Recognizing Bounds of Context Change in On-Line Learning

Helen Kaikova*, Vagan Terziyan**, Borys Omelayenko*

*Department of Artificial Intelligence,
Kharkov State Technical University of Radioelectronics,
Lenina av. 14, Kharkov, 310166, Ukraine,
e-mail: helen@kture.kharkov.ua

**Department of Computer Science and Information Systems,
University of Jyvaskyla, P.O.Box 35, FIN-40351, Jyvaskyla, FINLAND,
e-mail: sepi@jytko.jyu.fi

Abstract

The on-line algorithms in machine learning are intended to discover unknown function of the domain based on incremental observing of it instance by instance. These algorithms have a great ability for adaptation to each new situation they appear in time, or to a new context. In this paper we propose an algorithm for identifying the moments when the context of the 'environment' changes. The key idea is that if the predictor, which was based on some previous examples, becomes crucially inconsistent with new examples then we identify context change at the placement of this instance. If most of the predictors from diverse ensemble start to identify context change then we report total context change. We experimentally illustrate that this idea works very well on Vowel recognition dataset.

1 Introduction

A target function to be learned by a learner passes in time through many different contexts that influence its behavior. To match these changes the on-line algorithms track the target function, and have special schemes for fast adaptation, such as multiplicative weight updating in Winnow (Littlestone, 94) variants. We assume that target function implicitly contains another function in it: *context function*, which indicates recent active context, or, at least, places where context changes. We can improve knowledge extraction and benefit from the target function if we explicitly learn also the context function. In this paper we propose an algorithm for selection of instances, which indicate context change in target function, and are needed for learning of the context function.

The term 'context' is widely used and ill-defined term in many areas. It has two similar and strict definitions in the field of supervised machine learning. Both definitions divide the features of the objects to be learned into relevant, or predictive, features and contextual ones. Features that influence the prediction alone (without any other) are considered as predictive. Features that influence the prediction only when considered together with some other non-predictive features, which form their context, are considered as contextual. Irrelevant features do not influence the predictions in any combination with predictive or contextual.

In the present paper we use the form $s = (\vec{X}, Y)$ to represent the instance *e*, where vector $\vec{X} = (X_1, ..., X_m)$ represents the *m* features of the instance and *Y* denotes the class of the instance. We use x_i to represent the value of X_i and *y* to represent the value of *Y*.

Under the Turney's definition of context (Turney, 1996) the feature X_i is *strongly relevant* when the assertion $X_i = x_i$ in the context of the assignment for all features except X_i provides us with additional information which we can use to improve the prediction about value of the class *Y*.

This definition in terms of probability distributions looks as follows:

$$p(Y = y | X_i = x_i, S_i = s_i) \neq p(Y = y | S_i = s_i),$$

where each assignment S_i defines a context in which the feature X_i is relevant.

According to Widmer (Widmer, 1997), the feature X_i is *predictive* if the distribution of classes in examples with $X_i = x_i$ is significantly different (as determined by a χ^2 test) from the unconditioned distribution of classes. To define contextual features Widmer introduces 'meta-classes' $\hat{c}_{i,j}$: an instance *e* is in class $\hat{c}_{i,j}$ if feature $X_i = x_j$ is recognized as predictive at the time of classification of *e*. The feature X_i is *contextual* if the distribution of meta-classes $\hat{c}_{i,j}$ in examples with $X_i = x_j$ is significantly different (as determined by a χ^2 test) from the unconditional distribution of the $\hat{c}_{i,j}$.

Previous research deals mainly with the context of features. The paper (Widmer, 1997) brings the notion of context into the area of on-line learning, but the notion of contextual examples is still based on their contextual features. In MetaL(IB) (Widmer, 1997) two examples are considered to be in the same context if their contextual features have the same values. And also there is the lack of research about extracting and exploiting of contextual examples in on-line machine learning due to natural ability of on-line algorithms to adapt the changing situation.

In this paper we assume that the learner passes through different contexts in time, and we develop a formalism which is able to recognize changes between the contexts. For this we define some basic concepts in Chapter 2, then we introduce a formalism to evaluate bounds for some threshold parameters, which are necessary to identify context function, in Chapter 3, then we perform experimental investigation of this formalism in Chapter 4, and concluding in Chapter 5.

2 Basic concepts

In this paper we focus on on-line learning model, introduced in [Littlestone, 1988, Littleston & Warmuth, 1994]. According to this model learning process is divided into *trials* $t = 1,2,3,\ldots,T$. Each trial the learner receives the correspondent example s_i from the unknown a-priory sequence $S = \{s_1,\ldots,s_T\}$ of T examples and has to immediately predict the class, or label $y'_t \in Y$ of the example, where Y denotes the set of all possible classes, or labels. Then the learner receives the response from the 'environment' in a form of the correct class $y_t \in Y$ of example s_t, and updates its internal knowledge to classify the next example better.

In this paper we investigate weighted majority algorithm [Littlestone 1988, Littlestone&Warmuth 1994], that came together with on-line learning model. This algorithm assumes that there exists a set of n predictors $\{h_1, h_2, \ldots, h_n\}$, or hypotheses about correct labeling of input examples. The algorithm maintains a vector of predictors' weights $\{w_1, w_2, \ldots, w_n\}$ and for each trial it presents input example s_t to all predictors, and collects their opinions h_i. Then it calculates the sum $\sum_{i=1}^{n} w_i h_i; h_i = y$ of the weights of predictors that support predicted class y, separately for each possible class. The ensemble predicts class y'_t, for which the sum was maximal.

After making prediction on trial t each predictor h_i receives the correct class of the example and is said to suffer loss value l_i^t.

Definition 1: The *loss value* for *i*-th predictor ($i = 0,1,\ldots,n$) at *t*-th trial is as follows:

$$l_i^t = \begin{cases} 1, if\ h_i^t \neq y_t; \\ 0, if\ h_i^t = y_t. \end{cases}$$

After making a prediction the leaner receives the correct class of the example from the 'environment' and updates the weights with exponential rule: $w_i^{t+1} \leftarrow w_i^t \beta^{l_i^t}$, where β is a fixed parameter $0 < \beta < 1$.

Each example s_t is presented to the learner as a vector $s_t = (x_1,\ldots,x_m)$ of the features of the example. We use the scheme for generating the predictors h_i, similar to described in [Blum, 1997]. For each pair (x', x'') of features we generate a separate predictor $h(x', x'')$ that observes only the values of these two features and predicts the classes from this information. Totally we construct $\binom{m}{2}$ predictors from m features.

Assume that the algorithm has observed already sequence of *T* examples $\{s_1, s_2,\ldots, s_T\}$ from the set S during past *T* trials and now considers example S_{T+1} on trial *T+1*.

Definition 2: The *cumulative loss* for predictor h_i *i*-th predictor ($i = 0,1,\ldots,n$) during *T* trials is as follows:

$$L_i^T = \sum_{t=1}^{T} l_i^t .$$

We expand the basic ideas of the definitions of (Turney, 1996) and (Widmer, 1997) to define context of examples. The idea of the definition in (Turney, 1996) is that the feature is defined as strongly relevant if it provides us with additional information, which we can use to *improve* our prediction about the value of the class. So, the definition in (Turney, 1996) defines the feature as predictive if it *affects* the probability somehow, not necessary in the good direction. It is impossible to implement this idea for batch learning, because we do not know whether the feature improves or not the prediction. In batch learning paradigm, a learner gets the training instances with known correct classes at once, and only then it can classify test examples. Incremental learning assumes that a learner obtains examples one-by-one, classifies them, and only then receives the correct class to compare with its own prediction.

We consider the example as relevant if it influences the distribution of "real" classes from the set *Y* among the examples. This differs from the definitions of (Turney, 1996) and (Widmer, 1997), where the feature is recognized as relevant if it influences the distribution of classes predicted by the algorithm h_0^t without any respect whether these predictions correct ($h_0^t = y_t$) or not. The idea behind our following definition is to eliminate noisy and irrelevant examples, which are abounding in on-line learning tasks.

Definition 3: The *probability* P_i^t *of correct prediction* ($h_i^t = y_t$) is as follows:

$$P_i^1 = 1;\ P_i^t = 1 - \frac{1}{t-1} \cdot L_i^{t-1},\ t = 2,3,\ldots .$$

Definition 4: Example s_t from trial t is relevant with respect to ε to the predictor h_i^t if:

$$P_i^t - P_i^{t+1} \le \varepsilon,\ 0 < \varepsilon < 1 . \qquad (1)$$

The background of Definition 4 is the following: an example, which is relevant to some predictor, must increase the probability of correct classification done by this predictor. However in the same time the Definition allows considering even misclassified examples as relevant ones, if this misclassification keeps the probabilities in bound (1).

Definition 5: The predictor h_i *indicates context change* on trial T (example s_T) if a sequence $S = \{s_{T-r}, s_{T-r+1}, \ldots, s_{T-1}\}$, $S \subset E$ of r previous examples, which were relevant with respect to ε on the correspondent trials, begins to meet the following requirements starting from T-th trial:

$$L_i^{T-1} - L_i^{T-r-1} = r \quad \text{and} \quad P_0^{T-r} - P_0^T > \varepsilon. \qquad (2)$$

The left-hand condition of (2) requires that whole sequence S should be misclassified by i-th predictor. However this requirement alone does not indicate context change if the probability of correct classification for the collective predictor has not decreased essentially during last r trials as requires the right-hand condition of (2).

The idea of Definition 5 is the following. If a predictor has predict classes well during the past trials and now begins to make errors we suppose that something in the "environment" of this concrete predictor has changed. It does not matter if other predictors still work accurately. If we have diverse enough ensemble then we can believe that most of the hypotheses from it are not as much effected by this change as one we are talking about. In such case there is no need to report context change for the work of on-line algorithm. However if most of the predictors begin to predict wrong, then we suppose that the context changed rather than the hypotheses make correlated errors together.

3. Context change for on-line learning

First we prove a Theorem which bounds the loss that the on-line algorithm should get over the sequence of recent examples to be able to indicate context change on trial T according to Definition 5. Then we interpret our bounds and definitions for the Weighted Majority on-line algorithm. We end this chapter by the definition of a context function for Weighted Majority algorithm.

Theorem 1: If the predictor h_i^T indicates context change according to Definition 5, then the amount q of examples, misclassified by this algorithm during the sequence S, is bounded with the following:

$$\varepsilon \cdot (T-1) < q \le \varepsilon \cdot (T-1) + (1-\varepsilon).$$

Proof:

We first rewrite the right-hand condition of (2) according to Definition 3:

$$1 - \frac{1}{T-r-1} \cdot L_0^{T-r-1} - 1 + \frac{1}{T-1} \cdot L_0^{T-1} > \varepsilon,$$

$$\frac{1}{T-1} \cdot L_0^{T-1} - \frac{1}{T-r-1} \cdot L_0^{T-r-1} > \varepsilon,$$

$$\frac{T \cdot L_0^{T-1} - r \cdot L_0^{T-1} - L_0^{T-1} - T \cdot L_0^{T-r-1} + L_0^{T-r-1}}{(T-1) \cdot (T-r-1)} > \varepsilon,$$

$$\frac{T\cdot(L_0^{T-1}-L_0^{T-r-1})-r\cdot(L_0^{T-1}-L_0^{T-r-1})-(L_0^{T-1}-L_0^{T-r-1})}{(T-1)\cdot(T-r-1)}>\varepsilon,$$

$$\frac{(T-r-1)\cdot(L_0^{T-1}-L_0^{T-r-1})}{(T-1)\cdot(T-r-1)}>\varepsilon,$$

$$\frac{L_0^{T-1}-L_0^{T-r-1}}{T-1}>\varepsilon. \qquad (3)$$

Taking into account that amount q of examples, misclassified by the algorithm during the sequence S of r trials is equal to:

$$q=L_0^{T-1}-L_0^{T-r-1},$$

we can rewrite (3) as follows:

$\frac{q}{T-1}>\varepsilon$, and thus:

$$q>\varepsilon\cdot(T-1). \qquad (4)$$

On the other hand, according to Definition 5 the condition from the right-hand of (2) holds starting from T-th trial. We can write the following:

$$P_0^{T-r-1}-P_0^{T-1}\le\varepsilon.$$

Making similar transformations as above we receive:

$$1-\frac{1}{T-r-2}\cdot L_0^{T-r-2}-1+\frac{1}{T-2}\cdot L_0^{T-2}\le\varepsilon,$$

$$\frac{1}{T-2}\cdot L_0^{T-2}-\frac{1}{T-r-2}\cdot L_0^{T-r-2}\le\varepsilon,$$

$$\frac{T\cdot L_0^{T-2}-r\cdot L_0^{T-2}-2\cdot L_0^{T-2}-T\cdot L_0^{T-r-2}+2\cdot L_0^{T-r-2}}{(T-2)\cdot(T-r-2)}\le\varepsilon,$$

$$\frac{T\cdot(L_0^{T-2}-L_0^{T-r-2})-r\cdot(L_0^{T-2}-L_0^{T-r-2})-(L_0^{T-2}-L_0^{T-r-2})}{(T-2)\cdot(T-r-2)}\le\varepsilon,$$

$$\frac{(T-r-2)\cdot(L_0^{T-2}-L_0^{T-r-2})}{(T-2)\cdot(T-r-2)}\le\varepsilon,$$

$$\frac{L_0^{T-2}-L_0^{T-r-2}}{T-2}\le\varepsilon. \qquad (5)$$

Taking into account that amount of examples, misclassified by the algorithm during the sequence of r trials from $T-r-1$ to $T-1$ is equal to $q-1$, because the example s_{T-r-1} should be classified correctly according to Definition 5. It means that:

$$L_0^{T-2}-L_0^{T-r-2}=q-1,$$

and we can rewrite (5) as follows:

$$\frac{q-1}{T-2} \le \varepsilon \text{, and thus:}$$

$$q \le \varepsilon \cdot (T-2) + 1,$$

$$q \le \varepsilon \cdot (T-1) + (1-\varepsilon). \qquad (6)$$

From (4) and (6) we receive:

$$\varepsilon \cdot (T-1) < q \le \varepsilon \cdot (T-1) + (1-\varepsilon). \qquad (7)$$

Taking into account that $0 < \varepsilon < 1 << T$ we can locate no more than one integer within bounds (7).■

For example if $T = 100$ and $\varepsilon = 0.05$, then $q = 5$. It means that some predictor reports context change if it is making the uninterrupted sequence of misclassification errors during which the on-line algorithm makes 5 errors up to *100*-th trial.

Definition 6: *Context* of an on-line learned sequence of examples *changes* on trial T if the collective predictor h_0^T indicates context change.

According to Definitions (5) and (6) and Theorem 1, the context of an on-line learned sequence changes starting from example s_T if the on-line algorithm misclassifies previous q examples $s_{T-q}, s_{T-q+1}, \ldots, s_{T-1}$, where q is the integer within bounds (7).

Definition 7: *Context function CF(t)* of an on-lined learned sequence of examples is as follows:

$$CF(1) = 1;$$

$$CF(t) = \begin{cases} 1, & \text{if the context changes on trial } t; \\ 0, & \text{otherwise.} \end{cases}$$

This is the simplest possible context function, which only indicates places where context changes and gives no other information.

4. Experiments on Vowel dataset

Context function that indicates only context change can be used to identify speaker change in speech recognition task. Our task is not only to recognize the words, spoken by the speakers, but also to recognize when the speaker changes.

Vowel dataset is available from UCI Repository and contains information about the speakers that pronounce eleven vowels. Each speaker pronounces this set of vowels six times, and then new speaker continues. Thus, the dataset contains 11 classes, and each speaker is presented with 66 examples in the dataset. Each vowel is presented with ten numeric features that correspond to different frequencies in digitized vowels.

In our experiments we use sliding 1-Nearest Neighbor classifier as predictors $h_i(x', x'')$, each generated over a pair of features. Each classifier stores only 20 recent examples in the memory and predicts from them using simple Euclidean distance measure to find nearest neighbors. Parameter β was set to $\beta = \frac{1}{2}$.

We also set the number q of misclassified examples (Theorem 1) to $q = 5$. When the predictor generated wrong predictions for last q examples then it is marked as indicating context change, and context change flag is set to 1. If half or more of predictors identify context change then we report context change for the whole ensemble and clear individual context change flags of the predictors.

Figure 1 shows experimental results on the Vowel dataset. According to on-line learning model the learner receives the examples one-by-one, and first 66 examples belong to speaker 1, next 66 – to speaker 2, and so on up to speaker 15. The examples are shown on the horizontal axis in the figure, and the examples of context change are marked with ticks and numbers.

Upper line on the figure shows accuracy of the weighted majority ensemble, measured over the last 20 trials and marked on the left axis. Context change flag of the ensemble, referenced as context function in Definition 7, is presented with a lower line on the figure, that is marked over the right axis.

We see, that our algorithm identifies all the moments of speaker change.

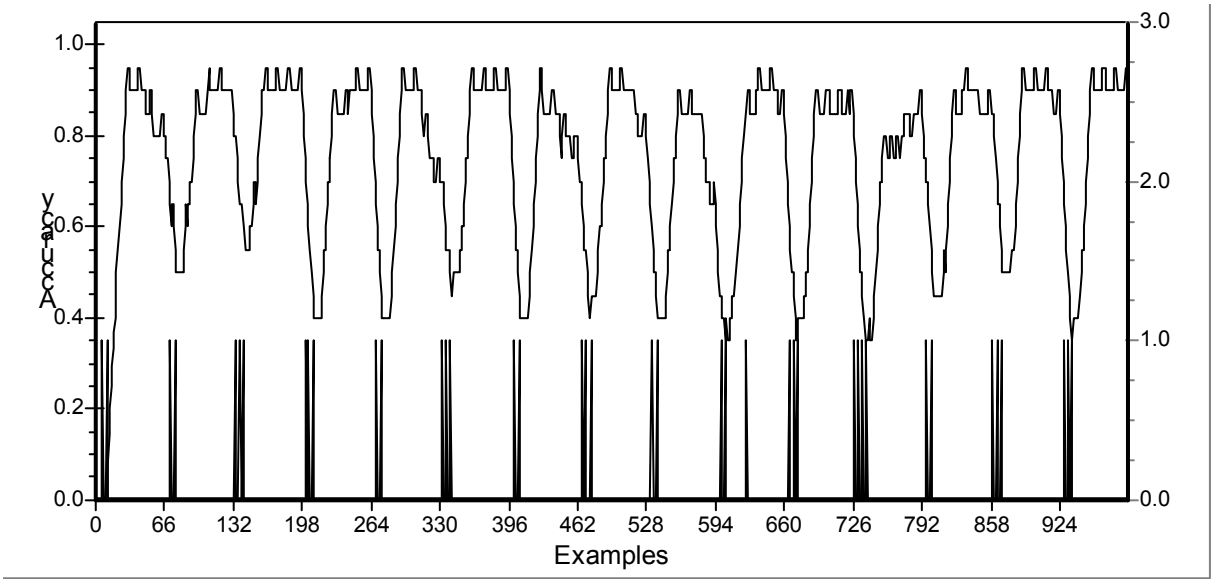

Figure 1. Vowel dataset

5. Conclusion

The proposed formalism in conjunction with weighted majority algorithm select instances indicating context change, and the experiments show that the formalism is able to identify all the moments of context change on the Vowel dataset.

The paper rises two questions. The first one is how to use context function derived from the selected examples to improve the classification accuracy? What types of context functions exist and which of them can be successfully learned with parameter ε?

Acknowledgement: This research is partly supported by the Grant from the Academy of Finland.

References

Blum, A. (1997), Empirical Support for WINNOW and Weighted-Majority Algorithms: Results on a Calendar Scheduling Domain, Machine Learning 26(1), pp. 5-24.

Littlestone, N. and Warmuth, M. K. (1994), The Weighted Majority Algorithm, Information and computation, 108(2):212-261.

Merz, C. and Murphy, P. (1998), UCI Repository of Machine Learning Databases.

Turney, P. (1996a), The identification of context-sensitive features: a formal definition of context for concept learning. In Proceedings of the Workshop on Learning in Context-Sensitive Domains at the Thirteenth International Conference on Machine Learning ICML-96, Bari, Italy, July 3-6.

Widmer, G. (1997), Tracking Context Changes through Meta-Learning. Machine Learning 27(3), p. 259-286.