Invited Paper

Journal of Information Processing Systems, Vol.8, No.2, June 2012 http://dx.doi.org/10.3745/JIPS.2012.8.2.191

An Adaptive Approach to Learning the Preferences of Users in a Social Network Using Weak Estimators

B. John Oommen*, Anis Yazidi** and Ole-Christoffer Granmo**

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 a probability of 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 a 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 technology.

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 that are 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 [5], multimedia [7], and TV programs [20], which are based on their respective contents. For instance, a comprehensive study for personalized service provisioning was performed by Naudet et al. from Bell Labs [20], where the authors designed an application for filtering the TV content provided to users' mobile devices based on their learned profiles. The application is based on the use of ontologies to capture content descriptions as well as the

※ A preliminary version of some of the results of this paper was presented at the AI'11, the 2011 Australasian Joint Conference on Artificial Intelligence, in Perth Australia, in December 2011.

Manuscript received March 16, 2012; accepted April 9, 2012.

Corresponding Author: B. John Oommen

* School of Computer Science, Carleton University, Ottawa, Canada (oommen@scs.carleton.ca). This author is also an *Adjunct* Professor with the Department of ICT, University of Agder, Grimstad, Norway.

** Department of ICT, University of Agder, Grimstad, Norway (anis.yazidi@uia.no, ole.granmo@uia.no)

users' interests. The latter interests are, in turn, mined by using a dedicated profiling engine presented in [1], which leverages Machine Learning (ML) techniques for user profiling. The work reported in [30] presented a “product” that recommends vendors' web pages by measuring the similarity between the user's profile and the vendor's web page when the user is in the vicinity of the vendor (seller). The user's profile is constructed through mining the history of his web log. Another example that falls in the class of mobility-aware applications is the PILGRIM system, which makes use of the user's location to recommend relevant web links [3]. In the same vein, the SMMART framework dynamically locates products that match the shopping preferences of a mobile user [13].

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 [7, 8]. A number of previous studies [12] 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 as being difficult and non-trivial.

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

In earlier works [10-12], 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 the 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 [12] that the GF can be an enhancement to the sliding window paradigm. In this sense, observations within each sliding window are weighted using a GF function.

Recently, Oommen and Rueda [23] 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 a Stochastic Learning Weak Estimation (SLWE), and is based on the principles of stochastic Learning Automata (LA) [19, 28]. The SLWE has found successful applications in many real-life problems that involve estimating distributions in non-stationary environments such as in adaptive encoding [24], route selection in mobile ad-hoc networks [22], and topic detection and tracking in multilingual online discussions [27]. In addition, the SLWE can be applied to solve emerging research problems such as in mining frequent itemsets in data-streams [4, 9, 14, 32], real-time monitoring of wireless biosensors [25], and dynamic load balancing [21]. Motivated by these successful applications of the SLWE in various areas, we consider employing the SLWE for solving the intriguing problem of tracking user's interests in the course of this study. The objective of the paper is to present a personalized Learning Preferences Manager, a *modus operandus* for capturing user's preferences. The latter will be able to cope with changes brought about by variations in the distribution of the user's interests, which will be where the SLWE plays a prominent part.

The rationale for choosing a weak estimator for non-stationary environments is that estimators that converge with probability 1 (e.g., the MLE and Bayesian estimates) cannot easily unlearn and adapt to the drift in interests. In our opinion, the appealing properties of the SLWE lies in its recursive *multiplication*-based update form that achieves the process of unlearning stale data by an order-of-magnitude faster than a traditional *addition*-based updating scheme. Moreover, unlike other legacy state-of-the-art approaches, the adaptability of the SLWE and its capability to cope with changing environments can be achieved using the choice of the scheme's fixed internal parameter, λ. In fact, the choice, λ, is not a critical issue, and it does not influence the convergence speed of the SLWE. Indeed, we present simulations results that demonstrate the superiority of the SLWE compared to GF, the sliding window, and the approach presented by Hossain *et al.* [8], which represent the state-of-the-art technology.

1.1 Contribution

The novel contributions of this paper, when it concerns user's profiling techniques, are the following:

- To the best of our knowledge, our current work presents the first attempt to apply a LA-inspired approach, such as the SLWE, to the real-life problem of tracking user's interests. We hope that the current study paves the way towards more applications of LA-based techniques to the realm of user profiling.
- Philosophically, our profile representation model is distantly related to the approach presented in [7, 8], where the authors utilized the history to update the affinity of the user's interests. However, a substantial difference from the latter studies is our novel categorization of the data items that constitute a profile into its so-called *disjunctive* and *conjunctive* data items. To the best of our knowledge, although profile update approaches in which the data items that are disjunctive have received a significant interest, the case of conjunctive data items remains largely unaddressed. In this work, we propose an adequate update form based on the principles of the SLWE for each case of these two cases[1].
- The model that we have adopted, namely that of the user's interests changing “abruptly," is, in itself, interesting. In fact, instead of presuming that the so-called environment's “switch” occurs with some fixed periodicity [23], we assume that changes in the distribution of the user's interests occur at *unknown random time instants*. Furthermore, we suppose that the distribution changes to a possibly new random distribution after the “switch." Such a model of the distribution's versatility is more realistic than the one that possesses a fixed periodicity-based changing model, and thus, this is more appropriate in the context of estimating the user's preferences. Clearly, the described settings represent a particularly challenging scenario for any approach that models and studies change detection! The experiments conducted and the results reported demonstrate that our approach exhibits lower errors and a faster adaptivity represent the state-of-the-art. technology.

[1] The reader will observe that the application of the SLWE has already been reported in the literature. However, its use for conjunctive and disjunctive data items is totally new.

- The model and technique that we have used here (for learning user's preferences) has been incorporated into a more comprehensive system whose architecture, design, and implementation details are found in [31].

1.2 Paper Organization

The rest of the paper is organized as follows: in Section 2, we report on a brief survey of the available results in tracking user's interests. Then, in Section 3, we present some of the theoretical properties of the SLWE. In Section 4, we introduce a formal model of the user's preferences, as well as an SLWE-based solution to the problem. Experimental results obtained by rigorously testing our solution for a variety of scenarios are presented in Section 5. Section 6 concludes the paper.

2. State of The Art

The core function of a personalized Learning Preferences Manager is to update the user's profile in a dynamic and incremental way. This is done so that the "Manager" can closely follow the real-time evolution of the user's interests. In fact, any user's interests are not constant over time, and therefore it is imperative that the system takes the profile's drifting 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 [29]. There are several studies that have dealt with the task of learning a user's interests. These include the use of a sliding window [17], aging examples [15], and a Gradual Forgetting (GF) function [10-12], 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 [17], 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 [12] suggested a linearly-decreasing function, $w = f(t)$, for decreasing the relative weights of the GF as follows:

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

where i denotes a counter of observations starting from the most recent one, n is the number of observations, and $k \in [0,1]$ 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 [12], proposed to apply the GF *within each sliding window*. Thus, in this case, the parameter n (e.g., the length of the observation sequence) in equation (1) was set to be equal to L, where L denotes the length of the window.

Apart from the sliding window and GF schemes, other approaches, which also deal with *change detection*, have also emerged. In general, there are two major competitive sequential change-point detection algorithms: Page's cumulative sum (CUSUM) [2] detection procedure and the Shiryaev¯Roberts¯Pollak detection procedure. In [26], 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 in time, and that there is no change. Inherent limitations of CUMSUM and the Shiryaev ¯ Roberts ¯ Pollak approaches for online implementation are the demanding computational and memory requirements. In contrast to the CUMSUM and the Shiryaev¯Roberts¯Pollak, the SLWE avoids the intensive computations of ratios and does not invoke hypothesis testing.

A particularly interesting recent study on learning user's interests in ambient media services (and in, consequently, locating relevant services) was reported in [8]. 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 [7, 8] by recording the so-called "scores," which represented his/her affinity of interests. In order to closely follow the evolution of the scores, the authors of [8] refined their proposed updating method defined earlier in [7] 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 [7].

We shall now discuss the family of weak estimators that were alluded to earlier, and proceed to explain how they can be used to solve the problem currently being studied.

3. Weak Estimators of Multinomial Distributions

The problem of estimating the parameters of a multinomial distribution has been efficiently solved by the recently introduced SLWE [23]. The multinomial distribution is characterized by two parameters, namely, the number of trials and a probability vector that determines the probability of a specific event (from a prespecified set of events) occurring. In this regard, we assume that the number of observations is the number of trials. Therefore, the problem is to estimate the latter probability vector associated with the set of possible outcomes or trials. Thus, we encounter the problem of estimating the latter probability vector that is associated with the set of possible outcomes.

Specifically, let X be a multinomially distributed random variable, which takes on values from the set $\{`1',\ldots,`r'\}$. We assume that X is governed by the distribution $S=[s_1,\ldots,s_r]^T$ as follows:

$X=`i'$ with probability s_i, where $\sum_{i=1}^{r} s_i = 1$.

Also, let $x(n)$ be a concrete realization of X at time `$n$'. The intention of the exercise is to estimate $S$, i.e., $s_i$ for $i=1,\ldots,r$. We achieve this by maintaining a running estimate $P(n)=[p_1(n),\ldots,p_r(n)]^T$ of $S$, where $p_i(n)$ is the estimate of $s_i$ at time `n', for $i=1,\ldots,r$. We omit the reference to time `n' in $P(n)$ whenever there is no confusion. Then, the value of $p_i(n)$ is updated as per the following simple rule (the rules for other values of $p_j(n), j \neq i$, are similar):

$$p_i(n+1) \leftarrow p_i + (1-\lambda)\sum_{j\neq i} p_j \quad when\ x(n) = i \tag{2}$$

$$\leftarrow \lambda p_i \qquad when\ x(n) \neq i. \tag{3}$$

The properties[2] of the estimator are catalogued below.

Theorem 1 *Let the parameter* S *of the multinomial distribution be estimated by* $P(n)$ *at time* `n' *as per equations (2) and (3). Then,* $E\left[P(\infty)\right] = S$.

Remarks: Theorem 1 explicitly states that rules (2) and (3) lead to a mean probability vector that asymptotically converges to the actual unknown probabilities. Although this behavior is asymptotic, we will empirically show presently that the rule quickly adapts to the changes in the distribution and, hence, works efficiently in non-stationary environments.

Proof. First of all, we can rewrite the updating rules given in equations (2) and (3) as follows:

$$p_i(n+1) \leftarrow p_i + (1-\lambda)(1-p_i)\ w.p. s_i \tag{4}$$

$$\leftarrow \lambda p_i \qquad w.p. \sum_{j\neq i} s_j. \tag{5}$$

Thus, the expected value of $p_i(n+1)$ given the estimated probabilities at time `n', P, is:

$$E\left[p_i(n+1) \mid P\right] = p_i s_i + (1-\lambda - p_i + \lambda p_i)s_i + \lambda p_i(1-s_i) \tag{6}$$

$$= p_i s_i + s_i - \lambda s_i - p_i s_i + \lambda p_i s_i + \lambda p_i - \lambda p_i s_i \tag{7}$$

$$= (1-\lambda)s_i + \lambda p_i. \tag{8}$$

Taking expectations a second time, we have:

$$E[p_i(n+1)] = (1-\lambda)s_i + \lambda E[p_i(n)]. \tag{9}$$

As $n \rightarrow \infty$, both equations $E\left[p_i(n+1)\right]$ and $E\left[p_i(n)\right]$ converge[3] to $E\left[p_i(\infty)\right]$, and

[2] The theoretical properties of the SLWE have been proven elsewhere [23] and are not a fundamental contribution of this paper. They have been included here with the consent of the Editor, and in the interest of this paper being a stand-alone publication.

[3] Observe that $\mathrm{E}[p_i]$ converges to a limit because the multiplying factor of the resultant linear difference equation is λ, which is both positive and strictly less than the unity.

hence we can write:

$$E[p_i(\infty)](1-\lambda) = (1-\lambda)s_i \tag{10}$$

$$\Rightarrow E[p_i(\infty)] = s_i. \tag{11}$$

The result follows since (11) is valid for every component p_i of P.

We now derive the explicit dependence of $E\left[P(n+1)\right]$ on $E\left[P(n)\right]$ and the consequences.

Theorem 2 *Let the parameter* S *of the multinomial distribution be estimated at time* `n ' *by* $P(n)$ *obtained by equations (2) and (3). Then,* $E\left[P(n+1)\right] = \mathbf{M}^T E\left[P(n)\right]$, *in which every off-diagonal term of the stochastic matrix,* $\mathbf{M}$, *has the same multiplicative factor,* $(1-\lambda)$. *Furthermore, the final solution of this vector difference equation is independent of* λ.

Remark: Theorem 2 states that the rules given by equations (2) and (3) are governed by a Markovian phenomenon in which the stochastic matrix has the same multiplicative factor and the solution to the final equation is independent of the parameter used in the algorithm. However, as we see below and in the empirical results, by selecting the parameter λ, the scheme leads to extremely good results. The convergence and eigenvalue properties of M follow.

Proof. From equation (8), we can write the conditional expected probability, $E\left[p_1(n+1) \mid P\right]$ as follows:

$$E\left[p_1(n+1) \mid P\right] = (1-\lambda)s_1 \sum_{j=1}^{r} p_j + \lambda p_1. \tag{12}$$

Similarly, for all other conditional expectations of $p_i(n+1)$, we have:

$$E\left[p_i(n+1) \mid P\right] = (1-\lambda)s_i \sum_{j=1}^{r} p_j + \lambda p_i. \tag{13}$$

Organizing the terms of (13) in a vectorial manner for all $i = 1,\ldots,r$, it can be seen that $E\left[P(n+1)\right] = \mathbf{M}^T E\left[P(n)\right]$, where the stochastic matrix $\mathbf{M}$, is:

$$\mathbf{M} = \begin{bmatrix} (1-\lambda)s_1+\lambda & (1-\lambda)s_2 & \cdots & (1-\lambda)s_r \\ (1-\lambda)s_1 & (1-\lambda)s_2+\lambda & \cdots & (1-\lambda)s_r \\ \vdots & \vdots & \ddots & \vdots \\ (1-\lambda)s_1 & (1-\lambda)s_2 & \cdots & (1-\lambda)s_r+\lambda \end{bmatrix}.$$

The limiting solution for $E\left[P(n)\right]$ is obtained by solving the vectorial difference equation, and taking the limit as n is increased to infinity[4].

$$E\left[P(\infty)\right] = \mathbf{M}^T E\left[P(\infty)\right].$$

To solve the above, we observe that every element of the matrix $\left(\mathbf{I}-\mathbf{M}\right)$ contains the term $(1-\lambda)$. By invoking this property and doing some straightforward algebraic manipulations, it turns out that:

$$E\left[P(\infty)\right] = S,$$

and the theorem is proved.

Theorem 3 *Let the parameter S of the multinomial distribution be estimated at time `n' by $P(n)$ obtained by equations (2) and (3). Then, all the non-unity eigenvalues of $\mathbf{M}$ are exactly λ, and thus the rate of convergence of P is fully determined by λ.*

Proof. To analyze the rate of convergence of the vector difference equation, we first find the eigenvalues of $\mathbf{M}$, namely $\xi_1, \xi_2, \ldots, \xi_r$. Without going into the algebraic details, it can be shown that $\mathbf{M}$ can be expressed as follows:

$$\mathbf{M} = \mathbf{FLF}^{-1}, \tag{14}$$

where:

$$\mathbf{F} = \begin{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} & \begin{bmatrix} -\frac{s_2}{s_1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} -\frac{s_3}{s_1} \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} & \cdots & \begin{bmatrix} -\frac{s_r}{s_1} \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \end{bmatrix}, \tag{15}$$

are the eigenvectors of $\mathbf{M}$, and $\mathbf{L} = diag(1, \lambda, \lambda, \ldots, \lambda)$, which contains the eigenvalues of $\mathbf{M}$. Thus, $\xi_1 = 1$, and $\xi_i = \lambda$ for $i = 2, \ldots, r$.

Consequently, the rate of convergence of the matrix determining the vector difference equation is *fully* determined by the second largest eigenvalue, which is λ, (since λ is an eigenvalue of multiplicity $r-1$). The result follows as a consequence of these arguments.

A small value of λ leads to fast convergence and a large variance. On the contrary, a large value of λ leads to slow convergence and a small variance. Although the derived results are asymptotic, and thus, are valid only as $n \to \infty$, realistically, and for all practical purposes, the convergence takes place after a relatively small value of n. If λ is even as "small" as 0.9,

[4] The solution exists since $\mathbf{M}$ has one eigenvalue, which is unity, and all the other eigenvalues of $\mathbf{M}$ are strictly less than the unity. We shall show this fact presently.

after 50 iterations, the variation from the asymptotic value will be of the order of 10^{-50}, because λ also determines the rate of convergence, which again, occurs in a geometric manner. In other words, even if the environment switches its multinomial probability vector after 50 steps, the SLWE will be able to track this change. Earlier experimental results reported in [23], as well as our current experimental results, demonstrate this fast convergence.

4. SLWE-BASED SOLUTION TO THE ADAPTATION OF A USER'S DRIFT IN INTERESTS

In this section, we devise a Learning Preferences Manager, which takes advantage of the SLWE updating scheme presented in Section 3, so as to accurately estimate the user's interest affinity in non-stationary environments. First, we will present our adapted model, as it pertains to the presentation of the user's profile. After that, we introduce two profile update methodologies based on whether the data items attached to an attribute are disjunctive or conjunctive.

4.1 Profile Representation

An essential element of the Learning Preference Manager is the Profile Representation. For instance, a possible representation model for a user's interests can be in terms of the topic hierarchies [6, 16]. We adopt the Profile Representation Model advocated by Hossain and his co-authors in [7, 8]. 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 [7, 8] is similar to that of [33], in the sense that it is based on $<\text{feature}, \text{weight}>$ pairs, except that in [7, 8], the authors have invoked a normalized score for the data items. We shall first briefly present the Profile Representation Model reported in [7, 8].

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 a type of movie, the set of possible attributes could be (movie genre, director name, etc.). An attribute, in turn, possesses a set of *data items*. For example, if the movie attribute “genre” has two data items, namely “action” and “comedy," a vector associated with the attribute (comedy affininity=0.7, action affininity=0.3) reflects that the user likes comedy movies more than action movies, with a relative weighting of 0.7 to 0.3. The update of the weights of the data items for a particular attribute is done in an incremental manner.

4.2 Profile Updating Method

In the quest to learn the user's dynamic profile, the Learning Preferences Manager is guided by so-called Relevance Feedback (RF) [18]. In this paper, we rely on the Service Usage History (analogous to the history maintained by the authors of [7, 8]) 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 [23] explained in the previous section, so as to update the score of the data item based on the usage history. Whenever a user selects a service, the metadata describing the service is used to update the score of 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 [7, 8] 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 (e.g., that which is needed to locate relevant information).

At this juncture, we distinguish two classes of data items that, in turn, require two different forms of update mechanisms. In fact, the data items related to a given attribute could be either semantically **disjunctive** or semantically **conjunctive**. We illustrate what we mean by the latter concepts by alluding to two simple examples.

4.2.1 Profile Update for Disjunctive Data Items

Data items of a particular attribute are said to be *disjunctive* if every service usage history can only be instantiated with the exclusive realization of one of the data items at a time. To illustrate the idea in simpler terms, consider the example of learning a user's preferences when it concerns a type of service 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 then becomes a suitable choice for managing the time-varying preferences.

It is crucial for the reader to observe that the SU approach presented in [7, 8] deals only with this specific case (e.g., disjunctive data items).

4.2.2 Profile Update for Conjunctive Data Items

Data items of a particular attribute are said to be *conjunctive* whenever every service usage history can be instantiated with one or more data items at a time. To illustrate this, consider the example of the service usage history corresponding to the services for movies. The attribute movie genre is associated with the data item set $S_{genre} = \{action, romantic, comedy, horror\}$. The latter data items are conjunctive (not disjunctive) in the sense that a movie's genre can be described with more than a single data item at a time. For instance, a movie genre could be “romantic” and “action packed” at the same time. Suppose that the user watches a movie that belongs to the genres "action" and "romance" at a given time instant \`$n$'. In this case, the weights of both the data items "action" and "romance" can be increased at time `$n+1$'.

In this case, a multinomial SLWE will not be able to update the different weights of the data items because it is not designed to increase the weights of more than a component at a time. Thus, a different methodology for updating the weights of the data items is needed, where more than a single data item's weight can be incremented at a time. To solve the problem, we propose to attach a binomial SLWE to each data item instead of having a multinonmial probability vector for each attribute, as in the case of disjunctive data items. In other words, a binomial probability vector will be attached to each of the data items in S_{genre}.

For the sake of clarity, we consider the above-mentioned example and describe the update at the subsequent instant `$n+1$' of each binomial probability vector as:

$$p_{action}(n+1) \leftarrow 1-\lambda(1-p_{action}(n)) \tag{16}$$

$$p_{romantic}(n+1) \leftarrow 1-\lambda(1-p_{romantic}(n)) \tag{17}$$

$$p_{comedy}(n+1) \leftarrow \lambda p_{comedy}(n) \tag{18}$$

$$p_{horror}(n+1) \leftarrow \lambda p_{horror}(n) \tag{19}$$

Once these binomial-based computations have been achieved, we then resort to an additional computation in order to normalize the weights of each data items. The normalization is, quite simply, given by: For $k \in S_{genre} = \{action,\ romantic,\ comedy,\ horror\}$

$$W_k(n+1) = \frac{p_k(n+1)}{\sum_{j \in S} p_j(n+1)} \tag{20}$$

Consequently W_k tracks, with a SLWE-philosophy, the ratio of the number of times the particular data items ($k \in S_{genre} = \{action,\ romantic,\ comedy,\ horror\}$) of the particular attribute (movie's genre) appears in the service usage within a given number of usage records to the total number of occurrences of the data items of S_{genre}.

In order to model this in a tangible (or realistic) way, we suppose that the occurrence of each data item in the usage history is controlled by a binomial distribution. We further suppose, that the occurrence of the data items are independent of each other. Let s_k be the binomial parameter that describes the occurrence of data item k in the usage history, where $k \in S_{genre} = \{action,\ romantic,\ comedy,\ horror\}$. With these assumptions, based on the results of the previous subsection, we easily derive the asymptotic weight:

$$E[W_k(\infty)] = \frac{s_k}{\sum_{j \in S} s_j}. \tag{21}$$

It is worth noting that whenever the data items corresponding to a given attribute are disjunctive, it is computationally more efficient (although only marginally) to employ a multinomial SLWE, instead of a set of binomial SLWEs.

4.2.3 Modelling Changes in the Interests

We suppose that at every time instance denoted by `n', the Learning Preferences Manager is fed by a service usage instance. We further assume that the distribution of the user's interests, which are 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. Furthermore, in the case of conjunctive data items, the binomial distribution attached to every data item switches to a possibly new value.

5. Experimental Results

To verify our computational model and our proposed solution, we have performed extensive simulations. However, in the interest of space and brevity, we report here only a subset of these results. We emphasize though that these results are both representative and typical. The obtained experimental results are conclusive, and demonstrate that our SLWE-based update schemes, when applied to tracking users' interests, outperforms the GF approach, the sliding window, and the SU.

In order to model the changes in the interests' distribution, we assume that at any given time instant the distribution of the user's preferences changes with probability 0.02. This implies that on average, a change occurs every 50 time instants. The reader should observe that our experimental results are based on synthetic data due to fact that it is difficult (if not, impossible) to obtain real-life data that describes a 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 that we have used to artificially indicate the changes in the user's interest distributions is strong enough to mimic real-life settings.

5.1 Disjunctive Data Items

To study the case of disjunctive data items, we assume that we are dealing with estimating the evolving user's interests' weights of data items of this type, namely, those which are associated with a given attribute. In the interest of completeness, we will present separate experimental results for the binomial and the multinomial cases.

5.1.1 Binomial Distribution

In this scenario, we assume that we are dealing with an attribute that possesses two data items. The problem of estimating the user's interests in this particular case now reduces to that of estimating the parameters of the corresponding binomial random variables. As mentioned above, numerous experiments were performed, although in the interest of brevity, we report here only on the results from four of these. Moreover, in order to render the comparison meaningful, we have simultaneously followed the GF and SU computation, and in each case we have utilized identical data streams, as in the case of the SLWE. Furthermore, in each case, the estimation algorithms were presented with random occurrences of the variables for $n = 400$ time instances. In the case of the SLWE, the true underlying value of s_1 was randomly assigned for the first step, and modified at random instants (determined with a switching probability of 0.02) using values drawn from a random variable that was uniformly distributed in $[0,1]$.

In order to demonstrate the superiority of the SLWE over the GF, which uses a sliding window, and the SU, we report the respective averages on an ensemble of experiments. The same experiment was repeated $1,000$ times with distinct random sequences, and the ensemble average at every time step was recorded. Clearly, by doing this, the variations of the estimates would be much smoother. In order to perform a fair evaluation, we adopted the same comparison approach as in [23], where the value of λ for the SLWE and the size of the window were randomly generated from uniform distributions in $[0.55,0.95]$ and $[20,80]$, respectively. The value of the "gradual forgetting" factor, k, for the GF was chosen to be 0.4 as suggested in [12].

The plots of the estimated probability p_1 for the SLWE, the GF, and SU for four cases are shown in Figures 1, 2, 3, and 4, where the values of λ are 0.763, 0.811, 0.811, and 0.563, and the sizes of the windows are 29, 46, 57, and 68, respectively.

Observe that both the GF and SU follow s_1 quite exactly prior to the first distribution change, but they are thereafter severely handicapped in tracking the variations. The weakness of the GF is accentuated in the cases in which the size of the window is larger, (e.g., 68). In contrast, the SLWE adjusts to the changes much more rapidly, as expected, in a geometric manner. In Table 1, we report the error rate associated with the four experiments (e.g., those plotted in Fig. 1, 2, 3, and 4).

Table 1. The effects of varying the window size and the updating parameter on the error of p_1 for the various schemes investigated for disjunctive data items

Figure No.	Error rate: SLWE	Error rate: GF	Error rate: MLE	Error rate: SU
Figure 1	0.0262	0.0864	0.1005	0.2894
Figure 2	0.0347	0.1418	0.1654	0.2713
Figure 3	0.0737	0.1364	0.1514	0.2813
Figure 4	0.0330	0.0779	0.0901	0.1561

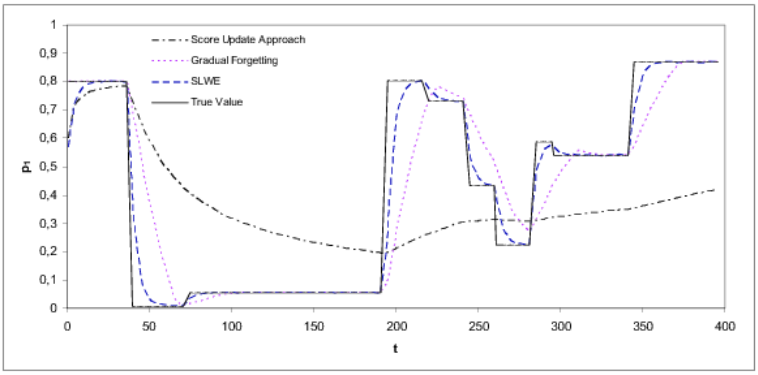


Fig. 1. Plots of the expected values of $p_1(n)$, at time `n' for disjunctive data items, which were estimated by using the SLWE, the GF, and the SU, in which the corresponding parameters $\lambda = 0.736$ and the window size is 29

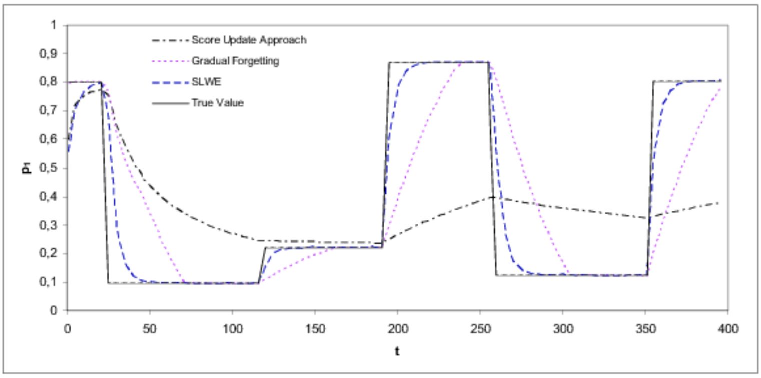


Fig. 2. Plots of the expected values of $p_1(n)$, at time `n' for disjunctive data items, which were estimated by using the SLWE, the GF, and the SU, in which the corresponding parameters $\lambda = 0.811$ and the window size is 46

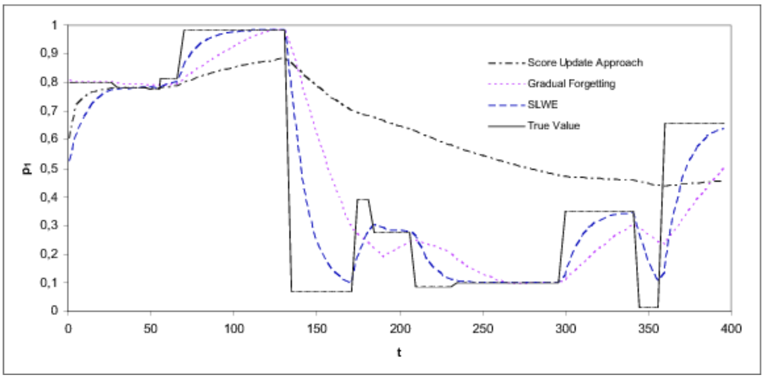


Fig. 3. Plots of the expected values of $p_1(n)$, at time `n' for disjunctive data items, which were estimated by using the SLWE, the GF and the SU, in which the corresponding parameters $\lambda = 0.912$ and the window size is 57

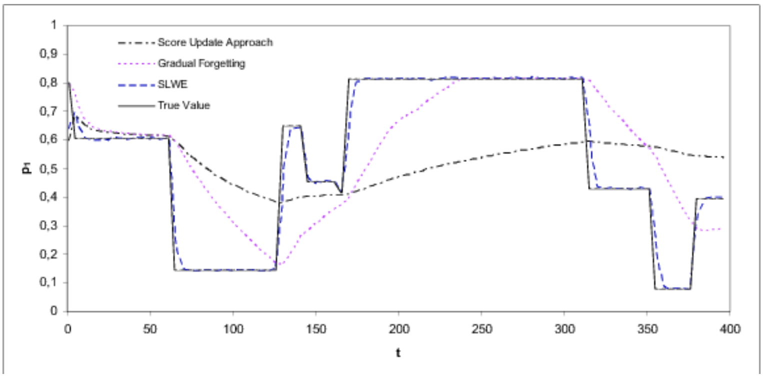


Fig. 4. Plots of the expected values of $p_1(n)$, at time `n' for disjunctive data items, which were estimated by using the SLWE, the GF, and the SU, in which the corresponding parameters $\lambda = 0.563$ and the window size is 68

We also include the error rate for the MLE with the same sliding window sizes that were used for the GF experiments. We can clearly observe that the SLWE yields a lower error rate than the other three approaches, namely the GF, the SU, and the MLE which uses a sliding window,. An additional remark confirming the results of [12] is the following: the GF augmented with a sliding window exhibits a lower error rate than the MLE with a sliding window. Thus, this confirms that the GF presents an enhancement to the basic sliding window approach.

5.1.2 Multinomial Distribution

We have also performed simulations for the case of disjunctive data items, where the user's interests follow a multinomial distribution, and where the parameters were estimated by following the SLWE, the GF. and the SU. We considered a multinomial random variable, X, which can take any of four different values, namely `1', `2', `3' or `4', whose characterizing parameters changed (randomly) at random time instants. As in the binomial case, we ran the estimators for 400 steps, repeated this $1,000$ times, and then computed the corresponding ensemble averages. For each experiment, we computed $\|P-S\|$, the Euclidean distance between P and S, which we reckoned as a measure of how good our estimate, P, was of S. The plots of the latter distance obtained from the SLWE, the GF, and the SU are depicted in Fig. 5, 6, 7, and 8, where the values of λ were 0.908, 0.903, 0.952, and 0.948, and the sizes of the windows were 35, 44, 63, and 76, respectively. The values for λ and the window size were obtained randomly from a uniform distribution in $[0.9, 0.99]$ and $[20, 80]$ respectively.

From these figures, we observe that the GF, the SU, and the SLWE converge to zero relatively quickly prior to the first instant when the distribution changes. However, this behavior is not present for subsequent (successive) distribution “switches." Rather, we notice that the GF is capable of tracking the changes of the parameters when the size of the window is small, or at least smaller than the intervals of constant probabilities. It is, however, not able to track the changes properly when the window size is relatively large. Since neither the magnitude nor the instants of the changes is known *a priori*, this scenario demonstrates the weakness of the GF, and its dependence on the knowledge of the input parameters. Again, such observations are typical.

In Table 2, we report the error rates associated with the experiments plotted in Figures 5, 6, 7,

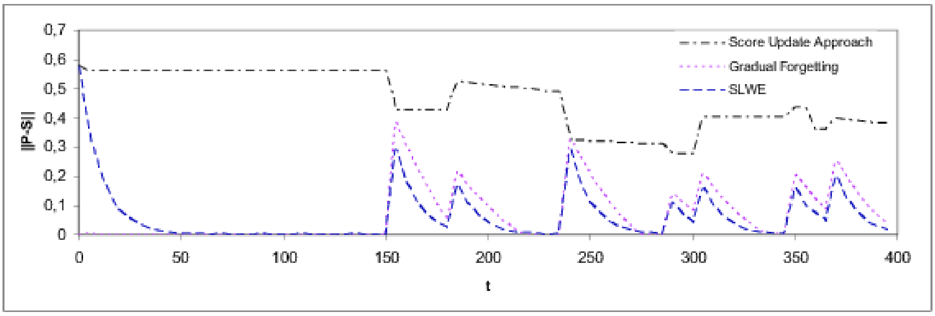


Fig. 5. Plot of the Euclidean norm $\|P-S\|$ (the Euclidean distance between P and S) for disjunctive data items for the SLWE, the GF, and the SU, where $\lambda = 0.908$ and w = 35

and 8. In Table 2, we also include the error rates for the MLE augmented with a sliding window. One can clearly observe that the SLWE exhibits a lower error rate than the GF, the SU, and the MLE.

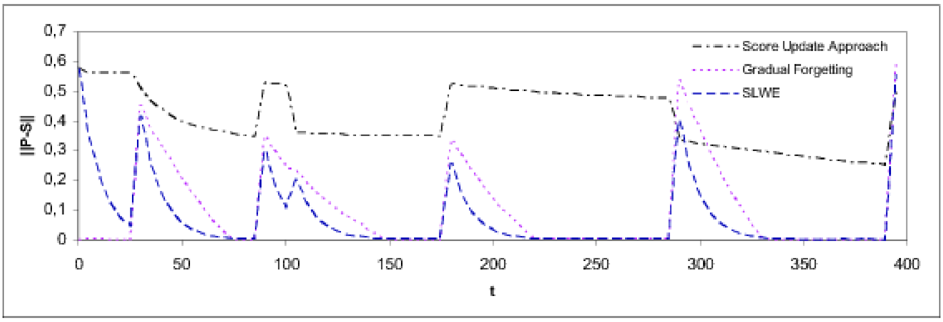


Fig. 6. Plot of the Euclidean norm $\|P-S\|$ (the Euclidean distance between P and S) for disjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.903$ and w = 44

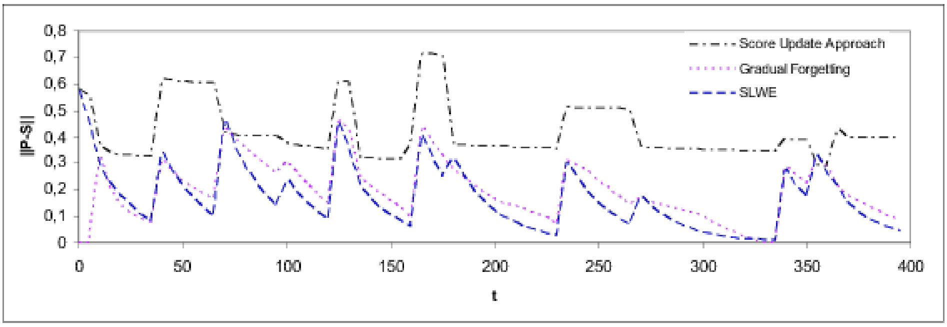


Fig. 7. Plot of the Euclidean norm $\|P-S\|$ (the Euclidean distance between P and S) for disjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.952$ and w = 63

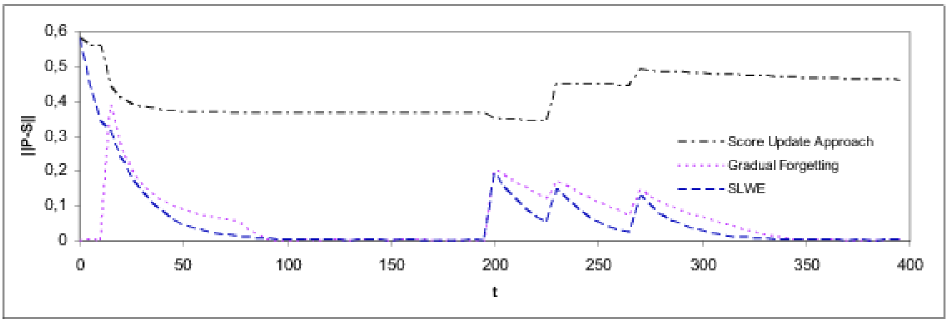


Fig. 8. Plot of the Euclidean norm $\|P-S\|$ (the Euclidean distance between P and S) for disjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.948$ and w = 76

Table 2. The effects of varying the window size and the updating parameter on the error rates for the various schemes investigated for disjunctive data items

Figure No.	Error rate: SLWE	Error rate: GF	Error rate: SW	Error rate: SU
Figure 5	0.0612	0.0724	0.0836	0.4606
Figure 6	0.0665	0.1006	0.1152	0.4037
Figure 7	0.1601	0.1893	0.2074	0.4175
Figure 8	0.0507	0.0567	0.0672	0.4165

5.2 Conjunctive Data Items

In this subsection, we present results from simulations that are related to conjunctive data items. We assume that for each data item, the probability of appearing in a given record history is binomial. In the experiments, we considered 4 data items and assumed that at random time instants the binomial distribution of each of the 4 data items changed, and that they did so randomly. We are interested in estimating the weights of the data items that reflect the respective ratios of their appearances in the usage history. As alluded to previously, W_i tracks the ratio of appearance of data item i in the usage history[5].

As in the previous experiments, we computed $\|P-S\|$, the Euclidean distance between P and S, which was used as a measure of how good our estimate, P, was of S. The plots of the latter distance obtained from the SLWE, the GF, and the SU are depicted in Fig. 9, 10, 11, and 12 where the values of λ were 0.579, 0.627, 0.771 and 0.809 respectively, and the sizes of the windows were 33, 39, 55, and 70, respectively. Here the value of λ for the SLWE and the size of the window were randomly generated from uniform distributions in $[0.55, 0.95]$ and $[20, 80]$ respectively.

In Table 3, we report the error rate associated with the experiments whose results were plotted in Fig. 9, 10, 11, and 12. We have also included the error rates for the MLE augmented with the sliding window in Table 3. In all brevity we can state that the results are conclusive: the SLWE exhibits a lower error rate than the GF, the SUm and the MLE in all the different settings.

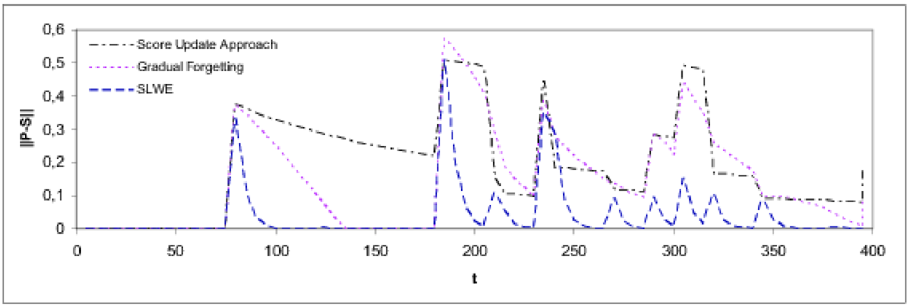


Fig. 9. Plot of the Euclidean norm $\|P-S\|$ (the Euclidean distance between P and S) for conjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.579$ and w = 33

[5] Since we estimate the quantity W in terms of P, we plot the variation of $P(n)$ instead of the variation of $W(n)$ in each of the graphs below.

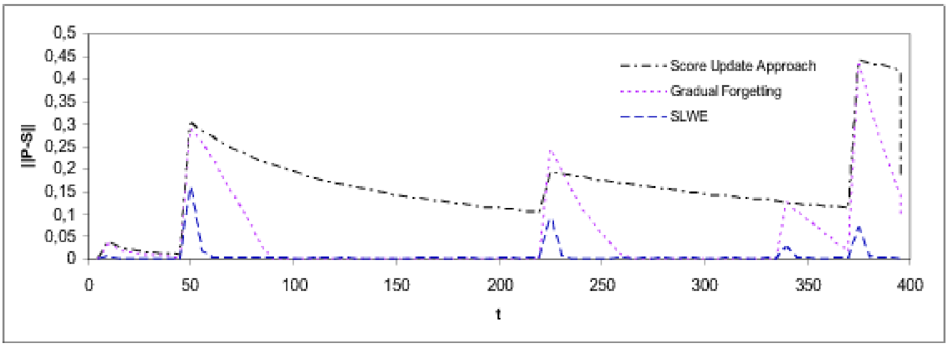


Fig. 10. Plot of the Euclidean norm $\|P-S\|$ (the Euclidean distance between P and S) for conjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.627$ and w = 39

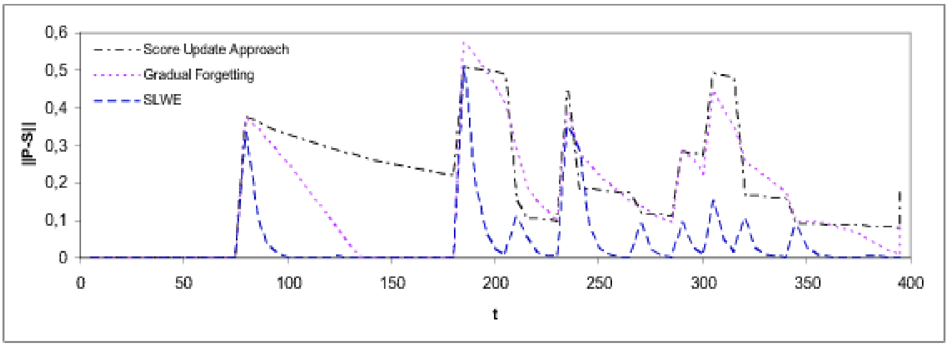


Fig. 11. Plot of the Euclidean norm $\|P-S\|$ (the Euclidean distance between P and S) for conjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.771$ and w = 55 respectively

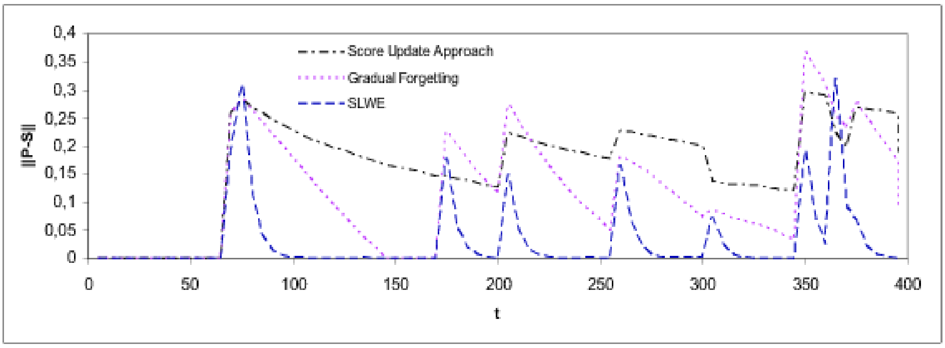


Fig. 12. Plot of the Euclidean norm $\|P-S\|$ (the Euclidean distance between P and S) for conjunctive data items, for the SLWE, the GF and the SU, where $\lambda = 0.809$ and w = 70

Table 3. The effects of varying the window size and the updating parameter on the error rates for the various schemes investigated for conjunctive data items

Figure	Error rate: SLWE	Error rate: GF	Error rate: MLE	Error rate: SU
Figure 9	0.0181	0.1030	0.1151	0.1556
Figure 10	0.0071	0.0533	0.0614	0.1609
Figure 11	0.0371	0.1412	0.1524	0.1864
Figure 12	0.0259	0.1158	0.1281	0.1633

6. CONCLUSION

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. The results of simulations based on synthetic data demonstrate 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 the learned profiles in order to perform efficient matchmaking between available services and the user's profile is a potential avenue for future research, for which we do indeed have some very promising initial results.

REFERENCES

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

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

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

[4] U. Bud and J. Lim, "*Distance functions to detect changes in data streams*",*JIPS*, Vol.2, No.1, 2006, pp.44-47.

[5] D. Godoy and A. Amandi, “*User profiling in personal information agents: a survey*”, *Knowl. Eng. Rev.*, Vol.20, 2005, pp.329-361.

[6] D. Godoy, S. Schiaffino, and A. Amandi, “*Interface agents personalizing web-based tasks*”, *Cognitive Systems Research, Special Issue on Intelligent Agents and Data Mining for Cognitive Systems*, Vol.5, No.3, 2004, pp.207-222.

[7] M. A. Hossain, P. K. Atrey, and A. El Saddik, “*Gain-based selection of ambient media services in pervasive environments*”,*Mob. Netw. Appl.*, Vol.13, No.6, 2008, pp.599-613.

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

[9] Y. Kim, W. Kim, and U. Kim, “*Mining frequent itemsets with normalized weight in continuous data streams*”, *JIPS*, Vol.6, No.1, 2010, pp.79-90.

[10] 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.

[11] 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.

[12] 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.

[13] S. Kurkovsky and K. Harihar, “*Using ubiquitous computing in interactive mobile marketing*”, *Personal Ubiquitous Comput.*, Vol.10, No.4, 2006, pp.227-240.

[14] J.-W. Lee and Y.-J. Lee, “*A knowledge discovery framework for spatiotemporal data mining*”, *JIPS*, Vol.2, No.2, 2006, pp.124-129.

[15] M. A. Maloof and R. S. Michalski, “*Selecting examples for partial memory learning*”, *Machine Learning*, Vol.41, 2000, pp.27-52.

[16] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, “*Ontological user profiling in recommender systems*”, *ACM Trans. Inf. Syst.*, Vol.22, No.1, 2004, pp.54-88.

[17] T. M. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, “*Experience with a learning personal assistant*”, *Commun. ACM*, Vol.37, No.7, 1994, pp.80-91.

[18] M. Montaner, B. López, and J. L. de la Rosa, “*A taxonomy of recommender agents on the internet*," *Artificial Intelligence Review*, Vol.19, 2003, pp.285-330.

[19] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*. 1em plus Prentice Hall, 1989.

[20] Y. Naudet, A. Aghasaryanb, S. Mignon, Y. Toms, and C. Senot, “*Ontology-based profiling and recommendations for mobile tv*," in *Semantics in Adaptive and Personalized Services*, ser. Studies in Computational Intelligence, M. Wallace, I. Anagnostopoulos, P. Mylonas, and M. Bielikova, Eds. Springer, Heidelberg, 2010, Vol.279, pp.23-48.

[21] I. Ong and H. Lim, “*Dynamic load balancing and network adaptive virtual storage service for mobile appliances*”, *JIPS*, Vol.7, No.1, 2011, pp.53-62.

[22] 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, 2010, pp.159-169.

[23] 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, 2006, pp.328-341.

[24] 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, 2006, pp.1196-1200.

[25] A. Saeed, M. Faezipour, M. Nourani, S. Banerjee, G. Lee, G. Gupta, and L. Tamil, “*A scalable wireless body area network for bio-telemetry*”, *JIPS*, Vol.5, No.2, 2009, pp.77-86.

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

[27] A. Stensby, B. J. Oommen, and O.-C. Granmo, “*Language detection and tracking in multilingual*

documents using weak estimators", in *Proceedings of the 2010 joint IAPR international conference on Structural, syntactic, and statistical pattern recognition*, ser. SSPR&SPR'10, 2010, pp.600-609.

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

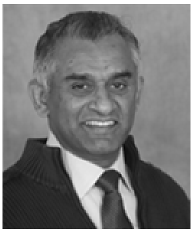
[29] G. Widmer, "*Tracking context changes through meta-learning*", *Mach. Learn.*, Vol.27, No.3, 1997, pp.259-286.

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

[31] A. Yazidi, O.-C. Granmo, B. J. Oommen, M. Gerdes, and F. Reichert, "*A user-centric approach for personalized service provisioning in pervasive environments*", *Submitted for Publication.*

[32] F. Yu, D. Oyana, W.-C. Hou, and M. Wainer, "*Approximate clustering on data streams using discrete cosine transform*", *JIPS*, Vol.6, No.1, 2010, pp.67-78.

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



Dr. John Oommen

He was born in Coonoor, India on September 9, 1953. He obtained his B.Tech. degree from the Indian Institute of Technology in Madras, India in 1975. He obtained his M.E. from the Indian Institute of Science in Bangalore, India in 1977. He then went on for his M.S. and Ph. D., which he obtained from Purdue University in West Lafayette, Indiana in 1979 and 1982 respectively. He joined the School of Computer Science at Carleton University in Ottawa, Canada in the 1981-82 academic year. He is still at Carleton and holds the rank of a full Professor. Since July 2006, he has been awarded the honorary rank of *Chancellor's Professor*, which is a lifetime award from Carleton University. His research interests include Automata Learning, Adaptive Data Structures, Statistical and Syntactic Pattern Recognition, Stochastic Algorithms and Partitioning Algorithms. He is the author of more than 375 refereed journal and conference publications, and is a fellow of the IEEE and a fellow of the IAPR. Dr. Oommen has also served on the Editorial Board of the IEEE Transactions on Systems, Man and Cybernetics, and Pattern Recognition.

Anis Yazidi

Anis Yazidi obtained his M.Sc. in 2008 and Ph.D. degree in 2008, both from thr University of Agder in Norway. He is currently a researcher in Teknova, Norway. His research interests include Machine Learning, Learning Automata, Machine Vision, and Artificial Intelligence and its industrial applications.

Ole-Christoffer Granmo

He was born in Porsgrunn, Norway. He obtained his M.Sc. in 1999 and Ph.D. in 2004, both from the University of Oslo in Norway. He is currently a Professor in the Department of ICT at the University of Agder in Norway. His research interests include Intelligent Systems, Stochastic Modelling and Inference, Machine Learning, Pattern Recognition, Learning Automata, Distributed Computing, and Surveillance and Monitoring. He is the author of more than 70 refereed journal and conference publications.