The Bayesian Logistic Regression in Pattern Recognition Problems
Under Concept Drift

Pavel Turkov, Olga Krasotkina
Tula State University, Tula, Russia
pavel.turkov@gmail.com
o.v.krasotkina@yandex.ru

Vadim Mottl
Computing Centre of the Russian Academy of Sciences
Moscow, Russia
vmottl@yandex.ru

Abstract

The practice always makes us face the challenge of processing pattern recognition data flows with time-varying target concept, i.e., changing statistical relationship between class memberships and observable characteristics of entities to be perceived by the recognition system. In this paper, a mathematical and algorithmic framework is proposed for handling the concept drift in pattern recognition problems on the basis of the Bayesian treatment of logistic regression as an appropriate mathematical instrument for inferring a time-varying decision rule. The pattern recognition procedure resulting from this approach is a numerical implementation of the general dynamic programming principle, and has the linear computational complexity with respect to the length of the time series, in contrast to the polynomial complexity of pattern recognition procedures of general kind.

1 Introduction

The standard traditional approach to the pattern recognition problem is caused by the tacit assumption that the statistical relationship between the observable characteristics of entities and their hidden class memberships remains the same at the stage of recognition as it was at the stage of training. However, it is typical for practice that properties of the data source change in time, maybe, even during the process of data acquisition, when collecting the training set is not yet accomplished. In the recent literature on data mining, such situations are referred to as concept drift.

A review of existing pattern-recognition methods under concept drift is to be found in [6]. All the methods may be conventionally classed in two groups.

The methods of the first group are based on a single classifier. In these methods, a sliding time window is applied to train a new model from the respective portion of instances. The window length may be constant, as, for example, in FLORA [14], or vary, then the method contains a drift detection mechanism, e.g., ADWIN [2].

The second group is that of the ensemble-based methods, in which several classifiers are learned from the training data and combined then by voting or weight voting. As a rule, algorithms of this group extract information from new training data by one of two ways:

– retraining the old ensemble members on new data, as in the Accuracy Weighted Ensemble (AWE) [13];
– dropping one worst classifier off the ensemble and adding a new classifier learned on incoming data, e.g., the Streaming Ensemble Algorithm (SEA) [11].

It cannot be denied that the existing algorithms are somewhat heuristic, and a great deal of this heuristics is caused by subconsciously exaggerated attention paid to the specificity of each particular application task. In addition, the ensemble-based methods are, as a rule, excessively complicated. In general, it should be concluded that a clear mathematical formulation of the concept drift problem in pattern recognition doesn’t exist.

In this paper, we propose a probabilistic framework for dealing with concept drift aspects in pattern recognition on the basis of the Bayesian treatment of logistic regression as an appropriate mathematical instrument for inferring a time-varying decision rule.

2 The Bayesian approach to the concept drift problem in pattern recognition

Let every instance of the universe \( \omega \in \Omega \) be presented by a point in the linear feature space \( x(\omega) = (x^1(\omega), \ldots, x^n(\omega)) \in \mathbb{R}^n \), and its hidden membership in one of two classes be specified index \( y(\omega) \in \{1, -1\} \). We use here the classical approach to the training problem [12]. According to this approach, the
model of the universe is based on the notion of a linear discriminant function in \( \mathbb{R}^n \), namely, a hyperplane with its direction vector \( a = \mathbb{R}^n \) and threshold \( b = \mathbb{R} \). It is assumed that primarily \( f(x(\omega)) = a^T x + b > 0 \) if \( y(\omega) = 1 \), and \( < 0 \) if \( y(\omega) = -1 \).

But such a model doesn’t take into account the presence of concept drift. The supposed drift of the target concept implies some changes in the universe, and the discriminant hyperplane must be assumed to change in time, too. Therefore, let the behavior of the universe with concept drift be described by time-varying hyperplane \( f_t(x(\omega)) = a_t^T x + b_t \), where \( a_t \) and \( b_t \) are some unknown functions of time. So, it is inevitable to consider every instance \( \omega \in \Omega \) jointly with the time point when it appears \( (\omega, t) \). As a result, the training set should have the form of a set of triplets \( \{(X_t, Y_t, t)\}_{t=1}^T \) where \( (X_t, Y_t) = \{(x_{k,t} \in \mathbb{R}^n, y_{k,t} = \pm 1)\}_{k=1}^{N_t} \) is the subset of instances having appeared simultaneously at the same time point \( t \).

Let us formulate now the probabilistic problem statement. We use the logistic regression approach [5], so, the posterior probabilities of two possible class memberships of an instance \( y_{j,t} = \pm 1 \) can be expressed as a logistic function of its feature vector \( x_{j,t} \):

\[
f(y_{j,t}|x_{j,t}, a_t, b_t) = \frac{\phi(y_{j,t}|x_{j,t}, a_t, b_t)}{\phi(y_{j,t} = 1|x_{j,t}, a_t, b_t) + \phi(y_{j,t} = -1|x_{j,t}, a_t, b_t)},
\]

where the parametric family of probability densities

\[
\phi(y_{j,t}|x_{j,t}, a_t, b_t) = \exp \left[ -\frac{1}{2\sigma^2} \left( 1 - y_{j,t} (a_t^T x_{j,t} + b_t) \right)^2 \right]
\]

is associated with the respective parametric family of discriminant hyperplanes \( a_t^T x_{j,t} + b_t \geq 0 \). For all training instances \( X_t \) and their class labels \( Y_t \) having independently appeared at the time point \( t \) the joint probability function is the product:

\[
\Phi(Y_t|X_t, a_t, b_t) = \prod_{j=1}^{N_t} f(y_{j,t}|x_{j,t}, a_t, b_t).
\]

The key element of our Bayesian approach to the concept drift problem is treating the time-varying parameters of the hyperplane as hidden random processes possessing the Markov property

\[
a_t = qa_{t-1} + \xi_t, M(\xi_t) = 0, M(\xi_t^2) = dI, \quad b_t = b_{t-1} + \nu_t, M(\nu_t) = 0, M(\nu_t^2) = d', \quad q = \sqrt{1-d}, 0 \leq q < 1,
\]

where variances \( d \) and \( d' \) determine the assumed hidden dynamics of the concept, and \( \xi_t \) and \( \nu_t \) are white noises with zero mathematical expectations.

Then, the a priori distribution density of the hidden sequence of hyperplane parameters will have the form:

\[
\Psi(a_t, b_t, t = 1, \ldots, T) = \prod_{t=1}^{T} \psi_t(a_t, b_t|a_{t-1}, b_{t-1}),
\]

\[
\psi_t(a_t, b_t|a_{t-1}, b_{t-1}) \propto \mathcal{N}(a_t|\sqrt{1-\hat{d}a_{t-1}}, dI)\mathcal{N}(b_t|b_{t-1}, d') =
\]

\[
= \frac{1}{d(n/2)(2\pi)^{n/2}} \frac{1}{2\pi d'} \exp \left( -\frac{1}{2d'} (b_t - b_{t-1})^2 \right) \times \exp \left( -\frac{1}{2d} (a_t - \sqrt{1-\hat{d}a_{t-1}})^T (a_t - \sqrt{1-\hat{d}a_{t-1}}) \right).
\]

It is clear that the a posteriori distribution density of the hidden sequence of hyperplane parameters will be proportional to the product

\[
P(a_t, b_t|Y_t, t = 1, \ldots, T) \propto \Psi(a_t, b_t, t = 1, T) \times \Phi(Y_t|X_t, a_t, b_t, t = 1, T).
\]

The sought-for sequence of time-varying parameters \( (a_t, b_t)_{t=1}^T \) is the maximum point of the joint distribution of the parameters and the training set:

\[
(a_t, b_t) = \arg \max_{a_t, b_t} P(a_t, b_t|Y_t, t = 1, \ldots, T) =
\]

\[
= \arg \max_{a_t, b_t} \Psi(a_t, b_t, t = 1, \ldots, T) \times \Phi(Y_t|X_t, a_t, b_t, t = 1, \ldots, T). \tag{1}
\]

It is easy to show that the maximum point of (1) coincides with the minimum point of the following optimization criterion

\[
J_T[(a_t, b_t)_{t=1}^T] =
\]

\[
= \sum_{t=1}^{T} \theta_t(a_t, b_t) + \sum_{t=2}^{T} \theta_t(a_{t-1}, b_{t-1}, a_t, b_t),
\]

where the first term

\[
\theta_t(a_t, b_t) = \sum_{j=N_{t-1}+1}^{N_t} C (1 - y_j (a_t^T x_j + b_t))^2 + \sum_{j=N_{t-1}+1}^{N_t} \log \left[ C (1 - (a_t^T x_j + b_t))^2 \right] + C (1 + (a_t^T x_j + b_t))^2 \right]
\]

fits the class indices of the training set \( y_t \), and the second term

\[
\theta_t(a_{t-1}, b_{t-1}, a_t, b_t) = \frac{1}{d}(a_t - \sqrt{1-\hat{d}a_{t-1}})^T (a_t - \sqrt{1-\hat{d}a_{t-1}}) + \frac{1}{2d'} (b_t - b_{t-1})^2
\]

is responsible for the assumed volatility of time-varying parameters of the discriminant hyperplane.
3 The approximate dynamic-programming based algorithm

It can be easily shown the criterion (2) is convex, and so we can use any methods for its optimization. But there are two problems associated with pattern recognition under concept drift: (a) incremental learning and (b) verifying the nonstationary model by a cross-validation procedure.

The incremental mode of learning is extremely important for non-stationary pattern recognition, because instances are entering the system over a long period of time, when the properties of the analyzed phenomenon may undergo considerable changes. Such problems require multiple recalculation of the optimal goal variables at the succession of time moments, and the computational complexity of the usual algorithm would grow proportionally to the squared length of the training time series.

Fortunately, the criterion (2) belongs to the class of so-called pair-wise separable function. The natural way of optimization for such objective functions is the classical dynamic programming procedure. In [8], in addition to the traditional “forward then back” dynamic programming procedure a new “forward against forward” scheme is proposed. For incremental learning and verifying the nonstationary model, this procedure has the linear computation complexity relative to the length of the data sequence.

The classical dynamic programming procedure is fundamentally based on the assumption that all the variables take values from a finite set, what allows for numerically recomputing the so-called Bellman functions and finding the optimal values. However, it is shown in [9] that, if all the parts of the pair-wise separable objective function are quadratic, the Bellman functions are quadratic, too, their parameters can be easily recalculated, and the dynamic programming procedure becomes possible.

But in our case the observable part (3) of criterion (2) will no longer be quadratic, and the dynamic programming method cannot be applied immediately. In order to save the computational advantages of the dynamic programming procedure, we resort here to the following trick: we heuristically replace the genuine nonquadratic first term of (2) by some appropriate quadratic approximation. So, we come to the following criterion of minimization by \((z_t)_{t=1}^T\):

\[
J(z_1, \ldots, z_T) = \sum_{t=1}^T \zeta_t(z_t) + \sum_{t=2}^T \gamma_t(z_{t-1}, z_t),
\]

where

\[
\zeta_t(z_t) = (z_t - z_t^0)^T Q_t (z_t - z_t^0),
\]

\[
\gamma_t(z_{t-1}, z_t) = (z_t - A z_{t-1})^T U(z_t - A z_{t-1}),
\]

\[
z_t = \begin{bmatrix} a_t \\ b_t \end{bmatrix}; z_t^0 = (Q_t^T Q_t)^{-1} Q_t \sum_{j=N_t-1}^{N_t} g_j, \\
Q_t = \sum_{j=N_t-1}^{N_t} C_{yj} g_j g_j^T, \quad g_j = \begin{bmatrix} y_j x_j \\ y_j \end{bmatrix}, \quad N_0 = 0, \\
U = \text{diag}(1/d, \ldots, 1/d, 1/d^T), \\
A = \text{diag}(\sqrt{1-d}, \ldots, \sqrt{1-d}, \sqrt{1-d^T}).
\]

In such a form, this objective function allows to apply the dynamic programming procedure, and the optimal values of target variables \(z_t\) at each time moment are recurrently evaluated from their previous values \(z_{t-1}\) without the necessity to store all the training instances.

4 Experimental Evaluation

The object of the experimental study in this paper is the problem of filtering e-mails. The behavior of advertisement distributors is constantly changing in time, and, as a result, the spam filters should adapt to the behavior of spammers. Thus, we come to a pattern recognition problem under concept drift.

We took the SPAM E-mail Database [10] from the repository UCI as the dataset. These data contain 4601 instances (e-mails) each of which is characterized by 58 features. The last feature is nominal, it specifies the membership of the instance in one of two classes: spam or non-spam. The values of other features are continuous, they indicate the frequency of occurrence for particular elements (words or characters) in the letter text, or measure the length of sequences of consecutive capital letters. The junk e-mails make 39.4% (1813 instances) of the entire dataset.

To carry out the experiments, we selected 3600 instances (4 groups of 400 instances each) from this database. The test set was composed by 1001 instances. We standardized the features before the experiments. To compare the obtained results, we used some concept drift algorithms from the software environment Massive Online Analysis (MOA) [3].

The first algorithm OzabBagASHT is a bagging-based one, which uses the Adaptive-Size Hoeffding Trees [4]. When the tree size exceeds some maximum value, it restarts building the tree from a new root. The main parameter of this algorithm is the number of models in the bag. For the choice of its optimal value 24, we made several trial experiments on the SPAM E-mail Database, and choose it by the minimum of the classification error.

The second algorithm OzabBagAdwin implements bagging using ADWIN [4]. ADWIN is a change detector and estimator that solves the problem of tracking the average of a stream of bits or real-valued numbers in a well-specified way. The model in the bag is a decision tree for streaming data with adaptive Naive Bayes
percentage of misclassification. In the experiment on the spam e-mail database, classification at leaves. As well as in the previous case, we chose the optimal number of models 18 by the minimum of the classification error in a set of experiments.

SingleClassifierDrift is the third algorithm, which uses the single classifier with EDDM [7] as drift detection method. The decision tree for streaming data with adaptive Naive Bayes classification at leaves was selected as the classifier for training.

The fourth algorithm AdaHoeffdingOptionTree is based on the adaptive decision option tree for streaming data with adaptive Naive Bayes classification at leaves. We set 50 as the maximum number of option paths per node (the influence of this parameter on the error is not detected).

Yet another algorithm chosen for our experiments is LimAttClassifier, which an the ensemble combining restricted hoeffding trees using stacking [1]. It produces a classification model based on the ensemble of restricted decision trees, where each tree is built from a distinct subset of the attributes. The overall model is formed by combining the log-odds of the predicted class probabilities of these trees using sigmoid perceptrons, with one perceptron per class. The minimum of the classification error was got under the following options: when one Adwin detects a change, the worst classifier is replaced; the number of attributes to use per model was 2.

And the last chosen algorithm is NormDistClassifierDrift – the method proposed by us. The parameters $C, d, d'$ were chosen after the trial tests with their different values by the minimum of the error: $C = 1; d = 10^{-8}; d' = 10^{-8}$. The final results presented in Table 1 show that the proposed algorithm turned out to be noticeably better than the other algorithms.

<table>
<thead>
<tr>
<th>Method</th>
<th>Percentage of misclassified instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>OzaBagASHT</td>
<td>22.278</td>
</tr>
<tr>
<td>OzaBagAdwin</td>
<td>20.879</td>
</tr>
<tr>
<td>SingleClassifierDrift</td>
<td>39.361</td>
</tr>
<tr>
<td>AdaHoeffdingOptionTree</td>
<td>23.876</td>
</tr>
<tr>
<td>LimAttClassifier</td>
<td>29.271</td>
</tr>
<tr>
<td>NormDistClassifierDrift</td>
<td>14.785</td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, we have proposed a strictly probabilistic framework for pattern recognition problems under concept drift. The framework results from the Bayesian approach to logistic regression coupled with the Markov assumption on the random drift of the discriminant hyperplane. The method having resulted from this approach exploits the basic principle of dynamic programming. In the experiment on the spam e-mail database, the method has shown an essentially smaller error percentage than the known methods.

6 Acknowledgements

The work is supported by grants of the Russian Foundation for Basic Research No. 11-07-00409 and No. 11-07-00634.

References


