# Gradual Forgetting for Adaptation to Concept Drift

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

**Abstract**

The paper presents a method for gradual forgetting, which is applied for learning drifting concepts. The approach suggests the introduction of a time-based forgetting function, which makes the last observations more significant for the learning algorithms than the old ones. The importance of examples decreases with time. Namely, the forgetting function provides each training example with a weight, according its appearance over time. The used learning algorithms are modified to be able to deal with weighted examples. Experiments are conducted with the STAGGER problem using NBC and ID3 algorithms. The results provide evidences that the utilization of gradual forgetting is able to improve the predictive accuracy on drifting concepts. The method was also implemented for a recommender system, which learns about user from observations. The results from experiments with this application show that the method is able to improve the system's adaptability to drifting user's interest.

**Keywords:** Concept Drift, Forgetting, Inductive Learning

## 1 Introduction

Recently, many systems have been developed that employ machine learning methods in real life applications. Numerous of this application learns real-life concepts, which tend to shift over time. For example the systems which aim at helping users to find pieces of information or other objects that are in accordance with their personal interests (e.g. [1], [2], [8], [11] etc). These systems utilize machine learning methods to acquire user's interests profiles. Often user's interests tend to change with time. The ability to adapt fast to the current user's interests is an important feature for the recommender systems.

The concept changes (aka drifting [10] or evolving [5] concept), whether gradual or abrupt, occur over time. The evidences for changes in a concept are represented by the training examples, which are distributed over time. Hence the old observation can become irrelevant to the current period thus the learned knowledge can be out-of-date. The systems use different forgetting mechanisms to cope with this problem (e.g. [2], [3], [5], [8], [13], [14] etc.) Usually, these methods forget abruptly. That means the examples that are irrelevant according to some time criteria (e.g. examples that are outdated) are deleted from the partial memory [6]. Hence, these instances are totally forgotten. The examples that remain in the partial memory are equally important for the learning algorithm.

This paper presents a method for gradual forgetting. Namely, it offers introduction of time-based forgetting function, which provides each example with a weight according its occurring time. The importance of the examples diminishes with time. Most of the inductive learning algorithms are designed to process all training examples as equally important. Hence some modifications are necessary in those algorithms, in order to allow them to deal with examples that are unequally significant. The relevant changes in the used learning algorithms (NBC, ID3 and a multi-strategy learning method for a recommender system) are suggested in sections 4 and 5.

## 2 Related Works

STAGGER [10] 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.

A software assistant for scheduling meetings is described in [8]. It employs machine learning methods (i.e. induction on decision tree) to acquire assumptions about individual habits of arranging meetings. The learning method uses a time window (last 180 examples) to adapt faster to the shifting preferences of the user. The newly generated rules are merged with old ones. The rules that perform poorly on the test set drop down the list.

The FLORA systems [14], which are also systems for coping with concept drift, use a forgetting technique with an adaptive time window. The window size and thus the rate of forgetting is supervised and dynamically adjusted by heuristics that monitor the learning process. For example during periods when the system performs well it increases the size of the window. If there is a decreasing in performance presumably due to some changes in the target concept, the system reduces the window size.

A method that can learn and track changing context, using meta-learning is presented in [13]. The assumption is that the domain provides explicit clues as to the current context (e.g., attributes with characteristic values). A two-level learning algorithm is presented that effectively adjusts to changing contexts by trying to detect (via meta-learning) contextual clues and using this information to focus the learning process.

In [2] a system that learns user's interests profiles by monitoring web and e-mail habits is presented. 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 the time. This research shows that user's interests can be tracked over time by measuring the similarity of interests across a period of time.

An intelligent agent called NewsDude that is able to adapt to changing user's interests is presented in [1]. 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 that 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.

An offline meta-learning algorithm for identification of hidden context is presented in [3]. The approach assumes that concepts are likely to be stable for some period of time. It uses batch learning and contextual clustering to detect stable concepts and to extract hidden context.

In [5] and [6] a method for selecting training examples for a partial memory learning system is described. The forgetting mechanisms of the method selects extreme examples that lie at the boundaries of concept descriptions and remove examples from partial memory that are irrelevant or outdated for the learning task. The method uses a time-based forgetting function to remove examples from partial memory, which are older than a certain age.

# 3 Gradual forgetting

Usually, the time-based forgetting mechanisms act on example level by forgetting examples that are not up to date. Consequently they use partial memory learning in case of time-based forgetting. A comparative review of forgetting mechanisms for partial memory learning can be found in [6]. Many systems work on knowledge level also, by evaluating the accuracy of learned rules and forget or try to adapt those which perform poorly (e.g. [6], [7] etc.). The time forgetting mechanisms on example level often use a so-called time window. The concept descriptions are learned from the newest observations only (e.g. only last $l$ examples are used for training) [2], [7]. 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 algorithm [14]. The time-forgetting mechanism in [5] uses a function for aging the examples and the ones that are older than a certain age are forgotten. These approaches 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 abrupt and total forgetting of old information, which in some cases can be valuable. To avoid loss of useful knowledge, learned from old examples some systems keep

old rules till they are competitive to the new ones [7]. Another approach learns a hybrid model consisting of both a short-term and long-term model of the user's interests [1]. The short-term model is learned on the most recent observations. This hybrid user model is flexible enough to account changes in user's interest and keeps track on user's long term interests also.

The current work offers a time-based forgetting mechanism. The main idea behind it is that the natural forgetting is a gradual process. That means, the last observations should be more important than the old ones and the importance of an observation should decrease with time. Therefore a gradual forgetting function $w = f(t)$ can be defined. It should produce a weight for each observation according its location in course of time. The calculated weights must be in an interval that is suitable for the inductive learning algorithms. The experience show that the following constrains: $w_i \geq 0$ and $\frac{\sum_{i=1}^{n} w_i}{n} = 1$ are appropriate for all tested learning algorithms. For the current consideration it can be assumed that examples occur on equal time steps. Various functions that model the process of forgetting can be defined. For example, a linear gradual forgetting function is defines as follows:

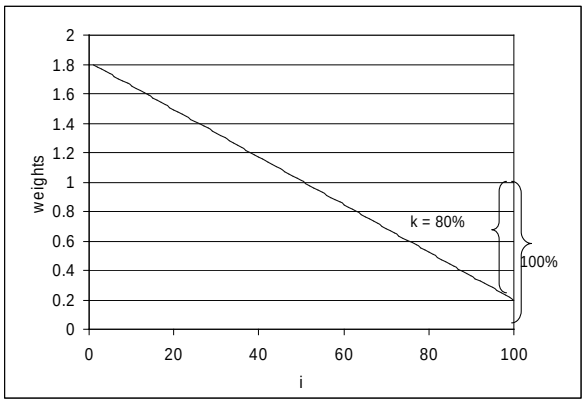


**Figure 1.** A linear gradual forgetting function. Where: n=100, k=80%.

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

where $i$ is a counter of observations starting from the most recent one and it goes back over time $i = \{1..n\}$; $n$ is the length of the observed sequence; $k \in [0,1]$ is a parameter that represents the percent of decreasing the weight of the first observation and consequently the percent of increasing the weight of the last one in comparison to the average (see Figure 1). By varying *k* the slope of the forgetting function can be adjusted. This forgetting function (1) was utilized in the experiment reported in the next sections.

The algorithms, which learn from examples, are primarily designed to treat the training examples as equally important. Therefore they should be modified to use weighted examples (in particular time-weighted examples). The suggested improvements for each of used learning methods are presented in the next sections.

# 4 Experiments with STAGGER Concepts

The initial experiments were done with an artificial learning problem that was defined and used by Schlimmer and Granger for testing STAGGER [9], one of the first concept drift tracking system. Many of the works dedicated to this problem used this data set for testing their systems (e.g. [3], [6], [12], [14], etc).

The instance space of a simple blocks world is defined by 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 and classified according to the current concept. The underlying concept is forced to change after every 40 training examples, in this way: (1)-(2)-(3).

The experiments are conducted as follows: A concept description is learned from the first $n = \{2..120\}$ examples. After each learning phase the predictive accuracy is tested on an independent test set of 100 instances, also generated randomly and classified according to the current concept. The results are averaged over 10 runs. The gradual forgetting was implemented for two learning algorithms: NBC and ID3 algorithms. For each of this learning algorithms two experiments are performed with the STAGGER data set. The first one employs full memory learning, using all currently available

instances as training set. The second one uses partial memory learning, by a fixed-size time window ($l = 30$). These experiments are reported in the following subsections.

### 4.1 Experiments with the ID3 Algorithm

Top Down Induction on Decision Tree (TDIDT) is one of the most widely applied learning algorithm. It is well known as ID3 [9]. The contingence tables are often used in the base of the algorithm [7]. An element $x^i_{j,k}$ in a

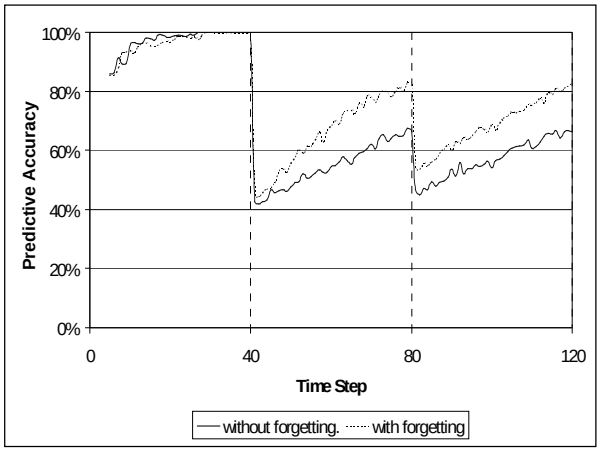


**Figure 3.** The STAGGER problem: The improvement of predictive accuracy of ID3 when the gradual forgetting is utilized.

contingence table presents the appearance of the attribute value $a_i = v^i_j$ in the subset of examples that belong to class $c_k$. The presented approach utilizes the weights of examples for calculating the confidence tables elements as follows: $x^i_{j,k} = \sum_{l=1}^{m} w_l b^{i,j}_l$, where: $b^{i,j}_l \in \{0,1\}$; $b^{i,j}_l = 1$ when for the example $e_l \in c_k$ the attribute value is $a_i = v^i_j$ and 0 otherwise; $w_l$ is the weight for the example $e_l$ calculated by a forgetting function. The different measures for goodness of split use the contingence tables in their calculations [7]. Hence, the application of forgetting weights in creating of contingence tables will influence the whole process of building the decision tree.

Figure 2 shows the results from first experiment (full memory learning). The utilization of the forgetting function (1) improves the average prediction accuracy from 69% to 77%.

Figure 3 present, the result from second experiment, which demonstrate that the presented approach can be an augmentation to the time window approach (i.e. the examples in the time window also can be weighted according to their appearance over the time). The simple time window improves the average predictive accuracy from 69% to 83%. The usage of a gradually forgetting function in it additionally improves the average predictive accuracy to 87%. 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 utilization of the gradual forgetting speedup the adaptation to concept changes.

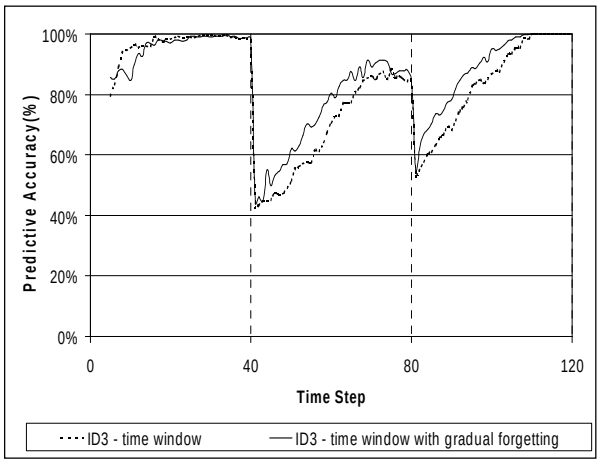


**Figure 2.** The STAGGER problem: The improvement in predictive accuracy of ID3 when the gradual forgetting is utilized in a time window.

### 4.2 Experiments with the NBC Algorithm

The Naïve Bayes Classifier (NBC) is a fast learning algorithm, which is also broadly used. The forgetting weights are utilized for NBC in calculating the probabilities. In order to calculate a probability it is necessary to count how many times an attribute value appears in a subset of examples. In current consideration the appearances of an attribute value in different examples are not equal important. Hence the counting of the appearance of a value in an

example $e_i$ should be multiplied with its weight $w_i$ (1).

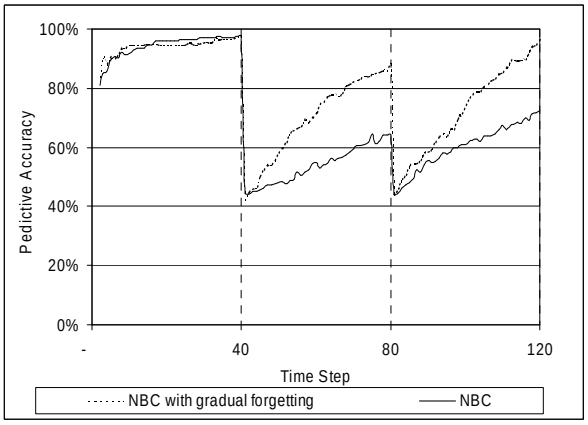


**Figure 4.** The STAGGER problem: The improvement of predictive accuracy of NBC when the gradual forgetting is utilized.

Figure 4 shows the results from the experiment using full memory. The utilization of the forgetting increases the average predictive accuracy from 69% to 79%.

Figure 5 shows the results from application of gradual forgetting in a time window. The employment of a time window improves the average predictive accuracy from 69% to 84%. The gradual forgetting in the time window additionally improves the average predictive accuracy to 88%. The concept shifts lead to a sudden decrease of the predictive accuracy of the learning algorithm. The experiments show again that the algorithms utilizing gradual forgetting are able to adapt faster when the concept changes appear.

# 5 Adaptation to Drifting Interests

A content-based recommender agent was developed [11]. In order to be unobtrusive, it learns individual interest profiles based on passive observations only. The central source of information about interests of a user is the sequence of objects he selected. A feature selection is employed to select those features that are extraordinarily important to the user for identifying relevant objects - i.e. an explicit user's profile. A probabilistic approach and an instance-based learning approach have been used for recommending. Both approaches have been modified to deal with a single class only. The feature selection additionally helps to improve the distance measure for instance-based learning. Moreover the feature selection is employed to focus the learning task. The developed algorithms have been implemented and evaluated in a real-world application - a WWW-based information system. A recent development for Internet browsing is presented in [12].

The presented forgetting mechanism is used to cope with drifting user's interests [4]. Since the feature selection plays a basic role in the developed methodology [11], the utilization of

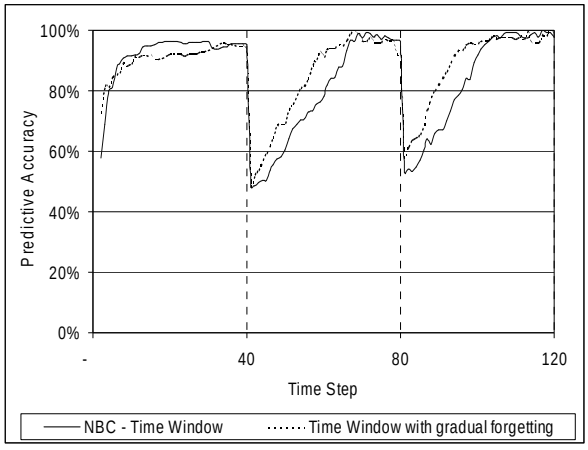


**Figure 5.** The STAGGER problem: The improvement in predictive accuracy of NBC when the gradual forgetting is utilized in a time window.

the forgetting for the feature selection should effect both, the explicit users' profiles and system's recommendations. In this case the forgetting weights are employed in counting the appearance of the features in user's selections by multiplying each its occurrence with the weight $w_i$ of the observation. Consequently, this reflects the estimation of features' significance. In this way the appearance of a feature in the resent observations become more valuable.

The experiment shows that the employment of gradual forgetting is able to influence both the generated explicit user profiles and the recommendations. The effect on the user profiles is that they include mainly the features that represent recent observations and those, which characterize interests that are stable over time. The average predictive accuracy of the

recommendations improves about 2%. This improvement may not look very large, but we should take into account that average predictive accuracy approaches 90% and in about 40% of the cases the system accuracy approaches 100%. In such cases the interests are stable and any improvement is actually impossible. When the user's interests are changed the predicting accuracy can drop down dramatically. After such changes, the utilization of presented forgetting mechanism results in faster adaptation to the new user's interests.

## 6 Conclusion

The presented approach introduces the gradual forgetting by weighting the training examples according to their appearance over time. The conducted experiments show that the presented method is applicable to different learning algorithms and is able to improve its adaptability to drifting concepts. Conducted experiments with IBL algorithm on STAGGER data set, which produces similar results like the presented in section 4, are reported in [4]. The experiments show also that the presented forgetting method can work in cooperation with other forgetting mechanisms for partial memory learning (e.g. time window). The "speed of forgetting" can be adjusted by varying *k* and it can be dynamically adapted using heuristics like those used by Widmer and Kubat for adapting the size of the time window [14]. 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. Grabtree I. Soltysiak S. (1998). Identifying and Tracking Changing Interests. International Journal of Digital Libraries, Springer Verlag, vol. 2, 38-53.
3. Harries M. B., Sammut C., Horn K. (1998). Extracting Hidden Context, Machine Learning 32, 101-126, Kluwer Academic Publishers.
4. Koychev, I. and Schwab I. (2000). Adaptation to Drifting User's Intersects - Proceedings ECML2000/MLnet workshop "ML in the New Information Age".
5. 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.
6. Maloof M. and Michalski R. (2000). Selecting examples for partial memory learning. *Machine Learning* (to appear).
7. Mingers J. (1989). An Empirical Comparison of Selected Measures for Decision Tree Induction, Machine Learning 3, 319-142, Kluwer Academic Publishers.
8. 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.
9. Qinlan J.R. (1986). Induction of Decision Trees, Machine Learning 1, 81-106 Kluwer Academic Publishers.
10. Schlimmer J., and Granger R. (1986). Incremental Learning from Noisy Data, Machine Learning1 (3), 317-357, Kluwer Academic Publishers
11. Schwab I., Pohl W. and Koychev, I. (2000). Learning to Recommend from Positive Evidence, Proceedings of Intelligent User Interfaces 2000, ACM Press.
12. Schwab I., Kobsa A. and Koychev I. (2000) Learning about Users from Observation, AAAI 2000 Spring Symposium: Adaptive User Interface.
13. Widmer G. (1997). Tracking Changes through Meta-Learning, Machine Learning 27, 256-286, Kluwer Academic Publishers
14. Widmer G. and Kubat M. (1996) Learning in the presence of concept drift and hidden contexts. Machine Learning 23: 69--101, Kluwer Academic Publishers