Pattern Recognition Letters 70 (2016) 38–44

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

An adaptive streaming active learning strategy based on instance weighting☆

Mohamed-Rafik Bouguelia*, Yolande Belaïd, Abdel Belaïd

Université de Lorraine – LORIA, UMR 7503, Vandoeuvre-les-Nancy F-54506, France

ARTICLE INFO

Article history:
Received 17 February 2015
Available online 27 November 2015

Keywords:
Classification
Data stream
Active learning
Instance weighting
Uncertainty

ABSTRACT

This paper addresses stream-based active learning for classification. We propose a new query strategy based on instance weighting that improves the performance of the active learner compared to the commonly used uncertainty strategies. The proposed strategy computes the smallest weight that should be associated with new instance, so that the classifier changes its prediction regarding this instance. If a small weight is sufficient to change the predicted label, then the classifier was uncertain about its prediction, and the true label is queried from a labeller. In order to determine whether the sufficient weight is "small enough", we propose an adaptive uncertainty threshold which is suitable for the streaming setting. The proposed adaptive threshold allows the stream-based active learner to achieve an accuracy which is similar to that of a fully supervised learner, while querying much less labels. Experiments on several public and real world data prove the effectiveness of the proposed method.

1. Introduction

In usual supervised learning methods, a classification model is built by performing several passes over a static dataset with sufficiently many labelled data. First, performing several passes over a static dataset is not possible in the case of data streams where data is massively and continuously arriving from an infinite-length stream. Because of that, several methods for data stream classification have been proposed [1]. Second, the fully supervised methods need a large enough set of manually labelled instances for the learning to be efficient, but manual labelling is costly and time consuming. Semi-supervised learning methods such as those surveyed in [11], can learn using both labelled and unlabelled data, and can therefore be used to alleviate the cost of labelling many instances. However, instead of randomly selecting the instances to be labelled, it is more interesting to let the learning algorithm chose which instances are more convenient for labelling. This is referred to as active learning [8]. To reduce the manual labelling cost, active learning methods query from a human labeller only the class labels of data which are informative (usually uncertain instances). Active learning methods are convenient for data stream classification, since unlabelled data is available cheaply from an infinite-length stream, whereas the class labels are expensive to obtain.

Active learning can use any baseline classifier and its main concern is: how to select instances for which labelling is required. In order to select informative instances, several active learning strategies have been studied in the literature. For example, in the query-by-committee strategy [5,6], a set of classifiers is used to predict the label of each unlabelled instance. The most informative instance is the one about which they most disagree. Another strategy is based on the expected model change [9] where the most informative instance is the one that would make the greatest change to the classification model if we know its label. Many other strategies have been proposed in the literature. We refer to [8] for a survey of active learning methods. However, the most widely used active learning strategies are the uncertainty based ones [4]. Uncertainty strategies select those instances for which the classifier is most uncertain how to label. These instances correspond to the ones that lie close the decision boundary of the current model. Uncertainty of an underlying model can be measured in several ways, based on posterior probabilities or scores returned by the baseline classifier.

The method we propose is different. First, we propose a new uncertainty measure that improves the performance of the active learner compared to the commonly used uncertainty measures. The proposed strategy determines the smallest importance weight required for the prediction to switch to another label. If a small weight is sufficient to change the prediction, then the classifier is uncertain, and the true label is queried from a labeller. Second, we propose an adaptive uncertainty threshold which is convenient for evolving streams and gives a compromise between the error rate and the number of queried labels.

☆ This paper has been recommended for acceptance by Prof. M.A. Girolami.

* Corresponding author. Tel.: +33 623 73 71 05.
E-mail address: mohamed.bouguelia@loria.fr, bouguelia.med.rafik@gmail.com (M.-R. Bouguelia).

http://dx.doi.org/10.1016/j.patrec.2015.11.010
0167-8655/

This paper is organised as follows. In Section 2 we give background on active learning by focusing on the uncertainty based strategies. In Section 3 we first describe our proposed uncertainty query strategy based on instance weighting, then, we propose an adaptive threshold for a stream-based active learning. In Section 4 we present the experimental evaluation. Finally, we conclude and explain the future work in Section 5.

2. Preliminaries

Let $X \subseteq \mathbb{R}^d$ be a d dimensional feature space. The input $x \in X$ is called an *instance* and represents a data point in the feature space X. Let Y be a finite set of classes where each class $y \in Y$ is presented as a discrete value called *class label*. The classifier is then presented as a classification function h that associates an instance $x \in X$ to a class $y \in Y$ (see Eq. (1)).

$$h : \left| \begin{array}{ccc} \mathbf{X} & \longrightarrow & \mathbf{Y} \\ x & \longmapsto & y = h(x) \end{array} \right. \tag{1}$$

The conditional probability $p(y|x)$ is called the posterior probability of class y for instance x. A discussion of specific classifiers properties is beyond the scope of this paper. We simply mention that most existing classifiers not only return the predicted class y but also gives a score or an estimate $\hat{p}(y|x)$ of the posterior probability. The main question that arises is how to get a good classifier h, i.e., how to learn h.

In the case of a supervised learning, the algorithm tries to model the relationship between inputs (instances) and outputs (classes) by learning the function $h : X \longrightarrow Y$ using a training dataset where each instance is previously labelled with its true class label. Suppose that we have a large amount of unlabelled instance. A labeller will manually label as much instances as possible to use them as a training dataset. The more we label, the better it is. However, labelling is costly and time consuming. Note that the instances to be labelled are randomly selected by the labeller, i.e., the learning is *passive*.

In order to enrich the quality of the training data and reduce the labelling cost, it is possible to allow interaction between the learning algorithm and the labeller, i.e., the learning is *active*. Unlike a passive learning, active learning allows to select the most informative instances to be manually labelled, in order to converge more quickly to a good classification function h.

In a *batch mode active learning*, the learner is given a set of unlabelled instances $U \subset X$ and has to iteratively select an instance $x \in U$ in order to query its corresponding class label $y \in Y$ from a labeller. In a *stream-based active learning*, at each time step t, the learner receives a new unlabelled instance $x \in X$ from an infinite-length data stream and has to decide (at time t) whether or not to query the corresponding class label $y \in Y$. If the label y of x is queried, the labelled instance (x, y) is used to update the classification model h. Otherwise, the classifier outputs the predicted label $y = h(x)$. In this way, the goal is to learn an efficient classification model h: $X \rightarrow Y$ using a minimal number of queried labels.

In order to determine for which instance the label should be queried, many active learning query strategies have been studied. The most common ones are the uncertainty sampling based strategies. With these strategies, the instances that are selected for manual labelling are those for which the model h is uncertain about their class. If an uncertain instance x is labelled manually and correctly, two objectives are met: (i) the classifier avoids to output a probable prediction error, and (ii) knowing the true class label of x would be useful to improve h and reduce its overall uncertainty (x is said to be informative).

Let $P_h(y|x)$ be the probability that x belongs to class y. This prediction probability can be directly returned by an intrinsically probabilistic classifier or can be obtained by normalising the prediction scores returned by any classifier. Let y_1 and y_2 be respectively the first and the second most probable classes of x. The most common uncertainty strategies simply select instances with a low confidence $C \in [0, 1]$ (or a high uncertainty $I = 1 - C$) according to one of the following measures:

- Prediction probability $C = P_h(y_1|x)$
- Margin $C = |P_h(y_1|x) - P_h(y_2|x)|$
- Conditional entropy $C = 1 - E$
  where $E = -\sum_{y \in Y} P_h(y|x) \times \log P_h(y|x)$

In the batch mode active learning, the most informative instance among a set of instances U is the one with the smallest confidence (i.e., the highest uncertainty). In the streaming setting, a new instance x is considered to be informative if its confidence is below a threshold $C < \theta$.

3. The proposed active learning method

In this section, we propose a new active learning strategy. The proposed method measures the uncertainty of the classifier using instance weighting. It determines the smallest weight that should be associated with an instance, for the prediction to switch from one label to another. If a small weight is sufficient to change the prediction, then the classifier is uncertain, and the true class label is queried from a labeller.

In Section 3.1 we give a quick overview on instance weighting for classification. In Section 3.2 we formally define our notion of the sufficient weight on which the proposed strategy is based. Section 3.3 shows how to measure the uncertainty by approximating the sufficient weight. Section 3.4 shows static and stream-based active learning using the proposed uncertainty measure. Then we present the proposed adaptive threshold for the stream setting in Section 3.5.

3.1. Instance weighting in classification

Weighting instances is a common practice in classification [3,10]. A weight associated with a labelled instance quantifies the relative importance of this instance compared to the others. Independently of the problem at hand, any baseline learner can be modified so that it gives more importance to instances with a higher weight, which means that the classifier puts more emphasis on getting these instances right. Therefore, instances with a high weight will have more impact on the learned model h. Very often classifiers such as SVM or logistic regression, try to minimise a loss function during learning. In this case, each instance can be weighted within this loss function by simply multiplying the loss and the weight associated to this instance, as described in [2]. Weighting an instance may also be implemented by performing a training on that instance several times instead of one time. Indeed, for several classifiers such as KNN, assigning a weight to an instance simply turns out to duplicate the instance several times, depending on the weight.

Instance weighting is usually used for domain adaptation problems [3] or for classification problems in the case of unbalanced data, by giving a higher weight to instances of minority classes. In this section, we use instance weighting to propose a new efficient active learning strategy.

3.2. The sufficient weight notion

Assume that we have only two classes y_1 and y_2, and that the model h predicts the label y_1 for the instance x, i.e., $h(x) = y_1$. Suppose that we force h to learn x but labelled with the opposite label y_2 (different from the predicted one which was y_1) and weighted by a weight $w \in [0, 1]$. Let us note $\bar{h}_w$ the obtained model. If a small weight w is sufficient for $\bar{h}_w$ to predict y_2, then the model h was uncertain about its first prediction y_1. In other words, the model h is uncertain