective recommendations.

The model is also based on the assumption that the direct experience of a participant with a service provider reflects the real trustworthiness of the service provider and therefore trustworthiness obtained by a referral can be deduced. In this sense, they suggest that the trust in the referrals that either overate or underate a service provider can be reduced.

2.1.3 Fighting Unfair Ratings

As alluded to previously, unfair ratings are an inherent problem for RSs. In this subsection, we present the state-of-the-art approaches related to filtering unfair ratings. In spite of the fact that the reliability of RSs is crucial to functioning ideally, very little work has been conducted in devising schemes that are resilient to unfair ratings that introduce bias into the RS. Indeed, the lack of such mechanisms typically implies compromising the RS itself because the reputation becomes misleading and it fails to reflect the true performance of the service.

2.1.3.1 Endogenous and Exogenous Discounting of Unfair Ratings

Jøsang and his coauthors [17] divided the approaches for handling unfair ratings into two categories, endogenous and exogenous. Endogenous approaches rely on the assumption that unfair feedback can be identified via its statistical properties. Therefore, endogenous approaches assign low weights or filter out the presumed unfair ratings. The idea behind endogenous approaches is to compare the ratings between themselves. For example, Bayesian RSs relies on the majority of the ratings of a seller to judge whether a new rating is fair or unfair. Therefore a rating is considered unfair if it is inconsistent with the majority of the ratings. The fundamental assumption is that unfair raters are the minority among the raters, and thus those advisors whose opinions deviate from the mainstream opinion are likely to be unfair.

The exogenous approaches rely on a rather different assumption than the endogenous approaches. The main assumption that they rely on is that the untrustworthiness of the inaccurate advisors can be determined via direct experience of the entity soliciting advices. The advisors should have a consistent experience about the service providers that is akin that of the entity soliciting advice. The TRAVOS model [25] and Yu and Singh's Weighted Majority Algorithm [26] fall into this category.

2.1.3.2 Dellarocas' Approach

Dellarocas [27] used elements from collaborative filtering to determine the nearest neighbors of an agent that exhibited similar ratings on commonly-rated subjects. He then applied a cluster filtering algorithm [28] to separate between honest ratings and dishonest ratings. Thus, the ratings data is separated into two clusters, i.e, the lower ratings cluster and the higher ratings cluster. Subsequently, the dishonest ratings were excluded from the reputation computation to eliminate the bias in the reputation score. It is worth noting that Dellarocas only considered the case of unfairly high ratings as those providing dishonest feedback. He did not consider the case of symmetric unfairly low ratings. Nevertheless, as is well known RSs may contain both unfairly positive and unfairly negative ratings.

2.1.3.3 Chen and Singh's Approach

Chen and Singh proposed a hierarchical RS that resorts to the concept of collaborative filtering for grouping raters according to the ratings they give to the same objects. The rationale behind proposing a hierarchy is to characterize each rater by an expertise that varies from one expertise domain to another. Each domain is, in turn, described by a node in the hierarchy.

A global reputation of a rater is computed as a combination of his local reputations in each expertise domain, where the local reputation is computed based on the similarity of the agent with the ratings provided by other agents. This approach mainly suffers from two problems. First, the model does not consider the direct experience of the agent with his advisors, even though that such information is crucial for determining the trustworthiness of the advisors. Secondly, the model possesses a heavy computational overhead.

2.1.3.4 TRAVOS Model

TRAVOS [25] is a RS that uses the foundations of Bayesian theory. In TRAVOS, an agent forms an opinion on the trustworthiness of an advice which is based on the history of the previous pieces of advice he was given. This opinion is characterized in terms of a Beta distribution.

When an agent aspires to interact with a service provider, he first checks whether he can predict the service provider's performance based solely on his own direct experience. If the confidence of the agent in his own experience is high, he relies solely on his own experience. Otherwise he solicits reports from other advisors who might have interacted with the service provider. The report from the corresponding advisors are combined together to yield a prediction of the service provider's performance, while assigning a low weight to inaccurate advisors. An interesting functionality in TRAVOS involves adjusting the ratings of the inaccurate advisors, so that the unfair effect can be mitigated.

2.1.3.5 Yu and Singh's Approach

Yu and Singh [26] used a version of the Weighted Majority Algorithm to detect and reduce the effect of deceptive and inaccurate ratings. The mechanism works by comparing the result of the actual interaction result with the information provided by the rater. The main idea of the algorithm in [26] is to assign a weight to every witness that reflects how credible he is. Before accessing a service the agent requests the predictions of the individual witnesses concerning the performance of service. The witnesses convey their predictions in the form of belief functions [26]. After accessing the service, the agent in question updates the weight of every witness based on the result of the interaction with the service. These weights are initialized with a value of unity and this value can be assimilated with the trustworthiness of the corresponding advisor. Deceptive agents will tend to submit inaccurate predictions, and thus their relative weights will decrease over time. Similarly, the weights of fair agents will increase over time. An aggregated prediction is computed by the agent in question as a weighted combination of the witnesses' predictions.

2.1.3.6 Deviation Test Method

Buchegger and Le Boudec [29] proposed a Bayesian reputation mechanism in which each node isolates malicious nodes by applying a deviation test methodology. According to the deviation test, each agent accepts second-hand information from other witnesses if the information does not deviate much from his own experience.

Their approach requires the agent to have enough *direct* experience with the services so that he can evaluate the trustworthiness of the reports of the witnesses. While this is a desirable option, unfortunately, in real life, such an assumption does not always hold, especially when the number of possible services is large.

2.1.3.7 Sen and Sajja's Approach

Sen and Sajja [30] proposed an algorithm to select a service provider to process a task by querying other user agents about their ratings of the available service providers. The main idea motivating their work is to select a subset of agents, who when queried, provide a minimum probabilistic guarantee that the majority of the queried agents provide correct reputation estimates. However, comprehensive experimental tests show that their approach is prone to the variation of the ratio of deceptive agents.

2.1.3.8 Iterated Filtering Approach

In [31], Witby and Jøsang presented a Bayesian approach to filter out dishonest feedback based on an iterated filtering approach. In their approach, the authors extended the so-called “Beta” RS earlier presented by Jøsang and Ismail [17].

In order to filter out dishonest feedback, the authors used an Iterated Filtering approach to exclude those ratings that are not in the majority.

The reputation of a service provider is expressed as a Beta distribution which is a probabilistic model to represent the probability of a binary event. In this vein, the reputation of a service provider is expressed in terms of a Beta distribution of which parameters are the number of positive ratings supporting a good performance and the number of negative ratings supporting a low performance, as cumulated from all witnesses.

The trustworthiness of a seller is then represented by the expected value of the beta function, which is the most likely probability value that the seller will act honestly in the future if all feedback is honest. In order to eliminate the bias introduced by the unfair agents, a feedback from a witness is considered as honest if it falls between the lower and upper boundaries of the cumulated reputation of the service provider.

2.1.3.9 Zoran and Aberer’s Appproach

Zoran And Aberer [32] proposed a probabilistic model to assess peer trustworthiness in peer-to-peer networks. In [32], they assumed that a peer can deduce the trustworthiness of other peers by comparing its own performance with reports of other peers about itself. Although such an assumption permits a feedback-evaluating mechanism, it is based on the fact that peers provide services to one another, thus permitting every party the right to play the role of a service provider and the service consumer (a reporting agent).

2.1.4 Collaborative Filtering Vs. Reputation Systems

The premise of Collaborative Filtering [33] is that the tastes and preferences of users might vary from one user to another, which consequently leads to differences in ratings. Therefore, the common approach is to find, for a given user, the like-minded users, based on the commonly-rated services. Similarity between users is often computed using the Pearson correlation, which corresponds to the conventional cosine similarity rule. A major difference between RSs and Collaborative Filtering is that the former supposes that the difference in ratings is due to unfair feedback, while the latter supposes that the difference can be explained by the difference in the tastes and preferences of the users.

Collaborative Filtering functions best in centralized systems, such as movies recommendation websites, or for books such as Amazon, where a huge amount of data is collected. It is not viable in small social networks, where the user is connected to a few direct friends from which he solicits advice because of the sparsity data problem [33]. Another inherent problem with Collaborative Filtering involves the so-called *cold start* which is what occurs when a user is new to the system and has not yet rated a sufficient number of items that can reveal his preferences. For an extensive review about Collaborative Filtering, the reader can refer to [33].

2.2 User Centric Vision In Pervasive Environment

In this subsection, we present some insight into the concept of Pervasive Computing pioneered by Mark Weiser, and we stress the importance of treating user's attention as a scarce resource in realizing unobtrusive applications.

In his seminal paper [34] titled "The Computer for the 21st Century", Weiser coined the term "Ubiquitous Computing" (also referred to as Pervasive Computing) to describe a new wave of computation that seamlessly supports the end user in the course of his daily activities, while disappearing into the fabric of our lives. According to Weiser: "*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*" According to this vision, the pervasive system should fade into the background, and the user will interact with it in a manner that does not require him to give his thoughts or much of his attention. In this sense, the user should focus on his current tasks instead of tediously spending time interacting with the system. Pervasive Computing defines a new interaction model between the user and his immediate daily environment.

Pervasive computing was boosted by the development in processing power, memory, miniaturization of devices, communication and distributed computing [35]. It depicts a picture of the future of computing for the next generation where environments of our daily-life are saturated with an abundance of devices, services, each of them possessing significant communication capabilities and computational power. These devices are connected forming an indistinguishable part of the environment. The multitude of devices are considered as being enabling technologies. Weiser [34] stressed the importance of a seamless technology that should create calm, and that should disappear into the fabric of our lives. Such a calm can be realized if the massive number of devices and associated computational power are enhanced with so-called *Ambient Intelligence* [36]. In fact, the intention is that the computers should be able to interpret the user's intentions, anticipate his behavior, while being silent and permitting him to focus on main tasks. For instance, an intelligent pervasive system could automatically switch on the heater in the house during winter as soon as the user is driving back home from work so that he can find his home warm and save energy at the same time. Another example is a system that, for a user, automatically books a hotel that meets his learned preferences, as soon as the user's calendar plans a business trip abroad. Such an adaptive behavior can be achieved by learning and observing the user's behavior or through precompiled forms.

Pervasive Computing stresses the importance of the concept of I-centricity, i.e, that of putting the user in the middle of the computation. Today, we are operating within the era of Computer-centric paradigm, and are moving towards the I-centric paradigm. Pervasive computing is all about putting the user, rather than a particular computing device, in the center of the computing activities.

According to the I-centric paradigm proposed by Wireless World Research Forum (WWRF), the service provision should be tailored to the actual needs of the user [3]. The I-centric vision promotes personalization, ambient awareness and adaptability as the core requirements of future services.

In the context of service provisioning, since September 2001, the WWRF has mobilized a significant effort in order to carry out the vision of I-centric communication. According to the WWWRF's vision, ambient services should be able to adapt to the needs of the user, so that the user is the center, instead of rigid and inflexible services that "*are unaware of actual customer needs or situations*". From this perspective, the WWRF presents visions for building services based on analyzing the user's needs. The system acts on behalf of the user, and autonomously reasons about his needs and interprets his intentions. Pervasive Computing environments have to recognize opportunities and take the initiative of notifying the user if there is a belief that an ambient service is relevant in the user's current situation. Environment monitoring and event notification are therefore essential elements. For example, instead of the user having to manually figure out which services are available when he moves across different domains, the system can make autonomous decisions, thus relieving him of the burden of "manual" service selection. Another example of this is the multimodal adapting of content while driving, such as the automatic reading of received SMS messages.

As Garlan writes: "*Increasingly, the bottleneck in computing is not its disk capacity, processor speed, or communication bandwidth, but rather the limited resource of human attention*" [1]. Filtering out irrelevant information has been a focal concern in Pervasive Computing. The main issue has been to reduce the cognitive load on the user when it comes to selecting services.

It is worth noting that Pervasive Computing is a multidisciplinary research arena that can benefit from a panoply of areas, including: Mobile Computing, Artificial Intelligence, Machine Learning, Communication Network, Social Science, Human Machine Interfaces, etc.

2.2.1 User's Attention as a Precious Resource

According to Garlan [1], the most precious resource in a computer system is no longer its processor, memory, disk, or network, but rather human attention. According to Moore's Law, processing capacity doubles almost every two years. As opposed to this, our attention does not follow Moore's Law. Human attention seems to be a bottleneck in this new era of computing.

In fact, the vision of Pervasive Computing envisages an explosion of resources, information and services that fall outside the boundary of the capacity of the user's attention. It is rational to say that today's computers distract the user, and they can easily be an overwhelming source of interruption and distraction. For example, how can a user find an interesting restaurant from an enormous number of candidates?

The user needs to be supported in order to deal with the overwhelming quantity of information and services latent in his ambient Pervasive Environment. Information overload is to be avoided by means of the design of unobtrusive applications able to operate in pervasive environments. The overload puts stringent constraints of the system to present to the user only information that is needed and useful in his current context. Pervasive systems should also reduce interruptions to the human attention in case of proactive applications. The user have limited attention, therefore he should not have to give it to things that do not matter! Simon [2], expresses along the line of Garlan, "*What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the overabundance of information sources that might consume it*".

A whole body of research has been devoted to designing distraction-free applications and to the area of preserving the user's attention in human-computer interactions.

In [37], cost and utility were considered in order to decide whether issuing a notification was beneficial. The cost of the notification was measured in terms of the perceived distraction of the user's current particular activity, and was compared to the utility. The latter, in turn, depends on the content of the message. This framework permits the system to compute the optimal timing for issuing the notification. The conclusion of the study was that the acceptability of non-urgent messages can be improved by taking into account the user's mental activity load at the time of notification.

In [38], the authors considered the user's acceptability of automatic notifications in a home environment. They differentiated between urgent messages that had to be delivered to the end user as soon as possible, and non-urgent messages that could be differed and postponed until the importance of the message had increased beyond a certain threshold.

A significant amount of research has been put into estimating the cost of interrupting the user. Some researchers express the cost of interruptions and notifications in terms of task performance. For example, traditional mobile devices available in the market issue a low battery warning sound whenever the battery is drained so that the user is alerted to charge it. Nevertheless, such warnings do not take into account the context of the user. For instance, the warning might result in annoying disturbance when the user is sleeping in the night. Thus, taking into account the user's situation is crucial in this context. For instance, Horvitz *et al.* from Microsoft [39] formulated notification issuing as a decision making problem under uncertainty, and resorted to inference using Bayesian Networks to reason about the user's attention. In their framework, the notification manager is an intelligent decision making engine that issues messages if the benefit of alerting the user exceeds the expected cost expressed in terms of disruption. Their probabilistic attentional model also permits the system to reason not only about the most appropriate time of notification but also about the modality and the best device to employ for alerting. The model integrates diverse sources of information as inputs, namely, sensor data that infers the attention level of the user, the user's location and situation, and it uses a module called "Context Server" to achieve this. From this perspective, the Context Server is responsible for gathering and analyzing data coming from the user's Microsfot Outlook calendar, infer the user's current activity, and invoke a Bayesian head tracking mechanism to get a clue about the user's attention level.

Another work along the same direction is reported in [40]. In this work, the authors employed data mining techniques to mine the user's primary activities and decide on the convenient time for interrupting the user by means of of a notification. In [41], the authors considered the availability of the user to deduce the call distraction.

In order to reduce the information load due to notifying the user about the reception of a new incoming e-mail, the results of [42] resort to assigning priorities to incoming e-mails. There are mainly two ways by which the priorities of received e-mail can be inferred: The first method is to learn theses priorities via a training phase, where the classifier is trained using some sample data. The second method is to infer the priorities by observing the user's interaction with the e-mail browser.

The study in [43] concluded that messages that are delivered during activity transitions are less distractive than those in the middle of the activities. The author of [43] claims that in an ubiquitous environment many heterogenous devices are competing for the user attentions. Thus, the vision of Ubiquitous Computing can result in disruption and information overload if countermeasures are not taken. To solve this, they used accelerometers to detect when the user's activities switched through postural and ambulatory activity transitions. Thus, the messages are delivered to the user during these inferred transitions, which was experimentally shown to augment the acceptability of the interruptions compared to the case where the message are delivered during other instants. The authors therefore suggested the development of similar systems that would postpone message delivery to instants of activity transitions.

2.2.2 Context Aware Service Provisioning

A fundamental element of personalization is context awareness. In this regard, "Context" includes any information that can be used to characterize the situation of a mobile user requesting a service. It could include numerous pieces of information such as the user's location (where), the time of presence (when), his current activity, his "mood" etc. The description of the concept of the "context" follows the pioneering works of [44, 45]. Indeed, Schmidt *et al.* [45] define context as: "*knowledge about the user's and IT device's state, including surroundings, situation, and to a less extent, location*". Dey, on the other hand, [44] defines context as: "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*". The author of [44] identifies three main streams of information to characterize context: spatial information such as location, speed, direction, physiological information such as heart rate, and environmental information such as noise and light level.

The user's location is the most-used contextual data in today's ambient aware applications. The context-aware application should be able to respond to different situations and to adapt to them. For example, as the user moves, the quality of streaming a video may degrade, and a context-aware system should be able to make an autonomous decision so as to maintain the quality of experience and to avoid the user from the burden of interruption. For example, the system might switch to another network that provides better quality or might provide the user with a lower fidelity of the quality of the video. In addition, context awareness facilitates trigging the execution of useful services or the issuing of relevant notifications in a proactive manner. Distractions pose even more of a problem in mobile environments than in desktop environments because mobile users must often continue walking, driving, or taking part in other real-world interactions [46].

In this vein, Smailagic and Siewiorek [46] write: "*A ubiquitous computing environment that minimizes distraction should therefore include a context-aware system able to read its users state and surroundings and modify its behavior on the basis of this information. The system can also act as a proactive assistant by linking information such as location and schedule derived from many contexts, making decisions, and anticipating users needs. Mobile computers that can exploit contextual information will significantly reduce demands on human attention*".

In pervasive computing, the environment is endowed with a multitude of sensors and computing entities that are able to sense the user's environment, analyze the stream of signals, and infer the activity and situation of the user. Understanding the user's context, helps the pervasive applications to adapt their behavior according to the situation of the user, enabling the invisibility of the pervasive system.

Inferring high-level context, such as the user's current activity, from low-level context data (for example, from sensor data) has been an active field of research. In order to infer context, usually, diverse pieces of information have to be acquired from distributed devices. The complexity of gathering and interpreting the contextual data is hidden from the applications and services. In this vein, it is worth noting that there are many pieces of middleware available in the marketplace, such as *Contory*, that can achieve the latter task. For a review of such context-aware middleware, we refer the reader to [47].

2.2.2.1 Context Aware Recommendations

Traditional Recommender System rely solely on the user's profile and the meta-description of the services to perform matchmaking and to present the user with the list of relevant services. As the recommendation process goes mobile, the user's traditional recommendation is no longer associated with his desktop. A number of research studies have been interested in providing context-aware service recommendations for mobile users. The most-used piece of contextual information used so far is the user's location, around which it articulates a class of context-aware services called Location Base Services (LBS). However, LBS rely mainly on the user'location, and do not encompass more context clues into the services recommendation process.

Context awareness permits the system to reason about the user's current tasks, and to infer his needs in a personalized fashion. The shortcoming of these studies is that they merely rely on the user's context combined with a *statically*-defined user's profile, in order to personalize the service recommendation. Examples of these techniques are various nomadic context-aware applications such as tourist guide applications [4] and mobile marketing and advertising applications [5].

A recent trend in devising context-aware applications is to integrate more contextual information, such as time and the companion, etc. This leads to an important general remark: We should emphasize that, in general, a user's interests are *context-dependent*. For example, recommendations about restaurants might be of interest to a certain user during weekends, when he is both close to the restaurant in question, and when he is not busy. Also, data related to tourist attractions can be of use in the context of tourism. Such contexts can be inferred from the fact that the user is on holidays (for example, from the user's calendar) and traveling. Therefore, a viable approach is to provide the user with the ability to specify that certain kinds of services (those of interest) are active in a particular context. The idea is relatively novel and has been recently applied in the *Dynamos* framework [7] where a user is permitted to specify several types of activities and their associated status, and to associate multiple interests to each of them.

The essence of this idea has also been utilized in the *Mobilife* project, where the user has different context-dependent sub-profiles [48]. In the *Mobilife* project [48], a user's profile can be separated into different sections that include personalized information relevant to different applications. The appropriate sub-profiles are then activated as a result of the changes to the context. By modeling these issues in this manner, the profile and context information are treated as meta-data that describe the user's specific instance.

The work reported in [10] presented a system that recommends vendors' web pages by measuring the similarity between the user's profile and the vendor's web page when the user is in the vicinity of the vendor (seller). The user's profile is constructed through mining the history of his web log. Another example of a mobility-aware application is the *PLIGRIM* system which makes use of the user's location to recommend relevant web links [11]. In the same vein, the *SMMART* framework dynamically locates products that match the shopping preferences of the mobile user [5]. Another example is the *Mycampus* [12] product, which is a system developed and implemented by Carnegie Mellon University. *Mycampus* offers several different types of services including context-aware recommender services, context-aware message filtering services, context-aware reminder services, collaboration applications, and community applications. For example, apart from the system being able to recommend nearby services such as restaurants or movies, it can remind users about things they need to purchase when they are close to a store. *Mycampus* can also send messages to the user when he is not busy.

We argue that relevant contextual information is truly pertinent and crucial in Recommender Systems and that it is important to take this information into account when providing recommendations.

2.2.3 Hybrid Service Provisioning System

The term Hybrid Service Provisioning System was coined in [7] to describe a service provisioning system which combines multiple aspects in order to recommend relevant services to the user. Theses aspects include:

- Context awareness: This takes into account the situation of the user by analyzing the different pieces of context information.

- Reputation Systems or Collaborative Filtering: A substantial amount of research has already been performed in the area of Recommender Systems, most existing approaches focus on recommending the most relevant items to users without taking into account any contextual information, such as time, location etc. Examples include applications that recommend interesting documents or interesting news to a user based only on his profile, and an online website that learns a user's video preferences so as to recommend to him videos similar to the ones he likes.

A Hybrid Recommender System for mobile devices goes beyond the issues related to context; rather it integrates other "dimensions", such as reputation, to render the recommendation more tailored to the user. Hybrid Recommender Systems leverage advantages from traditional recommenders, namely taking the benefits of RSs and Collaborative Filtering that are designed to work for desktop environments, and those of Context Aware applications that take into account the mobility of the user and his current situation and surrounding environment, so as to recommend services in a real time manner.

A pertinent study that falls in the class of Hybrid Recommender Systems is the *Dynamos* project [7]. The *Dynamos* approach is an example of a context-aware mobile application that can be used for recommending relevant services to the user. In [7], the authors designed a Hybrid Recommender System for notifying users about relevant services in a context-aware manner. The model is based on a peer-to-peer social functionality model, where the users can generate contextual notes and ratings, and attach them to services, or to the environments. They are also permitted to share these with their peers. The attached notes to the environment are delivered to other users whenever they are in the geographical vicinity of the entities associated with the notes. This research, however, assumed that the user was expected to explicitly describe his preferences by manually entering them. In this sense, the profile is defined by the user by explicitly specifying types of activities, and by associating multiple interests to them. Such an approach can be considered to be of a more "primitive" sort – it is not viable in pervasive environments where preferences change over time.

A concept akin to Hybrid Recommender System, is referred to as Mobile Multidimensional Recommender Systems [49]. The work reported in [49] provides a

Recommender System incorporating contextual information. Their model is multidimensional recommendation model that makes recommendations based on multiple dimensions and, therefore, extends the classical two-dimensional $<$Users, Items$>$ paradigm. An important research question involves determining how one can estimate unknown ratings in such multidimensional recommendation space.

An example to illustrate this concept is a smart shopping application, designed in [49], where the authors bridged the advances in Collaborative Filtering algorithms and contextual information (such as time, location within the store etc.) in order to recommend products in a real-time manner to the customer during his shopping experience. For example, a movie Recommender System would take into account the preferences deduced from the history of the movies rated by the user, as well as whether his companions (the companion of the user is a piece of the contextual information) are his parents or his friends in order to recommend a movie for the weekend.

The need of such pervasively intelligent recommendations is even more pertinent in the case of proactive Recommender Systems, where the information is pushed to the user causing interruptions. In this case, there is a benefit in having such a system where the reputation is taken into account so as to further reduce the cognitive load, and the distraction and irritability that can be caused by the risk of spamming.

A number of studies have been performed to design Hybrid Recommender Systems. A pioneering recent work was reported by Hossain *et al*. [6]. In this work, the authors proposed a gain-based media selection mechanism. In this regard, the gains obtained by ambient media services were estimated by combining the media's reputation, the user's context and the user's profile. As a result of such a modeling process, the service selection problem was formulated as a gain maximization problem. Thereafter, a combination of a dynamic and a greedy approach was used to solve the service selection problem. The authors of [6] did not present mechanisms to compute the reputation of the media services, thus, in effect, assuming that it is merely static. We argue that this assumption is not always valid, and that it is of paramount importance that the system tracks the variations in the reputation of the services since they, almost certainly, change over time.

In the results reported in [50], the authors presented a media content recommendation framework for smart phones that handles multiple dimensions, namely user's preferences, situation context and capability of the device in order to recommend media content to the user. The output of the recommendation is, interestingly, multidimensional such as format, frame rate, frame size, score, quality-of-service dimensions. A content-based approach was used to evaluate wether the user liked the media item or not in general. First, they used the content-based approach to measure the similarity between a media item and the preference of the user. The second step was to evaluate whether the media item was relevant in the user's current context. In order to perform the latter task, they adopted the Nave Bayes classifier approach to evaluate media items against the situation context while a rule-based approach was used to evaluate media items against the capability of the terminal. The items with the highest score were chosen to be presented to the user.

The *Daidalos* project [13] is an example of Hybrid Recommender System. The system is intended to operate"*everywhere*", and to provide access to services "*anytime*". This means that the "*users will constantly be exposed to services through different network-enabled channels. In such a world, service providers will be concerned mainly with getting users attention, and network providers will be concerned mainly with increased network use. Pervasive computing in* Daidalos *addresses an important piece of this puzzle supporting ordinary users.* Daidalos *presents a platform for personalized service delivery in pervasive environment*" [13]. The platform is composed of two layers which are tightly connected namely, the Service and Identity management layer, and the user experience management layer. The service and identity management is responsible for service discovery and composition as well as for the privacy-aware service access via the use of virtual identities, so as to protect the user's identity from being revealed to the service provider. The layer also performs a ranking of the discovered services as per the main parameters which are the context and the user's preferences. The choice of the service can be made by the user from the ranked list, or, in turn the *Daidalos* system can itself choose, for the user, the service possessing the highest ranking score. The user experience management layer includes two main components which are the Context management component and the Learning management component. The Context component is responsible for collecting contextual information. In fact, it collects raw contextual data and delivers inferred high level contextual data to the services and the applications requesting it. The second functionality involves maintaining an updated user profile through learning from the history of the interactions of the user with the services.

In the same vein, Moon Hee *et al.* [51], provided music recommendation considering the change of context as inferred from a combination of a fuzzy system, Bayesian networks, and utility theory.

2.2.4 Novelty of Information in Unobtrusive Applications

In this section, we will survey the state-of-the-art that relates to preserving user's attention while he is appraised of novel information. It is known that, we, as humans are interested in novel information that conveys new knowledge, and are simultaneously stressed and irritated by repetitive information.

Designing unobtrusive applications that are based on the concept of information novelty is an emerging research topic in information retrieval system and notification engines. Further, detecting novelty of information is a new frontier direction of research that, to the best of our knowledge, has not gained a significant amount of attention in the research community.

In the area of information retrieval systems, most of the legacy work relies solely on the user's profile in order to recommend services or information that match it to thus avoid overload of information. Few research studies take into account whether the information presented to the user as a result of the recommendation process is redundant. Since attention is an important resource, clearly, redundancy of information should be avoided.

The idea which we pursue is to not present the same information to the user unless the information entails novelty. The essence of this idea can be applied to information retrieval systems, such as document recommendation. The question is whether the content is similar to that possessed by previously observed documents. A whole body of research in document retrieval that incorporates the notion of novelty in text documents, for instance the work reported in [52] has employed detecting novelty in text documents for recommending novel blogs.

The research that is most closely related to novelty or redundancy detection in adaptive information filtering is perhaps the "First Story Detection" (FSD) task found in the research associated with Topic Detection and Tracking (TDT) [53]. A TDT system monitors a stream of chronologically-ordered documents, usually news stories. The FSD task is defined as detecting the first story that discusses a previously unknown event, where an event is defined as something that happens at some specific time and place. Online clustering approaches have been a common solution to the FSD task [54, 55, 56]. New stories are compared to clusters of stories about previously-known events. If the new story matches an existing cluster, it, supposedly, describes a known event; otherwise, it describes a new event.

One such system that includes traditional document filtering (for relevance) as well as a second stage novelty/redundancy detection is reported in [57]. Besides, novelty detection has previously been performed on text documents [58, 59] as well as to detect novel blog documents [52]. *Newsjunkie* [60] examines detecting novel information in a stream of news items dealing with the same story. *Newsjunkie* monitors a stream of news articles evolving over time on a common story, with the goal of highlighting truly informative updates, and of filtering out a large mass of articles that largely "relay more of the same". The authors of [60] used, for this purpose, a sliding window covering a number of preceding articles to estimate the novelty of the current one. Estimating distances between articles and those processed within preceding window of fixed-length, facilitates the comparison of scores.

In their paper on the nature of novelty detection, the authors [61] state that novelty detection proved helpful in personalized information filtering and that it was potentially helpful for other tasks that could return redundancies to the users. Suppressing repetitive notifications in the case of proactive-based applications based on Publish/Subscribe has been recognized to be a significant functionality. Among the few studies that dealt with this issue, we cite the work reported in [62] where the authors present a state persistence Publish/Subcrsribe model. In state-persistent models, notifications are delivered to the end-user only upon state transitions where the latter occurs only when a subject enters a given geographical region or leaves it. As opposed to this, stateless models create redundant notifications.

2.2.5 Learning Preferences

The problem of estimating the user's preferences is becoming increasingly essential for personalized applications which range from service recommender systems to the targeted advertising of services.

Utilizing the power of the Internet to affect marketing, business and politics *via* strategies applicable for social networking, is becoming increasingly important, especially in a user-driven universe. Over the last few years, the issue of maintaining users' profiles has become more crucial for designing and streamlining personalized applications ranging from service recommender systems to the advertising of targeted services.

Mastering and optimally utilizing the knowledge about a user's interests has led to promising applications in filtering and recommending documents [63], multimedia [6] and TV programs [8], based on their respective contents. For instance, a comprehensive study for personalized service provisioning was performed by Naudet *et al*. from Bell Labs [8], where the authors designed an application for filtering the TV content provided to users' mobile devices based on their learned profiles. The application is based on the use of ontologies to capture content descriptions as well as the users' interests. The latter interests are, in turn, mined using a dedicated profiling engine presented in [9], which leveraged Machine Learning (ML) techniques for user profiling. The work reported in [10] presented a "product" that recommends vendors' web pages by measuring the similarity between the user's profile and the vendor's web page when the user is in the vicinity of the vendor (seller). The user's profile is constructed through mining the history of his web log. Another example that falls in the class of mobility-aware applications is the *PLIGRIM* system, which makes use of the user's location to recommend relevant web links [11]. In the same vein, the *SMMART* framework dynamically locates products that match the shopping preferences of a mobile user [5].

Usually, constructing a user's profile involves applying estimation techniques to leverage the knowledge about his interests, which, in turn, is gleaned from the history of the services that he utilizes [6, 64]. A number of previous studies [65] have shown that a user's interests are not constant over time, and consequently, paradigms which are to be promising, should take into account the drift of these interests. The time varying nature of the distribution of the user's interests renders the problem of estimating them both difficult and non-trivial.

Thus, unlike traditional estimation problems where the underlying target distribution is stationary, estimating a user's interests, typically, involves non-stationary distributions. The consequent time varying nature of the distribution to be tracked imposes stringent constraints on the "*unlearning*" capabilities of the estimator used. Therefore, resorting to strong estimators that converge with probability 1 is inefficient since they rely on the assumption that the distribution of the user's preferences is stationary.

Tracking the dynamics of a user's interests is akin to a well-known problem in statistical *Pattern Recognition* (PR), namely that of estimating non-stationary distributions. Traditionally available methods that cope with non-stationary distributions resort to the so-called *sliding window* approach, which is a limited-time variant of the well-known Maximum Likelihood Estimation (MLE) scheme. The latter model is useful for discounting stale data in a stream of observations. Data samples arrive continually and only the most recent observations are used to compute the current estimates. Any data occurring outside the current window is forgotten and replaced by the new data. The problem with using sliding windows is the following: If the time window is too small the corresponding estimates tend to be poor. As opposed to this, if the time window is too large, the estimates prior to the change of the parameter have too much influence on the new estimates. Moreover, the observations during the entire window width must be maintained and updated during the process of estimation.

In earlier works [66, 67, 65], Koychev *et al*. introduced the concept of Gradual Forgetting (GF). The GF process relies on assigning weights that decrease over time to the observations. In this sense, the GF approach assigns most weight to the more recent observations, and a lower weight to the more-distant observations. Hence, the influence of old observations (on the running estimates) decreases with time. It was shown in [65] that GF can be an enhancement to the sliding window paradigm. In this sense, observations within each sliding window are weighted using a GF function.

Recently, Oommen and Rueda [14] proposed a strategy by which the parameters of a binomial/multinomial distribution can be estimated when the underlying distribution is non-stationary. The method is referred to as Stochastic Learning Weak Estimation (SLWE), and is based on the principles of stochastic Learning Automata (LA) [68, 69]. The SLWE has found successful applications in many real-life problems that involve estimating distributions in non-stationary environments such as in adaptive encoding [70], route selection in mobile ad-hoc networks [71], and topic detection and tracking in multilingual online discussions [72]. Motivated by these successful applications of the SLWE in various areas, in the course of this study, we consider employing the SLWE for solving the intriguing problem of tracking user's interests. One objective of the work (described later) is to present a personalized *Learning Preferences Manager*, which will involve the *modus operandus* for capturing user's preferences. The latter will be able to cope with changes brought about by variations in the distribution of the user's interests, which will be where the SLWE plays a prominent part.

The core function of a personalized *Learning Preferences Manager* is to update the user's profile in a dynamic and incremental way. This is done so that the Learning Preferences Manager can closely follow the real-time evolution of the user's interests. In fact, user's interests are in general not constant over time, and therefore it is imperative that the system takes the profile's drift into account. In this sense, whenever one attempts to represent the user's *current* interests, the most recent observations are more reliable than older ones. From a more general perspective, the task of learning the drifts in the user's interests corresponds to the problem of learning evolving concepts [73]. There are several studies that have dealt with the task of learning a user's interests. These include the use of a sliding window [74], aging examples [75], and a Gradual Forgetting (GF) function [66, 67, 65]. However, of all these, a sliding window approach is the most popular one. It consists of learning the description of the user's interests from the most recent observations, and thereafter, of discarding the observations that fall outside the window.

A substantial shortcoming of the sliding window approach is the choice of the window size. In [74], the authors adopted a fixed-size time window in order to learn a user's scheduling preferences. They empirically determined that a window size of 180 was a proper choice for their particular scheduling application. The GF, on the other hand, relies on assigning weights to the observations that decrease over time. Hence, the influence of older observations on the running estimates, decreases with time. The authors of [65] suggested a linearly-decreasing function, $w = f(t)$, for decaying the relative weights of the GF as follows:

$$w_i = \frac{-2k}{n-1}(i-1) + 1 + k, \tag{4}$$

where i denotes a counter of observations starting from the most recent one, n is the number of observations, $k \in [0,1]$ is a parameter that represents the percentage by which the weight of any subsequent observation is decreased, and consequently the percentage by which the weight of the most recent one, in comparison to the average, is increased. Thus k is a parameter that controls the slope of the forgetting function.

In order to achieve a synergy between the two approaches, namely GF and sliding window, Koychev in [65], proposed to apply the GF *within each sliding window*. Thus, in this case, the parameter n (i.e., the length of the observation sequence) in Equation (4) was set to be equal to L, where L denotes the length of the window.

Apart from the sliding window and GF schemes, other approaches, which also deal with *change detection*, have also emerged. In general, there are two major competitive sequential change-point detection algorithms: Page's cumulative sum (CUSUM) [76] detection procedure and the Shiryaev$-$Roberts$-$Pollak detection procedure. In [77], Shiryayev used a Bayesian approach to detect changes in the parameters distribution, where the change points were assumed to obey a geometric distribution. CUMSUM is motivated by a maximum likelihood ratio test for the hypotheses that a change occurred. Both approaches utilize the log-likelihood ratio for the hypotheses that the change occurred at the point, and that there is no change. Inherent limitations of CUMSUM and the Shiryaev$-$Roberts$-$Pollak approaches for on-line implementation are their demanding computational and memory requirements. In contrast to the CUMSU and the Shiryaev$-$Roberts$-$Pollak, the SLWE avoids the intensive computations of ratios, and do not invoke hypothesis testing.

A particularly interesting recent study for learning user's interests in ambient media services (and in, consequently, locating relevant services) was reported in

[64]. Hossain *et al.* devised the so-called Ambient Media Score Update method, which we shall refer to as SU for the rest of the thesis. The SU method was used to learn a user's changing interests [6, 64] by recording the so-called "*scores*", which represented his/her affinity of interests. In order to follow closely the evolution of the scores, the authors of [64] refined their proposed updating method defined earlier in [6] and updated the scores of the services at every time instant whenever the service was used. This was done instead of performing updates in a batch mode [6].

2.3 Learning Automata

The fundamental tool which we shall use in most of our research involves Learning Automata. Learning Automata[2] (LA) have been used in systems that have incomplete knowledge about the Environment in which they operate [79, 80, 81, 68, 82, 83, 84]. The learning mechanism attempts to learn from a *stochastic Teacher* which models the Environment. In his pioneering work, Tsetlin [85] attempted to use LA to model biological learning. In general, a random action is selected based on a probability vector, and these action probabilities are updated based on the observation of the Environment's response, after which the procedure is repeated.

The term "Learning Automata" was first publicized and rendered popular in the survey paper by Narendra and Thathachar. The goal of LA is to "determine the optimal action out of a set of allowable actions" [79]. The distinguishing characteristic of automata-based learning is that the search for the optimizing parameter vector is conducted in the space of probability distributions defined over the parameter space, rather than in the parameter space itself [86].

With regard to applications, the entire field of LA and stochastic learning has had a myriad of applications [80, 81, 68, 83, 84], which (apart from the many applications listed in these books) include solutions for problems in network and communications [87, 88, 89, 90], network call admission, traffic control, quality of

[2] A part of the review on LA included here is incorporated from a section of a preliminary version of a paper [78] published in this thesis, although the final published version did not include it for space consideration. The paper [78] was co-authored with Dr. Granmo and Dr. Oommen, where the LA review section in that paper is due to Dr. Oommen.

service routing, [91, 92, 93], distributed scheduling [94], training hidden Markov models [95], neural network adaptation [96], intelligent vehicle control [97], and even fairly theoretical problems such as graph partitioning [98]. Besides these fairly generic applications, with a little insight, LA can be used to assist in solving (by, indeed, learning the associated parameters) the stochastic resonance problem [99], the stochastic sampling problem in computer graphics [100], the problem of determining roads in aerial images by using geometric-stochastic models [101], and various location problems [102]. Similar learning solutions can also be used to analyze the stochastic properties of the random waypoint mobility model in wireless communication networks [103], to achieve spatial point pattern analysis codes for GISs [104], to digitally simulate wind field velocities [105], to interrogate the experimental measurements of global dynamics in magneto-mechanical oscillators [106], and to analyze spatial point patterns [107]. LA-based schemes have already been utilized to learn the best parameters for neural networks [96], optimizing QoS routing [93], and bus arbitration [88] – to mention a few other applications.

In the field of Automata Theory, an automaton [80, 81, 68, 83, 84] is defined as a quintuple composed of a set of states, a set of outputs or actions, an input, a function that maps the current state and input to the next state, and a function that maps a current state (and input) into the current output.

Definition 1: A LA is defined by a quintuple $\langle A, B, Q, F(.,.), G(.)\rangle$, where:

1. $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of outputs or actions that the LA must choose from, and $\alpha(t)$ is the action chosen by the automaton at any instant t.

2. $B = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is the set of inputs to the automaton. $\beta(t)$ is the input at any instant t. The set B can be finite or infinite. The most common LA input is $B = \{0, 1\}$, where $\beta = 0$ represents reward, and $\beta = 1$ represents penalty.

3. $Q = \{q_1, q_2, \ldots, q_s\}$ is the set of finite states, where $Q(t)$ denotes the state of the automaton at any instant t.

4. $F(.,.) : Q \times B \mapsto Q$ is a mapping in terms of the state and input at the instant t, such that, $q(t+1) = F[q(t), \beta(t)]$. It is called a *transition function*, i.e.,

a function that determines the state of the automaton at any subsequent time instant $t+1$. This mapping can either be deterministic or stochastic.

5. $G(.)$: is a mapping $G : Q \mapsto A$, and is called the *output function*. G determines the action taken by the automaton if it is in a given state as: $\alpha(t) = G[q(t)]$. With no loss of generality, G is deterministic.

If the sets Q, B and A are all finite, the automaton is said be *finite*.

The Environment, E, typically, refers to the medium in which the automaton functions. The Environment possesses all the external factors that affect the actions of the automaton. Mathematically, an Environment can be abstracted by a triple $\langle A, C, B \rangle$. A, C, and B are defined as follows:

1. $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the set of actions.

2. $B = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is the is the output set of the Environment. Again, we consider the case when $m = 2$, i.e., with $\beta = 0$ representing a "Reward", and $\beta = 1$ representing a "Penalty".

3. $C = \{c_1, c_2, \ldots, c_r\}$ is a set of penalty probabilities, where element $c_i \in C$ corresponds to an input action α_i.

The process of learning is based on a learning loop involving the two entities: the Random Environment (RE), and the LA, as described in Figure 2.3. In the process of learning, the LA continuously interacts with the Environment to process responses to its various actions (i.e., its choices). Finally, through sufficient interactions, the LA attempts to learn the optimal action offered by the RE. The actual process of learning is represented as a set of interactions between the RE and the LA.

The automaton is offered a set of actions, and it is constrained to choose one of them. When an action is chosen, the Environment gives out a response $\beta(t)$ at a time "t". The automaton is either penalized or rewarded with an unknown probability c_i or $1 - c_i$, respectively. On the basis of the response $\beta(t)$, the state of the automaton $\phi(t)$ is updated and a new action is chosen at (t+1). The penalty probability c_i satisfies:

$$c_i = Pr[\beta(t) = 1 | \alpha(t) = \alpha_i] \qquad (i = 1, 2, \ldots, R).$$

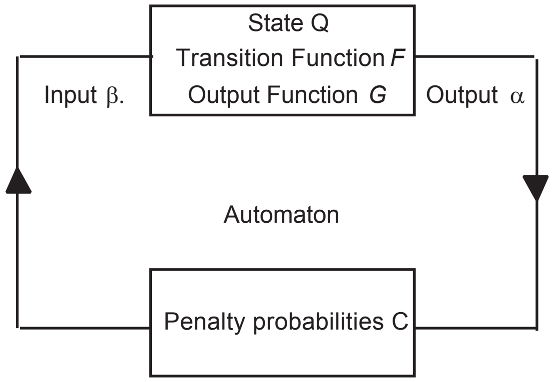


Figure 2.3: Feedback Loop of LA.

We now provide a few important definitions used in the field. $P(t)$ is referred to as the action probability vector, where, $P(t) = [p_1(t), p_2(t), \ldots, p_r(t)]^T$, in which each element of the vector.

$$p_i(t) = Pr[\alpha(t) = \alpha_i],\ i = 1, \ldots, r, \text{ such that } \sum_{i=1}^{r} p_i(t) = 1 \ \ \forall t. \tag{5}$$

Given an action probability vector, $P(t)$ at time t, the *average penalty* is:

$$\begin{aligned} M(t) &= E[\beta(t)|P(t)] = Pr[\beta(t) = 1|P(t)] \\ &= \sum_{i=1}^{r} Pr[\beta(t) = 1|\alpha(t) = \alpha_i]\ Pr[\alpha(t) = \alpha_i] \\ &= \sum_{i=1}^{r} c_i p_i(t). \end{aligned} \tag{6}$$

The average penalty for the "pure-chance" automaton is given by:

$$M_0 = \frac{1}{r} \sum_{i=1}^{r} c_i. \tag{7}$$

As $t \mapsto \infty$, if the average penalty $M(t) < M_0$, at least asymptotically, the automaton is generally considered to be better than the pure-chance automaton. $E[M(t)]$ is given by:

$$E[M(t)] = E\{E[\beta(t)|P(t)]\} = E[\beta(t)]. \tag{8}$$

A LA that performs better than by pure-chance is said to be *expedient*.

Definition 2: A LA is considered *expedient* if:

$$\lim_{t \mapsto \infty} E[M(t)] < M_0.$$

Definition 3: A LA is said to be *absolutely expedient* if $E[M(t+1)|P(t)] < M(t)$, implying that $E[M(t+1)] < E[M(t)]$.

Definition 4: A LA is considered *optimal* if $\lim_{t \mapsto \infty} E[M(t)] = c_l$,where $c_l = \min_i\{c_i\}$.

It should be noted that no optimal LA exist. Marginally sub-optimal performance, also termed above as ε-optimal performance, is what LA researchers attempt to attain. **Definition 5:** A LA is considered *ε-optimal* if:

$$\lim_{n \mapsto \infty} E[M(t)] < c_l + \varepsilon, \tag{9}$$

where $\varepsilon > 0$, and can be arbitrarily small, by a suitable choice of some parameter of the LA.

2.3.1 Classification of Learning Automata

2.3.1.1 Deterministic Learning Automata

An automaton is termed as a *deterministic automaton*, if both the transition function $F(.,.)$ and the output function $G(.)$ are deterministic. Thus, in a deterministic automaton, the subsequent state and action can be uniquely specified, provided the present state and input are given.

2.3.1.2 Stochastic Learning Automata

If, however, either the transition function $F(.,.)$, or the output function $G(.)$ are stochastic, the automaton is termed to be a *stochastic automaton*. In such an automaton, if the current state and input are specified, the subsequent states and actions cannot be specified uniquely. In such a case, $F(.,.)$ only provides the probabilities of reaching the various states from a given state.

In the first LA designs, the transition and the output functions were time invariant, and for this reason these LA were considered "Fixed Structure Stochastic Automata" (FSSA). Tsetlin, Krylov, and Krinsky [85] presented notable examples of this type of automata.

Later, Vorontsova and Varshavskii introduced a class of stochastic automata known in the literature as Variable Structure Stochastic Automata (VSSA). In the definition of a VSSA, the LA is completely defined by a set of actions (one of which is the output of the automaton), a set of inputs (which is usually the response of the Environment) and a learning algorithm, T. The learning algorithm [68] operates on a vector (called *the Action Probability vector*).

Note that the algorithm $T : [0,1]^R \times \mathrm{A} \times \mathrm{B} \rightarrow [0,1]^R$ is an updating scheme where A = $\{\alpha_1, \alpha_2, \ldots, \alpha_R\}$, $2 \leq \mathrm{R} < \infty$, is the set of output actions of the automaton, and B is the set of responses from the Environment. Thus, the updating is such that

P(t+1) = T(P(t), α(t), β(t)),

where P(t) is the action probability vector, α(t) is the action chosen at time t, and β(t) is the response it has obtained.

If the mapping T is chosen in such a manner that the Markov process has absorbing states, the algorithm is referred to as an absorbing algorithm. Many families of VSSA that posses absorbing barriers have been reported [68]. Ergodic VSSA have also been investigated [68, 15]. These VSSA converge in distribution and thus, the asymptotic distribution of the action probability vector has a value that is independent of the corresponding initial vector. While ergodic VSSA are suitable for non-stationary environments, absorbing VSSA are preferred in stationary environments.

Each distinct updating scheme, T, identifies a different type of learning algorithm, as follows:

- *Linear algorithms* are the ones in which $P(t+1)$ is a linear function of $P(t)$.

- *Nonlinear algorithms* are the ones in which $P(t+1)$ is a non-linear function of $P(t)$.

In a VSSA, if a chosen action α_i is rewarded, the probability for the current action is increased, and the probabilities for all the other actions are decreased. On the other hand, if the chosen action α_i is penalized, the probability of the current action is decreased, whereas the probabilities for the rest of the actions could, typically, be increased. This leads to the following different types of learning schemes for VSSA:

- **Reward-Penalty (RP):** In both the cases, when the automaton is rewarded as well as penalized, the action probabilities are updated.

- **Inaction-Penalty (IP):** When the automaton is penalized, the action probability vector is updated, whereas when the automaton is rewarded, the action probabilities are neither increased nor decreased.

- **Reward-Inaction (RI):** The action probability vector is updated whenever the automaton is rewarded, and is unchanged whenever the automaton is penalized.

A LA is considered to be a *continuous automaton* if the probability updating scheme T is continuous, i.e., the probability of choosing an action can be any real number in the closed interval $[0,1]$.

In a VSSA, if there are r actions operating in a stationary environment with $\beta = \{0,1\}$, a general action probability updating scheme for a continuous automaton is described below. We assume that the action α_i is chosen, and thus, $\alpha(t) = \alpha_i$. The updated action probabilities can be specified as:

$$\begin{aligned} \text{For } \beta(t) &= 0, \forall j \neq i\,, \quad p_j(t+1) = p_j(t) - g_j(P(t)) \qquad (10) \\ \text{For } \beta(t) &= 1, \forall j \neq i\,, \quad p_j(t+1) = p_j(t) + h_j(P(t)) \end{aligned}$$

Since $P(t)$ is a probability vector, $\sum_{j=1}^{r} p_j(t) = 1$. Therefore,

$$\begin{aligned} \text{When } \beta(t) &= 0, \quad p_i(t+1) = p_i(t) + \sum_{j=1, j\neq i}^{r} g_j(P(t)), \qquad (11) \\ \text{and when } \beta(t) &= 1, \quad p_i(t+1) = p_i(t) - \sum_{j=1, j\neq i}^{r} h_j(P(t)). \end{aligned}$$

The functions h_j and g_j are *nonnegative* and *continuous* in $[0,1]$, and obey:

$$\forall i = 1,2,\ldots,r, \quad \forall P \in (0,1)^R,\ 0 < g_j(P) < p_j, \tag{12}$$
$$\text{and} \qquad 0 < \sum_{j=1, j\neq i}^{r} [p_j + h_j(P)] < 1.$$

For *continuous linear* VSSA, the following four learning schemes are extensively studied in the literature. They are explained for the 2-action case; their extension to the r-action case, where $r > 2$, are straightforward, and can be found in [68].

For a 2-action LA, let

$$\begin{aligned} g_i(P(t)) &= a\ p_j(t) \\ \text{and } h_j(P(t)) &= b\ (1 - p_j(t)) \end{aligned} \tag{13}$$

In Equation (13), a and b are called the reward and penalty parameters, and they obey the following inequalities: $0 < a < 1$, $0 \leq b < 1$. Equation (13) will be used further to develop the action probability updating equations.

- The Linear Reward-Inaction Scheme (L_{RI}): In the case, $a > 0$ and $b = 0$. The L_{RI} scheme is ε-optimal as $a \mapsto 0$. The two actions L_{RI} scheme has the vectors $[1,0]^T$ and $[0,1]^T$ as two absorbing states. Indeed, with probability 1, it gets absorbed into one of these absorbing states. The L_{RI} scheme is both absolutely expedient, and ε-optimal [68].

- The Linear Inaction-Penalty Scheme (L_{IP}): It is characterized by $a = 0, b > 0$. L_{IP} is an ergodic scheme.

- The Symmetric Linear Reward-Penalty Scheme (L_{RP}): $a = b; a, b > 0$. L_{RP} is Ergodic and is, at best, expedient.

- The Linear Reward-ε-Penalty Scheme ($L_{R-\varepsilon P}$): $a > 0, b << a$. $L_{R-\varepsilon P}$ is ε-optimal as $a \mapsto 0$. In addition it is Ergodic. The $L_{R-\varepsilon P}$ is adequate for non-statinary environments.

2.3.2 Discretized Learning Automata

In practice, the relatively slow rate of convergence of these algorithms constituted a limiting factor in their applicability. In order to increase their speed of convergence, the concept of discretizing the probability space was introduced [15, 108]. This concept is implemented by restricting the probability of choosing an action to a finite number of values in the interval [0,1]. If the values allowed are equally spaced in this interval, the discretization is said to be linear, otherwise, the discretization is called non-linear. Following the discretization concept, many of the continuous VSSA have been discretized; indeed, discrete versions of almost all continuous automata have been reported [15].

2.3.3 Estimator Algorithms

Pursuit and Estimator-based LA were introduced to be faster schemes, characterized by the fact that they pursue what can be reckoned to be the *current* optimal action or the set of current optimal schemes [15]. The updating algorithm improves its convergence results by using the history to maintain an estimate of the probability of each action being rewarded, in what is called the *reward-estimate* vector. While, in non-estimator algorithms, the action probability vector is updated solely on the basis of the Environment's response, in a Pursuit or Estimator-based LA, the update is based on *both* the Environment's response and the *reward-estimate* vector. Families of Pursuit and Estimator-based LA have been shown to be faster than VSSA [86]. Indeed, even faster discretized versions of these schemes have been reported [79, 15].

2.3.4 Object Migrating Automaton (OMA)

As documented in the literature, the object partitioning problem involves partitioning a set of $|\mathbb{P}|$ objects into $|\mathbb{N}|$ groups or classes, where the main aim is to partition the objects into groups that mimic an underlying unknown grouping. In other words, the objects which are accessed together must reside in the same group [109]. In the special case when all the groups are required to contain the same number of objects, the problem is also referred to as the Equi-Partitioning Problem (EPP).

Many solutions involving *LA* have been proposed to solve the *EPP*, but the most efficient algorithm is the *Object Migrating Automaton* (*OMA*) [109]. The latter was first proposed by Oommen and Ma [109], and some modifications were added by Gale *et.al.* [110] to create the *Adaptive Clustering Algorithm* (*ACA*).

The Object Migrating Automaton (*OMA*) is an ergodic automaton that has R actions $\{\alpha_1, \ldots, \alpha_R\}$ representing the possible underlying classes. Each action α_i has its own set of states $\{\phi_{i1}, \phi_{i2} \ldots, \phi_{iM}\}$, where M is the depth of memory, and $1 \leq i \leq R$ represents the number of classes. ϕ_{i1} is called the most internal state and ϕ_{iM} is the boundary (or most external) state.

A set of W physical objects $\{A_1, A_2, \ldots, A_W\}$ is accessed by a random stream of queries, and the objects are to be partitioned into groups so that the frequently *jointly*-accessed objects are clustered together. The *OMA* utilizes W abstract objects $\{O_1, O_2, \ldots, O_W\}$ instead of migrating the physical objects. Each abstract object is assigned to a state belonging to an initial random group but in its boundary state. The objects within the automaton move from one action to another, and so, in this case, all the W abstract objects move around in the automaton. If the abstract objects O_i and O_j are in the action α_h, and the request accesses $< A_i, A_j >$, then the *OMA* will be rewarded by moving them towards the most internal state ϕ_{h1}. But a penalty arises if the abstract objects O_i and O_j are in different classes, say α_h and α_g, respectively. Assuming O_i is in $\zeta_i \; \varepsilon \; \{\phi_{h1}, \phi_{h2}, \ldots, \phi_{hM}\}$ and O_j is in $\zeta_j \; \varepsilon \; \{\phi_{g1}, \phi_{g2}, \ldots, \phi_{gM}\}$, they will be moved as follows:

- If $\zeta_i \neq \phi_{hM}$ and $\zeta_j \neq \phi_{gM}$, O_i and O_j are moved one state toward ϕ_{hM} and ϕ_{gM}, respectively.

- If exactly one of them is in the boundary state, the object which is not in the boundary state is moved towards *its* boundary state.

- If both of them are in their boundary states, one of them, say O_i is moved to the boundary state of the other object ϕ_{gM}. In addition, the closest object to them is moved to the boundary state ϕ_{hM}, so as to preserve an equal number of objects in each group.

It is important to point out that the random stream of queries contains information about an optimal partition, and the *OMA* attempts to converge to it. The automaton is said to have converged when all the objects in a class are in the deepest (or second deepest) most-internal state.

The *OMA* can be improved by the following: Assume that a pair of objects $< A_i, A_j >$ is accessed, where O_i is in the boundary state, while O_j is in a non-boundary state. In this case, a general check should be made to locate another object in the boundary state of the partition containing O_j. If there is an object, then swapping is done between this object and O_i in order to bring the two accessed objects into the same partition. In turn, instead of waiting for a long time to have these accessed objects in the same partition, the convergence speed can be increased by swapping the objects into the right partitions.

The formal algorithm for the *OMA* can be found in [110, 109]; it is shown in Algorithm 1.

Algorithm 1 Enhanced *OMA* [110]

Input: The abstract set of objects, a number of state per action, a sequence of random queries in form (O_i, O_j)

Output: A periodic clustering of the objects into R partitions

Notation: ζ_i is the state of the abstract object O_j. It is an integer in the range1$\ldots RN$, where, if $(h-1)N+1 \leq \zeta_i \leq hN$, then the object O_i is assigned to α_h.

Method:

Initialize ζ_i for $1 \leq i \leq W$ randomly among the boundary state of classes, each class having W/R objects.
for a sequence of T queries **do**
    Read query (A_i, A_j)
    if ((ζ_i div N)=(ζ_j div N)) **then**
        if (ζ_i mod N $\neq$ 1) **then**
            $\zeta_i = \zeta_i - 1$
        end if
        if (ζ_j mod N $\neq$ 1) **then**
            $\zeta_j = \zeta_j - 1$
        end if
    else
        if (((ζ_i mod N)$\neq$ 0) and ((ζ_j mod N)$\neq$ 0)) **then**
            $\zeta_i = \zeta_i + 1$
            $\zeta_j = \zeta_j + 1$
        else if (ζ_i mod N $\neq$ 0) **then**
            if (O_v: unaccessed object in group of O_i where ζ_v mod $N = 0$) **then**
                temp $=$ ζ_j
                $\zeta_j = \zeta_v$
                $\zeta_v =$ temp
            end if
            $\zeta_i = \zeta_i + 1$
        else if (ζ_j mod N $\neq$ 0) **then**
            if (O_v: unaccessed object in group of O_j where ζ_v mod $N = 0$) **then**
                temp $=$ ζ_i
                $\zeta_i = \zeta_v$
                $\zeta_v =$ temp
            end if
            $\zeta_j = \zeta_j + 1$
        else
            $temp = \zeta_i$
            $\zeta_i = \zeta_j$
            t = index of an unaccessed object in group of O_j where O_t is closest to ζ_j
            $\zeta_t = temp$
        end if
    end if
end for
return Partitions based on the states $\{\zeta_i\}$

End Algorithm Enhanced *OMA*

Chapter 3

Contributions

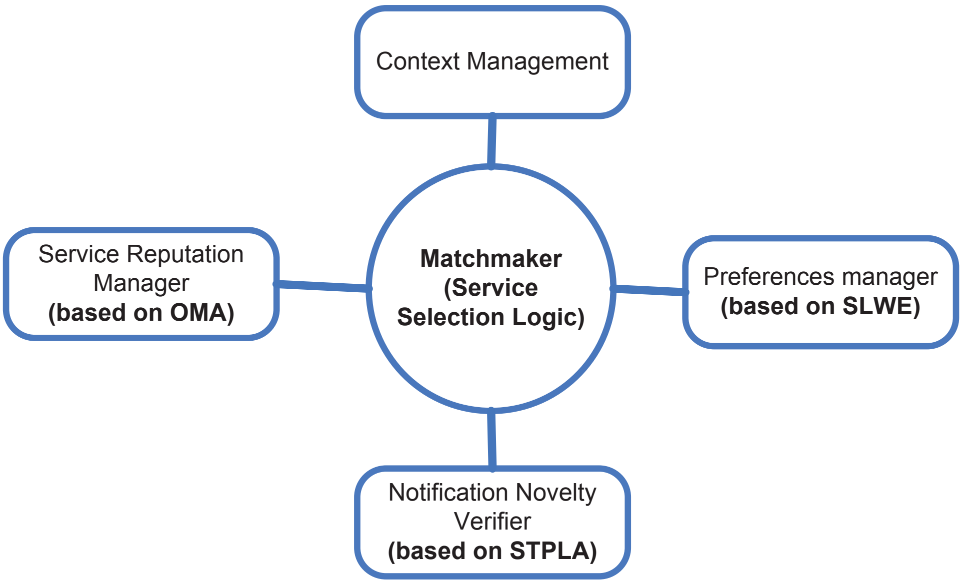


Figure 3.1: A simplified panoramic view of the modules of the architecture with reference to the LA tools that we have employed.

The figure above describes the main components of the architecture with reference to the LA tools that we have employed in the thesis. To be more specific, the Service Reputation Manager is based on the theory of OMA, and it has been described in detail in Appendix A. The Preferences Manager makes use of the theory of the SLWE, and its implementation details are documented in Appendix D. The Notification Novelty Verifier was introduced in Appendix C as a part of the overall architecture of the system. It is based on the theory of STPLA which was formulated earlier in Appendix B.

3.1 Contributions in Reputation Systems

3.1.1 Overview of the Contributions

As explained in the introductory chapter, one of the aims of this thesis is to design an unobtrusive User-Centric Service Provisioning System. In order to save the user's attention and to maximize the usefulness of the services accessed, the user needs to build his opinion about these services in the absence of any direct experience, and as a consequence, must rely on the experiences of his acquaintances.

In Appendix A, we present a paper which explains how we have designed a Service Reputation Manager for selecting high quality service providers, which is a cornerstone component of our architecture.

In the absence of direct experience, a user can rank the quality of the available services by optimizing the power of Word-of-Mouth communications. By allowing users to share their feedback in the form of ratings, it is possible for them to expediently obtain knowledge about the nature, quality and drawbacks of specific services by considering the experiences of other users. Traditional RSs, usually compute the reputation of a service as the average of all provided ratings. This corresponds, for instance, with the percentage of positive ratings in the *eBay* feedback form [111]. Such a simplistic approach of just blindly aggregating users' experiences may mislead the RS if some of the user's acquaintances are misinformed/deceptive users. Misinformed/deceptive users attempt to collectively subvert the system by providing either unfair positive ratings about a service, or by unfairly submitting negative ratings.

From this perspective, we can state a fundamental weakness about using RSs by virtue of the fact that they are prone to "ballot stuffing" and "badmouthing" in a competitive marketplace. Users who want to promote a particular product or service can flood the domain (i.e., the social network) with sympathetic votes, while those who want to get a competitive edge over a specific product or service can "badmouth" it unfairly. Thus, although these systems can offer generic recommendations by aggregating user-provided opinions, unfair ratings may degrade the trustworthiness of such systems. Additionally, changes in the quality of service, over time, can render previous ratings unreliable. In general, unfair ratings may degrade the trustworthiness of RSs, and changes in the quality of service, over time, can render previous ratings unreliable.

It is reasonable to assume that the acquaintances of the user can be divided into two classes: trustworthy acquaintances that provide accurate ratings, and unreliable acquaintances that provide unfair ratings. It follows that a good reputation manager component would seek to classify the acquaintances into one of these two classes so as to counter the detrimental effect of unfair ratings. In the paper found in Appendix A, we show how we have developed a Service Reputation Manager which is based on a concept analogous to collaborative filtering in order to separate between these two classes. The premise of the scheme was to separate the users' types by observing how they rate the same services. The latter scheme was designed in such a way that "similar" users would be in the same group by maximizing the "within-group" similarities and minimizing the "between-group" similarities.

In Appendix A, the latter problem was formulated in terms of the so-called Agent-Type Partitioning Problem, or $ATTP$ in brief. Further, we also formulated an algorithm to solve the ATTP. The aim of the solution to the $ATTP$ is to incrementally partition the users as being true/fair or deceptive, this is being done concurrently with their experiences being communicated. Thus, our scheme can be divided into two interleaving phases: A Partitioning Phase and a Service Selection Phase. The inputs to the Agent Partitioning phase are the reports communicated by the other agents, while the input to the Service Selection phase is the current partitioning of the agent. Decisions about whether the service is reliable or not are the output of the overall procedure. The decision is based on intelligently combining the feedback of deceptive and fair users. As the learning proceeds, we show that we can exclusively rely on the feedback from the fair users.

By suitably modeling reports about direct experiences involving a specific service as responses from the corresponding "Environment", our scheme intelligently groups agents according to the rating that they give to the same service. To formalize these responses, we define an agent to be "fair" (or "trustworthy") if it reports the service performance correctly with a probability $p > 0.5$. Similarly, an agent is said to be "deceptive" if it reports the inverted service performance with a probability $q > 0.5$. The beauty of our scheme is that although the identity of the reporting agents is unknown, "fair" agents will end up in the same group, while "deceptive" agents will converge to another group. In terms of LA terminology, in this case, the input to the environment is the ratings of the agents relative to the same services, while the action is a choice of the partition to which each agent belongs.

3.1.2 Related Work

Finding ways to solve the $ATTP$ and thus counter the detrimental influence of unfair ratings on a RS has been a focal concern of a number of studies [29, 27, 112, 30, 26, 113]. Dellarocas [27] used elements from collaborative filtering to determine the nearest neighbors of an agent that exhibited similar ratings on commonly-rated subjects. He then applied a cluster filtering approach to filter out the most likely unfairly positive ratings. However, his approach handles only the case of extremely positive ratings (i.e, "ballot stuffing") while it does not tackle the case of extremely negative ratings (i.e, "badmouthing").

Sen and Sajja [30] proposed an algorithm to select a service provider to process a task by querying other user agents about their ratings of the available service providers. The main idea motivating their work is to select a subset of agents, who when queried, gives a minimum probabilistic guarantee that the majority of the queried agents provide correct reputation estimates. However, comprehensive experimental tests show that their approach is prone to the variation of the ratio of deceptive agents.

Buchegger and Le Boudec [29] tackled the latter issue as follows: They proposed a Bayesian reputation mechanism in which each node isolates malicious nodes by applying a deviation test methodology. Their solution requires the agent to have enough *direct* experience with the services so that he can evaluate the trustworthiness of the reports of the witnesses. While this is a desirable option, unfortunately, in real life, such an assumption does not always hold, specially when the number of possible services is large.

In [114], Chen and Singh evaluated the quality of feedbacks assuming that a feedback is credible if it is consistent with the majority of feedbacks for a given user. Their approach, though promising, unfortunately, suffers from a deterioration in the performance when the ratio of deceptive agents is high.

In [26], Yu and Singh devised a modified weighted majority algorithm to combine reports from several witnesses to determine the ratings of another agent. The main shortcoming of the work reported in [26] is its relatively slow rate of convergence.

In contrast, in [31], Witby and Jøsang presented a Bayesian approach to filter out dishonest feedback based on an iterated filtering approach. In their approach, the authors extended the so-called "Beta" RS presented by Jøsang and Ismail [17].

The authors of [32] proposed a probabilistic model to assess peer trustworthiness in P2P networks. Their model, which, in one sense, is similar to ours, differs from our work because the authors of [32] assume that a peer can deduce the trustworthiness of other peers by comparing its own performance with reports of other peers about itself. Though such an assumption permits a feedback-evaluating mechanism, it is based on the fact that peers provide services to one another, thus permitting every party the right to play the role of a service provider and the service consumer (a reporting agent). Our approach makes a clear distinction between these parties – the service provider and the reporting agent.

Regan *et al.* [115, 116] tackled the problem of RSs by responding to the behavior of deceptive agents. The main idea of their work is that an agent can learn, over time, the trustworthiness of the advices of other agents using a Bayesian framework. Nevertheless, such an approach supposes that an agent has sufficient experience with other agents so that he can determine its trustworthiness. In this direction, the approach proposed by Regan *et al.* bears a marked similarity to the "slow learning" property of the trustworthiness of the recommending agents as in the Weighted Majority Approach [26].

3.1.3 Summary of our Contributions in RSs

In Appendix A, we proposed to solve the problem using tools provided by Learning Automata (LA), which have proven properties capable of learning the optimal action when operating in unknown stochastic environments. Furthermore, they combine rapid and accurate convergence with low computational complexity. In addition to its computational simplicity, unlike most reported approaches, our scheme does not require prior knowledge of the *degree* of any of the above mentioned problems associated with RSs. Instead, it gradually learns the identity and characteristics of the users which provide fair ratings, and of those who provide unfair ratings, even when these are a consequence of them making unintentional mistakes.

Comprehensive empirical results show that our LA-based scheme efficiently handles any degree of unfair ratings (as long as these ratings are binary). Furthermore, if the quality of services and/or the trustworthiness of the users change, our scheme is able to robustly track such changes over time. In addition, the scheme is ideal for decentralized processing. Accordingly, we believe that our LA-based scheme forms a promising basis for improving the performance of RSs in general.

Apart from the above, we demonstrate the applicability of LA to RSs – thus providing a promising real-time solution to the service selection problem. To our knowledge, our work documented in Appendix A, presents the first reported LA-based solution for any problem within the field of RSs. Since our solution is based on LA, it is both computationally simple and memory efficient. Regarding computation (Please see [Appendix A, Fig. 14], [Appendix A, Fig. 15] and [Appendix A, Fig. 16]) one can observe a mere linear increase in the computation as a function of the underlying internal parameters of the scheme. Regarding memory, ([Appendix A, Fig. 11]) we see that the scheme obtains a near-optimal performance with a memory size as small as 5. This result demonstrates that our scheme is memory efficient.

With regard to the field of LA itself, our scheme presents twofold contributions to traditional LA-based partitioning algorithms [117, 118, 119]. First of all, we do not impose the constraint that an equal number of agents must reside in the same partition. Secondly, we experimentally demonstrate that the partitioning still yields accurate results when the environment is stochastic. In this sense, the parameter used here to decide whether two agents are reckoned "similar" is stochastic, while the latter parameter was assumed *constant* by the authors of [117, 118, 119].

Most importantly, however, our scheme maximizes the likelihood of selecting high quality services in the presence of an *unknown* ratio of deceptive agents. Indeed, not only does the scheme not require the *a priori* knowledge of the ratio of the deceptive agents, but it is also very robust to extremely high ratios of such deceptive agents!

From Appendix A (see Fig. 6) we can observe how "immune" our system is to the percentage of deceptive agents. The simulations results demonstrate that the scheme is truly "immune" to varying the proportions of fair and deceptive agents. In fact, even if all agents are deceptive, the average performance is stable and again achieves near-optimal values that approach the index of the high performance services. In our opinion, this is quite remarkable!

In addition, an experimental comparison with other approaches have been performed. We have included a comparative evaluation with other popular approaches for dealing with deceptive agents ratings, namely, with Yu and Singh's weighted majority method [26], and Sen and Sajja's reinforcement learning approach [30].

In Appendix A, we remark from studying Fig. 12, that our proposed approach exhibits a faster convergence speed than the weighted majority algorithm. The slow convergence speed of the weighted majority algorithm can be explained by the gradual reduction of the weight of deceptive agents. This gradual modification of weights leads to the weighted average being "polluted" by the deceptive agents for a considerable amount of time, before their detrimental effect is filtered out.

Further from Fig. 13 of Appendix A, we observe that Sen and Sajja's algorithm suffers from a performance decline as the number of deceptive agents increases. As opposed to this, the average performance of our scheme remains near-optimal in every setting. This concludes our report on the contributions of this thesis in the area of Reputation Systems.

3.2 Contributions in On-line Discovery and Tracking of Spatio-Temporal Event Patterns

3.2.1 Overview of the Contributions

From our literature survey, we state that designing unobtrusive application that are based on the concept of information novelty is an emerging research topic in information retrieval system and notification engines. In fact, an excessive number of event notifications can quickly render the functionality of event-sharing to be obtrusive. Rather, any notification of events that provides redundant information to the application/user can be seen to be an unnecessary distraction.

In Appendix B, we present a paper that we have published in this regard. In that paper, we introduce a new scheme for discovering and tracking noisy spatio-temporal event patterns, with the purpose of suppressing reoccurring patterns, while discerning novel events. A scenario that we resort to is *Presence Sharing*. *Presence Sharing* is a ubiquitous service in which distributed mobile devices periodically broadcast their identity via short-range wireless technology such as BlueTooth or WiFi [120]. Applications that utilize *Presence Sharing* have been used in social contexts to maintain an "in touch" feeling strengthening social relations [121], as well as in work environments to enhance collaboration between colleagues [122]. It is worth noting that the contribution in question constitutes the basis for the Notification Novelty Checker component of our unobtrusive architecture presented in greater detail in Appendix C.

The solution we propose is to try to discern the nature of the events encountered. In fact, if we can discern that an event is repeating (even though this repetition is non-periodic), then it must be given less weight, while non-repeating events must be assigned a greater weight. In order to detect novel events we devise a scheme that is based on maintaining a collection of hypotheses, each one conjecturing a specific spatio-temporal event pattern. A dedicated LA – the *Spatio-Temporal Pattern LA* (STPLA) – is associated with each hypothesis. The STPLA decides whether its corresponding hypothesis is true by observing events as they unfold, processing evidence for and/or against the correctness of the hypothesis. In this case, as far as the field of LA is concerned, the action is one of doing a Suppress or Notify task. The input to the LA is a stream of events, namely the encounters. The response of the environment is a Penalty or Reward according to whether the encounter could be predicted or not by an hypothesized spatio-temporal pattern.

3.2.2 Related Work

A number of earlier studies have investigated various techniques for discovering the periodicity of time patterns, such as the episode discovery algorithm found in [123]. However, episode discovery, and other related approaches, suffer from the limitation that they assume unperturbed patterns that exhibit an *exact* periodicity. Unfortunately, the real-life unfolding of events is typically noise ridden. On one hand, regular events may get cancelled, introducing what we define as *omission noise*, and on the other, events may arise spontaneously and unexpectedly, without being part of a periodic pattern, introducing *inclusion noise*.

A pioneering work which was reported in [124], introduced the concept of *off-line* mining of partially periodic events. Partially periodic events are characterized by irregular periodicity that is disrupted by noise. Even though the concept of locating partially periodic events is common in many real-life applications, few research studies have been reported in this direction. In [124], the authors introduced a chi-squared test for discovering partially periodic patterns. Another interesting study on mining partially periodic patterns has been recently reported in [125]. In [125, 126], the Frequent and Periodic Activity Miner (FPAM) algorithm was introduced for finding repetitive patterns in a resident's activities in smart-environment applications. FPAM is essentially an *off-line* algorithm. However, in order to be able to detect changes in the patterns of the resident activities, FPAM was applied at scheduled regular mining sessions [125, 126]. A main shortcoming of the approaches discussed in [124, 125, 126] is that they operate in an *off-line* fashion. This is in contrast to our application where deciding whether to suppress event notifications is achieved "instantaneously", in an *on-line* manner, even as the events are unfolding. Indeed, we argue that any realistic scheme should discover and adapt to patterns as they appear and evolve in an *on-line* manner, without relying on extensive *off-line* data mining.

3.2.3 Summary of the Contributions

To the best of our knowledge, in the paper included in Appendix B, we present the first reported *on-line* approach for discovering and tracking of spatio-temporal patterns in noisy sequences of events. As mentioned above, a pioneering work which was reported in [124], introduced the concept of *off-line* mining of partially periodic events. Extensive simulations results confirm that our scheme outperforms a this state-of-the-art scheme, namely the FPAM [125]. The robustness of the STPLA to inclusion and as well as to omission noise, constitutes a unique property when it is compared to the FPAM.

Moreover, the simulations results reported in Appendix B show that our scheme possesses an excellent ability to cope with non-stationary environments. Interestingly, we found that using a Balanced memory STPLA with memory depth as small as 5 yields quite good results in *almost every* environment setting! Therefore, the question of determining (or tuning) the proper internal configuration of the STPLA does not constitute a critical issue to ensure adaptivity in dynamic environments. This is in contrast to the "modified" FPAM presented in [127], where the performance is dependent on a judicious choice of the size of the sliding window.

Regarding computation and memory requirements, our scheme is based on a *team* of finite automata, rendering it computationally efficient with a minimal memory footprint.

To conclude, in the paper included in Appendix B, we present the first reported approach to suppressing redundant notifications in applications related to pervasive environments. We believe that our work paves the way towards more research avenues for devising unobtrusive application by making use of the potential offered by machine learning techniques. The usefulness and feasibility of our application was demonstrated through a working prototype.

3.3 Contributions in Designing User-centric Architecture for Personalized Service Provisioning in Pervasive Environments

3.3.1 Overview of the Contributions

Throughout this thesis, we have argued and demonstrated that identifying those services that deserve the attention of the user is becoming an increasingly challenging task [1]. To respond to this challenge, in the paper included in Appendix C, we argue that service recommendation should rely on a multi-criteria decision maker that combines different aspects (dimensions) of the system/environment in order to decide, on behalf of the user, whether a service is relevant or not. "Relevance", we propose in this paper should be determined based on a user-centric approach that collectively combines the reputation of the service, the user's current context, the user's profile, as well as a record of the history of recommendations. The decision making mechanism should also be adaptive in the sense that it is able to cope with users' contexts that are changing, and the drifts in the users' interests, while it simultaneously can track the reputations of services, and suppress repetitive notifications based on the history of the recommendations.

In the paper presented in Appendix C, we present an instantiation of our architecture for a real-life, day-to-day scenario involving a proactive location-based application which provides an ensemble of services. In the scenario, the goal is to build a personalized and context-aware decision maker that delivers narrowly-targeted notifications to the user about relevant services in his environment.

3.3.2 Related Work

The rich availability of services in pervasive environments has the effect of overburdening the system's service selection task. From the perspective of Pervasive Computing promoted by Mark Weiser, the intention of incorporating more advanced

[1] The reader will observe that in this section, we will repeat a few portions of text from the previous chapter. This is done in the interest of permitting the chapter to be a stand-alone unit, and to avoid having the reader go back and forth. We believe that this is the ideal way to present this material.

technology should be that it provides the user the possibility of operating in a *calm* frame of mind [34]. Filtering out irrelevant information has been a focal concern in a number of studies. The main issue has been to reduce the cognitive load on the user when it comes to selecting services. It is well known that "pushing" (or downloading) notifications messages to users can cause interruptions and distractions. Users who receive irrelevant notifications may become dissatisfied with their recommendation service. According to the I-centric paradigm proposed by the Wireless World Research Forum (WWRF), the service provision should be tailored to the actual needs of the user [3]. The I-centric vision promotes personalization, ambient awareness, and adaptability as the core requirements of future services.

A number of studies have been performed to realize this user-centric vision. A pioneering recent work was performed by Hossain *et al.* [6]. In this work, the authors proposed a gain-based media selection mechanism. In this regard, the gains obtained by the ambient media services were estimated by combining the media's reputation, the user's context and the user's profile. As a result of such a modeling process, the service selection problem was formulated as a gain maximization problem. Thereafter, a combination of a dynamic and a greedy approach was used to solve the problem. There are some fundamental differences between the study of [6] and the approach that we have proposed in Appendix C. From an architectural point of view, our work is based on a Publish/Subscribe paradigm in order to realize matchmaking between available services and the user's preferences. Moreover, the authors of [6] did not present mechanisms to compute the reputation of the media services, thus, in effect, assuming that it is merely static.

A pertinent study that falls in the same class of our work is the *Dynamos* project [7]. This approach is an example of a context-aware mobile application that can be used for recommending relevant services to the user. In [7], the authors designed a hybrid recommender system for notifying users about relevant services in a context-aware manner. The model is based on a peer-to-peer social functionality model, where the users can generate contextual notes and ratings, and attach them to services, or to the environments. They are also permitted to share these pieces of information with their peers. The attached notes to the environment are delivered to other users whenever they are in the spatial vicinity of the entities associated with the notes. A main difference between their work and what we propose is the way by which preferences are described. Their work assumed that the user was expected to explicitly describe his preferences by manually entering them. In this sense, the profile is defined by the user by explicitly specifying the types of activities and associating to them multiple interests. Such an approach can be considered to be a more a "primitive" approach – it is not viable in pervasive environments where preferences change over time. Moreover, the issue of suppressing repetitive notifications was not addressed in [7].

A comprehensive study for personalized service provision was performed by Naudet *et al.* from Bell Labs [8]. In [8], Naudet *et al.* designed an application for filtering the TV content provided to users' mobiles based on their learned profiles. The application is based on the use of ontologies to capture content descriptions as well as the users' interests. The latter interests are, in turn, mined using a dedicated profiling engine presented in [9], which leveraged Machine Learning techniques for user profiling.

3.3.3 Summary of the Contributions

In the paper included in Appendix C, we demonstrate how we construct a hybrid recommender system that minimizes the distraction to a user's attention while, simultaneously, maximizing the hit ratio of the service notifications. In accordance with the multiple dimensions that affect the decision making process, we have also defined a set of enabler components. The synergy between these enabler components is ensured through a Publish/Subscribe architecture.

Through our architecture, we demonstrate that AI can be of great support in the field of pervasive computing. In fact, our architecture paves the way towards more integration of AI in unobtrusive applications that adapt their behavior to the user needs.

Repetitively reissuing of the same notification regarding the same service is usually regarded as a nuisance to the user's attention. We have addressed the issue of suppressing repetitive notifications in a social mobile application. With regard to recommender systems, to the best of our knowledge, the question of suppressing repetitive notifications has not been addressed before in the literature.

The paper in Appendix C report the results of simulations conducted. These results demonstrate the efficiency of our design in reducing the unobtrusiveness that might be caused by traditional service recommendation systems. The design avoids flooding the user with irrelevant information.

3.4 Contributions in Learning the Preferences of Users

3.4.1 Overview of the Contributions

The problem of estimating the preferences of users is fundamental for personalized applications which range from service recommender systems to the targeted advertising of services. However, unlike traditional estimation problems where the underlying target distribution is stationary, estimating a user's interests, typically, involves non-stationary distributions.Therefore, resorting to strong estimators that converge with probability 1 is inefficient since they rely on the assumption that the distribution of the user's preferences is stationary. In the paper included in Appendix D, we propose a method by which we can use a family of stochastic-learning based *Weak* estimators for learning and tracking user's time varying interests.

Recently, Oommen and Rueda [14] have proposed a strategy by which the parameters of a binomial/multinomial distribution can be estimated when the underlying distribution is non-stationary. The method is referred to as Stochastic Learning Weak Estimation (SLWE), and is based on the principles of stochastic LA [68, 69]. The SLWE has found successful applications in many real-life problems that involve estimating distributions in non-stationary environments such as in adaptive encoding [70], route selection in mobile ad-hoc networks [71], and topic detection and tracking in multilingual online discussions [72]. Motivated by these successful applications of the SLWE in various areas, in Appendix D, we consider employing it for solving the intriguing problem of tracking user's interests.

The objective of the work in that paper included in Appendix D is to present a personalized *Learning Preferences Manager*, i.e, a *modus operandus* for capturing user's preferences. The latter will be able to cope with changes brought about by variations in the distribution of the user's interests, which will be where the SLWE plays a prominent part. In the quest to learn the user's dynamic profile, the *Learning Preferences Manager* is guided by so-called *Relevance Feedback* (RF) [128]. We rely on the *Service Usage History* (analogous to the history maintained by the authors of [6, 64]) as the main source of the RF. In fact, a common approach towards constructing a user's profile is through non-intrusively monitoring the history of the usage of his services. A *Service Usage History* (also known as the *Interaction His-*

tory), contains the history of the services used by the user over time. For example, when the user has used a certain service at a certain time instant, the *Learning Preferences Manager* refines and revises the user's profile based on the current instance of the usage history, which, in turn, is automatically and unobtrusively observed in the background. To obtain an index to measure this, the sum of the scores of a data item for a given attribute is made to be equal to unity. To now quantify this, we have opted, in the paper included in Appendix D, to use the SLWE [14] to update the score of the data item based on the usage history. Whenever a user selects a service, the metadata describing the service is used to update the score of data item. Thus, for example, if a user currently views a "action" movie, the scheme would increase the weight associated with the data item "action".

In the paper mentioned here, we distinguish two classes of data items that, in turn, require two different forms of update mechanisms. In fact, the data items related to a given attribute could be either semantically **disjunctive** or semantically **conjunctive**. Both of these cases have been systematically studied and analyzed.

3.4.2 Related Work

The core function of a personalized *Learning Preferences Manager* is to update the user's profile in a dynamic and incremental way. This is done so that the Learning Preferences Manager can closely follow the real-time evolution of the user's interests. In fact, any user's interests are not constant over time, and therefore it is imperative that the system takes the profile's drift into account. In this sense, whenever one attempts to represent the user's *current* interests, the most recent observations are more reliable than older ones. From a more general perspective, the task of learning the drifts in the user's interests corresponds to the problem of learning evolving concepts [73]. There are several studies that have dealt with the task of learning a user's interests. These include the use of a sliding window [74], aging examples [75], and a Gradual Forgetting (GF) function [66, 67, 65] etc. However, of all these, a sliding window approach is the most popular one. It consists of learning the description of the user's interests from the most recent observations, and thereafter, of discarding the observations that fall outside the window.

A substantial shortcoming of the sliding window approach is the choice of the window size. In [74], the authors adopted a fixed-size time window in order to learn a user's scheduling preferences. They empirically determined that a window size of 180 was a proper choice for their particular scheduling application. The GF, on the other hand, relies on assigning weights to the observations that decrease over time. Hence, the influence of older (more "stale") observations on the running estimates, decreases with time. The authors of [65] suggested a linearly-decreasing function, $w = f(t)$, for decaying the relative weights of the GF. In order to achieve a synergy between both the two approaches, namely GF and sliding window, Koychev, in [65], proposed to apply the GF *within each sliding window*. Thus, in this case, the parameter n (i.e., the length of the observation sequence) was set to be equal to L, where L denotes the length of the window.

Apart from the sliding window and GF schemes, other approaches, which also deal with *change detection*, have also emerged. In general, there are two major competitive sequential change-point detection algorithms: Page's cumulative sum (CUSUM) [76] detection procedure and the Shiryaev−Roberts−Pollak detection procedure. In [77], Shiryayev used a Bayesian approach to detect changes in the parameters distribution, where the change points were assumed to obey a geometric distribution. CUMSUM is motivated by a maximum likelihood ratio test for the hypotheses that a change occurred. Both approaches utilize the log-likelihood ratio for the hypotheses that the change occurred at the point, and that there is no change. Inherent limitations of CUMSUM and the Shiryaev−Roberts−Pollak approaches for on-line implementation are the demanding computational and memory requirements. In contrast to the CUMSU and the Shiryaev−Roberts−Pollak, the SLWE avoids the intensive computations of ratios, and do not invoke hypothesis testing.

It is worth noting that change detection techniques and those involving weak estimators differ from a conceptual point of view. In fact, change detections techniques are interested in determining the random instances when a change in the underlying distribution occurs. As opposed to this, weak estimators aspire to compute accurate estimates by unlearning old (stale) estimates without necessarily inferring the instant of change. In this vein, in change detection techniques, a common approach to track the time-varying estimates is to rely *only* on the data samples that occurred after the inferred change instant in order to compute the estimates.

A particularly interesting recent study for learning user's interests in ambient media services (and in, consequently, locating relevant services) was reported in [64]. Hossain *et al* devised the so-called Ambient Media Score Update method. The SU method was used to learn a user's changing interests [6, 64] by recording the so-called "scores", which represented his/her affinity of interests. In order to follow closely the evolution of the scores, the authors of [64] refined their proposed updating method defined earlier in [6], and updated the scores of the services at every time instant whenever the service was used. This was done instead of performing updates in a batch mode [6].

3.4.3 Overview of the Contributions

To the best of our knowledge, our work documented in the paper included in Appendix D presents the first attempt to apply a LA-inspired approach, such as the SLWE, to the real-life problem of tracking user's interests.

Philosophically, our profile representation model is distantly related to the approach presented in [6, 64], where the authors utilized the history to update the affinity of the user's interests. However, a substantial difference from the latter studies is our novel categorization of the data items that constitute a profile, into its so-called *disjunctive* and *conjunctive* data items. To the best of our knowledge, although profile update approaches in which the data items which are disjunctive have received a significant interest, the case of conjunctive data items remains largely unaddressed. We propose an adequate update form based on the principles of the SLWE, for each case of these two cases. It is worth noting that numerous applications of the SLWE has already been reported in the literature. However, its use for conjunctive and disjunctive data items is totally new.

The model which we have adopted, namely that of the user's interests changing "abruptly", is, in itself, interesting. In fact, instead of presuming that the so-called environment's "switch" occurs with some fixed periodicity [14], we assume that changes in the distribution of the user's interests occur at *unknown random time instants*. Furthermore, we suppose that the distribution changes to a possibly new random distribution after the switch. Such a model of the distribution's versatility is more realistic than the one which possesses a fixed periodicity-based changing model, and this is thus more appropriate in the context of estimating the user's preferences. Clearly, the described settings represents a particularly challenging scenario for any approach which models and studies change detection! The experiments conducted and the results reported, demonstrate that our approach exhibits lower error and faster adaptivity than the state-of-the-art.

3.5 Contributions in the field of Random Walk-Jump Process

3.5.1 Overview of the Contributions

Although Random Walks (RWs) with single-step transitions have been extensively studied for almost a century [129], problems involving the analysis of RWs that contain interleaving random steps and random "jumps" are intrinsically hard. In the paper presented in Appendix E, we have considered the analysis of one such fascinating RW, where every step is paired with its counterpart random jump. Apart from this RW being conceptually interesting, it also has applications in testing of entities (components or personnel), where the entity is never allowed to make more than a pre-specified number of *consecutive* failures. The paper in this appendix contains the analysis of the chain, some fascinating limiting properties, and simulations that justify the analytic results.

In the paper included in Appendix E, we consider the scenario when we are given the task of testing an error-prone component. At every time step, the component is subject to failure, where the event of failure occurs with a certain probability, q. The corresponding probability of the component not failing[2] is p, where $p = 1 - q$. Further, like all real-life entities, the component can operate under two modes, either in the *Well-Functioning* mode, or in the *Mal-Functioning* mode. At a given time step, we aim to determine if the component is behaving well, i.e, in the *Well-Functioning* mode, or if it is in the *Mal-Functioning* mode, which are the two states of nature. It is not unreasonable to assume that both these hypotheses are mutually exclusive, implying that only one of these describes the state of the component at a given time step, thus excluding the alternative. Our paper considers a possible strategy for determining the appropriate hypothesis for the state of nature.

To achieve this, suppose that the current maintained hypothesis conjectures that the component is in a *Mal-Functioning* mode. This hypothesis is undermined and systematically replaced by the hypothesis that the component is in its *Well-Functioning* mode if it succeeds to realize a certain number N_1 of successive re-

[2]The latter quantity can also be perceived to be the probability of the component *recovering* from a failure, i.e., if it, indeed, had failed at the previous time instant.

coveries (or successes). In the same vein, suppose that the current hypothesis conjectures that the component is in its *Well-Functioning* mode. This hypothesis, on the other hand, is invalidated and systematically replaced by the hypothesis that the component is in its *Mal-Functioning* mode if the component makes a certain number $N_2 + 1$ of successive failures. In our paper, we argue that such a hypothesis testing paradigm is most appropriately modeled by a RW in which the random steps and jumps are interleaving.

3.5.2 Related Work

The theory of RWs and its applications have been central to the research of stochastic processes since the start of the last century. From the recorded literature, one perceives that the pioneering treatment of a one-dimensional RW was due to Karl Pearson in 1905 [130].

It is pertinent to mention that the available results pertaining to RWs in which the chain can move from any state N to *non-neighboring* states $N+k$ or $N-k$ are scanty. Indeed, the analysis of the corresponding MCs is almost impossible in the most general case, *except when certain conditions like time reversibility can be invoked*. Finally, such chains fall completely outside the scope of the so-called family of "birth-and-death" processes, because even in these cases, the number of individuals does not drastically fall or increase in a single time unit (except in the case when one models catastrophes - but even here, MCs of this type are unreported).

3.5.3 Summary of the Contributions

In our work documented in Appendix E, we have analyzed a novel RW with interleaving steps and jumps, whose applications domains have been linked above. Also, as alluded to above, although RWs with single-step transitions have been extensively studied for almost a century [129], problems involving the analysis of RWs that contain interleaving random steps and random "jumps" are intrinsically hard. In our paper, every step is paired with its counterpart random jump. The obtained theoretical results are novel and quantify the asymptotic behavior of the chain. Morover, the paper also contains additional theoretical results regarding the symmetry properties of the chain.

The associated simulations results show that our scheme posseses an excellent ability to cope with non-stationary environments.

The RW we devised has applications in the testing of entities (components or personnel), because we can constrain the entity to never be allowed to make more than a pre-specified number of *consecutive* failures. The results also present avenues for further research in issues related to exploration and exploitation in multi-arm bandit type problems.

3.6 Contributions in Designing a Novel Stochastic Discretized Weak Estimator

3.6.1 Overview of the Contributions

The task of designing estimators that are able to track time-varying distributions has found promising applications in many real-life problems. A particularly interesting family of distributions are binomial/multiomial distributions. Existing approaches resort to sliding windows that track changes by discarding old observations. In Appendix F, we report a novel estimator, referred to as the Stochastic Discretized Weak Estimator (SDWE), that is based on the principles of LA. The estimator is able to estimate the parameters of a time-varying binomial distribution using finite memory. It tracks changes in the distribution by operating a controlled random walk on a discretized space. The steps of the estimator are discretized so that the updates are done in jumps, and thus the convergence speed is increased. The analogous results for the binomial distribution have also been extended for the multinomial case. Interestingly, the estimator possesses a low computational complexity that is independent of the number of the parameters of the multinomial distribution. Our experimental results demonstrate that the new estimator is able cope with non-stationary environments with high adaptation rate and accuracy.

As alluded to above, the devised SDWE is based on the theory of LA [68, 84], and in particular to the family of *ergodic* and *discretized* LA. Ergodic automata are known to better adapt to non-stationary environments where the reward probabilities are time dependent. In a parallel vein, with respect to the set of values that the action probabilities can take, LA typically fall into one of two categories, namely Continuous and Discretized. Continuous LA permit the action probabilities to take any value in the interval $[0,1]$. In practice, the relatively slow rate of convergence of these algorithms constituted a limiting factor in their applicability. In order to increase their speed of convergence, the concept of discretizing the probability space was introduced in [15, 108]. This concept is implemented by restricting the probability of choosing an action to be one of a finite number of values in the interval $[0,1]$. If the values allowed are equally spaced in this interval, the discretization is said to be *linear*, otherwise, the discretization is called *non-linear*. Following the discretization concept, many of the continuous VSSA have been discretized; indeed, discretized versions of almost all continuous automata have been reported [15]. Families of Pursuit and Estimator-based LA have been shown to be faster than VSSA [86]. As a matter of a fact, even faster discretized versions of these schemes have been reported [79, 15]. In brief, our estimator relies on the principle of discretization in order to hasten the convergence speed, and on the phenomenon of ergodicity to be able to cope with non-stationary distributions.

3.6.2 Related Work

Estimation is a fundamental and substantial issue in statistical problems. Estimators generally fall into various categories including the sort of Maximum Likelihood Estimates (MLE) and the Bayesian family of estimates. The MLE and Bayesian estimates are well-known for having good computational and statistical properties. However, the basic premise for establishing the quality of estimates is based on the assumption that the parameters being estimated do not change with time, i.e, the distribution is assumed to be stationary. Thus, it is desirable that the estimate converges to the true underlying parameter with probability 1, as the number of samples increases.

Consider, however, the scenario when the parameter being estimated changes with time. Thus, for example, let us suppose that the Bernoulli trials leading to binomially distributed random variable were done in a time-varying manner, where the parameter *switched*, for example periodically. In the binomial case, it implies that the *parameter* of the binomial distribution *switches*, for example periodically. Such a scenario demonstrates the behavior of a non-stationary environment. Thus, in this case, the goal of an estimator scheme would be to estimate the parameter, and to be able to adapt to any changes occurring in the environment. In other words, the algorithm must be able to detect the changes and estimate the new parameter after a *switch* has occurred in the environment. If one uses strong estimators (i.e., estimators that converge w.p. 1), it is impossible for the learned parameter to change rapidly from the value to which it has converged to, and thus resulting in poor time varying estimates.

Recently, Oommen and Rueda [14] have presented a strategy by which the parameters of a binomial/multinomial distribution can be estimated when the underlying distribution is non-stationary. The method has been referred to as the Stochastic Learning Weak Estimator (SLWE), and is based on the principles of continuous stochastic LA. A part from the SLWE, there is a number of continous estimators such as moving average, but they are all continous. As opposed to this, our scheme presented in Appendix F, uses the alluded to learning principles which resort to discretizing the probability space [131, 132, 79, 108], and performing a controlled random walk on this discretized space. By virtue of this discretization, the convergence rate has been shown to increase.

Apart from the SLWE and those described in [14], the literature also reports estimators involving the families of moving average estimators. But as they essentially work within the realm of continuous *time series analysis*, they are outside of the scope of this thesis, and we will thus not elaborate on them here.

3.6.3 Summary of the Contributions

To the best of our knowledge, the SDWE is the first reported discretized estimator that is able to track a time varying binomial/multnimoial distribution.

The scheme uses the discretization principle of LA, and operates on a controlled random walk on this space. Indeed, by virtue of discretization, our SDWE yields faster convergence speed than the analogous continuous weak estimators. In fact, comprehensive simulation results demonstrate that the new estimator is able to cope with non-stationary environments with high adaptation rate and accuracy. In addition, the results suggest that the SDWE outperforms the MLEW as well as the SLWE.

The SDWE also posseses a low computational complexity, measured in terms of the number of updates per time step to the estimates vector. Interestingly, this is independent of the number of parameters of the multinomial distribution to be estimated. In fact, the SDWE makes at most 2 updates per time step, thus rendering the worst case complexity to be constant or $O(1)$. To the best of our knowledge, this characteristic is unique when compared to other estimators including the acclaimed SLWE which posseses a complexity of $O(r)$, where r is the number of parameters of the multinomial variable.

A part from being computationally efficient, the scheme is also memory efficient and can be implemented using simple finite state machines. The possible applications of the SDWE for classification and language detection is currently being investigated.

Chapter 4

Conclusion and Future Research

In this thesis, we presented a user-centric architecture for recommending relevant services to the mobile user in pervasive environments. The aim of our architecture was to reduce the cognitive load on the user when it concerns selecting services. The architecture is adaptive in the sense that it is able to cope with users' contexts that are changing and with the drifts in the users' interests, while it simultaneously can track the reputations of services, and suppress repetitive notifications based on the history of the recommendations.

In this vein, we devised a Reputation Manager that identifies reputable services in the presence of a significant ratio of deceptive referrals. While most of the legacy approaches are vulnerable to the ratio of inaccurate recommenders, the RS that we have proposed was shown to be robust to the undermining effect of inaccurate recommenders. In addition, it utilized the power of Word of Mouth (WoM) communications in an optimal way in the absence of direct experience.

Designing a Novelty Checker for suppressing redundant notifications involves solving a fascinating problem of on-line discovery and tracking of noisy Spatio-Temporal Patterns. In this regard, we presented a novel solution to this problem using the principles of LA. The solution was based on a new family of RWs with interleaving jumps. Beside the application domain, in a separate study, we also examined the properties of this RW with interleaving jumps. The results that we have obtained constitute a significant contribution to the field of RWs.

With regard to user's profiling, we proposed a method by which we can use a family of SLW estimators for learning and tracking user's time varying interests. Since increasing the learning speed of the SLWE is a problem in itself, we tackled this issue and reported the first discretized version of the SLWE. This discretized weak estimator has the property that it can provide significant benefits to the Learning Preferences Manager. In addition, we presented an instantiation of our architecture for a real-life, day-to-day scenario involving a proactive location-based application which provides an ensemble of services.

As our overall conclusion, in the thesis, we have demonstrated that LA can be a great support tool to the field of Pervasive Computing. We hope that the thesis can be a key driver for more sophisticated integration of LA in the design of unobtrusive applications that adapt their behavior to the user needs.

Several overall research directions for further investigation arise from this thesis. They are discussed below.

4.1 Future work in RSs

1. In this thesis, we solved the $ATTP$ with mutually exclusive and exhaustive groups of agents. An intriguing future research direction involves solving the $ATTP$ for **several** groups of deceptive and fair agents, where each individual group is characterized by an unknown probability of truthfully communicating the reports. In addition, the question of fusing the experience of these different "heterogenous" groups is an open research question.

2. In the proposed solution to the $ATTP$, the parameter used to decide whether two agents are reckoned "similar" is stochastic, while the latter parameter was assumed *constant* by the authors of [117, 118, 119]. We believe that such a transformation can be integrated into the OMA in order to realize novel adaptive clustering algorithms, where the similarity measure between two elements is a stochastic parameter, and where the algorithm is only allowed to perceive *instantiations* of the latter random variable. We believe that this interesting idea would result in solving a wide range of clustering problems.

3. The solution to the $ATTP$ problem which we proposed in the thesis is based on the principles of Fixed Structure Stochastic Automaton (FSSA). An ap-

pealing alternative solution to the latter problem is to resort to the family of Variable Structure Stochastic Automata (VSSA) instead of FSSA. A future research direction is to attempt to model the "ties" between agents using a graph where the edges represent a similarity measure between the corresponding nodes. Such a similarity measure can be updated using an appropriate VSSA scheme. In order to partition the agents into fair and deceptive groups, we believe that we can invoke a minimum-cut approach [133].

4. Most legacy approaches in RSs, discard the ratings from deceptive agents instead of trying to intelligently make use of them. An extremely promising research avenue is to attempt to make use of the unfair ratings after performing some "adjustment" operations. In the study proposed in [26], the authors considered the case of continuous ratings, and reported some "deterministic" models of malicious manipulation of the ratings in a trust network that included complementary models, exaggerated positive models and exaggerated negative models. We propose that future research could involve inferring the model for manipulating the ratings that each malicious agent in the system is adopting, so that we can "adjust" the system's behavior as per the unfair ratings and render them fair. This task could include identifying the deceptive agents, and classifying them into different classes according to the pattern of the malicious behavior that they exhibit. Then, for each class of malicious behavior, an appropriate adjustment of the ratings can be performed to "transform" the deceptive ratings into informative ratings.

5. In many real-life situations, the perceived performance of the service by the user does not merely depend on its "intrinsic" quality but can also be seen to be dependent on extrinsic factors such as the current load of the service. For example, a Web-Service might perform poorly when accessed by a significant number of users. In such settings, identifying "intrinsically" high quality services in the presence of deceptive agents will be increasingly difficult since, in the case of negative rating, it is uncertain wether such a negative rating is due to its "intrinsically" low quality or to an excessive load that degrades the performance. Moreover, if a RS considers only the "intrinsic" quality of

a service without taking into account the load of the service, most users in the system would attempt to access “top ranked services” in the same time, and thus, the perceived performance will degrade. Finding ways to distribute and/or schedule the access to the services so as to improve the user’s quality of experience is an interesting research direction.

4.2 Future work in the field of noisy Spatio-Temporal Patterns

1. In this thesis, we reported a solution to tracking noisy Spatio-Temporal Patterns using a controlled RW. A fascinating avenue for future work is to investigate the application of the SLWE as an alternative solution to the problem, and to compare its performance to our existing approach.

2. Our approach relies on the assumption that the noisy patterns belong to a set of hypotheses. A research direction that is worth pursuing would be to design an “unguided” approach for tracking Spatio-Temporal patterns, i.e, one that would not require the hypotheseses space.

3. In the field of data mining, *association rule learning* is a well-known approach for mining *frequent itemsets* [134]. The generalization of our approach to learn the set of evolving *association rules* in a dynamic stream of data is a particularly intriguing open research question.

4.3 Future Work in Designing User-centric Architecture for Personalized Service Provisioning in Pervasive Environments

1. In this thesis, we tested our architecture using a simulation framework in which we chose, as a performance metric, the ratio of relevant notifications. In a future work, it would be beneficial to conduct a user study in order to assess the acceptance of our architecture from the perspective of an end-user.

In such a study, privacy concerns related to the sharing of user's personal information with service providers should be addressed.

2. In the study [6], the authors proposed a gain-based media selection mechanism. The gains obtained by ambient media services were estimated by combining the media's reputation, the user's context and the user's profile using different weights. A future research possibility is to base our multi-criteria decision-making mechanism for recommending relevant services to the user based on a simple computation of the gains of the corresponding services. Additional parameters can be taken into account for evaluating the gain of the service such as its cost in terms of money or battery life.

4.4 Future work in Learning User's Preferences

1. In the work reported in [8], Naudet *et. al* introduced the concept of the Quantity of Affiliation (QoA), which is defined as the degree of affiliation of a content item to a given concept. From example, the movie "Shrek" can have a quantity of affiliation: Animation = 0.9, Comedy = 0.8 [8]. This concept is similar to that involving conjunctive data-items that we proposed for our Learning Preference Manager. The main difference is that in the case of conjunctive data-items, the QoA is only allowed to take binary values. An interesting research direction would thus be to design a novel SLWE update mechanism that is able to handle the case of continuous QoA. We believe that such an SLWE will present a significant contribution both to the family of Weak Estimator algorithms and to our Learning Preferences Manager.

2. The evaluation of the SLWE-based solution for learning preferences was performed using simulated data sets. As a future work, we intend to test our solution for different real-life data sets.

4.5 Future Work in SDWE

1. With regard to the SDWE, an interesting research direction is to attempt to control the step size of the SDWE in an on-line manner. Since, between two environment "switches", the environment is stationary, a viable intuitive idea is to decrease the step size between two environment "switches", so that the variance is reduced. On the other hand, as soon as an environment "switch" is detected, we suggest that the step size should be increased for a limited number of time iterations, so as to avoid getting trapped in the old estimates and to allow for fast migration towards the new target value. To achieve this task, a classical change-point detection algorithm (Such as Page's cumulative sum (CUSUM) [76] detection procedure or the Shiryaev−Roberts−Pollak detection procedure) can be employed in conjunction with the SDWE in order to control the step size. It is worth mentioning that the same idea could be applied to control the update parameter λ of the SLWE.

2. In many real-life scenarios, when the environment executes a "switch", the new target parameter being tracked and estimated is sampled from a given distribution, which is usually unknown. A Meta-Learning algorithm that tries to learn the latter *distribution* from which the target parameter is drawn at environment switches, is worth investigating. For example, imagine that the target parameter of a binomial distribution can take two values according to two-state Markov model. The steps of the SDWE can be chosen in a non-linear manner so that "big" jumps are realized between the two possible values of the target parameter, implying that the RW does not lose time in exploring the regions that are unlikely to contain the changing parameter of the target distribution.

4.6 Future Work in RWs with Interleaving Steps

1. In the thesis, we considered the analysis of ergodic RWs with interleaving jumps. As a future work, we intend to analyze the properties of the latter RW in the presence of absorbing barriers.

2. In our study of RWs with interleaving jumps, we assumed that an entity can operate **only under two modes**, either in the *Well-Functioning* mode, or in the *Mal-Functioning* mode. A natural extension of our work is to generalize the results to the case of several modes. In this sense, on can envisage multiple modes where the entity is "upgraded" to the next "better" mode after a certain number of consecutive successes, while it is "downgraded" to a "worse" mode after a specified number of consecutive failures. It should be increasingly difficult for the walker to advance from one mode to a "better mode".

3. The generalization for a researcher to use the same strategy to know when an AI scheme should switch from "Exploration" to "Exploitation" is an extremely interesting avenue for future research.

Bibliography

[1] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, "Project aura: Toward distraction-free pervasive computing," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22–31, 2002.

[2] H. A. Simon, *Designing Organizations for an Information-Rich World*, pp. 37–72. 1971.

[3] S. Arbanowski, P. Ballon, K. David, O. Droegehorn, H. Eertink, W. Kellerer, H. van Kranenburg, K. Raatikainen, and R. Popescu-Zeletin, "I-centric communications: Personalization, ambient awareness, and adaptability for future mobile services," *IEEE Communications Magazine*, vol. 42, no. 9, pp. 63–69, 2004.

[4] E. Kaasinen, "User needs for location-aware mobile services," *Personal Ubiquitous Comput.*, vol. 7, no. 1, pp. 70–79, 2003.

[5] S. Kurkovsky and K. Harihar, "Using ubiquitous computing in interactive mobile marketing," *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 227–240, 2006.

[6] M. A. Hossain, P. K. Atrey, and A. El Saddik, "Gain-based selection of ambient media services in pervasive environments," *Mob. Netw. Appl.*, vol. 13, no. 6, pp. 599–613, 2008.

[7] O. Riva and S. Toivonen, "The dynamos approach to support context-aware service provisioning in mobile environments," *J. Syst. Softw.*, vol. 80, no. 12, pp. 1956–1972, 2007.

[8] Y. Naudet, A. Aghasaryanb, S. Mignon, Y. Toms, and C. Senot, "Ontology-based profiling and recommendations for mobile tv," in *Semantics in Adaptive and Personalized Services* (M. Wallace, I. Anagnostopoulos, P. Mylonas, and M. Bielikova, eds.), vol. 279 of *Studies in Computational Intelligence*, pp. 23–48, Springer Berlin / Heidelberg, 2010.

[9] A. Aghasaryan, S. Betgé-Brezetz, C. Senot, and Y. Toms, "A profiling engine for converged service delivery platforms," *Bell Lab. Tech. J.*, vol. 13, no. 2, pp. 93–103, 2008.

[10] W.-S. Yang, H.-C. Cheng, and J.-B. Dia, "A location-aware recommender system for mobile shopping environments," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 437–445, 2008.

[11] M. Brunato and R. Battiti, "Pilgrim: A location broker and mobility-aware recommendation system," in *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, (Washington, DC, USA), p. 265, IEEE Computer Society, 2003.

[12] S. Norman, G. Fabien, and O. B. Kwon, "Ambient intelligence: The mycampus experience," in *Ambient Intelligence and Pervasive Computing* (T. Vasilakos and W. Pedrycz, eds.).

[13] N. Taylor, P. Robertson, B. Farshchian, K. Doolin, I. Roussaki, L. Marshall, R. Mullins, S. Druesedow, and K. Dolinar, "Pervasive computing in daidalos," *IEEE Pervasive Computing*, vol. 10, pp. 74 –81, jan.-march 2011.

[14] B. J. Oommen and L. Rueda, "Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments," *Pattern Recogn.*, vol. 39, no. 3, pp. 328–341, 2006.

[15] B. J. Oommen and M. Agache, "Continuous and discretized pursuit learning schemes: Various algorithms and their comparison," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 31, pp. 277–287, 2001.

[16] J. W. Creswell, *Research design : qualitative, quantitative, and mixed method approaches*. Sage Publications, 2nd ed ed., 2003.

[17] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.

[18] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web*, WWW '03, (New York, NY, USA), pp. 640–651, ACM, 2003.

[19] S. Brin, R. Motwani, L. Page, and T. Winograd, "What can you do with a Web in your Pocket?," *Data Engineering Bulletin*, vol. 21, no. 2, pp. 37–47, 1998.

[20] J. A. Golbeck, *Computing and applying trust in web-based social networks*. PhD thesis, College Park, MD, USA, 2005. AAI3178583.

[21] G. Zacharia and P. Maes, "Trust management through reputation mechanisms," *Applied Artificial Intelligence*, vol. 14, no. 9, pp. 881–907, 2000.

[22] M. E. Glickman, "Parameter estimation in large dynamic paired comparison experiments," *Applied Statistics*, vol. 48, pp. 377–394, 1999.

[23] A. Jøsang, "A logic for uncertain probabilities," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 9, pp. 279–311, June 2001.

[24] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of the 1997 workshop on New security paradigms*, NSPW '97, (New York, NY, USA), pp. 48–60, ACM, 1997.

[25] W. Teacy, J. Patel, N. Jennings, and M. Luck, "Travos: Trust and reputation in the context of inaccurate information sources," *Autonomous Agents and Multi-Agent Systems*, vol. 12, pp. 183–198, 2006.

[26] B. Yu and M. P. Singh, "Detecting deception in reputation management," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, (Melbourne, Australia), pp. 73–80, ACM, 2003.

[27] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proceedings of the 2nd*

ACM conference on Electronic commerce, (Minneapolis, Minnesota, United States), pp. 150–157, ACM, 2000.

[28] P. Macnaughton-smith, W. T. Williams, M. B. Dale, and L. G. Mockett, "Dissimilarity analysis: a new technique of hierarchical sub-division," *Nature*, vol. 202, pp. 1034–1035, June 1964.

[29] S. Buchegger and J. Le Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," in *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[30] S. Sen and N. Sajja, "Robustness of reputation-based trust: boolean case," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, (Bologna, Italy), pp. 288–293, ACM, 2002.

[31] A. Whitby, A. Jøsang, and J. Indulska, "Filtering out unfair ratings in bayesian reputation systems," in *Proceedings of the Third International Joint Conference on Autonomous Agenst and Multi Agent Systems*, (NY, USA), pp. 106–117, 2004.

[32] Z. Despotovic and K. Aberer, "A probabilistic approach to predict peers performance in P2P networks," in *Cooperative Information Agents VIII*, pp. 62–76, 2004.

[33] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *10th International World Wide Web Conference*, Association of Computing Machinery, Association of Computing Machinery, 2001.

[34] M. Weiser, "The computer for the twenty-first century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[35] S. Kurkovsky, "Pervasive computing: Past, present and future," in *Information and Communications Technology, 2007. ICICT 2007. ITI 5th International Conference on*, pp. 65–71, dec. 2007.

[36] W. Weber, J. M. Rabaey, and E. Aarts, *Ambient Intelligence*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[37] M. H. Vastenburg, D. V. Keyson, and H. Ridder, “Considerate home notification systems: a field study of acceptability of notifications in the home,” *Personal Ubiquitous Comput.*, vol. 12, pp. 555–566, November 2008.

[38] M. H. Vastenburg, D. V. Keyson, and H. de Ridder, “Considerate home notification systems: A user study of acceptability of notifications in a living-room laboratory,” *International Journal of Human-Computer Studies*, vol. 67, no. 9, pp. 814 – 826, 2009.

[39] E. Horvitz, C. Kadie, T. Paek, and D. Hovel, “Models of attention in computing and communication: from principles to applications,” *Commun. ACM*, vol. 46, pp. 52–59, March 2003.

[40] D. S. McCrickard and C. M. Chewar, “Attuning notification design to user goals and attention costs,” *Commun. ACM*, vol. 46, pp. 67–72, March 2003.

[41] K. S. Nagel, J. M. Hudson, and G. D. Abowd, “Predictors of availability in home life context-mediated communication,” in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, CSCW ’04, (New York, NY, USA), pp. 497–506, ACM, 2004.

[42] S. Whittaker, V. Bellotti, and J. Gwizdka, “Email in personal information management,” *Commun. ACM*, vol. 49, pp. 68–73, January 2006.

[43] J. Ho and S. S. Intille, “Using context-aware computing to reduce the perceived burden of interruptions from mobile devices,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI ’05, (New York, NY, USA), pp. 909–918, ACM, 2005.

[44] A. K. Dey, “Understanding and using context,” *Personal Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, 2001.

[45] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. V. Laerhoven, and W. V. d. Velde, “Advanced interaction in context,” in *HUC ’99: Proceedings*

of the 1st international symposium on Handheld and Ubiquitous Computing, (London, UK), pp. 89–101, Springer-Verlag, 1999.

[46] A. Smailagic and D. Siewiorek, “Application design for wearable and context-aware computers,” *Pervasive Computing, IEEE*, vol. 1, pp. 20–29, oct-dec 2002.

[47] K. K. Khedo, “Context-aware systems for mobile and ubiquitous networks,” in *ICNICONSMCL ’06: Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, (Washington, DC, USA), p. 123, IEEE Computer Society, 2006.

[48] M. Sutterer, O. Droegehorn, and K. David, “User profile management on service platforms for ubiquitous computing environments,” in *VTC Spring*, pp. 287–291, 2007.

[49] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, “Incorporating contextual information in recommender systems using a multidimensional approach,” *ACM Trans. Inf. Syst.*, vol. 23, pp. 103–145, January 2005.

[50] Z. Yu, X. Zhou, D. Zhang, C.-Y. Chin, X. Wang, and J. Men, “Supporting context-aware media recommendations for smart phones,” *IEEE Pervasive Computing*, vol. 5, pp. 68–75, July 2006.

[51] M.-H. Park, H.-S. Park, and S.-B. Cho, “Restaurant recommendation for group of people in mobile environments using probabilistic multi-criteria decision making,” in *Computer-Human Interaction* (S. Lee, H. Choo, S. Ha, and I. Shin, eds.), vol. 5068 of *Lecture Notes in Computer Science*, pp. 114–122, Springer Berlin / Heidelberg, 2008.

[52] H. Liang, F. Tsai, and A. Kwee, “Detecting novel business blogs,” in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, pp. 1–5, dec. 2009.

[53] Y. Yang, J. Zhang, J. Carbonell, and C. Jin, “Topic-conditioned novelty detection,” in *Proceedings of the eighth ACM SIGKDD international conference*

on Knowledge discovery and data mining, KDD ’02, (New York, NY, USA), pp. 688–693, ACM, 2002.

[54] J. Allan, R. Papka, and V. Lavrenko, “On-line new event detection and tracking,” in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’98, (New York, NY, USA), pp. 37–45, ACM, 1998.

[55] J. Allan, V. Lavrenko, and H. Jin, “First story detection in tdt is hard,” in *Proceedings of the ninth international conference on Information and knowledge management*, CIKM ’00, (New York, NY, USA), pp. 374–381, ACM, 2000.

[56] N. Stokes and J. Carthy, “Combining semantic and syntactic document classifiers to improve first story detection,” in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’01, (New York, NY, USA), pp. 424–425, ACM, 2001.

[57] Y. Zhang, J. Callan, and T. Minka, “Novelty and redundancy detection in adaptive filtering,” in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’02, (New York, NY, USA), pp. 81–88, ACM, 2002.

[58] A. Kwee, F. Tsai, and W. Tang, “Sentence-level novelty detection in english and malay,” in *Advances in Knowledge Discovery and Data Mining* (T. Theeramunkong, B. Kijsirikul, N. Cercone, and T.-B. Ho, eds.), vol. 5476 of *Lecture Notes in Computer Science*, pp. 40–51, Springer Berlin / Heidelberg, 2009.

[59] Y. Zhang, F. S. Tsai, and A. T. Kwee, “Multilingual sentence categorization and novelty mining,” *Information Processing and Management*, vol. 47, no. 5, pp. 667–675, 2011.

[60] E. Gabrilovich, S. Dumais, and E. Horvitz, “Newsjunkie: providing personalized newsfeeds via analysis of information novelty,” in *Proceedings of the*

13th international conference on World Wide Web, WWW '04, (New York, NY, USA), pp. 482–490, ACM, 2004.

[61] L. Zhao, M. Zhang, and S. Ma, "The nature of novelty detection," *Inf. Retr.*, vol. 9, pp. 521–541, November 2006.

[62] H. K. Y. Leung, I. Burcea, and H.-A. Jacobsen, "Modeling location-based services with subject spaces," in *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research*, CASCON '03, pp. 171–181, IBM Press, 2003.

[63] D. Godoy and A. Amandi, "User profiling in personal information agents: a survey," *Knowl. Eng. Rev.*, vol. 20, pp. 329–361, December 2005.

[64] M. A. Hossain, J. Parra, P. K. Atrey, and A. El Saddik, "A framework for human-centered provisioning of ambient media services," *Multimedia Tools and Applications*, vol. 44, pp. 407–431, 2009.

[65] I. Koychev and I. Schwab, "Adaptation to drifting user's interests," in *In Proceedings of ECML2000 Workshop: Machine Learning in New Information Age*, pp. 39–46, 2000.

[66] I. Koychev, "Gradual forgetting for adaptation to concept drift," in *In Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning*, pp. 101–106, 2000.

[67] I. Koychev and R. Lothian, "Tracking drifting concepts by time window optimisation," in *Research and Development in Intelligent Systems XXII* (M. Bramer, F. Coenen, and T. Allen, eds.), pp. 46–59, Springer London, 2006.

[68] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. Prentice-Hall, Inc., 1989.

[69] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers, 2004.

[70] L. Rueda and B. J. Oommen, "Stochastic automata-based estimators for adaptively compressing files with nonstationary distributions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 5, pp. 1196 –1200, 2006.

[71] B. J. Oommen and S. Misra, "Fault-tolerant routing in adversarial mobile ad hoc networks: anefficient route estimation scheme for non-stationary environments," *Telecommunication Systems*, vol. 44, pp. 159–169.

[72] A. Stensby, B. J. Oommen, and O.-C. Granmo, "Language detection and tracking in multilingual documents using weak estimators," in *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*, 2010.

[73] G. Widmer, "Tracking context changes through meta-learning," *Mach. Learn.*, vol. 27, no. 3, pp. 259–286, 1997.

[74] T. M. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, "Experience with a learning personal assistant," *Commun. ACM*, vol. 37, no. 7, pp. 80–91, 1994.

[75] M. A. Maloof and R. S. Michalski, "Selecting examples for partial memory learning," *Machine Learning*, vol. 41, pp. 27–52, 2000.

[76] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes: theory and application*. Prentice-Hall, Inc., 1993.

[77] A. N. Shiryayev, *Optimal Stopping Rules*. Springer, 1978.

[78] A. Yazidi, O.-C. Granmo, and B. J. Oommen, "Service selection in stochastic environments: a learning-automaton based solution," *Applied Intelligence*, pp. 1–21, 2011. 10.1007/s10489-011-0280-5.

[79] M. Agache and B. J. Oommen, "Generalized pursuit learning schemes: New families of continuous and discretized learning automata," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, pp. 738–749, December 2002.

[80] S. Lakshmivarahan, *Learning Algorithms Theory and Applications*. Springer-Verlag, 1981.

[81] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Applications*. Oxford: Pergamon Press, 1994.

[82] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, “Learning automata: Theory, paradigms, and applications,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, pp. 706–709, December 2002.

[83] A. S. Poznyak and K. Najim, *Learning Automata and Stochastic Optimization*. Berlin: Springer-Verlag, 1997.

[84] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Boston: Kluwer Academic, 2003.

[85] M. L. Tsetlin, *Automaton Theory and the Modeling of Biological Systems*. New York: Academic Press, 1973.

[86] M. A. L. Thathachar and P. S. Sastry, “Varieties of learning automata: An overview,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, pp. 711–722, December 2002.

[87] S. Misra and B. J. Oommen, “GPSPA: A new adaptive algorithm for maintaining shortest path routing trees in stochastic networks,” *International Journal of Communication Systems*, vol. 17, pp. 963–984, 2004.

[88] M. S. Obaidat, G. I. Papadimitriou, A. S. Pomportsis, and H. S. Laskaridis, “Learning automata-based bus arbitration for shared-edium ATM switches,” *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 32, pp. 815–820, 2002.

[89] B. J. Oommen and T. D. Roberts, “Continuous learning automata solutions to the capacity assignment problem,” *IEEE Transactions on Computers*, vol. C-49, pp. 608–620, 2000.

[90] G. I. Papadimitriou and A. S. Pomportsis, “Learning-automata-based TDMA protocols for broadcast communication systems with bursty traffic,” *IEEE Communication Letters*, pp. 107–109, 2000.

[91] A. F. Atlassis, N. H. Loukas, and A. V. Vasilakos, “The use of learning algorithms in ATM networks call admission control problem: A methodology,” *Computer Networks*, vol. 34, pp. 341–353, 2000.

[92] A. F. Atlassis and A. V. Vasilakos, “The use of reinforcement learning algorithms in traffic control of high speed networks,” *Advances in Computational Intelligence and Learning*, pp. 353–369, 2002.

[93] A. V. Vasilakos, M. P. Saltouros, A. F. Atlassis, and W. Pedrycz, “Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques,” *IEEE Transactions on Systems, Man and Cybernetics: Part C*, vol. 33, pp. 297–312, 2003.

[94] F. Seredynski, “Distributed scheduling using simple learning machines,” *European Journal of Operational Research*, vol. 107, pp. 401–413, 1998.

[95] J. Kabudian, M. R. Meybodi, and M. M. Homayounpour, “Applying continuous action reinforcement learning automata (CARLA) to global training of hidden markov models,” in *Proceedings of the International Conference on Information Technology: Coding and Computing , ITCC’04*, (Las Vegas, Nevada), pp. 638–642, 2004.

[96] M. R. Meybodi and H. Beigy, “New learning automata based algorithms for adaptation of backpropagation algorithm pararmeters,” *International Journal of Neural Systems*, vol. 12, pp. 45–67, 2002.

[97] C. Unsal, P. Kachroo, and J. S. Bay, “Simulation study of multiple intelligent vehicle control using stochastic learning automata,” *Transactions of the Society for Computer Simulation International*, vol. 14, pp. 193–210, 1997.

[98] B. J. Oommen and E. d. S. Croix, “Graph partitioning using learning automata,” *IEEE Transactions on Computers*, vol. C-45, pp. 195–208, 1995.

[99] J. J. Collins, C. C. Chow, and T. T. Imhoff, “Aperiodic stochastic resonance in excitable systems,” *Physical Review E*, vol. 52, pp. R3321–R3324, 1995.

[100] R. L. Cook, “Stochastic sampling in computer graphics,” *ACM Trans. Graph.*, vol. 5, pp. 51–72, 1986.

[101] M. Barzohar and D. B. Cooper, “Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 707–722, 1996.

[102] M. L. Brandeau and S. S. Chiu, “An overview of representative problems in location research,” *Management Science*, vol. 35, pp. 645–674, 1989.

[103] C. Bettstetter, H. Hartenstein, and X. Prez-Costa, “Stochastic properties of the random waypoint mobility model,” *Journal Wireless Networks*, vol. 10, pp. 555–567, 2004.

[104] B. S. Rowlingson and P. J. Diggle, *SPLANCS: Spatial Point Pattern Analysis Code in S-Plus*. University of Lancaster, North West Regional Research Laboratory, 1991.

[105] M. Paola, “Digital simulation of wind field velocity,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 74-76, pp. 91–109, 1998.

[106] J. P. Cusumano and B. W. Kimble, “A stochastic interrogation method for experimental measurements of global dynamics and basin evolution: Application to a two-well oscillator,” *Nonlinear Dynamics*, vol. 8, pp. 213–235, 1995.

[107] A. Baddeley and R. Turner, “Spatstat: An R package for analyzing spatial point patterns,” *Journal of Statistical Software*, vol. 12, pp. 1–42, 2005.

[108] M. A. L. Thathachar and B. J. Oommen, “Discretized reward-inaction learning automata,” *Journal of Cybernetics and Information Science*, pp. 24–29, Spring 1979.

[109] B. Oommen and D. Ma, "Deterministic Learning Automata Solutions to the Equipartitioning Problem," *IEEE Transaction Computer*, vol. 37, no. 1, pp. 2–13, 1988.

[110] W. Gale, S. Das, and C. Yu, "Improvements to an Algorithm for Equipartitioning," *IEEE Trans. Comput.*, vol. 39, no. 5, pp. 706–710, 1990.

[111] A. Schlosser, M. Voss, and L. Bruckner, "On the simulation of global reputation systems," *Journal of Artificial Societies and Social Simulation*, vol. 9, no. 1, pp. 1–23, 2005.

[112] J. Mundinger and J.-Y. Le Boudec, "Analysis of a reputation system for mobile Ad-Hoc networks with liars," *Perform. Eval.*, vol. 65, no. 3-4, pp. 212–226, 2008.

[113] G. Zacharia and P. Maes, "Trust management through reputation mechanisms," *Applied Artificial Intelligence*, vol. 14, no. 9, pp. 881–907, 2000.

[114] M. Chen and J. P. Singh, "Computing and using reputations for internet ratings," in *Proceedings of the 3rd ACM conference on Electronic Commerce*, (Tampa, Florida, USA), pp. 154–162, ACM, 2001.

[115] K. Regan, P. Poupart, and R. Cohen, "Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change," *International Conference on Machine Learning*, vol. 1, 2006.

[116] K. Regan and R. Cohen, "Designing adaptive buying agents in electronic markets using advice from other buyers to model seller reputation," *The Journal of Business and Technology*, vol. 1, 2005.

[117] B. J. Oommen and E. V. de St. Croix, "Graph partitioning using learning automata," *IEEE Trans. Comput.*, vol. 45, no. 2, pp. 195–208, 1996.

[118] B. J. Oommen and D. C. Y. Ma, "Deterministic learning automata solutions to the equipartitioning problem," *IEEE Trans. Comput.*, vol. 37, no. 1, pp. 2–13, 1988.

[119] B. J. Oommen and D. C. Y. Ma, "Stochastic automata solutions to the object partitioning problem," *The Computer Journal*, vol. 34, pp. A105–A120, Dec. 1992.

[120] L. P. Cox, A. Dalton, and V. Marupadi, "SmokeScreen: flexible privacy controls for presence-sharing," in *Proceedings of the 5th international conference on Mobile systems, applications and services*, (San Juan, Puerto Rico), pp. 233–245, ACM, 2007.

[121] N. Eagle and A. Pentland, "Social serendipity: mobilizing social software," *Pervasive Computing, IEEE*, vol. 4, no. 2, pp. 28–34, 2005.

[122] L. Holmquist, J. Falk, and J. Wigstrm, "Supporting group collaboration with interpersonal awareness devices," *Personal and Ubiquitous Computing*, vol. 3, pp. 13–21, Mar. 1999.

[123] G. Youngblood and D. Cook, "Data mining for hierarchical model creation," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 4, pp. 561–572, 2007.

[124] S. Ma and J. L. Hellerstein, "Mining partially periodic event patterns with unknown periods," in *Proceedings of the 17th International Conference on Data Engineering*, (Washington, DC, USA), pp. 205–214, IEEE Computer Society, 2001.

[125] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: adapting the smart home to the user," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 39, no. 5, pp. 949–959, 2009.

[126] P. Rashidi and D. J. Cook, "Keeping the intelligent environment resident in the loop," in *Proceedings of the International Conference on Intelligent Environments*, pp. 1–9, 2008.

[127] A. Yazidi, O.-C. Granmo, and B. J. Oommen, "Learning automaton based on-line discovery and tracking of spatio-temporal event patterns," *Submitted for publication*.

[128] M. Montaner, B. Lpez, and J. L. de la Rosa, “A taxonomy of recommender agents on the internet,” *Artificial Intelligence Review*, vol. 19, pp. 285–330, 2003.

[129] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. Wiley, 3 ed., January 1968.

[130] K. Pearson, “The Problem of the Random Walk,” *Nature*, vol. 72, p. 342, August 1905.

[131] J. K. Lanctôt and B. J. Oommen, “Discretized estimator learning automata,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-22, pp. 1473–1483, November/December 1992.

[132] B. J. Oommen and E. Hansen, “The asymptotic optimality of discretized linear reward-inaction learning automata,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-14, May/June 1986.

[133] D. R. Karger and C. Stein, “A new approach to the minimum cut problem,” *J. ACM*, vol. 43, pp. 601–640, July 1996.

[134] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, “Algorithms for association rule mining a general survey and comparison,” *SIGKDD Explor. Newsl.*, vol. 2, pp. 58–64, June 2000.

Appendix A

Paper I

Title:	Service Selection in Stochastic Environments: A Learning-Automaton Based Solutions
Authors:	**Anis Yazidi**[1], Ole-Christoffer Granmo[2] and B. John Oommen[1,2]
Affiliations:	[1]University of Agder, Faculty of Engineering and Science, P. O. Box 509, NO-4898 Grimstad, Norway [2]School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6
Journal:	*Vol. 36, 2012, pp. 617-637.*

Appl Intell (2012) 36:617–637
DOI 10.1007/s10489-011-0280-5

Service selection in stochastic environments: a learning-automaton based solution

Anis Yazidi · Ole-Christoffer Granmo · B. John Oommen

Published online: 17 February 2011

Abstract In this paper, we propose a novel solution to the problem of identifying services of high quality. The reported solutions to this problem have, in one way or the other, resorted to using so-called "Reputation Systems" (RSs). Although these systems can offer generic recommendations by aggregating user-provided opinions about the quality of the services under consideration, they are, understandably, prone to "ballot stuffing" and "badmouthing" in a competitive marketplace. In general, unfair ratings may degrade the trustworthiness of RSs, and additionally, changes in the quality of service, over time, can render previous ratings unreliable. As opposed to the reported solutions, in this paper, we propose to solve the problem using tools provided by Learning Automata (LA), which have proven properties capable of learning the optimal action when operating in unknown stochastic environments. Furthermore, they combine rapid and accurate convergence with low computational complexity. In addition to its computational simplicity, unlike most reported approaches, our scheme does not require prior knowledge of the *degree* of any of the above mentioned problems associated with RSs. Instead, it gradually learns the identity and characteristics of the users which provide fair ratings, and of those who provide unfair ratings, even when these are a consequence of them making unintentional mistakes.

Comprehensive empirical results show that our LA-based scheme efficiently handles any degree of unfair ratings (as long as these ratings are binary—the extension to non-binary ratings is "trivial", if we use the S-model of LA computations instead of the P-model). Furthermore, if the quality of services and/or the trustworthiness of the users change, our scheme is able to robustly track such changes over time. Finally, the scheme is ideal for decentralized processing. Accordingly, we believe that our LA-based scheme forms a promising basis for improving the performance of RSs in general.

Keywords Reputation systems · Learning automata · Stochastic optimization

The first author gratefully acknowledges the financial support of the *Ericsson Research*, Aachen, Germany, and the third author is grateful for the partial support provided by NSERC, the Natural Sciences and Engineering Research Council of Canada. A preliminary version of this paper was presented at IEA/AIE'10, the 2010 International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Cordoba, Spain, in June 2010.

A. Yazidi · O.-C. Granmo · B.J. Oommen (✉)
Department of ICT, University of Agder, Grimstad, Norway
e-mail: oommen@scs.carleton.ca

A. Yazidi
e-mail: anis.yazidi@uia.no

O.-C. Granmo
e-mail: ole.granmo@uia.no

B.J. Oommen
School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6

1 Introduction

1.1 Problem formulation

With the abundance of services available in today's world, identifying those of high quality is becoming increasingly difficult. Since this, typically, involves the comprehensive marketplace, an entire body of research has recently come to the forefront, namely, the study of so-called "Reputation Systems" (RSs). Such systems have attracted a lot of attention during the last decade in academia as well as in the industry, because they present a, hopefully, transparent method by which the user community, within a social network, can rank the quality of the services in question. RSs have also emerged as an efficient approach to handle trust in online services, and can be used to collect information about the performance of services in the absence of direct experience.

In this paper we intend to model and study how experiences can be shared between users in a social network, where the medium of collaboration is a RS. The basic premise, of course, is that it is possible for users to expediently obtain knowledge about the nature, quality and drawbacks of specific services by considering the experiences of other users. Indeed, the information that people used to share with their friends on campus or over a cup of coffee is now being broadcasted online, and one can thus, easily and electronically, take advantage of the comments of thousands of people participating in the social network. For instance, in a social network of tourists, sharing the experiences concerning restaurants can certainly prove advantageous.

The above premise is true if the basis of the decision is accurate, up-to-date and fair. Unless a person is naive, he must accept the fact that every user may not communicate his experiences truthfully. In fact, the social network and the system itself might contain misinformed/deceptive users who provide either unfair positive ratings about a subject or service, or who unfairly submit negative ratings. Such "deceptive" agents, who may even submit their inaccurate ratings innocently, have the effect that they mislead a RS that is based on blindly aggregating users' experiences. Furthermore, when the quality of services and the nature of users change over time, the challenge is further aggravated.

From this perspective, we can state a fundamental paradox[1] about using RSs by virtue of the fact that they are prone to "ballot stuffing" and "badmouthing" in a competitive marketplace. Users who want to promote a particular product or service can flood the domain (i.e., the social network) with sympathetic votes, while those who want to get a competitive edge over a specific product or service can "badmouth" it unfairly. Thus, although these systems can offer generic recommendations by aggregating user-provided opinions, unfair ratings may degrade the trustworthiness of such systems. Additionally, changes in the quality of service, over time, can render previous ratings unreliable. In general, unfair ratings may degrade the trustworthiness of RSs, and changes in the quality of service, over time, can render previous ratings unreliable.

This problem, of separating "fair" and "unfair" agents for a specific service, is called the Agent-Type Partitioning Problem ($ATPP$). Put in a nutshell, in this paper, we propose to solve the above mentioned paradoxical $ATPP$ using tools provided by Learning Automata (LA), which have powerful potential in efficiently and quickly learning the optimal action when operating in unknown stochastic environments. It adaptively, and in an on-line manner, gradually learns the identity and characteristics of the users who provide fair ratings, and of those which provide unfair ratings, even when these are a consequence of them making unintentional mistakes.

The solutions provided here have been subjected to rigorous experimental tests, and the results presented are, in our opinion, both novel and conclusive.

1.2 Reputation systems: state of the art

Finding ways to solve the $ATPP$ and thus counter the detrimental influence of unfair ratings on a RS has been a focal concern of a number of studies [3, 5, 13, 22, 28, 30]. Dellarocas [5] used elements from collaborative filtering to determine the nearest neighbors of an agent that exhibited similar ratings on commonly-rated subjects. He then applied a cluster filtering approach to filter out the most likely unfairly positive ratings. Sen and Sajja [22] proposed an algorithm to select a service provider to process a task by querying other user agents about their ratings of the available service providers. The main idea motivating their work is to select a subset of agents, who when queried, gives a minimum probabilistic guarantee that the majority of the queried agents provide correct reputation estimates. However, comprehensive experimental tests show that their approach is prone to the variation of the ratio of deceptive agents. As opposed to these, Zacharia [30] proposed a game-theoretic model to solve the trust problem in online markets. In [29], the authors present a heuristic methodology to reduce the computational complexity of ratings prediction in a trust network topology while maintaining accuracy. Their approach relies on experimentally verifying that the trust network exhibits the so called "small-worldness" network property. The authors verified the validity of their approach through experimental data extracted from online trust sites. Apart from online markets applications, the concept of trust was shown to be useful to avoid access to fraudulent and malicious web sites [15]. In [15], the authors presented a proxy-based approach that makes use of safety ratings provided by McAfee SiteAdvisor in order to prevent access to untrustworthy web sites.

It is worth noting that combining reports from different witnesses is akin to the problem of fusioning possibly conflicting sources of information [2, 7, 10]. Buchegger and Le Boudec [3] tackled the latter issue as follows: They proposed

[1] Instead of relying purely on traditional information sources, a user can opt to take advantage of social networks in the form of RSs to get more reliable recommendations. But instead, he risks ending up with even worse reliability than what was offered with traditional information sources because misinformed/deceptive users may "contaminate" the RSs. Hence the paradox!

a Bayesian reputation mechanism in which each node isolates malicious nodes by applying a deviation test methodology. Their approach requires the agent to have enough *direct* experience with the services so that he can evaluate the trustworthiness of the reports of the witnesses. While this is a desirable option, unfortunately, in real life, such an assumption does not always hold, specially when the number of possible services is large. In [4], Chen and Singh evaluated the quality of feedbacks assuming that a feedback is credible if it is consistent with the majority of feedbacks for a given user. Their approach, though promising, unfortunately, suffers from a deterioration in the performance when the ratio of deceptive agents is high. In [28], Yu and Singh devised a modified weighted majority algorithm to combine reports from several witnesses to determine the ratings of another agent. The main shortcoming of the work reported in [28] is its relatively slow rate of convergence. In contrast, in [27], Witby and Jøsang presented a Bayesian approach to filter out dishonest feedback based on an iterated filtering approach. In their approach, the authors extended the so-called "Beta" reputation system presented by Jøsang and Ismail [9]. The authors of [6] proposed a probabilistic model to assess peer trustworthiness in P2P networks. Their model, which, in one sense, is similar to ours, differs from the present work because the authors of [6] assume that a peer can deduce the trustworthiness of other peers by comparing its own performance with reports of other peers about itself. Though such an assumption permits a feedback-evaluating mechanism, it is based on the fact that peers provide services to one another, thus permitting every party the right to play the role of a service provider and the service consumer (a reporting agent). Our approach, which we briefly describe in the next section and then explain in greater detail subsequently, makes a clear distinction between these parties—the service provider and the reporting agent.

1.3 Overview of our solution

In this paper, we provide a novel solution to the above problems, and in particular to the $ATPP$, based on Learning Automata (LA), which can learn the optimal action when operating in unknown stochastic environments. Furthermore, they combine rapid and accurate convergence with low computational complexity. In addition to its computational simplicity, unlike most reported approaches, our scheme does not require prior knowledge of the *degree* of any of the above mentioned problems with RSs. Rather, it adaptively, and in an on-line manner, gradually learns the identity and characteristics of the users who provide fair ratings, and of those which provide unfair ratings, even when these are a consequence of them making unintentional mistakes.

Learning is achieved by interacting with a so-called "Environment", and by processing its responses to the actions that are chosen. Such automata have various applications such as parameter optimization, statistical decision making and telephone routing [14, 18–20]. Narendra and Thathachar [14] have dedicated a book that reviews the families and applications of LA, and a brief survey of this field is included here in the interest of completeness.

By suitably modeling reports about direct experiences involving a specific service as responses from the corresponding "Environment", our scheme intelligently groups agents according to the rating that they give to the same service. To formalize these responses, we define an agent to be "fair" (or "trustworthy") if it reports the service performance correctly with a probability $p > 0.5$. Similarly, an agent is said to be "deceptive" if it reports the inverted service performance with a probability $q > 0.5$. The beauty of our scheme is that although the identity of the reporting agents is unknown, "fair" agents will end up in the same group, while "deceptive" agents will converge to another group.

Unlike most existing reported approaches that only consider the feedback from "fair" agents as being informative, and which simultaneously discard the feedback from "unfair" agents, in our work we attempt to intelligently combine (or fuse) the feedback from fair and deceptive agents when evaluating the performance of a service. Moreover, we do not impose the constraint that we need a priori knowledge about the ratio of deceptive agents. Consequently, unlike most of existing work, that suffer from a decline in the performance when the ratio of deceptive agents increases, our scheme is robust to the variation of this ratio. This characteristic phenomenon of our scheme is unique.

1.4 Contributions of this paper

The novel contributions of this paper, when it concerns RSs, are the following:

- We demonstrate the applicability of LA to RSs—thus providing a promising real-time solution to this paradoxical problem. To our knowledge, the paper presents the first reported LA-based solution for any problem within the field of RSs.
- Since our solution is based on LA, it is both computationally simple and memory efficient.
- With regard to the field of LA itself, our scheme presents twofold contributions to traditional LA-based partitioning algorithms [17, 19, 20]. First of all, we do not impose the constraint that an equal number of agents must reside in the same partition. Secondly, we experimentally demonstrate that the partitioning still yields accurate results when the environment is stochastic. In this sense, the parameter used here to decide whether two agents are reckoned "similar" is stochastic, while the latter parameter was assumed *constant* in [17, 19, 20].

- Most importantly, however, our scheme maximizes the likelihood of selecting high quality services in the presence of an *unknown* ratio of deceptive agents. Indeed, not only does the scheme not require the a priori knowledge of the ratio of the deceptive agents, but it is also very robust to extremely high ratios of such deceptive or even malicious agents!

We conclude this section by mentioning that our results probably represent the state-of-the-art!

1.5 Paper Organization

Earlier, in Sect. 1.2 we presented a brief survey of the available solutions for dealing with fair and unfair agents in RSs. The rest of the paper is organized as follows. First of all, in Sect. 2, we present a formal statement of the problem. Then, in Sect. 3 we present a brief overview of the field of LA. Thereafter, in Sect. 4 we present our solution, which is the LA-based scheme for selecting services in stochastic environments. Experimental results obtained by rigorously testing our solution for a variety of scenarios and for agents with different characteristics, are presented in Sect. 5. Section 6 concludes the paper.

2 Modeling the problem

Let us consider a population of L services (or service providers), $\mathcal{S} = \{S_1, S_2, \dots, S_L\}$. We also assume that the social network (or pool of users) consists of N parties (synonymously called "agents") $\mathcal{U} = \{u_1, u_2, \dots, u_N\}$. Each service S_l has an associated quality, which, in our work is represented by an "innate" probability of the service provider performing exceptionally well whenever its service is requested by an agent. This probability is specified by the quantity θ_l, assumed to be unknown to the users/agents. For a given interaction instance between user agent u_i and service S_l, let x_{il} denote the performance value, which, for the sake of formalism, is assumed to be generated from a distribution referred to as the *Performance Distribution* of S_l. After the service has been provided, the user/agent u_i observes the performance x_{il}, where $x_{il} \in \{0, 1\}$. Since we intend to reduce our problem to a *maximization* problem, we assume that '0' denotes the lowest performance of the service, while '1' denotes its highest performance.[2]

At this juncture, after the agent has experienced the quality of the service, he communicates his experience to the rest of the network. Let y_{il} be the report that he transmits to other agents after he experiences x_{il}, where,[3] $y_{il} \in \{0, 1\}$. It is here that we have to model the genuineness of an agent communicating his evaluation accurately. To do this, we assume that agent u_i communicates his experience, x_{il}, truthfully to other agents in the population, with a probability p_i. In other words, p_i denotes the probability that agent u_i is not misreporting his experience. For ease of notation, we let $q_i = 1 - p_i$, which represents the probability that agent u_i does, in fact, misreport his experience. The intention for this symbolism should be obvious, because clearly, $p_i = Prob(x_{il} = y_{il})$.

Observe that as a result of this communication model, a "deceptive" agent will probabilistically tend to report low performance experience values for high performance services and vice versa. Our aim, then, is to incrementally partition the agents as being true/fair or deceptive, concurrently with their experiences being communicated to us. Furthermore, at the same time as the agents are being partitioned, our aim is to use the present state of the ongoing partitioning as a basis for decision making when selecting services. Thus, our scheme can be divided into two interacting phases, namely, an agent partitioning phase and a service selection phase. The input to the agent partitioning phase is the reports communicated by the other agents, while the input to the service selection phase is the current agent partitioning. Decisions about whether to access a service or not are the output of the overall procedure, with the goal of making the decisions that maximize the service performance experienced by the agent that acts upon those decisions.

Note that we do not assume that an agent can access any service, any time he wants. Rather, we assume that service access is spatially and temporally restricted. Thus, the true nature of services and agents are only gradually revealed, i.e., as the overall system of agents and services are observed over time. Also, note that we assume that some of the agents spend some time on exploration, and not all of their time purely on exploiting the experiences communicated by other agents. This is necessary in order for the overall system to discover the nature of new services when they are introduced, as well as detecting the new nature of an old service that changes nature.

Formally, the Agent-Type Partitioning Problem $(ATPP)$, can be stated as follows: A social network consists of N agents, $\mathcal{U} = \{u_1, u_2, \dots, u_N\}$, where each agent u_i is characterized by a fixed but unknown probability p_i of him reporting his experience truthfully. The ATPP involves partitioning $\mathcal{U}$ into 2 (mutually exclusive and exhaustive) groups so as to obtain a 2-partition $\mathbb{G}_k = \{G_i \mid i = 1, 2\}$, such that each group, G_i, of size, N_i, exclusively contains only the

[2]The extension to non-binary ratings (for example when x_{il} is a real number in the unit interval) is "trivial", if we use the S-model of LA computations instead of the P-model.

[3]We mention, in passing, that other researchers, have used the notation y_{il} to signify the rating of the service.

agents of its own type, i.e., which either communicate truthfully or deceptively.

Since the set of all possible solutions is isomorphic to the set of all possible subsets of $\mathcal{U}$, we conjecture that the problem of determining the optimal 2-partition is NP-hard.

To simplify the problem, we assume that every p_i can assume one of two[4] possible values from the set $\{p_d, p_f\}$, where $p_d < 0.5$ and $p_f > 0.5$. Then, agent u_i is said to be fair if $p_i = p_f$, and is said be deceptive if $p_i = p_d$.

Based on the above, the set of fair agents is $\mathcal{U}_f = \{u_i | p_i = p_f\}$, and the set of deceptive agents is $\mathcal{U}_d = \{u_i | p_i = p_d\}$.

Let y_{il} be a random variable defined as below:

$$y_{il} = \begin{cases} 1 & \text{w.p } p_i.\theta_l + (1 - p_i).(1 - \theta_l) \\ 0 & \text{w.p } p_i.(1 - \theta_l) + (1 - p_i).\theta_l. \end{cases} \quad (1)$$

Consider the scenario when two agents u_i and u_j utilize the same service S_l and report on it. Then, based on the above notation, their reports relative to the service S_l are y_{il} and y_{jl} respectively, where:

$$\begin{aligned} &Prob(y_{il} = y_{jl}) \\ &\quad = Prob[(y_{il} = 0 \wedge y_{jl} = 0) \vee (y_{il} = 1 \wedge y_{jl} = 1)] \\ &\quad = Prob[(y_{il} = 0 \wedge y_{jl} = 0)] \\ &\qquad + Prob[(y_{il} = 1 \wedge y_{jl} = 1)] \\ &\quad = Prob(y_{il} = 0) \cdot Prob(y_{jl} = 0) \\ &\qquad + Prob(y_{il} = 1) \cdot Prob(y_{jl} = 1). \end{aligned}$$

Throughout this paper, we shall denote $Prob(y_{il} = y_{jl})$ to be the probability that the agents u_i and u_j will agree in their appraisal. This quantity has the following property.

Theorem 1 *Let u_i and u_j two agents. If both u_i and u_j are of the same nature (either both deceptive agents or both fair), then $Prob(y_{il} = y_{jl}) > 0.5$. Similarly, if u_i and u_j are of different nature, then $Prob(y_{il} = y_{jl}) < 0.5$.*

Proof The proof is straightforward. □

We shall now proceed to present a brief overview of LA, the toolkit to solve the $ATPP$.

[4] Generalizing this so that each p_i can be an element of a set $\{p_{d_1}, p_{d_2}, \ldots p_{d_J}, p_{f_1}, p_{f_2}, \ldots p_{f_M}\}$, where every $p_{d_i} < 0.5$ and every $p_{f_j} > 0.5$ is rather trivial. It merely involves extending the arguments presented here for all possible pairs $\langle p_{d_i}, p_{f_j}\rangle$. Notice that in the same vein, agent u_i would be considered fair if $p_i \in \{p_{f_1}, p_{f_2}, \ldots p_{f_M}\}$, and he would be deceptive if $p_i \in \{p_{d_1}, p_{d_2}, \ldots p_{d_J}\}$.

3 Stochastic learning automata

Learning Automata[5] (LA) have been used in systems that have incomplete knowledge about the Environment in which they operate [1, 14, 21, 25]. The learning mechanism attempts to learn from a *stochastic Teacher* which models the Environment. In his pioneering work, Tsetlin [26] attempted to use LA to model biological learning. In general, a random action is selected based on a probability vector, and these action probabilities are updated based on the observation of the Environment's response, after which the procedure is repeated.

The term "Learning Automata" was first publicized by Narendra and Thathachar [14]. The goal of LA is to "determine the optimal action out of a set of allowable actions" [1]. The distinguishing characteristic of automata-based learning is that the search for the optimizing parameter vector is conducted in the space of probability *distributions* defined over the parameter space, rather than in the parameter space itself [24].

In the first LA designs, the transition and the output functions were time invariant, and for this reason these LA were considered "Fixed Structure Stochastic Automata" (FSSA). Tsetlin, Krylov, and Krinsky [26] presented notable examples of this type of automata. The solution we present here, essentially falls within this family and so we shall explain this family in greater detail in Sect. 3.1.

Later, Vorontsova and Varshavskii [14] introduced a class of stochastic automata known in the literature as Variable Structure Stochastic Automata (VSSA). In the definition of a VSSA, the LA is completely defined by a set of actions (one of which is the output of the automaton), a set of inputs (which is usually the response of the Environment) and a learning algorithm, T. The learning algorithm [14] operates on a vector (called *the Action Probability vector*)

$$P(t) = [p_1(t), \ldots, p_R(t)]^{\mathrm{T}},$$

where $p_i(t)$ $(i = 1, \ldots, R)$ is the probability that the automaton will select the action α_i at time 't', $p_i(t) = Pr[\alpha(t) = \alpha_i]$, $i = 1, \ldots, R$, and it satisfies

$$\sum_{i=1}^{R} p_i(t) = 1 \quad \forall t.$$

Note that the algorithm $T : [0, 1]^R \times A \times B \to [0, 1]^R$ is an updating scheme where $A = \{\alpha_1, \alpha_2, \ldots, \alpha_R\}$, $2 \le R < \infty$, is the set of output actions of the automaton,

[5] In the interest of completeness, we have included a brief review of the field of LA here. The review found in the earlier version of the paper has been abridged as per the desire of the referees. The list of applications of LA is also extensive, but omitted here in the interest of brevity and the advice of the referees.

and B is the set of responses from the Environment. Thus, the updating is such that

$$P(t+1) = T(P(t), \alpha(t), \beta(t)),$$

where $P(t)$ is the action probability vector, $\alpha(t)$ is the action chosen at time t, and $\beta(t)$ is the response it has obtained.

3.1 Fundamentals of FSSA

Since the solution to the $ATPP$ which we present here essentially falls within the family of FSSA, we explain them now in greater detail. A $FSSA$ is a quintuple ($\underline{\alpha}$, $\underline{\Phi}$, $\underline{\beta}$, F, G) where:

- $\underline{\alpha} = \{\alpha_1, \ldots, \alpha_R\}$ is the set of actions that it must choose from.
- $\underline{\Phi} = \{\phi_1, \ldots, \phi_S\}$ is a set of states.
- $\underline{\beta} = \{0, 1\}$ is its set of inputs. The '1' represents a penalty, while the '0' represents a reward.
- F is a map from $\Phi \times \beta$ to Φ. It defines the transition of the internal state of the automaton on receiving an input. F may be stochastic.
- G is a map from Φ to α, and it determines the action taken by the automaton if it is in a given state. With no loss of generality, G is deterministic.

As discussed above, the automaton is offered a set of actions, and it is constrained to choose one of them. When an action is chosen, the Environment gives out a response $\beta(t)$ at a time 't'. The automaton is either penalized or rewarded with an unknown probability c_i or $1 - c_i$, respectively. On the basis of the response $\beta(t)$, the state of the automaton $\phi(t)$ is updated and a new action is chosen at $(t+1)$. The penalty probability c_i satisfies:

$$c_i = Pr[\beta(t) = 1 | \alpha(t) = \alpha_i] \quad (i = 1, 2, \ldots, R).$$

The basic idea used to solve the $ATPP$ is based on a sub-class of LA solutions that has been used to solve the object partitioning problem [8, 16]. As documented in the literature, the object partitioning problem involves partitioning a set of $|\mathbb{P}|$ objects into $|\mathbb{N}|$ groups or classes, where the main aim is to partition the objects into groups that mimic an underlying unknown grouping. In other words, the objects which are accessed together must reside in the same group [16]. In the special case when all the groups are required to contain the same number of objects, the problem is also referred to as the Equi-Partitioning Problem (EPP). Many solutions involving LA have been proposed to solve the EPP, but the most efficient algorithm is the *Object Migrating Automaton* (OMA) [16]. The latter was first proposed by Oommen and Ma [16], and some modifications were added by Gale et al. [8] to create the *Adaptive Clustering Algorithm* (ACA). Since the OMA is, in one sense, the prior art on which our present solution is built, and since we have compared our scheme with the OMA, we briefly describe its design here.

3.2 Object migrating automaton (OMA)

The Object Migrating Automaton (OMA) is an ergodic automaton that has R actions $\{\alpha_1, \ldots, \alpha_R\}$ representing the possible underlying classes. Each action α_i has its own set of states $\{\phi_{i1}, \phi_{i2}, \ldots, \phi_{iM}\}$, where M is the depth of memory, and $1 \leq i \leq R$ represents the number of classes. ϕ_{i1} is called the most internal state and ϕ_{iM} is the boundary (or most external) state.

A set of W physical objects $\{A_1, A_2, \ldots, A_W\}$ is accessed by a random stream of queries, and the objects are to be partitioned into groups so that the frequently *jointly*-accessed objects are clustered together. The OMA utilizes W abstract objects $\{O_1, O_2, \ldots, O_W\}$ instead of migrating the physical objects. Each abstract object is assigned to a state belonging to an initial random group but in its boundary state. The objects within the automaton move from one action to another, and so, in this case, all the W abstract objects move around in the automaton. If the abstract objects O_i and O_j are in the action α_h, and the request accesses $\langle A_i, A_j \rangle$, then the OMA will be rewarded by moving them towards the most internal state ϕ_{h1}. But a penalty arises if the abstract objects O_i and O_j are in different classes, say α_h and α_g, respectively. Assuming O_i is in $\zeta_i \ \varepsilon \ \{\phi_{h1}, \phi_{h2}, \ldots, \phi_{hM}\}$ and O_j is in $\zeta_j \ \varepsilon \ \{\phi_{g1}, \phi_{g2}, \ldots, \phi_{gM}\}$, they will be moved as follows:

- If $\zeta_i \neq \phi_{hM}$ and $\zeta_j \neq \phi_{gM}$, O_i and O_j are moved one state toward ϕ_{hM} and ϕ_{gM}, respectively.
- If exactly one of them is in the boundary state, the object which is not in the boundary state is moved towards *its* boundary state.
- If both of them are in their boundary states, one of them, say O_i is moved to the boundary state of the other object ϕ_{gM}. In addition, the closest object to them is moved to the boundary state ϕ_{hM}, so as to preserve an equal number of objects in each group.

It is important to point out that the random stream of queries contains information about an optimal partition, and the OMA attempts to converge to it. The automaton is said to have converged when all the objects in a class are in the deepest (or second deepest) most-internal state.

The OMA can be improved by the following: Assume that a pair of objects $\langle A_i, A_j \rangle$ is accessed, where O_i is in the boundary state, while O_j is in a non-boundary state. In this case, a general check should be made to locate another object in the boundary state of the partition containing O_j. If there is an object, then swapping is done between this object and O_i in order to bring the two accessed objects into the same partition. In turn, instead of waiting for a long time to have these accessed objects in the same partition, the convergence speed can be increased by swapping the objects into the right partitions.

The formal algorithm for the OMA is found in [8, 16], and omitted here in the interest of space.

3.3 Similarities between the $ATPP$ and the EPP

The idea behind using LA as a tool to solve the $ATPP$ comes from the elegance of using them to solve the EPP. The similarity between the EPP and the $ATPP$ render LA as one of the promising candidate tools to solve the latter. This is because:

- As in the case of the EPP, the $ATPP$ is (possibly) NP-hard, primarily due to the exponential growth in the number of partitions of objects/agents.
- The EPP dictates that each partition must have the same number of objects. It is easy to see that an analogous condition can be imposed with the $ATPP$ if the number of fair and deceptive agents are equal. Relaxing this constraint will be one of our major challenges.
- The EPP and $ATPP$ seek to partition the objects/agents into groups that mimic the underlying unknown groups of objects and agents respectively. In the case of the EPP, the objects which are accessed together more frequently by a random sequence of queries are said to be in the same partition. As opposed to this, in the $ATPP$, the agents which are similar to each other (by being either fair or deceptive), are required to be in the same group so as to maximize the "within-group" and to minimize the "between-group" similarities.

We now highlight the *differences* between the two problems.

3.4 Limitations of the OMA in the $ATPP$ context

The reported instances of the OMA are not directly applicable for the $ATPP$. To develop our solution, we highlight the main restrictions, and the necessary enhancements which must be added to the OMA in order for it to be useful in our present application domain.

- First of all, unlike the EPP, the $ATPP$ does not require the number of agents in each group to be the same.
- In case of the $ATPP$, the user does not have access to the stream of random queries. Rather, the only available data is the set of instances when the appraisal of one agent concurs with that of another. It is thus apparent that we have to artificially "generate" a sequence of "queries" (or pairs) which can be used to operate on a machine similar to the OMA. The above restriction has a "two-edged" implication. First of all, in the EPP, the user usually requests the system to obtain a query pair of the form $\langle O_i, O_j \rangle$. However, in the $ATPP$, it is our responsibility, while designing the algorithm, to determine which agents should be deemed similar or dissimilar, and the reader will observe that this determination is a problem to be solved in its own right. Secondly, in the OMA, the placement of the objects in the automaton and the stream of random queries, together, serve to either reward or penalize the automaton. However, in the case of the $ATPP$, the question of obtaining a reward/penalty response is not provided by the user, but it has to be *inferred*. This again has to be solved.
- Unlike the EPP, which has no way of penalizing "non-accessed elements", a solution to the $ATPP$ must develop a strategy for penalizing such agents by considering how similar the agents within the same groups are. Clearly, this is superfluous for the EPP because, in that problem, the automaton is absolutely dependent on the user's queries. In the present problem, it is crucial that an automaton can quantify how fitting an agent is for any given group.
- The optimal partition for the EPP yields crucial information in the stream of random queries. As opposed to this, in the context of the $ATPP$, the system has no notion of how to characterize the optimal partition. This renders the problem of adapting the OMA to solve the $ATPP$ more difficult.
- In the same vein, the definition of the optimal partition for the EPP is quite different from that of the corresponding solution for the $ATPP$. In the case of the EPP, all objects which are accessed together more frequently should be in the same partition, while in the $ATPP$ all agents which *respond* in a similar way should be in the same group.
- The criteria which are used to reward and penalize the automaton in the EPP is quite unlike the one used for the $ATPP$. In the EPP, the automaton is rewarded or penalized based on the (unknown) probability of any two objects being jointly accessed. But in the context of the $ATPP$, the automaton is reward or penalized by "conducting a comprehensive study" of the relation between the individual agents, which, furthermore, may or may not participate in a communication within the social network.
- Although the EPP and $ATPP$ utilize analogous rules for a reward phenomenon, as we shall see, they differ in performing the penalty rules. In case of the EPP, the automaton enforces the rule that the pertinent object migrates, if and only if at least one of the accessed objects is at the boundary state of the different partitions. As opposed to this, in the $ATPP$, the automaton enforces the rule that the agents are migrated to the alternate group whenever the task of migrating is dictated by their *joint* appraisal probabilities.

- The automaton used to solve the EPP is said to have converged, when all the objects are found in the most (or the last two) internal states of each partition. However, we propose that the convergence in the $ATPP$ occurs when the measure of their clusters is satisfactory. For example, if all the agents in both class are in their most internal states, the within-cluster distance of each cluster would be zero, and the between-cluster distance would be $2M$. We can then say that convergence has occurred if the weighted sum of the within-cluster distance and their between-cluster distance is greater than a given threshold.

4 A LA-based solution to the $ATPP$

This section describes, in fair detail, all the aspects and algorithmic issues associated with our LA-based solution to the $ATPP$. To initiate this, in Sect. 4.1 we first explicitly state the inputs, outputs and goals of the entire exercise. Then, in Sect. 4.2, we present a formal overview of the solution, including the components of its 7-tuple formulation. Each of the elements of this automaton are formally explained here. Since the formulation of the LA has to explicitly declare the responses it makes to Rewards and Penalties, Sect. 4.3 explicitly defines these both diagrammatically and algorithmically. This is followed, in Sect. 4.4, by a description of how the user concentrates his observation on a sliding window of interactions with other agents in the social network, and a simple example (in Sect. 4.6) concludes this section.

4.1 Inputs, outputs and goals

In order to develop our LA-based solution to the $ATPP$, it is appropriate that we re-iterate what the corresponding inputs and outputs are. The input to our automaton, is the set of user agents $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ and the reports that are communicated within the social network. With regard to the output, we intend to partition $\mathcal{U}$ into two sets, namely, the set of fair agents $\mathcal{U}_f = \{u_i | p_i = p_f\}$, and the set of deceptive agents $\mathcal{U}_d = \{u_i | p_i = p_d\}$. The intuitive principle that we use is that the agents that have the same nature (fair or deceptive) will report similar experiences about the same service, and we shall attempt to infer this phenomenon to, hopefully, migrate them to the same partition. Observe that since agents of different nature report dissimilar experiences about the same service, we hope to also infer *this*, and, hopefully, force them to converge into different partitions.

4.2 Formal definition of the LA-based $ATPP$ solution

We define the Agent Migrating Partitioning Automaton ($AMPA$) as a 7-tuple as below: ($\mathcal{U}$, $\underline{\Phi}$, $\underline{\alpha}$, $\underline{\beta}$, $\mathbb{Q}$, $\mathbb{G}$, $\mathbb{W}$), where

- $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ is the set of agents.
- $\underline{\Phi} = \{\phi_1, \phi_2, \ldots, \phi_{2M}\}$ is the set of states.
- $\underline{\alpha} = \{\alpha_1, \alpha_2\}$ is the set of actions, each representing a group into which the elements of $\mathcal{U}$ fall.
- $\underline{\beta} = \{‘0’, ‘1’\}$ is the set of responses, where ‘0’ represents a *Reward*, and ‘1’ represents a *Penalty*.
- $\mathbb{Q}$ is the transition function, which specifies how the agents should move between the various states. This function is quite involved and will be explained in detail presently.
- The function $\mathbb{G}$ partitions the set of states for the groups. For each group, α_j, there is a set of states $\{\phi_{(j-1)M+1}, \ldots, \phi_{jM}\}$, where M[6] is the depth of memory. Thus,

$$G(\phi_i) = \alpha_j \quad (j-1)M + 1 \le i \le jM. \tag{2}$$

  This means that the agent in the automaton chooses α_1 if it is in any of the first M states, and that it chooses α_2 if it is in any of the states from ϕ_{M+1} to ϕ_{2M}. We assume that $\phi_{(j-1)M+1}$ is the most internal state of group α_j, and that ϕ_{jM} is the boundary state. These are called the states of “*Maximum Certainty*” and “*Minimum Certainty*”, respectively.
- $\mathbb{W} = \{W_l^D(t)\}$, where, $W_l^D(t)$ = {Last D records prior to instant t relative to service S_l }.

  Our aim is to infer from $\mathbb{W}$ a similarity list of agents deemed to be collectively similar. From it we can, based on the window of recent events, obtain a list of pairs of the form $\langle u_i, u_j\rangle$ deemed to be similar. The question of how $\mathbb{W}$ is obtained will be discussed later.

To see how all these components flow together, we shall now explain how the learning cycle is performed—which is the central kernel where the $ATPP$ is solved.

The learning phase is the core of the clustering. The $AMPA$ model is initialized by placing all the agents at the boundary state of their initially randomly-chosen groups. This indicates that the $AMPA$ is initially uncertain of the placement of the agents, because the different states within a given group quantify the measure of certainty that the scheme has for a given agent belonging to that group. As the learning cycle proceeds, similar agents will be rewarded for their being together in the same group, and they will be penalized by either moving toward their boundary state, or to another group, as will be clarified presently.

[6]Generally speaking, the depth of the memory, M, in the LA could play an important role in determining the accuracy of the LA, while the eigenvalues of the underlying chain would determine the machine’s rate of convergence. To the best of our knowledge, we are not aware of any method used to determine the best value for M except the trial-and-error approach. The effect of varying M will be explained in Sect. 5.

4.3 Reward and penalty transitions

Philosophically, we mention that since we require that all the elements of $\mathcal{U}$ move among the states of the machine, it is distinct from traditional FSSA, which, being a traditional finite state machine, always finds itself in only one of a finite number of states. Also, if agent u_i is in action α_j, it signifies that it is in the sub-partition whose index is j. Moreover, if the states occupied by the nodes are given, the sub-partitions can be trivially obtained by invoking (2).

Let $\zeta_i(t)$ be the index of the state occupied by agent u_i at the t^{th} time instant. Based on $\{\zeta_i(t)\}$ and (2), let us suppose that the automaton decides a current partition of $\mathcal{U}$ into sub-partitions. Using this notation we shall later describe the transition map of the automaton. Since the intention of the learning process is to collect "similar" agents into the same sub-partition, the question of "inter-agent similarity" (i.e., inferring which two agents should be grouped together) is rather crucial. In the spirit of Theorem 1, we shall reckon that two agents are similar if they are of the same nature, implying that the corresponding probability of them agreeing is greater than 0.5.

We now consider the reward and penalty scenarios separately.

4.3.1 Transitions for rewards

(a) Whenever two agents u_i and u_j test the same service, if their corresponding reports are identical (either both '0' or both '1'), and they currently belong to the same partition, the automaton (and, in particular, u_i and u_j) is rewarded. This mode of rewarding is called the *RewardAgreeing* mode depicted in Fig. 1, and the algorithm is formally given in Algorithm 1 titled "Reward_Agreeing_Nodes".

(b) As opposed to this, if u_i and u_j are dissimilar and they currently belong to distinct partitions, the automaton (and again, in particular, u_i and u_j) is rewarded. This mode of rewarding is called the *RewardDisagreeingMode*. The algorithm is identical to the formal algorithm given in Algorithm 1, and is thus omitted to avoid repetition.

By way of explanation, more specifically, on being rewarded, since the agents u_i and u_j are in the "correct" group, say α_p, with regard to the agent it is being compared with, both of them are moved towards the most internal state of that group, one step at a time. Observe that it does not matter whether either (or both) of them is in a boundary state. The overall scheme is given in Algorithm 2 titled "Reward_Agent".

4.3.2 Transitions for penalties

Here we encounter three cases as listed below.

Algorithm 1 Procedure Reward_Agreeing_Nodes

Input: ζ_i and ζ_j: the indices of the states where the agents R_i and R_j are located in the LA.
Output: The updated values of ζ_i and ζ_j.
Method:

```
1: Reward_Agent(ζ_i)
2: Reward_Agent(ζ_j)
3: return ζ_i and ζ_j
```

End Procedure Reward_Agreeing_Nodes

Algorithm 2 Procedure Reward_Agent

Input: ζ_i, which represents the index of the state where the agent R_i is located in the LA.
Output: The updated value of ζ_i.
Method:

```
1: if (ζ_i mod M ≠ 1) then
2:     ζ_i = ζ_i − 1
3: end if
4: return ζ_i
```

End Procedure Reward_Agent

(a) Whenever two agents u_i and u_j test the same service, if their corresponding reports are identical (either both '0' or both '1'), and they currently belong to distinct partitions, the automaton (and, in particular, u_i and u_j) is penalized. More specifically, this case is encountered when two similar agents, u_i and u_j, are allocated in distinct groups say, α_a and α_b respectively (i.e., R_i is in state ζ_i, where $\zeta_i \in \{\phi_{(a-1)M+1}, \ldots, \phi_{aM}\}$, and R_j is in state ζ_j, where $\zeta_j \in \{\phi_{(b-1)M+1}, \ldots, \phi_{bM}\}$). This mode of rewarding is called the *PenalizeAgreeing* mode depicted in Fig. 2, and the algorithm is formally given in Algorithm 3 titled "PenalizeAgreeingNodes".

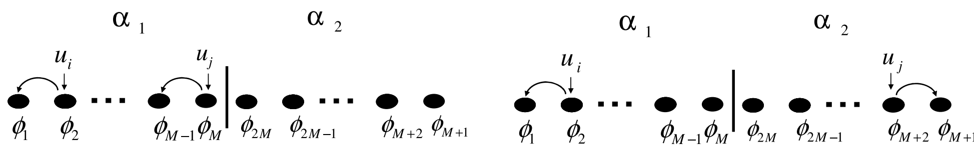
Algorithm 3 Procedure Penalize_Agreeing_Nodes

Input: ζ_i and ζ_j: the indices of the states where the agents R_i and R_j are located in the LA.
Output: The updated values of ζ_i and ζ_j.
Method:

```
1: Penalize_Agent(ζ_i)
2: Penalize_Agent(ζ_j)
3: return ζ_i and ζ_j
```

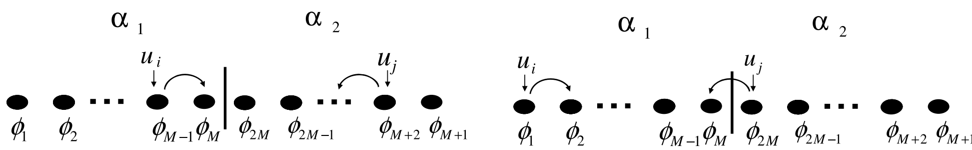
End Procedure Penalize_Agreeing_Nodes

(b) However, if u_i and u_j are both assigned to the same sub-partition, but they should rather be assigned to distinct groups, the automaton is penalized. Analogous to the above, this mode of penalizing is called the *PenalizeDisagreeing* mode, because, in this mode, agents which are



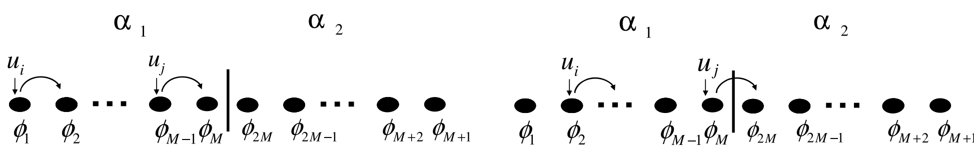
(a) Sim_1 (b) Sim_2

Fig. 1 (**a**) RewardAgreeing Mode: This is the case when both agents u_i and u_j belong to the same partition. Observe that it does not matter whether either (or both) of them is in a boundary state. (**b**) Reward Disagreeing Mode: This is the case when both agents u_i and u_j belong to different partitions. Again, observe that it does not matter whether either (or both) of them is in a boundary state



(a) Sim_1 (b) Sim_2

Fig. 2 PenalizeAgreeing Mode: This is the case when both agents u_i and u_j belong to different partitions. In (**a**), neither of them is in a boundary state. As opposed to this, in (**b**), the figure depicts the case when one of them, u_j, is in a boundary state



(a) Sim_1 (b) Sim_2

Fig. 3 PenalizingDisagreeing mode: This is the case when both agents u_i and u_j belong to the same partition, when, in actuality, they should belong to distinct partitions. In (**a**), neither of them is in a boundary state. As opposed to this, in (**b**), the figure depicts the case when one of them, u_j, is in a boundary state

Algorithm 4 Procedure Penalize_Agent

Input: ζ_i, which represents the index of the state where the agent R_i is located in the LA.
Output: The updated value of ζ_i.
Method:

```
if (ζi mod M ≠ 0) then
    ζi = ζi + 1
else
    if (ζi = M) then
        ζi = 2M
    else
        ζi = M
    end if
end if
return ζi
```

End Procedure Penalize_Agent

actually dissimilar are assigned to the same subpartition and they are therefore penalized. This is depicted in Fig. 3, and the algorithm is identical to the formal algorithm given in Algorithm 3. It is again omitted to avoid repetition.

(c) In both these cases, if the agent in question is in a boundary state, it is subsequently moved to the boundary state of the alternate choice.

Again, by way of explanation, on being penalized, if the agents u_i and u_j are in the same group, say α_a, both of them are moved *away* from the most internal state of that group, one step at a time. See Figs. 2 and 3 and Algorithm 4 titled "Penalize_Agent". If either of them is in a boundary state, it switches actions to go to the boundary state of the alternate action.

4.4 The window of observations

Since we adopt a simple interaction model, at each time instant a random agent chooses a random service to interact with, and can report his experience to the rest of agents.

We now address the question of recording the reports associated with the various agents, which in turn, involves the set $\mathbb{W}$, defined above. In our work, we adopt a "tuple-based" window to store the reports for a given service [12], where the size of the window is quantified in terms of the number of tuples. We have opted to do this because it is easier to deal with tuple-based windows, since the size of each window in terms of the number of tuples is fixed. As opposed to this, a time-based window would be specified in terms of time units, where the size of each window instance may vary based on the arrival process. Observe that our approach is consistent with the work of Shapiro [23] where it was proven that in an environment in which peers can change their behavior over time, the efficiency of a reputation mechanism is maximized by giving higher weights on recent ratings and where older (stale) ratings are discounted. Clearly, this is equivalent to enforcing a sliding window.

In the partitioning phase, the agent in question observes the reports of other agents. Based on the similarity of the reports relative to the same service, the agent in question partitions the reporters into two sets. Obviously, the reporters are either deceptive or fair. Despite the ability of our LA-based clustering algorithm to separate between the two groups $\mathcal{U}_f$ and $\mathcal{U}_d$, the agent can not determine which of the two groups is the fair one, ($\mathcal{U}_f$), and which is the deceptive one, ($\mathcal{U}_d$), unless he tries the services himself. Intuitively, if the agents knows this information, he can just take the inverse of the reports of the "liars" as being trustworthy, while he considers the rest of the reports, obtained from the fair agents, as also being trustworthy.

With regard to the set $\mathbb{W}$, the decision making procedure that maximizes the likelihood of choosing high performance services, works as follows. Every agent stores the last D reports seen so far. Thus, the agent in question maintains, for every service, a sliding window over the last D reported experiences, which guarantees gathering the most recent reports. Let

$$W_l^D(t) = \{\text{Last } D \text{ records prior to instant } t \\ \text{relative to service } S_l\}$$

At time instant t, $W_l^D(t)$ contains the D tuples with the largest time stamps (where, if the total number d of reports seen so far is smaller than the length of the window D, the vector contains these d elements). Clearly, $W_l^D(t)$ stores the most recent D tuples.[7] Also, let $W_l^D[k]$ denote the record of index k in the vector, or the $(D-k)^{th}$ last record.

4.5 The decision making phase

In the spirit of what we have developed so far, we assume that the services belong to two categories: high performance services and low performance services. A high performance service is a service with a high[8] value of θ, and similarly, a low performance service is a service with a low value of θ. We suppose that agent u aspires to interact with high performance services. Therefore, every time u desires to access a service, u creates a list L of the recommended services by applying a majority voting method, as explained below. Based on this, u chooses a random service among the elements of this created list. In order to create a list of the high performance services, for every service S_l, agent u evaluates the feedback from agents that may have directly interacted with service S_l during the last D interactions. We adopt the terminology of a "witness" to denote an agent solicited for providing its feedback. In this sense, at instant t, agent u examines the service history vector $W_l^D(t)$ that contains the last reports of the witnesses regarding the performance of service S_l. For every report in the vector $W_l^D(t)$, agent u should take the reverse of the report as true if he believes that the witness is a "liar", and consider the rest of the reports as being trustworthy. Following such a reasoning, given D trustworthy reports about a given service, we can apply a deterministic majority voting to determine if the service is of high performance or of low performance. Obviously, if the majority of the agents assign the service a ranking of '1', the service is assumed to be of a high performance, and consequently, it is added to the list L.

However, a potential question is that of determining which partition is the deceptive one, and which involves the fair agents. In order to differentiate between the partitions we design a LA that learns which of the partitions is deceptive and which is fair—based on the result of the interaction between agent u with the selected service S_l. The automaton is rewarded whenever agent u selects a recommended service from the list L and the result of the interaction is a high performance (meaning '1'). Similarly, the automaton is penalized whenever agent u selects a recommended service from the list L and the result of the interaction is a low performance (meaning '0'). Again, we suppose that agent u in question is initially assigned to the boundary state. We observe the following:

- If agent u is in class α_j then u supposes that all the agents in α_j are fair, and the agents in the alternate class are deceptive.
- Whenever agent u decides to interact with a high performance service, he creates the list L of recommended services, and proceeds to choose a random service from L.
- If the result of the interaction is '1', a reward is generated, and the agent u goes one step towards the most internal state of class α_j.

[7] In future, unless there is ambiguity, for ease of notation, we shall omit the time index t.

[8] In the section which describes the simulations performed, a typical value that we choose for high performance services is $\theta = 0.8$, and for low performance services is $\theta = 0.2$.

- If the result of the interaction is '0', a penalty is generated and agent u goes one step towards the boundary state of class α_j.
- If agent u is already in the boundary state, he switches to the alternate class.

The formal algorithm which puts all the pieces of this puzzle together, follows in Algorithm 5. Observe that the agent whom we are interested in (say, u, who is in state ξ) invokes this. He periodically,[9] with a periodicity T, invokes the *Service Selection* module given formally in Algorithm 6. Since, as we alluded to earlier, we are working with a simple interaction model, our solution assumes that at every time instant, a random agent u_i is allowed to experience the quality of a random service, and that he communicates his experience to the rest of the network. This report will serve as an input to the *OMA-Based-Partitioning*, given formally in Algorithm 7. Consequently, the agent u will be able to continuously invoke an "intelligent" partitioning strategy between his consecutive accesses to the services, and also incrementally partition the set of reporting agents. The automaton associated with the agent u will converge to the action which yields the minimum penalty response in an expected sense. In our case, the automaton will converge to the class containing the fair agents, while the deceptive agents will converge to the alternate class.

Algorithm 5 Main_Algorithm

Input: $\mathcal{U} = \{u_1, \ldots, u_N\}$, the set of agents, and T, a parameter which specifies the access periodicity of the system.
Output: Choice of an accessed service at every T^{th} time instant.
Method:

for Every time instant n **do**
  if ((n mod $T = 0$)) **then**
    Service_Selection($\mathcal{U}$)
  else
    OMA_Based_Partitioning($\mathcal{U}$)
  end if
end for
End Main_Algorithm

4.6 Example

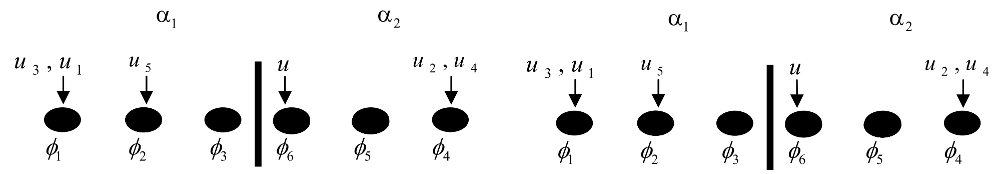
To show how the proposed decision making works we provide an example (see Fig. 4). In this example, we suppose that the current partition is depicted by Fig. 4(a). Agents u_1, u_3 and u_5 belong to a different partition than agents u_2 and u_4. We suppose that the agent in question, u, desires to interact with a high performance service. u examines the report vector $W_l^D(t)$. Moreover, we suppose that $W_l^D(t)$ is composed of $\{y_{1l} = 0, y_{3l} = 0, y_{5l} = 0, y_{2l} = 1, y_{4l} = 0\}$. Since u is in state ϕ_6, it belongs to partition α_2, and therefore assumes that agents u_2 and u_4 are fair agents, since they belong to the same partition as he. Similarly, u assumes that agents u_1, u_3 and u_5 are deceptive. Therefore, u inverts the reports received from the "assumed deceptive agents", namely ($y_{1l} = 0, y_{3l} = 0, y_{5l} = 0$) and trusts the reports from the "assumed fair agents", namely ($y_{2l} = 1, y_{4l} = 0$). Applying the majority voting method, four 1's and one 0, leads to a majority of 1, meaning that the service S_l is assumed a priori to be a high performance service. We suppose that u

Algorithm 6 Procedure Service_Selection

Input: A current partition of $\mathcal{U}\{u_1, \ldots, u_N\}$ into subpartitions. Note that ζ, which represents the state occupied by agent u.
Output: The updated value of ζ.
Method:

for every service S_l in the pool of available services **do**
  Initialize $vote \leftarrow 0$
  Initialize $L \leftarrow \emptyset$
  W_l^D: report vector relative to S_l
  for $k \leftarrow 1$ to $D - 1$ **do**
    $j \leftarrow$ Index of the agent associated to record $W_l^D[k]$
    if ((u and u_j are in same group)$\wedge$($W_l^D[k] = 1$)) **then**
      $vote := vote + 1$
    else
      if ((u and u_j are in different groups)$\wedge$($W_l^D[k] = 0$)) **then**
        $vote := vote + 1$
      end if
    end if
  end for
  if ($vote \geq D/2$) **then**
    $L \leftarrow L \cup S_l$
  end if
end for
S_r: Service chosen at random from the list L
x_r: Result of the interaction as u accesses S_r
ζ: Index of the state occupied by agent u
if ($x_r = 1$) **then**
  Reward_Agent(ζ)
else
  Penalize_Agent(ζ)
end if
End Procedure Service_Selection

[9] Here, we suppose that the agent u aspires to access a high performance service with a pre-defined fixed frequency, for example, after every T time instances. Observe that we could just as easily have resorted to a Poisson distribution.



(a) Sim_1 (b) Sim_2

Fig. 4 An example of how the decision making process works. In (**a**) we encounter the so-called *PenalizeAgreeing* mode. Here the agents u_i and u_j belong to different partitions, and neither of them is in a boundary state. As opposed to this, in case (**b**), the LA encounters the *PenalizeAgreeing* mode, when both agents u_i and u_j belong to different partitions, and one of them, u_j, is in a boundary state

Algorithm 7 Procedure OMA_Based_Partitioning

Input: $U = \{u_1, \ldots, u_N\}$ is the set of agents to be partitioned.
Output: Partitioning agents into two sub-partitions.
Method:

A random agent i chooses a random service S_l
x_{il}: Result of the interaction
y_{il}: Reported experience
Update the vector W_l^D
for $k \leftarrow 1$ to $D - 1$ **do**
  $j \leftarrow$ Index of the agent associated to record $W_l^D[k]$
  if $(y_{il} = y_{jl})$ **then**
    if (u_i and u_j are in same group) **then**
      Reward_Agreeing_Nodes(i, j)
    else
      Penalize_Agreeing_Nodes(i, j)
    end if
  else
    if (u_i and u_j are in same group) **then**
      Penalize_Disagreeing_Nodes(i, j)
    else
      Reward_Disagreeing_Nodes(i, j)
    end if
  end if
end for
End Procedure OMA_Based_Partitioning

selects S_l after creating the list L. In addition, suppose that the result of the interaction is ranked as being a "low" performance. Therefore, the automaton depicted in Fig. 4(b) is penalized. Since u is at the boundary state, ϕ_6, u switches from its current partition, and is assigned to the state ϕ_3. Being in state ϕ_3 (and consequently in partition α_1), u assumes that agents u_2 and u_4 are deceptive agents since they now belong to a different partition than he. Similarly, u now assumes that agents u_1, u_3 and u_5 are fair, and the system is ready for the next interaction!

5 Experimental results

The performance[10] of our scheme for RSs has been tested by simulation in a variety of parameter settings, and the results that we have obtained are truly conclusive. To quantify the quality of the scheme, we measured the average performance of the selected services over all interactions, and this was used as the performance index. All the results reported have been obtained after averaging across 1,000 runs, where every run consisted of 40,000 steps. In other words, each time i the agent in question, guided by our learning scheme, selects a service, the resulting performance x_i is either 0 or 1. Thus, assuming that the agent selects a total of n services across all runs and time steps, we define the average performance to be $\sum_{i=1}^{n} x_i$.

The interactions between the agents and the services were generated at random, and at every time instant, a random agent was made to select a random service. In our current experimental setting, the number of agents was 20 and the number of services in the pool of available services was 100. The services belonged to two categories, namely, high performance services with $\theta \geq 0.5$, and low performance services with $\theta \leq 0.5$. The agent in question (i.e., the one which we are interested in) periodically accessed a service every 1,000 runs—as per the above-mentioned procedure.

We report now the results obtained by testing the scheme in a variety of settings.[11]

5.1 Performance in static environments

We first report the results for environments which are static. Figure 5 shows the average performance of the service selection scheme when the testing was done over 40,000 runs. In this particular setting, 10% of the services were high

[10] We are grateful to the feedback from the anonymous Referees, whose comments helped to improve this section significantly.

[11] We have done experiments for numerous settings and scenarios. In the interest of brevity, we merely report a few representative (and typical) experimental results, so that the power of our proposed methodology can be justified.

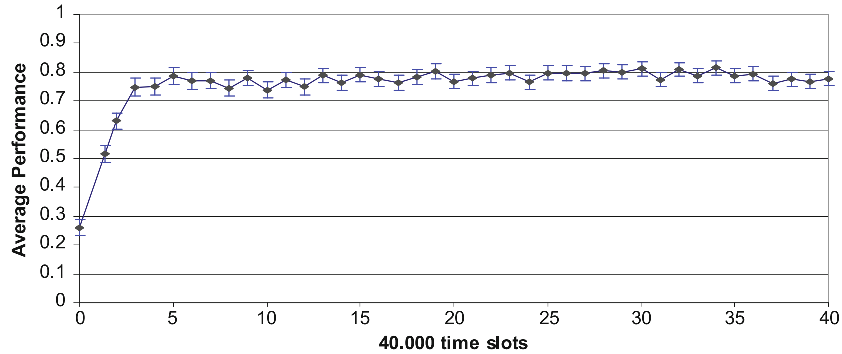


Fig. 5 The behavior of the $AMPA$, measured in terms of the average performance, in an environment when the behavior of the agents is static

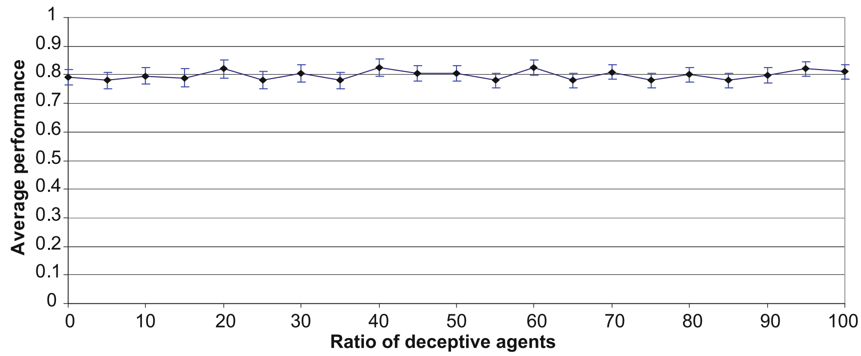


Fig. 6 The behavior of the $AMPA$ in an environment when the ratio of fair/deceptive agents is varied. The figure shows the variation of the average performance under different ratios of deceptive agents

performance services with $\theta = 0.8$, and 90% were low performance services with $\theta = 0.2$. Further, 15 of the reporting agents were deceptive with $p_d = 0.2$, while 5 were fair agents with $p_f = 0.8$. The depth of memory used for the LA was $M = 10$, and the length of the sliding window was 100. The results that we have obtained are shown in Fig. 5, which demonstrates the ability of the approach to accurately infer correct decisions in the presence of the deceptive agents. In Fig. 5, 95% confidence intervals for the averages performance are also plotted. Observe that the scheme achieves a near-*optimal* index that asymptotically approaches the performance of the high performance services, i.e., $\theta = 0.8$.

5.2 Immunity to the proportion of deceptive agents

We now consider the problem of investigating how "immune" our system is to the percentage of deceptive agents. Figure 6 presents the average performance of the system (over all interactions) when the ratio of deceptive agents is varied. Observe that Fig. 6 are also reports the 95% confidence intervals for these averages performance indices. The reader will agree that the simulations results demonstrate that the scheme is truly "immune" to varying the proportions of fair and deceptive agents. In fact, even if all agents are deceptive (i.e., this is equivalent to a ratio of 100%), the average performance is stable and again achieves near-optimal values that approach the index of the high performance services, $\theta = 0.8$. In our opinion, this is quite remarkable!

5.3 Varying the spread between deceptive and fair agents

To further demonstrate the power of the scheme, we have considered the effect of varying the spread between deceptive and fair agents. To analyze this, in this experiment, 50% of the services provided were set to be "high performance" services. Figure 7 displays the average performance when p_f and p_d were set to the following pairs: (0.8, 0.2), (0.6,0.4), and (0.5,0.5). The reader should observe that as the spread between the fair and deceptive agents is decreased, the environment becomes increasingly more "difficult", rendering the task of differentiating between them to be more exacting. In the particular case where $p_f = 0.5$ and $p_d = 0.5$, the process of choosing the services is totally random, which results in a theoretical performance of 0.5, because,

$$P(\theta = 0.8) \times 0.8 + P(\theta = 0.2) \times 0.5$$
$$= 0.5 \times 0.8 + 0.5 \times 0.2 = 0.5.$$

However, in every case, the $AMPA$ seems to asymptotically attain near-optimal solutions.

5.4 Periodically changing service performance

To investigate the behavior of the $AMPA$ with performances which changed with time, we first considered the scenario when these changes were made periodically. Indeed, we achieved this by changing all the service performances periodically every 5,000 runs. Further, the changes

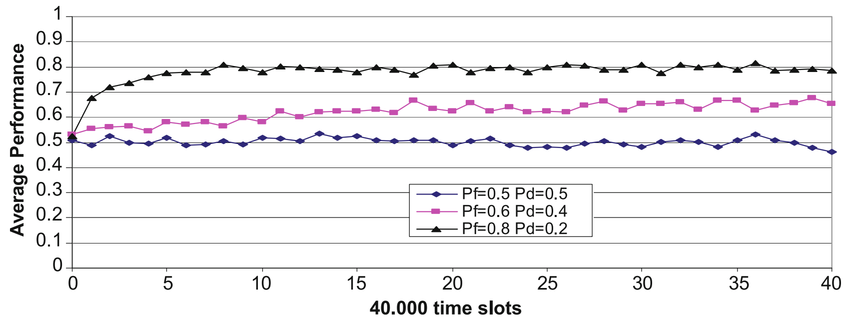


Fig. 7 The variation of the performance of the $AMPA$ with increasing spreads between the trustworthiness of deceptive and fair agents

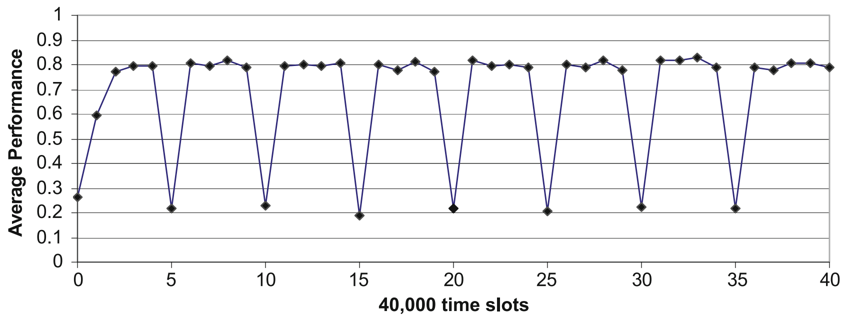


Fig. 8 The performance of the $AMPA$ with periodically changing service performances

were made "drastic", i.e., by inverting them from their prior values as per:

$$\theta_{l,new} = 1 - \theta_{l,old}.$$

In the simulation settings, we used the following parameters: There were 10% high performance services with $\theta = 0.8$, and 90% low performance services with $\theta = 0.1$. Further, we assumed that 15 of the reporting agents were deceptive, with $p_d = 0.2$, while 5 were fair agents with $p_f = 0.8$. The memory depth used for the LA was $M = 10$, and the length of the sliding window length was 100. From the results shown in Fig. 8, the reader will observe that the scheme is able to adapt favorably to such changes. Indeed, from Fig. 8, we notice that as the behavior of the services changed (i.e., at every 5,000th step), the subsequent access by agent u resulted in choosing a low performance service. However, the scheme was well able to adapt to that change, and that, rather rapidly, because of the state changes of the $AMPA$ and the sliding window approach. In fact, as the window slides, the reports related to older (i.e., "stale") service performances were discounted, and replaced by more recent reports, which, in turn, better reflected the current performance of the services. Again, we believe that the way by which the $AMPA$ tracks this change is quite remarkable.

5.5 Immunity to the ratio of low performance services

The next scenario we report is the $AMPA$'s immunity to the ratio of low performance services. Simulations results demonstrate that the scheme is immune to varying this ratio. In this case, we observe that for the system to achieve near-optimal performance, the time window should be chosen to be relatively large. In fact, it turns out that the agent in question still opts to choose high performance services even if their ratio is as low as 10%. Figure 9 depicts the average performance (over time) with variations in the ratio of low performance services. In our experiments, we utilized the following ratios: 10%, 30%, 50%, 70%, and 90%, and in the simulation, 15 of the reporting agents were deceptive with $p_d = 0.2$, while 5 of the agents were fair with $p_f = 0.8$. The reader will observe that the $AMPA$ converges to near-optimal performance in *every* single setting!

5.6 Varying the spread between high/low performance services

To further demonstrate the property of the $AMPA$ to be insensitive to the spread between high and low performance services, we conducted a set of experiments in which these parameters were varied. Figure 10 depicts the average performance when the indices of the high and low performance services were set to the following pairs: $(0.8, 0.2)$, $(0.7, 0.3)$, $(0.6, 0.4)$ and $(0.5, 0.5)$ respectively. In this experiment, 50% of the services provided were set to be "high performance" services. The reader will observe that as the spread decreased, it was (understandably!!) increasingly difficult for the scheme to accurately select only high performance services. Of course, in the limit when we encounter the particular case of $(0.5, 0.5)$, the process of choosing the services turns out to be totally random, which resulted in a theoretical performance of 0.5.

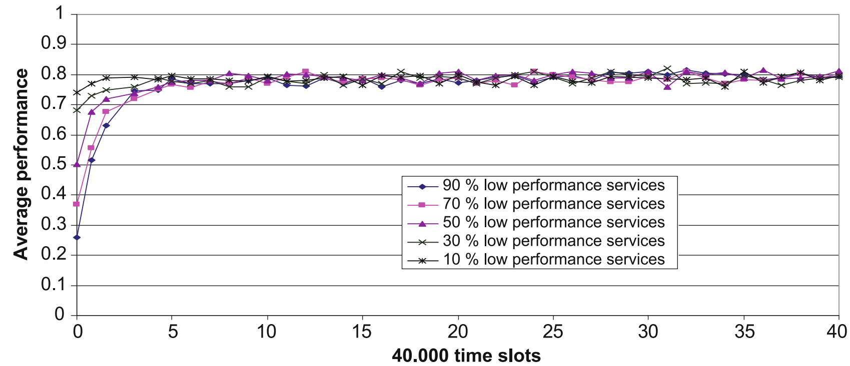


Fig. 9 The performance of the $AMPA$ with variations in the ratio of low performance services

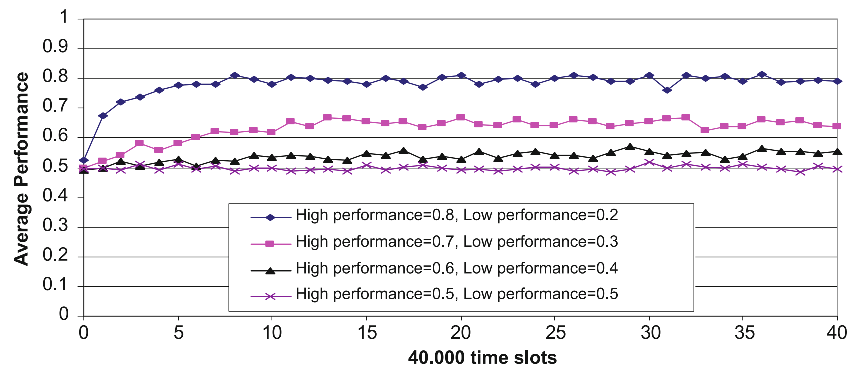


Fig. 10 The performance of the $AMPA$ as the spread between the high and low performance services is decreased

5.7 Effect of changing the memory size

The final result that we report confirms the basic theory of LA, which asserts that the performance of the machine increases with the size of the memory. To test the effect of the machine's memory, numerous experiments were conducted in a number of environmental settings. In this experiment, 50% of the services provided were set to be "high performance" services. The results obtained in every case was identical, namely, that the performance increased with the memory. Indeed, Fig. 11 illustrates that decreasing the memory depth from 5 to 2 results in a lower performance. Again, as anticipated by the theoretical results, a smaller memory undermines the quality of the partitioning process, making it difficult for the $AMPA$ to accurately differentiate between the deceptive and fair agents.

5.8 Experimental comparison

In this section, we compare[12] our proposed approach with two popular algorithms for dealing with deceptive agents, namely Yu and Singh's weighted majority method [28], and Sen and Sajja's approach [22]. The main idea of the algorithm in [28] is to assign a weight to every witness that reflects how credible he is. Before accessing a service, the agent in question, u, requests the predictions of the individual witnesses concerning the service performance. The witnesses convey their predictions to u in the form of belief functions [28]. After accessing the service, the agent in question updates the weight of every witness based on the result of interacting with the service. Deceptive agents will tend to submit inaccurate predictions and thus their relative weights will decrease over time. Similarly, the weights of fair agents will increase over time. An aggregated prediction, denoted as λ in [28], is computed by the agent in question as a weighted combination of the witnesses' predictions. In order to be able to compare our algorithm with the scheme proposed in [28], we were forced to add a *Service Selection Procedure* to Yu and Singh's weighted majority method, i.e., one that is analogous to the service selection procedure presented in our Algorithm 6. In the service selection procedure, whenever agent u desires to access a service, u creates a list L of the recommended services that have an aggregated prediction λ greater[13] than $\frac{1}{2}$.

In Fig. 12, we report the evolution of the average performance for our approach, and compare it with Yu and Singh's weighted majority algorithm. In the simulation settings, we used the following parameters: There were 10% high performance services with $\theta = 0.8$, and 90% low performance services with $\theta = 0.2$. Moreover, we assumed that 15 of the re-

[12] We are grateful to the anonymous referees who suggested such a comparison.

[13] Using the value $\frac{1}{2}$ as a decision threshold is commonly used in Weighted Majority Algorithms [11].

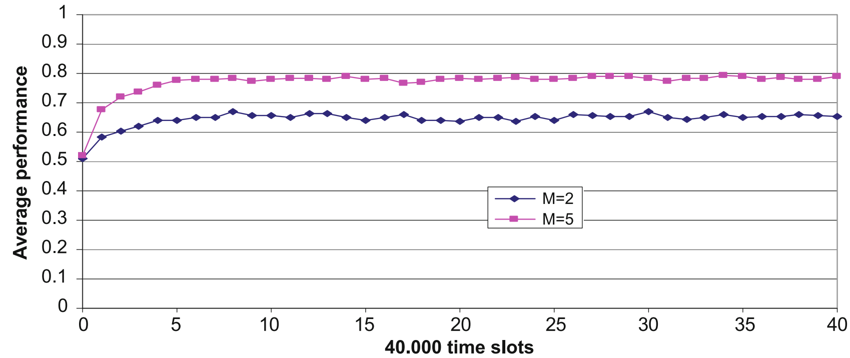


Fig. 11 The variation of the performance of the $AMPA$ by reducing the length of the memory

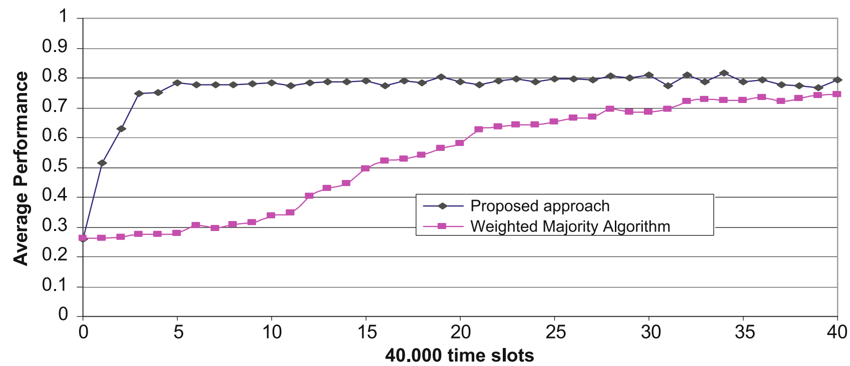


Fig. 12 The comparison of the performance of the $AMPA$ with the Weighted Majority Algorithm

porting agents were deceptive, with $p_d = 0.2$, while 5 were fair agents with $p_f = 0.8$. For the weighted majority algorithm, we adopted the number of episodes to be $H = 10$, as described in [28]. From Fig. 12, we remark that our proposed approach exhibits a faster convergence speed than the weighted majority algorithm. The slow convergence speed of the weighted majority algorithm can be explained by the gradual reduction of the weight of deceptive agents. This gradual modification of weights leads to the weighted average being "polluted" by the deceptive agents for a considerable amount of time, before their detrimental effect is filtered out.

Table 1 summarizes the average performance at time instant 20,000, for different ratios of deceptive agents, where the total number of agents was set to be 20. Table 1 shows that the average performance for the weighted majority algorithm monotonically decreases as the ratio of deceptive agents increases. This also demonstrates that the speed of convergence of the weighted majority algorithm decreases monotonically, as we increase the ratio of deceptive agents. On the other hand, Table 1 also reports that the learning speed of our proposed approach is not affected at all by the increased ratio of deceptive agents, thus demonstrating the superiority of our approach.

In addition to comparing our algorithm to the weighted majority algorithm, we have also evaluated another popular approach, namely the approach proposed by Sen and Sajja in [22]. In [22], the authors made use of a reinforcement learning technique to locate high performance service providers. A fundamental assumption in the latter approach is that the majority of agents are fair. When this assumption is true, the probability guarantee defined in [22] is obtained by querying all witnesses.[14] In Fig. 13, we report a comparison of the average performance under varying ratios of deceptive agents. In the experimental settings, 50% of the services were assumed to provide high performance. We observe that Sen and Sajja's algorithm suffers from a performance decline as the number of deceptive agents increases. As opposed to this, from Fig. 13, we observe that the average performance of our scheme remains near-optimal in every setting. It is worth noting that the robustness of our approach (when dealing with high ratios of deceptive agents) is not due to inverting the ratings of deceptive agents. In fact, we could exclusively base our predictions on the ratings of fair agents and discard the ratings of deceptive ones. However, intelligently combining the ratings from both fair and deceptive agents when evaluating the performance of a service, reduces the variance of the resulting aggregate.

5.9 Evaluation of computational efficiency

The issue of addressing the computational complexity of our solution is not easily answered. The real issue is that the problem itself is NP-Hard because it can be reduced to the

[14] In the interest of brevity, the full details of this approach are omitted. We refer the reader to [22] for further details about the scheme.

Table 1 A comparison of the performances of the $AMPA$ and the method due to Yu and Singh [28] after 20,000 time steps

Deceptive agents ratio	Weighted majority Average	Weighted majority Standard deviation	Proposed approach Average	Proposed approach Standard deviation
10%	0.786	0.013	0.796	0.013
20%	0.752	0.014	0.795	0.013
30 %	0.718	0.014	0.792	0.013
40%	0.701	0.014	0.790	0.013
50%	0.674	0.015	0.794	0.013
60%	0.635	0.015	0.793	0.013
70%	0.611	0.015	0.798	0.013
80%	0.581	0.016	0.797	0.013
90%	0.547	0.016	0.796	0.013

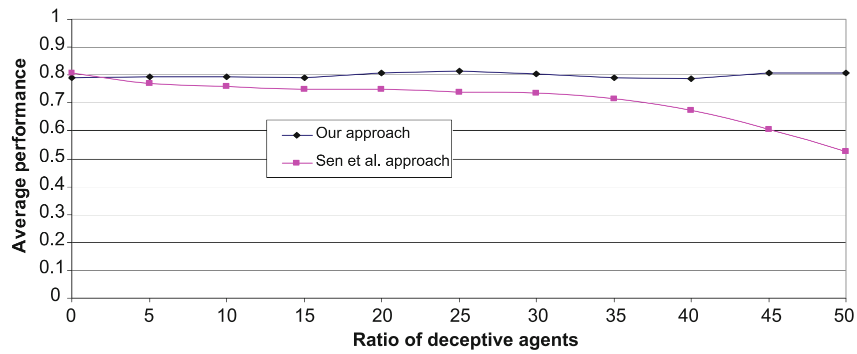


Fig. 13 The variation of the average performance of the $AMPA$ and the method of Sen and Sajja [22] when the ratio of fair agents is decreased

partitioning problem, and so, if we merely stated that at each step we required a *linear* number of operations, we would be presenting an unfair picture to the reader. Indeed, generally speaking, the time complexity of a fixed-structure LA depends on the depth of the memory, M, which plays an important role in determining the accuracy of the LA, while the consequent eigenvalues of the underlying chain (also dependent on M) would determine the rate of convergence. In practice, the LA is assumed to have converged when all objects are in (or close to) the most internal states in each partition. Even though one needs but a linear number of moves per iteration, to the best of our knowledge, we are not aware of any method that can be used to pre-determine the number of iterations required for a solution to reach these most internal states. Hence we present below, what we believe is the most appropriate (fair and scientific) method of reporting the complexity of the solution.

In order to assess the computational efficiency of our approach, we also measured the execution times on a laptop PC containing a 2.1 GHz Intel Core Duo CPU with 2 GB of RAM, running Windows XP 2002. All the algorithms were implemented using Java, and the code was compiled using the Sun Java compiler (javac). Observe that our main algorithm consists of two main procedures, namely the Service_Selection Procedure and the OMA_Based_Partitioning Procedure, whose detailed descriptions are respectively found in Algorithm 6 and Algorithm 7. In this perspective, we analyzed the computational efficiency of both these procedures separately.

Note that in all of the experiments, the execution time is expressed in milliseconds. Figure 14 depicts the evolution of the execution time of the Procedure OMA_Based_Partitioning when the size of the sliding window was varied from 10 to 100, and when we fixed the number of services to 100. From Fig. 14, we observe that the required execution time for the OMA_Based_Partitioning increases linearly as the size of the sliding window increases. Note that we achieve high accuracy with relatively small window sizes, and it is not unreasonable to reckon that this linear increase in computation time is rather insignificant in real-world applications.

In the rest of this section, we evaluate the performance of the Procedure Service_Selection. Here, we measured the execution times by varying the number of services and for varying sliding window sizes. In Fig. 15, we fixed the window size to be 100 and varied the number of services from 10 to 100. From the figure, we note that the execution time of Procedure Service_Selection increases almost linearly as we increase the number of services, which is also quite commendable!

Similarly, in Fig. 16, we display the results when the number of services was fixed to be 100, and we varied the

Fig. 14 Execution time of the Procedure OMA_Based_Partitioning with varying Sliding Window sizes

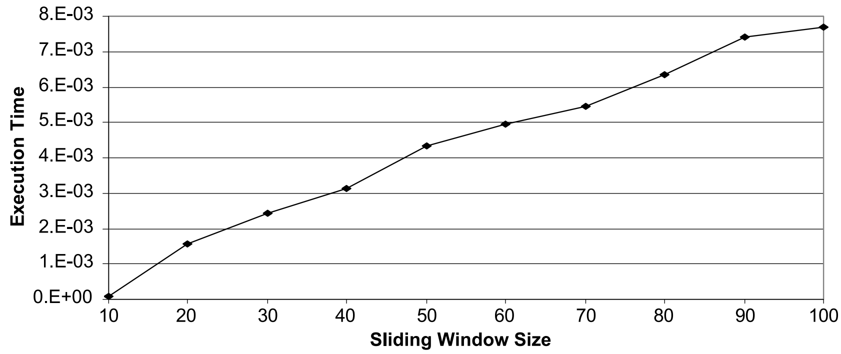


Fig. 15 Execution time of the Procedure Service_Selection with varying number of services

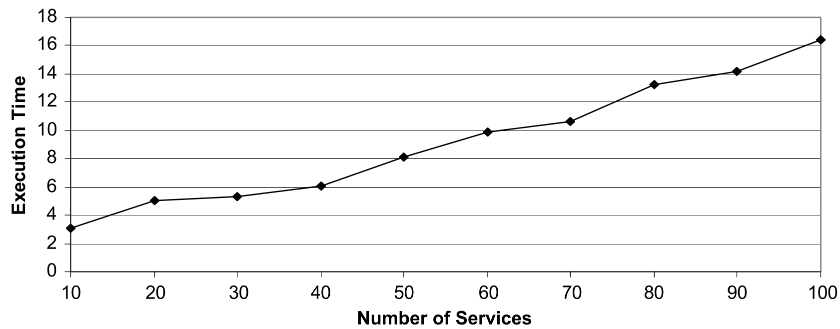
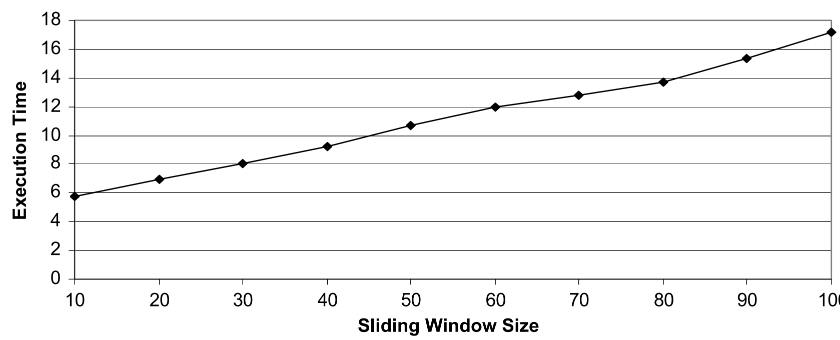


Fig. 16 Execution time of the Procedure Service_Selection with varying Sliding Window sizes



window size from 10 to 100. From this figure, we note that by increasing the window size, the execution time increases as well, again, almost linearly!

5.10 Utilizing the ATPP to real-world applications

The ultimate intention of this paper is to have the algorithm functional in a real-world device. The problems with testing and incorporating it on real-world applications are many and listed below:

1. First of all, it is extremely hard, if not impossible to get real-life data on which we can test the algorithm. Indeed, we would like to perform a real-world study with "real" good and bad services, and with "real" people taking the role of truth-tellers and liars. However, the question, really, is one of finding service providers and users who will willingly participate in such an experiment, and the logistics of this exercise is unsurmountable.
2. Secondly, even if are able to "recruit" service providers for this task, it will be impossible to find real-life users who will serve as "ballot-stuffers" or "badmouthers".
3. Even if people serve in these capacities, the results of the tests should, in actuality, be verified using psychological and cognitive criteria. Unfortunately, such a study would have to be carefully designed in order to provide reliable results, and would require human resources and knowledge in the latter domains that remain outside the scope of the present project. We respectfully submit that his is a multi-disciplinary project in its own right.
4. From an optimistic perspective, we remark that by introducing a large degree of noise in our simulations, we are able to "stress" the schemes quite severely, thus mimicking the "nuisances" of the real world. This is what we have done in this paper.
5. In this regard, we have included a comparative evaluation with other popular approaches for dealing with deceptive agents ratings, namely, Yu and Singh's weighted majority

method [28] and Sen and Sajja's reinforcement learning approach [22] (see Sect. 5.8). From these evaluations, we believe that it is clear that our scheme provides the same kind of functionality as what is required from practical cases. One should, however, note that our scheme outperforms the latter schemes in the experiments.

Finally, we conclude this section by remarking that from a more realistic viewpoint, now that the problem setup has been clarified, we believe that it is, presently, much clearer how the scheme fits into a practical case. Thus, a more restricted real-world study is planned as a next step in cooperation with *Ericsson Research*, where we intend to build a mobile phone-based prototype of the scheme presented in this paper, with a focus on learning from a real-life social network. Briefly stated, in this prototype each mobile phone will be equipped with an instance of ATPP. Furthermore, the ratings users submit about services must be accessible by each mobile phone, preferably based on lazy propagation of ratings on a per need basis. Thus, as ratings of other users are observed by a given instance of ATPP, and the respective user obtains direct experience from his own interaction with services, ATPP updates its state accordingly, as demonstrated in the simulations of this paper. Another possible practical application of our algorithm is stated in the paper by Sen and Sajja [22] where user agents need to select processor agents to achieve processor tasks. This avenue of research is also currently being investigated.

6 Conclusions

In this paper, we have considered an extremely pertinent problem in the area of "Reputation Systems" (RSs), namely the one of identifying services of high quality. Although these RSs offer generic recommendations by aggregating user-provided opinions about the quality of the services under consideration, they are prone to "ballot stuffing" and "badmouthing" in a competitive marketplace. Clearly, such unfair ratings may degrade the trustworthiness of RSs, and additionally, changes in the quality of service, over time, can render previous ratings unreliable. In this paper, we have presented a novel solution for the problem using tools provided by the family of Learning Automata (LA). Unlike most reported approaches, our scheme does not require prior knowledge of the *degree* of any of the above mentioned problems associated with RSs. Instead, it gradually learns the identity and characteristics of the users which provide fair ratings, and of those who provide unfair ratings, even when these are a consequence of them making unintentional mistakes.

Comprehensive empirical results show that our LA-based scheme efficiently handles any degree of unfair binary ratings. Furthermore, if the quality of services and/or the trustworthiness of the users change, our scheme is able to robustly track such changes over time. The paper also contains a detailed comparison of the method with the state-of-the-art. Finally, the strategy is ideal for decentralized processing, and so, as such, we believe that our LA-based scheme forms a promising basis for improving the performance of RSs.

A possible extension of our work is to develop the analogous methodology for continuous reports instead of boolean. Also, in this work, every agent in a social network can communicate with all other agents. In practice, though, most of the time, one agent may resort to his friends for partial information of available services. The question of how we can devise a solution for the service reputation effectively and efficiently (in this setting) is a research task in itself, and is an interesting problem for future work.

References

1. Agache M, Oommen BJ (2002) Generalized pursuit learning schemes: new families of continuous and discretized learning automata. IEEE Trans Syst Man Cybern, Part B, Cybern 32(6):738–749
2. Altincay H (2006) On the independence requirement in Dempster-Shafer theory for combining classifiers providing statistical evidence. Appl Intell 25:73–90. doi:10.1007/s10489-006-8867-y
3. Buchegger S, Le Boudec J (2004) A robust reputation system for P2P and mobile ad-hoc networks. In: Proceedings of the second workshop on the economics of peer-to-peer systems
4. Chen M, Singh JP (2001) Computing and using reputations for internet ratings. In: Proceedings of the 3rd ACM conference on electronic commerce, Tampa, FL, USA. ACM, New York, pp 154–162
5. Dellarocas C (2000) Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In: Proceedings of the 2nd ACM conference on electronic commerce, Minneapolis, Minnesota, USA. ACM, New York, pp 150–157
6. Despotovic Z, Aberer K (2004) A probabilistic approach to predict peers performance in P2P networks. In: Cooperative information agents VIII, pp 62–76
7. Faceli K, de Carvalho A, Rezende S (2004) Combining intelligent techniques for sensor fusion. Appl Intell 20:199–213. doi:10.1023/B:APIN.0000021413.05467.20
8. Gale W, Das S, Yu C (1990) Improvements to an algorithm for equipartitioning. IEEE Trans Comput 39(5):706–710
9. Jøsang A, Ismail R, Boyd C (2007) A survey of trust and reputation systems for online service provision. Decis Support Syst 43(2):618–644
10. Li X, Dai X, Dezert J, Smarandache F (2010) Fusion of imprecise qualitative information. Appl Intell 33:340–351. doi:10.1007/s10489-009-0170-2
11. Littlestone N, Warmuth MK (1994) The weighted majority algorithm. Inf Comput 108(2):212–261
12. Mayur Datar M, Gionis A, Indyk P, Motwani R (2002) Maintaining stream statistics over sliding windows. SIAM J Comput 31(6):1794–1813
13. Mundinger J, Le Boudec J-Y (2008) Analysis of a reputation system for mobile ad-hoc networks with liars. Perform Eval 65(3–4):212–226

14. Narendra KS, Thathachar MAL (1989) Learning automata: an introduction. Prentice-Hall, New Jersey
15. Obied A, Alhajj R (2009) Fraudulent and malicious sites on the web. Appl Intell 30:112–120. doi:10.1007/s10489-007-0102-y
16. Oommen B, Ma D (1988) Deterministic learning automata solutions to the equipartitioning problem. IEEE Trans Comput 37(1):2–13
17. Oommen BJ, de St Croix EV (1996) Graph partitioning using learning automata. IEEE Trans Comput 45(2):195–208
18. Oommen BJ, Fothergill C (1993) Fast learning automaton-based image examination and retrieval. Comput J 36(6):542–553
19. Oommen BJ, Ma DCY (1988) Deterministic learning automata solutions to the equipartitioning problem. IEEE Trans Comput 37(1):2–13
20. Oommen BJ, Ma DCY (1992) Stochastic automata solutions to the object partitioning problem. Comput J 34(6):A105–A120
21. Poznyak AS, Najim K (1997) Learning automata and stochastic optimization. Springer, Berlin
22. Sen S, Sajja N (2002) Robustness of reputation-based trust: boolean case. In: Proceedings of the first international joint conference on autonomous agents and multiagent systems: part 1, Bologna, Italy. ACM, New York, pp 288–293
23. Shapiro C (1982) Consumer information, product quality, and seller reputation. Bell J Econ 13(1):20–35
24. Thathachar MAL, Sastry PS (2002) Varieties of learning automata: an overview. IEEE Trans Syst Man Cybern, Part B, Cybern 32(6):711–722
25. Thathachar MAL, Sastry PS (2003) Networks of learning automata: techniques for online stochastic optimization. Kluwer Academic, Boston
26. Tsetlin ML (1973) Automaton theory and the modeling of biological systems. Academic Press, New York
27. Whitby A, Jøsang A, Indulska J (2005) Filtering out unfair ratings in bayesian reputation systems. J Manag Res 4(2):48–64
28. Yu B, Singh MP (2003) Detecting deception in reputation management. In: Proceedings of the second international joint conference on autonomous agents and multiagent systems, Melbourne, Australia. ACM, New York, pp 73–80
29. Yuan W, Guan D, Lee Y-K, Lee S (2010) The small-world trust network. Appl Intell 1–12. doi:10.1007/s10489-010-0230-7
30. Zacharia G, Maes P (2000) Trust management through reputation mechanisms. Appl Artif Intell 14(9):881–907

Appendix B

Paper II

Title:	Learning Automaton Based On-line Discovery and Tracking of Spatio-Temporal Event Patterns
Authors:	**Anis Yazidi**[1], Ole-Christoffer Granmo[2] and B. John Oommen[1,2]
Affiliations:	[1]University of Agder, Faculty of Engineering and Science, P. O. Box 509, NO-4898 Grimstad, Norway [2]School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6
Journal:	*To appear in IEEE Transactions on Systems, Man and Cybernetics. (Accepted January 10, 2012).*

Learning Automaton Based On-line Discovery and Tracking of Spatio-Temporal Event Patterns *

Anis Yazidi[†], Ole-Christoffer Granmo[‡], B. John Oommen[§]

Abstract

Discovering and tracking of spatio-temporal patterns in noisy sequences of events is a difficult task that has become increasingly pertinent due to recent advances in ubiquitous computing, such as community-based social networking applications. The core activities for applications of this class include the sharing and notification of events, and the importance and usefulness of these functionalities increases as event-sharing expands into larger areas of one's life. Ironically, instead of being helpful, an excessive number of event notifications can quickly render the functionality of event-sharing to be obtrusive. Indeed, any notification of events that provides redundant information to the application/user can be seen to be an unnecessary distraction. In this paper, we introduce a new scheme for discovering and tracking noisy spatio-temporal event patterns, with the purpose of suppressing reoccurring patterns, while discerning novel events. Our scheme is based on maintaining a collection of hypotheses, each one conjecturing a specific spatio-temporal event pattern. A dedicated Learning Automaton (LA) – the *Spatio-Temporal Pattern LA* (STPLA) – is associated with each hypothesis. By processing events as they unfold, we attempt to infer the correctness of each hypothesis through a real-time guided random walk. Consequently, the scheme we present is computationally efficient, with a minimal memory footprint. Furthermore, it is ergodic, allowing adaptation. Empirical results involving extensive simulations demonstrate the STPLA's superior convergence and adaptation speed, as well as an ability to operate successfully with noise, including both the erroneous inclusion and omission of events. An empirical comparison study was performed and confirms the superiority of our scheme compared to a similar state of art approach [28]. In particular, the robustness of the

*This work was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. A preliminary version of some of the results of this paper was presented at PRICAI-2010, the 2010 Pacific Rim International Conference on Artificial Intelligence, Deagu, Korea, September 2010 [34]. The list of authors in [34] is different from the list of authors of this paper, because the other authors of [34] did not contribute after the conference paper was published, especially in the rigorous theoretical analysis found here.

[†]This author can be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: anis.yazidi@uia.no.

[‡]This author can also be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: ole.granmo@uia.no.

[§]*Chancellor's Professor* ; *Fellow: IEEE* and *Fellow: IAPR*. This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. E-mail address: oommen@scs.carleton.ca.

STPLA to inclusion as well as to omission noise constitutes a unique property compared to other related approaches. Additionally, the results included, which involve a so-called "*Presence Sharing*" application, are both promising and in our opinion, impressive. It is thus our opinion that the proposed STPLA scheme is, in general, ideal for improving the usefulness of event notification and sharing systems, since it is capable of significantly, robustly and adaptively suppressing redundant information.

Keywords: *Learning Automata, Spatio-Temporal Pattern Recognition*

1 Introduction

Presence Sharing is a ubiquitous service in which distributed mobile devices periodically broadcast their identity via short-range wireless technology such as BlueTooth or WiFi [8]. The whole problem of *Presence Sharing* is intricately bound to the issue of the recording and processing of "events" involving the entities included within the social network. Applications that utilize *Presence Sharing* have been used in social contexts to maintain an "in touch" feeling strengthening social relations [9], as well as in work environments to enhance collaboration between colleagues [11].

Typically, "events" occurring in the real world can be characterized as being in one of two classes, i.e., "Stochastically Episodic" (SE) and "Stochastically Non-Episodic" (SNE). This is a distinction that is especially pertinent in simulation, where it is customary for one to model the behaviour of accidents, telephone calls, network failures etc. using their respective probability distributions, even though they follow no known pattern. Indeed, events of these families happen all the time, and so can be termed as being "stochastically non-episodic". As opposed to this, there is a whole class of events that can stochastically occur in a non-anticipated manner. These so-called "stochastically episodic" events include earthquakes, nuclear explosions etc. The difficulty with modelling SE events is that most of the observations appear as noise. However, when the SE event does occur, its magnitude and features far overshadow the background, as one observes after a seismic event. The modelling and simulation of such SE events in the presence of a constant stream of SNE events is a relatively new field [5, 6], where the authors model the SE and SNE events *simultaneously* in such a way that the effect of an SE event is perceived through the "lens" of the underlying background of SNE events.

Since events are almost omnipresent, one has to consider the observation due to Garlan *et al.* [10], who state that the most precious resource in a computer system is no longer its processor, memory, disk, or network, but rather human attention. Thus, our aim in this paper is to address a fundamental challenge concerning the above class of applications: *How can one harvest the benefit of event-sharing without distracting the application user with redundant notifications*? The solution we propose is to try to discern the nature of the events encountered[1]. Of course, the events may not

[1]To exemplify the usefulness of such a strategy, consider the nuisance caused by being notified every time one

be drastically SE or SNE, as in the case of earthquakes or nuclear explosions. However, if we can discern that an event is repeating (even though this repetition is non-periodic), it is still of a SNE nature which must be given less weight, while non-repeating events (which are in one sense, SE) must be assigned a greater weight. Thus, the question we resolve involves demonstrating how we can enhance the *Presence Sharing* experience by weighting the SE and SNE events appropriately.

Apart from the "*Presence Sharing*" application example, we elucidate the relevance of our approach for tracking noisy spatio-temporal event patterns, which we shall explain in great detail in the rest of the paper, by providing two specific examples of applications one form the field of Ambient Assisted Living and the other from the Internet of Things.[2] In the area of Ambient Assisted Living [30], alerts could be trigged if an elderly inhabitant does not visit the kitchen according to normal habits. The alerts are suppressed if, in turn, the person returns to the normal habits of visiting the kitchen. Such automatic alerts could serve to monitor the habits of the elderly inhabitants in order to assist them in case of need. The Internet of Things [4] envisions an environment saturated with the presence of an abundant number of wireless sensors. In the later field, a possible application of our scheme involves a sensor that is expected to regularly submit reports to a data collection center according to a predefined periodicity. However, some sporadic submission failures might occur, stemming from link failure, collision, congestion, low sensor energy, sensor failure etc. Consequently, the reports might not be regularly received by the data collection center according to the expected periodicity. The sensor is considered "well-functioning" when the reports are received according to a *noisy* periodic pattern. In the other hand, the sensor is deemed to be "mal-functioning" is the reports significantly drift from the regular pattern. In the later case, alerts could be issued in order to allow some counter measures to be taken. In addition to the aforementioned applications, we suggest that our approach has benefits in the area of video-surveillance [2] where detecting spatio-temporal activities constitutes a major research topic.

1.1 Related Work

A number of earlier studies have investigated various techniques for discovering the periodicity of time patterns, such as the episode[3] discovery algorithm found in [35]. However, episode discovery, and other related approaches, suffer from the limitation that they assume unperturbed patterns that exhibit an exact periodicity. Unfortunately, the real-life unfolding of events is typically noise ridden. On the one hand, regular events may get cancelled, introducing what we define as *omission*

meets a colleague at work, which is a repeating pattern, or a SNE event. In contrast, it would be far more useful to be promptly notified whenever the same colleague unexpectedly appears in your vicinity after a travel abroad. This would be non-repeating pattern, or an SE event.

[2] We are extremely grateful for the anonymous referee who pointed out these two applications of our approach both in the area of Ambient Assisted Living and in Internet of Things.

[3] The expression "episode" used in this setting must not be confused with the class of SE and SNE events described in the earlier paragraph.

noise, and on the other, events may arise spontaneously and unexpectedly, without being part of a periodic pattern, introducing *inclusion noise*.

A pioneering work which was reported in [16], introduced the concept of *off-line* mining of partially periodic events. Partially periodic events are characterized by irregular periodicity that is disrupted by noise. Despite that finding partially periodic events is common in many real life applications, few are the research studies that have been reported in this direction. In [16], the authors introduced a chi-squared test for discovering partially periodic patterns. A particularly interesting study on mining partially periodic patterns has been recently reported in [28]. In [28, 27], the frequent and periodic activity miner (FPAM) algorithm was introduced for finding repetitive patterns in resident's activities in smart-environment applications. FPAM is originally an *off-line* algorithm. However, in order to be able to detect changes in the patterns of the resident activities, FPAM was applied at regular mining sessions schedule [28, 27]. A main shortcoming of the later approaches [16, 28, 27] is that they operate in an *off-line* fashion. This is contrast to our application where deciding whether to suppress event notifications must often be done instantaneously, in an *on-line* manner, as the events are unfolding. Indeed, we argue that any realistic scheme should discover and adapt to patterns as they appear and evolve on-line, without relying on extensive off-line data mining. To the best of our knowledge, we report the first *on-line* approach for mining partially periodic events.

1.2 Paper Contribution

Our paper presents a set of novel contributions that we summarize in the following:

- To the best of our knowledge, we present the first reported *on-line* approach for discovering and tracking of spatio-temporal patterns in noisy sequences of events.

- Extensive simulations results confirm that our scheme outperforms a similar recent state of the art scheme, namely FPAM [28]. The robustness of the Spatio-Temporal Pattern LA (STPLA) to inclusion and as well as omission noise constitutes a unique property compared to FPAM. In this perspective, FPAM is shown to be fragile to variations in the omission noise.

- Simulations results show that our scheme possesses an excellent ability to cope with non-stationary environments. Interestingly, we found that using a Balanced memory STPLA with memory depth as small as 5 yields quite good results in *almost every* environment settings! Therefore, finding the proper internal configuration of the STPLA does not constitute a critical issue to ensure adaptivity in dynamic environment. This is in contrast to "modified" FPAM presented in section 4 where the performance is dependent on a judicious choice of the sliding window size.

- We present the first reported approach to suppressing redundant notifications in pervasive environments application. We believe that our work paves the way towards more research interest in devising unobtrusive application by making use of the potential offered by machine learning techniques. We believe that such unobtrusive applications [10] will constitute an emerging trend in future mobile applications. The usefulness and feasibility of our application was demonstrated through a working prototype.
- Since our solution is based on LA, it is both computationally simple and memory efficient.

1.3 Paper Organization

The paper is organized as follows. In Section 2, we present our overall approach to on-line discovery and tracking of spatio-temporal event patterns, in which the so-called Learning Automata (LA) plays a crucial role. The scheme is designed to deal with noisy spatio-temporal event patterns, when event patterns are evolving with time. Section 3 contains the theoretical analysis of our scheme as well as some fascinating limiting properties. We continue in Section 4 by evaluating our scheme using an extensive range of static and dynamic *noisy* event patterns. The experiments demonstrate the scheme's superior convergence and adaptation speed, as well as an excellent ability to operate successfully with noise, including both erroneous inclusion and omission of events. In addition, we report a detailed comparison of our STPLA with FPAM.

In order to highlight the applicability of our scheme, we present a "*Presence Sharing*" application prototype in Section 5 where we also summarize some initial user experiences. Finally, Section 6, concludes the paper and also provide pointers for further work.

2 On-line Discovery and Tracking of Spatio-Temporal Event Patterns

The method which we propose is based on the theory of LA. For an extensive overview of this theory, we refer the reader to an excellent book by Narendra and Thathachar [21], as well as to comprehensive list of other related books treating in details the theory of LA [13, 20, 26, 31].

In all brevity, we state that our scheme is based on maintaining a collection of hypotheses, each one conjecturing a specific spatio-temporal event pattern. A dedicated LA, which we coin the *Spatio-Temporal Pattern LA* (STPLA), is associated with each hypothesis. The STPLA decides whether its corresponding hypothesis is true by observing events as they unfold, processing evidence for and/or against the correctness of the hypothesis. To explain this, we first address hypothesis management, and then proceed with the details of the STPLA.

2.1 Hypothesis Management

The premise of our discussions is the following: In order to reduce distraction, events should only be signalled when they are SE. This means that they cannot be anticipated, obey no known stochastic distribution, and possess an element of "surprise", i.e., they can not be easily predicted by the recipient[4]. An event can either be sporadic, arising spontaneously, or it can be part of a spatio-temporal pattern, making it occur regularly. In either case, if it cannot be explained by any of the spatio-temporal patterns that are known by the recipient, the recipient should be notified. However, when the event constitutes a part of an ongoing spatio-temporal pattern, it is really non-episodic (or SNE) in nature. We require that this phenomenon be discovered as soon as possible, so that the events generated from this pattern can be suppressed before the pattern loses its novelty to the recipient.

In our proposed scheme, when an event is observed, all potentially interesting patterns that *could* have produced the event are identified. We refer to these potential patterns as *hypotheses*. The reader will thus observe that our approach is based on the concept of predefined pattern structures, as advocated in [14], rather than trying to look for patterns with unknown structure. Thus, in this spirit, we consider a discrete world of m spatial location primitives $L = \{l_1, l_2, \dots, l_m\}$ and of n discrete time primitives $T = \{t_1, t_2, \dots, t_n\}$ of appropriate granularity. By way of example, the location primitives could be "University", "Office", or "Gym Club", while the time primitives could be "Mondays", "Tuesdays", "Weekends", and "Daily". The location and time primitives are combined from their cross-product spaces to produce spatio-temporal patterns.

Thus, the resulting spatio-temporal pattern space would be an exhaustive enumeration of 4×3 (12) relevant combinations namely, {"Mondays at University", "Mondays at Office", "Mondays at Gym Club", "Tuesdays at University", "Tuesdays at Office", "Tuesdays at Gym Club", "Weekends at University", "Weekends at Office", "Weekends at Gym Club", "Daily at University", "Daily at Office", "Daily at Gym Club"}.

Each spatio-temporal pattern of the latter form is seen as a hypothesis, conjecturing that the respective pattern specifies an ongoing stream of events. In the following, we assume that there are r such hypotheses, represented as a set $H = \{h_1, h_2, \dots, h_r\}$. Observe that although the cardinality of this set might get large, the computational efficiency and small memory footprint of our LA (as seen presently), effectively handles the size of the state space. Obviously, the maximum value of r is $m \times n$. Note too that the novelty of this present work is not the above indicated structuring of the spatio-temporal pattern space, which is a well-known approach used in typical calendar systems. Rather, it is the learning scheme we propose[5] for determining whether a given spatio-temporal event

[4]Events should, of course, also match the interest profile of the recipient. We will, in this paper, assume that all events are of interest, as long as they are novel. On-line adaptive learning of interest profiles will be addressed in another forthcoming paper.

[5]Using the techniques presented in [5, 6], we are currently investigating how one-class classifiers can be used to

pattern can be found in a stream of events, in an on-line manner, and under noisy conditions.

2.2 Learning Automaton Based On-line Discovery and Tracking of Spatio-Temporal Event Patterns

We base our work on the principles of LA [21]. LA have been used in systems that have incomplete knowledge about the Environment in which they operate [1, 13, 20, 21, 26, 31]. The learning mechanism attempts to learn from a *stochastic Teacher* which models the Environment. In his pioneering work, Tsetlin [32] attempted to use LA to model biological learning.

In the first LA designs, the pioneering ones are those that belong to the fixed-structure stochastic automata (FSSA) families. In FSSA, the transition and the output functions were time invariant, and for this reason these LA were considered Fixed Structure Stochastic Automata (FSSA). Tsetlin, Krylov, and Krinsky [32] presented notable examples of this type of automata. The solution we present here, essentially falls within this family.

With respect to applications, the entire field of LA and stochastic learning has had a myriad of applications including pattern recognition, statistical decision making, parameter optimization, distributed scheduling, training hidden Markov models, neural network adaptation, graph partitioning, string taxonomy, network call admission control, quality of service routing, network congestion control [21, 1, 29, 12, 17, 23, 18, 22, 24, 25, 3, 33, 19].

Generally stated, an LA chooses a sequence of actions offered to it by a random *environment*. The environment can be seen as a generic *unknown* medium that responds to each action with some sort of reward or penalty, usually *stochastically*. Based on the responses from the environment, the aim of the LA is to find the action that minimizes the expected number of penalties received.

Designing an LA strategy for solving a given problem involves modeling the actions, simulating the transforming functions, and representing the system's output to have reward or penalty responses. This is where the creativity of the researcher becomes apparent! In this regard, before we proceed with describing the STPLA itself, it is necessary for us to first define the environment that we are dealing with.

2.2.1 Spatio-Temporal Pattern Environment:

The purpose of the Spatio-Temporal Pattern Environment is to provide feedback to the individual STPLA about the validity of their respective hypotheses. In all brevity, at each time instant matching the time primitive t_i, if an event takes place and an STPLA conjectures the presence of that event at location l_j according to the corresponding periodicity, it informs the environment about this successful prediction. The environment responds in turn with a Reward that "enforces"

learn the most appropriate hypothesis. This would assume that the patterns which can be anticipated constitute the SNE events, and the set of SE events, which cannot be anticipated, constitutes the one-class to be recognized.

the STPLA hypothesis conjecturing that the event is a part of the spatio-temporal pattern attached to location l_j. Conversely, if the STPLA expects the presence of the event at the same location as a part of a spatio-temporal pattern while in reality the event does not take place, this non-occurrence of the event is submitted to the environment. The environment interprets the latter incorrect prediction as a Penalty and therefore responds with a *Penalty* instead to the automaton. In other words, the STPLA is penalized if an event is predicted, but does not actually take place. From a high level perspective, a success "enforces" the correctness of the hypothesis maintained by the STPLA while a penalty "weakens" it.

An example of the latter reward policy is illustrated in Fig. 1.

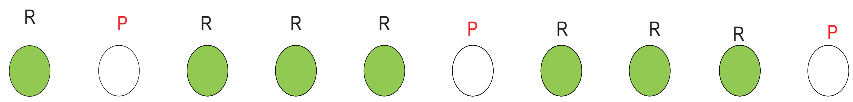


Figure 1: The feedback sequence (R-Reward, P-Penalty) for a daily event hypothesis, where the green circles correspond to the days where the meetings take place while the white circles correspond to the days where the meetings are cancelled

Fig.1 illustrates general reinforcement scheme for events generated from a daily meeting. Please note that the green circles refer to the days where the meetings actually take place while the white circles represent the days where there are no meetings. The same Figure depict the history of the meetings occurrences during a period of 10 days, where the meetings occurred daily except for the 2^{nd}, the 6^{th} and the 10^{th} days, depicted by white circles.

The STPLA that hypothesizes a daily meeting will be rewarded each time a meeting takes place (green circle) because of its ability to correspondingly predict the daily event. An important challenge that we address in this paper, however, is how to deal with spatio-temporal event patterns that are affected by noise. In the figure, for example, some of the daily meetings may be cancelled (depicted by white circles) due to external conditions, such as when the participants are unavailable. Thus, when meetings are cancelled, the STPLA maintaining the daily meeting hypothesis will get penalized because of its incorrect prediction. In a similar vein, so-called "straggler" events, not being part of any periodic pattern, can also occur in a sporadic and spontaneous manner.

From the above example it can be seen that we face two kinds of noise:

Omission Error: This is an error which occurs when an event that forms a part of a periodic spatio-temporal pattern is randomly left out. In other words, the event was supposed to have taken place according to the pattern, but did not. Notice the SE nature of this event – it is not something that could have been anticipated.

Inclusion Error: This is an error which occurs when an event that occurs is not part of a periodic (anticipated) pattern, but rather arises sporadically and spontaneously. Again, one must observe the SE nature of this event.

By way of example, Alice may cancel a regular meeting with Bob due to ill health. However, Alice may still meet Bob sometime outside of the regular meeting schedule – purely by chance (e.g., an accidental meeting in the canteen). In this manner, we can appropriately model both these kinds of noise.

In this paper, we have assumed that omission and inclusion errors are uncorrelated.[6]

It should be mentioned that in the theory of LA, the LA is only allowed to observe the responses of the Environment, and not the underlying stochastic model that generates the responses of the Environment. In other words, one works with the Environment being treated as the "Black Box model". The aim of the solution we have designed is not to detect the noise and an of itself, but rather to design a LA that is resilient to the noise that might rather mislead the system in achieving task of discovering the Spatio-Temporal patterns.

2.2.2 The Spatio-Temporal Pattern Learning Automaton (STPLA):

We now introduce the STPLA that we have designed to discover and track spatio-temporal patterns. In brief, the task of an STPLA is to decide whether a specific spatio-temporal pattern hypothesis is true. By observing events as they unfold, the correctness of an hypothesis is decided.

The STPLA can be designed to model arbitrarily general SE and SNE events. But due to space limitations, in this paper, we confine our design and implementation details to events which can be characterized *deterministically*.

The STPLA is inspired by the family of fixed structured LA [21]. Accordingly, a STPLA can be defined in terms of a quintuple [21]:

$$\{\underline{\Phi}, \underline{\alpha}, \underline{\beta}, \mathcal{F}(\cdot,\cdot), \mathcal{G}(\cdot,\cdot)\}.$$

Here, $\underline{\Phi} = \{\phi_1, \phi_2, \ldots, \phi_s\}$ is the set of internal automaton states. $\underline{\alpha} = \{\alpha_1,\ \alpha_2, \ldots,\ \alpha_r\}$ is the set of automaton actions. Further, $\underline{\beta} = \{\beta_1, \beta_2, \ldots, \beta_m\}$ is the set of inputs that can be given to the automaton. An output function $\alpha_t = \mathcal{G}[\phi_t]$ determines the action performed (or chosen) by the automaton given the current automaton state. Finally, a transition function $\phi_{t+1} = \mathcal{F}[\phi_t, \beta_t]$ determines the new state of the automaton from: (1) The current state of the automaton and (2) The response of the environment.

Based on the above generic framework, the crucial issue is to design automata that can learn the optimal action when interacting with the environment. Several designs have been proposed in

[6] We are thankful to an anonymous referee for pointing out an interesting research direction that would emerge if the omission and inclusion errors were correlated. In fact, the referee suggested studying the effect of the degree correlations between the omission and inclusion errors on the efficiency of the STPLA. We strongly believe that it is a promising research direction.

the literature, and the reader is referred to [21] for an extensive treatment. In this paper, since we target the learning of spatio-temporal patterns, our goal is to design an LA that is able to discover and track such patterns over time. Briefly stated, we construct an automaton with

- States: $\underline{\Phi} = \{1, 2, \ldots, N_1, N_1 + 1, \ldots, N_1 + N_2 + 1\}$.
- Actions: $\underline{\alpha} = \{Notify, Suppress\}$.
- Inputs: $\underline{\beta} = \{Reward, Penalty\}$.

Fig. 2 specifies the state space of STPLA as well as the $\mathcal{G}$ and $\mathcal{F}$ matrices. The $\mathcal{G}$ matrix can be

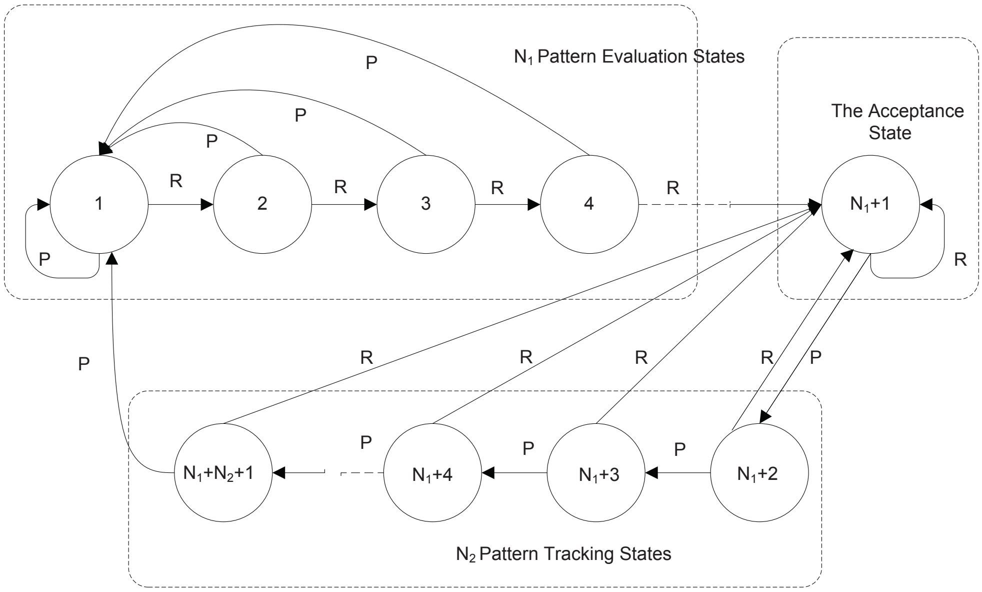


Figure 2: The state transition map and the output function of a STPLA

summarized as follows. If the automaton state lies in the set $\{1, \ldots, N_1\}$, which we refer to as the *Pattern Evaluation States*, then the LA will choose the action "Notify". If, on the other hand, the state is either $N_1 + 1$ or one of the states in the set $\{N_1 + 2, \ldots, N_1 + N_2 + 1\}$, it will choose the action "Suppress". We refer to the state $N_1 + 1$ as the *Pattern Acceptance State*, and the states $\{N_1 + 2, \ldots, N_1 + N_2 + 1\}$ as the *Pattern Tracking States* for reasons explained presently. Note that since we initially do not know whether a pattern is present, we set the initial state of our automaton to 1.

Formally, G is defined as follows:

$$G(\phi_i) = \begin{cases} Notify, & \text{if } \phi_i \in \{1, 2, \ldots, N_1\} \\ Suppress, & \text{if } \phi_i \in \{N_1 + 1, N_1 + 2, \ldots, N_1 + N_2 + 1\} \end{cases}$$

The state transition matrix $\mathcal{F}$ determines how the learning proceeds. We follow the LA nomenclature used in [21] and denote $\mathcal{F}(0)$ the transition matrix in case of Reward and $\mathcal{F}(1)$ the transition matrix in case Penalty.[7]

The matrix $\mathcal{F}(0)$ is defined as:

$$\mathcal{F}(0) = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \end{pmatrix}$$

The matrix $\mathcal{F}(1)$ is given by:

$$\mathcal{F}(1) = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

In brief, the learning is divided into three parts:

Pattern Evaluation: In the Pattern Evaluation part, the goal of the LA is to discover the presence of the spatio-temporal event pattern associated with the maintained hypothesis, without being distracted by omission and inclusion errors. In this phase, the state transitions illustrated in the figure are such that any deviance from the hypothesized pattern, modelled as a Penalty (P), causes a jump back to state 1. Conversely, only a systematic presence of the pattern hypothesized, modelled as a pure sequence of Rewards (R), will allow the LA to pass into the Pattern Acceptance part.

Pattern Acceptance: In the Pattern Acceptance part, consisting of state N_1+1, the hypothesized pattern has been confirmed with high probability.

Pattern Tracking: In the Pattern Tracking part, consisting of states $\{N_1+2, \dots, N_1+N_2+1\}$, the goal is to detect when the discovered pattern disappears, without getting distracted by

[7] In LA theory, "0" usually refers to Reward while "1" is used for Penalty.

omission errors. Thus, this part is the "opposite" of the Pattern Evaluation part in the sense that a pure sequence of Penalties is required to "throw" the LA back into the Patter Evaluation part again, while a single Reward reconfirms the pattern, returning the LA to the Pattern Acceptance part of the state space.

In other words, the automaton attempts to incorporate past deterministic responses when deciding on a sequence of actions.

In classical LA applications (such as in solving bandit-like problems), the response of the environment depends on the action currently chosen by the LA. However, in the particular application described in this paper, this is not the case. Rather, in fact, the chosen action by the LA (to either *Suppress* or *Notify* the user) does not have any effect on the next response of the Environment (which is, in turn, modeled as an occurrence or not of a predicted event). In other words, we can argue that our LA attempts to perform online mining of a stream of events, where the occurrence of the events within the stream is *totally* independent of the action chosen by the LA.

We define the "*Ensemble*" characteristic of a set of STPLA as follows: *An event is only signalled to the recipient when all of the STPLA that maintain hypotheses that are consistent with the event, collectively find themselves in the Pattern Evaluation part of the state space.* As soon as one of the STPLA can deterministically[8] explain an event as being part of the corresponding hypothesized spatio-temporal event pattern, that particular event will be suppressed and no notification will be issued to the recipient.

3 Theoretical Results

To render the problem tangible, we suppose that the Rewards are stochastic. Let p denote the reward probability and $q = 1 - p$ denote the Penalty probability.

Given a hypothesized pattern, p and q have different interpretations according to the following two cases:

- If the events follow a regular pattern than q could be assimilated to omission noise.
- If the events are hazardous and do not follow a regular periodicity, then p could be assimilated to inclusion noise. In this case, p is a noise that might mislead the learning process and result in false positives.

Theorem 1. *Let P_1 the probability to be in the Pattern Evaluation states, meaning the pattern is not learned yet. P_1 is given by the following expressions:*

$$P_1 = \frac{(1 - p^{N_1})q^{N_2}}{(1 - p^{N_1})q^{N_2} + p^{N_1 - 1}(1 - q^{N_2 + 1})} \tag{1}$$

[8] The system can easily be generalized for SE and SNE events by rendering the transitions stochastic.

Let P_2 the probability to overcome the Pattern Evaluation states, meaning the pattern is already learned. P_2 can be expressed by:

$$P_2 = \frac{(1-q^{N_2+1})p^{N_1-1}}{(1-p^{N_1})q^{N_2} + p^{N_1-1}(1-q^{N_2+1})} \tag{2}$$

Proof: To prove these results, we shall analyze the properties of the underlying Markov Chain that describes the behavior of the walker. By investigating the various transition considerations, we see that matrix of transition probabilities, M, is given by:

$$M = \begin{pmatrix} q & p & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ q & 0 & p & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q & 0 & \dots & 0 & p & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & p & q & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & p & 0 & q & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & p & 0 & \dots & 0 & q \\ q & 0 & \dots & 0 & p & 0 & 0 & \dots & 0 \end{pmatrix}$$

The reader should observe the transitions into the non-adjacent states, i.e., those which represent the jumps.

We shall now compute π_i the stationary (or equilibrium) probability of the chain being in state i. Clearly M represents a single closed communicating class whose periodicity is unity. The chain is thus ergodic, and the limiting probability vector is given by the eigenvector of M^T corresponding to the eigenvalue unity. The vector of steady state (equilibrium) probabilities $\Pi = [\pi_1, \dots, \pi_{N_1+N_2+1}]^T$ can be thus computed by solving $M^T\Pi = \Pi$.

Consider first the stationary probability of being in state 1, π_1. By expanding the first row we see that this is expressed by the following equation:

$$\begin{aligned} \pi_1 &= q\pi_1 + q\pi_2 + \dots + q\pi_{N_1} + q\pi_{N_1+N_2+1} \\ &= q\sum_{k=1}^{N_1} \pi_k + q\pi_{N_1+N_2+1}. \end{aligned} \tag{3}$$

For $2 \leq k \leq N_1$, the stationary probability π_k is given by a straightforward first-order difference equation, Eq. (4):

$$\pi_k = p\pi_{k-1}. \tag{4}$$

By applying recurrence, the Eq. (4) can be rewritten as:

$$\pi_k = p^{k-1}\pi_1. \tag{5}$$

By expanding the $(N+1)^{st}$ row, we can see that the probability of being in the "acceptance" state π_{N_1+1} is given by Eq. (6):

$$\pi_{N_1+1} = p\pi_{N_1} + p\pi_{N_1+1} + p\sum_{k=1}^{N_2}\pi_{N_1+1+k}. \tag{6}$$

Again, for $N_1 + 1 \leq k \leq N_1 + N_2 + 1$, the steady probabilities are given by:

$$\pi_k = q\pi_{k-1}. \tag{7}$$

By applying recurrence, Eq. (7) can be written for $1 \leq k \leq N_2 + 1$ as:

$$\pi_{N_1+1+k} = q^k\pi_{N_1+1}. \tag{8}$$

P_1 is given by Eq. (9):

$$\begin{aligned} P_1 &= \sum_{k=1}^{N_1}\pi_k \\ &= \sum_{k=1}^{N_1}p^{k-1}\pi_1 \\ &= \frac{1-p^{N_1}}{1-p}\pi_1. \end{aligned} \tag{9}$$

Similarly, P_2 can be expressed by:

$$\begin{aligned} P_2 &= \sum_{k=N_1+1}^{N_1+N_2+1}\pi_k \\ &= \sum_{k=0}^{N_2} q^k\pi_{N_1+1} \\ &= \frac{1-q^{N_2+1}}{p}\pi_{N_1+1}. \end{aligned} \tag{10}$$

Using Eq. (9) and replacing it in Eq. (3) we obtain:

$$\pi_1 = q\sum_{k=1}^{N_1}\pi_k + q^{N_2+1}\pi_{N_1+1}. \tag{11}$$

Therefore, we obtain:

$$\pi_1 = (1 - p^{N_1})\pi_1 + q^{N_2+1}\pi_{N_1+1}. \tag{12}$$

From the above, we can deduce the equation that relates π_1 and π_{N_1+1}:

$$\pi_{N_1+1} = \frac{p^{N_1}}{q^{N_2+1}}\pi_1. \tag{13}$$

Using the values of P_1 and P_2 and the fact that $P_1 + P_2 = 1$, and after carrying out some simple algebraic manipulations we obtain:

$$\pi_1 = \frac{q^{N_2+1}}{(1 - p^{N_1})q^{N_2} + p^{N_1-1}(1 - q^{N_2+1})}, \tag{14}$$

and

$$\pi_{N_1+1} = \frac{p^{N_1}}{(1 - p^{N_1})q^{N_2} + p^{N_1-1}(1 - q^{N_2+1})}, \tag{15}$$

whence:

$$P_1 = \frac{(1 - p^{N_1})q^{N_2}}{(1 - p^{N_1})q^{N_2} + p^{N_1-1}(1 - q^{N_2+1})}, \tag{16}$$

and

$$P_2 = \frac{(1 - q^{N_2+1})p^{N_1-1}}{(1 - p^{N_1})q^{N_2} + p^{N_1-1}(1 - q^{N_2+1})}, \tag{17}$$

which concludes the proof. □

3.1 Balanced Memory STPLA

With regards to the commonly used terminology in the field of FSSA, N_1 corresponds to the memory depth of the "Notify" action, while N_2+1 corresponds to the memory depth of the "Suppress" action.

In this section, we consider the particular case where the memory depth of both actions: "Notify" and "Suppress" are equals, i.e $N_1 = N_2 + 1 = N$. The later case defines what we call a Balanced Memory STPLA. The idea behind such definition lies the following explanation: in the absence of a-priori information about the existence or absence of an underlying pattern, employing a Balanced Memory STPLA seems a reasonable choice since the equal memory depth of the actions $\{Notify, Suppress\}$ does not favor any of them. In this section, we give two theorems relative to the convergence of the Balanced memory STPLA. In the rest of the paper, we make an abuse notation and call N the memory depth of the Balanced memory STPLA, where $N = N_1 = N_2 + 1$.

Theorem 2. *For a Balanced Memory STPLA, if $p > 0.5$ then the notification probability P_1 approaches 0 as the memory depth N tends to infinity. Formally,* $\lim_{N\to+\infty} P_1 = 0$

Proof:

Consider the quotient $\frac{P_1}{P_2}$. To prove this result, we first compute its limit as N tends to infinity for $p > 0.5$.

$$\frac{P_1}{P_2} = \frac{(1-p^N)q^{N-1}}{(1-q^N)p^{N-1}}. \tag{18}$$

Since $p > 0.5$, we have the condition that $q/p < 1$. Therefore $\lim_{N\to\infty}(q/p)^{N-1} = 0$. On the other hand, $\lim_{N\to\infty} \frac{1-p^N}{1-q^N} = 1$.

Therefore $\lim_{N\to\infty} \frac{P_1}{P_2} = 0$. Thus, we conclude that $\lim_{N\to\infty} P_1 = 0$, and the result is proved. $\square$

The analogous result for the case when $p < 0.5$ follows.

Theorem 3. *For a Balanced Memory STPLA, if $p < 0.5$ then the notification probability P_1 approaches 1 as the memory depth N tends to infinity. Formally,* $\lim_{N\to+\infty} P_1 = 1$

Proof: The proof is similar to the proof of Theorem 2, except that we rather consider the quotient $\frac{P_2}{P_1}$.

We remark that $p/q < 1$ for $p < 0.5$, and thus, $\lim_{N\to\infty}(p/q)^{N-1} = 0$. Moreover, we see that $\lim_{N\to\infty} \frac{1-q^N}{1-p^N} = 1$. Therefore, $\lim_{N\to\infty} \frac{P_2}{P_1} = 0$, and consequently $\lim_{N\to\infty} P_2 = 0$. Hence the result! $\square$

4 Experiments

In order to evaluate our scheme, we have applied it to both an event simulation system as well as to a real world prototype. This section reports the results obtained using the simulation, while the next section covers the prototype.

Since one of our main aims is handling noisy patterns, we intend to impose "stress" onto our scheme by using a wide range (percentage or degrees) of omission and inclusion errors. We will use q to denote the probability of event omission, while p denotes the probability of event inclusion. We also investigate how the number of states N_1 and N_2 affect the LA's speed and the accuracy.

As a performance criterion, we have chosen the probability of issuing a notification (alert) when an event takes place. We refer to this probability as P_1. Intriguingly, when a spatio-temporal pattern produces events, P_1 should be minimized, while when events are novel, P_1 should be maximized. We will presently see that our scheme achieves both. For instance, consider an event that occurs daily, with the possibility, however, that events may get cancelled (causing omission errors). In that case, our scheme should quickly stop alerting the user about these events. In contrast, when novel sporadic events occur, even on a daily basis, our scheme should rather always produce alerts, so that the user is notified about these novel events. Thus, by monitoring our scheme in terms of the index P_1 using various scenarios, we can capture its overall performance. The STPLA has been extensively studied for different settings. In the interest of brevity, we here report a selection of the most pertinent experiments. In addition, we report the results of comparison of STPLA with

FPMA. The result of the comparison is conclusive and demonstrate the superiority of our scheme compared to FPMA.

4.1 Performance After Convergence

Table 1 summarizes the performance after convergence, with a wide range of event inclusion probabilities, p, event omission probabilities, q, *Pattern Evaluation States*, N_1, and *Pattern Tracking States*, N_2. The resulting performance is then reported in terms of P_1, with P_1 being estimated by averaging over $1,000$ experiments, each consisting of $100,000$ iterations.

In the case of daily patterns, we have varied the omission error probabilities from $q = 0.05$ to $q = 0.2$, thus covering a spectrum of small to high degrees of omission noise. Intuitively, as we increase q, more omission noise is introduced and consequently it becomes increasingly difficult for the STPLA to discern the presence of an underlying spatio-temporal pattern.

In the case when no patterns are present, we have allowed random encounters to appear with probabilities from $p = 0.05$ to $p = 0.2$. Similarly, as we increase p, more inclusion noise is introduced and consequently the STPLA becomes more prone to false positives by wrongly signalling the presence of a spatio-temporal pattern while in reality, no patterns are present.

For the memory settings, we used values of N_1 and N_2 ranging from 1 to 5. We modified the internal memory of the STPLA in order to produce different types of "bias":

- The first three memory settings in Table 1, namely $(1, 5)$, $(2, 5)$ and $(3, 5)$, represent the case of an introduced "bias" towards the hypothesis conjecturing the presence of the pattern since $N_1 < N_2 + 1$.

- The case of $(N_1, N_2) = (5, 4)$ corresponds to a balanced memory, since $N_1 = N_2 + 1 = 5$. In this case, implying that the scheme is not biased towards any of the two hypothesizes (presence or absence of the spatio-temporal pattern).

- The last three memory settings in Table 1, namely $(5, 3)$, $(5, 2)$ and $(5, 1)$, illustrate the case of a bias towards the hypothesis conjecturing the absence of the pattern since $N_1 > N_2 + 1$ in this case.

From Table 1, we remark that for the Balanced memory STPLA characterized by $N_1 = 5$ $N_2 = 4$ $(N = 5)$, we are able to obtain a very high accuracy, with the scheme producing a negligible number of superfluous notifications to the user, while alerting the user of almost all novel events, even with high degrees of both omission and inclusion errors.

	Daily Pattern			No Underlying Pattern		
	$q = 0.05$	$q = 0.1$	$q = 0.2$	$p = 0.05$	$p = 0.1$	$p = 0.2$
(N_1, N_2)						
$(1, 5)$	1.5E-8	9.9E-7	6.4E-5	0.735	0.531	0.262
$(2, 5)$	3.2E-8	2.1E-6	1.4E-4	0.983	0.925	0.680
$(3, 5)$	4.9E-8	3.3E-6	2.4E-4	0.999	0.992	0.916
$(4, 5)$	6.7E-8	4.7E-6	3.6E-4	0.999	0.999	0.982
$(5, 5)$	8.6E-8	6.2E-6	5.2E-4	0.999	0.999	0.996
$(5, 4)$	1.7E-6	6.2E-5	2.6E-3	0.999	0.999	0.997
$(5, 3)$	3.4E-5	6.2E-4	0.012	0.999	0.999	0.998
$(5, 2)$	6.9E-4	6.2E-3	0.062	0.999	0.999	0.998
$(5, 1)$	0.0137	0.059	0.254	0.999	0.999	0.999

Table 1: Alert probability P_1 under varying conditions

4.2 Learning Speed vs. Learning Accuracy

We now consider how the number of states used by the STPLA affects learning speed. Fig. 3 reports the probability of producing alerts after the introduction of a new event pattern, with a very high degree of omission errors: $q = 0.2$.

In the first set of experiments, we fixed $N_1 = 5$ while varying N_2. In order to understand the effect of the internal memory of the STPLA on the rate of convergence, we report the number of required iterations to reach a value that is 95% of the final value of P_1. In Fig. 3, we experimentally found that it took *only* 28 time instants to reach 95% of the final value of P_1 for a memory $N_2 = 1$. When we fixed N_2 to 3 in Fig. 3, 95% of P_1 was attained within 43 iterations. Similarly, we chose N_2 to be 5 and it took 67 time instants to converge to 95% of P_1.

We remark that as we increased the memory N_2, the STPLA spent more time to converge to the asymptotic value of P_1, however, it becomes more accurate. Indeed, for $N_2 = 5$, each event observed reduces the probability of alerting the user in significant leaps, until probability zero is approached (P_1 should be minimized in this case). While for $N_2 = 1$, the accuracy is significantly inferior to the one reported for $N_2 = 5$, since the final value of P_1 is 0.24. As seen, even under such extreme conditions of omission errors $q = 0.2$, the learning speed decreases slightly with the number of states, however, the accuracy increases drastically.

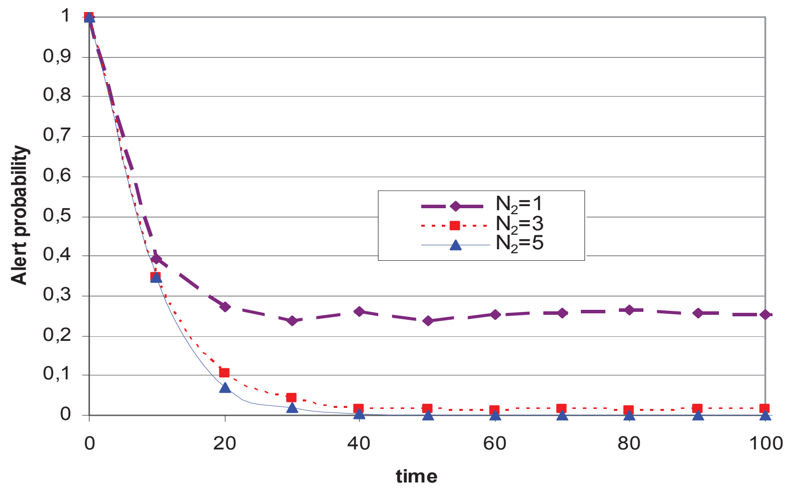


Figure 3: Evolution of the alert probability P_1 over time when a daily pattern present and with a Pattern Evaluation state space of $N_2 = 1, 3, 5$. In this setting the alert probability P_1 should be minimized in a online manner

Similarly, in Fig. 4 we observe the behaviour in a situation where novel events are occurring with probability $p = 0.2$, however, with no spatio-temporal pattern being present.

As alluded previously, when no underlying pattern is present, we would like P_1 to be maximized, and therefore P_1 shall attain 1. In the second set of experiments, we fixed $N_2 = 5$ while varying N_1. In Fig. 4, we experimentally found that it took *only* 21 time instants to reach 95% of P_1 for a memory $N_1 = 5$. In addition, for $N_1 = 5$, the accuracy is high because the final value of P_1 approaches 1.

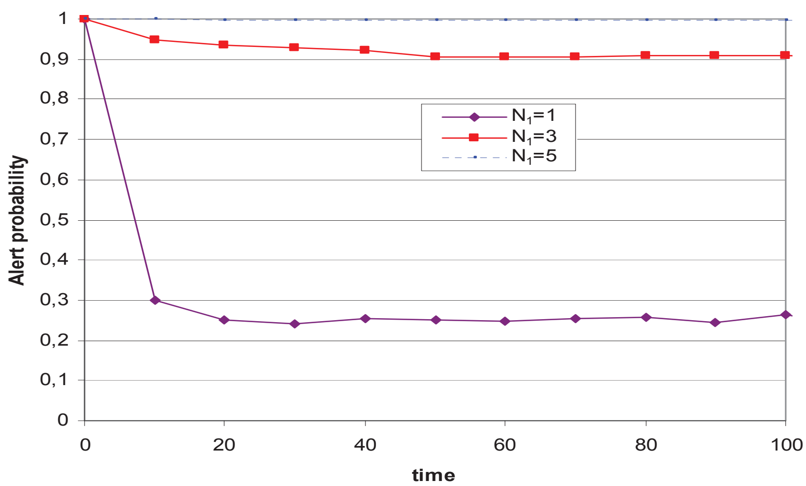


Figure 4: Evolution of the alert probability P_1 over time with novel events occurring with probability $p = 0.2$, with no daily pattern present, and with a Pattern Evaluation state space of $N_1 = 1, 3, 5$. In this setting P_1 should be maximized

In Fig. 4, 95% of the final value of P_1 was attained within 52 iterations when we fixed N_1 to 3.

Similarly, we chose N_1 to be 1 and it took 72 time instants to converge to 95% of P_1. Therefore, from this second set of experiments we conclude that when increasing the Pattern Evaluation state space N_1 from 1 to 5 states, the probability of reporting the novel events approaches unity, and the convergence rate is hastened reducing the "learning delay".

4.3 Robustness to Omission Noise

In Fig. 5, we plot the performance of STPLA when working in an environment with a spatio-temporal pattern being present, however, with events of the spatio-temporal pattern missing due to omission errors. The experiment demonstrates how the number of redundant alerts will change as the probabilities of omission error varies from $q = 0.0$ to $q = 0.2$. In this experiment, we used of a Balanced memory STPLA, with a memory depth $N = 3$. From Fig. 5, we observe that our scheme is extremely robust to different levels of omission noise. In fact, interestingly, the number of redundant alerts remains extremely negligible as we increase the omission error probability. Therefore, our STPLA achieves a near optimal performance even under an omission error probability as high as 0.2.

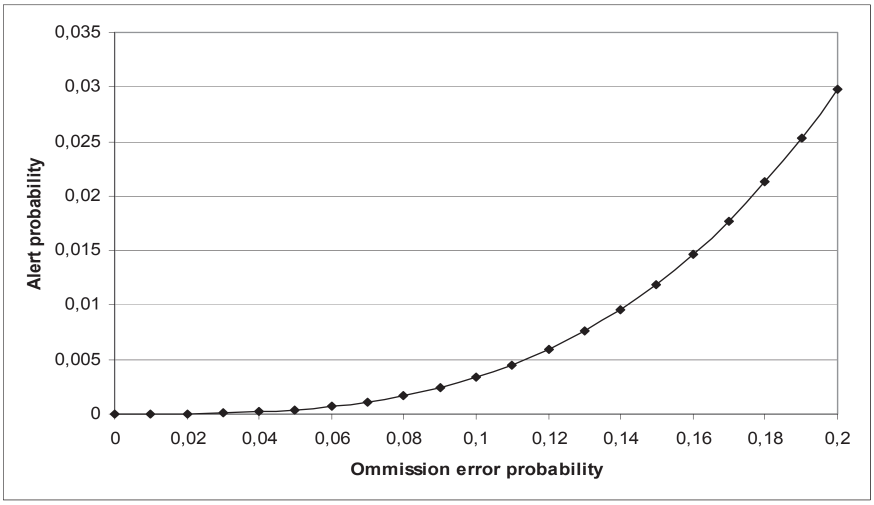


Figure 5: Alert probability P_1 plotted as a function of the omission error probability, in the case of an underlying spatio-temporal pattern

4.4 Robustness to Inclusion Noise

In Fig. 6, we plot the performance of STPLA when working in an environment in the absence of an underlying spatio-temporal pattern, however, with events of the spatio-temporal pattern present due to inclusion errors. This experiment illustrates how the number of alerts changes as the probabilities of inclusion error varies from $p = 0.0$ to $p = 0.2$. In this experiment, we made use of a Balanced memory STPLA, with a memory depth $N = 3$. As we increase the probability of inclusion error, the alert probability remains in a close neighborhood of the optimal value of alert probability in the absence of an underlying spatio-temporal pattern, which is merely 1 in this case. Therefore, we conclude that the STPLA is robust to the different levels of inclusion noise.

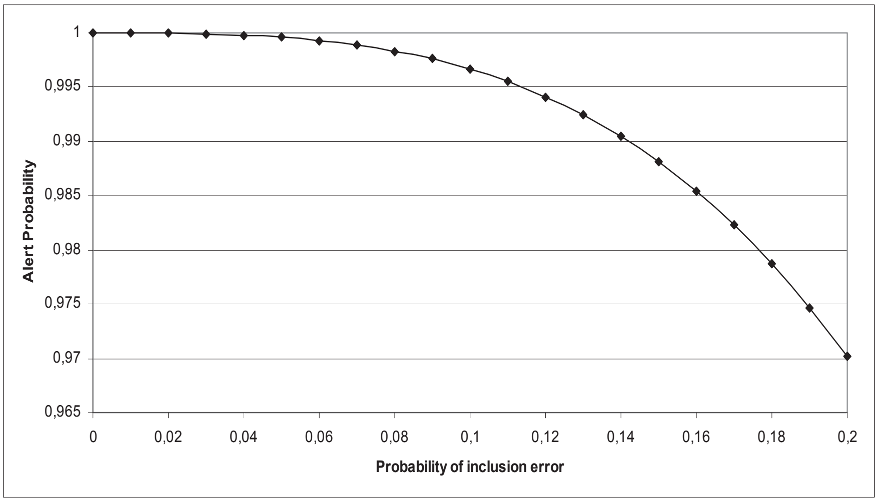


Figure 6: Alert probability P_1 plotted as a function of the inclusion error probability, in the absence of an underlying pattern

4.5 Performance in dynamic Environment

To investigate the ability of our scheme to track spatio-temporal patterns that change with time, we have conducted several experiments in dynamic environments. In all brevity, we report here a representative configuration, where spatio-temporal patterns end after a certain time period, while new ones are introduced every 100^{th} iteration. We modelled this by using an omission error probability of $q = 0.2$ when a pattern was present, and with an inclusion error probability of $p = 0.2$ when no pattern was present. We use a Balanced Memory STPLA characterized by the internal parameter $N = 5$. Employing a Balanced Memory STPLA is convenient in a non-stationary environment since it is not biased towards any of the actions $\{Notify, Suppress\}$.[9]

To be more specific, between iterations 0 and 100 we simulate a noisy daily pattern affected by an omission noise $q = 0.2$ while, between iterations 100 and 200 the daily pattern is absent, only "straggler" events can sporadically occur with an inclusion error probability $p = 0.2$, and so on. In all the experiments, the reader should note that we are using the notation iteration to denote the time instantiation measured in terms of the granularity of a day.

Fig. 7 depicts how the STPLA scheme adapts to the presence and absence of patterns over time. For instance, prior to time instant 100, the probability q was equal to 0.2, implying the presence of a daily pattern affected by omission noise. As seen, the STPLA quickly learns to suppress these events while achieving a near optimal performance despite the high omission error probability. When the

[9]Note that this experiment is different from the experiment reported in conference version of the paper where we fixed (N_1, N_2) to (3, 1).

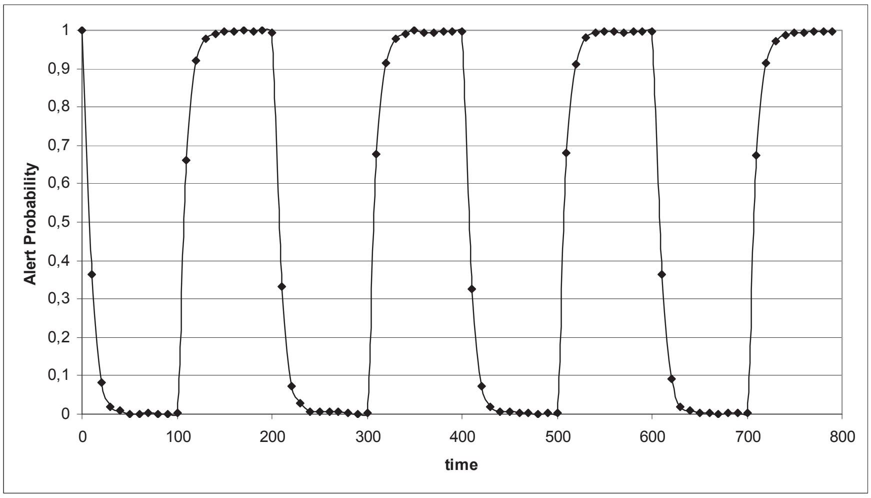


Figure 7: Evolution of the alert probability in a dynamic environment with alternating phases of presence of pattern and absence of pattern every 100^{th} iteration

pattern disappears after 100 time steps, being replaced with novel events only, we observe how quickly the STPLA changes from suppressing the events to alerting the user of them.

We thus conclude by stating that the empirical results confirm the power of STPLA both in noisy and non-stationary environments.

4.6 Comparison with FPAM

The absence of a reported literature work on *online* discovery and tracking of spatio-temporal patterns in noisy sequences presents a hindrance to comparing our STPLA to a state of art approach. To counter this problem, we introduce a modification to the FPAM algorithm [28, 27], rendering it able to operate in an *on-line* manner. In fact, a straightforward extension of FPAM is using a sliding window. It is worth noting that the concept of sliding window is akin to the idea of regular scheduled mining sessions adopted in FPAM [28, 27] in order to detect changes in the pattern. In order to decide wether an event is a part of a temporal pattern, we apply the original FPAM to mine the last W days where W is the size of the time window. To make the comparison fair, we need to choose the right internal parameters for both approaches, namely STPLA and FPAM. When it comes to the FPAM, its performance depends on the internal parameter ζ which represents a predefined threshold that determines the percentage of expected occurrences of an event. According to [28], an "optimal" choice of ζ that yields quite good performance lies between 90% and 95%. Therefore, we fixed ζ to 90% in our experimental settings. Similarly, we used a Balanced memory STPLA characterized by a memory depth $N = 5$. In fact, we have empirically found that using a Balanced Memory STPLA with a memory depth N equal to 5 gives quite good results. By adopting such "optimal" choice of the internal parameters of the STPLA and FPAM, the comparison is made meaningful. In this section, we report the experimental results of the comparison of our scheme to

Omission Noise	STPLA	FPAM
q=0.05	1.73E-6	0.062
q=0.1	6.2E-5	0.353
q=0.15	5.4E-4	0.680

Table 2: Comparing the Alert probability P_1 for the STPLA and FPAM in the presence of daily pattern and under varying omission noise.

Inclusion Noise	STPLA	FPAM
p=0.05	0.999	0.999
p=0.1	0.999	0.999
p=0.15	0.999	0.999

Table 3: Comparing the Alert probability P_1 for the STPLA and FPAM in the absence of daily pattern and under varying inclusion noise.

the online version of the FPAM algorithm.

4.6.1 Comparison under varying noise

In this experiment, we compare the asymptotic properties of our STPLA and the FPAM. We fixed the sliding window size of the FPAM to 30 days. P_1 was averaged over $1,000$ experiments, each consisting of $100,000$ iterations. In Table 2, we report the alert probability P_1 for both approaches under varying omission noise. In the experiment, P_1 should be minimized since we assume the presence of an underlying pattern, and consequently we desire to suppress the alerts. From Table 2 it is clear that the STPLA outperforms the FPAM under different levels of noises by keeping P_1 close to zero. Therefore, we conclude that the STPLA is resilient to level of omission noise. This is in contrast to the FPAM that is shown to be very sensitive to the increase of omission noise. In fact, P_1 in the case of FPAM dramatically increases from 0.062 to 0.68 as we increase the omission error probability q from 0.05 to 0.15. In other words, the FPAM fails to efficiently suppress repetitive alerts as the magnitude of omission noise increases. Therefore, a major weakness of the FPAM compared to our algorithm is its inability to efficiently cope with omission error, resulting in a low performance. We shall emphasize that the later weakness is not due to the introduction of a sliding window but is rather inherent to the FPAM algorithm [28, 27]. In fact, using a threshold ζ for detecting irregular periodicity patterns renders the scheme very prone to the increase of omission noise probability.

In Table 3, we report the alert probability P_1 for both approaches under varying inclusion noise. In the experiment, since we assume the absence of an underlying pattern, every event should be considered as novel and therefore P_1 should be maximized. From Table 3, it is clear that our STPLA and FPAM perform equally well. In fact, both schemes achieve a near optimal performance of P_1 that approaches 1.

4.6.2 Comparison under non-stationary environment

In this section, we study the effect of the sliding window size on the learning speed of the FPMA. In the experiments, we used a balanced memory STPLA with $N = 5$. We modelled the dynamic environment by using an omission error probability of $q = 0.05$ when a pattern was present, and with an inclusion error probability of $p = 0.05$ when no pattern was present. For each experiment, we conducted 1000 simulations each consisting of 800 iterations and we report the ensemble average of P_1. In order to make the comparison meaningful, the same stream of events is used in each of the 1000 simulations. Moreover , every 100^{th} iteration the environment switches. For instance, between iterations 0 and 100 the daily pattern is absent, only "straggler" events can sporadically occur with the inclusion error probability. Between iterations 100 and 200, we simulate a noisy daily pattern affected by an omission noise and so on.

Note that this model of dynamic environment follows in the same vein as in the subsection 4.5 Observe that the FPAM suppresses efficiently the alerts in the first window, but it is thereafter severely handicapped in adapting to switches in the environment. In Fig. 9 and Fig. 10, we observe that the weakness of the FPAM is accentuated in the cases in which the size of the sliding window increases. We also remark from Fig. 8 that FPAM algorithm adjusts P_1 to the optimal value in the corresponding environment much more quickly using a smaller sliding window of size 30.

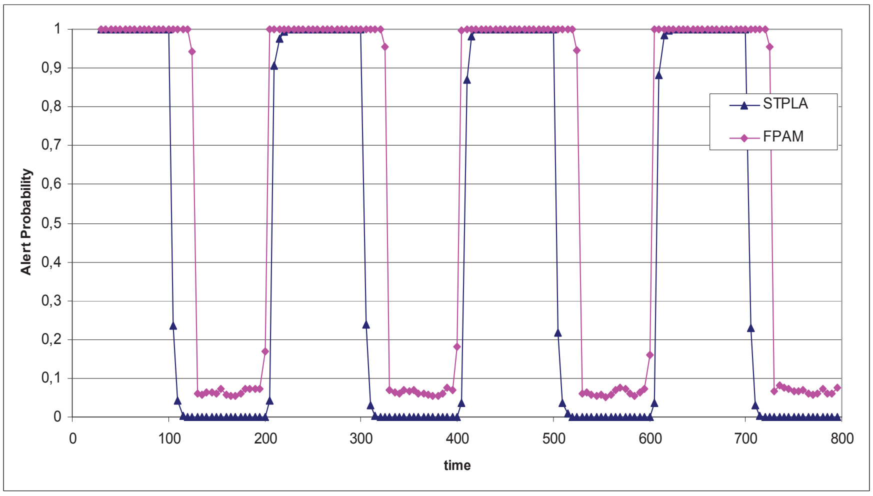


Figure 8: Performance comparison of the STPLA and FPMA in a dynamic environment with a sliding window size of 30 and a memory size 5 respectively, when the environment switches between the absence of pattern and the presence of pattern every 100^{th} iteration

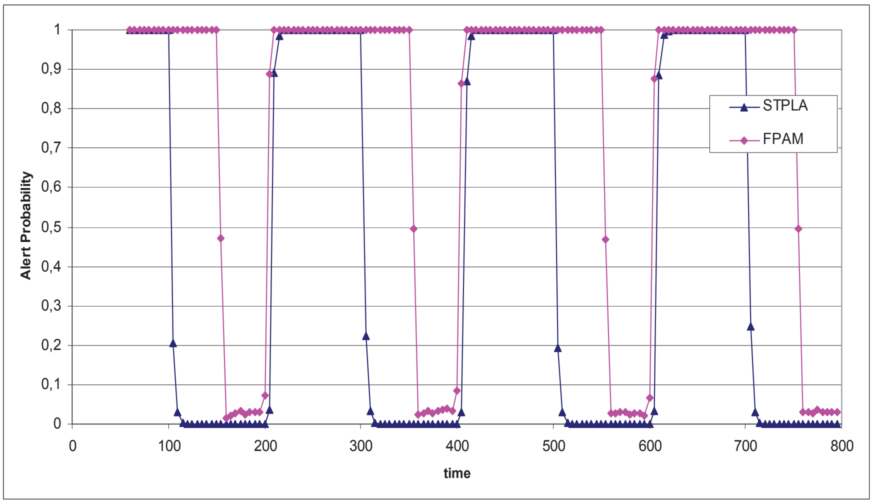


Figure 9: Performance comparison of the STPLA and FPMA in a dynamic environment with a sliding window size of 60 and a memory size 5 respectively, when the environment switches between the absence of pattern and the presence of pattern every 100^{th} iteration

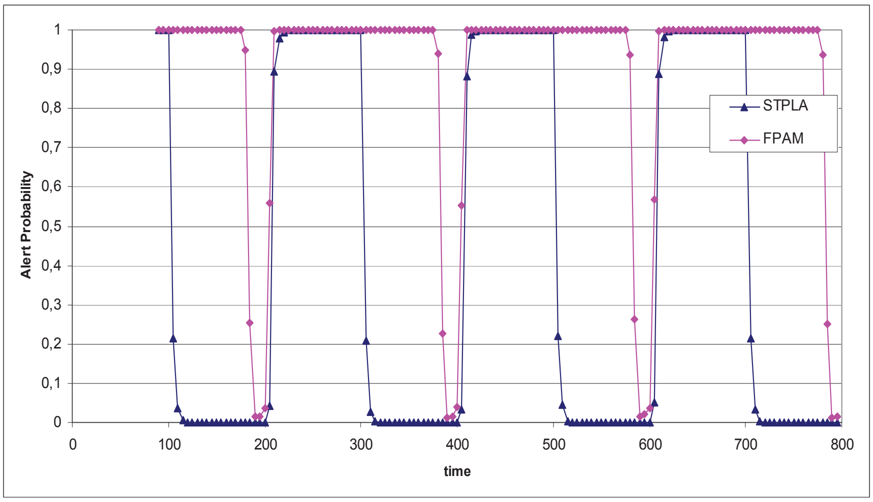


Figure 10: Performance comparison of the STPLA and FPMA in a dynamic environment with a sliding window size of 90 and a memory size 5 respectively, when the environment switches between the absence of pattern and the presence of pattern every 100^{th} iteration

From Fig. 9 and Fig. 10 we observe that while the performance of the modified FPAM in a dynamic environment is sensitive to the window size, the ability of our STPLA to adapt to changes in dynamic environments can be achieved using a fixed-memory configuration (namely N=5). fine tuning the STPLA internal parameters's (memory depth) is not required to enhance its adaptivity in dynamic environments. The simulation results suggests that a Balanced memory STPLA with memory depth from the set $\{4, 5\}$ yields quite good performance.

5 Prototype

In addition to the empirical results presented in the previous section, we have also implemented a social networking application and conducted real-life tests.

A key requirement of our community based social networking application demands that users can be made aware of the *Presence* of their friends at anytime and anywhere using their mobiles' sensing capabilities.

We now provide a brief description of our prototype, the details of whose implementation can be found in [15]. Our prototype system consists of two mobile phones: *HTC P3300* and *Sony Ericsson X1*, both of which are equipped with Wi-Fi modules. An ad-hoc network is established to provide a communication platform where our proposed solution for a "Friend Reminder" service runs.

This design is based on the "SmokeScreen" architecture [8], which introduces an effective approach to resolve privacy issues of *Presence Sharing*. In brief, we allow the user to specify exactly which of his friends can see the signal of his *Presence*. From a privacy perspective, we believe that the control of the user-related information should be fully under his own control. Thus, every user should be able to authorize the specific people who have the right to reveal his user-related information, and to also isolate other users. Accordingly, we let every pair of friends share a symmetric key. It is worth noting that the number of keys stored per user's mobile increases linearly with the number of contacts he has, while the number of keys within a group of users exhibits a quadratic growth with the group size.

The users must be synchronized to independently update the *Presence* signal and broadcast it periodically. Note that the update is deterministic so that every pair of participating users (for example Alice and Bob) can predict and interpret the time varying broadcast *Presence* signal. The *Presence* signal might vary on the hour and is known only to Alice and Bob, thus preventing impersonation attacks. As alluded to previously, we employed a symmetric key per pair of social contacts. Consequently, the size of the broadcast *Presence* signal increases linearly with the number of social contacts. In order to alleviate this problem, we have used Bloom filters to reduce the size of the *Presence* signal [7], and thus the operation of *Presence* detection reduces to the Bloom filter match operation. The choice of the memory settings depends on the application, and changing the memory settings can introduce a *bias* towards one of the two hypotheses, namely the absence or the presence of the spatio-temporal event pattern. For example, in the case of a "Reminder" , where the choice of the memory settings can be specified by the user, we suggest that if the user wants to reduce the number of notifications, it is possible then to choose N_1 and N_2 such that the memory depth of the "Suppress" action is higher than the memory depth of the "Notify" action (i.e. $N_2 + 1 > N_1$).

Based on the above architecture, we implemented our STPLA scheme on each mobile phone, allowing suppression of *Presence* notification when the *Presence* is part of a regular pattern. In all brevity, the STPLA scheme made the "Friend Notification Service" less obtrusive by only alerting the user of novel events, but suppressed alerts for regular meetings (e.g., for weekly lectures).

6 Conclusion

In this paper, we have presented the *Spatio-Temporal Pattern Learning Automaton* (STPLA) for the on-line discovery and tracking of patterns in noisy event streams. Our scheme is based on a team of finite automata, rendering it computationally efficient with a minimal memory footprint. The advantages of our approach was demonstrated through extensive simulations, as well as a prototype running on mobile devices. We provide an analytical analysis of the STPLA which was shown to confirm the experimental results. The scheme demonstrated excellent performance under different noise levels and in various dynamic settings. Thus, the power of STPLA was confirmed both in static and dynamic environments. An empirical comparison study was performed and confirms the superiority of our scheme compared to the FPAM approach [28]. We thus believe the STPLA forms an ideal framework for notification suppression in event notification based systems. In our opinion, our "Friend Reminder" prototype opens new avenues towards realizing *unobtrusive* community-based social networking applications by making use of Artificial Intelligence techniques. As a future work, we intend to extend our prototype to learning interest profiles and adaptive service recommendations.

References

[1] M. Agache and B. J. Oommen, "Generalized pursuit learning schemes: New families of continuous and discretized learning automata," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 32, no. 6, pp. 738–749, December 2002.

[2] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, vol. 43, pp. 1–43, Apr. 2011.

[3] A. F. Atlasis and A. V. Vasilakos, "The use of reinforcement learning algorithms in traffic control of high speed networks," in *Advances in Computational Intelligence and Learning*, ser. International Series in Intelligent Technologies, H.-J. Zimmermann, G. Tselentis, M. Someren, and G. Dounias, Eds. Springer Netherlands, 2002, vol. 18, pp. 353–369.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[5] C. Bellinger, "Simulation and pattern classification for rare and episodic events," Master's thesis, Carleton University, Ottawa, Ontario, 2010.

[6] C. Bellinger and B. J. Oommen, "On simulating episodic events against a background of noise-like non-episodic events," in *Proceedings of the Summer Simulation Multiconference (SummerSim '10)*, 2010, pp. 452–460.

[7] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters: A Survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, 2003.

[8] L. P. Cox, A. Dalton, and V. Marupadi, "SmokeScreen: flexible privacy controls for presence-sharing," in *Proceedings of the 5th international conference on Mobile systems, applications and services*, 2007, pp. 233–245.

[9] N. Eagle and A. Pentland, "Social serendipity: mobilizing social software," *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 28–34, 2005.

[10] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, "Project aura: Toward distraction-free pervasive computing," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22–31, 2002.

[11] L. Holmquist, J. Falk, and J. Wigström, "Supporting group collaboration with interpersonal awareness devices," *Personal and Ubiquitous Computing*, vol. 3, no. 1, pp. 13–21, Mar. 1999.

[12] J. Kabudian, M. R. Meybodi, and M. M. Homayounpour, "Applying continuous action reinforcement learning automata (CARLA) to global training of hidden markov models," in *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '04)*, 2004, pp. 638–642.

[13] S. Lakshmivarahan, *Learning Algorithms Theory and Applications*. Springer-Verlag, 1981.

[14] C. Lee, M. Chen, and C. Lin, "Progressive partition miner: an efficient algorithm for mining general temporal association rules," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 1004–1017, 2003.

[15] M. Lin and X. Wen, "Handling Relations in a Ubiquitous Computing Environments," Master's thesis, University of Agder, Norway, June 2009.

[16] S. Ma and J. L. Hellerstein, "Mining partially periodic event patterns with unknown periods," in *Proceedings of the 17th International Conference on Data Engineering*, 2001, pp. 205–214.

[17] M. R. Meybodi and H. Beigy, "New learning automata based algorithms for adaptation of backpropagation algorithm pararmeters," *International Journal of Neural Systems*, vol. 12, no. 2, pp. 45–68, 2002.

[18] S. Misra and B. J. Oommen, "GPSPA: A new adaptive algorithm for maintaining shortest path routing trees in stochastic networks," *International Journal of Communication Systems*, vol. 17, pp. 963–984, 2004.

[19] S. Misra, B. J. Oommen, S. Yanamandra, and M. S. Obaidat, "Random early detection for congestion avoidance in wired networks: A discretized pursuit learning-automata-like solution," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 40, no. 1, pp. 66–76, 2010.

[20] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Applications.* Oxford: Pergamon Press, 1994.

[21] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction.* Prentice Hall, 1989.

[22] M. S. Obaidat, G. I. Papadimitriou, A. S. Pomportsis, and H. S. Laskaridis, "Learning automata-based bus arbitration for shared-edium ATM switches," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 32, no. 6, pp. 815–820, 2002.

[23] B. J. Oommen and E. V. De St. Croix, "Graph partitioning using learning automata," *IEEE Transactions on Computers*, vol. 45, no. 2, pp. 195–208, 1996.

[24] B. J. Oommen and T. D. Roberts, "Continuous learning automata solutions to the capacity assignment problem," *IEEE Transactions on Computers*, vol. 49, no. 6, pp. 608–620, jun 2000.

[25] G. I. Papadimitriou and A. S. Pomportsis, "Learning-automata-based TDMA protocols for broadcast communication systems with bursty traffic," *IEEE Communication Letters*, vol. 4, no. 3, pp. 107–109, 2000.

[26] A. S. Poznyak and K. Najim, *Learning Automata and Stochastic Optimization.* Berlin: Springer-Verlag, 1997.

[27] P. Rashidi and D. J. Cook, "Keeping the intelligent environment resident in the loop," in *Proceedings of the 4th International Conference on Intelligent Environments*, 2008, pp. 1–9.

[28] ——, "Keeping the resident in the loop: adapting the smart home to the user," *IEEE Transactions on Systems, Man and Cybernetics: Part A*, vol. 39, no. 5, pp. 949–959, 2009.

[29] F. Seredynski, "Distributed scheduling using simple learning machines," *European Journal of Operational Research*, vol. 107, no. 2, pp. 401–413, 1998.

[30] H. Sun, V. D. Florio, N. Gui, and C. Blondia, "Promises and challenges of ambient assisted living systems," in *Proceedings of the 6th International Conference on Information Technology: New Generations (ITNG '09)*, 2009, pp. 1201–1207.

[31] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Boston: Kluwer Academic, 2003.

[32] M. L. Tsetlin, *Automaton Theory and the Modeling of Biological Systems*. New York: Academic Press, 1973.

[33] A. V. Vasilakos, M. P. Saltouros, A. F. Atlassis, and W. Pedrycz, "Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques," *IEEE Transactions on Systems, Man and Cybernetics: Part C*, vol. 33, no. 3, pp. 297–312, 2003.

[34] A. Yazidi, O.-C. Granmo, M. Lin, X. Wen, B. J. Oommen, M. Gerdes, and F. Reichert, "Learning automaton based on-line discovery and tracking of spatio-temporal event patterns," in *Proceedings of the 11th Pacific Rim international conference on Trends in artificial intelligence (PRICAI '10)*, 2010, pp. 327–338.

[35] G. M. Youngblood and D. J. Cook, "Data mining for hierarchical model creation," *IEEE Transactions on Systems, Man, and Cybernetics: Part C*, vol. 37, no. 4, pp. 561–572, 2007.

Appendix C

Paper III

Title:	An Intelligent Architecture for Service Provisioning in Pervasive Environments
Authors:	**Anis Yazidi**[1], Ole-Christoffer Granmo[1], B. John Oommen[1,2], Martin Gerdes[3] and Frank Reichert[1]
Affiliations:	[1]University of Agder, Faculty of Engineering and Science, P. O. Box 509, NO-4898 Grimstad, Norway [2]School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6 [3]Ericsson Research, Aachen, Germany.
Conference:	*Proceedings of International Symposium on Innovations in Intelligent Systems and Applications*, Istanbul, Turkey, June 2011, pp. 524-530.

An Intelligent Architecture for Service Provisioning in Pervasive Environments

Anis Yazidi
Dept. of ICT
University of Agder, Grimstad, Norway
anis.yazidi@uia.no

Ole-Christoffer Granmo
Dept. of ICT
University of Agder, Grimstad, Norway
ole.granmo@uia.no

B. John Oommen
School of Computer Science
Carleton University, Ottawa, Canada
oommen.scs@carleton.ca

Frank Reichert
Dept. of ICT
University of Agder, Grimstad, Norway
frank.reichert@uia.no

Martin Gerdes
Ericsson Research, Aachen, Germany
martin.gerdes@ericsson.com

***Abstract*—The vision of pervasive environments is being realized more than ever with the proliferation of services and computing resources located in our surrounding environments. Identifying those services that deserve the attention of the user is becoming an increasingly-challenging task. In this paper, we present an adaptive multi-criteria decision making mechanism for recommending relevant services to the mobile user. In this context, "*Relevance*" is determined based on a user-centric approach that combines both the reputation of the service, the user's current context, the user's profile, as well as a record of the history of recommendations. Our decision making mechanism is adaptive in the sense that it is able to cope with users' contexts that are changing and drifts in the users' interests, while it simultaneously can track the reputations of services, and suppress repetitive notifications based on the history of the recommendations. The paper also includes some brief but comprehensive results concerning the task of tracking service reputations by analyzing and comprehending Word-of-Mouth communications, as well as by suppressing repetitive notifications. We believe that our architecture presents a significant contribution towards realizing intelligent and personalized service provisioning in pervasive environments.**

I. Introduction

The environment in which we live in today is truly "pervasive". The proliferation of services and computing resources, indeed, makes the very dream of computing in such a pervasive environment realizable. However, this task has numerous real-life hurdles. Most prominent among these is the task of identifying those services that deserve the attention of the user. Ironically, as the services and tools become more pervasive, this task, in itself, is becoming increasingly-challenging due to the fact that:

1) The increasing number of services can overwhelm the attention of even the most educated user. It is, rather, plausible that an arbitrary user is *not even aware* of the services at his disposal.
2) The changes of a user's preferences and needs over time, renders the task of predicting his current services/interests extremely difficult.
3) The result of the interaction of the user with any specific service is usually uncertain. It surely depends on the performance of the latter. As low performance services can provoke his dissatisfaction, it is mandatory that an expedient system must be capable of identifying reputable (and disreputable) services [14], [16].

The complexity of understanding what services could be interesting and important enough to justify disturbing the user, is the main challenge of our research. To respond to this challenge, we argue that service recommendation should rely on a multi-criteria decision maker that combines different aspects (dimensions) of the system/environment in order to decide, on behalf of the user, whether a service is relevant or not. "Relevance", we propose, should be determined based on a user-centric approach that collectively combines the reputation of the service, the user's current context, the user's profile, as well as a record of the history of recommendations. This is precisely what we have attempted to achieve in our endeavor, and we thus believe that our architecture presents a significant contribution towards realizing intelligent personalized service provisioning in pervasive environments.

To clarify things, we shall present an instantiation of our architecture to a real-life, day-to-day scenario involving a proactive location-based application which provides an ensemble of services. In the scenario, the goal is to build a personalized and context-aware decision maker that delivers narrowly-targeted notifications to the user about relevant services in his environment. Nevertheless, even though this instantiation is specific, the proposed architecture is generic and can be applied to recommend a wide range of services.

Before we proceed we would like to mention that it is impossible to *comprehensively* describe the design and implementation of the entire system in a single paper. The system

B. J. Oommen is a *Chancellor's Professor*, a *Fellow : IEEE* and a *Fellow : IAPR*. The Author also holds an *Adjunct Professorship* with the Dept. of ICT, University of Agder, Norway. The first author gratefully acknowledges the financial support of the *Ericsson Research*, Aachen, Germany, and the third author is grateful for the partial support provided by NSERC, the Natural Sciences and Engineering Research Council of Canada.

978-1-61284-922-5/11/$26.00 ©2011 IEEE

which we propose contains numerous modules which deal with inter-user communications, the ranking of services, inferring the dependability of other users within a social network, communication from the system to the user, discovering and recording reputations etc. Each of these modules, in itself, is a contribution in its own right. The pertinent results describing some of these modules have already been published, and the results concerning the other modules are currently being compiled. Thus, we emphasize that while this paper contains the design and implementation details of the overall architecture, we will briefly describe the functionality of some of the component modules, and omit the details which are found in the associated citations. The reader should note that a more detailed description of all of these components and the overall system will be found in the doctoral thesis of the first author [13].

The remainder of this paper is organized as follows. In Section II, we describe some reported studies which are closely related to our approach. Then, in Section III, we present the details of the architecture by explaining the functionality of each of the components and their mutual interactions. Section IV reports the results of simulations conducted. These results demonstrate the efficiency of our design in reducing the unobtrusiveness that might be caused by traditional service recommendation systems. Concluding remarks and future lines of research are outlined in the conclusion.

II. Related work

The rich availability of services in pervasive environments has the effect of over-burdening the system's service selection task. According to the vision of pervasive computing promoted by Mark Weiser, the intention of incorporating more advanced technology should be that it provides the user the possibility of operating in a *calm* frame of mind [12]. "Increasingly, the bottleneck in computing is not its disk capacity, processor speed, or communication bandwidth, but rather the limited resource of human attention" [4]. Filtering out irrelevant information has been a focal concern in a number of studies. The main issue has been to reduce the cognitive load on the user when it comes to selecting services. It is well known that "pushing" (or downloading) notifications messages to users can cause interruptions and distractions. Users who receive irrelevant notifications may become dissatisfied with their recommendation service. According to the I-centric paradigm proposed by Wireless World Research Forum (WWRF), the service provision should be tailored to the actual needs of the user [3]. The I-centric vision promotes personalization, ambient awareness and adaptability as the core requirements of future services.

A number of studies have been performed to realize this user-centric vision. A pioneering recent work was performed by Hossain *et al*. [5]. In this work, the authors proposed a gain-based media selection mechanism. In this regard, the gains obtained by ambient media services were estimated by combining the media's reputation, the user's context and the user's profile. As a result of such a modeling process, the service selection problem was formulated as a gain maximization problem. Thereafter, a combination of a dynamic and a greedy approach was used to solve the problem. There are some fundamental differences between the study of [5] and the approach that we have proposed in this paper. From an architectural point of view, our work is based on a Publish/Subscribe paradigm in order to realize matchmaking between available services and the user's preferences[1]. Moreover, the authors of [5] did not present mechanisms to compute the reputation of the media services, thus, in effect, assuming that it is merely static. We argue that this assumption is not always valid, and that it is of paramount importance that the system tracks the variations in the reputation of the services since they, almost certainly, change over time.

A pertinent study that falls in the same class of our current work is the *Dynamos* project [10]. The *Dynamos* approach is an example of a context-aware mobile application that can be used for recommending relevant services to the user. In [10], the authors designed a hybrid recommender system for notifying users about relevant services in a context-aware manner. The model is based on a peer-to-peer social functionality model, where the users can generate contextual notes and ratings, and attach them to services, or to the environments. They are also permitted to share these with their peers. The attached notes to the environment are delivered to other users whenever they are in the spatial vicinity of the entities associated with the notes. A main difference between their work and what we propose is the way by which preferences are described. Their work assumed that the user was expected to explicitly describe his preferences by manually entering them. In this sense, the profile is defined by the user by explicitly specifying the types of activities and associating multiple interests to them. Such an approach can be considered to be a more "primitive" approach – it is not viable in pervasive environments where preferences change over time. Moreover, the issue of suppressing repetitive notifications was not addressed in [10].

A comprehensive study for personalized service provision has been performed by Naudet *et al*. from Bell Labs [8]. In [8], Naudet *et al*. designed an application for filtering the TV content provided to users' mobiles based on their learned profiles. The application is based on the use of ontologies to capture content descriptions as well as the users' interests. The latter interests are, in turn, mined using a dedicated profiling engine presented in [1], which leveraged Machine Learning (ML) techniques for user profiling.

The motivations behind our work are the following:

1) First of all, most of the reported context-aware recommendation systems do not consider the reputation of the services when issuing recommendations. In order to ensure that our hybrid recommender systems is unobtrusive, we need to locate reputable services. The success

[1] Adopting a "Push" based approach does not limit the applicability of our approach. In fact, the paradigm is still valid and can function in a "Pull" based manner, as in the *Dynamos* project [10].

of reputation systems (such as *Ebay*) suggests that there are significant latent benefits in the convergence of these ideas in pervasive environments.

2) Secondly, in order to ensure minimal user distraction, the system should be able to track the changes in a user's interests, over time. In fact, static approaches, where the user manually defines his interest's domains, are usually not expedient as the user's needs and interests change over time. Therefore, appropriate ML techniques are needed for adapting to changing interests by *inconspicuously* monitoring service usage.
3) Thirdly, repetitively reissuing the same notification regarding the same service is usually regarded as a nuisance to the user's attention. In [14], we addressed the issue of suppressing repetitive notifications in a social mobile application. With regards to recommender systems, to the best of our knowledge, the question of suppressing repetitive notifications has not been addressed before in the literature.

Stemming from these observations, we construct a hybrid recommender system that minimizes the distraction to a user's attention while, simultaneously, maximizing the hit ratio of the service notifications. In accordance with the multiple dimensions that affect the decision making process, we have also defined a set of enabler components. The synergy between these enabler components is ensured through a Publish/Subscribe architecture.

All of these issues will be crystallized in the next section where we describe the architecture of our proposed system.

III. ARCHITECTURE

The main goal of our multi-criteria decision maker is to pro-actively notify the user about relevant services. In this section, we present the different components which articulates the architecture of our system.

A. Context-Dependent Service Category Activation Rules

Within our framework, the "context" includes any information that can be used to characterize the situation of a mobile user requesting a service. It could include numerous pieces of information including the user's location (where), the time of presence (when), his current activity, his "mood" etc.

We should emphasize that in general, a user's interests are *context-dependent*. For example, recommendations about restaurants might be of interest to a certain user during weekends, when he is both close to the restaurant in question and when he is not busy. Therefore, a viable approach is to provide the user with the ability to specify that certain kinds of services (those of interest) are active in a particular context. This is the approach that we have adopted in the current study. The idea is relatively novel and has been recently applied in the *Dynamos* framework [10]. In [10], a user is permitted to specify several types of activities and their associated status, and to associate multiple interests to each of them.

With regard to specifics, in this paper, we will adopt a *two-level filtering approach* in order to support efficient matching between the available services and the user's profile. The first level of filtering is based on the user's context and is called *Context-Dependent Service Category Based Filtering*. The concepts here are akin to those found in [10], where for *each service category* (for example, restaurants, shopping, tourist attractions etc.) the user specifies the context attributes needed to make this category valid. Note that this sort of filtering is static, and can be implemented using fixed rules or stereotypes. Consequently, since the rules are static, they can be entered by the user or can be given by a template, while the names of the interests can be predefined based on a service taxonomy. From this perspective, this filtering is coarse, since we retain the *service category* such as restaurants, but do not consider refining the service recommendation by considering sub-categories of restaurants, such as Italian Pizza restaurants, Japanese restaurants, etc. In order to realize a more diversified service category matching, we integrate a wider range of pieces of contextual information, and not only location. The context attributes are mainly:

- Where: The location of the user.
- When: The time context.
- What: The activity of the user.
- What Mood: The mood of the user.

We define a function F, that statically maps a set of context attributes to a service Category as:

$$F : C_{location} \times C_{time} \times C_{activity} \times C_{mood} \mapsto ServiceCategories$$

An example of a Context-Dependent Service Categories Activation Rule, based on the inferred context attributes is:

*F(location = *, time = weekend evening, user activity = walking, mood= *) = Restaurants*

B. Learning Preferences Manager

In the previous subsection, we explained our approach to filter the available services based on their categories using Context-Dependent Service Category Activation Rules. Obviously, the category-based filtering will reduce the number of eventual services that might be of interest to the user. Nevertheless, such a filtering is coarse and needs further refinement. Therefore, we propose to carry out a second level service filtering which performs an even closer match. In this sense, the second level filtering re-filters the services *via* a finer granularity, based on the learned interests in the sub-categories. In fact, it is important that we want to model not only a *general user's interests* such as restaurants, shops, movies etc., but also the *sub-categories* of these interests that are relevant to a given user. In [15], we had presented a novel, personalized *Learning Preferences Manager* that is able to adapt to changes brought about by variations in the distribution of the user's interests, using the principles of weak estimation. This module is a fundamental component of our architecture. In the quest to learn the user's dynamic profile, the Learning Preferences Manager is guided by so-called *Relevance Feedback* (RF) [7]. In this paper, we rely on the *Service Usage History* maintained by the authors of [5], [6] as the main source of the RF. A

Service Usage History (also known as the *Interaction History*), contains the history of the services used by the user over time. For example, when the user has used a certain service at a certain time instant, the Learning Preferences Manager refines and revises the user's profile based on the current instance of the usage history, which, in turn, is automatically and unobtrusively observed in the background. To now quantify this, we have recommended the use of a Weak Estimator (devised by Oommen *et al*. [9]) so as to update the score of the data-item based on the usage history.

C. Service Reputation Manager

In this section, we introduce the Service Reputation Manager [14], [16], which is a cornerstone component of our architecture. Reputation is a particularly important criterion for filtering services.

With the abundance of services available in a pervasive environment, identifying those of high quality is a crucial task. When services are pervasive, in order to maximize the usefulness of the services accessed, the user needs to build his opinion about these services in the absence of direct experience, and as a consequence, must rely on the experiences of his acquaintances. In fact, through leveraging the power of Word-of-Mouth communications, our hybrid recommender system permits us to identify reliable services possibly deserving the user's attention. Traditional reputation systems, usually compute the reputation of a service as the average of all provided ratings. This corresponds, for instance, with the percentage of positive ratings in the *eBay* feedback form [11]. Such a simplistic approach of just blindly aggregating users' experiences may mislead the reputation system if some of the user's acquaintances are misinformed/deceptive users. Misinformed/deceptive users attempt to collectively subvert the system by providing either unfair positive ratings about a service, or by unfairly submitting negative ratings. Since an alternate way to interpret unfair ratings is to consider the unreliable referrals as coming from people with different tastes, such "deceptive" agents may even submit their inaccurate ratings innocently – due to differences in tastes. Our system can easily become intrusive and ultimately become unusable if the "trust component" (or equivalently, the Service Reputation Manager) does not deal with unfair ratings of this sort. The risk of attacks from malicious users is a crucial issue that we have incorporated in our system, which is especially pertinent in a competitive marketplace.

It is reasonable to assume that the acquaintances of the user can be divided into two classes: trustworthy acquaintances that provide accurate ratings, and unreliable acquaintances that provide unfair ratings. It follows that a good reputation manager component would seek to classify the acquaintances in one of these two classes so as to counter the detrimental effect of unfair ratings. In [14], [16], some of the authors of this present paper developed a Service Reputation Manager which is based on a concept analogous to collaborative filtering in order to separate between these two classes. The premise of the scheme in that paper was to separate the users' types by observing how they rate the same services. The latter scheme was designed in such a way that these users would be in the same group by maximizing the "within-group" similarities and minimizing the "between-group" similarities.

D. Notification Novelty Checker

The last phase of our decision maker is a module whose task is to identify if triggering a service notification will be perceived by the user as being "repetitive" information. In [17], we have argued that the user's activities can be modeled to follow some "noisy" periodic pattern, and so we can, in turn, affect the services notifications to be periodic as well. This argument will be true unless we suppress repetitive information. Consequently, any notification about a service that provides redundant information to the user can be regarded as being an unnecessary distraction. Arguing along the same vein, in this paper, we propose that we can incorporate here the same approach that we have used for suppressing repetitions in the friendly reminder application of [17]. In [17], we introduced a new scheme for discovering and tracking noisy spatio-temporal event patterns, with the purpose of suppressing re-occurring patterns, while discerning novel events. Our scheme is based on maintaining a collection of hypotheses, each one conjecturing a specific spatio-temporal event pattern. A dedicated Learning Automaton (LA) – the *Spatio-Temporal Pattern LA* (STPLA) – is associated with each hypothesis. Whenever a user receives a service notification related to a given service, a STPLA is instantiated, and this is attached to the latter service notification in order to learn the periodicity of the context in which the service is available to the user. By processing events as they unfold, we attempt to infer the correctness of each hypothesis through a real-time guided so-called random Walk/Jump process.

E. Service Notification Based on a Publish/Subcribe Paradigm

Now that the individual modules have been explained, we state that the overall architecture of or system would be as described pictorially in Figure 1.

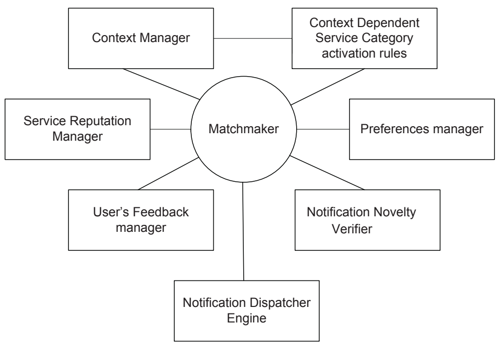


Fig. 1. The high-level architecture of our system.

Our requirement of offering highly pertinent information through a push-based approach to the user can be well supported through the Publish/Subscribe paradigm [2]. The system can be deployed using a Publish/Subscribe System, which puts all the pieces of this puzzle together. A Publish/Subcribe model consists of information providers, who publish events to the system, and of information consumers, who subscribe to events of interest within the system. A Publish/Subscribe architecture ensures the timely notification of events to the interested subscribers. Note too that the use of a Publish/Subcrsibe server will enhance privacy, since no user-sensitive private information need be transmitted to the service providers.

A notification is issued whenever the service matches the user's subscription. In other words, this occurs whenever the following conditions should be met:

- A spatial filter reports that the service is in the user's vicinity. We agree that in the case of location-based services, the knowledge of the user's context is the most differentiating information within this context.
- The Service Reputation module returns the truth value of whether the service is reputable, as per the approach defined in [14], [16], and briefly explained in Section III-C.
- The service description matches the user's profile according to the above-mentioned two-level filtering approach. To identify service items of interest, the matching process consists of two steps. First, for each service, its associated category is matched with the set of active service categories. These service categories, generally, specify the business branches of the service (e.g., Restaurants, Shops). After applying the context-dependent category activation rules, only the services belonging to the active categories are maintained. Moreover, the service subcategory should match the second filter characterized by a finer granularity, namely the Learning Preferences Manager.
- The Novelty Detection module reports that the eventual service notification would not be repetitive by checking wether the notification is a part of a spatio-temporal pattern.

IV. Experimental results

To demonstrate the proof of these concepts, in this section, we present results of simulations that we have conducted, that puts into a nutshell all the components of the proposed architecture. To do this, we have adopted a Discrete Event Simulation methodology. The performance metric to assess our architecture is the hit ratio, denoted $\alpha(t_n)$, and defined at any given time instance, t_n, as the ratio of the relevant notifications delivered to the user at time t_n. We define a relevant (or equivalently, non-distractive) notification as one where:

- The service matches the user's profile
- The notification is not repetitive
- The user's interaction with the service leads to the user's satisfaction.

By virtue of the above, we consequently regard a distractive notification as one that is either repetitive, or if the interaction with the service does not lead to the user's satisfaction due to its low performance value [14], [16], or if the recommended service does not match the current user's interests.

In the same vein, we define the "Distraction" ratio (denoted $\beta(t_n)$), at any given time instance t_n, as the ratio of distractive notifications delivered to the user at time t_n. Clearly $\alpha(t_n) + \beta(t_n) = 1$.

To demonstrate the power of our architecture, we compare our approach to an *Unguided Recommendation System*, that delivers to the user notifications regarding services that match only his contextual preferences based on the Context Based Service Category static filtering. In other words, we suppose that the *Unguided Recommendation System* performs only coarse contextual filtering. For example, in the case of the notification of location-based services, the *Unguided Recommendation System* sends restaurant suggestions every time the user is close to a restaurant without learning his profile, without suppressing repetitive alerts and without checking the reputation of the service. In our simulation, we considered delivering only a single notification per location[2].

We assume that the user's mobility follows a given noisy periodic spatio-temporal pattern. As explained previously in Section III-D, the location and time primitives are combined from their cross-product spaces to produce spatio-temporal patterns. Let us suppose that the mobile user in question, u, visits a given location R according to a weekly spatio-temporal pattern characterized by an omission noise $q = 0.1$. We suppose that a pool of services S is available in the visited area, for eventual access by the user.

At this juncture, it is important to remind the reader that, for the sake of clarity, we use two time granularities (or two time scales) for different events in our Discrete Event Simulation model. In fact, at the granularity of a week, namely at time instances t_n (n denotes the week index), the user visits the location R, and therefore, it is likely that service notifications can take place. On the other hand, at the lower time scale (or equivalently, at the finer time granularity) of a day, we assume that other possible events can take place, such as the generation of Relevance Feedback that serves as input to the Learning Preferences Manager, or the submission of a service rating by user in $\mathcal{U}$ that serves as input to the Service Reputation Manager.

We further assume that the mobile user u possesses a set of acquaintances $\mathcal{U}$ that communicate their experiences regarding the performance of the available pool of services, S. We assume that at discrete time instances, the acquaintances in $\mathcal{U}$ communicate their ratings to u. In the absence of direct experience from the user, the feedback provided by the acquaintances serves as input to the Service Reputation Manager, referred to in Section III-C.

For the sake of clarity and simplicity, we assume that the user preferences fall into two categories C_1 and C_2. We further assume that the underlying distribution of the weights of the

[2]It is possible to adopt a top-N recommendation approach in order to not overwhelm the user with a long list of services and thus limit the size of the list.

preferences that reflect the affinity of user's interest in each of the preferences categories C_1 and C_2 follows a binomial distribution [15]. Therefore, the problem of estimating the user's interests in this particular case is modeled as the estimation of the parameters for binomial random variables. The Relevance Feedback concerning the preferences categories C_1 and C_2 is generated according to the true underlying value of s_1 and s_2. The intention of the Learning Preferences Manager is to estimate S, i.e., s_i for $i = 1, 2$. We achieve this by maintaining a running estimate $P(n) = [p_1(n), p_2(n)]^T$ of S, where $p_i(n)$ is the estimate of s_i at time granularity 'n', where n denotes the day index. Note that we assume that the Relevance Feedback is available at the finer time granularity of a day.

If $s_i > s_j$, we say that category C_i represents the user's preferred interest category, and thus assume that only services that belong to category C_i are of interest to the user. All the services belonging to category C_j will not be of interest to the user, and notifying him about these services will result in a distraction. Consequently, the matchmaking of the preferences will rely on the same simple mechanism, and recommend the services whose estimated category weight is larger between the two categories. The reader should observe that this simple rule is similar to decision rules in classifiers, where the decision maker has to decide on a hypothesis on the state of nature between two exclusive hypotheses. However, a more sophisticated preferences matchmaking approach, analogous to the one in [5], [6] that is based on assessing a linear combination of the weights, can be easily adopted in combination with our Learning Preferences Manager [15].

An important parameter that must be specified is the rate at which the Relevance Feedback occurs. We suppose that at the finer time granularity of a day, a Relevance Feedback is generated according the underlying distribution, S. Therefore, the estimated weights of C_1 and C_2 are tracked and updated at the granularity of a day.

We further model the performances of services as either being High Performance or Low Performance as reported in [14], [16]. We also assume that the services either belong to C_1 or C_2. Therefore, we will have a combination of 4 exclusive classes of services in the current experiments:

- 25 High performance services that belong to C_1
- 25 High performance services that belong to C_2
- 25 Low performance services that belong to C_1
- 25 Low performance services that belong to C_2.

If, for example, C_1 represents the current preferred interest category, the Recommendation System will recommend services to the user that are both of high performance, and that belong to category C_1. In the simulation settings, we assume that the user possesses 40 acquaintances in his social network – 20 of which are deceptive and the remaining 20 are trustworthy [14], [16]. Furthermore, the trustworthy user's acquaintances are characterized with $p = 0.8$, while the deceptive ones have $p = 0.2$. In all the experiments, we configure the STPLA with $N_1 = 5$ and $N_2 = 5$. The high performance services have an performance probability of 0.8, while the low performance services have are characterized by the performance probability of 0.2 [14], [16].

As alluded to previously, we suppose that the user's mobility follows a weekly periodic noisy pattern, and thus we conducted the simulations for a period of 40 week instances.

We report now the results obtained by testing our proposed architecture in a variety of settings[3].

In this experiment, we compared the distraction ratio as well as the hit ratio obtained by our approach with the respective ratios obtained by utilizing an Unguided Recommendation System. The results were obtained from an ensemble of 100 simulations, and we report $\alpha(t_n)$, where n denotes the week index. In Figure 2(a), we report the hit ratio, and in Figure 2(b), we report the distraction ratio. The preferences were assumed static, and thus, in other words, we employed the same underlying distribution for the weights of the preferences. We also assumed that the current preferred services category was C_1. We supposed that at a finer time granularity, namely, at a daily basis, each of the acquaintances submitted a rating for a randomly chosen service among the pool of available services. We observe from Figure 2(b) that the distraction ratio asymptotically approaches the value 0.2 and that the hit ratio approaches the value 0.8. These values can be explained by the fact that our architecture tends to recommend only the 25 High performance services that belong to C_1 as time advances.

Furthermore, we remark from Figure 2(a) and its counterpart Figure 2(b) that the performances achieved by utilizing our proposed architecture improves almost uniformly over time, and that it outperforms the Unguided Recommendation System.

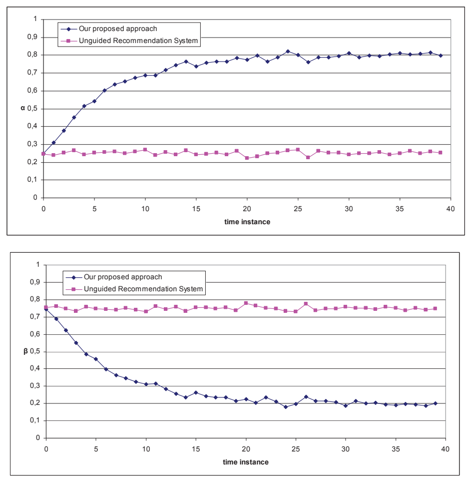


Fig. 2. (a) The evolution of the hit ratio in the case of our proposed approach and the Unguided Recommendation System, when the performance probabilities of the high and low performance services are 0.8 and 0.2 respectively. (b) The evolution of the distraction ratio in the case of our proposed approach and the Unguided Recommendation System for the same settings.

[3]We have done experiments for numerous settings and scenarios. For the sake of brevity, we report in this paper few of them, more simulations results are found in [**?**].

The results for another set of experiments are shown in Figure 3(a) and Figure3(b). In these experiments, we changed the settings by assuming that the high performance services had an performance probability of 0.7, while the low performance services were characterized by the performance probability of 0.3. Whereas in Figure 3(a), we report the hit ratio, in Figure 3(b), we report the distraction ratio. As in the case of the previous figures, the convergence of the graphs to their optimal levels is clear from these figures too.

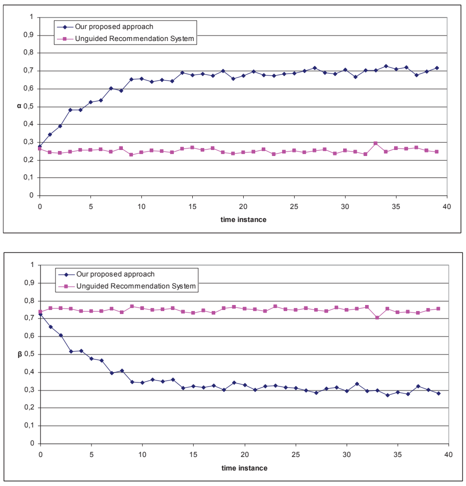


Fig. 3. (a) The evolution of the hit ratio in the case of our proposed approach and the Unguided Recommendation System, when the performance probabilities of the high and low performance services are 0.7 and 0.3 respectively. (b) The evolution of the distraction ratio in the case of our proposed approach and the Unguided Recommendation System for the same settings.

V. Conclusion

In this paper we have considered the problem of computing in pervasive environments, and in particular, in identifying those services that deserve the attention of the user. We have presented an adaptive multi-criteria decision making mechanism for recommending relevant services to the mobile user, where "*Relevance*" is determined based on a user-centric approach that combines both the reputation of the service, the user's current context, the user's profile, as well as a record of the history of recommendations. We have proposed the architecture of a system that builds a personalized and context-aware application that delivers narrowly targeted information to the user, while being unobtrusive. The design avoids flooding the user with irrelevant information. We have conducted simulations and reported results that suggest that our architecture can significantly reduce unobtrusiveness. To gain more insights into the acceptance of the system by an end user, in the future, we propose that the system be deployed into a real-life application domain, which also incorporates a user study.

References

[1] A. Aghasaryan, S. Betgé-Brezetz, C. Senot, and Y. Toms, "A profiling engine for converged service delivery platforms," *Bell Lab. Tech. J.*, vol. 13, no. 2, pp. 93–103, 2008.

[2] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra, "Matching events in a content-based subscription system," in *PODC '99: Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing*. New York, NY, USA: ACM, 1999, pp. 53–61.

[3] S. Arbanowski, P. Ballon, K. David, O. Droegehorn, H. Eertink, W. Kellerer, H. van Kranenburg, K. Raatikainen, and R. Popescu-Zeletin, "I-centric communications: Personalization, ambient awareness, and adaptability for future mobile services," *IEEE Communications Magazine*, vol. 42, no. 9, pp. 63–69, 2004.

[4] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, "Project aura: Toward distraction-free pervasive computing," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22–31, 2002.

[5] M. A. Hossain, P. K. Atrey, and A. El Saddik, "Gain-based selection of ambient media services in pervasive environments," *Mob. Netw. Appl.*, vol. 13, no. 6, pp. 599–613, 2008.

[6] M. A. Hossain, J. Parra, P. K. Atrey, and A. El Saddik, "A framework for human-centered provisioning of ambient media services," *Multimedia Tools and Applications*, vol. 44, pp. 407–431, 2009.

[7] M. Montaner, B. Lopez, and J. L. de la Rosa, "A taxonomy of recommender agents on the internet," *Artificial Intelligence Review*, vol. 19, pp. 285–330, 2003.

[8] Y. Naudet, A. Aghasaryanb, S. Mignon, Y. Toms, and C. Senot, "Ontology-based profiling and recommendations for mobile tv," in *Semantics in Adaptive and Personalized Services*, ser. Studies in Computational Intelligence, M. Wallace, I. Anagnostopoulos, P. Mylonas, and M. Bielikova, Eds. Springer Berlin / Heidelberg, 2010, vol. 279, pp. 23–48.

[9] B. J. Oommen and L. Rueda, "Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments," *Pattern Recogn.*, vol. 39, no. 3, pp. 328–341, 2006.

[10] O. Riva and S. Toivonen, "The dynamos approach to support context-aware service provisioning in mobile environments," *J. Syst. Softw.*, vol. 80, no. 12, pp. 1956–1972, 2007.

[11] A. Schlosser, M. Voss, and L. Brckner, "On the simulation of global reputation systems," *Journal of Artificial Societies and Social Simulation*, vol. 9, 2005.

[12] M. Weiser, "The computer for the twenty-first century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[13] A. Yazidi, "Intelligent learning automata-based strategies applied to personalized service provisioning in pervasive environments," Ph.D. dissertation, Department of ICT, University of Agder, Grimstad, Norway, 2011.

[14] A. Yazidi, O.-C. Granmo, M. Lin, X. Wen, B. J. Oommen, M. Gerdes, and F. Reichert, "Learning automaton based on-line discovery and tracking of spatio-temporal event patterns," in *PRICAI 2010: Trends in Artificial Intelligence*, ser. Lecture Notes in Computer Science, B.-T. Zhang and M. Orgun, Eds. Springer Berlin / Heidelberg, 2010, vol. 6230, pp. 327–338.

[15] A. Yazidi, O.-C. Granmo, and B. J. Oommen, "An adaptive approach to learning the preferences of users in a social network using weak estimators," *Submitted for publication*.

[16] ——, "Service selection in stochastic environments: a learning-automaton based solution," *To Appear in Applied Intelligence*.

[17] ——, "A learning automata based solution to service selection in stochastic environments." in *Proceedings of the Twenty Second International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems (IEA-AIE 2010)*, ser. Lecture Notes in Artificial Intelligence, 2010, pp. 209–218.

Appendix D

Paper IV

Title:	An Adaptive Approach to Learning the Preferences of Users in a Social Network Using Weak Estimators
Authors:	**Anis Yazidi**[1], Ole-Christoffer Granmo[2] and B. John Oommen[1,2]
Affiliations:	[1]University of Agder, Faculty of Engineering and Science, P. O. Box 509, NO-4898 Grimstad, Norway [2]School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6
Journal:	*To appear in Journal of Information Processing Systems. (Accepted March 16, 2012).*

An Adaptive Approach to Learning the Preferences of Users in a Social Network Using Weak Estimators

Anis Yazidi*, Ole-Christoffer Granmo†, B. John Oommen‡

Abstract

Since a social network, by definition, is so diverse, the problem of estimating the preferences of its users is becoming increasingly essential for personalized applications which range from service recommender systems to the targeted advertising of services. However, unlike traditional estimation problems where the underlying target distribution is stationary, estimating a user's interests, typically, involves non-stationary distributions. The consequent time varying nature of the distribution to be tracked imposes stringent constraints on the "*unlearning*" capabilities of the estimator used. Therefore, resorting to strong estimators that converge with probability 1 is inefficient since they rely on the assumption that the distribution of the user's preferences is stationary. In this vein, we propose to use a family of stochastic-learning based *Weak* estimators for learning and tracking user's time varying interests. Experimental results demonstrate that our proposed paradigm outperforms some of the traditional legacy approaches that represent the state-of-the-art.

Keywords : *Weak estimators, User's Profiling, Time Varying Preferences*

1 Introduction:

Utilizing the power of the internet to affect marketing, business and politics *via* strategies applicable for social networking, is becoming increasingly important, especially in a user-driven universe. Over the last few years, the issue of maintaining users' profiles has become more crucial for designing and streamlining personalized applications ranging from service recommender systems to the advertising of targeted services. Mastering and optimally utilizing the knowledge about a user's interests has led to promising applications in filtering and recommending documents [4], multimedia [6] and TV

*This author can be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: `anis.yazidi@uia.no`.

†This author can also be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: `ole.granmo@uia.no`.

‡*Chancellor's Professor* ; *Fellow: IEEE* and *Fellow: IAPR*. This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. E-mail address: `oommen@scs.carleton.ca`.

programs [17], based on their respective contents. For instance, a comprehensive study for personalized service provisioning was performed by Naudet *et al.* from Bell Labs [17], where the authors designed an application for filtering the TV content provided to users' mobile devices based on their learned profiles. The application is based on the use of ontologies to capture content descriptions as well as the users' interests. The latter interests are, in turn, mined using a dedicated profiling engine presented in [1], which leveraged Machine Learning (ML) techniques for user profiling. The work reported in [25] presented a "product" that recommends vendors' web pages by measuring the similarity between the user's profile and the vendor's web page when the user is in the vicinity of the vendor (seller). The user's profile is constructed through mining the history of his web log. Another example that falls in the class of mobility-aware applications is the *PLIGRIM* system, which makes use of the user's location to recommend relevant web links [3]. In the same vein, the *SMMART* framework dynamically locates products that match the shopping preferences of a mobile user [11].

Usually, constructing a user's profile involves applying estimation techniques to leverage the knowledge about his interests, which, in turn, is gleaned from the history of the services that he utilizes [6, 7]. A number of previous studies [10] have shown that a user's interests are not constant over time, and consequently, paradigms which are to be promising, should take into account the drift of these interests. The time varying nature of the distribution of the user's interests renders the problem of estimating them both difficult and non-trivial.

Tracking the dynamics of a user's interests is akin to a well-known problem in statistical *Pattern Recognition* (PR), namely that of estimating non-stationary distributions. Traditionally available methods that cope with non-stationary distributions resort to the so-called *sliding window* approach, which is a limited-time variant of the well-known Maximum Likelihood Estimation (MLE) scheme. The latter model is useful for discounting stale data in data stream observations. Data samples arrive continually and only the most recent observations are used to compute the current estimates. Any data occurring outside the current window is forgotten and replaced by the new data. The problem with using sliding windows is the following: If the time window is too small the corresponding estimates tend to be poor. As opposed to this, if time window is too large, the estimates prior to the change of the parameter have too much influence on the new estimates. Moreover, the observations during the entire window width must be maintained and updated during the process of estimation.

In earlier works [8, 9, 10], Koychev *et al.* introduced the concept of Gradual Forgetting (GF). The GF process relies on assigning weights that decrease over time to the observations. In this sense, the GF approach assigns most weight to the more recent observations, and a lower weight to the more-distant observations. Hence, the influence of old observations (on the running estimates) decreases with time. It was shown in [10] that the GF can be an enhancement to the sliding window paradigm. In this sense, observations within each sliding window are weighted using a GF function.

Recently, Oommen and Rueda [19] have proposed a strategy by which the parameters of a binomial/multinomial distribution can be estimated when the underlying distribution is non-stationary. The method is referred to as Stochastic Learning Weak Estimation (SLWE), and is based on the principles of stochastic Learning Automata (LA) [16, 23]. The SLWE has found successful applications in many real-life problems that involve estimating distributions in non-stationary environments such as in adaptive encoding [20], route selection in mobile ad-hoc networks [18], and topic detection and tracking in multilingual online discussions [22]. Motivated by these successful applications of the SLWE in various areas, in the course of this study, we consider employing the SLWE for solving the intriguing problem of tracking user's interests. The objective of the paper is to present a personalized *Learning Preferences Manager*, a *modus operandus* for capturing user's preferences. The latter will be able to cope with changes brought about by variations in the distribution of the user's interests, which will be where the SLWE plays a prominent part.

The rationale for choosing a weak estimator for non-stationary environments is that estimators that converge with probability 1 (e.g. the MLE and Bayesian estimates) cannot easily unlearn and adapt to the drift in the interests. In our opinion, the appealing properties of the SLWE lies in its recursive *multiplication*-based update form, that achieves the process of unlearning stale data, by an order-of-magnitude faster than a traditional *addition*-based updating scheme. Moreover, unlike other legacy state-of-the-art approaches, the adaptability of the SLWE and its capability to cope with changing environments, can be achieved using the choice of the scheme's fixed internal parameter, λ. In fact, the choice λ is not a critical issue, and it does not influence the convergence speed of the SLWE. Indeed, we present simulations results that demonstrate the superiority of the SLWE compared to GF, the sliding window, and the approach presented by Hossain *et al.* [7], which represent the state-of-the-art.

1.1 Contribution

The novel contributions of this paper, when it concerns user's profiling techniques, are the following:

- To the best of our knowledge, our current work presents the first attempt to apply a LA-inspired approach, such as the SLWE, to the real-life problem of tracking user's interests. We hope that the current study paves the way towards more applications of LA-based techniques to the realm of user profiling.

- Philosophically, our profile representation model is distantly related to the approach presented in [6, 7], where the authors utilized the history to update the affinity of the user's interests. However, a substantial difference from the latter studies is our novel categorization of the data items that constitute a profile, into its so-called *disjunctive* and *conjunctive* data items. To the best of our knowledge, although profile update approaches in which the data items which are disjunctive have received a significant interest, the case of conjunctive data items remains largely unaddressed. In this work, we propose an adequate update form based on the principles of the SLWE, for each case of these two cases[1].

- The model which we have adopted, namely that of the user's interests changing "abruptly", is, in itself, interesting. In fact, instead of presuming that the so-called environment's "switch" occurs with some fixed periodicity [19], we assume that changes in the distribution of the user's interests occur at *unknown random time instants.* Furthermore, we suppose that the distribution changes to a possibly new random distribution after the "switch". Such a model of the distribution's versatility is more realistic than the one which possesses a fixed periodicity-based changing model, and this is thus more appropriate in the context of estimating the user's preferences. Clearly, the described settings represents a particularly challenging scenario for any approach which models and studies change detection! The experiments conducted and the results reported, demonstrate that our approach exhibits lower error and faster adaptivity than the state-of-the-art.

- The model and technique which we have used here (for learning user's preferences) has been incorporated into a more comprehensive system whose architecture, design and implementation details are found in [26].

1.2 Paper Organization

The rest of the paper is organized as follows. In Section 2, we report a brief survey of the available results in tracking user's interests. Then, in Section 3, we present some of the theoretical properties of the SLWE. In Section 4, we introduce a formal model of the user's preferences as well as an SLWE-based solution to the problem. Experimental results obtained by rigorously testing our solution for a variety of scenarios are presented in Section 5. Section 6 concludes the paper.

2 State of The Art

The core function of a personalized *Learning Preferences Manager* is to update the user's profile in a dynamic and incremental way. This is done so that the "Manager" can closely follow the real-time evolution of the user's interests. In fact, any user's interests are not constant over time, and therefore it is imperative that the system takes the profile's drift into account. In this sense, whenever one attempts to represent the user's *current* interests, the most recent observations are more reliable than older ones. From a more general perspective, the task of learning the drifts

[1]The reader will observe that the application of the SLWE has already been reported in the literature. However, its use for conjunctive and disjunctive data items is totally new.

in the user's interests corresponds to the problem of learning evolving concepts [24]. There are several studies that have dealt with the task of learning a user's interests. These include the use of a sliding window [14], aging examples [12], and a Gradual Forgetting (GF) function [8, 9, 10] etc. However, of all these, a sliding window approach is the most popular one. It consists of learning the description of the user's interests from the most recent observations, and thereafter, of discarding the observations that fall outside the window.

A substantial shortcoming of the sliding window approach is the choice of the window size. In [14], the authors adopted a fixed-size time window in order to learn a user's scheduling preferences. They empirically determined that a window size of 180 was a proper choice for their particular scheduling application. The GF, on the other hand, relies on assigning weights to the observations that decrease over time. Hence, the influence of older (more "stale") observations on the running estimates, decreases with time. The authors of [10] suggested a linearly-decreasing function, $w = f(t)$, for decaying the relative weights of the GF as follows:

$$w_i = \frac{-2k}{n-1}(i-1) + 1 + k, \tag{1}$$

where i denotes a counter of observations starting from the most recent one, n is the number of observations, $k \in [0, 1]$ is a parameter that represents the percentage by which the weight of any subsequent observation is decreased, and consequently the percentage by which the weight of the most recent one, in comparison to the average, is increased. Thus k is a parameter that controls the slope of the forgetting function.

In order to achieve a synergy between both the two approaches, namely GF and sliding window, Koychev in [10], proposed to apply the GF *within each sliding window*. Thus, in this case, the parameter n (i.e., the length of the observation sequence) in equation (1) was set to be equal to L, where L denotes the length of the window.

Apart from the sliding window and GF schemes, other approaches, which also deal with *change detection*, have also emerged. In general, there are two major competitive sequential change-point detection algorithms: Page's cumulative sum (CUSUM) [2] detection procedure and the Shiryaev−Roberts−Pollak detection procedure. In [21], Shiryayev used a Bayesian approach to detect changes in the parameters distribution, where the change points were assumed to obey a geometric distribution. CUMSUM is motivated by a maximum likelihood ratio test for the hypotheses that a change occurred. Both approaches utilize the log-likelihood ratio for the hypotheses that the change occurred at the point, and that there is no change. Inherent limitations of CUMSUM and the Shiryaev−Roberts−Pollak approaches for on-line implementation are the demanding computational and memory requirements. In contrast to the CUMSU and the Shiryaev−Roberts−Pollak, the SLWE avoids the intensive computations of ratios, and do not invoke hypothesis testing.

A particularly interesting recent study for learning user's interests in ambient media services (and in, consequently, locating relevant services) was reported in [7]. Hossain *et al* devised the so-called Ambient Media Score Update method, which we shall refer to as SU for the rest of the paper. The SU method was used to learn a user's changing interests [6, 7] by recording the so-called "scores", which represented his/her affinity of interests. In order to follow closely the evolution of the scores, the authors of [7] refined their proposed updating method defined earlier in [6] and updated the scores of the services at every time instant whenever the service was used. This was done instead of performing updates in a batch mode [6].

We shall now discuss the family of weak estimators alluded to earlier, and proceed to explain how they can be used to solve the problem currently being studied.

3 Weak Estimators of Multinomial Distributions

The problem of estimating the parameters of a multinomial distribution has been efficiently solved by the recently introduced SLWE [19]. The multinomial distribution is characterized by two parameters, namely, the number of trials, and a probability vector that determines the probability of a specific event (from a prespecified set of events) occurring. In this regard, we assume that the number of observations is the number of trials. Therefore, the problem is to estimate the latter probability vector associated with the set of possible outcomes or trials. Thus, we encounter the problem of estimating the latter probability *vector* associated with the set of possible outcomes.

Specifically, let X be a multinomially distributed random variable, which takes on the values from the set $\{`1', \ldots, `r'\}$. We assume that X is governed by the distribution $S = [s_1, \ldots, s_r]^T$ as follows:

$X = `i'$ with probability s_i, where $\sum_{i=1}^{r} s_i = 1$.

Also, let $x(n)$ be a concrete realization of X at time 'n'. The intention of the exercise is to estimate S, i.e., s_i for $i = 1, \ldots, r$. We achieve this by maintaining a running estimate $P(n) = [p_1(n), \ldots, p_r(n)]^T$ of S, where $p_i(n)$ is the estimate of s_i at time 'n', for $i = 1, \ldots, r$. We omit the reference to time 'n' in $P(n)$ whenever there is no confusion. Then, the value of $p_i(n)$ is updated as per the following simple rule (the rules for other values of $p_j(n), j \neq i$, are similar):

$$p_i(n+1) \leftarrow p_i + (1-\lambda)\sum_{j \neq i} p_j \quad \text{when } x(n) = i \tag{2}$$

$$\leftarrow \lambda p_i \quad \text{when } x(n) \neq i. \tag{3}$$

The properties of the estimator are catalogued below.

Theorem 1. *Let the parameter S of the multinomial distribution be estimated by $P(n)$ at time 'n'* *as per equations (2) and (3). Then,* $E[P(\infty)] = S$.

Remark:Theorem 1 explicitly states that rules (2) and (3) lead to a mean probability vector which asymptotically converges to the actual unknown probabilities. Although this behavior is asymptotic, we will empirically show presently that the rule quickly adapts to the changes in the distribution and, hence, works efficiently in non-stationary environments.

Proof. First of all, we can rewrite the updating rules given in equations (2) and (3) as follows:

$$\begin{aligned} p_i(n+1) &\leftarrow p_i + (1-\lambda)(1-p_i) \quad \text{w.p. } s_i && (4)\\ &\leftarrow \lambda p_i \quad\quad\quad\quad\quad\quad \text{w.p. } \sum_{j\neq i} s_j\,. && (5)\end{aligned}$$

Thus, the expected value of $p_i(n+1)$ given the estimated probabilities at time 'n', P, is:

$$\begin{aligned} \mathrm{E}\left[p_i(n+1)|P\right] &= p_i s_i + (1-\lambda-p_i+\lambda p_i)s_i + \lambda p_i(1-s_i) && (6)\\ &= p_i s_i + s_i - \lambda s_i - p_i s_i + \lambda p_i s_i + \lambda p_i - \lambda p_i s_i && (7)\\ &= (1-\lambda)s_i + \lambda p_i\,. && (8)\end{aligned}$$

Taking expectations a second time, we have:

$$\mathrm{E}[p_i(n+1)] = (1-\lambda)s_i + \lambda \mathrm{E}[p_i(n)]\,. \quad (9)$$

As $n \to \infty$, both equations $\mathrm{E}\left[p_i(n+1)\right]$ and $\mathrm{E}\left[p_i(n)\right]$ converge[2] to $\mathrm{E}\left[p_i(\infty)\right]$, and hence we can write:

$$\begin{aligned} \mathrm{E}[p_i(\infty)](1-\lambda) &= (1-\lambda)s_i && (10)\\ \Rightarrow \mathrm{E}[p_i(\infty)] &= s_i\,. && (11)\end{aligned}$$

The result follows since (11) is valid for every component p_i of P. □

We now derive the explicit dependence of $\mathrm{E}\left[P(n+1)\right]$ on $\mathrm{E}\left[P(n)\right]$ and the consequences.

Theorem 2. *Let the parameter S of the multinomial distribution be estimated at time 'n' by $P(n)$ obtained by equations (2) and (3). Then, $E\left[P(n+1)\right] = \mathbf{M}^T E\left[P(n)\right]$, in which every off-diagonal*

[2] Observe that $\mathrm{E}[p_i]$ converges to a limit because the multiplying factor of the resultant linear difference equation is λ, which is both positive and strictly less than unity.

term of the stochastic matrix, $\mathbf{M}$*, has the* same *multiplicative factor,* $(1-\lambda)$*. Furthermore, the final solution of this* vector *difference equation is independent of* λ*.*

Remark: Theorem 2 states that the rules given by equations (2) and (3) are governed by a Markovian phenomenon in which the stochastic matrix has the same multiplicative factor and the solution to the final equation is independent of the parameter used in the algorithm. However, as we see below and in the empirical results, by selecting the parameter λ, the scheme leads to extremely good results. The convergence and eigenvalue properties of M follow.

Proof. From equation (8), we can write the conditional expected probability, $\mathrm{E}\left[p_1(n+1)|P\right]$ as follows:

$$\mathrm{E}\left[p_1(n+1)|P\right] = (1-\lambda)s_1 \sum_{j=1}^{r} p_j + \lambda p_1 \,. \tag{12}$$

Similarly, for all other conditional expectations of $p_i(n+1)$, we have:

$$\mathrm{E}\left[p_i(n+1)|P\right] = (1-\lambda)s_i \sum_{j=1}^{r} p_j + \lambda p_i \,. \tag{13}$$

Organizing the terms of (13) in a vectorial manner for all $i = 1, \ldots, r$, it can be seen that $\mathrm{E}\left[P(n+1)\right] = \mathbf{M}^T \mathrm{E}\left[P(n)\right]$, where the stochastic matrix $\mathbf{M}$, is:

$$\mathbf{M} = \begin{bmatrix} (1-\lambda)s_1+\lambda & (1-\lambda)s_2 & \cdots & (1-\lambda)s_r \\ (1-\lambda)s_1 & (1-\lambda)s_2+\lambda & \cdots & (1-\lambda)s_r \\ \vdots & \vdots & \ddots & \vdots \\ (1-\lambda)s_1 & (1-\lambda)s_2 & \cdots & (1-\lambda)s_r+\lambda \end{bmatrix}.$$

The limiting solution for $\mathrm{E}\left[P(n)\right]$ is obtained by solving the vectorial difference equation, and taking the limit as n is increased to infinity[3].

$$\mathrm{E}\left[P(\infty)\right] = \mathbf{M}^T \mathrm{E}\left[P(\infty)\right].$$

To solve the above, we observe that every element of the matrix $(\mathbf{I}-\mathbf{M})$ contains the term $(1-\lambda)$. By invoking this property and doing some straightforward algebraic manipulations, it turns out that:

$$\mathrm{E}\left[P(\infty)\right] = S,$$

and the theorem is proved. $\square$

[3]The solution exists since $\mathbf{M}$ has one eigenvalue which is unity, and all the other eigenvalues of $\mathbf{M}$ are strictly less than unity. We shall show this fact presently.

Theorem 3. *Let the parameter S of the multinomial distribution be estimated at time 'n' by $P(n)$ obtained by equations (2) and (3). Then, all the non-unity eigenvalues of* $\mathbf{M}$ *are exactly λ, and thus the rate of convergence of P is* fully *determined by λ.*

Proof. To analyze the rate of convergence of the vector difference equation, we first find the eigenvalues of $\mathbf{M}$, namely $\xi_1, \xi_2, \ldots, \xi_r$. Without going into the algebraic details, it can be shown that $\mathbf{M}$ can be expressed as follows:

$$\mathbf{M} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^{-1}\,, \tag{14}$$

where:

$$\mathbf{\Phi} = \left[\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \middle| \begin{bmatrix} -\frac{s_2}{s_1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \middle| \begin{bmatrix} -\frac{s_3}{s_1} \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \middle| \cdots \middle| \begin{bmatrix} -\frac{s_r}{s_1} \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right], \tag{15}$$

are the eigenvectors of $\mathbf{M}$, and $\mathbf{\Lambda} = diag(1, \lambda, \lambda, \ldots, \lambda)$, which contains the eigenvalues of $\mathbf{M}$. Thus, $\xi_1 = 1$, and $\xi_i = \lambda$ for $i = 2, \ldots, r$.

Consequently, the rate of convergence of the matrix determining the vector difference equation is *fully* determined by the second largest eigenvalue, which is λ, (since λ is an eigenvalue of multiplicity $r-1$). The result follows. □

A small value of λ leads to fast convergence and a large variance. On the contrary, a large value of λ leads to slow convergence and a small variance. Although the derived results are asymptotic, and thus, are valid only as $n \to \infty$, realistically, and for all practical purposes, the convergence takes place after a relatively small value of n. If λ is even as "small" as 0.9, after 50 iterations, the variation from the asymptotic value will be of the order of 10^{-50}, because λ also determines the rate of convergence, which again, occurs in a geometric manner. In other words, even if the environment switches its multinomial probability vector after 50 steps, the SLWE will be able to track this change. Earlier experimental results reported in [19] as well as our current experimental results demonstrate this fast convergence.

4 SLWE-based Solution to Adaptation to User's Interests Drift

In this section, we devise a *Learning Preferences Manager* which takes advantage of the SLWE updating scheme presented in Section 3, so as to accurately estimate the user's interest affinity in non-stationary environments. First, we will present our adapted model, as it pertains to the presentation of the user's profile. Thereafter, we introduce two profile update methodologies based on whether the data items attached to an attribute are disjunctive or conjunctive.

4.1 Profile Representation

An essential element of the Learning Preference Manager is the *Profile Representation*. For instance, a possible representation model for a user's interests can be in terms of the topic hierarchies [5, 13]. We adopt the Profile Representation Model advocated by Hossain and his co-authors in [6, 7]. It is important to remark that in these publications, the latter Profile Representation Model was mainly devised for representing the user's preferences in content media. Nevertheless, the model can be easily applied to encompass a wider set of interests. It should also be noted that the model reported in [6, 7] is similar to that of [27] in the sense that it is based on <feature, weight> pairs, except that in [6, 7], the authors have invoked a normalized score for the data items. We shall first briefly present the Profile Representation Model reported in [6, 7].

The user's affinity of interests in a service type, such as movies, or restaurants, is represented by a set of *attributes*. For example, for a repository of services of type movie, the set of possible attributes could be {movie genre, director name, etc.}. An attribute, in turn, possesses a set of *data items*. For example, if the movie attribute "genre" has two data items, namely "action" and "comedy", a vector associated with the attribute (comedy affininity=0.7, action affininity=0.3) reflects that the user likes comedy movies more than action movies, with a relative weighting of 0.7 to 0.3. The update of the weights of the data items for a particular attribute is done in an incremental manner.

4.2 Profile Updating method

In the quest to learn the user's dynamic profile, the *Learning Preferences Manager* is guided by so-called *Relevance Feedback* (RF) [15]. In this paper, we rely on the *Service Usage History* (analogous to the history maintained by the authors of [6, 7]) as the main source of the RF. In fact, a common approach towards constructing a user's profile is through non-intrusively monitoring the history of the usage of his services. A *Service Usage History* (also known as the *Interaction History*), contains the history of the services used by the user over time. For example, when the user has used a certain service at a certain time instant, the *Learning Preferences Manager* refines and revises the user's profile based on the current instance of the usage history, which, in turn, is automatically and unobtrusively observed in the background. To obtain an index to measure this, the sum of the scores of a data item for a given attribute is made to be equal to unity. To now quantify this, we have opted to use the SLWE [19] explained in the previous section, so as to update the score of the data item based on the usage history. Whenever a user selects a service, the metadata describing the service is used to update the score of data item. Thus, for example, if a user currently views a "action" movie, the scheme would increase the weight associated with the data item "action".

Apart from the updating mechanism, our strategy can also be seen to be philosophically related to the approach presented in [6, 7] in which the authors utilized the history to update the affinity of the user's interests. We believe that this will facilitate the ease of the retrieval of personalized information, and help alleviate the user's cognitive load, i.e., that which is needed to locate relevant information.

At this juncture, we distinguish two classes of data items that, in turn, require two different forms of update mechanisms. In fact, the data items related to a given attribute could be either semantically **disjunctive** or semantically **conjunctive**. We illustrate what we mean by the latter concepts by alluding to two simple examples.

4.2.1 Profile Update for Disjunctive Data Items

Data items of a particular attribute are said to be *disjunctive* if every service usage history can only be instantiated with the exclusive realization of one of the data items at a time. To illustrate the idea in simpler terms, consider the example of learning a user's preferences when it concerns a type of services such as restaurants. In this case, we can consider the attribute genre of the restaurant, with the data items being, for example, Chinese, Italian, Indian, French etc.

The latter data items correspond to a possible semantic taxonomy of restaurants according to their genre. Whenever a user interacts with a service of type restaurant, a *Service Usage History* instance is submitted to the *Learning Preference Manger* where the restaurant is described by a single exclusive attribute, such as Italian. Consequently, the weight of the latter data item can be incremented while the weight of the remaining data items of the same attribute can be decremented. Therefore, a multinomial SLWE is a viable option for estimating the evolving weights of the data items. Proceeding to make inferences from these weak estimators becomes then a suitable choice for managing the time-varying preferences.

It is crucial for the reader to observe that the SU approach presented in [6, 7] deals only with this specific case, i.e., of disjunctive data items.

4.2.2 Profile Update for Conjunctive Data Items

Data items of a particular attribute are said to be *conjunctive* whenever every service usage history can be instantiated with one *or more* data items at a time. To illustrate this, consider the example of the service usage history corresponding to the services for movies. The attribute movie genre is associated with the data item set $S_{genre} = \{action, romantic, comedy, horror\}$. The latter data items are conjunctive (not disjunctive) in the sense that a movie's genre can be described with more than a single data item at a time. For instance, a movie genre could be "romantic" and "action packed" at the same time. Suppose that the user watches a movie that belongs to the genres *action* and *romance* at a given time instant 'n'. In this case, the weights of both the data items *action* and *romance* can be increased at time '$n+1$'.

In this case, a multinomial SLWE will not be able to update the different weights of the data items because it is not designed to increase the weights of more than a component at a time. Thus, a different methodology for updating the weights of the data items is needed, where more than a single data item's weight can be incremented at a time. To solve the problem, we propose to attach a binomial SLWE to each data item instead of having a multinonmial probability vector for each attribute, as in the case of disjunctive data items. In other words, a binomial probability vector will be attached to each of data items in S_{genre}.

For the sake of clarity, we consider the above-mentioned example and describe the update at the subsequent instant '$n+1$' of each binomial probability vector as:

$$p_{action}(n+1) \leftarrow 1 - \lambda(1 - p_{action}(n)) \tag{16}$$

$$p_{romantic}(n+1) \leftarrow 1 - \lambda(1 - p_{romantic}(n)) \tag{17}$$

$$p_{comedy}(n+1) \leftarrow \lambda p_{comedy}(n) \tag{18}$$

$$p_{horror}(n+1) \leftarrow \lambda p_{horror}(n) \tag{19}$$

Once these binomial-based computations have been achieved, we then resort to an additional computation in order to normalize the weights of each data items. The normalization is, quite simply, given by: For $k \in S_{genre} = \{action, romantic, comedy, horror\}$

$$W_k(n+1) = \frac{p_k(n+1)}{\sum_{j \in S} p_j(n+1)} \tag{20}$$

Consequently W_k tracks, with a SLWE-philosophy, the ratio of the number of times the particular data items ($k \in S_{genre} = \{action, romantic, comedy, horror\}$) of the particular attribute (movie's genre) appears in the service usage within a given number of usage records, to the total number of occurrences of the data items of S_{genre}.

In order to model this in a "tangible" (or realistic) way, we suppose that the occurrence of each data item in the usage history is controlled by a binomial distribution. We further suppose, that the occurrence of the data items is independent of each other. Let s_k be the binomial parameter that describes the occurrence of data item k in the usage history, where $k \in S_{genre} = \{action, romantic, comedy, horror\}$. With these assumptions, based on the results of the previous subsection, we easily derive the asymptotic weight:

$$\mathrm{E}\left[W_k(\infty)\right] = \frac{s_k}{\sum_{j \in S} s_j}. \tag{21}$$

It is worth noting that whenever the data items corresponding to a given attribute are disjunctive, it is computationally more efficient (although only marginally) to employ a multinomial SLWE – instead of a set of binomial SLWEs.

4.2.3 Modelling changes in the Interests

We suppose that at every time instance 'n', the *Learning Preferences Manager* is fed by a service usage instance. We further assume that the distribution of the user's interests, relative to a given attribute, undergoes an abrupt change at a random time instance with an unknown probability p. In the case of disjunctive data items, we assume that the parameters of the multinomial distribution change to yield a new distribution. Further, in the case of conjunctive data items, the binomial distribution attached to every data item switches to a possibly new value.

5 Experimental results

To verify our computational model and our proposed solution, we have performed extensive simulations. However, in the interest of space and brevity, we report here only a subset of these results. We emphasize though that these results are both representative and typical. The obtained experimental results are conclusive, and demonstrate that our SLWE-based update schemes, when applied to tracking users' interests, outperforms the GF approach, the sliding window, and the SU.

In order to model the changes in the interests' distribution, we assume that at any given time instant, the distribution of the user's preferences changes with probability 0.02. This implies that on *average*, a change occurs every 50 time instants. The reader should observe that our experimental results are based on synthetic data due to fact that it is difficult (if not *impossible*) to obtain real-life data that describe user's preferences. Indeed, no existing organization will disclose or share such data because of the implied privacy and security considerations. However, we believe that the model which we have used to "artificially" indicate the changes in the user's interest distributions is strong enough to mimic real-life settings.

5.1 Disjunctive Data Items

To study the case of disjunctive data items, we assume that we are dealing with estimating the evolving user's interests' weights of data items of this type, namely, those which are associated with a given attribute. In the interest of completeness, we will present separate experimental results for the binomial and the multinomial cases.

Figure No.	Error rate: SLWE	Error rate: GF	Error rate: MLE	Error rate: SU
Figure 1	0.0262	0.0864	0.1005	0.2894
Figure 2	0.0347	0.1418	0.1654	0.2713
Figure 3	0.0737	0.1364	0.1514	0.2813
Figure 4	0.0330	0.0779	0.0901	0.1561

Table 1: The effects of varying the window size and the updating parameter on the error of p_1 for the various schemes investigated for disjunctive data items.

5.1.1 Binomial Distribution

In this scenario, we assume that we are dealing with an attribute which possesses two data items. The problem of estimating the user's interests in this particular case now reduces to that of estimating the parameters of the corresponding binomial random variables. As mentioned above, numerous experiments were performed, although in the interest of brevity, we report here only the results from four of these. Moreover, in order to render the comparison meaningful, we have simultaneously followed the GF and SU computation, and in each case we have utilized identical data streams as in the case of the SLWE. Further, in each case, the estimation algorithms were presented with random occurrences of the variables for $n = 400$ time instances. In the case of the SLWE, the true underlying value of s_1 was randomly assigned for the first step, and modified at random instants (determined with a switching probability of 0.02) using values drawn from a random variable which was uniformly distributed in $[0, 1]$.

In order to demonstrate the superiority of the SLWE over the GF that uses a sliding window, and the SU, we report the respective averages on an ensemble of experiments. The same experiment was repeated $1,000$ times with distinct random sequences, and the ensemble average at every time step was recorded. Clearly, by doing this, the variations of the estimates would be much smoother. In order to perform a fair evaluation, we adopted the same comparison approach as in [19], where the value of λ for the SLWE and the size of the window were randomly generated from uniform distributions in $[0.55, 0.95]$ and $[20, 80]$ respectively. The value of the "gradual forgetting" factor, k, for the GF was chosen to be 0.4 as suggested in [10].

The plots of the estimated probability p_1 for the SLWE, the GF and SU for four cases are shown in Figures 1, 2, 3 and 4, where the values of λ are 0.763, 0.811, 0.811 and 0.563, and the sizes of the windows are 29, 46, 57 and 68 respectively.

Observe that both the GF and SU follow s_1 *quite exactly* prior to the first distribution change, but they are thereafter severely handicapped in tracking the variations. The weakness of the GF is accentuated in the cases in which the size of the window is larger, e.g. 68. In contrast, the SLWE adjusts to the changes much more rapidly, as expected, in a geometric manner. In Table 1, we report the error rate associated with the four experiments – i.e., those plotted in Figures 1, 2, 3 and 4. We also include the error rate for the MLE with the same sliding window sizes as used for the GF experiments. Clearly, we observe that the SLWE yields a lower error rate than the other three approaches, namely the GF, the SU and the MLE which uses a sliding window. An additional remark confirming the results of [10] is the following: The GF augmented with a sliding window exhibits a lower error rate than the MLE with a sliding window. Thus, this confirms that the GF presents an enhancement to the basic sliding window approach.

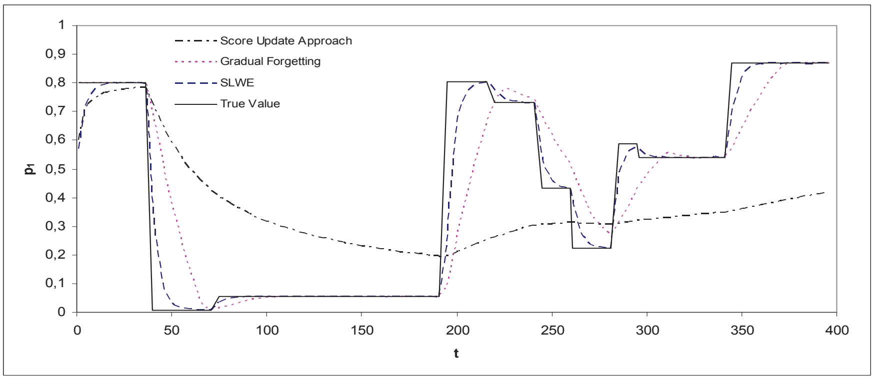


Figure 1: Plots of the expected values of $p_1(n)$, at time 'n' for disjunctive data items, which were estimated by using the SLWE, the GF and the SU, in which the corresponding parameters $\lambda = 0.736$ and the window size is 29.

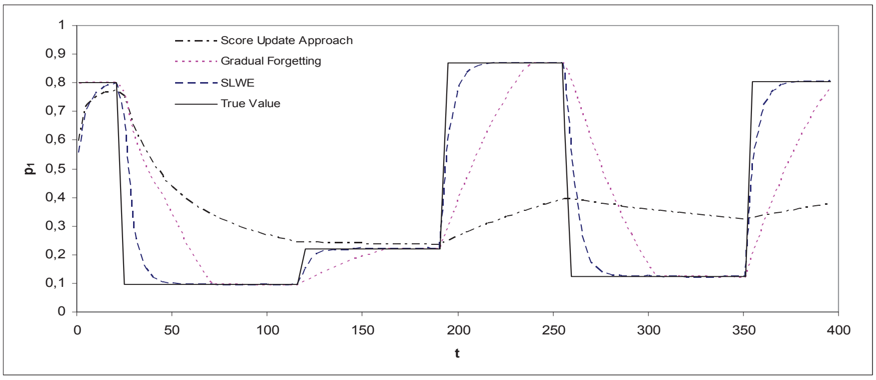


Figure 2: Plots of the expected values of $p_1(n)$, at time 'n' for disjunctive data items, which were estimated by using the SLWE, the GF and the SU, in which the corresponding parameters $\lambda = 0.811$ and the window size is 46.

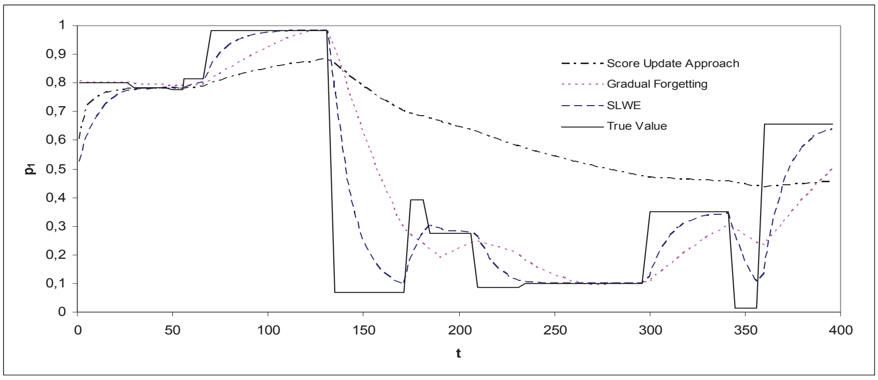


Figure 3: Plots of the expected values of $p_1(n)$, at time 'n' for disjunctive data items, which were estimated by using the SLWE, the GF and the SU, in which the corresponding parameters $\lambda = 0.912$ and the window size is 57.

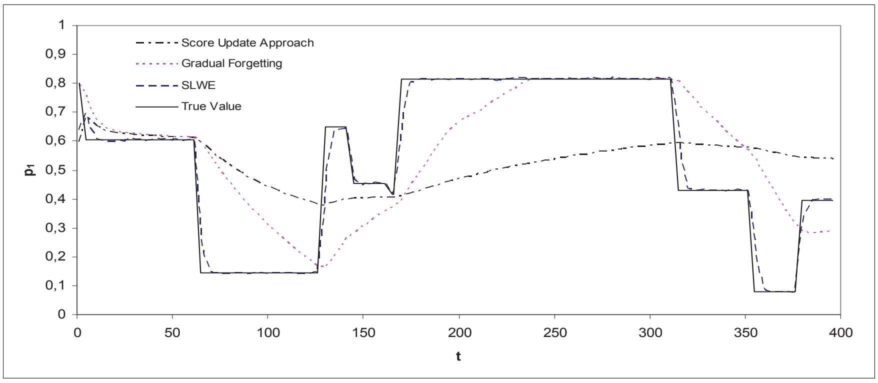


Figure 4: Plots of the expected values of $p_1(n)$, at time 'n' for disjunctive data items, which were estimated by using the SLWE, the GF and the SU, in which the corresponding parameters $\lambda = 0.563$ and the window size is 68.

5.1.2 Multinomial Distribution

We have also performed simulations for the case of disjunctive data items, where the user's interests follow a multinomial distribution, and where the parameters were estimated by following the SLWE, the GF and the SU. We considered a multinomial random variable, X, which can take any of four different values, namely '1', '2', '3' or '4', whose characterizing parameters changed (randomly) at random time instants. As in the binomial case, we ran the estimators for 400 steps, repeated this $1,000$ times, and then computed the corresponding ensemble averages. For each experiment, we computed $||P - S||$, the *Euclidean distance* between P and S, which we reckoned as a measure of how good our estimate, P, was of S. The plots of the latter distance obtained from the SLWE, the GF and the SU are depicted in Figures 5, 6, 7 and 8, where the values of λ were 0.908, 0.903, 0.952 and 0.948, and the sizes of the windows were 35, 44, 63 and 76 respectively. The values for λ and the window size were obtained randomly from a uniform distribution in $[0.9, 0.99]$ and $[20, 80]$ respectively.

Figure No.	Error rate: SLWE	Error rate: GF	Error rate: SW	Error rate: SU
Figure 5	0.0612	0.0724	0.0836	0.4606
Figure 6	0.0665	0.1006	0.1152	0.4037
Figure 7	0.1601	0.1893	0.2074	0.4175
Figure 8	0.0507	0.0567	0.0672	0.4165

Table 2: The effects of varying the window size and the updating parameter on the error rates for the various schemes investigated for disjunctive data items.

From these figures, we observe that the GF, the SU and the SLWE converge to zero relatively quickly prior to the first instant when the distribution changes. However, this behavior is not present for subsequent (successive) distribution "switches". Rather, we notice that the GF is capable of tracking the changes of the parameters when the size of the window is small, or at least smaller than the intervals of constant probabilities. It is, however, not able to track the changes properly when the window size is relatively large. Since neither the magnitude nor the instants of the changes is known *a priori*, this scenario demonstrates the weakness of the GF, and its dependence on the knowledge of the input parameters. Again, such observations are typical.

In Table 2, we report the error rates associated with the experiments plotted in Figures 5, 6, 7 and 8. We also include the error rates for the MLE augmented with a sliding window in Table 2. Clearly, one observes that the SLWE exhibits a lower error rate than the GF, the SU and the MLE.

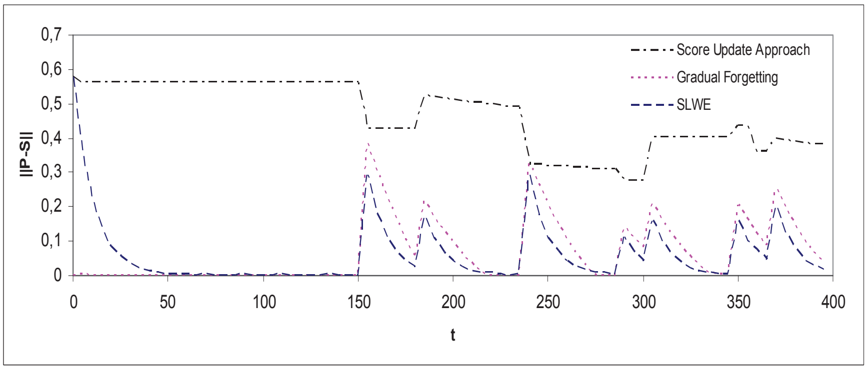


Figure 5: Plot of the Euclidean norm $||P - S||$ (the Euclidean distance between P and S) for disjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.908$ and w = 35.

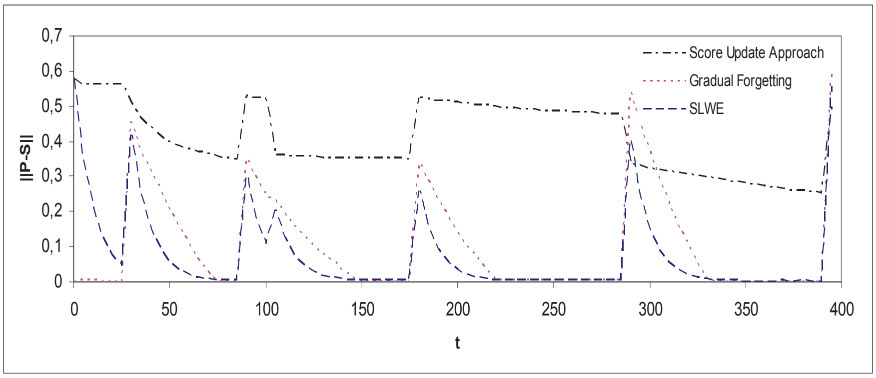


Figure 6: Plot of the Euclidean norm $||P - S||$ (the Euclidean distance between P and S) for disjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.903$ and w = 44.

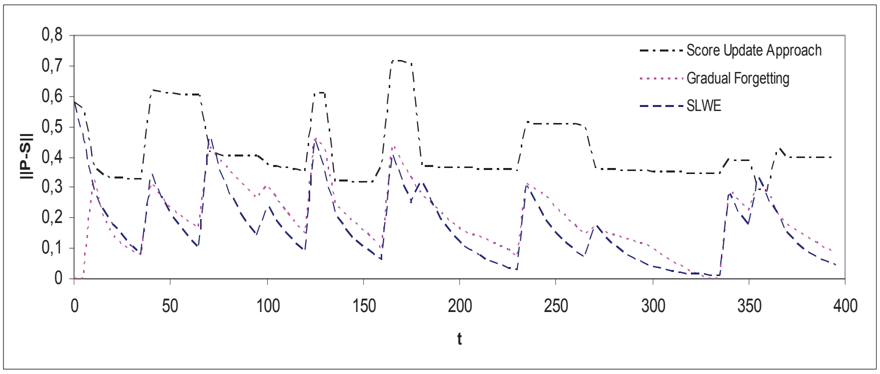


Figure 7: Plot of the Euclidean norm $||P - S||$ (the Euclidean distance between P and S) for disjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.952$ and w = 63.

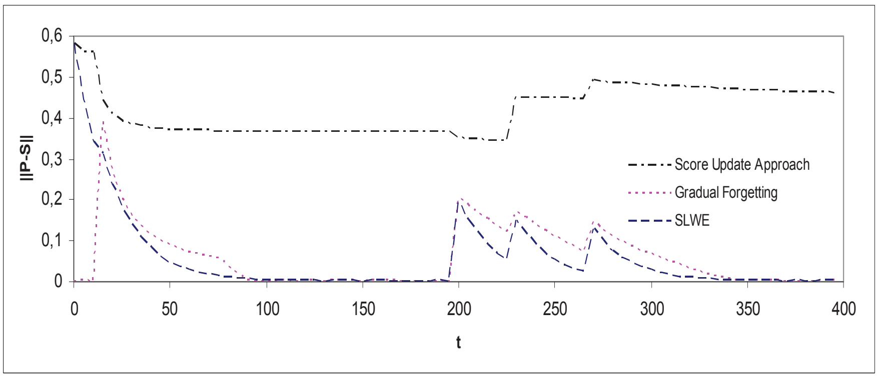


Figure 8: Plot of the Euclidean norm $||P - S||$ (the Euclidean distance between P and S) for disjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.948$ and w = 76.

5.2 Conjunctive data items

In this subsection, we present simulations results related to conjunctive data items. We assume that for each data item, the probability of appearing in a given record history is binomial. In the experiments, we considered 4 data items and assumed that at random time instants, the binomial

Figure	Error rate: SLWE	Error rate: GF	Error rate: MLE	Error rate: SU
Figure 9	0.0181	0.1030	0.1151	0.1556
Figure 10	0.0071	0.0533	0.0614	0.1609
Figure 11	0.0371	0.1412	0.1524	0.1864
Figure 12	0.0259	0.1158	0.1281	0.1633

Table 3: The effects of varying the window size and the updating parameter on the error rates for the various schemes investigated for conjunctive data items.

distribution of each of the 4 data items changed, and that, randomly. We are interested in estimating the weights of the data items that reflect the respective ratios of their appearances in the usage history. As alluded to previously, W_i tracks the ratio of appearance of data item i in the usage history[4].

As in the previous experiments, we computed $||P - S||$, the *Euclidean distance* between P and S, which was used as a measure of how good our estimate, P, was of S. The plots of the latter distance obtained from the SLWE, the GF and the SU are depicted in Figures 9, 10, 11 and 12 where the values of λ were 0.579, 0.627, 0.771 and 0.809 respectively, and the sizes of the windows were 33, 39, 55 and 70 respectively. Here the value of λ for the SLWE and the size of the window were randomly generated from uniform distributions in $[0.55, 0.95]$ and $[20, 80]$ respectively.

In Table 3, we report the error rate associated with the experiments whose results were plotted in Figures 9, 10, 11 and 12. We have also included the error rates for the MLE augmented with the sliding window in Table 3. In all brevity we can state that the results are conclusive: The SLWE exhibits a lower error rate than the GF, the SU and the MLE in all the different settings.

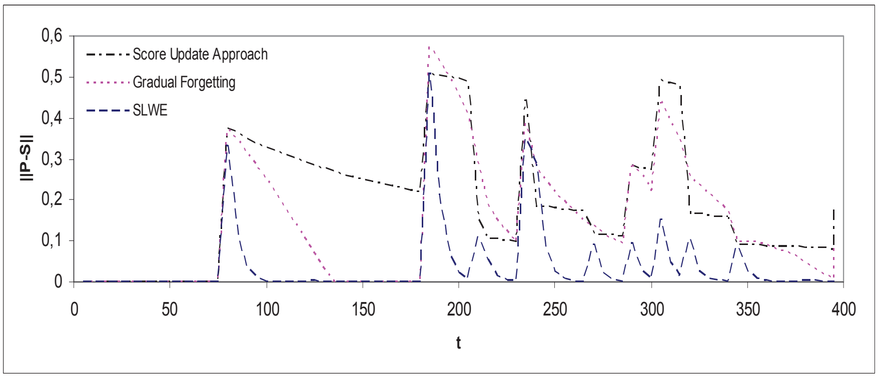


Figure 9: Plot of the Euclidean norm $||P - S||$ (the Euclidean distance between P and S) for conjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.579$ and w = 33.

[4]Since we estimate the quantity W in terms of P, we plot the variation of $P(n)$ instead of the variation of $W(n)$ in each of the graphs below.

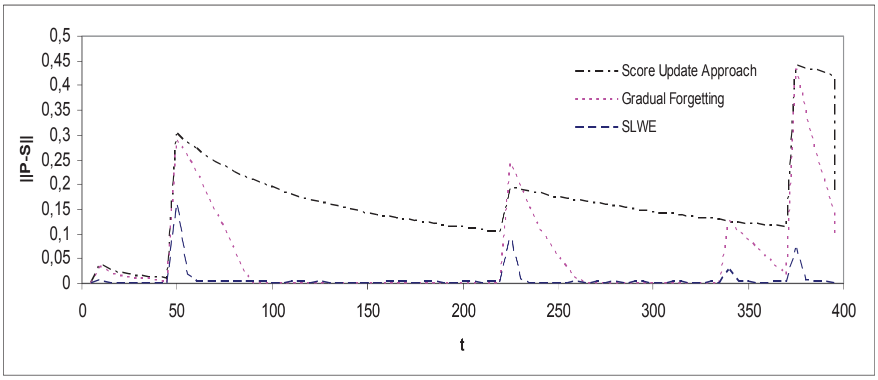


Figure 10: Plot of the Euclidean norm $||P - S||$ (the Euclidean distance between P and S) for conjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.627$ and w = 39.

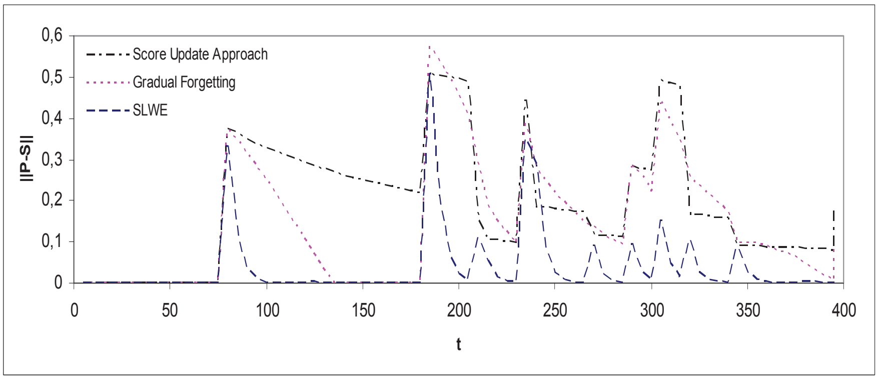


Figure 11: Plot of the Euclidean norm $||P - S||$ (the Euclidean distance between P and S) for conjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.771$ and w = 55 respectively.

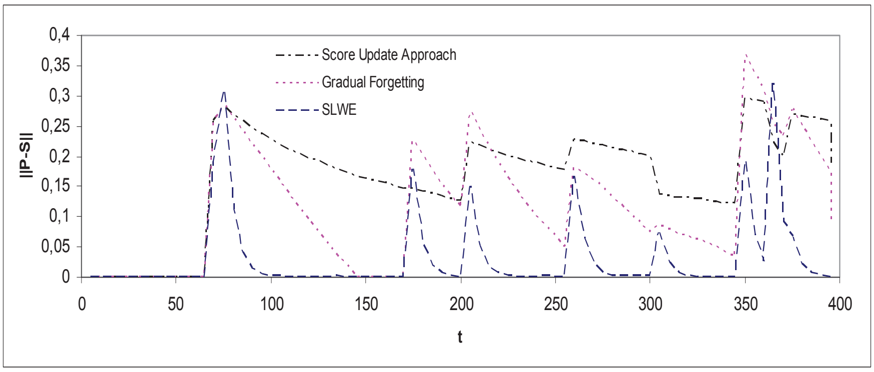


Figure 12: Plot of the Euclidean norm $||P - S||$ (the Euclidean distance between P and S) for conjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.809$ and w = 70.

6 Conclusions

In this paper we have studied the complex problem of having a social network adapt with the preferences of its users. The premise for this study is that the diversity of a social network cannot be accurately modeled by a *static* set of preferences. Thus, the problem of "estimating" the preferences of its users is becoming increasingly essential for personalized applications which range from service recommender systems to the targeted advertising of services. This being the case, one observes that a traditional estimation strategy, (for estimating the user's interests) which works when the underlying target distribution is stationary, is unsuitable for dynamic non-stationary environments. We have therefore argued that resorting to strong estimators that converge with probability 1 is inefficient since they rely on the assumption that the distribution of the user's preferences is stationary. Consequently, we have proposed the use of a family of stochastic-learning based *weak* estimators for learning and tracking the user's time varying interests. To solve the problem, we have approached the problem by modeling the user's interests using the concept of data items. Thereafter, we have devised two cohesive models for updating the score of the data items in the user's profile depending on whether the data items associated with a given attributed are disjunctive or conjunctive. Simulations results based on synthetic data demonstrates the superiority of our proposed weak estimator-based update methods when compared to the state-of-the-art methods involving "Gradual Forgetting", the Ambient Media Score Update method (SU), and the Maximum Likelihood Estimation (MLE) scheme augmented with a sliding window.

The problem of utilizing of the learned profiles in order to perform efficient matchmaking between available services and the user's profile is a potential avenue for future research, for which we do, indeed, have some very promising initial results.

References

[1] A. Aghasaryan, S. Betgé-Brezetz, C. Senot, and Y. Toms, "A profiling engine for converged service delivery platforms," *Bell Lab. Tech. J.*, vol. 13, no. 2, pp. 93–103, 2008.

[2] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes: theory and application.* Prentice-Hall, Inc., 1993.

[3] M. Brunato and R. Battiti, "Pilgrim: A location broker and mobility-aware recommendation system," in *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications.* Washington, DC, USA: IEEE Computer Society, 2003, p. 265.

[4] D. Godoy and A. Amandi, "User profiling in personal information agents: a survey," *Knowl. Eng. Rev.*, vol. 20, pp. 329–361, 2005.

[5] D. Godoy, S. Schiaffino, and A. Amandi, "Interface agents personalizing web-based tasks," *Cognitive Systems Research, Special Issue on Intelligent Agents and Data Mining for Cognitive Systems*, vol. 5, no. 3, pp. 207–222, 2004.

[6] M. A. Hossain, P. K. Atrey, and A. El Saddik, "Gain-based selection of ambient media services in pervasive environments," *Mob. Netw. Appl.*, vol. 13, no. 6, pp. 599–613, 2008.

[7] M. A. Hossain, J. Parra, P. K. Atrey, and A. El Saddik, "A framework for human-centered provisioning of ambient media services," *Multimedia Tools and Applications*, vol. 44, pp. 407–431, 2009.

[8] I. Koychev, "Gradual forgetting for adaptation to concept drift," in *Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning*, 2000, pp. 101–106.

[9] I. Koychev and R. Lothian, "Tracking drifting concepts by time window optimisation," in *Research and Development in Intelligent Systems XXII*, M. Bramer, F. Coenen, and T. Allen, Eds. Springer London, 2006, pp. 46–59.

[10] I. Koychev and I. Schwab, "Adaptation to drifting user's interests," in *Proceedings of ECML2000 Workshop: Machine Learning in New Information Age*, 2000, pp. 39–46.

[11] S. Kurkovsky and K. Harihar, "Using ubiquitous computing in interactive mobile marketing," *Personal Ubiquitous Comput.*, vol. 10, no. 4, pp. 227–240, 2006.

[12] M. A. Maloof and R. S. Michalski, "Selecting examples for partial memory learning," *Machine Learning*, vol. 41, pp. 27–52, 2000.

[13] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, "Ontological user profiling in recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 54–88, 2004.

[14] T. M. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, "Experience with a learning personal assistant," *Commun. ACM*, vol. 37, no. 7, pp. 80–91, 1994.

[15] M. Montaner, B. López, and J. L. de la Rosa, "A taxonomy of recommender agents on the internet," *Artificial Intelligence Review*, vol. 19, pp. 285–330, 2003.

[16] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction.* Prentice Hall, 1989.

[17] Y. Naudet, A. Aghasaryanb, S. Mignon, Y. Toms, and C. Senot, "Ontology-based profiling and recommendations for mobile tv," in *Semantics in Adaptive and Personalized Services*, ser. Studies in Computational Intelligence, M. Wallace, I. Anagnostopoulos, P. Mylonas, and M. Bielikova, Eds. Springer, Heidelberg, 2010, vol. 279, pp. 23–48.

[18] B. J. Oommen and S. Misra, "Fault-tolerant routing in adversarial mobile ad hoc networks: an efficient route estimation scheme for non-stationary environments," *Telecommunication Systems*, vol. 44, pp. 159–169, 2010, 10.1007/s11235-009-9215-4.

[19] B. J. Oommen and L. Rueda, "Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments," *Pattern Recogn.*, vol. 39, no. 3, pp. 328–341, 2006.

[20] L. Rueda and B. J. Oommen, "Stochastic automata-based estimators for adaptively compressing files with nonstationary distributions," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 5, pp. 1196 –1200, 2006.

[21] A. N. Shiryayev, *Optimal Stopping Rules*. Springer, 1978.

[22] A. Stensby, B. J. Oommen, and O.-C. Granmo, "Language detection and tracking in multilingual documents using weak estimators," in *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*, ser. SSPR&SPR'10, 2010, pp. 600–609.

[23] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers, 2004.

[24] G. Widmer, "Tracking context changes through meta-learning," *Mach. Learn.*, vol. 27, no. 3, pp. 259–286, 1997.

[25] W.-S. Yang, H.-C. Cheng, and J.-B. Dia, "A location-aware recommender system for mobile shopping environments," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 437–445, 2008.

[26] A. Yazidi, O.-C. Granmo, B. J. Oommen, M. Gerdes, and F. Reichert, "A user-centric approach for personalized service provisioning in pervasive environments," *Submitted for Publication.*

[27] Z. Yu, X. Zhou, D. Zhang, C.-Y. Chin, X. Wang, and J. Men, "Supporting context-aware media recommendations for smart phones," *IEEE Pervasive Computing*, vol. 5, pp. 68–75, July 2006.

Appendix E

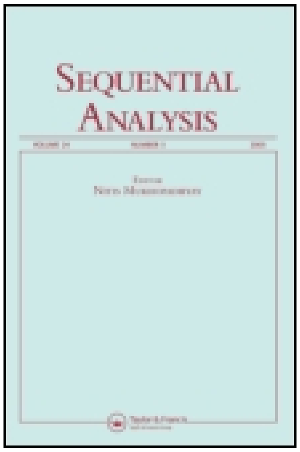
Paper V

Title: On the Analysis of a Random Interleaving Walk-Jump Process

Authors: **Anis Yazidi**[1], Ole-Christoffer Granmo[2] and B. John Oommen[1,2]

Affiliations: [1]University of Agder, Faculty of Engineering and Science, P. O. Box 509, NO-4898 Grimstad, Norway
[2]School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6

Journal: *Sequential Analysis*, Vol. 31, 2012, pp. 190-218.



Sequential Analysis: Design Methods and Applications
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/lsqa20

On the Analysis of a Random Interleaving Walk-Jump Process with Applications to Testing

Anis Yazidi [a], Ole-Christoffer Granmo [a] & B. John Oommen [a b]

[a] Department of ICT, University of Agder, Grimstad, Norway

[b] School of Computer Science, Carleton University, Ottawa, Canada

Available online: 07 Nov 2011

Sequential Analysis, 30: 457–478, 2011

ISSN: 0747-4946 print/1532-4176 online
DOI: 10.1080/07474946.2011.619104

On the Analysis of a Random Interleaving Walk–Jump Process with Applications to Testing

Anis Yazidi[1], Ole-Christoffer Granmo[1], and B. John Oommen[1,2]
[1]Department of ICT, University of Agder, Grimstad, Norway
[2]School of Computer Science, Carleton University, Ottawa, Canada

Abstract: Although random walks (RWs) with single-step transitions have been extensively studied for almost a century as seen in Feller (1968), problems involving the analysis of RWs that contain interleaving random steps and random "jumps" are intrinsically hard. In this article, we consider the analysis of one such fascinating RW, where every step is paired with its counterpart random jump. In addition to this RW being conceptually interesting, it has applications in testing of entities (components or personnel), where the entity is never allowed to make more than a prespecified number of *consecutive* failures. The article contains the analysis of the chain, some fascinating limiting properties, and simulations that justify the analytic steady-state results. Some simulation results for the chain's transient behavior are also included. Finally, a comparative testing against a hidden Markov model shows that within the testing framework, the results of our model are competitive, if not superior. As far as we know, the entire field of RWs with interleaving steps and jumps is novel, and we believe that this is a pioneering article in this field.

Keywords: Ergodic random processes; Random processes; Random walks with jumps.

Subject Classifications: 60J10; 60J20; 68M15.

1. INTRODUCTION

The theory of random walks (RWs) and their applications have gained increasing research interest since the start of the last century. From the recorded literature, one perceives that the pioneering treatment of a one-dimensional RW was due to Karl Pearson in 1905. In this context, we note that a RW is usually defined as a trajectory involving a series of successive random steps, which are, quite naturally, modeled using Markov chains (MCs). MCs are probabilistic structures that possess

Received March 7, 2011, Revised August 17, 2011, and August 23, 2011, Accepted August 27, 2011
Recommended by R. Keener
Address correspondence to B. John Oommen, School of Computer Science, Carleton University, Ottawa, OT, K1S 5B6, Canada; E-mail: oommen@scs.carleton.ca

the so-called Markov property—which, informally speaking, implies that the next "state" of the walk depends on the current state and not on the entire past states (or history). The latter property is also referred to as the *lack of memory* property, which imparts to the structure practical consequential implications because it permits the modeler to predict how the chain will behave in the immediate and distant future and to thus quantify its behavior.

RWs have been utilized in a myriad of applications stemming from areas as diverse as biology, computer science, economics, and physics. For instance, concrete examples of these applications in biology are the epidemic models described in Berg (1993), the Wright-Fisher model (Wade and Goodnight, 1998), and the Moran model in Nowak (1996), among others. RWs arise in the modeling and analysis of queuing systems as seen in Gross and Harris (1998), ruin problems as seen in Takacs (1969), risk theory as shown in Paulsen (1983), and sequential analysis and learning theory as demonstrated in Bower (1994). In addition to the above-mentioned *classical* application of RWs, recent applications include mobility models in mobile networks given in Camp et al. (2002), collaborative recommendation explained in Fouss et al. (2007), web search algorithms highlighted in Altman and Tennenholtz (2005), and reliability theory for both software and hardware components as seen in Bishop and Pullen (1991, please see pp. 83–111).

RWs can be broadly classified in terms of their Markovian representations. Generally speaking, RWs are either ergodic or possess absorbing barriers. In the simplest case, the induced MC is ergodic, implying that, sooner or later, each state will be visited (with probability 1), independent of the initial state. In such MCs, the limiting distribution of being in any state is independent of the corresponding initial distribution. This feature is desirable when the directives dictating the steps of the chain are a consequence of interacting with a nonstationary environment, allowing the walker to not get trapped into choosing any single state. Thus, before one starts the analysis of an MC, it is imperative that one understands the nature of the chain; that is, whether it is ergodic, which will determine whether or not it possesses a stationary distribution.

A RW can also possess absorbing barriers. In this case, the associated MC has a set of transient states that it will sooner or later never visit again and, thus, it cannot be ergodic. When the walker reaches an absorbing barrier, it is "trapped" and is destined to remaind there forever. RWs with two absorbing barriers have also been applied to analyze problems akin to the two-choice bandit problems in Oommen (1986) and the gambler's ruin problem in Takacs (1969), and their generalizations to chains with multiple absorbing barriers have their analogous extensions.

Although RWs are traditionally considered to be unidimensional (i.e., on the line), multidimensional RWs operate on the plane or in a higher dimensional space.

The most popularly studied RWs are those with single-step transitions. The properties of such RWs have been extensively investigated in the literature. A classical example of a RW of this type is the ruin problem in Takacs (1969). In this case, a gambler starts with a fortune of size s. The gambler decides to play either until he is ruined (i.e., his fortune decreases to 0) or until he has reached a fortune of M. At each step, the gambler has a probability (chance), p, of incrementing his fortune by a unit and a chance $q = 1 - p$ of losing a unit. The actual capital possessed by the gambler is represented by a RW on the line of integers from 0 to M, with the states 0 and M serving as the respective absorbing barriers. Of course, the game changes drastically to be ergodic if a player is freely given a unit of wealth if he is bankrupt (his fortune is 0) or forfeits a unit if he attains the maximum wealth of M, in which case the respective boundaries are said to be *reflecting*.

It is pertinent to mention that the available results pertaining to RWs in which the chain can move from any state N to *non-neighboring* states $N+k$ or $N-k$ are scant. Indeed, the analysis of the corresponding MCs is almost impossible in the most general case, *except when certain conditions like time reversibility can be invoked*. Finally, such chains fall completely outside the scope of the so-called family of 'birth-and-death' processes, because even in these cases, the number of individuals does not drastically fall or increase in a single time unit (except in the case when one models catastrophes—but even here, MCs of this type are unreported).

Although RWs with single-step transitions, such as the ruin problem, have been extensively studied for almost a century, as one can observe from Feller (1968), problems involving the analysis of RWs containing *interleaving* random steps and random jumps are intrinsically hard. In this article, we consider the analysis of one such fascinating RW, where every step is paired with its counterpart random jump. In addition to this RW being conceptually interesting, it has applications in the testing of entities (components or personnel), where the entity is never allowed to make more than a prespecified number of *consecutive* failures.

To motivate the problem, consider the scenario when we are given the task of testing an error-prone component. At every time step, the component is subject to failure, where the event of failure occurs with a certain probability, q. The corresponding probability of the component not failing is p, where $p = 1 - q$. The latter quantity can also be perceived to be the probability of the component *recovering* from a failure; that is, if it indeed had failed at the previous time instant. Further, like all real-life entities, the component can operate under two modes, either in the *well-functioning* mode, or in the *malfunctioning* mode. At a given time step, we aim to determine whether the component is behaving well, that is, in the *well-functioning* mode, or whether it is in the *malfunctioning* mode, which are the two states of nature. It is not unreasonable to assume that both of these hypotheses are mutually exclusive, implying that only one of these describes the state of the component at a given time step, thus excluding the alternative. Let us now consider a possible strategy for determining the appropriate hypothesis for the state of nature.

Suppose that the current maintained hypothesis conjectures that the component is in a *malfunctioning* mode. This hypothesis is undermined and systematically replaced by the hypothesis that the component is in its *well-functioning* mode if it succeeds to realize a certain number N_1 of successive recoveries (or successes). In the same vein, suppose that the current hypothesis conjectures that the component is in its *well-functioning* mode. This hypothesis, on the other hand, is invalidated and systematically replaced by the hypothesis that the component is in its *malfunctioning* mode if the component makes a certain number $N_2 + 1$ of successive failures. We shall show that such a hypothesis testing paradigm is most appropriately modeled by an RW in which the random steps and jumps are interleaving. To the best of our knowledge, such a modeling paradigm is novel. Further, the analysis of such a chain is unreported in the literature.

An analogous way to illustrate the application of the problem is by considering the scenario when the knowledge of personnel in an organization is tested, for example, *via* a questionnaire. In order to test the knowledge of the person concerned, she or he is continuously confronted with questions at discretized time steps. The person can either answer the current question correctly, with a probability p, or erroneously, with a probability $q = 1 - p$. At any given time step, we conjecture the hypothesis that the person concerned is in a *learning phase*, or, alternatively, that she or he has crossed the learning hurdle and is in a *knowledgeable phase*. Suppose that the current hypothesis conjectures that the person is in the *learning phase*. The latter hypothesis is invalidated and replaced by the hypothesis that she or he is in a *knowledgeable phase* if she or he realizes a certain number, say N_1, of successive correct answers in the questionnaire. In the same vein, suppose that the current hypothesis conjectures that the person is in the *knowledgeable phase*. This hypothesis is, in turn, undermined and systematically replaced by the hypothesis conjecturing that she or he is in the *learning phase* if she or he makes a certain number, say $N_2 + 1$, of successive wrong answers. At this juncture, we point out that the generalization for a researcher to use the same strategy to know when an artificial intelligence (AI) scheme should switch from 'exploration' to 'exploitation' is an extremely interesting avenue for future research. The applicability of the results produced here for this application domain should thus be evident to the reader.

Having stated the above applications, we feel that it is appropriate to mention the following, to prevent the reader from possible misunderstandings about the 'intent' and 'mission' of the article. With regard to the content and overall 'mission', a reader may work with the premise that we have proposed a new strategy for solving problems such as the ones discussed above—which are easily and aptly modeled as 'change-point detection' problems. However, we emphasize that this premise is not entirely accurate. Rather, we have intended to propose a solution to a family of MCs, and one of the *applications* of this solution is a new strategy for 'change-point detection'.

Also, one may believe that the techniques for solving such MCs are 'well understood' and 'easy to solve'. Indeed, though an arbitrary solution for any MC is standard and well defined (i.e., the eigenvector of M^T for the eigenvalue unity), the actual solution for obtaining this eigenvector for *specific, arbitrary* chains is not known except when there is a difference equation between the stationary probabilities of the state indices. Furthermore, the solution is also known when the chain possesses some specific phenomena, like the property of 'time reversibility'. Otherwise, we emphasize that the closed-form analytic solution of such random interleaving walk/jump processes is not known. Thus, this is one of the fundamental contributions of the article, and the applications within the testing-related and change-point detection-related domains should be considered secondary.

By way of nomenclature, throughout this article, we shall refer to p as the 'reward probability' and to q as the 'penalty probability', where $p + q = 1$.

2. PROBLEM FORMULATION

2.1. Specification of the State Space and Transitions

We consider the following RW with jumps (RWJ) as depicted in Figure 1. Let $X(t)$ denote the index of the state of the walker at a discrete time instant t. The details of the RW can be catalogued as follows:

1. First of all, observe that the state space of the RW contains $N_1 + N_2 + 1$ states.

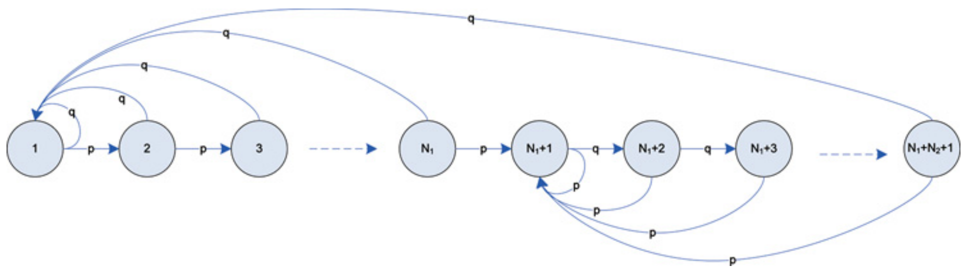


Figure 1. State transitions of the random walk with jumps. (Figure is provided in color online.)

2. The states whose indices are in the set $\{1, \ldots, N_1\}$ are paired with their counterpart random jump to state 1.
3. The states whose indices fall in the range between the integers $\{N_1 + 1, \ldots, N_1 + N_2\}$ are paired with *their* counterpart random jump to state $N_1 + 1$.
4. Finally, the state whose index is $N_1 + N_2 + 1$ is linked to both states 1 and $N_1 + 1$.
5. Essentially, whenever the walker is in a state $X(t) = i$ which belongs to the set $\{1, \ldots, N_1\}$, he has a chance, p, of advancing to the neighboring state $i + 1$, and a chance $q = 1 - p$ of performing a random jump to state 1. Similarly, whenever he is in a state $X(t) = i$ in the set $\{N_1 + 1, \ldots, N_1 + N_2\}$, the walker has a chance, q, of advancing to the neighbor state $i + 1$ and a chance p of operating a random jump to state N_1. However, whenever the walker is in state $N_1 + N_2 + 1$, he has a probability p of jumping to state $N_1 + 1$ and a probability q of jumping to state 1.
6. These rules describe the RWJ completely.

The reader will observe a marginal asymmetry in the assignment of our states. Indeed, one could query: Why should we operate with N_1 and $N_2 + 1$ states in the corresponding modes, instead of N_1 and N_2 respectively? Would it not have been "cleaner" to drop the extra state in the latter case; that is, to use N_2 states instead of $N_2 + 1$? The reason why we have allowed this asymmetry is because we have consciously intended to put emphasis on the so-called 'acceptance state', which counts as unity. Our position is that this is also a more philosophically correct position—because the acceptance state is really one that has already obtained a success.

2.2. Application to Component Testing

As briefly alluded to earlier, from a philosophical point of view, the testing of a component can be modeled by the RWJ presented in Section 2.1. At each time step, the entity is either subject to a success or a failure and is either supposed to be in the *well-functioning* or *malfunctioning* mode. From a high-level perspective, a success "enforces" the hypothesis that the entity is *well-functioning* while simultaneously "weakening" the hypothesis that it is *malfunctioning*. On the other hand, a failure 'enforces' the hypothesis that the entity is deteriorating, that is, *malfunctioning*, while 'weakening' the hypothesis that it is *well-functioning*. It is worth noting that states whose indices are in the set $\{1, \ldots, N_1\}$ serve to memorize the number of consecutive successes that have occurred so far. In other words, if the walker is in state i ($i \in \{1, \ldots, N_1\}$), this implies that we can deduce that the walker has passed the test i consecutive times. Similarly, states whose indices are in the set $\{N_1 + 1, \ldots, N_1 + N_2 + 1\}$ present an indication of the number of consecutive failures that have occurred. In this case, if the walker is in state $N_1 + i$ where $0 < i \leq N_2 + 1$, we can infer that the walker has made i consecutive failures so far.

We present the following mapping of the states of the RW $\{1, \ldots, N_1 + N_2 + 1\}$ to the set of hypotheses $\{$*well-functioning*, *malfunctioning*$\}$ as follows. The mapping is divided into two parts:

Malfunctioning states: We refer to the states $\{1, \ldots, N_1\}$ as being the so-called *malfunctioning* states, because whenever the index $X(t)$ of the current state of the walker is in that set, we conjecture that 'the hypothesis that the component is in its *malfunctioning* mode' is true. In this phase, the state transitions illustrated in the figure are such that any deviance from the hypothesis is modeled by a successful transition to the neighboring state, whereas a failure causes a jump back to state 1. Conversely, only a pure uninterrupted sequence of N_1 successes will allow the walker to pass into the set of *well-functioning* states.

Well-functioning states: We refer to the states $\{N_1 + 1, \ldots, N_1 + N_2 + 1\}$ as the *well-functioning* states, because when in this set of states, we conjecture the hypothesis that the component is in its *well-functioning* mode. More specifically, we refer to state $N_1 + 1$ as being an "acceptance" state because, informally speaking, whenever the walker is in that state, the conjectured hypothesis that the component is *well-functioning* has been confirmed with highest probability. In particular, within theses state, the goal is to detect when the entity deteriorates, causing it to degrade into one of the *malfunctioning* states. These states can be perceived to be the 'opposite' of the *malfunctioning* states in the sense that an uninterrupted sequence of failures is required to "throw" the walker back into the *malfunctioning* mode, whereas a single success reconfirms the conjectured hypothesis that the component is functioning well, forcing the walker to return to the *well-functioning* state space.

We shall now present a detailed analysis of the above RWJ.

3. THEORETICAL RESULTS

The analysis of the above-described RWJ is particularly difficult because the stationary (equilibrium) probabilities of being in any state is related to the stationary probabilities of *non-neighboring* states. In other words, it is not easy to derive a simple difference equation that relates the stationary probabilities of the neighboring states. To render the analysis more complex, we observe that the RW does not possess any time-reversibility properties either.

However, by studying the peculiar properties of the chain, we have succeeded in solving for the stationary probabilities, which, in our opinion, is far from trivial. The proof of the result follows.

Theorem 3.1. *For the RWJ described by the Markov chain given in Figure* 1, P_1, *the probability of the walker being in the* malfunctioning *mode is given by the following expression*:

$$P_1 = \frac{(1 - p^{N_1})q^{N_2}}{(1 - p^{N_1})q^{N_2} + p^{N_1 - 1}(1 - q^{N_2+1})}. \tag{3.1}$$

Similarly, P_2, the probability of being in the well-functioning *mode (or exiting from the* malfunctioning *mode) is*:

$$P_2 = \frac{(1-q^{N_2+1})p^{N_1-1}}{(1-p^{N_1})q^{N_2} + p^{N_1-1}(1-q^{N_2+1})}. \tag{3.2}$$

Proof. To prove these results, we shall analyze the properties of the underlying MC that describes the behavior of the walker. By investigating the various transition considerations, we see that matrix of transition probabilities, M, is given by:

$$M = \begin{pmatrix} q & p & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ q & 0 & p & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q & 0 & \dots & 0 & p & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & p & q & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & p & 0 & q & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & p & 0 & \dots & 0 & q \\ q & 0 & \dots & 0 & p & 0 & 0 & \dots & 0 \end{pmatrix}$$

The reader should observe the transitions into the nonadjacent states; that is, those that represent the jumps.

We shall now compute π_i, the stationary (or equilibrium) probability of the chain being in state i. Clearly, M represents a single closed communicating class whose periodicity is unity. The chain is thus ergodic, and the limiting probability vector is given by the eigenvector of M^T corresponding to the eigenvalue unity. The vector of steady-state (equilibrium) probabilities $\Pi = [\pi_1, \dots, \pi_{N_1+N_2+1}]^T$ can be thus computed by solving $M^T\Pi = \Pi$.

Consider first the stationary probability of being in state 1, π_1. By expanding the first row we see that this is expressed by the following equation:

$$\begin{aligned} \pi_1 &= q\pi_1 + q\pi_2 + \dots + q\pi_{N_1} + q\pi_{N_1+N_2+1} \\ &= q\sum_{k=1}^{N_1}\pi_k + q\pi_{N_1+N_2+1}. \end{aligned} \tag{3.3}$$

For $2 \le k \le N_1$, the stationary probability π_k is given by a straightforward first-order difference equation, equation (3.4):

$$\pi_k = p\pi_{k-1}. \tag{3.4}$$

By applying recurrence, equation (3.4) can be rewritten as:

$$\pi_k = p^{k-1}\pi_1. \tag{3.5}$$

By expanding the $(N+1)$st row, we can see that the probability of being in the 'acceptance' state π_{N_1+1} is given by equation (3.6):

$$\pi_{N_1+1} = p\pi_{N_1} + p\pi_{N_1+1} + p\sum_{k=1}^{N_2}\pi_{N_1+1+k}. \tag{3.6}$$

Again, for $N_1 + 1 \leq k \leq N_1 + N_2 + 1$, the steady-state probabilities are given by:

$$\pi_k = q\pi_{k-1}. \tag{3.7}$$

By applying recurrence, equation (3.7) can be written for $1 \leq k \leq N_2 + 1$ as:

$$\pi_{N_1+1+k} = q^k \pi_{N_1+1}. \tag{3.8}$$

Using equations (3.5) and (3.8), and replacing them in equation (3.3) we obtain:

$$\pi_1 = q \sum_{k=1}^{N_1} p^{k-1} \pi_1 + q^{N_2+1} \pi_{N_1+1}. \tag{3.9}$$

Therefore, we obtain:

$$\pi_1 = (1 - p^{N_1})\pi_1 + q^{N_2+1} \pi_{N_1+1}. \tag{3.10}$$

From the above, we can deduce the equation that relates π_1 and π_{N_1+1}:

$$\pi_{N_1+1} = \frac{p^{N_1}}{q^{N_2+1}} \pi_1. \tag{3.11}$$

Consequently, P_1 is given by equation (3.12):

$$\begin{aligned} P_1 &= \sum_{k=1}^{N_1} \pi_k \\ &= \sum_{k=1}^{N_1} p^{k-1} \pi_1 \\ &= \frac{1 - p^{N_1}}{1 - p} \pi_1. \end{aligned} \tag{3.12}$$

Similarly, P_2 can be expressed by:

$$\begin{aligned} P_2 &= \sum_{k=N_1+1}^{N_1+N_2+1} \pi_k \\ &= \sum_{k=0}^{N_2} q^k \pi_{N_1+1} \\ &= \frac{1 - q^{N_2+1}}{p} \pi_{N_1+1}. \end{aligned} \tag{3.13}$$

Using the values of P_1 and P_2 and the fact that $P_1 + P_2 = 1$, and after carrying out some simple algebraic manipulations, we obtain:

$$\pi_1 = \frac{q^{N_2+1}}{(1 - p^{N_1})q^{N_2} + p^{N_1-1}(1 - q^{N_2+1})}, \tag{3.14}$$

and

$$\pi_{N_1+1} = \frac{p^{N_1}}{(1-p^{N_1})q^{N_2} + p^{N_1-1}(1-q^{N_2+1})}, \tag{3.15}$$

whence:

$$P_1 = \frac{(1-p^{N_1})q^{N_2}}{(1-p^{N_1})q^{N_2} + p^{N_1-1}(1-q^{N_2+1})}, \tag{3.16}$$

and

$$P_2 = \frac{(1-q^{N_2+1})p^{N_1-1}}{(1-p^{N_1})q^{N_2} + p^{N_1-1}(1-q^{N_2+1})}, \tag{3.17}$$

which concludes the proof. □

3.1. 'Balanced Memory' Strategies

Although the results obtained above are, in one sense, pioneering, the question of understanding *how* the memory of the scheme should be assigned is interesting in its own right. To briefly address this, in this section we consider the particular case where N_1 and $N_2 + 1$ are both equal to the same value, N. In this case, if $p = q = 1/2$, one can trivially confirm that $P_1 = P_2 = 1/2$, implying that the scheme is not biased toward either of the two modes, the *malfunctioning* or the *well-functioning* mode. In practice, employing a 'balanced memory' strategy seems to be a reasonable choice because having equal memory depth (or number of states) for the *malfunctioning* and *well-functioning* modes eliminates any bias toward any of the conjectured hypotheses.

Theorem 3.2. *For a 'balanced memory' strategy in which $N_1 = N_2 + 1 = N$, the probability, P_1, of being in the* malfunctioning *mode approaches* 0 *as the memory depth N tends to infinity whenever $p > 0.5$. Formally,* $\lim_{N\to\infty} P_1 = 0$.

Proof. Consider the quotient $\frac{P_1}{P_2}$. To prove this result, we first compute its limit as N tends to infinity for $p > 0.5$.

$$\frac{P_1}{P_2} = \frac{(1-p^N)q^{N-1}}{(1-q^N)p^{N-1}}. \tag{3.18}$$

Dividing the numerator and denominator by p^{2N} we obtain:

$$\frac{P_1}{P_2} = \frac{(1/p^N - 1)(q/p)^{N-1}}{1/p^N - (q/p)^N}. \tag{3.19}$$

Since $p > 0.5$, we have the condition that $q/p < 1$. Therefore, $\lim_{N\to\infty}(q/p)^{N-1} = 0$. On the other hand, $\lim_{N\to\infty} \frac{(1/p^N-1)}{1/p^N-(q/p)^N} = 1$.

Therefore, $\lim_{N\to\infty} \frac{P_1}{P_2} = 0$. Thus, we conclude that $\lim_{N\to\infty} P_1 = 0$, and the result is proved. □

The analogous result for the case when $p < 0.5$ follows.

Theorem 3.3. *For a 'balanced memory' strategy in which $N_1 = N_2 + 1 = N$, the probability, P_1, of being in the* malfunctioning *mode approaches unity as the memory depth N tends to infinity whenever $p < 0.5$. Formally,* $\lim_{N\to\infty} P_1 = 1$.

Proof. The proof is similar to the proof of Theorem 3.2, except that we consider the quotient $\frac{P_2}{P_1}$. By dividing the numerator and denominator by q^{2N}, we get the following expression:

$$\frac{P_2}{P_1} = \frac{(1/q^N - 1)(p/q)^{N-1}}{1/q^N - (p/q)^N}. \tag{3.20}$$

We remark that $p/q < 1$ for $p < 0.5$ and, thus, $\lim_{N\to\infty}(p/q)^{N-1} = 0$. Moreover, by arguing in an analogous manner we can see that $\lim_{N\to\infty} \frac{(1/q^N-1)}{1/q^N-(p/q)^N} = 1$. Therefore, $\lim_{N\to\infty} \frac{P_2}{P_1} = 0$ and, consequently, $\lim_{N\to\infty} P_2 = 0$. Hence the result. □

3.2. Symmetry Properties

The MC describing the RW with Jumps is described by its state occupation probabilities and the overall mode probabilities, P_1 and P_2. It is trivial to obtain P_1 from P_2 and vice versa for the symmetric 'balanced memory' case – one merely has to replace p by q and do some simple transformations. However, the RW also possesses a fascinating property when it concerns the underlying state occupation probabilities – the $\{\pi\}$'s themselves. Indeed, we shall derive one such interesting property of the scheme in Theorem 3.4, which specifies a rather straightforward (but non-obvious) method to deduce the equilibrium distribution of a 'balanced memory' scheme possessing a reward probability p from the equilibrium distribution of the counterpart 'balanced memory' scheme possessing a reward probability of $1 - p$.

Theorem 3.4. *Let $\Pi = [\pi_1, \ldots, \pi_{2N}]^T$ be the vector of steady state probabilities of a balanced scheme characterized by a reward probability p and penalty probability q. Let $\Pi' = [\pi'_1, \ldots, \pi'_{2N}]^T$ be the vector of steady state probabilities of a balanced scheme possessing a reward probability $p' = 1 - p$ and penalty probability $q' = 1 - q$. Then Π' can be deduced from Π using the following transformation*:

$$\pi'_k = \pi_{\sigma(k)} \quad \textit{for } k \textit{ with } 1 \le k \le 2N,$$

where σ is a circular permutation of the set $S = \{1, 2, \ldots, 2N\}$ defined by

$$\sigma(k) = \begin{cases} 2N, & \textit{if } k = N \\ (k+N)(\textit{mod}\, 2N), & \textit{Otherwise} \end{cases}$$

Proof. We shall first prove the theorem for states π'_1 and π'_{N+1}. Using equations (3.14) and (3.15) and replacing $(N_1, N_2 + 1)$ by (N, N), we deduce that: $\pi'_1 = \pi_{N+1} = \pi_{\sigma(1)}$ and $\pi'_{N+1} = \pi_1 = \pi_{\sigma(N+1)}$.

Now, consider the hypothesis for k such that $1 < k \le N$. By a simple substitution we see that for all k $(1 < k < N)$, $\sigma(k) = (k+N)(\mathrm{mod}\, 2N) = k + N$, and for $k = N$, $\sigma(N) = 2N$.

We use equation (3.5) to write

$$\begin{aligned}\pi'_k &= p'^{k-1}\pi'_1 \\ &= q^{k-1}\pi_{N+1}.\end{aligned} \tag{3.21}$$

However, as per equation (3.8), for k such that $2 \leq k \leq N$:

$$q^{k-1}\pi_{N+1} = \pi_{N+k}.$$

Therefore, for k such that $2 \leq k \leq N$:

$$\pi'_k = \pi_{N+k} = \pi_{\sigma(k)}$$

Now, we treat the case where k is bounded as per $N < k \leq 2N$ separately. For this case, first of all, we remark that σ can be expressed differently. Indeed, if k satisfies $N < k \leq 2N$, it can be seen that $\sigma(k) = (k+N)(\bmod 2N) = k - N$. Considering this, we now apply equation (3.8) to yield:

$$\begin{aligned}\pi'_{N+k} &= q'^{k-1}\pi'_{N+1} \\ &= p^{k-1}\pi_1 \\ &= \pi_k.\end{aligned} \tag{3.22}$$

The result is proven by a straightforward change of variables, since we can easily deduce that for $N < k \leq 2N$: $\pi'_k = \pi_{k-N} = \pi_{\sigma(k)}$. □

4. EXPERIMENTAL RESULTS

Apart from the above theoretical results, we have also rigorously tested the RWJ which we have studied in various experimental settings. The simulations have been grouped into three sets. The first of these, experimentally verifies the accuracy of the theoretical expressions of the steady-state probabilities derived above. The second demonstrates the transient properties of the chain. Finally, some simulations have been included to compare our scheme with a hidden Markov model (HMM) scheme.

The question of why one should *experimentally* verify the accuracy of correctly-derived theoretical expressions is truly pertinent. The point is that although the final steady-state probabilities are independent of the starting state, in reality, the *time* it takes to converge to these is dependent on where one starts, and on the size of the machine (MC). Thus, in computer simulations (for example, in the case of Learning Automata), it does occur that the machine converges to the final steady-state accuracy only after 200,000 iterations. If one terminates the simulations prematurely, one observes a difference between the anticipated steady-state value and the one that is observed. We emphasize, though, that our simulations show that for the MC that we investigate here, such a difference is unobservable. We believe that any thorough study of a MC should also include such an experimental verification, which is why this has been undertaken.

In this section, we present some experimental results for cases where the RWJ has been simulated. The goal of the exercise was to understand the sensitivity of the MC to changes in the memory size, the properties of P_1 as a function of the reward probability, the limiting (asymptotic) behavior of the walk, and the characteristics of the RW in nonstationary environments. Although the chain has been simulated for a variety of settings, in the interest of brevity, we present here only a few typical sets of results – essentially, to catalogue the overall conclusions of the investigation.

4.1. Sensitivity of the RWJ to Changes in the Memory Size: Theoretical and Simulation Results

The first study that we undertook has twofold purposes, namely:

– to understand the characteristics of the chain for changes in the memory size.
– to investigate how exact the simulation values match the closed form expression.

The results obtained have been in recorded in Table 1, which summarizes the performance of the MC, for a wide range of reward probabilities, p, numbers of *malfunctioning* states, N_1, and *well-functioning* states, N_2+1. The resulting performance is then reported in Table 1 in terms of the asymptotic mode occupation probability P_1, where P_1 is obtained using closed form expression of Theorem 3.1 as well as simulations. In the simulation settings, we report the average value of P_1 as obtained from an ensemble of 10 experiments, each consisting of 10,000 iterations. From Table 1, we observe a close match between the theoretical results and the simulation, which also explains the rapid convergence to the optimal values of P_1.

From the experimental results and the closed form expression results we can conclude the following:

1. In spite of the fact that we have used a fairly low number of iterations, and that we averaged over a small ensemble size, the simulation results and the theoretical results are almost identical. As mentioned above, this is also due to the fast convergence of our RWJ for small memory settings.

Table 1. The values of P_1 using the closed form expression and the corresponding experimental results for different values of the memory size and the reward probability

	$p=0.9$		$p=0.8$		$p=0.5$		$p=0.2$		$p=0.1$	
(N_1, N_2)	Form.	Simulation	Form.	Simulation	Form.	Simulation	Form.	Simulation	Form.	Simulation
(1, 5)	9.9E-7	9.7E-7	6.39E-5	6.2E-5	0.015	0.016	0.262	0.259	0.531	0.535
(2, 5)	2.11E-6	2.13E-6	1.44E-4	1.3E-4	0.045	0.045	0.680	0.682	0.925	0.92
(3, 5)	3.34E-6	3.56E-6	2.44E-4	2.4E-4	0.1	0.11	0.916	0.912	0.992	0.993
(4, 5)	4.71E-6	4.58E-6	3.68E-4	3.8E-4	0.192	0.19	0.982	0.98	0.999	0.999
(5, 5)	6.24E-6	6.14E-6	5.25E-4	5E-4	0.329	0.33	0.996	0.992	0.999	0.999
(5, 4)	6.24E-5	6.54E-5	2.62E-3	2.1E-3	0.5	0.51	0.997	0.994	0.999	0.999
(5, 3)	6.23E-4	6.41E-4	0.012	0.01	0.674	0.673	0.998	0.999	0.999	0.999
(5, 2)	6.2E-3	6.2E-3	0.062	0.064	0.815	0.815	0.998	0.998	0.999	0.999
(5, 1)	0.059	0.057	0.254	0.261	0.911	0.912	0.99	0.991	0.999	0.999

2. For any given values of N_1 and N_2, the value of P_1 generally decreases monotonically with p. Thus, when N_1 and N_2 are 3 and 5 respectively, the value of P_1 increases from its lowest value of 3.34×10^{-6} for $p = 0.9$, to 0.992 for $p = 0.1$. This is as expected.
3. For any given value of p, if the value of N_2 is fixed, the value of P_1 generally increases monotonically with N_1. Thus, when p is 0.1 and $N_2 = 5$, the value of P_1 increases from its lowest value of 0.531 for $N_1 = 1$ to 0.999 for $N_1 = 5$. This, too, is as expected.
4. Finally, in order to observe the effect of the size of the memory, we consider the column in Table 1 for a reward probability $p = 0.5$. Note that the configuration $(N_1, N_2) = (5, 4)$ corresponds to a balanced memory scheme. If $N_2 + 1 > N_1$, the walk is concentrated in the *well-functioning* states, which has the effect of minimizing P_1. On the other hand, if $N_1 > N_2 + 1$, the walk is concentrated in the *malfunctioning* states, which, in turn, has the effect of maximizing P_1. One can thus easily deduce the effect of the memory on the bias of the MC.

4.2. P_1 as a Function of p, the Reward Probability

In the second set of experiments, we analyzed the value of P_1 as a function of the reward probability, p, for different memory configurations of a balanced memory setup. We report here the cases when N was equal to 3, 5, and 10. By observing the plot of P_1 (see Figure 2), we see that this is a monotonically decreasing function of p, which possesses an inflection point at $p = 1/2$, which confirms the conclusions of Theorems 3.1 and 3.2. Further, from Figure 2 we see that for values of p such that $p > 0.5$, P_1 decreases significantly and tends toward 0 as we increase N from 3 to 10. Conversely, for values of p such that $p < 0.5$, we observe that P_1 increases and tends toward unity as we increase N from 3 to 10. This, too, confirms our earlier theoretical results.

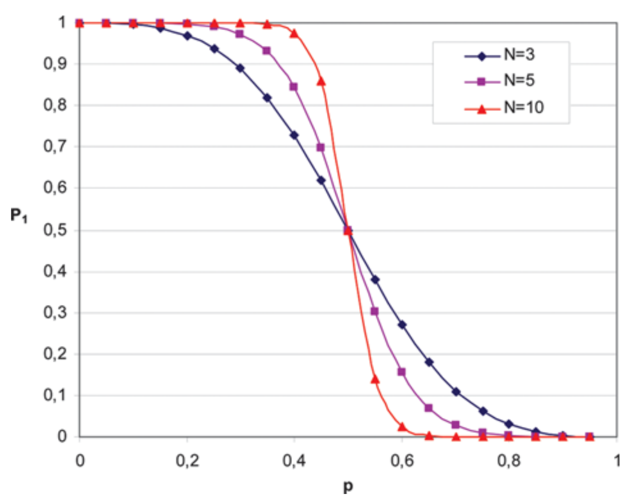
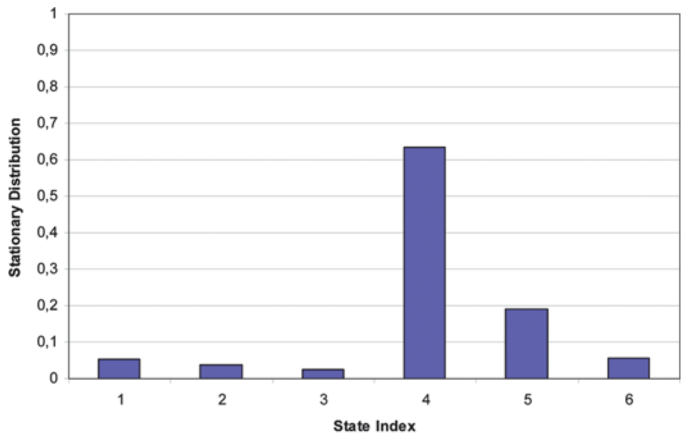


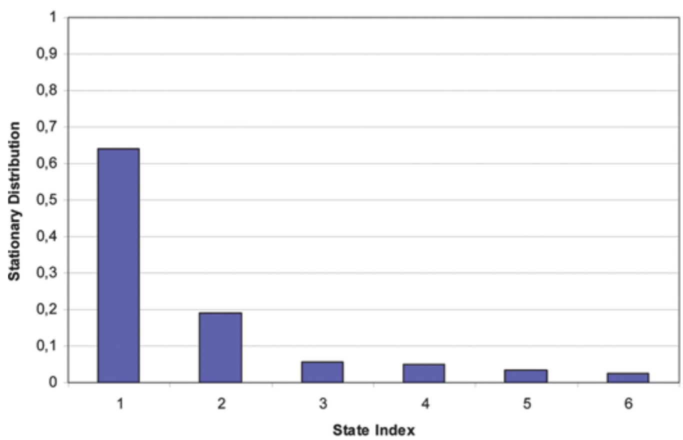
Figure 2. Plot of P_1 as a function of p, the reward probability. (Figure is provided in color online.)

4.3. Limiting Behavior of the RWJ

In our studies, we were also interested in understanding the limiting behavior of the RWJ. To investigate this, we simulated various balanced memory schemes. Indeed, we see that the series π_i monotonically decreases with the state index for $p < 0.5$. In other words, $\pi_1 > \pi_2 > \ldots \pi_{2N}$. Figure 3 depicts the steady-state (equilibrium) distribution associated with two balanced memory chains, each possessing six states ($N = 3$). In the first case (Figure 3(a)), the reward probability was $p = 0.3$, and in the second (Figure 3(b)), the reward probability was $p' = 1 - p = 0.7$. The steady probability of each state was estimated by averaging over 1,000 experiments, each consisting of 100,000 iterations. The reader will appreciate the confirmation of Theorem 3.4 as illustrated by Figure 3. In fact, the steady distribution for $p = 0.3$ can be easily deduced from the steady distribution of $p' = 1 - p = 0.7$.



(b)



(a)

Figure 3. (a) The stationary distribution for a reward probability $p = 0.7$ and (b) the corresponding stationary distribution for a reward probability $p = 0.3$. (Figure is provided in color online.)

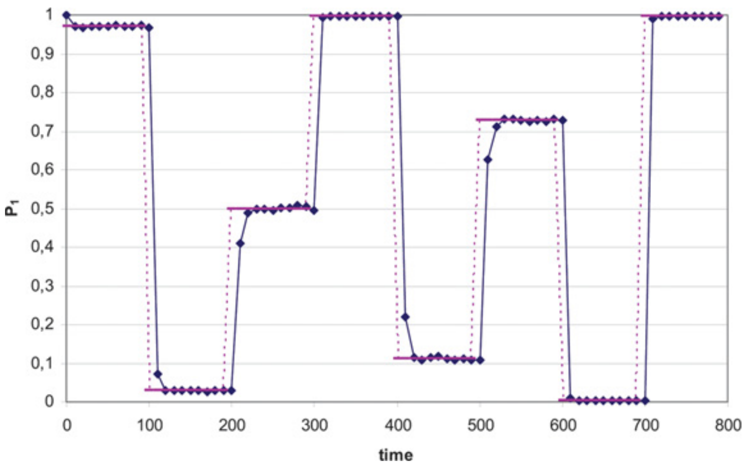


Figure 4. Ability of the scheme to track the target distribution with $N = 3$. (Figure is provided in color online.)

4.4. Investigation of the MCJ for Nonstationary Environments

In this experiment, we were interested in understanding the characteristics of the chain when interacting with a nonstationary environment (i.e., when p changed with time). To do this, we modeled a nonstationary environment by altering the reward probability, p, governing the state transitions at every 100th time slot. Then, as before, P_1 was estimated by averaging over 1,000 experiments, each consisting of 100,000 iterations. In particular, the reward probabilities that we used in the experiments were drawn sequentially every 100 time instants from the reward vector $R = [0.2, 0.8, 0.5, 0.1, 0.7, 0.4, 0.9, 0.1]$. To be more specific, between time instants 0 and 100 the reward probability was equal to 0.2, between instants 100 and 200 the reward probability was equal to 0.8, and so on. In Figure 4, we have plotted the average value of P_1 over the set of experiments using a continuous line and the target distribution of P_1 using a discontinuous (dashed) line, when $N = 3$, where the specific target distribution of P_1 was explicitly calculated using equation (3.16), in which we substituted p with the respective element of the reward vector, R, currently used. For example, between instants 0 and 100, because the reward probability was $p = 0.2$, the target distribution of P_1 was computed to be 0.9701. Similarly, between instants 100 and 200, because the reward probability was $p = 0.8$, the corresponding target distribution was seen to be $P_1 = 0.029$. In Figure 5, we have reported the results of redoing the same experiment (as in Figure 4) but with a different memory $N = 5$. From Figures 4 and 5 we observe that the instantaneous value of P_1 tracks the target distribution drawn in discontinuous line in a in near-optimal manner, which we believe is quite fascinating. The use of this to track the time-varying testing of components is obvious.

4.5. Rate of Convergence

This set of experiments was conducted so that we could better understand the transient behavior of the chain and thus perceive the rate of convergence of P_1 for

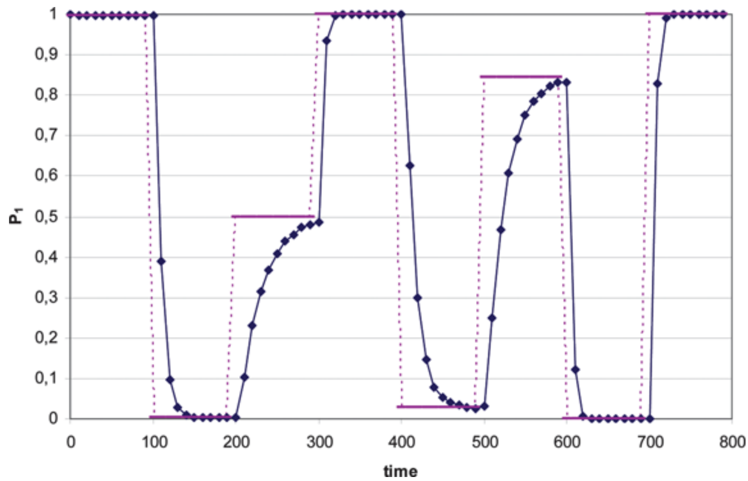


Figure 5. Ability of the scheme to track the target distribution with $N = 5$. (Figure is provided in color online.)

different memory configurations of a balanced memory setup. To do this we fixed the reward probability, p, to be 0.8 while we increased the memory N from $N = 3$ to $N = 10$. The quantity P_1 was then estimated by averaging it over 1,000 experiments. In order to understand the effect of the internal memory of the RWJ on the rate of convergence, we report the number of required iterations to reach a value that 95% of the final value of P_1. In all of the following experiments, the initial state $X(0)$ was set to be state 1.

From Figure 6(a), we see that it took *only* 18 time instants to reach 95% of P_1 for a memory $N = 3$. This, we believe, is remarkable. In Figure 6(b), 95% of P_1 was attained within 55 iterations when we fixed N to 5. Similarly, in Figure 6(c), we chose N to be 7 and it took 123 time instants to converge to 95% of P_1. Finally, in Figure 6(d), when we set N to 10, the required number of iterations was 326.

We remark that as we increased the memory, the RW took more time to converge to the optimal value of P_1. This is, of course, understandable.

4.6. Comparing HMM to RWJ

To compare our model with another existing solution for change-point detection, we now include some simulation results in which we model real-life components using an HMM, as proposed in Rabiner and Juang (1986). In the same vein as our RWJ, we assume that the component is either *well-functioning* or *malfunctioning*. The reader should note that in this section, we consider the real state of the component, which might differ from the one hypothesized by the RWJ—which could conjecture that the component is in a *malfunctioning* mode or in *well-functioning* mode.

We consider an HMM as shown in Figure 7. The two-state Markov model is used where the states correspond to the *well-functioning state* (W) and the *malfunctioning state* (M), respectively. Within this framework, without loss of generality, when the component is in the W state, the failure probability is within

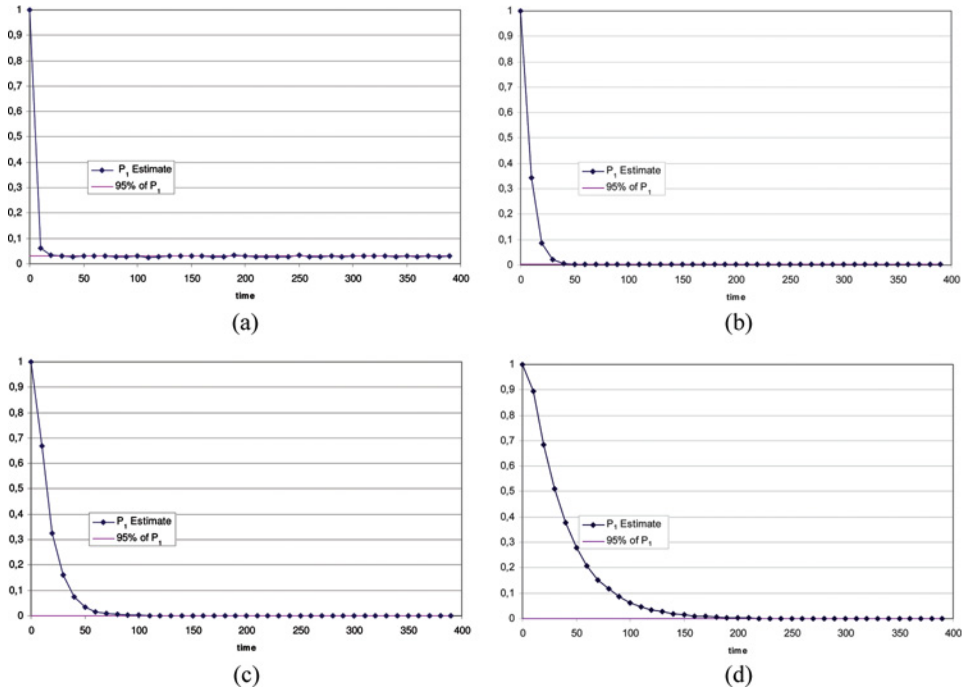


Figure 6. This figure depicts the transient behavior of the chain as a function of time, where we plot (a) $P_1(t)$ for a memory size $N = 3$, (b) $P_1(t)$ for a memory size $N = 5$, (c) $P_1(t)$ for a memory size $N = 7$, and (d) $P_1(t)$ for a memory size $N = 10$. (Figure is provided in color online.)

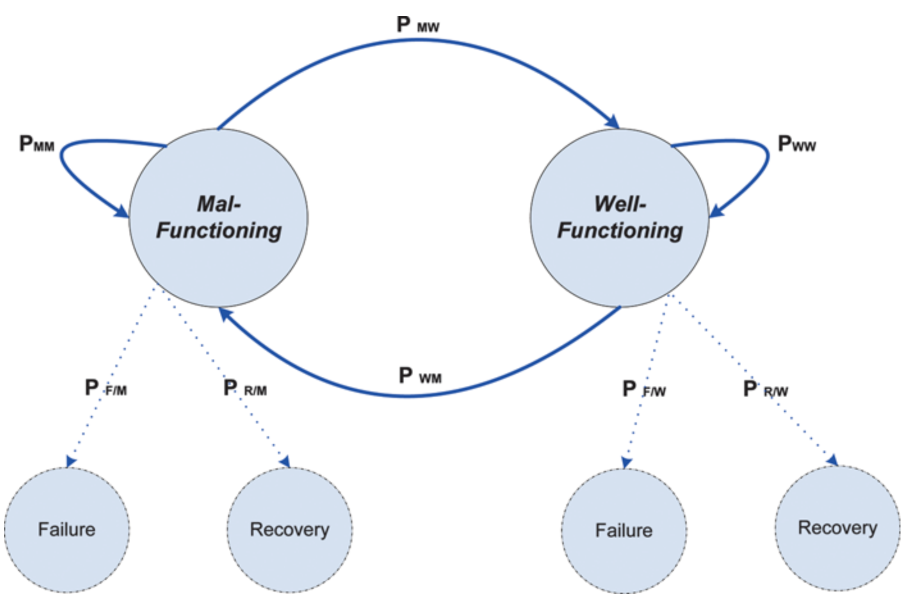


Figure 7. HMM model. (Figure is provided in color online.)

an acceptable level (usually low), and when it is the M state, the component has a high failure probability.

With this HMM model, the state transition probability is given by:

$$\begin{pmatrix} P_{MM} & P_{MW} \\ P_{WM} & P_{WW} \end{pmatrix},$$

where $P_{MW} = P(\text{current state is } M \,|\, \text{previous state is } W)$, etc.

As reported in the classical literature about HMMs, we assume that the states themselves are unobservable but that only the output, which is dependent on the states, is visible. In our case, the output is either the event *failure* or the event *recovery*. Given the parameters of the model, an HMM prediction algorithm, namely, the Viterbi algorithm (explained in Rabiner and Juang, 1986), infers the most likely sequence of states that produced a given output sequence. For the rest of this article, in the interest of simplicity, we shall also refer to the HMM prediction algorithm as HMM. The emission probabilities in the *malfunctioning* state are:

$$P_{F/M} = P(\text{Failure} \,|\, \text{State is } M), \quad \text{and}$$
$$P_{R/M} = P(\text{Recovery} \,|\, \text{State is } M),$$

where $P_{F/M} + P_{R/M} = 1$. Similarly, the emission probabilities in the well-functioning state are

$$P_{F/W} = P(\text{Failure} \,|\, \text{State is } W), \quad \text{and}$$
$$P_{R/W} = P(\text{Recovery} \,|\, \text{State is } W),$$

where, obviously, $P_{F/W} + P_{R/W} = 1$.

Note that the HMM gives the optimal performance when the parameters of the HMM are known. We now compare the performance of our RWJ with that of the HMM, where for a performance metric we use the ratio of correct predictions of the states of the component. In fact, both the HMM and our RWJ are allowed only to observe, at discrete time instants, the events recovery or failure. Based on these events, the HMM and the RWJ try to predict the state of the component.

In the experiments, we chose the following transition matrix:

$$\begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}.$$

We report below the average values of the ratio of correct predictions for both our RWJ and the HMM. Observe that both algorithms are fed with the same sequence of observations of length 200, and the results obtained from an ensemble average of 100 replicated experiments is reported Table 2 for different parameters of the emission probabilities.

We tested the RWJ for different parameters of the internal memory, where we empirically chose the value $N = 2$ because we experimentally observed that the RWJ was able to achieve better performance than for other memory configurations. As a primary observation, we remark that the RWJ yielded an acceptable performance that approached the *optimal* performance achieved by the HMM. Further, as the gap between $P_{R/M}$ and $P_{R/W}$ was reduced, one observes that it is increasingly difficult

Table 2. Ratio of correct predictions for the RWJ studied here and the HMM for different emission probabilities

$(P_{R/M}, P_{R/W})$	RWJ	HMM
(0.1, 0.9)	0.813	0.932
(0.15, 0.9)	0.788	0.92
(0.2, 0.8)	0.748	0.86
(0.25, 0.8)	0.72	0.85
(0.3, 0.7)	0.67	0.76
(0.4, 0.7)	0.622	0.698
(0.4, 0.6)	0.6	0.62
(0.5, 0.5)	0.5	0.52

for both the HMM and the RWJ to infer the correct states of the component, resulting in a decline in the performance of both algorithms. For example, when $(P_{R/M}, P_{R/W}) = (0.4, 0.6)$, the RWJ yielded a ratio of right prediction of 0.6 and the HMM achieved the index of 0.62.

4.6.1. Degrading the Parameters of the HMM

We now consider the cases when the the parameters of the HMM are degraded. In this experiment, we fixed the memory size to be $N = 2$ and gradually altered the underlying transitions probability of the HMM, namely, the emission probability $P_{R/M}$. With regard to settings, the HMM was initially configured with $P_{R/M} = 0.1$ and $P_{R/W} = 0.9$. Thereafter, at every 50th time instant, we increased $P_{R/M}$ by 0.05. The total number of iterations that we considered was 200. We observe that as we altered the parameters of the HMM with time, the optimal HMM prediction algorithm was actually operating under inaccurate parameters. As depicted in Figure 8, at around instant 150, our RWJ prediction ratio outperformed the HMM.

These results demonstrate that, under dynamic environments where the parameters of the HMM change over time, the optimal performance of the HMM is dependent on tracking the correct transitions probabilities between the states, which has to be achieved by continuously training the HMM. As opposed to this, our RWJ does not require a knowledge and/or estimation of the underlying transition probabilities. Rather, the RWJ is able to adapt to changes brought about by the dynamic nature of the environment.

4.6.2. Improving Performance via Biased Memory

We now consider the effect of improving the performance of both schemes by adding a bias to their memory requirements. To do this, we performed experiments where the component was more likely to be in the *well-functioning* state. In other words, the values of the transition probabilities were chosen in such a way that $P(W) > P(M)$ or, equivalently, $P_{MW}/(P_{MW} + P_{WM}) > P_{WM}/(P_{MW} + P_{WM})$. In this case, the component will spend most of its time in the *well-functioning* state. Note that this setting leads to experiments that are different from the above where the transition matrix was chosen to satisfy $P(W) = P(M)$. Given this fact, we observe that the performance of the RWJ can be drastically improved when compared to

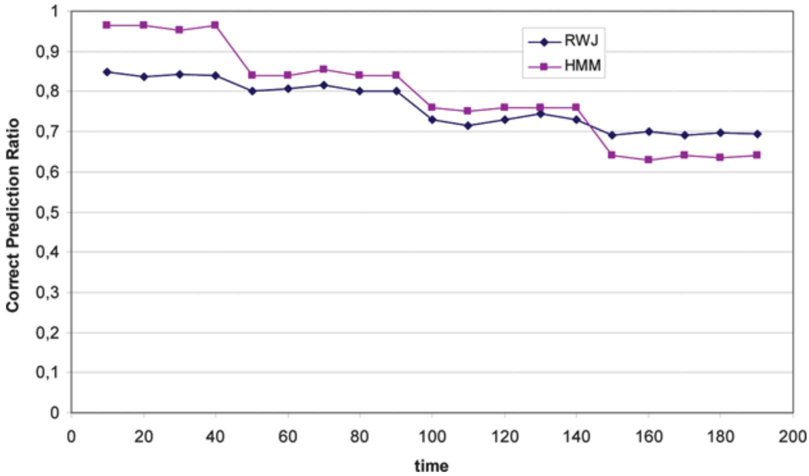


Figure 8. Plot of the performance of the HMM and the RWJ when the transition probabilities between the states are changing. (Figure is provided in color online.)

Table 3. Ratio of correct predictions for both the RWJ and the HMM for different values of $P(W)$

$P(W)$	(N_1, N_2)	RWJ	HMM
0.95	(1, 4)	0.92	0.95
0.9	(1, 3)	0.84	0.87
0.85	(1, 2)	0.81	0.83
0.8	(2, 3)	0.78	0.81
0.75	(2, 3)	0.71	0.76
0.7	(2, 3)	0.672	0.69

the previous case where the $P(W) = P(M)$. In fact, we can modify the memory parameters N_1 and N_2 in such a way that $N_2 + 1 > N_1$, thus rendering the predictions of the RWJ to be biased toward the hypothesis of being in the *well-functioning* mode. Indeed, experimentally we observed that the RWJ approaches the optimal values of the HMM. In Table 3, we report some of the results of the ratio of predictions for different values of $P(W)$ and for different settings of the memory (N_1, N_2), where the values of (N_1, N_2) were chosen empirically.

As an overall observation, we see that our RWJ achieves a near optimal prediction ratio while using a small memory trace. In addition, the RWJ does not need to maintain the history of the observation or require a training phase to infer the right values of the transitions probabilities. We thus assert that the RWJ is a viable alternative for predicting the state of the component, even though in some situations we may have to sacrifice the prediction accuracy marginally.

5. CONCLUSIONS

In this article, we have analyzed a novel random walk with interleaving steps and jumps, which has potential applications in the area of component testing. Although, as explained in Feller (1968), RWs with single-step transitions have been extensively studied for almost a century, problems involving the analysis of RWs that contain interleaving random steps and random 'jumps' are intrinsically hard. In this article, we have considered the analysis of one such fascinating RW, where every step is paired with its counterpart random jump. As mentioned, the RW has applications in the testing of entities (components or personnel), because we can constrain the entity to never be allowed to make more than a prespecified number of *consecutive* failures. The article contains and detailed analysis of the chain, some fascinating limiting properties, a numerous simulations that justify the analytic results. The article also includes some simulation results for the chain's transient behavior. Finally, we have also conducted a comparative testing against a hidden Markov model, which demonstrated that within the testing framework, the results of our model are competitive if not superior.

We believe that the entire field of RWs with interleaving steps and jumps is novel, and we believe that this is a pioneering article in this field.

ACKNOWLEDGMENTS

The first author gratefully acknowledges the financial support of *Ericsson Research*, Aachen, Germany, and the third author is grateful for the partial support provided by NSERC, the Natural Sciences and Engineering Research Council of Canada. We are also grateful to the anonymous referees, the Associate Editor, and the Editor-in-Chief for their comments on an earlier version of the article.

REFERENCES

Altman, A. and Tennenholtz, M. (2005). Ranking Systems: The Pagerank Axioms, In *Proceedings of the Sixth ACM Conference on Electronic Commerce*, M. Kearns and M. Reiter, eds., pp. 1–8, New York: ACM.

Berg, H. C. (1993). *Random Walks in Biology*, revised edition, Princeton: Princeton University.

Bishop, P. G. and Pullen, F. D. (1991). A Random Walk Through Software Reliability Theory, In *Mathematical Structures for Software Engineering*, B. De Neuman, D. Simpson, and G. Slater, eds., New York: Clarendon.

Bower, G. H. (1994). A Turning Point in Mathematical Learning Theory, *Psychological Review* 101: 290–300.

Camp, T., Boleng, J., and Davies, V. (2002). A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communications & Mobile Computing* 2: 483–502.

Feller, F. (1968). *An Introduction to Probability Theory and Its Applications*, vol. 1, third edition, New York: Wiley.

Fouss, F., Pirotte, A., Renders, J.-M., and Saerens, M. (2007). Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation, *IEEE Transactions on Knowledge and Data Engineering* 19: 355–369.

Gross, D. and Harris, C. M. (1998). *Fundamentals of Queueing Theory*, New York: Wiley.

Nowak, M. A. (1996). *Evolutionary Dynamics: Exploring the Equations of Life*, Cambridge: Harvard University.

Oommen, B. J. (1986). Absorbing and Ergodic Discretized Two-Action Learning Automata, *IEEE Transactions on Systems, Man, and Cybernetics* 16: 282–293.

Paulsen, J. (1983). Ruin Theory with Compounding Assets – A Survey, *Insurance: Mathematics and Economics* 22: 3–16.

Pearson, K. (1905). The Problem of the Random Walk, *Nature* 72: 342.

Rabiner, L. and Juang, B. (1986). An Introduction to Hidden Markov Models, *IEEE ASSP Magazine* 3: 4–16.

Takacs, L. (1969). On the Classical Ruin Problems, *Journal of American Statistical Association* 64: 889–906.

Wade, M. J. and Goodnight, C. J. (1998). The Theories of Fisher and Wright in the Context of Metapopulations: When Nature Does Many Small Experiments, *Evolution* 52: 1537–1553.

Appendix F

Paper VI

Title: A New Family of Stochastic Discretized Weak Estimators operating in Non-Stationary Environments

Authors: **Anis Yazidi**[1], B. John Oommen[1,2] and Ole-Christoffer Granmo[2]

Affiliations: [1]University of Agder, Faculty of Engineering and Science, P. O. Box 509, NO-4898 Grimstad, Norway
[2]School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6

Journal: *To be submitted for publication.*

A New Family of Stochastic Discretized Weak Estimators Operating in Non-Stationary Environments*

Anis Yazidi†, B. John Oommen‡, Ole-Christoffer Granmo§

Abstract

The task of designing estimators that are able to track time-varying distributions has found promising applications in many real-life problems. A particularly interesting family of distributions are the binomial/multiomial distributions. Existing approaches resort to sliding windows that track changes by discarding old observations. In this paper, we report a novel estimator referred to as the Stochastic Discretized Weak Estimator (SDWE), that is based on the principles of Learning Automata (LA). In brief, the estimator is able to estimate the parameters of a time varying binomial distribution using finite memory. The estimator tracks changes in the distribution by operating on a controlled random walk in a discretized probability space. The steps of the estimator are discretized so that the updates are done in jumps, and thus the convergence speed is increased. The analogous results for binomial distribution have also been extended for the multinomial case. Interestingly, the estimator possesses a low computational complexity that is independent of the number of parameters of the multinomial distribution. The paper briefly reports conclusive experimental results that demonstrate the ability of the SDWE to cope with non-stationary environments with high adaptation rate and accuracy.

Keywords : *Weak Estimators, Learning Automata, Non-Stationary Environments*

1 Introduction

Estimation is a fundamental and substantial issue in statistical problems. Estimators generally fall into various categories including the Maximum Likelihood Estimates (MLE) and the Bayesian family

*The first author gratefully acknowledges the financial support of the *Ericsson Research*, Aachen, Germany, and the third author is grateful for the partial support provided by NSERC, the Natural Sciences and Engineering Research Council of Canada. A preliminary version of this paper will be presented at ICNC'12, the 2012 International Conference on Computing, Networking and Communications, Hawaii, USA, in February 2012.

†This author can be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: `anis.yazidi@uia.no`.

‡*Chancellor's Professor* ; *Fellow: IEEE* and *Fellow: IAPR*. This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. E-mail address: `oommen@scs.carleton.ca`.

§This author can also be contacted at: Department of ICT, University of Agder, Grimstad, Norway. E-mail address: `ole.granmo@uia.no`.

of estimates. The MLE and Bayesian estimates are well-known for having good computational and statistical properties. However, the basic premise for establishing the quality of estimates is based on the assumption that the parameters being estimated do not change with time, i.e, the distribution is assumed to be stationary. Thus, it is desirable that the estimate converges to the true underlying parameter with probability 1, as the number of samples increases.

Consider, however, the scenario when the parameter being estimated changes with time. Thus, for example, let us suppose that the Bernoulli trials leading to binomially distributed random variable were done in a time-varying manner, where the parameter *switched*, for example, periodically, to possibly a new random value. In the binomial case, this implies that the parameter of the binomial distribution *switches* periodically. Such a scenario demonstrates the behavior of a non-stationary environment. Thus, in this case, the goal of an estimator scheme would be to estimate the parameter, and to be able to adapt to any changes occurring in the environment. In other words, the algorithm must be able to detect the changes and estimate the new parameter after a *switch* has occurred in the environment. If one uses strong estimators (i.e., estimators that converge w.p. 1), it is impossible for the learned parameter to change rapidly from the value to which it has converged, resulting in poor time-varying estimates.

As opposed to the traditional MLE and Bayesian estimators, we propose a novel discretized weak estimator, referred to as the Stochastic Discretized Weak Estimator (SDWE), that can be shown to converge to the true value fairly quickly, and "*unlearn*" what it has learned so far to adapt to the new, "*switched*" environment. The convergence of the estimate is weak, i.e., with regard to the first and second moments. The analytic results derived and the empirical results obtained demonstrate that the SDWE estimator is able cope with non-stationary environments with high adaptation rate and accuracy.

With regard to their applicability, apart from the problem being of importance in its own right, weak estimators admit a growing list of applications in various areas such as intrusion detection systems in computer networks [19], spam filtering [23], ubiquitous computing [9], fault tolerant routing [14], adaptive encoding [16], and topic detection and tracking in multilingual online discussions [18].

2 State-of-the-Art

Traditionally available methods that cope with non-stationary distributions resort to the so-called *sliding window* approach, which is a limited-time variant of the well-known MLE scheme. The latter model is useful for discounting stale data in data stream observations. Data samples arrive continuously and only the most recent observations are used to compute the current estimates. Any data occurring outside the current window is forgotten and replaced by the new data. The problem with using sliding windows is the following: If the time window is too small the corresponding estimates tend to be poor. As opposed to this, if time window is too large, the estimates prior to the change of the parameter have too much influence on the new estimates. Moreover, the observations during the entire window width must be maintained and updated during the process of estimation.

Apart from the sliding window approach, many other methods have been proposed, which deal with the problem of detecting change points during estimation. In general, there are two major competitive sequential change-point detection algorithms: Page's cumulative sum (CUSUM) [2] detection procedure and the Shiryaev-Roberts-Pollak detection procedure. In [17], Shiryayev used a Bayesian approach to detect changes in the parameters distribution, where the change points were assumed to obey a geometric distribution. CUMSUM is motivated by a maximum likelihood ratio test for the hypotheses that a change occurred. Both approaches utilize the log-likelihood ratio for the hypotheses that the change occurred at the point, and that there is no change. Inherent limitations of CUMSUM and the Shiryaev-Roberts-Pollak approaches for on-line implementation are the demanding computational and memory requirements. In contrast to the CUMSU and the Shiryaev−Roberts−Pollak approaches, our SDWE avoids the intensive computations of ratios, and does not invoke hypothesis testing.

In earlier works [5, 6, 7], Koychev *et al.* introduced the concept of Gradual Forgetting (GF). The GF process relies on assigning weights that decrease over time to the observations. In this sense, the GF approach assigns most weight to the more recent observations, and a lower weight to the more-distant observations. Hence, the influence of old observations (on the running estimates) decreases with time. It was shown in [7] that the GF can be an enhancement to the sliding window paradigm. In this sense, observations within each sliding window are weighted using a GF function.

Recently, Oommen and Rueda [15] presented a strategy by which the parameters of a binomial/multinomial distribution can be estimated when the underlying distribution is non-stationary. The method has been referred to as the Stochastic Learning Weak Estimator (SLWE), and is based on the principles of *continuous* stochastic Learning Automata (LA). As opposed to this, our scheme resorts to *discretizing* the probability space [1, 8, 13, 20], and performing a controlled random walk on this discretized space. It is well known in the field of LA that discretized schemes achieve faster convergence speed than continuous schemes [1, 12]. By virtue of discretization, our estimator realizes fast adjustments of the running estimates by jumps, and thus it is able to robustly track changes in the parameters of the distribution after a switch has occurred in the environment. It is worth noting that the concept of discretizing the probability space was pioneered by Thathachar and Oommen in their study on reward-inaction LA [20], and since then that it has catalyzed a significant research in the design of discretized LA [1, 3, 4, 8, 13]. Recently, there has been an upsurge of research interest in solving resource allocation problems based on novel discretized LA [3, 4]. In [3, 4], the authors proposed a solution to the class of *Stochastic Nonlinear Fractional Knapsack* problems where resources had to be allocated based on incomplete and noisy information. The latter solution was applied to resolve the web-polling problem, and to the problem of determining the optimal size required for estimation.

To the best of our knowledge, our SDWE is the first reported discretized counterpart of the continuous SLWE [15]. The numerous successful applications of the continuous SLWE reported in [14, 16, 18, 23] motivates our SDWE, and elucidates its relevance for tracking non-stationary distributions in real-life problems.

2.1 Contributions of the Paper

In this paper, we provide a novel discretized estimator based on the principles of LA. The scheme presents a number of notable contributions that can be summed up as follows:

- To the best of our knowledge, the SDWE is the first reported discretized estimator that is able to track a time varying binomial/multnimoial distribution.
- The scheme uses the discretizing principle of LA, and operates by means of a controlled random walk on the probability space. Indeed, by virtue of discretization, our SDWE yields faster convergence speed than analogous continuous weak estimators.
- The SDWE also possesses a low computational complexity, measured in terms of the number of updates per time step to the estimates vector. Interestingly, this index is independent of the number of parameters of the multinomial distribution to be estimated. In fact, the SDWE makes at most *two* updates per time step, thus rendering the worst case complexity to be constant or $O(1)$. To the best of our knowledge, this characteristic is unique when compared to the other estimators including the acclaimed SLWE which possesses a complexity of $O(r)$, where r is the number of parameters of the multinomial variable.
- Apart from being computationally efficient, the scheme is memory efficient and can be implemented using simple finite state machines.
- Finally, with regard to the design and analysis of Random Walks (RWs), we submit that a fundamental contribution of this paper is the manner in which we have designed the estimation process for the multinomial distribution, by reducing/projecting the latter onto multiple binomial state spaces. This issue will be discussed, in detail, later.

3 Properties of the SDWE with Regard to LA

Our devised SDWE is based on the theory of LA [10, 22], and in, particular, on the family of *ergodic* and *discretized* LA. In fact, according to their Markovian representation, automata fall into two categories: ergodic automata and automata possessing absorbing barriers. Such automata are locked into a barrier state after a finite number of iterations. Many families of automata that posses absorbing barriers have been reported [10]. Ergodic automata have also been investigated in [10, 12]. These automata converge in distribution and thus, the asymptotic distribution of the action probability vector has a value that is independent of the corresponding initial vector. While ergodic LA are suitable for non-stationary environments, absorbing automata are preferred in stationary environments. In fact, ergodic automata are known to better adapt to non-stationary environments where the reward probabilities are time dependent. In the next section, we shall present how our automata is ergodic. In a parallel vein, with respect to the values that the action probabilities can take, the LA typically fall into one of the two categories, namely, Continuous and Discretized. Continuous LA permit the action probabilities to take any value in the interval $[0, 1]$. In practice, the relatively slow rate of convergence of these algorithms constituted a limiting factor in their applicability. In order to increase their speed of convergence, the concept of discretizing the probability space was introduced in [12, 20]. This concept is implemented by restricting the probability of choosing an action to be one of finite number of values in the interval $[0, 1]$. If the values allowed are equally spaced in this interval, the discretization is said to be *linear*, otherwise, the discretization is called *non-linear*. Following the discretization concept, many of the continuous Variable Structure Stochastic Automata (VSSA) have been discretized; indeed, discretized versions of almost all continuous automata have been reported [12, 11]. Families of Pursuit and Estimator-based LA have been shown to be faster than VSSA [21]. As a matter of a fact, even faster discretized versions of these schemes have been reported [1, 12].

We shall presently argue how our automata is linearly discretized. In brief, our estimator relies on the principle of discretization in order to hasten the convergence speed, and on the phenomenon of ergodicity to be able to cope with non-stationary distributions.

4 The Estimator for Binomial Distributions

We assume that we are estimating the parameters of a binomial distribution. The binomial distribution is characterized by two parameters, namely the number of trials and the parameter characterizing each Bernoulli trial. We assume that the number of observations is the number of trials. We seek to estimate the Bernoulli parameter for each trial.

Let X be a binomially distributed random variable, which takes on the value either “1” or “2”. We choose to use these values instead of the more common used notation “0” or “1” to make the notation consistent when we consider the multinomial case. It is assumed that the distribution of X is characterized by the parameter $S = [s_1, s_2]^{\mathrm{T}}$. In other words,

$X =$ “1” with probability s_1

$X =$ "2" with probability s_2, where $s_1 + s_2 = 1$.

Let $x(t)$ be a concrete realization of X at time 't'. We intend to estimate S, i.e, s_i for $i = 1, 2$. We achieve this by maintaining a running estimate of $P(t) = [p_1(t), p_2(t)]^{\mathrm{T}}$ of S where $p_i(t)$ represents the estimate of s_i at time n, for $i = 1, 2$. Our proposed SDWE works in a discretized manner. In fact, we enforce the condition that $p_i(t)$ takes values from a finite set, i.e, $p_i(t) \in \{0, 1/N, 2/N, \ldots, 1\}$, where N is a user-defined integer parameter. N is called resolution parameter and determines the stepsize Δ ($\Delta = 1/N$) relevant to the updating of the estimates. A larger value of N will ultimately imply a more accurate convergence to the unknown parameter S. However, a small value of N will hasten the convergence rate but sacrificing some accuracy.

Initially, we assign $p_1(0) = p_2(0) = N/2$, where N is assumed to be an even integer. Thereafter, the value of $p_1(n)$, is updated as follows:

If $x(t) =$ "1" and $rand() \leq 1 - p_1(t)$ and $0 \leq p_1(t) < 1$

$$p_1(t+1) := p_1(t) + \frac{1}{N} \tag{1}$$

If $x(t) =$ "2" and $rand() \leq 1 - p_2(t)$ and $0 < p_1(t) \leq 1$

$$p_1(t+1) := p_1(t) - \frac{1}{N} \tag{2}$$

$$p_1(t+1) := p_1(t) \text{ Otherwise,} \tag{3}$$

where $p_2(t+1) = 1 - p_1(t+1)$ and $rand()$ is a uniform random number generator function. In the interest of simplicity, we omit the time index t, whenever there is no confusion and thus, P implies $P(t)$.

We state below two fundamental theorems concerning our scheme. The first theorem is about the distribution of the vector P which estimates S as per Equations (1), (2) and (3). We affirm that P converges in distribution. The mean of P is shown to converge exactly to the mean of S. The second theorem is about the variance of the estimate, describing the rate of convergence in relation to the variance, and its dependence on N. We will show that a small N results in fast convergence and a large variance, and that a large value of N will lead to slow convergence and a small variance. As we will see, the choice of this user-defined learning parameter, N, summarizes the trade off between the speed and the corresponding accuracy.

Since the proofs of the theorems are rather intertwined, we shall prove them together.

Theorem 1 *Let X be a binomially distributed random variable, and $P(t)$ be the estimate of S at time t obtained by Equation (1), (2) and (3). Then $E[P(\infty)] = S$.*

Theorem 2 *Let X be a binomially distributed random variable, and $P(t)$ be the estimate of S at time t. Then the algebraic expression for the variance of $P(\infty)$ is fully determined by N. Moreover, when $N \to \infty$ the variance tends to zero, implying mean square convergence.*

Proof:

We now present the proofs of Theorems 1 and 2.

In Theorem 1, our aim is to prove that as the index t is increased indefinitely, the expected value of the $p_1(t)$ converges towards s_1, implying that: $\lim_{t\to\infty} E[p_1(t)] \to s_1$.

We shall prove the above by analyzing the properties of the underlying Markov chain, which is specified by the rules (1), (2) and (3). In brief, rules (1), (2) and (3) obey the Markov chain with transition matrix $H = [h_{ij}]$, where.

$h_{j,j-1} = s_2 \frac{j}{N}$, $0 < j \leq N$,

$h_{j,j+1} = s_1(1 - \frac{j}{N})$, $0 \leq j < N$,

$h_{j,j} = 1 - h_{j,j-1} - h_{j,j+1}$, $0 < j < N$,

and, accordingly

$h_{0,0} = 1 - h_{0,1}$

$h_{N,N} = 1 - h_{N,N-1}$.

We shall now compute π_k, the stationary (or equilibrium) probability of the chain being in state k. Clearly H represents a single closed communicating class whose periodicity is unity. The chain is ergodic, and the limiting probability vector is given by the eigenvector of H^T corresponding to the eigenvalue unity.

The vector of steady state probabilities $\Pi = [\pi_1, \ldots, \pi_N]^T$ can be computed using $H^T\Pi = \Pi$ as:

$$\begin{bmatrix} h_{0,0} & h_{1,0} & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ h_{1,0} & h_{1,1} & h_{1,2} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & h_{2,1} & h_{2,2} & h_{2,3} & 0 & \cdot & \cdot & 0 \\ \vdots & \cdot & \ddots & \ddots & \ddots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & h_{k,k-1} & h_{k,k} & h_{k,k+1} & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot & \cdot & h_{N-1,N-2} & h_{N-1,N-1} & h_{N-1,N} \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & h_{N,N-1} & h_{N,N} \end{bmatrix}^T \begin{bmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \cdot \\ \vdots \\ \cdot \\ \pi_{N-1} \\ \pi_N \end{bmatrix} = \begin{bmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \cdot \\ \vdots \\ \cdot \\ \pi_{N-1} \\ h\pi_N \end{bmatrix} \quad (4)$$

Consider first the stationary probability of being in state 0, π_0. Expanding the first row of Equation (4) yields:

$$\pi_0 h_{0,0} + \pi_1 h_{1,0} = \pi_0 \implies \pi_1 = \frac{(1 - h_{0,0})}{h_{1,0}}\pi_0 = \frac{h_{0,1}}{h_{1,0}}\pi_0. \quad (5)$$

Expanding the second row of Equation (4) and substituting (5) yields:

$$\pi_0 h_{0,1} + \pi_1 h_{1,1} + \pi_2 h_{2,1} = \pi_1 \implies \pi_2 = \frac{h_{1,2}}{h_{2,1}}\pi_1. \tag{6}$$

Arguing in the same manner, and after some algebraic simplifications, we obtain the recurrence:

$$\pi_k = \frac{h_{k-1,k}}{h_{k,k-1}}\pi_{k-1}. \tag{7}$$

Using Equation (7) and substituting $h_{k-1,k}$ and $h_{k,k-1}$ gives:

$$\begin{aligned}
\pi_k &= \pi_0 \prod_{i=1}^{k} \frac{h_{i-1,i}}{h_{i,i-1}} \\
&= \pi_0 \prod_{i=1}^{k} \frac{s_1(1-\frac{i-1}{N})}{s_2\frac{i}{N}} \\
&= \pi_0 \prod_{i=1}^{k} (\frac{N-i+1}{i})(\frac{s_1}{s_2})^k \\
&= \binom{N}{k}(\frac{s_1}{s_2})^k \pi_0.
\end{aligned} \tag{8}$$

Consider the sum $\sum_{k=0}^{N} \pi_k$. Using Equation (8) and applying the Binomial theorem gives:

$$\begin{aligned}
\sum_{k=0}^{N} \pi_k &= \pi_0 \sum_{k=0}^{N} \binom{N}{k}(\frac{s_1}{s_2})^k \\
&= \pi_0(1+\frac{s_1}{s_2})^N \\
&= \frac{\pi_0}{s_2^N}.
\end{aligned}$$

Using the fact that $\sum_{k=0}^{N} \pi_k$ sums to unity, we obtain:

$$\pi_0 = s_2^N. \tag{9}$$

Replacing Equation (9) in Equation (8), we obtain a closed form expression for the stationary probability π_k as:

$$\pi_k = \binom{N}{k} s_1^k s_2^{N-k} \tag{10}$$

Let X^* be the limiting index of the state of the random walker. From the above, clearly X^* is binomially distributed, with parameters s_1 and N.

Hence, $E[X^*] = Ns_1$ and $var[X^*] = Ns_1 s_2$.

Consequently, $E[p_1(\infty)] = \frac{1}{N}E[X^*] = s_1$, and $var[p_1(\infty)] = \frac{1}{N^2}var[X^*] = \frac{1}{N}s_1 s_2$.

Thus, Theorem 1 is proved. Using the fact that $\lim_{N\to\infty} var[p_1(\infty)] = \lim_{N\to\infty} \frac{1}{N}s_1 s_2 = 0$, we see that Theorem 2 is also proved.

5 The Estimator for Multinomial Distributions

In this section, we shall consider the problem of estimating the parameters of a multinomial distribution, which is a generalization of the binomial case introduced earlier. The multinomial distribution is characterized by two parameters, namely, the number of trials, and a probability vector which determines the probability of a specific event. In this regard, we assume that the number of observations is the number of trials. Thus, we deal with the problem of estimating the latter probability vector associated with the set of possible outcomes.

Let X be a multinomially distributed random variable, taking values from the set {"1","2", ..., "r"}. Again, we assume that X is governed by the distribution $S = [s_1 \ldots s_r]^T$ as follows:

X = "i" with probability s_i, where $\sum_{i=1}^{r} s_i = 1$.

Let $x(t)$ be a concrete realization of X at time "t". The task at hand is to estimate S, i.e., s_i for $i = 1 \ldots r$. We achieve this by maintaining a running estimate $\mathrm{P(t)} = [\mathrm{p_1(t)}, \ldots, \mathrm{p}_R\mathrm{(t)}]^{\mathrm{T}}$ of S, where $p_i(t)$ is the estimate of s_i at time t. Again we omit the time reference t in $P(t)$ whenever this does not lead to confusion.

We assume that the resolution is a multiple of the number of parameters r, i.e, $N = r\delta$ where δ is an integer. Therefore, for all i, we initialize $p_i(0)$ as per the following $p_i(0) = \frac{\delta}{N} = \frac{1}{r}$.

Generalizing the update scheme used in the binomial case, we get the following update scheme for the multinomial case:

If $x(t)$ = "i" and $rand() \leq 1 - p_i(t)$ and $0 \leq p_i(t) < 1$

$$\text{(i) } p_i(t+1) := p_i(t) + \frac{1}{N}. \tag{11}$$

Randomly choose p_j, $j \neq i$, according to the normalized probability $\frac{p_j(t)}{\sum\limits_{k\neq i} p_k(t)}$ and update p_j as:

$$\text{(ii) } p_j(t+1) := p_j(t) - \frac{1}{N} \tag{12}$$

$$P(t+1) := P(t) \text{ Otherwise.} \tag{13}$$

As before, we shall state two theorems about the properties of the estimate first and prove them together.

Theorem 3 *Let X be a multinomially distributed random variable, and $P(t)$ be the estimate of S at time t obtained by Equations (11), (12) and (13). Then $E[P(\infty)] = S$.*

Theorem 4 *Let X be a multinomially distributed random variable, and $P(t)$ be the estimate of S at time t obtained by Equations (11), (12) and (13). Then algebraic expression for the variance of $P(\infty)$ is fully determined by N. Moreover, when $N \to \infty$ the variance tends to zero, implying mean square convergence.*

Having stated the theorems, we shall now present the proofs of Theorems 3 and 4.

Proof: Our aim is to prove that as the index t is increased indefinitely, the expected value of the $p_i(t)$ converges towards s_i, for all $1 \leq i \leq r$ implying that: $\lim_{t\to\infty} E[p_i(t)] \to s_i$.

The proof is analogous to the proof of the binomial case.

Let us consider $p_i(t)$ for the index i throughout the argument. Indeed, we shall prove the above by analyzing the properties of the underlying Markov chain associated to $p_i(t)$, which is specified by the rules (11), (12) and (13).

In brief, rules (11), (12) and (13) obey the Markov chain with transition matrix H^i. Let us consider $p_i(t)$, in order to specify the exact expression of H^i, *its* Markov chain. Consider the transitions that the discretized running estimate $p_i(t)$ exhibits. Note that the transitions happen between adjacent states. We can easily remark that: $h^i_{j,j+1} = s_i(1 - \frac{j}{N})$, $0 \leq j < N$.

The transition from state j to $j-1$, for $0 < j \leq N$, is more complicated to compute. The latter happens with a certain probability at time t if $x(t) =$ "k" takes place, for $i \neq k$, and $p_k(t)$ is increased at time instant t. In this sense, if $p_k(t)$ is incremented at time t, then $p_i(t)$, $i \neq k$, will be *possibly* decremented with a probability equal to $\frac{p_i(t)}{\sum_{\substack{l=1 \\ l\neq k}}^r p_l(t)}$ which corresponds to the normalized probability of the selection of $p_i(t)$ among $\{p_l(t), l \neq k\}$.

In the following, we show that Markov chain is time-homogeneous by demonstrating that the terms of the transition matrix H^i do not depend on time! In addition, we shall show that the matrix H^i does not depend on the state of random walk $p_j(t)$ described by H^j, for $i \neq j$. In other words, the transitions of the random walk attached to $p_i(t)$ are "*decoupled*" from those of $p_j(t)$, for $i \neq j$.

Therefore:

$h^i_{j,j-1} = \sum_{\substack{k=1 \\ k\neq i}}^r s_k(1 - p_k)\frac{p_i}{\sum_{\substack{l=1 \\ l\neq k}}^r p_l}, 0 < j \leq N.$

We note that, $\sum_{\substack{l=1 \\ l\neq k}}^r p_l = 1 - p_k$. Moreover, whenever the random walker i is in state j, $p_i = \frac{j}{N}$. Therefore $h^i_{j,j-1}$, for $0 < j \leq N$ is expressed as:

$h^i_{j,j-1} = \sum_{\substack{k=1 \\ k\neq i}}^r s_k(1 - p_k)\frac{\frac{j}{N}}{\sum_{\substack{l=1 \\ l\neq k}}^r p_l} = \sum_{\substack{k=1 \\ k\neq i}}^r s_k \frac{j}{N}.$

We note too that, $\sum_{\substack{k=1 \\ k\neq i}}^r s_k = 1 - s_i$.

Therefore, $h^i_{j,j-1} = (1 - s_i)\frac{j}{N}$.

We now resume our argument by seeing that H^i is defined by:

$h^i_{j,j+1} = s_i(1 - \frac{j}{N})$, $0 \leq j < N$,

$h^i_{j,j-1} = (1 - s_i)\frac{j}{N}$, $0 < j \leq N$,

$h^i_{j,j} = 1 - h_{j,j-1} - h_{j,j+1}$, $0 < j < N$,

and, accordingly

$h^i_{0,0} = 1 - h^i_{0,1}$

$h^i_{N,N} = 1 - h^i_{N,N-1}$.

Therefore, H^i does not depend on time.

We shall now compute π^i_k the stationary (or equilibrium) probability of the chain being in state k. Clearly H^i represents a single closed communicating class whose periodicity is unity. The chain is ergodic, and the limiting probability vector is given by the eigenvector of ${H^i}^T$ corresponding to the eigenvalue unity.

Let Π^i denote the vector of steady state probabilities. This vector, $\Pi^i = [\pi^i_1, \ldots, \pi^i_N]^T$, can be computed using ${H^i}^T \Pi^i = \Pi^i$.

Using the results from the binomial case, we can easily deduce that:

$\pi^i_k = \binom{N}{k} s_i^k (1 - s_i)^{N-k}$.

Let X^{i*} be the limiting index of the state of the random walker i. From the above, clearly X^{i*} is binomially distributed, with parameters s_i and N.

Hence, $E[X^{i*}] = Ns_i$ and $var[X^{i*}] = Ns_i(1 - s_i)$.

Consequently, $E[p_i(\infty)] = \frac{1}{N}E[X^{i*}] = s_i$ and $var[p_i(\infty)] = \frac{1}{N^2}var[X^{i*}] = \frac{1}{N}s_i(1 - s_i)$.

Note too that, amazingly enough, the convergence does not involve the number of states.

Thus, Theorem 3 is proved. Using the fact that $\lim_{N\to\infty} var[p_i(\infty)] = \lim_{N\to\infty} \frac{1}{N}s_i(1 - s_i) = 0$, we prove Theorem 4.

Remarks:

A few remarks regarding our method for updating the estimates are not out of place. Indeed:

- The philosophy behind the form of the update defined by Equations (11), (12) and (13) is to, eventually, increase one component of the vector of running estimates by a quantity equal to the stepsize Δ while decreasing another component by the same quantity Δ. This is done in order to ensure the sum of the components equals unity.

- The SDWE makes at most 2 updates per time step, thus rendering the worst case to be constant or $O(1)$. Interestingly, the estimator possesses a low computational complexity that is independent of the number of the parameters of the multinomial distribution.

- It is pertinent to mention that although the rationale for updating is similar to that used for the Stochastic Nonlinear Fractional Knapsack algorithm [4], there are fundamental differences. Unlike the latter, where the LA was made *artificially* ergodic by excluding the end-states from the set of possible probability values, our estimator is truly ergodic[1]. The reason for excluding the the end states in [4] was because the LA in [4] uses its "estimates" to select "actions" with frequencies proportional to the estimates. Consequently, enforcing an absorbing condition for the end states in [4] would imply always selecting the same action from that time instant onwards, and never selecting the alternate one again. As opposed to this, our current automaton does not perform/choose actions (events) with frequencies proportional to the estimate vector. Rather, since this present LA is intended for estimation, it is, indeed, the environment which "produces" events independent of the LA, and which informs the LA about these events. The LA then, in turn, estimates S based on the observed events. In other words, the difference between the two LA is that the one in [4] performs actions in the environment in order to maximize the number of rewards it receives, while the present LA observes the environment to estimate *its* properties.

- Apart from the above, it is pertinent to mention that the purpose of introducing the hierarchy in the knapsack problem was, first and foremost, to achieve better scalability when it concerns the learning speed. The hierarchy proposed in [4] improved the learning speed dramatically compared to the approach presented in [3] because of the hierarchical configuration of the underlying LA. The question of introducing a hierarchical philosophy in the setting of the current paper remains unsolved.

- One of the most significant contributions of the paper, in our opinion, is the rather elegant manner by which we have designed the RW for the estimation process in the multinomial case. By intelligently designing the RW transitions in the discretized probability space, we have succeeded in deriving and analyzing the multinomial estimates in a manner identical to what was involved for solving the binomial estimation case. To further explain this, we mention that if the transitions of the RW associated with p_i were directly dependent on the RW associated with p_j for $i \neq j$, the state space would have been unmanageable, namely of the order of $O(N^2)$. However, the effect of our specific design and representation helped in reducing the magnitude of the state space of the RW for the multinomial case from $O(N^2)$ to $O(N)$. This further assisted in *decoupling* the transition matrices for each component of the multinomial vector, which in fact, is an elegant alternative to the concept of using a team of *normalized* binomial estimators for solving multionomial estimation problems[2]. In addition,

[1]If the LA in the present paper would have observed the event "i" with probability p_i instead of probability s_i (for $i = 1, 2$), the present LA would have also been absorbing!

[2]The reader can easily see that this is also an alternate, although more cumbersome, way to decouple the updates

we believe that the unique RW transitions used here for multinomial estimates, can inspire the design of novel ways for generalizing two-action learning machines into multiaction machines!

6 Experimental Results

In this section, we evaluate the new family of discretzied estimators in non-stationary environments and compare this approach to traditional Maximum Likelihood estimation methods which use the sliding window (MLEW) and the SLWE. In order to confirm the superiority of our scheme, we have conducted extensive simulations results under different parameter settings. In the interest of brevity, we merely cite a few specific experimental results.

6.1 Comparison with the MLEW for the Case of Binomial Random Variables

The estimation of the parameters for binomial random variables has been extensively tested for numerous binomial distributions. However, in the interest of brevity, we include only the results from two of these experiments here. We adopt the same simulation methodology as in [15], using randomly generated values of N. To assess the efficiency of the estimation methods in a fair way, we randomly chose values for the resolution N and for the window width from the intervals $[6, 16]$ and $[20, 80]$, respectively. To get a *smooth* result, we performed ensembles of $1,000$ simulations each consisting of 400 time steps. The true underlying value of was randomly changed every 50 time steps. The plots from each experiment are presented in Figures 1 and 2 where the values for N are 6 and 8 respectively, and the sizes of the windows are 32 and 57, respectively. The results we show here are typical. The reason for operating with larger window sizes than 50 is that we generally have no knowledge about the environment with regard to the frequency or magnitude of the changes. We observe that when the window size of the MLEW becomes larger, e.g., 57, the estimation algorithm is unable to track the changes in the environment, yielding poor overall accuracy. In Figure 2, we see that the MLEW approximates the true value fairly well for the first 50 time steps, but is thereafter unable to track the variations. In Figure 1, when the window size is smaller, the MLEW is capable of tracking the changes, but not nearly as fast nor as accurate as the SDWE. Even when the SDWE uses a resolution parameter as low as 6, it clearly outperforms the MLEW with regard to the convergence speed.

of the different components of vector of multinomial estimates.

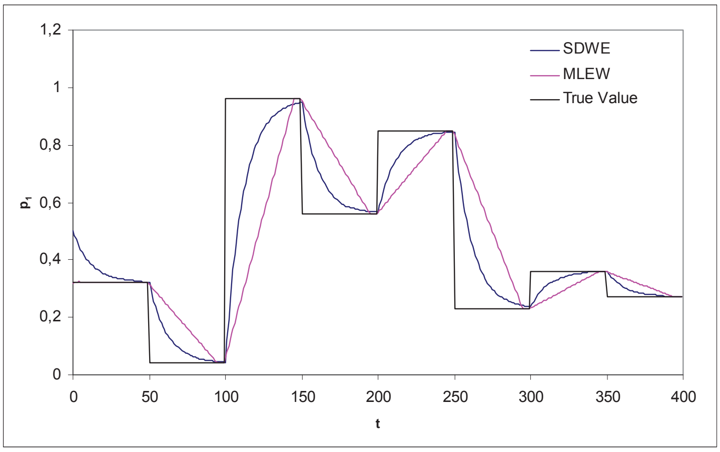


Figure 1: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and MLEW, using $N = 6$ and $w = 29$.

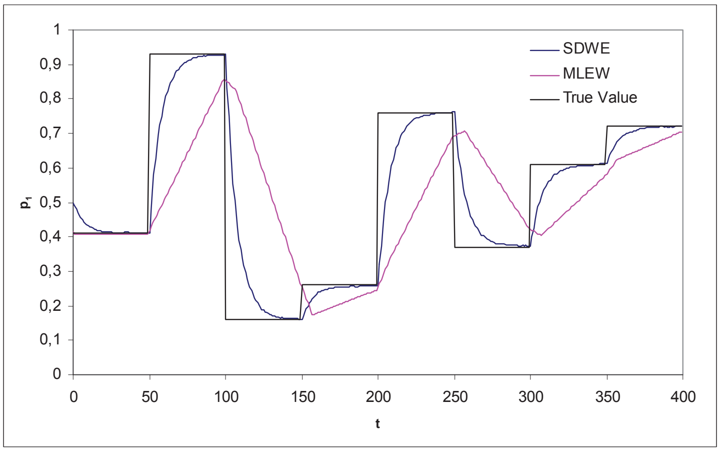


Figure 2: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and MLEW, using $N = 8$ and $w = 57$.

6.2 Comparison with MLWE for the Case of Multinomial Random Variables

We have also performed simulations for multinomial random variables, where the parameters were estimated by following the SDWE and the MLEW. We considered a multinomial random variable, X, which can take any of three different values, namely 1, 2, or 3, whose probability values change (randomly) every 50 steps. As in the binomial case, we ran the estimators for 400 steps, repeated this 1,000 times, and then took the ensemble average of P. We computed $||P - S||$, the Euclidean distance between P and S, which was intended to be a measure of how good our estimate, P, was of S. The plots of the latter distance obtained from the SDWE and the MLEW are depicted in Figures 3 and 4, where the values for N are 6 and 12, and the sizes of the windows are 67 and 34, respectively. The values for N and the window size were obtained randomly from a uniform distribution in $[6, 15]$ and $[20, 80]$.

It is clear from both figures that the SDWE is faster in tracking the changes than the MLEW. Note that the MLEW converges faster than the SDWE only during the first 50 time instants (before the first environment switch). However, this behavior is not present in successive epochs. In Figure 3, we remark that a window size as large as 67 slows down the convergence speed of the MLEW. This is due to the fact that the larger the window size, the greater is the effect of the stale data to jeopardize the running estimate. From both Figures 3 and 4, we observe that a large value of N yields low variance, i.e., the accuracy is high. The problem is that the rate of convergence is slower than when we are using a lower value of N. A low value of N results in a faster convergence, but it yields a higher variance from the true underlying parameter. This confirms that the choice of the user-defined learning parameter, N, reduces to a trade off between the speed and the corresponding accuracy. Similar results are observed for the window width parameter of the MLEW as well. We notice that the MLEW is capable of tracking the changes of the parameters when the size of the window is small, or at least smaller than the intervals of constant probabilities. The latter, however, is not able to track the changes properly when the window size is relatively large. Since neither the magnitude nor the frequency of the changes is known *a priori*, this scenario demonstrates the weakness of the MLEW, and its dependence on the knowledge of the input parameters. Such observations are typical.

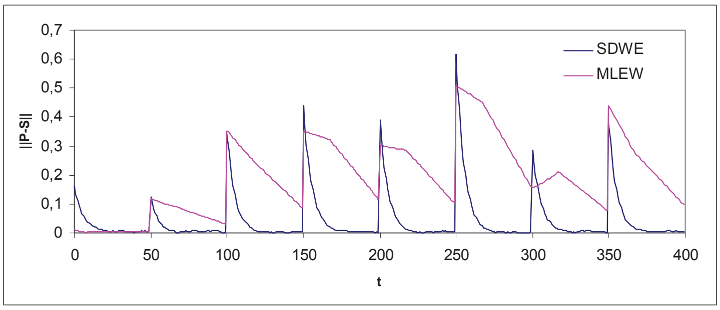


Figure 3: Plot of the Euclidean norm P-S (or the Euclidean distance between P and S), for both the SDWE and MLEW, where N is 6 and the window size is 67 respectively.

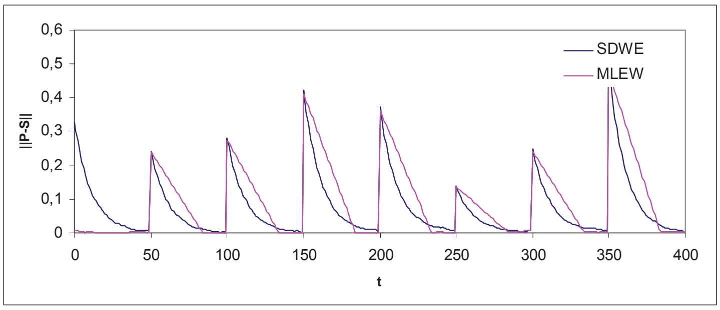


Figure 4: Plot of the Euclidean norm P-S (or the Euclidean distance between P and S), for both the SDWE and MLEW, where N is 12 and the window size is 34 respectively.

6.3 Comparison with SLWE for the case of Binomial random variables

We considered a binomial random variable, X, which can take any of two different values, namely 1, 2 where the probability changes (randomly) every 50 steps. In order to compare the performance SDWE and the performance of the SLWE in a fair way, we adopted the same method as in the case of the SDWE and the MLEW reported in [15]. We randomly chose values for the resolution N and for the parameter λ of the SLWE [15] from the intervals $[8, 16]$ and $[0.9, 1]$, respectively. It was reported in [15] that using values of λ for the SLWE drawn from $[0.9, 1]$ yields fast convergence speed and good accuracy. Again, we conducted an ensemble of $1,000$ simulations, each consisting of 400 time steps. In the interest of brevity, we include only the results from two of these experiments here. The expected value of the estimates of $p_1(t)$, obtained from the SDWE and SLWE, using $N = 8$ and $\lambda = 0.938$ is depicted in Figure 5. Similarly, the estimates of $p_1(t)$, obtained from the SDWE and SLWE, using $N = 12$ and $\lambda = 0.9623$ is depicted in Figure 6. Clearly, the SDWE is significantly faster than the SLWE. This is parallel to the results known in the field of LA where discretized LA have been reported to outperform their continuous counterparts.

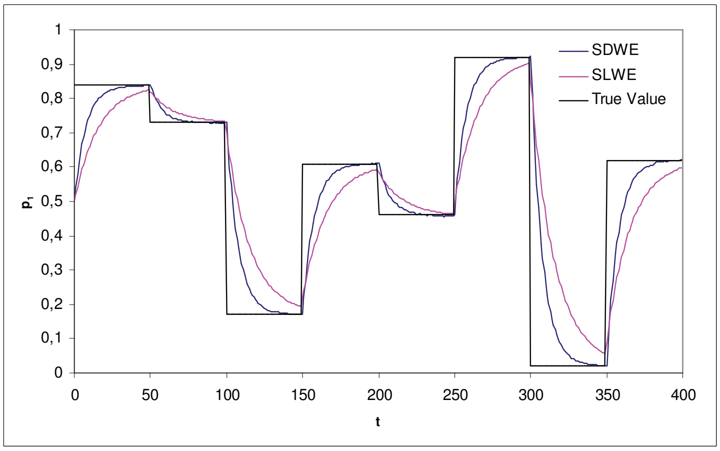


Figure 5: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and SLWE, using $N = 8$ and $\lambda = 0.938$ respectively.

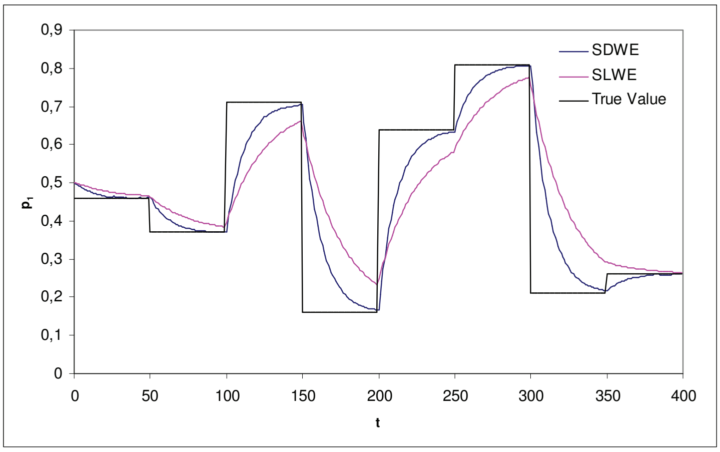


Figure 6: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and SLWE, using $N = 12$ and $\lambda = 0.9623$ respectively.

6.4 Comparison with SLWE for the Case of Multinomial Random Variables

For the multinomial case, we again performed ensembles of $1,000$ simulations each consisting of 400 time steps. The true underlying value of S was randomly changed every 50 time steps. We computed $||P - S||$, the Euclidean distance between P and S, and include here only the results from two experiments. We randomly chose values for the resolution N and for the parameter λ of the SLWE [15] from the intervals $[6, 15]$ and $[0.9, 1]$, respectively. Figure 7 illustrates the case of $N = 12$ for the SDWE and $\lambda = 0.931$ for the SLWE, whereas Figure 8 depicts the results for $N = 9$ and $\lambda = 0.942$.

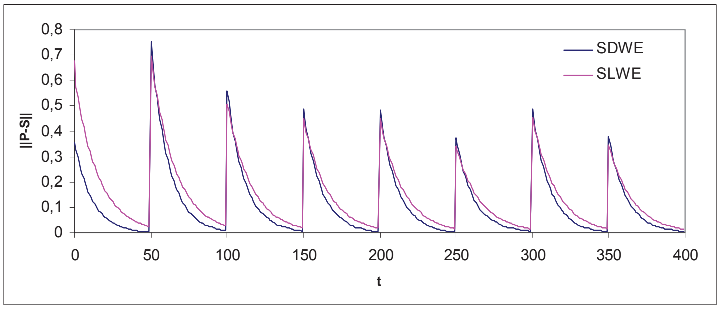


Figure 7: Plot of the Euclidean norm P-S (or the Euclidean distance between P and S), for both the SDWE and SLWE, where N is 12 and the $\lambda = 0.931$ respectively.

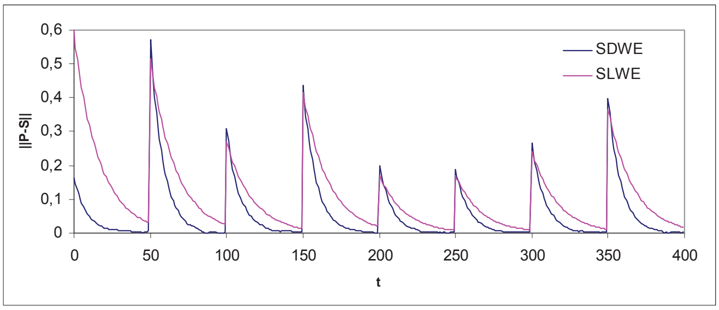


Figure 8: Plot of the Euclidean norm P-S (or the Euclidean distance between P and S), for both the SDWE and SLWE, where N is 9 and $\lambda = 0.942$ respectively.

From both figures 7 and 8, we observe that the SDWE outperforms the SLWE in the multinonmial case both in speed and accuracy. In the case of the SLWE, we note that a small value for λ, yields less accuracy, but faster convergence. Similarly, for the SDWE, large values of N yield more accuracy but slower convergence speed.

7 Conclusion

This paper has presented a novel discretized estimator that is able to cope with non-stationary binomial/multinomial distributions using finite memory. To the best of our knowledge, our SDWE is the first reported discretized counterpart of the SLWE. Through this paper, it was shown that discretizing the probability space offers a new promising approach for the design of weak estimators. In fact, comprehensive simulation results demonstrate that the new estimator is able to cope with non-stationary environments with both a high adaptation rate and accuracy. In addition, the results suggest that the SDWE outperforms the MLEW as well as the SLWE. As a future work, we propose to study the performance of the estimator in real-life applications. Indeed, the possible application of the SDWE for classification and language detection is *currently* being investigated.

References

[1] M. Agache and B. J. Oommen, "Generalized pursuit learning schemes: New families of continuous and discretized learning automata," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, no. 6, pp. 738–749, December 2002.

[2] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes: theory and application.* Prentice-Hall, Inc., 1993.

[3] O.-C. Granmo, B. Oommen, S. Myrer, and M. Olsen, "Learning automata-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 1, pp. 166 –175, feb. 2007.

[4] O.-C. Granmo and B. J. Oommen, "Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata," *IEEE Transactions on Computers*, vol. 59, pp. 545–560, 2009.

[5] I. Koychev, "Gradual forgetting for adaptation to concept drift," in *Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning*, 2000, pp. 101–106.

[6] I. Koychev and R. Lothian, "Tracking drifting concepts by time window optimisation," in *Research and Development in Intelligent Systems XXII*, M. Bramer, F. Coenen, and T. Allen, Eds. Springer London, 2006, pp. 46–59.

[7] I. Koychev and I. Schwab, "Adaptation to drifting user's interests," in *Proceedings of ECML2000 Workshop: Machine Learning in New Information Age*, 2000, pp. 39–46.

[8] J. K. Lanctôt and B. J. Oommen, "Discretized estimator learning automata," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-22, no. 6, pp. 1473–1483, November/December 1992.

[9] M. May, B. Berendt, A. Cornujols, J. Gama, F. Giannotti, A. Hotho, D. Malerba, E. Menesalvas, K. Morik, R. Pedersen, L. Saitta, Y. Saygin, A. Schuster, and K. Vanhoof, "Research challenges in ubiquitous knowledge discovery," in *Next Generation of Data Mining (Chapman Hall Crc Data Mining and Knowledge Discovery Series)*, 1st ed. Chapman Hall CRC, 2008.

[10] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction.* New Jersey: Prentice-Hall, 1989.

[11] B. J. Oommen, "Absorbing and ergodic discretized two-action learning automata," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-16, pp. 282–293, March/April 1986.

[12] B. J. Oommen and M. Agache, “Continuous and discretized pursuit learning schemes: Various algorithms and their comparison,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 31, pp. 277–287, 2001.

[13] B. J. Oommen and E. Hansen, “The asymptotic optimality of discretized linear reward-inaction learning automata,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-14, no. 3, May/June 1986.

[14] B. J. Oommen and S. Misra, “Fault-tolerant routing in adversarial mobile ad hoc networks: an efficient route estimation scheme for non-stationary environments,” *Telecommunication Systems*, vol. 44, pp. 159–169, 2010.

[15] B. J. Oommen and L. Rueda, “Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments,” *Pattern Recogn.*, vol. 39, no. 3, pp. 328–341, 2006.

[16] L. Rueda and B. J. Oommen, “Stochastic automata-based estimators for adaptively compressing files with nonstationary distributions,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 5, pp. 1196 –1200, 2006.

[17] A. N. Shiryayev, *Optimal Stopping Rules*. Springer, 1978.

[18] A. Stensby, B. J. Oommen, and O.-C. Granmo, “Language detection and tracking in multilingual documents using weak estimators,” in *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 600–609.

[19] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blazek, and H. Kim, “A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods,” *IEEE Transactions on Signal Processing*, vol. 54, pp. 3372–3382, 2006.

[20] M. A. L. Thathachar and B. J. Oommen, “Discretized reward-inaction learning automata,” *Journal of Cybernetics and Information Science*, pp. 24–29, Spring 1979.

[21] M. A. L. Thathachar and P. S. Sastry, “Varieties of learning automata: An overview,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 32, no. 6, pp. 711–722, December 2002.

[22] ——, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Boston: Kluwer Academic, 2003.

[23] J. Zhan, B. J. Oommen, and J. Crisostomo, "Anomaly detection in dynamic systems using weak estimators," *ACM Trans. Internet Technol.*, vol. 11, pp. 3:1–3:16, July 2011.