

# The Effect of History on Modeling Systems' Performance: The Problem of the Demanding Lord

George Giannakopoulos\*, Themis Palpanas\*

*\*Department of Information Engineering and Computer Science  
University of Trento, Italy*

*Email: ggianna@disi.unitn.it, themis@disi.unitn.eu*

**Abstract**—In several concept attainment systems, ranging from recommendation systems to information filtering, a sliding window of learning instances has been used in the learning process to allow the learner to follow concepts that change over time. However, no analytic study has been performed on the relation between the size of the sliding window and the performance of a learning system. In this work, we present such an analytic model that describes the effect of the sliding window size on the prediction performance of a learning system based on iterative feedback. Using a signal-to-noise approach to model the learning ability of the underlying machine learning algorithms, we can provide good estimates of the average performance of a modeling system independently of the supervised machine learning algorithm employed. We experimentally validate the effectiveness of the proposed methodology with detailed experiments using synthetic and real datasets, and a variety of learning algorithms, including Support Vector Machines, Naive Bayes, Nearest Neighbor and Decision Trees. The results validate the analysis and indicate very good estimation performance in different settings.

**Keywords**-concept drift; user modeling; adaptive learning; “demanding lord” problem

## I. INTRODUCTION

In the literature, researchers have expressed the inability of several classifiers to follow changing concepts (e.g., in user preferences [1]). The implied change of context that changes a target concept, its effects on classification, and the evolution of learning methods in order to tackle the change that have been studied in the machine learning and the stream mining community as the problem of “concept drift”. Drifting concepts appear in a variety of settings in the real world, e.g., the state of a free market or the traits of the most viewed movie.

The questions we answer with this study are the following. How can we model the expected performance of learning algorithms based on knowledge of the characteristics of the abrupt concept drift (also termed “concept shift”), such as the period of occurrence of these drifts? How can we estimate the performance of a learner for different window sizes and concept change periods, regardless of the underlying learning algorithm?

To answer these questions, we focus on the functional relation between the window size and the average performance of a learning system. In summary, we make the

following contributions. We offer a formulation — under the label “the problem of the demanding lord” — and analytic solution of the problem of estimating the average performance in learning systems in the presence of abrupt concept drift (concept shift). We describe a methodology to approximately estimate the performance of a learning algorithm as a function of signal-to-noise in the training set, regardless of the learning algorithm idiosyncrasies. This allows practitioners to easily optimize the performance of learning systems. To the best of our knowledge, this is the first systematic approach for estimating the average performance of learning algorithms in this setting. This approach provides the basis for further analytic study of the connection between average performance of an incremental learning system and the noise in the training set.

In the following sections, we present the related work (Section II) and we formulate the problem faced (Section III). We then elaborate on the proposed analytic methodology (Section IV). Then, we experimentally validate the analysis (Section V) and we conclude with a discussion on our findings (Section VI).

## II. RELATED WORK

There have been several studies with different assumptions on the speed or type of drift. The drift can be gradual (termed “drift”), instantaneous (termed “abrupt drift” or “shift”), or a function of time [2], where a parameter indicates the speed of the drift.

In an early influential work, the problem of “concept attainment” in the presence of noise was indicated and studied in the STAGGER system [3]. The system approximated a (boolean expression) concept based on examples, through weighted symbolic characterizations. A backtracking methodology allowed changing the current description of the target concept to account for the drift. In [4], a full-memory (i.e., all remembering) incremental-learning speed-efficient system is presented, aiming to find concept descriptions that are both characteristic (wide coverage) and discriminative (high precision).

A focused study of the mistake rate of a learning algorithm that updates its estimate based on the most recent examples [5] identifies bounds for this rate, based on the

number of recent examples. In [6] the authors connect the VC-dimension ( $d$ ) of a target concept to the difficulty of attaining the target concept.

In later works, we find approaches where either hard-coded thresholds are used [7] based on trial-and-error, or the window is adjusted whenever a shift is detected [8], possibly based on some optimization scheme [9], [10]. The window size can also be adjusted based on heuristics [2], or the observations in the window can be assigned different weights over time [11]. Recently, researchers have also used “local windows” in sub-parts of models, as in [12] where an incremental decision tree uses local sub-concept adaptive window sizes. Another approach uses multiple competing windows of different sizes [13], that try to tackle the problem of differentiating noise from virtual drift from actual concept drift. In [14], two classifiers are used, one with full memory and one with partial, fixed-size memory, in a *paired learning* approach, where the partial memory *reactive* learner causes a reset to the full-memory classifier in key time-points.

Other approaches use a window differently or not at all. In user modeling, instead of a window, constantly updated weights on terms are used to follow change in user interests in [15]. In [16], the proposed system learns based on extreme examples and batch learning. Ensemble-based approaches exist, such as the case of boosting [17], or the Concept Drift Committee of decision trees [18], or the case where an EM algorithm assigns weights to ensemble classifiers, which are created and disposed of over time as needed [19].

In this work, motivated by our related studies on window size and its effect on user modeling [20], [21], we provide an analytic framework that allows the a-priori estimation of an optimal window size for the case of periodic concept shifts, overcoming the heuristic or algorithm-specific approaches of the literature. Another major contribution is based on the proposal of an algorithm-agnostic signal-to-noise function (see Section IV-A) as a description of the connection between noisy input and the performance of a learning algorithm.

### III. PROBLEM FORMULATION

We call the problem we analyze the “problem of the demanding lord” (see Table I for an overview of the analogy). The idea is that there is a demanding lord that requires a meal every day from his good servant. The servant tries to estimate a classification of the meals his lord likes, based on his reactions to previous meals. Each day the servant offers a set of meals and gets the full set of reactions from the lord as feedback. The lord, however, may change his preferences. We want to determine how many of the lord’s latest answers the servant needs to remember in order to maximally satisfy the lord on average over time.

We can differentiate servants from their policy of learning  $\mathbb{P}$  and by the number of reactions  $r$  they take into account. *The finite-memory servant* remembers the last  $r$  reactions

Description	Symbol	Demanding lord analogy and other notes
User	$\mathbb{W}$	demanding lord
User modeling system	$\mathbb{H}$	servant
System iteration	$d$	days of service, $d \in \mathbb{N}^*$
Concept instance	$G$	meal
Feedback instance	$A$	lord’s reaction to meal
Learning method	$\mathbb{P}$	learning policy (i.e., the way the servant learns)
Memory window size	$r$	# of the latest reactions the servant remembers
Period of shift	$T_s$	days between two consecutive interest shifts

Table I  
PROBLEM FORMULATION ANALOGY AND MAIN SYMBOLS

only. *The all-remembering servant* ( $r \rightarrow \infty$ ) remembers all his lord’s reactions. Therefore, a servant can be described as the pair  $\mathbb{H} \equiv \langle \mathbb{P}, r \rangle$ . The lord can be described based on the probability distribution  $p(d)$  of an occurring shift, over the days elapsed from the last shift:  $\mathbb{W} \equiv \langle p(d) \rangle$ .

We make some assumptions that facilitate the representation of the problem. First, the lord  $\mathbb{W}$  periodically changes his interests through what we call an *interest shift*, or simply *shift*. This implies that:  $p(d) = 1$ , if  $d = kT_s, k \in \mathbb{N}^*$  else  $p(d) = 0$ . Also, a shift is radical, so that no information is valid concerning reactions on the previous sets of meals. This makes sure that we can judge noise when detecting a shift. For a given day  $d$  and a set of offered meals  $\mathbb{G}_d = \{G_1, G_2, \dots, G_n\}, n > 0$  the set of the lord’s reactions on that day is  $\mathbb{A}_d = \{A_1, A_2, \dots, A_n\}$  containing the reactions mapped to each one of the  $n$  meals.

In the following elaboration we refer to Figure 2 to visualize the described states. In Figure 1 we provide the legend of the corresponding symbols. Each given day  $d_c$ , the servant  $\mathbb{H}$  uses the  $r$  last feedback sets (see Figure 2)  $\mathbb{A}_{d_c-r}, \mathbb{A}_{d_c-r+1}, \dots, \mathbb{A}_{d_c-1}$  to learn, using his training policy  $\mathbb{P}$ , to estimate meals. We call this set of feedback sets the training set  $\mathbb{T}$  of the servant. In a given point in time  $d_c$  the servant is trained using only valid information (the white circles in Figure 2), if within the last  $r$  days, no shift has occurred. Otherwise, if a shift occurred on day  $d_s$ , before the current day  $d_c$ ,  $d_c - d_s \leq r$ , then the servant has some no-longer-valid feedback set  $\mathbb{N} \subset \mathbb{T}$  (noise, shown as black circles in Figure 2) and some valid  $\mathbb{S} \subset \mathbb{T}$  (signal), and  $\mathbb{T} = \mathbb{S} \cup \mathbb{N}$ . The period of the shift will be noted as  $T_s$ , i.e., every shift happens exactly  $T_s$  days after the previous one<sup>1</sup>. The first interest shift happens on day  $d = T_s$ . We start with this assumption of periodicity, to facilitate the formulation of the problem. Later, we verify whether the results of our analysis are also valid for random shift frequency.

If  $|\cdot|$  is the operator of the size of a set, then we let  $S = |\mathbb{S}|$  and  $N = |\mathbb{N}|$  represent the signal magnitude and the noise magnitude of a training set  $\mathbb{T}$ . We also allow  $\mathbb{S} = \emptyset \Rightarrow S = 0, \mathbb{N} = \mathbb{T} \Rightarrow N = r$ , when all the training set is not valid any longer because a shift has just occurred. Correspondingly,  $\mathbb{N} = \emptyset \Rightarrow N = 0, \mathbb{S} = \mathbb{T} \Rightarrow S = r$ , when no change has

<sup>1</sup>In the case of random shifts,  $T_s$  can be approximated by the expected number of days between two consecutive interest shifts.

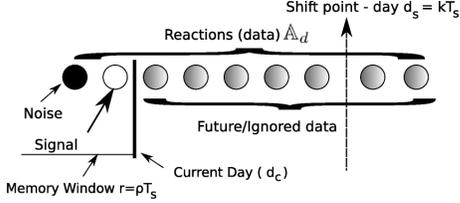


Figure 1. Legend, explaining the symbols used in the following illustrations.

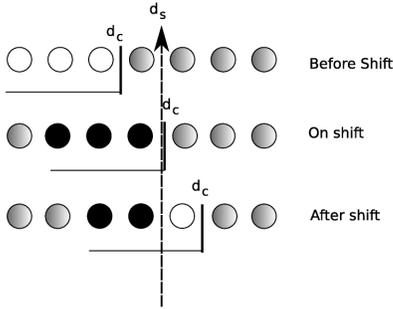


Figure 2. The validity of training data over time. When the current day is after an interest shift, all data before the interest shift become invalid (i.e., noise).

occurred within the last  $r$  days (for more intuition on why this is the case see Figure 2).

Given the above definitions, we define the signal to noise ratio  $Z$  of a given moment in time, as  $Z = \log' S - \log' N$ , where  $\log' x = \log(1 + x)$  returns for a given  $x$  the natural logarithm of  $x + 1$ , to return a value also for  $x = 0$ .

Let us consider that the servant training set size  $r$  is a ratio  $\rho$  of the shift period  $T_s$ :  $r = \rho T_s$ . We call this ratio  $\rho$ , i.e., the memory window-to-shift period ratio, *characteristic ratio* of a given servant  $\mathbb{H}$ . For a given servant,  $\mathbb{H}$  and a given lord  $\mathbb{W}$  we support that the servant's average performance on predicting the preferences of the lord is a function  $f(\mathbb{W}, \mathbb{H}, \rho)$  or, equivalently,  $f(p(d), \mathbb{P}, \rho)$ . This means that we consider the performance to be a function of the shift probability distribution, the learning algorithm and the characteristic ratio.

#### IV. ANALYSIS OF WINDOW SIZE EFFECT

We perform an analysis of the effect of the memory (window) size of a learner on its performance, in the presence of concept drift. The analysis is based on the estimation of a function connecting signal-to-noise in the training set to the expected performance of a given learner (Section IV-A). Then, we illustrate how the mathematic relation of the window size of a learner and the signal-to-noise ratio for every iteration of a modeling system with periodic concept shifts allows the estimation of the average performance of the learning system over time. The described analysis can be used, in conjunction with concept shift detection methods or

various related shift indicators to optimize the  $\rho$  parameter, for a given user and learning algorithm.

##### A. Characteristic Transfer Function

To describe the  $\mathbb{P}$  component of the predictive function  $f(p(d), \mathbb{P}, \rho)$ , we consider that each learning algorithm is described by a function which indicates the impact of signal-to-noise ratio  $Z$  in the training set to the average performance  $\bar{f}$  of the algorithm. Given an algorithm with a minimum performance of  $\underline{m}$  and a maximum of  $\overline{M}$  for a given domain, the form of the function is:

$$\bar{f}(Z) = \underline{m} + (\overline{M} - \underline{m}) \frac{1}{1 + b \times \exp(-c \times Z)} \quad (1)$$

where  $\exp(x) = e^x$  and the constants  $b \in \mathbb{R}, c \in \mathbb{R}$  are parameters of the *sigmoid* function. We call the  $\bar{f}$  function the *characteristic transfer function (CTF)* of the learning algorithm. In the case where  $\underline{m} = 0, \overline{M} = 1$ , Equation 1 takes the form

$$\bar{f}_N(Z) = \frac{1}{1 + b \times \exp(-c \times Z)} \quad (2)$$

which we call the *normal characteristic transfer function (NCTF)* and it represents a normalized version of the CTF.

The intuition behind the sigmoid form of the CTF is discussed further in [22]. We consider the CTF to be characteristic of an algorithm for a given dataset. We expect that the sigmoid can be estimated from training sets of varying noise levels ( $Z$ ) and, then, it can be used as a known function for the given algorithm (see Section V). We emphasize that we do not make any specific assumption for the underlying distribution of training instances.

##### B. Average Performance and Characteristic Ratio

Given the estimation of the CTF, which describes the  $\mathbb{P}$  component of the servant, we need to find the relation between  $\rho$  and  $\bar{f}(Z)$ . We examine the case of the short-memory servant, where  $\rho \leq 1$ , i.e., the lord is not expected to have an interest shift too often.

For the case where  $\rho \leq 1$ , we can calculate the signal to noise ratio, studying different key iteration intervals as follows. In the beginning  $d = 0, S = 0, N = 0$ . In the interval  $0 < d < T_s, S = \min(d, r), N = 0$ . This happens because the maximum number of training instances are  $r$  i.e.,  $S + N = r \Rightarrow N = r - S$ . After the first shift, everything we know so far is suddenly useless, i.e., noise. Thus, if  $[a]_b$  is the integer part of the division  $\frac{a}{b}$  (the modulo operator), it stands that:  $d \in \{d' \mid [d']_{T_s} = 0\} \Rightarrow S = 0, N = r$ . The sum of training instances are  $r$  at most; also, everything that is not signal, is necessarily noise. Thus, while  $d \in \{d' \mid d' > T_s, 0 < [d']_{T_s} < T_s\}$  it stands that  $S = \min([d]_{T_s}, r), N = r - S$ . The function  $\min()$  is the minimum function. In Figure 3 we illustrate an example case ( $\rho = 1$ , and  $T_s = 3$ ).

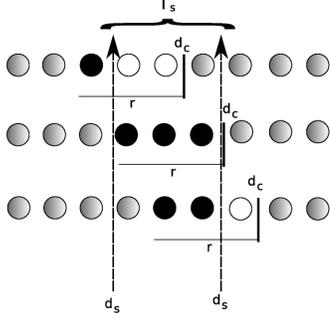


Figure 3. Consecutive days of a short-memory servant.  $\rho = 1, T_s = 3$ .

From the above we deduce that  $S, N$  are actually  $S(d), N(d)$  functions of the current day. Every day  $d$ ,  $Z$  is:  $Z(d) = \log'(S(d)) - \log'(r - S(d)) = \log'(S(d)) - \log'(\rho T_s - S(d))$ . The expected value  $\bar{Z}$  of  $Z$  is:

$$\bar{Z} = E(Z) = \sum_i Z_i q(Z_i) \quad (3)$$

where  $q(Z_i)$  is the probability of occurrence of  $Z_i$  and  $Z_i, 0 \leq i < r = \rho T_s$  indicates possible values of  $Z$ .

A day  $d$  can be modeled related to the last shift that occurred<sup>2</sup>. So, if  $d$  is  $k_1$  days after the last shift, which occurred on day  $k_0 T_s$ , then we can express  $d$  as  $d = k_0 T_s + k_1$ . Then, for a chosen, arbitrarily big  $k_0 \gg 0$ ,  $k_0 \simeq k_0 + 1$ , it stands that  $d = k_0 T_s + k_1, k_0 \gg 0, 0 \leq k_1 < \rho T_s$ , i.e., after many shifts, all the values  $Z_i, i \neq \rho T_s - 1$  have a probability of approximately

$$q(Z_i) = \frac{k_0}{k_0 T_s} \Rightarrow q(Z_i) \simeq \frac{1}{T_s} \quad (4)$$

since they appear  $k_0$ , or  $k_0 + 1$  times (depending on  $k_1$ ) in  $k_0 T_s$  days. For the value  $Z_{\rho T_s - 1} \equiv Z_{max} = \log'(\rho T_s) - \log'0$  (which is a *constant*) — i.e., the maximum value of  $Z$  — the probability of occurrence of  $Z_{max}$  is

$$q(Z_{max}) = 1 - \sum_{i=0}^{\rho T_s - 2} q(Z_i) \Rightarrow q(Z_{max}) = 1 - \rho + \frac{1}{T_s} \quad (5)$$

Therefore, Equation 3, setting  $R_{tot} = \sum_{i=0}^{\rho T_s - 1} Z_i$  gives

$$\bar{Z} = \frac{R_{tot}}{T_s} + (1 - \rho) Z_{max} \quad (6)$$

Using the same methodology, we calculate the performance of the system, using Equation 1. Given the fact that performance  $f$  is a strictly monotonic function of  $Z$ , the unique values of  $f$  are exactly the same in number, as the  $Z$  possible values. In addition, each value  $f(Z_i)$  holds the same probability of appearance as  $Z_i$ . This means that, similarly

<sup>2</sup>We note that the possible values of  $Z$ , that appear before the first shift, can be ignored (see [22] for details).

to Equation 3, the expected value of the performance of the system is  $\bar{f} = E(f(Z)) = \sum_{i=0}^{\rho T_s - 1} f_i q'(f_i)$ , where  $f_i$  is the  $i$ -th possible distinct value of  $f(Z)$ , with  $q'(f_i) \equiv q(Z_i)$ . Given Equation 1:  $f_i = \underline{m} + (\bar{M} - \underline{m}) \frac{1}{1 + b \times \exp(-c \times Z_i)}$ . Thus, for  $\bar{f}$  we get the following equation<sup>3</sup>.

$$\bar{f} = \underline{m} + (\bar{M} - \underline{m}) \left( \frac{1}{T_s} \sum_{i=0}^{\rho T_s - 2} \bar{f}_N(Z_i) + (1 - \rho + \frac{1}{T_s}) \bar{f}_N(Z_{max}) \right) \quad (7)$$

We have argued that the optimization of a system, consists of the following process steps: estimation of the CTF, detection of the interest shift period and optimization of the  $\rho$  parameter. If the proposed process stands, then all problems that can be expressed through the problem of the demanding lord can have an a-priori estimation of performance for any given algorithm, with the effort of estimating the characteristic transfer function of the learning algorithm and the calculation of the expected shift period of the user, which of course are non-trivial tasks.

## V. EXPERIMENTAL EVALUATION

In order to validate the effectiveness of the proposed approach, we perform a set of experiments with two aims. First, we want to check *whether the estimation of the characteristic transfer function (CTF) of an algorithm can be achieved*. Second, we want to determine *whether the analytic estimator of signal-to-noise converges to the actual average signal-to-noise of the system*. If it converges, the estimator can optimize  $\rho$  online, after detected shifts.

Before the evaluation, we give an example of a Signal-to-Noise to Mean Performance graph — drawn from a single test case — in Figure 4. The left graph indicates the average performance per signal to noise, including also confidence interval bars within 95% confidence level. In our experiments, we search for a good-enough CTF, using only  $Z$  values that have enough support<sup>4</sup> as shown in the right graph of Figure 4.

The datasets we report on in this study are the following. The first dataset is based on the STAGGER method evaluation dataset [3], which shifts the target concept every 40 iterations of the system. The second, real world dataset we use is based on the Ifo Business Survey<sup>5</sup>, which describes some aspects of the business climate in Germany. Each record (i.e., instance) contains six questionnaire-derived measurements representing the *balance* values and the *index* values of the business situation, the business outlook and the business climate. Based on two instance features, the study classifies the state of the market as “boom”, “downswing”, “recession”,

<sup>3</sup>For details see [22].

<sup>4</sup>We use  $Z$  values that have appeared enough times to have a standard error in their mean performance estimation below 0.05.

<sup>5</sup>Ifo Business Survey - The Ifo Business Climate for Germany since January 1991, Ifo Institute: [http://www.cesifo-group.de/portal/page/portal/ifoHome/a-winfo/d6zeitreihen/15reihen/\\_reihenkt](http://www.cesifo-group.de/portal/page/portal/ifoHome/a-winfo/d6zeitreihen/15reihen/_reihenkt)

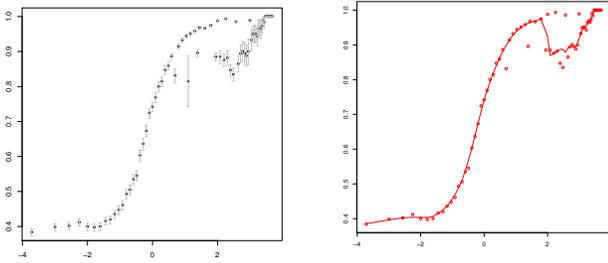


Figure 4. Mean Performance (vertical axis) per signal-to-noise ratio (horizontal axis): Means plot with confidence bars (left) and LOWESS regression plot for well-supported points (right).

“upswing”, which label the current target concept based on the current iteration. For more information on the datasets, see [22].

*Searching for a Good Sigmoid:* Given an equation of sigmoid type (see Equation 1), we need to identify the parameters that best describe a set of observed data points. In our case, the data points are measured performance values for given signal-to-noise ratios. The search in the parameter space is performed by a genetic algorithm (GA), searching for an approximate good set of parameters<sup>6</sup>. The fitness function of the GA used is based on the Kolmogorov-Smirnov (KS) goodness-of-fit  $D$  statistic. The K-S test statistic  $D$  is expected to have a low value if two sets of samples from distributions are more likely to originate from the same underlying distribution. In our case, the two compared distributions are the actual and estimated values of performance, corresponding to the possible  $Z$  values.

To determine whether the sigmoid estimation is good, we perform a five-fold cross-validation of our sigmoid estimation process. The training set is used to determine the CTF and the test set to determine the collinearity (through a Pearson test) between the estimation and the real values of the performance with the given CTF. A high collinearity value indicates that the CTF is a good estimate.

*Discussion on the CTF:* Observing the data in these runs, we identified an important aspect of the learners, illustrated in Figure 5: sometimes the sigmoid is shifted to the left or to the right. Given this trait, we better expressed and approximated the CTF by the form:

$$\bar{f}(Z) = \underline{m} + (\bar{M} - \underline{m}) \frac{1}{1 + b \times \exp(-c \times (Z - d))} \quad (8)$$

which adds a parameter  $d$  that models the horizontal position shift, improving CTF estimation.

In Table II we illustrate the Pearson correlation values indicating how collinear the performance values from the estimated sigmoid CTFs are to the actual values. In order

<sup>6</sup>Non-linear regression, which we tried first, failed to converge in some cases for the given setting and had to be abandoned.

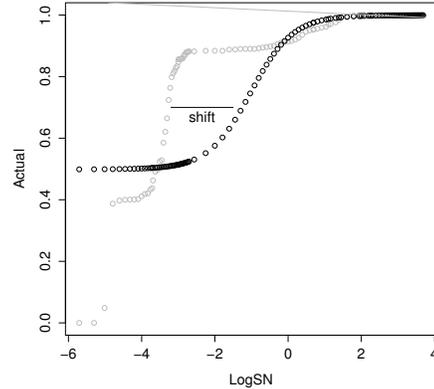


Figure 5. Shift in the Sigmoid. Gray: true points, Black: estimation.

Setting		Correlation Quantiles					
STAGGER Dataset		Naive Bayes			J48 Decision Tree		
$T_s$	$\rho$	1st Q.	Mean	3rd Q.	1st Q.	Mean	3rd Q.
40	0.50	0.94	<b>0.94</b>	0.95	0.86	<b>0.92</b>	0.96
40	0.85	0.88	<b>0.90</b>	0.89	0.87	<b>0.89</b>	0.91
40	1.00	0.70	<b>0.76</b>	0.79	0.68	<b>0.77</b>	0.81
IFO Dataset		SVM			NN		
$\bar{T}_s$	$\rho(r)$	1st Q.	Mean	3rd Q.	1st Q.	Mean	3rd Q.
8.46	0.59 (5)	0.72	<b>0.76</b>	0.90	0.84	<b>0.86</b>	0.94
8.64	1.04 (9)	0.58 (0.08)	<b>0.77</b>	0.93	0.81	<b>0.84</b>	0.94

Table II  
CORRELATION OF TRUE PERFORMANCE TO CTF-BASED ESTIMATE.  
 $P$ -Value of tests  $< 0.1$ .

to elaborate on the whole set of results over all folds, we provide the 1st and 3rd quantiles of the collinearity values, as well as the mean value. The results on both the STAGGER and the market dataset indicate the consistently high correlation, and thus success, of the estimation to the actual data points<sup>7</sup>. We should note, however, that in the worst case of the market dataset — where neither of our prerequisites the correlation is not statistically significant ( $p$ -value  $> 0.05$ ), even though it is rather strong. This indicates that there is still room for improvement in CTF estimation. Lastly, we note that the estimation time for a CTF is on the order of a few seconds per fold ( $< 1000$  iterations).

*Estimation of Performance over Time:* In this section we study whether the estimated and real average performance of a *demanding lord system* (DLS) converge over time. Using the best performing estimated CTF (in terms of collinearity to the actual performance), we follow a learning system over time recalculating on every iteration the average period of the shift, the estimated performance and the actual performance, based on the feedback.

Given the above information we plot graphs of iteration (x-axis) vs. delta (y-axis) between the estimation and the actual value (absolute error). We measure the Spearman and

<sup>7</sup>For the specifics on the evaluation process for the market dataset please consult [22].

Synthetic dataset: Boolean Concept Dataset - 6000 Iterations			
Setting		Spearman	Pearson
Bayes	40-20	-0.3022561	-0.272744
Bayes	40-40	-0.9611965	-0.482552
Real dataset: Business Climate, 212 Iterations			
Setting		Spearman	Pearson
SVM	Random-5	-0.1777558	-0.3815536
SVM	Random-9	-0.3168430	-0.4193718

Table III  
CORRELATION OF ITERATION NUMBER AND DELTA (10-FOLD  
VALIDATION).  $P$ -value of tests  $< 0.05$ .

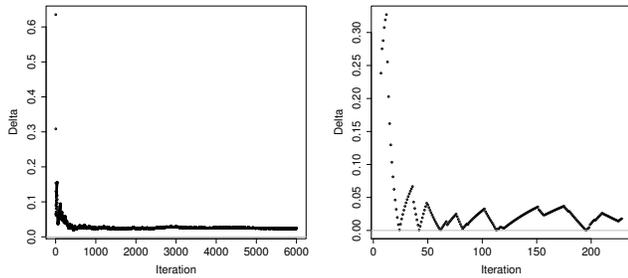


Figure 6. Convergence (10-fold validation) of the average delta.  $\rho < 1$ . Left to right: Boolean Concept and Real Dataset.

the Pearson correlations, indicating rank and linear correlation correspondingly between iteration and delta. In the case of the real dataset, where the shift period changes over time, the system takes into account on every iteration the *average* period of the shifts. We perform 10-fold validation, also increasing the number of iterations during which we examine the system, for the synthetic dataset. We can see the Boolean Concept dataset graph (exhibiting periodic shifts), Figure 6-left and the Business Climate dataset graph (aperiodic shifts) in Figure 6-right. We observe (refer to Table III) that the performance estimation converges (negative values for correlation) well within the statistical significance level of 99%. It is very interesting that the convergence also happens in the random shift dataset (Table III, Business Climate dataset), which indicates that the estimator is robust.

It appears that the estimation error falls to levels below 5% rather quickly (few hundreds of iterations), which indicates that the average performance can be estimated quite early, allowing for early optimization of the memory window.

## VI. CONCLUSIONS

In this study, we have shown that on a partial memory online learning system, we can estimate a good memory window to maximize performance for a given series of instances, regardless of the underlying learner. The analysis we have performed offers a basis for studying learning algorithms from a signal-to-noise-response perspective. For a given dataset, calculating a good CTF allows practitioners to optimize a system's learning performance, without exhaustive experiments.

## REFERENCES

- [1] G. I. Webb, M. J. Pazzani, and D. Billsus, "Machine learning for user modeling," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1, pp. 19–29, Mar. 2001.
- [2] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69–101, 1996.
- [3] J. Schlimmer and R. Granger, "Incremental learning from noisy data," *Machine learning*, vol. 1, no. 3, pp. 317–354, 1986.
- [4] R. Reinke and R. Michalski, "Incremental learning of concept descriptions: A method and experimental results," *Machine intelligence*, vol. 11, pp. 263–288, 1988.
- [5] A. Kuh, T. Petsche, and R. Rivest, "Learning time-varying concepts," in *Proceedings of the 1990 conference on Advances in neural information processing systems 3*. Morgan Kaufmann Publishers Inc., 1990, p. 189.
- [6] D. P. Helmbold and P. M. Long, "Tracking drifting concepts by minimizing disagreements," *Machine Learning*, vol. 14, no. 1, pp. 27–45, 1994. [Online]. Available: <http://dx.doi.org/10.1007/BF00993161>
- [7] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski, "Experience with a learning personal assistant," *Communications of the ACM*, vol. 37, no. 7, pp. 80–91, 1994.
- [8] B. I. Crabtree and S. Soltysiak, "Identifying and tracking changing interests," *International Journal on Digital Libraries*, vol. 2, no. 1, pp. 38–53, 1998.
- [9] I. Koychev and R. Lothian, "Tracking drifting concepts by time window optimisation," in *Proceedings of AI-2005, the Twenty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Citeseer, 2005, pp. 46–59.
- [10] L. Kuncheva and I. Žliobaitė, "On the window size for classification in changing environments," *Intelligent Data Analysis*, vol. 13, no. 6, pp. 861–872, 2009.
- [11] I. Koychev and I. Schwab, "Adaptation to drifting user's interests," in *Proc. of ECML2000 Workshop: Machine Learning in New Information Age*. Citeseer, 2000, pp. 39–45.
- [12] M. Núñez, R. Fidalgo, and R. Morales, "Learning in environments with unknown dynamics: Towards more robust concept learners," *The Journal of Machine Learning Research*, vol. 8, pp. 2595–2628, 2007.
- [13] M. Lazarescu, S. Venkatesh, and H. Bui, "Using multiple windows to track concept drift," *Intelligent Data Analysis*, vol. 8, no. 1, pp. 29–59, 2004.
- [14] S. Bach and M. Maloof, "Paired learners for concept drift," in *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on, 2008*, pp. 23–32. [Online]. Available: 10.1109/ICDM.2008.119
- [15] D. Widyantoro, T. Ioerger, and J. Yen, "An adaptive algorithm for learning changes in user interests," in *Proceedings of the eighth international conference on Information and knowledge management*. ACM New York, NY, USA, 1999, pp. 405–412.
- [16] M. Maloof and R. Michalski, "Selecting examples for partial memory learning," *Machine Learning*, vol. 41, no. 1, pp. 27–52, 1999.
- [17] M. Scholz and R. Klinkenberg, "Boosting classifiers for drifting concepts," *Intelligent Data Analysis*, vol. 11, no. 1, pp. 3–28, 2007.
- [18] K. Stanley, "Learning concept drift with a committee of decision trees," *Computer Science Department, Univ. of Texas-Austin*, 2001.
- [19] F. Chu, Y. Wang, and C. Zaniolo, "An adaptive learning approach for noisy data streams," in *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on, 2004*, pp. 351–354.
- [20] G. Giannakopoulos and T. Palpanas, "Content and type as orthogonal modeling features: a study on user interest awareness in entity subscription services," *International Journal of Advances on Networks and Services*, vol. 3, no. 2, 2010. [Online]. Available: <http://www.iit.demokritos.gr/~ggianna/Publications/adaptiveSubscription.pdf>
- [21] G. Giannakopoulos and T. Palpanas, "Adaptivity in entity subscription services," in *Proceedings of ADAPTIVE2009*, Athens, Greece, 2009.
- [22] G. Giannakopoulos and T. Palpanas, "DISI-10-052: The effect of history on modeling systems' performance: The problem of the demanding lord," DISI, University of Trento, Tech. Rep., 2010. [Online]. Available: <http://disi.unitn.eu/~themis/publications/demandinglord-tr10.pdf>