Stochastic Discretized Learning-based Weak Estimation: A Novel Estimation Method for Non-Stationary Environments[*]

Anis Yazidi[†], B. John Oommen[‡], Geir Horn[§], 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. 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 discretized 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. Further, the state transitions are both *state-dependent* and *randomized*. As far as we know, such a scheme is both novel and pioneering. The results which have first been proven for binomial distributions have subsequently been extended for the multinomial case, using which they can be applied to any univariate distribution using a histogram-based scheme. The most outstanding and pioneering contribution of our work is that of achieving multinomial estimation without relying on a set of binomial estimators, and where the underlying strategy is truly randomized. Interestingly, the estimator possesses a low computational complexity that is independent of the number of parameters of the multinomial distribution. The generalization of these results for other distributions has also been alluded to.

[*] The first author gratefully acknowledges the partial 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 very grateful to the the Associate Editor and Referees of the paper for their valuable comments. They have improved the paper significantly. A very preliminary version containing a few of the initial results found in this paper was presented at ICNC'12, the 2012 International Conference on Computing, Networking and Communications, Hawaii, USA, in February 2012.

[†] Author's status: *Associate Professor*. This author can be contacted at: Oslo and Akershus University College, Department of Computer Science, Pilestredet 35, Oslo, Norway. The work was partially done while he was working as a Senior Researcher with Teknova AS, Sorlandet Kunnskapspark, Gimlemoen 19, 4630 Kristiansand, Norway. E-mail: Anis.Yazidi@hioa.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 be contacted at: University of Oslo, Faculty of Mathematics and Natural Sciences, Ole-Johan Dahls hus. P.O. Box 1080 Blindern, Oslo, Norway. E-mail address: Geir.Horn@mn.uio.no.

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

The paper briefly reports conclusive experimental results that prove the ability of the SDWE to cope with non-stationary environments with high adaptation 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 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, within this premise, 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 a binomially-distributed random variable were done in a time-varying manner, where the parameter *switched*, for example, periodically, to possibly, a new random value. Such a scenario exhibits the behavior of a *non-stationary* environment. Consequently, in this setting, 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 to estimate the new parameter after a *switch* has occurred in the environment, and hopefully, this must be done "reasonably quickly". 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, in the vein of the known results for non-stationary environments (surveyed presently), 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 to also "*unlearn*" what it has learned so far so as 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 a 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 [33], spam filtering [41], ubiquitous computing [20], fault tolerant routing [25], adaptive encoding [29], and topic detection and tracking in multilingual online discussions [32]. We submit that the DSWE can be used expediently in all of these application domains. Further, such an estimation scheme opens avenues towards a large set of real life applications, as in tracking changing user preferences [26] and variations in ubiquitous environments, and as an integral component of a hybrid framework applicable for service recommendation systems [39]. Indeed, we also are currently investigating the application of our estimator to inferring drifts in other user-interest domains.

2 Related Work and 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[1].

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 [31], Shiryayev used a Bayesian approach to detect changes in the parameter's distribution, where the change points were assumed to obey a geometric distribution. CUMSUM is motivated by a maximum likelihood ratio test for the hypothesis 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 [14, 16, 17], 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 [17] 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.

[1] Enhancements to this philosophy will be discussed later in this section.

As opposed to the above-mentioned papers, in the last decades, a wide range of techniques for estimation in dynamically changing environments have appeared. We provide here a brief overview of representative *on-line* approaches particularly relevant to the family of techniques pioneered in the present paper. For a more detailed and comprehensive treatment of these, we refer the reader to recent surveys [8, 18], which include approaches based on *adaptive windowing*, *aging factors*, *instance selection* and *instance weighting*. Additionally, there is a distinction between *passive* approaches, which intrinsically adapt to changes, and *active* approaches that actively search for changes.

With regard to categorizing these approaches, they differ in terms of their memory management and forgetting mechanisms [8]. A memory management module aims to control the data points that are to be kept in memory for learning. As opposed to this, *forgetting* refers to the task of erasing previously-learned information. One data management strategy, referred to as "instance selection", is based on storing instances relevant to the current situation. Here, the simplest approach is to only keep a single data point in memory, with the task of forgetting being modeled by the gradual dilution of its relevance through adaptation of the model's parameters [6]. Such a strategy can be expanded to store multiple data points, using a window of fixed size. The data points in the window are used to establish the current estimate, and at each time step, the oldest data point is replaced with the newest one, using a FIFO policy. Slow changes are best captured by a larger window, while rapid changes require a correspondingly smaller window. To address this conflict, an alternative scheme involves variable-sized sliding windows, referred to as *adaptive windowing* [38]. In brief, the overall strategy is to allow the window size to grow over time, and use a dedicated change detector to reset the window size when a change is detected, resulting in windows with an adaptive size. More advanced approaches carefully pick out prototypes, representing the concept being tracked [3], thus handling concept drift by replacing the prototypes themselves to reflect changes.

Another strategy, which avoids windowing, involves using *aging* factors. In this approach, while training data spans all of the data points, each data point is assigned a weight that reflects its importance. Recent data points are prioritized, for instance using a fading factor that gradually reduces the influence of older data points [4]. The decay can be linear [15] or exponential [13]. By using weights combined with instance selection, referred to as instance weighting, one can take advantage of the ability of support vector machines to process weighted instances [13].

Most of the above approaches are passive in the sense that they gradually adjust to changes, without explicitly pinpointing them. In contrast, another class of approaches actively searches for changes using change detectors. One solution is to maintain an ensemble of estimators, combined using voting, such as the dynamic integration of classifiers found in [37]. The impact of each estimator is based on assessing them on recent data, prioritizing the one that produces the least error, where the change of priority indicates an underlying change in the distribution of data points. One can also monitor estimation error using various mechanisms, or the raw data itself. Changes in estimation error signals a change in the distribution of data points, triggering re-estimation [28]. In all brevity, changes are detected based on comparing sections of data, using statistical analysis to detect distributional changes, i.e., abrupt or gradual changes in the mean of the data points when compared with a baseline mean with a random noise component. One option is also to keep a reference window and compare recent windows with the reference window to detect changes [7]. This can, for example, be done based on comparing the probability distributions of the reference window and the recent window using Kullback-Leibler divergence [5, 30].

Using a completely different tool-set, Oommen and Rueda [27] 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, 19, 24, 34], 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 their continuous counterparts [1, 23]. By virtue of discretization, our estimator realizes fast adjustments of the running estimates by performing "jumps", and it is thus able to robustly and quickly track changes in the parameters of the distribution after a switch has occurred in the environment. Historically, the concept of discretizing the probability space was pioneered by Thathachar and Oommen in their study on Reward-Inaction LA [34], and since then, it has catalyzed a significant research in the design of discretized LA [1, 9, 10, 19, 24]. Recently, there has been an upsurge of research interest in solving resource allocation problems based on novel discretized LA [9, 10], in which 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 thereafter applied to resolve the web-polling problem, and to the problem of determining the optimal size required for training a classifier. Our present paper is a distant relative of these schemes.

To the best of our knowledge, our SDWE is the first reported discretized[2] counterpart of the continuous SLWE [27]. The numerous successful applications of the continuous SLWE and the solution to stochastic point location using, for example, a hierarchical [40] strategy, reported in [25, 29, 32, 41] motivates our SDWE, and elucidates its relevance for tracking non-stationary distributions in real-life problems.

[2] We emphasize that the main thrust of this paper is not to merely demonstrate the superiority of the SDWE to the SLWE etc. Rather, we submit that the entire phenomenon of utilizing the concepts of discretization in weak estimation is unexplored, and it is precisely here that we have our primary contributions.

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 summarized 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, for updating the vector of estimates. 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 Ergodic and Discretized Properties of the SDWE

Our devised SDWE is based on the theory of LA [21, 36], 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. The latter automata get locked into a barrier state after a finite number of iterations. Many families of automata that posses absorbing barriers have been reported [21]. Ergodic automata have also been investigated in [21, 23]. 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 the reasons why our LA-based scheme is ergodic.

Moving now to a disjoint vein, with respect to the values that the action probabilities can take, the families of LA typically fall into one of the two categories, namely, Continuous or 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 [23, 34]. 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 Variable Structure Stochastic Automata (VSSA) have been discretized; indeed, discretized versions of almost all continuous automata have been reported [22, 23]. Families of Pursuit and Estimator-based LA have been shown to be faster than VSSA [35]. As a matter of a fact, even faster discretized versions of these schemes have been reported [1, 23].

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 A Direct Estimator for Binomial Distributions

In this section and the next, we submit the theoretical foundations on which the SDWE is developed. First of all, in Section 4.2, we shall formally prove that the so-called direct (or naively discretized) solution will not work. Then, in Section 5, we present a more-elaborate state-dependent RW-based solution that will, indeed, yield accurate results in terms of the convergence, in expectation, to the correct underlying mean. We also explain the philosophy behind the design of the estimator based on the analogy with *birth-death* processes. The resulting discrete estimator can therefore, truly, be perceived as a completely novel estimator.

4.1 Definitions

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 symbols instead of the more commonly 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, and

$X =$ "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 $p_i(t)$ *only* 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 the Resolution parameter, and determines the stepsize Δ (where $\Delta = 1/N$) relevant to the updating of the estimates.

4.2 The Direct Solution

In the continuous version of the SLWE pioneered by Oommen *et al* [27], the updating rule of the estimate is given by the following rule:

$$p_1(t+1) = \begin{cases} 1 - \lambda\,(1 - p_1(t)) & \text{if } x(t) = \text{“1”} \\ \lambda p_1(t) & \text{if } x(t) = \text{“2”}. \end{cases} \qquad (1)$$

The continuous SLWE avoids the maximum-likelihood estimation principle and is thus able to cope with non-stationary environments. As a starting point, to motivate the results of this paper, we shall consider the issue of whether we can use the analogous principle of updating the probability vector (*albeit*, in a discretized manner) to achieve the same result. From a philosophical perspective, one would imagine that by updating the probability vector in an identical dsicretized manner and using these probabilities to estimate S, one could obtain a reasonable estimate of S. This is more plausible because it turns out that the updating rule parallels a maximum likelihood mechanism in which the state index can be used to count the number of occurrences of the event. However, ironically, unlike the maximum-likelihood scheme, the probability estimated by such a mechanism does not converge to the true (unknown) underlying probability. This result is quite astonishing, but it really is quite straightforward if one considers the properties of the underlying Markov Chain.

Based on the latter updating rules, the obvious approach analogous to the SLWE for updating the estimate would be to increase the estimate by the step size if $x(t) =$ "1" and decrease the estimate by the step size if $x(t) =$ "2". Such a strategy would lead to the following rules:

$$p_1(t+1) = \begin{cases} p_1(t) + \frac{1}{N} & \text{if } 0 \le p_1(t) < 1 \text{ and } x(t) = \text{“1”}\ , \\ p_1(t) - \frac{1}{N} & \text{if } 0 < p_1(t) \le 1 \text{ and } x(t) = \text{“2”} \\ p_1(t) & \text{Otherwise.} \end{cases} \qquad (2)$$

The Markov chain for the above-specified updating scheme has $N+1$ states $k(t) \in \{0, \ldots, N\}$ estimating the underlying probability for observing $x(t) =$ "1", s_1, as $p_1(t) = \frac{k(t)}{N}$. This Markov chain is depicted in Figure 1. We now state the following foundational result that $P(n)$ does not converge to $S(n)$, and it is proven below.

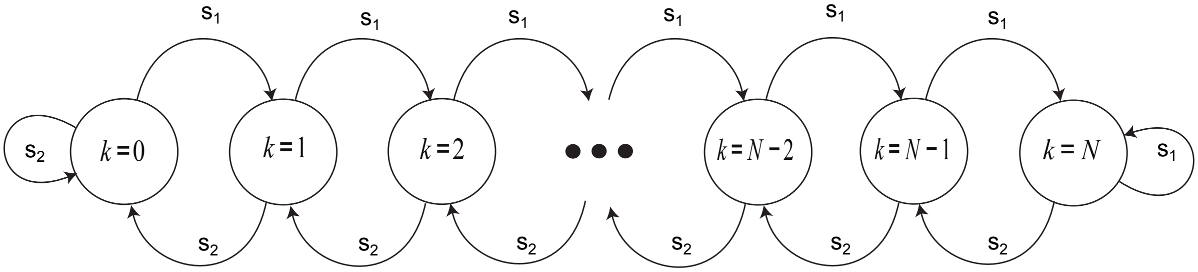


Figure 1: The Markov chain of the direct mapping of the SLWE onto a discretized space.

Theorem 1 *The stationary distribution defined by the Markov chain describe by Figure 1 is:*

$$\pi_s = \left(\frac{s_1}{s_2}\right)^s \pi_0 \qquad 0 \leq s \leq N, \tag{3}$$

where π_0 is a normalizing constant such that $\sum_{s=0}^{N} \pi_s = 1$, and is given by:

$$\pi_0 = \frac{1 - \rho/(1-\rho)}{1 - [\rho/(1-\rho)]^{N+1}}. \tag{4}$$

Proof: Since the underlying probabilities are s_1 and s_2, one can perceive that the stationary state probabilities π_s of the Markov chain in Figure 1 must obey the following recurrence relation for the non-extremal states:

$$\pi_s = s_1 \pi_{s-1} + s_2 \pi_{s+1}. \tag{5}$$

Equation (3) is proven by induction. Consider first the basic case for π_0 by setting $s = 0$. Then, if we use the simplifying notation that $\rho = s_1$ (and $(1-\rho) = s_2$), and if we substitute the solution into the recurrence relation (5), we get:

$$\begin{aligned}
\pi_s &= \rho\pi_{s-1} + (1-\rho)\pi_{s+1} \\
&= \rho\left(\frac{\rho}{1-\rho}\right)^{s-1}\pi_0 + (1-\rho)\left(\frac{\rho}{1-\rho}\right)^{s+1}\pi_0 \\
&= \frac{\rho^s}{(1-\rho)^{s-1}}\pi_0 + \frac{\rho^{s+1}}{(1-\rho)^{s+1}}\pi_0 - \frac{\rho^{s+2}}{(1-\rho)^{s+1}}\pi_0 \\
&= \frac{\pi_0}{(1-\rho)^{s+1}}\left[\rho^s(1-\rho)^2 + \rho^{s+1} - \rho^{s+2}\right] \\
&= \frac{\pi_0}{(1-\rho)^{s+1}}\left[\rho^s\left(1-2\rho+\rho^2\right) + \rho^{s+1} - \rho^{s+2}\right] \\
&= \frac{\pi_0}{(1-\rho)^{s+1}}\left[\rho^s - 2\rho^{s+1} + \rho^{s+2} + \rho^{s+1} - \rho^{s+2}\right] \\
&= \frac{\rho^s(1-\rho)}{(1-\rho)^{s+1}}\pi_0 = \frac{\rho^s}{(1-\rho)^s}\pi_0 \\
&= \left(\frac{\rho}{1-\rho}\right)^s\pi_0,
\end{aligned}$$

which shows that the equations are accurate for the internal states. The stationary probability of the extreme state must be computed directly from the recurrence relation for the final state as this is different from the relation for the internal states. To do this, we observe that:

$$\pi_N = \rho\pi_{N-1} + \rho\pi_N.$$

Now sorting the state probabilities, we get:

$$\pi_N - \rho\pi_N = \rho\pi_{N-1}, \text{ whence } (1-\rho)\pi_N = \rho\pi_{N-1}.$$

Simplifying the above (after re-arranging), we see that:

$$\begin{aligned}
\pi_N &= \left(\frac{\rho}{1-\rho}\right)\pi_{N-1} = \left(\frac{\rho}{1-\rho}\right)\left[\left(\frac{\rho}{1-\rho}\right)^{N-1}\pi_0\right] \\
&= \left(\frac{\rho}{1-\rho}\right)^N\pi_0.
\end{aligned}$$

To obtain the normalizing constant, let $a = \rho/(1-\rho)$. Then the sum of π's over all states is

$$\sum_{s=0}^{N}\pi_s = \sum_{s=0}^{N}a^s\pi_0 = \pi_0\sum_{s=0}^{N}a^s$$

Thus, from the formula for the sum of a finite power series,

$$= \pi_0\left[\frac{1-a^{N+1}}{1-a}\right]$$

Note that the the normal requirement that $|a| < 1$, which is necessary for the convergence of an infinite power series sums, is not needed for *finite* sums. Since this sum must be unity for a probability distribution, the normalising constant can be found as

$$\pi_0 = \frac{1}{\dfrac{1-a^{N+1}}{1-a}} = \frac{1-a}{1-a^{N+1}} = \frac{1-\rho/(1-\rho)}{1-[\rho/(1-\rho)]^{N+1}}.$$

Hence the result.

Theorem 2 *Following the update rules given by Eq. (2), as $N \to \infty$, $E[P(\infty)]$ converges to $[1,0]^T$ if $s_1 > 1/2$, and to $[0,1]^T$ if $s_1 < 1/2$ independent of the true value of S.*

Proof: By observing that the asymptotic probability mass on the states follows a geometric distribution, the expected value of the distribution (i.e., the expected value for the estimate) is easily obtained for $\rho \neq 1/2$ as:

$$E[\hat{\rho}] = 1 - \frac{1}{1-a^{N+1}} + \frac{1}{N}\left(\frac{1}{1-a} - \frac{1}{1-a^{N+1}}\right), \tag{6}$$

where $a = \rho/(1-\rho)$.

The result follows directly from (6) since when $\rho < 1/2$, it implies that $a < 1$, and so as $N \to \infty$ the first two terms cancel, and the factor $1/N$ eventually takes the last term to zero independent of the true value of ρ. Similarly, if $\rho > 1/2$ then $a > 1$, and as N grows, the term a^{N+1} will grow faster than $1/N$ diminishes. Hence the expected estimate can quickly be approximated by $1-1/(N(a-1))$, which pushes the estimate towards unity as N increases indefinitely.

Remark: These counter intuitive effects clearly highlight the fact that it is not so straightforward to "discretize" a continuous scheme because the main problems with such a strategy for devising a discretized scheme is that the mean is asymptotically always at the boundary values. Interestingly, the work in [11] further demonstrates that even for finite values of N, the expected estimate will never be close to S. Indeed, even for moderate values of N, the mean will be biased towards the extreme values – basically rendering such a discretization methodology to be useless. This motivates the search for a completely different, non-trivial mechanism whereby one can obtain a discretized SLWE. This is the subject matter of the rest of the paper.

5 Design of the SDWE for the binomial case

In order to derive a discretized solution that really works, it is necessary to step back and revisit the motivation for using discretized automata, and in particular, the phenomenon that continuous automata have the property that they only approach the extreme probability values asymptotically.

This property has not been captured by the previous attempt on discretizing the SLWE which was discussed in the last section. This observation leads to the revelation that it must be increasingly difficult for the automaton to move to a more extreme state even as it gets closer to the boundary states represented by the indices zero or N. Consider the case where $s_1 \approx 1$. In such a case, the observations will primarily consist of a sequence of $x(\cdot) =$ "1", and although it should be harder for the automaton to move towards the final state, it seems incorrect to *reduce* the state at time t upon another observation of $x(t) =$ "1". The solution to this dilemma is to let the automaton keep the current state with increasing probability the closer the state is to either of the extreme states.

Stochastic processes of this type are typically categorized as *birth-death* processes in the literature, stemming from early studies of discrete population dynamics, where the growth depends on the number of individuals in the population [12]. Assuming that the individuals compete for food, an individual's risk of dying of starvation over an epoch of observations will increase with a growing population, whereas epoch survival will be almost assured when there are few individuals. In addition, one can safely assume that the birth rates depend on the population size. Thus, an equilibrium will be achieved when the birth rate balances the death rate for a given population size admitted by the availability of the food resources.

With this as a backdrop, at this juncture, we present the equations motivating the design of such a discretized estimator.

Initially, we assign $p_1(0) = p_2(0) = N/2$, where N is assumed to be an even integer. Thereafter, the value of $p_1(t)$, is updated as follows:

If $x(t) =$ "1" and $rand() \leq 1 - p_1(t)$ and $0 \leq p_1(t) < 1$

$$p_1(t+1) := p_1(t) + \frac{1}{N} \tag{7}$$

If $x(t) =$ "2" and $rand() \leq 1 - p_2(t)$ and $0 < p_1(t) \leq 1$

$$p_1(t+1) := p_1(t) - \frac{1}{N} \tag{8}$$

$$p_1(t+1) := p_1(t) \text{ Otherwise,} \tag{9}$$

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 Eqs. (7), (8) and (9). 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 variance.

Since the proofs of the theorems are rather intertwined, we shall prove them together.

Theorem 3 *Let X be a binomially distributed random variable, and $P(t)$ be the estimate of S at time t obtained by Eqs. (7), (8) and (9). Then $E[P(\infty)] = S$.*

Theorem 4 *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 3 and 4.

In Theorem 3, 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 given by Eqs. (7), (8) and (9). In brief, rules given by Eqs. (7), (8) and (9) obey the Markov chain with transition matrix $H = [h_{kl}]$, where.

$h_{k,k-1} = s_2\frac{k}{N}$, $0 < k \leq N$,

$h_{k,k+1} = s_1(1 - \frac{k}{N})$, $0 \leq k < N$,

$h_{k,k} = 1 - h_{k,k-1} - h_{k,k+1}$, $0 < k < N$,

$= (1 - \frac{k}{N}) + (2\frac{k}{N} - 1)s_1$, $0 < k < N$,

and, accordingly

$h_{0,0} = 1 - h_{0,1}$

$h_{N,N} = 1 - h_{N,N-1}$.

The associated Markov Chain[3] is depicted in Figure 2.

[3] In the population dynamics formulation of the problem, the following two concepts can be defined from the above definitions [12]: $Birthrate = s_1(1 - \frac{k}{N})$, and $Deathrate = s_2\frac{k}{N}$.

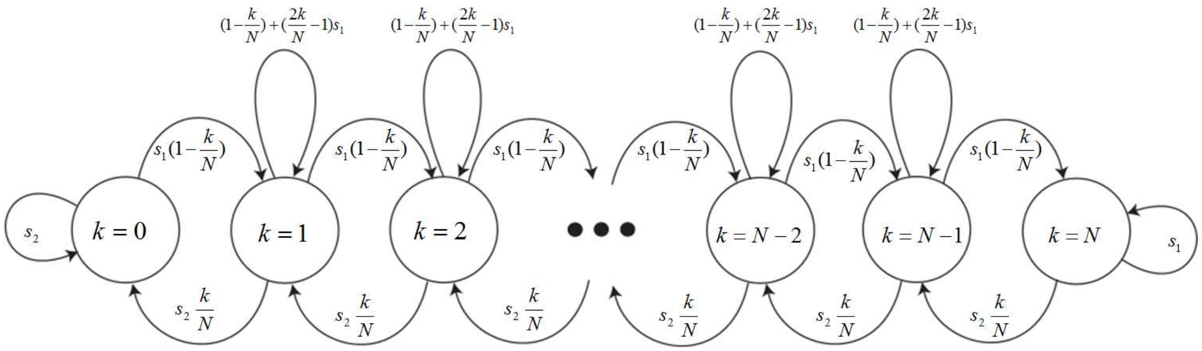


Figure 2: The Markov chain of the SDWE where k denotes the state index.

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:

$$\begin{bmatrix} h_{0,0} & h_{1,0} & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ h_{1,0} & h_{1,1} & h_{1,2} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & h_{2,1} & h_{2,2} & h_{2,3} & 0 & \cdot & \cdot & 0 \\ \vdots & \cdot & \ddots & \ddots & \ddots & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & h_{k,k-1} & h_{k,k} & h_{k,k+1} & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot & \cdot & h_{N-1,N-2} & h_{N-1,N-1} & h_{N-1,N} \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & h_{N,N-1} & h_{N,N} \end{bmatrix}^T \begin{bmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \cdot \\ \vdots \\ \cdot \\ \pi_{N-1} \\ \pi_N \end{bmatrix} = \begin{bmatrix} \pi_0 \\ \pi_1 \\ \pi_2 \\ \cdot \\ \vdots \\ \cdot \\ \pi_{N-1} \\ h\pi_N \end{bmatrix} \tag{10}$$

Consider first the stationary probability of being in state 0, π_0. Expanding the first row of Eq. (10) yields:

$$\pi_0 h_{0,0} + \pi_1 h_{1,0} = \pi_0 \implies \pi_1 = \frac{(1-h_{0,0})}{h_{1,0}} = \frac{h_{0,1}}{h_{1,0}}\pi_0. \tag{11}$$

Expanding the second row of Eq. (10) and substituting Eq. (11) yields:

$$\pi_0 h_{0,1} + \pi_1 h_{1,1} + \pi_2 h_{2,1} = \pi_1 \implies \pi_2 = \frac{h_{1,2}}{h_{2,1}}\pi_1. \tag{12}$$

Arguing in the same manner, and after some algebraic simplifications, we obtain the recurrence:

$$\pi_k = \frac{h_{k-1,k}}{h_{k,k-1}} \pi_{k-1}. \tag{13}$$

Using Eq. (13) and substituting $h_{k-1,k}$ and $h_{k,k-1}$ gives:

$$\begin{aligned}
\pi_k &= \pi_0 \prod_{i=1}^{k} \frac{h_{i-1,i}}{h_{i,i-1}} \\
&= \pi_0 \prod_{i=1}^{k} \frac{s_1(1-\frac{i-1}{N})}{s_2 \frac{i}{N}} \\
&= \pi_0 (\frac{s_1}{s_2})^k \prod_{i=1}^{k} (\frac{N-i+1}{i}) \\
&= \binom{N}{k} (\frac{s_1}{s_2})^k \pi_0. \qquad (14)
\end{aligned}$$

Consider the sum $\sum_{k=0}^{N} \pi_k$. Using Eq. (14) 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[4] that $\sum_{k=0}^{N} \pi_k$ sums to unity, we obtain:

$$\pi_0 = s_2^N. \tag{15}$$

Replacing Eq. (15) in Eq. (14), we obtain a closed form expression for the stationary probability π_k as:

$$\pi_k = \binom{N}{k} s_1^k s_2^{N-k} \tag{16}$$

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 $Var[X^*] = Ns_1s_2$.

Consequently, $E[p_1(\infty)] = \frac{1}{N}E[X^*] = s_1$, and $var[p_1(\infty)] = \frac{1}{N^2}Var[X^*] = \frac{1}{N}s_1s_2$.

[4] The reader should observe that Eq. (14) is quite similar to the closed form expression we obtained for the so-called "direct" solution (Eq. (3)), except that we have now also obtained a *state dependent* combinatorial term. It is this combinatorial term, obtain as a consequence of the state-dependent transitions, that makes this discretization work!

Thus, Theorem 3 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 4 is also proved.

6 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. There is a straight-forward mechanism to extend the binomial result for the multinomial scenario, and can be indeed described as follows: Let us assume that $\mathrm{P(t)} = [\mathrm{p_1(t)}, \ldots, \mathrm{p}_r\mathrm{(t)}]^{\mathrm{T}}$ is the current estimate of S. At any single time instant we observe that this represents $\binom{r}{2}$ pairs of binomial sequences. Therefore when an event i occurs, the probability p_i can be increased based on a binomial distribution with regards to any or all the $\binom{r}{2}$ pairs. So a simple-minded strategy is to associate event i with any other event j so that when event i occurs, p_i is increased and p_j is decreased maintaining $p_i + p_j$ to have the same value as it had before the event i occurred. Notice thus that P will remain to be probability vector and consequently the proof of convergence of this updating rule will follow as a consequence of all the $\binom{r}{2}$ binomial sequences converging to their correction proportions.

However, rather than work with *binomial pairs* we now propose a more elegant solution which works with the multinomial sequence in its entirety, as Oommen and Rueda did in [27]. However, we emphasize the fascinating aspect that the Markov Chain obtained in the continuous scenario is completely distinct from the one which we encounter here. Unfortunately, although the asymptotic convergence properties are proven, the analogous Eigenvalue/Eigenvector properties are, as yet, unsolved – they are currently under investigation. As in the LA field, the discretized results that we have here are experimentally superior to their continuous counterparts.

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 *randomized* updating 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 $0 \leq p_i(t) < 1$

$$\text{(i) } p_i(t+1) := p_i(t) + \frac{1}{N}. \tag{17}$$

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:

$$\text{(ii) } p_j(t+1) := p_j(t) - \frac{1}{N} \tag{18}$$

$$P(t+1) := P(t) \text{ Otherwise.} \tag{19}$$

As before, we shall state two theorems about the properties of the estimate first and prove them simultaneously.

Theorem 5 *Let X be a multinomially distributed random variable, and $P(t)$ be the estimate of S at time t obtained by Eqs. (17), (18) and (19). Then $E[P(\infty)] = S$.*

Theorem 6 *Let X be a multinomially distributed random variable, and $P(t)$ be the estimate of S at time t obtained by Eqs. (17), (18) and (19). 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 5 and 6.

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 given in Eqs. (17), (18) and (19).

In brief, the rules given in Eqs. (17), (18) and (19) 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 observe 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 ${H^i}^T \Pi^i = \Pi^i$.

Since the expressions are identical, we can use the results from the binomial case to 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 5 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 see that Theorem 6 is also proven.

Remarks:

A few remarks regarding our method for updating the estimates are not out of place. Indeed:

- The reader will observe the very subtle point that our proofs of the theorems is based on the assumption[5] that the parameter of the environment remains "stationary long enough" for such a estimation scheme to be meaningful. This is unavoidable because, as far as we know, there are no available methods for formally analyzing estimation methods for *strictly non-stationary* environments. This is because, one has to first determine a model for the non-stationarity and for the way the underlying parameter would change within this model, before an estimation scheme can be devised. Since this is not possible in a general setting, what one rather needs is a method of estimation that quickly converges to the current parameter of the environment in a "weak" (i.e., not with probability '1') manner, and that would be able to quickly unlearn if the parameter changes. This is precisely what the SLWE and SDWE are able to do.

- To clarify the above, we explain the dilemma of proving the results for stationary environments, and then using them for non-stationary domains. The results we have derived are asymptotic, and are thus, valid only as $n \to \infty$. While this could prove to be a handicap, realistically, and for all practical purposes, the convergence takes place after a relatively small value of n. Thus, for the SLWE, 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, and this occurs in a geometric manner [21]. In other words, even if the environment switches its Bernoulli parameter after 50 steps, the SLWE will be able to track this change. This same observation is valid for the SDWE, for example, for relatively small values of the resolution of discretization. Observe too that we do not need to introduce the use of a "sliding window".

- The philosophy behind the form of the update defined by Eqs. (17), (18) and (19) 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 that 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.

[5] We are extremely grateful to the anonymous Referee who brought this up.

- It is pertinent to mention that although the rationale for updating is similar to that used for the Stochastic Nonlinear Fractional Knapsack algorithm [10], 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[6]. The reason for excluding the the end states in [10] was because the LA in [10] uses its "estimates" to select "actions" with frequencies proportional to the estimates. Consequently, enforcing an absorbing condition for the end states in [10] would imply always selecting the same action from that time instant onwards, and never selecting the alternate one thereafter. 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 [10] 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 [10] improved the learning speed dramatically compared to the approach presented in [9] 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 this paper is, in our opinion, 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 $\mathrm{O}(N^2)$. 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 $\mathrm{O}(N^2)$ to $\mathrm{O}(N)$. 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[7]. In addition, we believe

[6] 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!

[7] The reader can easily see that this is also an alternate, although more cumbersome, way to decouple the updates

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!

- The issue of how we can obtain a generalization of these results for other distributions is very interesting. As a first step, we propose approximating the distribution using a equi-depth or equi-width histogram, whereby the original distribution itself can be approximately modeled as a multinomial distribution by considering its relative proportions in each of the histogram bins. More elegant solutions of this are currently being investigated.

7 Experimental Results: Artificial and Real-lie Data

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. Also, while Sections 7.1-7.3 deal with artificial data, Section 7.4 presents the results for a real-life application domain.

7.1 Comparison with the MLEW of Binomial distributions

The estimation of the parameters for binomial random variables has been extensively tested for numerous binomial distributions, and we cite the results from two of these experiments here. To do this, in the interest of standardization, we have adopted the same simulation methodology used in [27], 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. Further, to enable us to obtain a *smooth* result, we performed ensembles of $1,000$ simulations each consisting of 400 time steps. The true underlying value of the quantity to be estimated was randomly changed every 50 time steps. The plots from each experiment are presented in Figures 3 and 4 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 a poor overall accuracy. In Figure 4, 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 3, when the window size is smaller, the MLEW is capable of tracking the changes, but not

of the different components of a vector of multinomial estimates.

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.

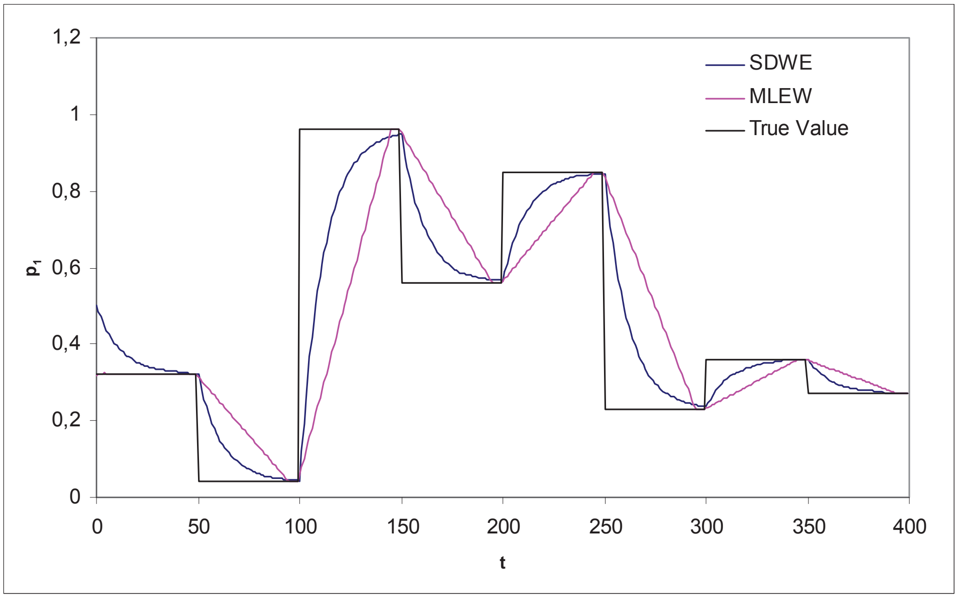


Figure 3: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and MLEW, using $N = 6$ and $w = 32$.

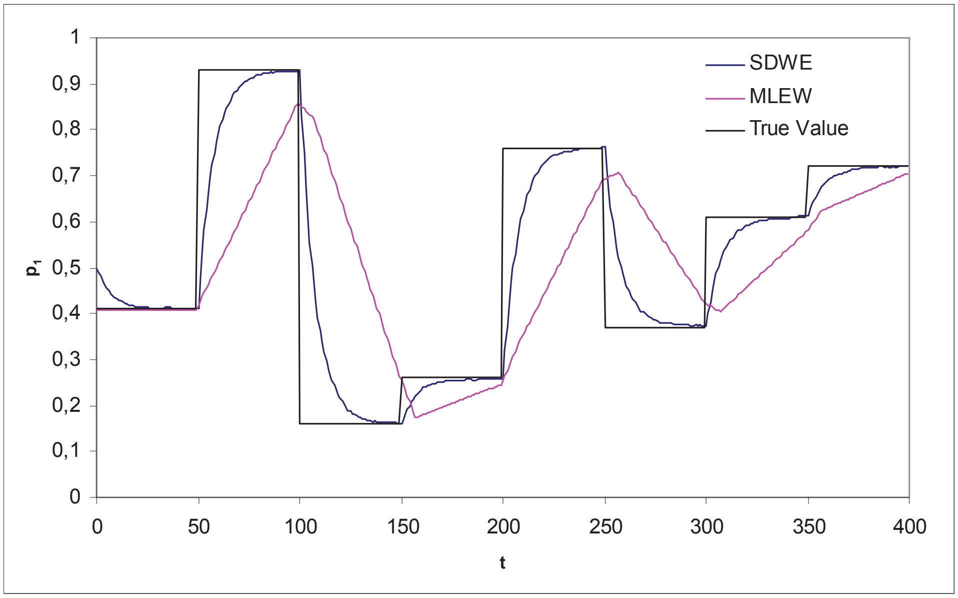


Figure 4: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and MLEW, using $N = 8$ and $w = 57$.

In Figure 5, we keep the same window size and resolution as previously done in Figure 3, but we instead increase the periodicity by which the environment switches from 50 to 200 iterations. The intention here is to simulate a slowly-varying environment. Interestingly, the SDWE seems to have higher adaptivity than the SLWE just after the environment switch, and then, as time proceeds, the MLEW succeeds to converge to the new true parameter of the environment in the order of 40 iterations, i.e, after some delay larger than the window size. The fact that the periodicity, i.e., 200, is much larger than the window size, 32, favors the MLEW after an order of 40 instants subsequent to the switch. While the SDWE moves faster towards the true value, it is unable to get of rid of some fluctuations in the neighborhood of the true parameter characterizing the environment – in spite of the fact that we are averaging over an ensemble of 1000 experiments. These fluctuations of the SDEW merely reflect a larger variance than the MLEW. Similar findings are observed in Figure 6 where we increase the periodicity to 200, as opposed to the value of 50 used for Figure 4.

Thus, we conclude that the SDWE seems to be more convenient for "fast-varying" environments, which opens avenues towards devising methods that combine the MLEW and SDWE schemes so as to take advantage of the best properties of them both. Under a slowly varying, a simple idea is to use the SDWE with a low resolution in order to quickly adapt to the new environment after the switch, and to subsequently resort to the MLEW so as to reduce the variance. Another alternative, is to resort to the SDWE by starting with a low resolution so as to converge faster, and to thereafter increase the resolution to "stabilize" around the true parameter.

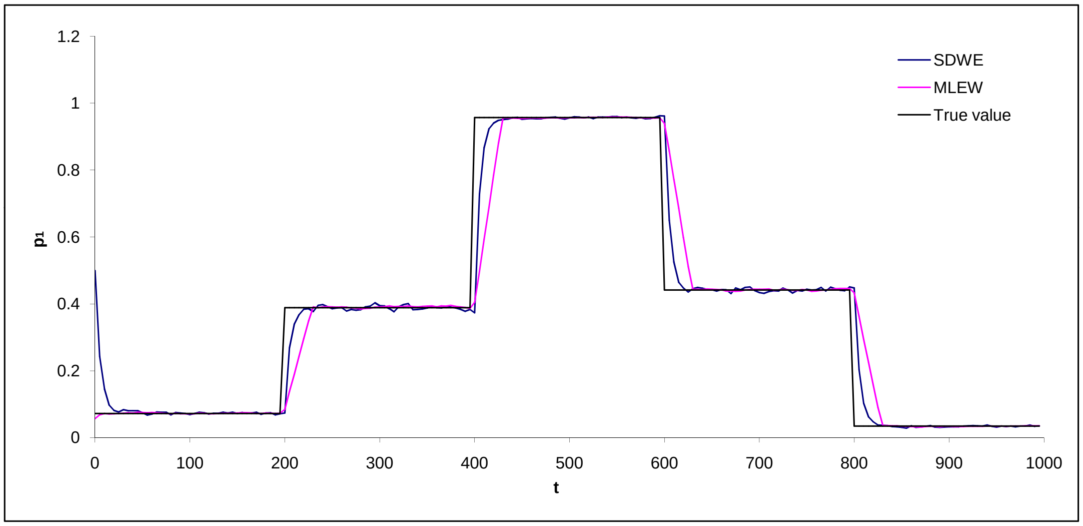


Figure 5: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and MLEW, using $N = 6$ and $w = 32$ for a slowly varying environment.

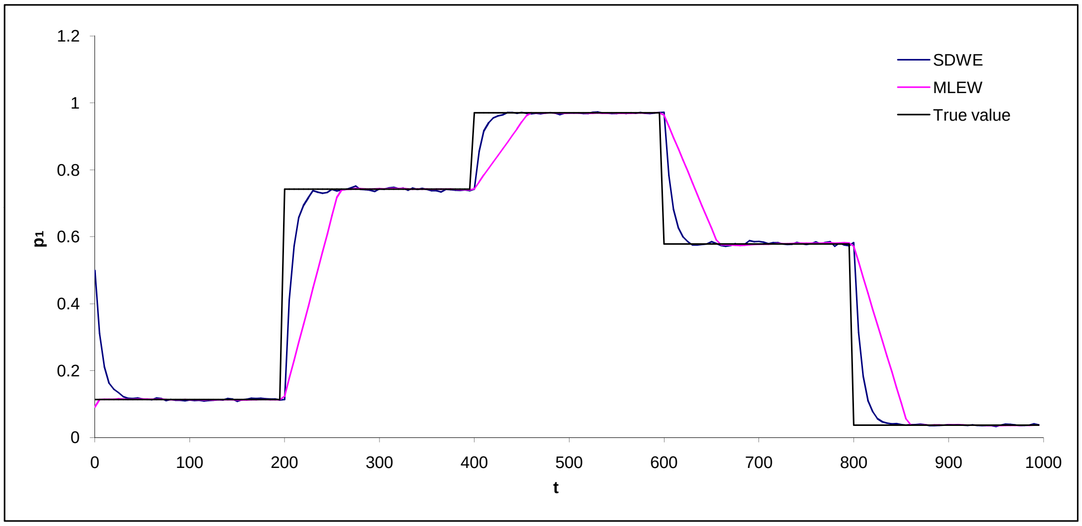


Figure 6: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and MLEW, using $N = 8$ and $w = 57$ for a slowly varying environment

7.1.1 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 evaluated the estimators for 400 steps, and repeated this $1,000$ times, after which the ensemble average of P was evaluated. To report the quality of an estimator, we computed the index $||P - S||$, i.e., 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 7 and 8, 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 7, 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 7 and 8, we observe that a large value of N yields low variance. The problem, however, is that the rate of convergence is slower than when we used a lower value of N. A low value of N results in a faster convergence, but it also yielded 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 variance.

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.

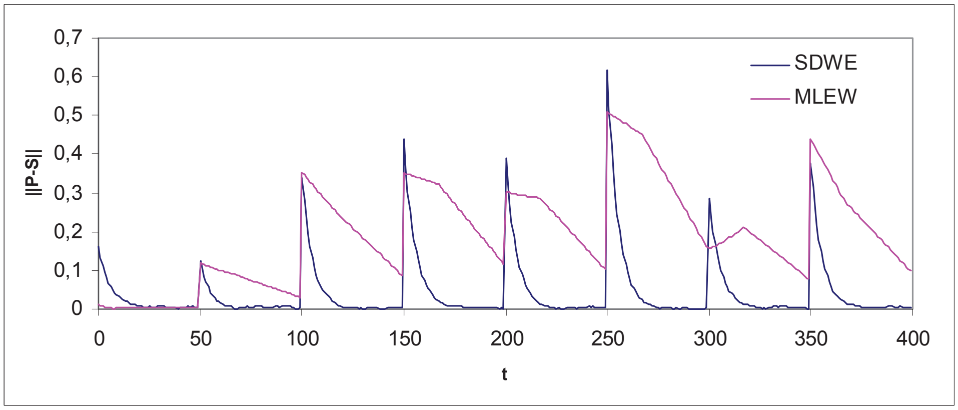


Figure 7: Plot of the Euclidean norm P-S (or the Euclidean distance between P and S), for both the SDWE and MLEW, where N is 6 and the window size is 67 respectively.

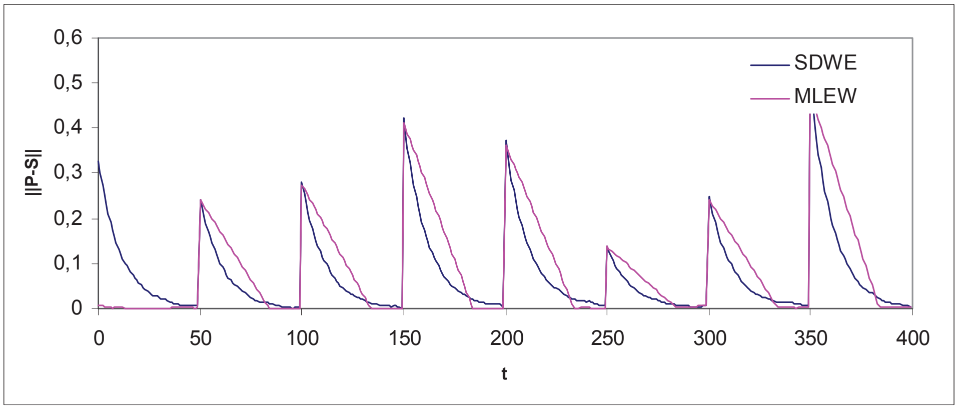


Figure 8: Plot of the Euclidean norm P-S (or the Euclidean distance between P and S), for both the SDWE and MLEW, where N is 12 and the window size is 34 respectively.

7.2 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 [27]. We randomly chose values for the resolution N and for the parameter λ of the SLWE [27] from the intervals $[8, 16]$ and $[0.9, 1]$, respectively. It was reported in [27] 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 9. Similarly, the estimates of $p_1(t)$, obtained from the SDWE and SLWE, using $N = 12$ and $\lambda = 0.9623$ is depicted in Figure 10. 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.

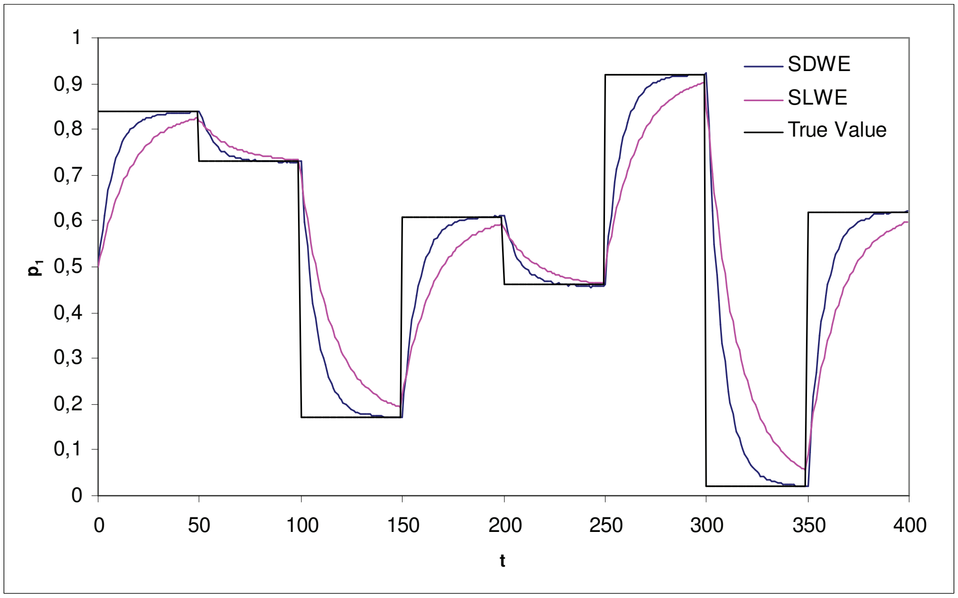


Figure 9: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and SLWE, using $N = 8$ and $\lambda = 0.938$ respectively.

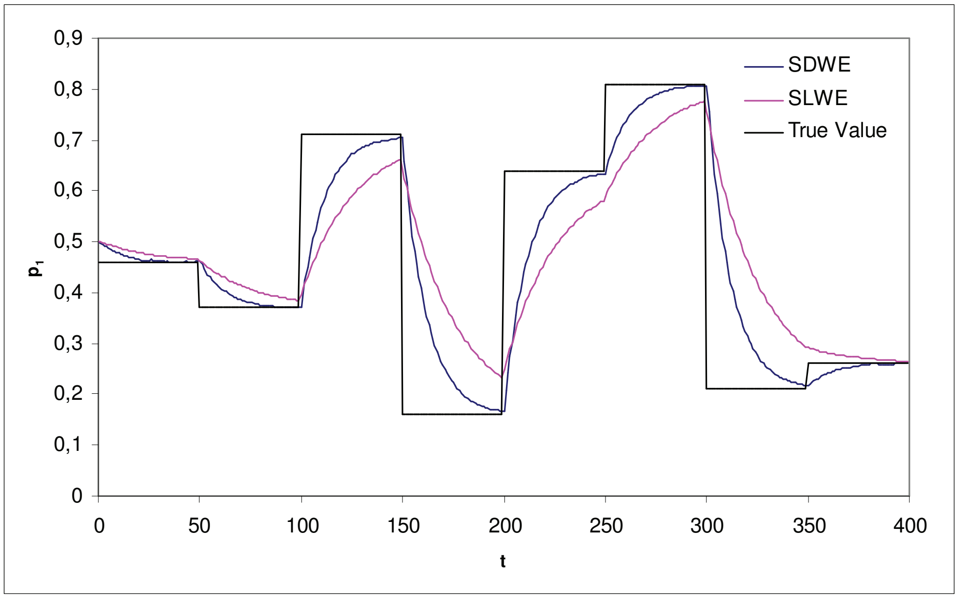


Figure 10: The expected value of the estimates of $p_1(t)$, obtained from the SDWE and SLWE, using $N = 12$ and $\lambda = 0.9623$ respectively.

7.3 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 [27] from the intervals $[6, 15]$ and $[0.9, 1]$, respectively. Figure 11 illustrates the case of $N = 12$ for the SDWE and $\lambda = 0.931$ for the SLWE, whereas Figure 12 depicts the results for $N = 9$ and $\lambda = 0.942$.

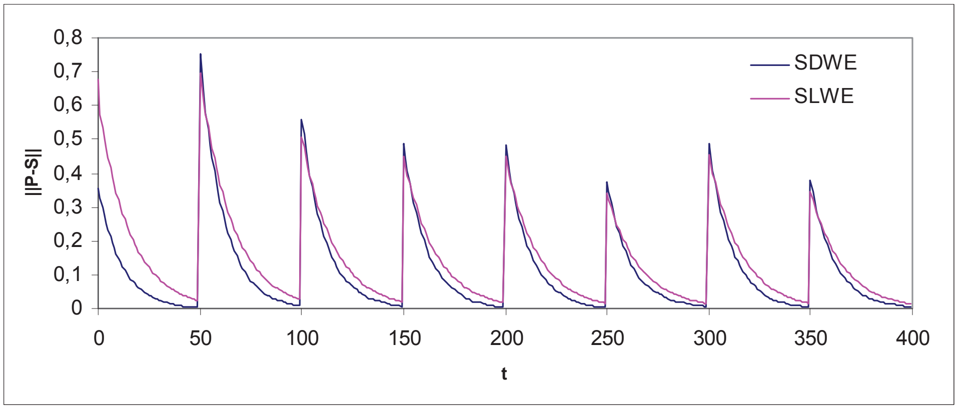


Figure 11: Plot of the Euclidean norm P-S (or the Euclidean distance between P and S), for both the SDWE and SLWE, where N is 12 and the $\lambda = 0.931$ respectively.

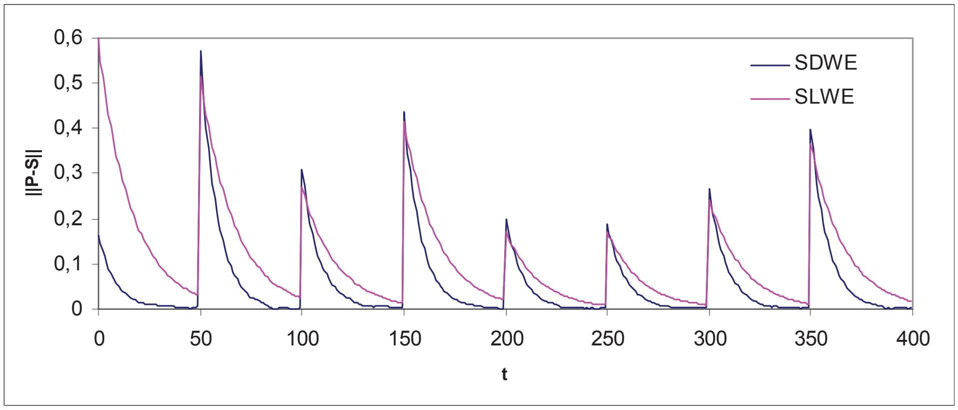


Figure 12: Plot of the Euclidean norm P-S (or the Euclidean distance between P and S), for both the SDWE and SLWE, where N is 9 and $\lambda = 0.942$ respectively.

From both Figures 11 and 12, 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 a higher accuracy but slower convergence speed.

7.4 Real-life Application of the Scheme: Language Detection

The solution that we have proposed is not only theoretical in nature. Indeed, we have also utilized it in a real-life language detection domain involving the languages English and Arabic[8].

In order to simulate real life data, we downloaded text from two on-line newspapers, where one of them was in Arabic (Aljazeera.net) and the other was in English (BBC.com). To render the problem non-trivial, the text from both newspapers was scrambled in a random manner. The scrambled text was then transformed into a series of bits where each character in the text corresponds to 8 bits according to the ASCII code. This series of bits was given to our algorithm where the aim of the exercise was to track the occurrence of the bit '1' which served as the recognizing feature, whence we attempted to achieve the classification of the corresponding piece of text. An example of the scrambled text is given in Table 1, where the respective change instants of the language from Arabic to English and *vice versa* are also recorded. Here the change instant denotes the index of the bit where a change of the language in the text occurs.

We report here the results of three experiments, each performed with three different resolutions. Figure 13(a), Figure 13(b) and Figure 13(c), report the variation of the estimate of the occurrence of the bit '1' for the respective resolutions $N = 10$, $N = 20$ and $N = 40$.

The reader can easily observe that in the three figures, the trend of the estimate changes as soon

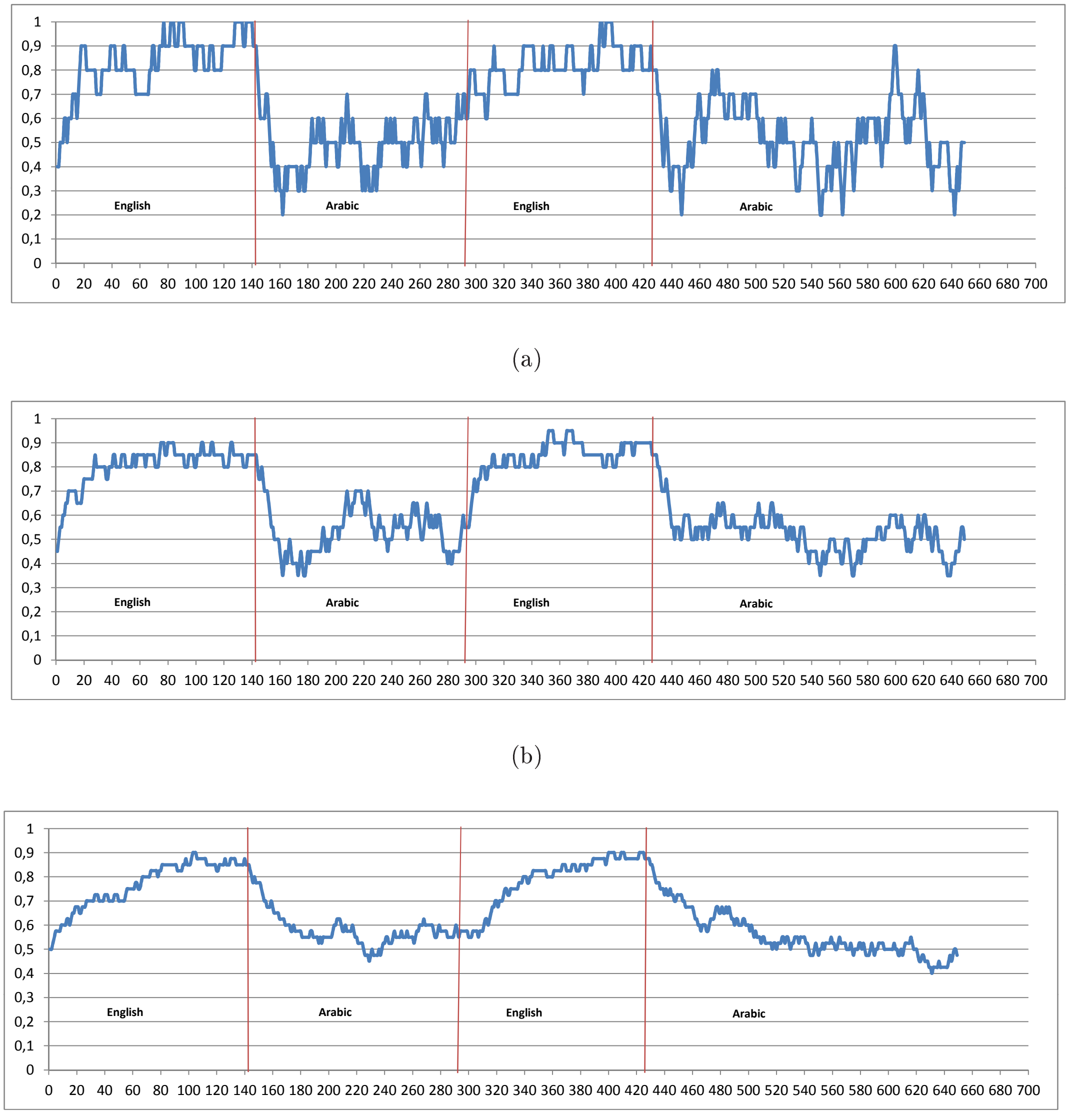
[8] As requested by one of the anonymous Referees, the data used for these experiments is available at https://www.dropbox.com/s/u4p8uop1que3ey8/text.txt?dl=0.

as there is a language change. The occurrence probability of ‘1’ for English stabilizes around 0.9, while for Arabic it seems to stabilize around 0.5. Interestingly enough, even for a resolution as small as $N = 10$ (see Figure 13(a)), the fluctuations are much more considerable than for the resolution $N = 20$ and $N = 40$, specially between time instants 600 and 620. The latter “considerable” fluctuations are explained by the fact that the level of discretization is characterized by a resolution parameter as small as 10. However, the fluctuations are almost unobservable as N increases. Thus, in Figure 13(c), we observe that the resolution is increased to $N = 40$, the fluctuations are extremely minor. Indeed, the curve is much smoother than for the case when $N = 10$. Furthermore, it takes around 40 bits for the scheme to “converge” and stabilize after each language switch. Thus, for example, while a language change took place at the instant time around 140, the estimate converged around the time instant 180 .

The convergence of the estimate and ability to track the language change is, in our opinion, quite fascinating.

Table 1: The time instances when a change in the language (from English to Arabic and *vice versa*) occurred.

Text	Bit number
oncountrytofailtorepayal	1 - 140
يسبوكصناعةأولطائر	140 - 293
Greecebecamethefi	293 - 425
ملمسؤولونفيشركة	425 - 649



(a)

(b)

(c)

Figure 13: This figure depicts the variation of the estimate for different values of the resolution N: (a) For $N = 10$, (b) for $N = 20$, and (c) for $N = 40$.

8 Conclusion

This paper has presented a novel discretized estimator that is able to cope with non-stationary binomial/multinomial distributions using finite memory. After demonstrating that a straightforward naive discretization of the continuous SLWE is inadequate, we have proposed a novel randomized discretization strategy, namely, the SDWE. To the best of our knowledge, our SDWE is the first reported discretized counterpart of the SLWE. Thus, through this paper, we have 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.

The most outstanding contribution of our work is that of achieving multinomial estimation without relying on a set of binomial estimators, and where the underlying strategy is truly randomized. We believe that this contribution is particularly of a pioneering sort.

The paper also reports the results of the estimate being able to track the changes in a corpus of text interspersed by English and Arabic fragments. The change detection observed is quite fascinating.

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] A. Bouchachia, "Fuzzy classification in dynamic environments," *Soft Computing*, vol. 15, no. 5, pp. 1009–1022, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00500-010-0657-0

[4] E. Cohen and M. Strauss, "Maintaining time-decaying stream aggregates," in *Proc. 22nd ACM Symposium on Principles of Database Systems*, 2003.

[5] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi, "An information-theoretic approach to detecting changes in multi-dimensional data streams," in *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*. Citeseer, 2006.

[6] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '00. New York, NY, USA: ACM, 2000, pp. 71–80. [Online]. Available: http://doi.acm.org/10.1145/347090.347107

[7] A. Dries and U. Rückert, "Adaptive concept drift detection," *Stat. Anal. Data Min.*, vol. 2, no. 5, pp. 311–327, Dec. 2009.

[8] J. a. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 44:1–44:37, Mar. 2014. [Online]. Available: http://doi.acm.org/10.1145/2523813

[9] 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.

[10] 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.

[11] G. Horn, "An interdiciplinary approach to optimisation in parallel computing," Ph.D. dissertation, Institute of informatics , University of Oslo, Oslo, Norway, 2012.

[12] D. G. Kendall, "Stochastic processes and population growth," *J. Roy. Statist. Soc. Ser. B.*, vol. 11, pp. 230–264, 1949.

[13] R. Klinkenberg, "Learning drifting concepts: Example selection vs. example weighting," *Intell. Data Anal.*, vol. 8, no. 3, pp. 281–300, Aug. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1293831.1293836

[14] 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.

[15] ——, "Gradual forgetting for adaptation to concept drift," in *Proceedings of ECAI 2000 Workshop on Current Issues in Spatio-Temporal Reasoning*, 2000.

[16] 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.

[17] 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.

[18] P. Kulkarni and R. Ade, "Incremental learning from unbalanced data with concept class, concept drift and missing features: A review," *International Journal of Data Mining and Knowledge Management Process (IJDKP)*, vol. 4, no. 6, pp. 15–29, Nov. 2014.

[19] 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.

[20] M. May, B. Berendt, A. Cornuï£¡jols, 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.

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] B. J. Oommen, A. Yazidi, and O.-C. Granmo, "An adaptive approach to learning the preferences of users in a social network using weak estimators." *Journal of Information Processing Systems*, vol. 8, no. 2, 2012.

[27] 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.

[28] G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand, "Exponentially weighted moving average charts for detecting concept drift," *Pattern Recognition Letters*, vol. 33, no. 2, pp. 191 – 198, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865511002704

[29] 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, October 2006.

[30] R. Sebastião and J. a. Gama, "Change detection in learning histograms from data streams," in *Proceedings of the Aritficial Intelligence 13th Portuguese Conference on Progress in Artificial*

Intelligence, ser. EPIA'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 112–123. [Online]. Available: http://dl.acm.org/citation.cfm?id=1782254.1782265

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

[32] A. Stensby, B. J. Oommen, and O.-C. Granmo, "The use of weak estimators to achieve language detection and tracking in multilingual documents," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 04, p. 1350011, 2013.

[33] 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, September 2006.

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

[35] 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.

[36] ——, *Networks of Learning Automata: Techniques for Online Stochastic Optimization.* Boston: Kluwer Academic, 2003.

[37] A. Tsymbal, M. Pechenizkiy, P. Cunningham, and S. Puuronen, "Dynamic integration of classifiers for handling concept drift," *Inf. Fusion*, vol. 9, no. 1, pp. 56–68, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.inffus.2006.11.002

[38] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996. [Online]. Available: http://dx.doi.org/10.1007/BF00116900

[39] A. Yazidi, O.-C. Granmo, B. J. Oommen, M. Gerdes, and F. Reichert, "A user-centric approach for personalized service provisioning in pervasive environments," *Wireless Personal Communications*, vol. 61, no. 3, pp. 543–566, 2011.

[40] A. Yazidi, O.-C. Granmo, B. J. Oommen, and M. Goodwin, "A novel strategy for solving the stochastic point location problem using a hierarchical searching scheme," *To appear in IEEE Transactions on Systems, Man and Cybernetics*, 2014.

[41] 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.