Tracking the Preferences of Users Using Weak Estimators

Anis Yazidi[1], Ole-Christoffer Granmo[1], and B. John Oommen[2,⋆]

[1] Dept. of ICT, University of Agder, Grimstad, Norway
[2] School of Computer Science, Carleton University, Ottawa, Canada

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 [2], multimedia [4] and TV programs [14], based on their respective contents.

⋆ Chancellor's Professor; Fellow : IEEE and 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.

D. Wang and M. Reynolds (Eds.): AI 2011, LNAI 7106, pp. 799–808, 2011.

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 [4,5]. A number of previous studies [8] 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.

Recently, Oommen and Rueda [16] 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) [13,20]. The SLWE has found successful applications in many real-life problems that involve estimating distributions in non-stationary environments such as in adaptive encoding [17], route selection in mobile ad-hoc networks [15], and topic detection and tracking in multilingual online discussions [19]. 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.

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 in the user's interests corresponds to the problem of learning evolving concepts [21]. There are several studies that have dealt with the task of learning a user's interests. These include the use of a sliding window [11], aging examples [9], and a Gradual Forgetting (GF) function [6,7,8] 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 [11], 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 [8] suggested a linearly-decreasing function, $w = f(t)$, for decaying the relative weights of the GF as follows:

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

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 [8], 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) [1] detection procedure and the Shiryaev−Roberts−Pollak detection procedure. In [18], 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 [5]. 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 [4,5] 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 [5] refined their proposed updating method defined earlier in [4] 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 [4].

3 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 [16], 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.

3.1 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 [3,10]. We adopt the Profile Representation Model advocated by Hossain and his co-authors in [4,5]. 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 [4,5] is similar to that of [23] in the sense that it is based on <feature, weight> pairs, except that in [4,5], the authors have invoked a normalized score for the data items. We shall first briefly present the Profile Representation Model reported in [4,5]. 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.

3.2 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) [12]. In this paper, we rely on the *Service Usage History* (analogous to the history maintained by the authors of [4,5]) 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 [16], 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 the data item. Thus, for example, if a user currently views an "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 [4,5] 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.

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 [4,5] deals only with this specific case, i.e., of disjunctive data items.

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 items 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:

$$p_{action}(n+1) \leftarrow 1 - \lambda(1 - p_{action}(n)) \tag{2}$$

$$p_{romantic}(n+1) \leftarrow 1 - \lambda(1 - p_{romantic}(n)) \tag{3}$$

$$p_{comedy}(n+1) \leftarrow \lambda p_{comedy}(n) \tag{4}$$

$$p_{horror}(n+1) \leftarrow \lambda p_{horror}(n) \tag{5}$$

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 $k \in S_{genre} = \{action, romantic, comedy, horror\}$

$$W_k(n+1) = \frac{p_k(n+1)}{\sum_{j \in S} p_j(n+1)} \tag{6}$$

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 $k \in S_{genre}$. With these assumptions, based on the results of the previous subsection, we easily derive the asymptotic weight:

$$\mathrm{E}\left[W_k(\infty)\right] = \frac{s_k}{\sum_{j \in S} s_j}. \tag{7}$$

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.

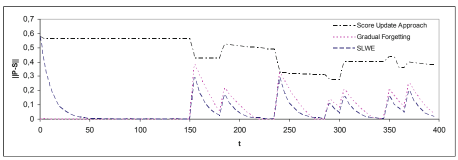
Modelling changes in the Interests. We suppose that at every time instance 'n', the *Learning Preferences Manager* is fed by a service usage instance. We further assume that the distribution of the user's interests, relative to a given attribute, undergoes an abrupt change at a random time instance with 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.

4 Experimental Results

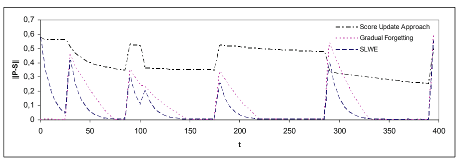
To verify our computational model and our proposed solution, we have performed extensive simulations. However, in the interest of space and brevity, we report here only a subset of these results. Due to space limitations, experimental results concerning the case of disjunctive data items are omitted in this paper and are found in [22]. 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. 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 multinomial case. 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. 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 1(a), 1(b), 1(c) and 1(d), 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. 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 1, we report the error rates associated with the experiments plotted in Figures 1(a), 1(b), 1(c) and 1(d). We also include the error rates for the MLE augmented with a sliding window in Table 1. Clearly, one observes that the SLWE exhibits a lower error rate than the GF, the SU and the MLE.

Table 1. The effects of varying the window size and the updating parameter on the error rates for the various schemes investigated for disjunctive data items

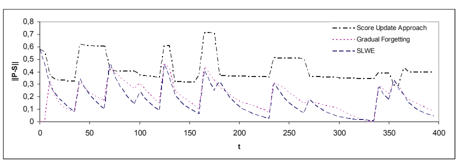
Figure No.	Error rate: SLWE	Error rate: GF	Error rate: SW	Error rate: SU
Figure 1(a)	0.0612	0.0724	0.0836	0.4606
Figure 1(b)	0.0665	0.1006	0.1152	0.4037
Figure 1(c)	0.1601	0.1893	0.2074	0.4175
Figure 1(d)	0.0507	0.0567	0.0672	0.4165



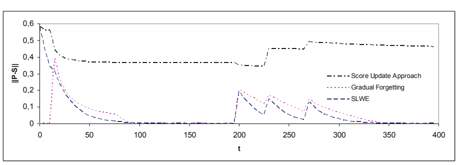
(a)



(b)



(c)



(d)

Fig. 1. 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 (a) $\lambda = 0.908$ and w = 35, (b) $\lambda = 0.903$ and w = 44, (c) $\lambda = 0.952$ and w = 63 and (d) $\lambda = 0.948$ and w = 76

5 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 attribute 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 match-making 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. Basseville, M., Nikiforov, I.V.: Detection of abrupt changes: theory and application. Prentice-Hall, Inc. (1993)
2. Godoy, D., Amandi, A.: User profiling in personal information agents: a survey. Knowl. Eng. Rev. 20, 329–361 (2005)
3. Godoy, D., Schiaffino, S., Amandi, A.: Interface agents personalizing web-based tasks. Cognitive Systems Research, Special Issue on Intelligent Agents and Data Mining for Cognitive Systems 5(3), 207–222 (2004)
4. Hossain, M.A., Atrey, P.K., El Saddik, A.: Gain-based selection of ambient media services in pervasive environments. Mob. Netw. Appl. 13(6), 599–613 (2008)
5. Hossain, M.A., Parra, J., Atrey, P.K., El Saddik, A.: A framework for human-centered provisioning of ambient media services. Multimedia Tools and Applications 44, 407–431 (2009)
6. Koychev, I.: Gradual forgetting for adaptation to concept drift. In: Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning, pp. 101–106 (2000)

7. Koychev, I., Lothian, R.: Tracking drifting concepts by time window optimisation. In: Bramer, M., Coenen, F., Allen, T. (eds.) Research and Development in Intelligent Systems XXII, pp. 46–59. Springer, London (2006)
8. Koychev, I., Schwab, I.: Adaptation to drifting user's interests. In: Proceedings of ECML 2000 Workshop: Machine Learning in New Information Age, pp. 39–46 (2000)
9. Maloof, M.A., Michalski, R.S.: Selecting examples for partial memory learning. Machine Learning 41, 27–52 (2000)
10. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Ontological user profiling in recommender systems. ACM Trans. Inf. Syst. 22(1), 54–88 (2004)
11. Mitchell, T.M., Caruana, R., Freitag, D., McDermott, J., Zabowski, D.: Experience with a learning personal assistant. Commun. ACM 37(7), 80–91 (1994)
12. Montaner, M., Lpez, B., de la Rosa, J.L.: A taxonomy of recommender agents on the internet. Artificial Intelligence Review 19, 285–330 (2003)
13. Narendra, K.S., Thathachar, M.A.L.: Learning Automata: An Introduction. Prentice Hall (1989)
14. Naudet, Y., Aghasaryanb, A., Mignon, S., Toms, Y., Senot, C.: Ontology-Based Profiling and Recommendations for Mobile TV. In: Wallace, M., Anagnostopoulos, I.E., Mylonas, P., Bielikova, M. (eds.) Semantics in Adaptive and Personalized Services. SCI, vol. 279, pp. 23–48. Springer, Heidelberg (2010)
15. Oommen, B.J., Misra, S.: Fault-tolerant routing in adversarial mobile ad hoc networks: an efficient route estimation scheme for non-stationary environments. Telecommunication Systems 44, 159–169 (2010), 10.1007/s11235-009-9215-4
16. Oommen, B.J., Rueda, L.: Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments. Pattern Recogn. 39(3), 328–341 (2006)
17. Rueda, L., Oommen, B.J.: Stochastic automata-based estimators for adaptively compressing files with nonstationary distributions. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36(5), 1196–1200 (2006)
18. Shiryayev, A.N.: Optimal Stopping Rules. Springer, Heidelberg (1978)
19. Stensby, A., Oommen, B.J., Granmo, O.-C.: Language Detection and Tracking in Multilingual Documents Using Weak Estimators. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 600–609. Springer, Heidelberg (2010)
20. Thathachar, M.A.L., Sastry, P.S.: Networks of Learning Automata: Techniques for Online Stochastic Optimization. Kluwer Academic Publishers (2004)
21. Widmer, G.: Tracking context changes through meta-learning. Mach. Learn. 27(3), 259–286 (1997)
22. Yazidi, A., Granmo, O.C., Oommen, B.J.: An adaptive approach to learning the preferences of users in a social network using weak estimators. Unabridged version of this paper (submitted for publication)
23. Yu, Z., Zhou, X., Zhang, D., Chin, C.Y., Wang, X., Men, J.: Supporting context-aware media recommendations for smart phones. IEEE Pervasive Computing 5, 68–75 (2006)