Adaptation to Drifting User's Interests

Ivan Koychev and Ingo Schwab

GMD FIT.MMK, D-53754 Sankt Augustin, Germany, phone: +49 2241 14 2194 fax: +49 2241 14 2146, {Ivan.Koychev,Ingo.Schwab}@gmd.de

Abstract

In recent years, many systems have been developed which aim at helping users to find pieces of information or other objects that are in accordance with their personal interests. In these systems, machine learning methods are often used to acquire the user interest profile. Frequently user interests drift with time. The ability to adapt fast to the current user's interests is an important feature for recommender systems. This paper presents a method for dealing with drifting interests by introducing the notion of gradual forgetting. Thus, the last observations should be more "important" for the learning algorithm than the old ones and the importance of an observation should decrease with time. The conducted experiments with a recommender system show that the gradual forgetting improves the ability to adapt to drifting user's interests. Experiments with the STAGGER problem provide additional evidences that gradual forgetting is able to improve the prediction accuracy on drifting concepts (incl. drifting user's interests).

Keywords:

Interest Drift, User Profiling, User Adaptivity, Recommender Systems

1 Introduction

In the last few years, many approaches and systems for recommending information, products and other items have been developed (e.g., (Billsus and Pazzani, 1999), (Lieberman, 1997), (Mladenic, 1996), (Pazzani and Billsus, 1997), etc.). These systems try to help users in finding pieces of information or other objects in which the users will presumably be interested (Kobsa, Koenemann and Pohl, 2000). There have been mainly two different approaches so far. Feature-based (aka "content-based") filtering systems take individual preferences for certain objects features of into account. Clique-based (aka "collaborative") filtering systems instead typically build on similarities between users with respect to the objects in which users implicitly or explicitly express an interest.

Recommender systems use machine learning methods for acquiring interest profiles. It is assumed that the set of information objects can be divided into classes (e.g., "interesting" and "not interesting"). In many systems users must provide examples for both classes in an initial training phase, on the basis of which a classification algorithm is learned inductively (e.g. (Pazzani and Billsus, 1997)). Thereafter, the classification algorithm can determine whether new information objects belong to the "interesting" or to the "not interesting" class. Such explicit rating requires additional user effort and keeps users from performing their real tasks, both of which is undesirable. According to observations by Carroll and Rosson (Carroll and Rosson 1987), users are unlikely to engage in such additional efforts even when they know that they would profit in the long run. Conclusions about user interests should therefore not only exclusively rely on user ratings, but also take observations about the user into account as far as possible.

We developed a content-based recommendation component (Schwab, Pohl and Koychev, 2000). In order to be unobtrusive, it learns individual interest profiles based on passive observations only. The central source of information about users' interests is the sequence of objects they selected. Selections are thereby made from the currently reachable information subspace, i.e. the set of selectable objects with common properties. In similar situations, other systems use heuristics to determine positive and negative evidence of the user's information interest (i.e. unselected objects are negative examples (Mladenic, 1996) (Lieberman, 1997)). However, we claim that unselected objects which are interesting to the user may exist (they may have just been overlooked or will perhaps be visited later). Classifying them as negative examples is dangerous and can cause too much noise in the training data. It is more suitable to take only selected objects as examples for the "interesting" class, and to disregard objects that have not been selected. Unfortunately, in this case standard classification methods are not applicable. Thus, for learning interest profiles we had to modify existing ones. In our project, a probabilistic approach and an instance-based learning approach have been used (Schwab, Pohl and Koychev, 2000). Both approaches have been modified to deal with a single class only by employing a notion of similarity or distance. However, it is difficult to use these learning results to characterize individual user's preferences explicitly, which is a desirable feature of user modeling systems (Kobsa, Koenemann and Pohl, 2000). Therefore, we developed a third mechanism that aims at selecting those features that are extraordinarily important to the user for identifying relevant objects, i.e. *explicit user's profile*. It turned out that this feature selection method additionally helps to improve the distance measure for instance-based learning. Moreover, feature selection can be combined with both probabilistic and instance-based learning to focus the learning task. The developed algorithms have been implemented and evaluated in a real-world application ELFI - a WWW-based information system that provides information about research grants. In this system, additional calls for proposals are recommended based on those the user had already browsed so far.

During the experiments we discovered that user interests are not permanent and can drift with time. This paper explains our approach, which copes with this problem. The related works about drifting user interests are mentioned in the next section. In a more general consideration the works dedicated to drifting concepts are relevant to the current topic of adapting to changing users interests. Hence, some important works about learning drifting concepts are also cited in the next section.

2 Related Works

A software assistant for scheduling meetings is described in (Mitchell et. al., 1994). It employs machine learning methods (e.g. induction on decision trees) to acquire assumptions about individual habits for arranging meetings. The learning method uses a time window (last 180 examples) to adapt faster to the shifting user's scheduling preferences. The newly generated rules are merged with old ones. The rules that perform poorly on the test set drop down the list.

A system, which learns user's interest profiles by monitoring web and e-mail habits, is presented in (Grabtree and Soltysiak, 1998). A clustering algorithm is used to detect user interests, which are then clustered to form interest themes. User profiles must also adapt to changing interests of the users over time. This research shows that user's interests can be tracked over time by measuring the similarity of interests in a time period.

An intelligent agent called NewsDude that is able to adapt to changing user's interests is presented in (Billsus and Pazzani, 1999). It learns two separate user models: one represents the user's short-term interests and the other one represents the user's long-term interests. The short-term model is learned from the most recent observations only. It represents user models, which can adjust more rapidly to the user's changing interests. If the short-term model cannot classify the story at all, it is passed on to the long-term model. The purpose of the long-term user model is to model the user's general preferences for news stories and compute predictions for stories that could not be classified by the short-term model.

STAGGER (Schlimmer and Granger, 1986) is an incremental learning system that dynamically tracks changes of concepts. STAGGER uses a connectionist representation scheme employing nodes to represent Boolean attributes and Bayesian-weighted connections to associate attribute nodes to a concept node. STAGGER learns and tracks changing concepts by adding new attribute nodes or adjusting the connection weights for the concept's connections.

The FLORA systems (Widmer and Kubat, 1996), which are also systems for coping with concept drifts, use a forgetting technique with an adaptive window size. They forget those examples, which are older, then a threshold. The window size and thus the rate of forgetting is supervised and dynamically adjusted by heuristics that monitor the learning process.

A method which can learn and track changing contexts, using meta-learning is presented in (Widmer, 1997). The assumption is that the domain provides explicit clues for the current context (e.g., attributes with characteristic values). A two-level learning algorithm is presented which effectively adjusts to changing contexts by trying to detect (via meta-learning) contextual clues and use this information to focus the learning process.

An offline meta-learning algorithm for identifying hidden contexts is presented in (Harries, Sammut and Horn, 1998). The approach assumes that concepts are likely to be stable for periods of time. It uses batch learning and contextual clustering to detect stable concepts and to extract a hidden context.

In (Maloof and Michalski, 1995) a method for selecting training examples for a partial memory learning system is described. Further development of that method is presented in (Maloof. and Michalski, 2000). The forgetting mechanisms of the method selects extreme examples that lie at the boundaries of concept descriptions and remove examples from the partial memory that are irrelevant or outdated for the learning task. The method uses a time-based forgetting function to remove examples from the partial memory, which are older than a certain age.

3 Interest Drift

It is assumed that a sequence of selected objects is given. The user aims at selecting documents, which are "interesting" for him. Therefore, user's selections are regarded as relatively independent tries, occurring over the time, to find interesting documents. As mentioned before, often the user's interests drift with time. Hence, the last observations represent the current user's interests better than older ones. The most often used approach to deal with this problem is the so-called time window. It learns the concept description only from the newest observations (e.g. only last l examples are used for training (Grabtree and Soltysiak, 1998), (Mitchell et. al., 1994)). An improvement of this approach is the use of heuristics to adjust the size of the window according to the current predictive accuracy of the system (Widmer and Kubat, 1996). The time-forgetting mechanism in (Maloof and Michalski, 2000)) uses a time-based function for aging the examples and the ones that are older than a certain age are forgotten. This approach totally "forgets" the observations that are outside the given window or older than certain age. The examples, which remain in the partial memory, are equally important for the learning algorithms. This is a complete forgetting of old information, which in some cases can be valuable. To avoid loss of useful knowledge, learned from old examples, some of these systems keep old rules as long as they are competitive to the new ones (Mitchell et. al. 1994).

Another approach is used in (Billsus and Pazzani, 1999) where a hybrid user model consisting of both a short-term and long-term model of the user's interests. The method employs the short-term model first. It is based on the most recent observations. If a story cannot be classified with the short-term model, the long-term model is used. This hybrid user model is useful in domains where the long-term user's interests are quite wide and short-term interests are changing fast as it is in the case of news stories.

In order to cope with the problem of interest drifts this paper suggests another approach. The main idea behind it is that the natural forgetting is a gradual process. Namely, the last observations should be more important than the old ones and the importance of an observation should decrease over time. To realize this we defined a gradual forgetting function $w = f(t)$, which is able to produce weights for each observation according to its arising time. A basic assumption for most of the learning algorithms is that all training examples are equally important. Therefore they should be modified to be able to utilize time-weighted examples.

In our methodology the feature selection plays a basic role (Schwab, Pohl and Koychev, 2000). Other algorithms use the results from the feature selection. Hence, the application of the gradual forgetting for the feature selection will effects both the explicit user profiles and system recommendations. For the feature selection it is necessary to count how often a feature j is appearing in the user's selections. To introduce the gradual forgetting this number is calculated by using the following formula $c_j = \sum_{i=1}^{n} w_i a_i^j$. i is a counter for observations starting from the most recent selection and it goes back over time; n is the length of the observed sequence of user actions; $a_i^j \in \{0,1\}$ are the feature values in the Boolean vectors which represent the observed user's selections; w_i are the weights calculated by the forgetting function for the observations. The calculated weights should be in an interval that is suitable for the used learning algorithm. For example it could be a linear function:

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

$k \in [0,1]$ is a parameter, which represents the percents of decreasing the weight of the first observation in comparison to the last one. By varying k the slope of the forgetting function can be adjusted.

4 Experiments

4.1 Experiments with Recommendations for ELFI Users

First experiments have been done with usage data from ELFI users. A detailed explanation of the used learning algorithms and its evaluation can be found in (Schwab, Pohl and Koychev, 2000). In this experiment the combination of feature selection and IBL with a weighted distance measure performs best. Therefore this combination was used in the recent experiments. In these experiments our goal was to investigate how the weights for gradual forgetting are able to influence both the generated explicit user profile and the predictive accuracy of recommendations. The explicit user profiles mainly include the features that represent recent observations and those, which characterise interests that are stable over time. The conducted experiments with recommendations were performed as follows: First for a given user *n* initial selections are used as training examples and then test on the next *n+1* example. Second the effect on a fixed time window is considered. In both cases the introduction of the gradual forgetting function (1) results in an improvement of the average predicting accuracy. The average improvement is about 2%. This improvement may not look very large, but we should take into account that the average predictive accuracy reaches 90% and in about 40% of the cases the system accuracy is 100%. Then the interests are stable and improvement is nearly impossible. When the user's interests change the prediction accuracy can drop down even less than 10%. After these gaps, the presented approach is able to adapt faster to the new user's interests.

4.2 Experiments with STAGGER Concepts

The presented approach for gradual forgetting was also tested on an artificial learning problem that was defined and used by Schlimmer and Granger for testing STAGGER (Schlimmer and Granger, 1986), one of the first concept drift tracking system. Many of the works dedicated to this problem used this data set for testing their systems

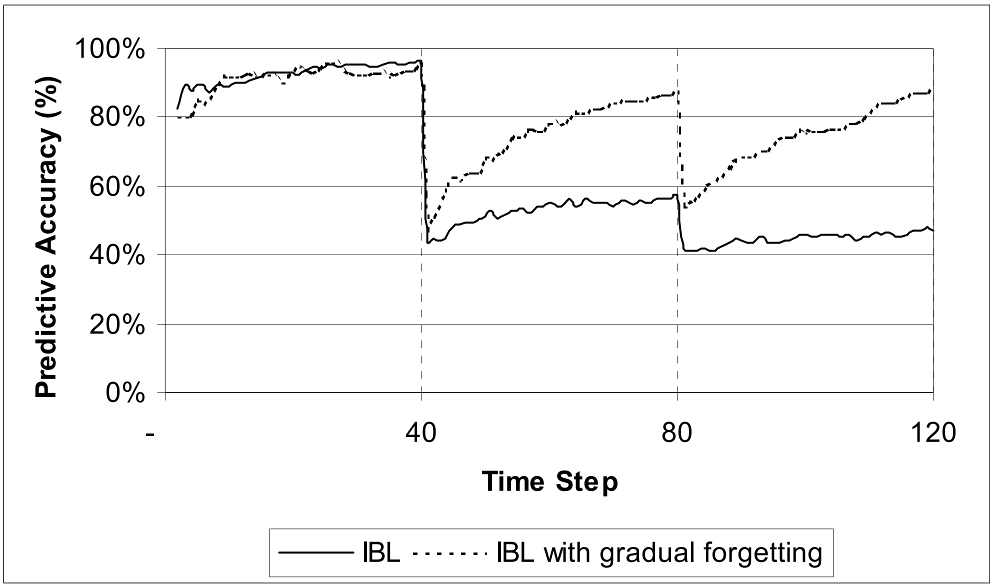

Figure 1. The STAGGER problem: The improvement of predictive accuracy when a gradual forgetting function is utilized.

(Widmer and Kubat, 1996), (Widmer, 1997), (Maloof and Michalski, 2000), (Harries, Sammut and Horn, 1998). As the IBL is the main algorithm that is used for recommending we experimented how the utilization of a gradual forgetting function (1) is able to influence the prediction accuracy of a simple nearest neighbor algorithm. In this simple toy example the feature selection does not make sense. Hence the gradual forgetting weights are used to influence the similarity measure for the IBL algorithm directly.

The instance space of a simple blocks world is defined by the three attributes ***size*** = *{small, medium, large}*, ***color*** = *{red, green, blue}*, and ***shape*** = *{square, circular, triangular}*. There is a sequence of three target concepts (1) ***size*** = *small and* ***color*** = *red*, (2) ***color*** = *green or* ***shape*** = *circular* and (3) ***size*** = *(medium or large)*. 120 training instances are generated randomly, labeled according to the hidden concept. The underlying concept is forced to change every 40 training examples. After processing each instance, the predictive accuracy is tested on an independent test set

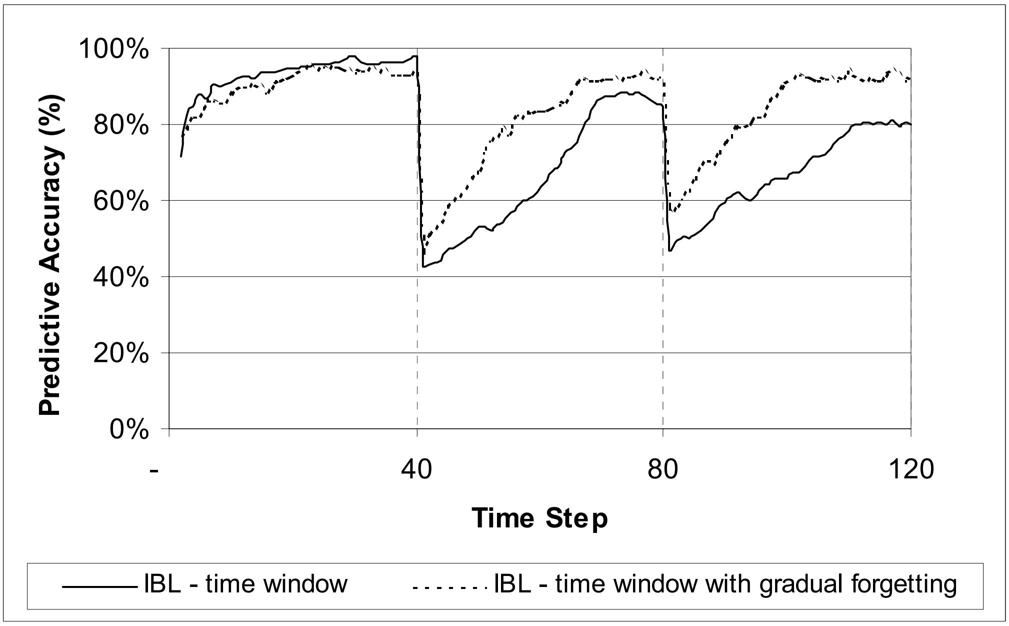

Figure 2. The STAGGER problem: The improvement in predictive accuracy when a gradual forgetting function is utilized in a time window.

of 100 instances (also generated randomly and classified according to current concept). The results are averaged over 10 runs. Two experiments are performed with this data set: The first one uses the observed instances as training set and the second one uses a fixed size time window. Figure 1 shows the results from the first experiment which uses a gradual forgetting function (*k=40%*). The prediction accuracy improves from 63% to 80%. Figure 2 shows that the presented approach can be an enhancement to the time window approach (i.e. the examples in the time window can be weighted according to their appearance over the time). The simple time window increases the average predictive accuracy from 63% to 76%. Using a gradual forgetting function additionally rises the average prediction accuracy to 85%. The dotted vertical lines indicate where the underlying concept changes. It can be seen that the dramatic concept shifts lead to a sharp decrease of the predictive accuracy.

After such falls the learning algorithms, which employ gradual forgetting function, adjust faster.

5 Conclusion

The presented approach introduces the gradual forgetting by weighting the training examples according to their appearance over time. As a result from the experiments we can conclude that the offered method can be successfully used for learning drifting user interests. In a more general consideration the suggested forgetting mechanism can be utilized for tracking drifting concepts and other relevant problems that occur over time. In a similar way the gradual forgetting can be integrated into other inductive learning methods (e.g. conducted experiments with a decision tree algorithm were quite successful). The "speed of forgetting" can be adjusted by varying k and it can be dynamically adapted using heuristics similar to those used by Widmer and Kubat for adapting the size of the time window (Widmer and Kubat, 1996). Furthermore, other types of forgetting functions can be defined (e.g. logarithmic, exponential etc).

References

1. Billsus D. and Pazzani M. J. (1999). A Hybrid User Model for News Classification. In *Kay J. (ed.), Proceedings of the Seventh International Conference on User Modeling* (UM '99), Springer-Verlag, pp. 99-108.
2. Carroll J. and Rosson M. B. (1987). The paradox of the active user. In J.M. Carroll (Ed.), *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. Cambridge, MA, MIT Press.
3. Grabtree I. Soltysiak S. (1998). Identifying and Tracking Changing Interests. International Journal of *Digital Libraries*, Springer Verlag, vol. 2, 38-53.
4. Harries M. B., Sammut C., Horn K. (1998). Extracting Hidden Context, Machine Learning 32, 101-126.
5. Kobsa A., Koenemann J. and Pohl W. (2000) *Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships*. Forthcoming.
6. Lieberman H. (1997) Autonomous Interface Agents. Proceedings of the ACM Conference on Computers and Human Interface, CHI-97, Atlanta, Georgia.
7. Maloof M. and Michalski S. (1995). Learning evolving concepts using a partial memory approach, Working Notes of the AAAI Fall Symposium on Active Learning, 70-73.
8. Maloof M. and Michalski R. (2000). Selecting examples for partial memory learning. *Machine Learning* (to appear).
9. Mitchell T., Caruana R., Freitag D., McDermott, J. and Zabowski D. (1994) Experience with a Learning Personal Assistant. Communications of the ACM 37.7 81-91.
10. Mladenic D. (1996). Personal WebWatcher: Implementation and Design. Technical Report IJS-DP-7472, Department of Intelligent Systems, J. Stefan Institute, Slovenia.
11. Pazzani M. and Billsus D. (1997). Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27, 313-331.
12. Schlimmer J., and Granger R. (1986). Incremental Learning from Noisy Data, Machine Learning1 (3), 317-357.
13. Schwab I., Pohl W. and Koychev I. (2000). *Learning to Recommend from Positive Evidence*, Proceedings of Intelligent User Interfaces 2000, ACM Press, 241 - 247.
14. Widmer G. (1997). Tracking Changes through Meta-Learning, Machine Learning 27, 256-286.
15. Widmer G. and Kubat M. (1996) Learning in the presence of concept drift and hidden contexts. Machine Learning 23: 69-101.