Dynamic Programming for Bayesian Logistic Regression Learning under Concept Drift

Pavel Turkov¹, Olga Krasotkina¹, and Vadim Mottl^{2,*}

¹ Tula State University, Tula, Russia

{pavel-turkov,o.v.krasotkina,vmottl}@yandex.ru

² Computing Center of the Russian Academy of Sciences, Moscow, Russia

Abstract. A data stream is an ordered sequence of training instances arriving at a rate that does not permit to permanently store them in memory and leads to the necessity of online learning methods when trying to predict some hidden target variable. In addition, concept drift often occurs, what means that the statistical properties of the target variable may change over time. In this paper, we present a framework of solving the online pattern recognition problem in data streams under concept drift. The framework is based on the application of the Bayesian approach to the probabilistic pattern recognition model in terms of logistic regression, hidden Markov model and dynamic programming.

Keywords: online learning, concept drift, logistic regression, hidden Markov model.

1 Introduction

A data stream is an ordered sequence of instances that arrive at a rate that does not permit to permanently store them in memory [1]. Unfortunately the standard approach to pattern recognition learning is caused by the tacit assumption that the entire available information is to be processed at once as a single data chunk. However, in the case of a data stream, such a traditional data mining concept would require infinite storage and running time. The data stream leads to the necessity of online learning methods, which assume the updating of an existing classifier or pool of classifiers. The main goal of online learning is to incorporate the knowledge that is intrinsically present in previously observed chunks of data into the existing system.

Pioneer researches concentrated on the creation of algorithms for implementation of online learning with respect to different types of classifiers. The majority of them considered a single particular classifier and confined themselves to a direct tuning of its parameters [2,3]. Many authors proposed one-pass versions of batch learning algorithms. Later, ensembles of classifiers were additionally applied to the problems which had been originally assumed to be solved by single classifier-based incremental learning algorithms [1], [4,5].

* We would like to acknowledge support from grants of the Russian Foundation for Basic Research 11-07-00409, 11-07-00634 and 12-01-31524.

In some applications the target concept may change while being analyzed, in the literature such a problem is known as the concept drift problem. There are some methods for pattern recognition under concept drift [6], that use a sliding window to choose a group of new instances to train a model. Such methods include a single classifier like [7,8] or exploit an ensemble of classifiers [9].

However, it is to be noted that the accurate mathematical statement of concept drift does not exist. The known algorithms are somewhat heuristic, and each specific heuristics is determined by the specificity of the particular practical task.

In this paper, we propose a probabilistic Bayesian approach to the online pattern recognition problem based on treating the concept drift as a hidden Markov process and exploiting the general principle of dynamic programming.

2 Background

Let each of real-world objects hiddenly belong to one of two classes $y = \pm 1$ and be accessible to immediate observation only through its real-valued feature vector $\mathbf{x} \in \mathbb{R}^n$. The classical linear approach to the pattern recognition problem is underlain by observer's concept that there exists a linear discriminant function $\mathbf{a}^T \mathbf{x} + b \geq 0$, namely, a discriminant hyperplane, determined by its direction vector in the feature space $\mathbf{a} \in \mathbb{R}^n$ and threshold $b \in \mathbb{R}$, such that primarily $\mathbf{a}^T \mathbf{x} + b > 0$ if $y = 1$ and < 0 if $y = -1$. It is required to infer the hyperplane's parameters $(\hat{\mathbf{a}}, \hat{b})$ from a finite unordered training set $\{(\mathbf{x}_j, y_j), j = 1, \dots, N\}$ that contains information on both feature vectors and class-memberships of objects.

The problem of batch learning under concept drift differs from the classical statement of the pattern recognition problem in two aspects. First, the training set is no longer treated as completely unordered. Instead, a data stream of single training batches of individual size is considered $\{(\mathbf{x}_{j,t}, y_{j,t}), j = 1, \dots, N_t\}$, which arrive sequentially in time $t = 1, 2, 3, \dots$ but are unordered each inside itself. Second, the observer's concept tolerates a relatively slow drift of the unknown discriminant hyperplane, i.e., its parameters may change in time $(\mathbf{a}_t, b_t)$.

We use the logistic regression approach [10], so, the probabilities of two possible class memberships of an instance $y_{j,t} = \pm 1$ can be expressed as a logistic functions of its feature vector $\mathbf{x}_{j,t}$

$$f(y_{j,t} | \mathbf{x}_{j,t}, \mathbf{a}_t, b_t) = \frac{1}{1 + \exp[-(1/\sigma^2)y_{j,t}(\mathbf{a}_t^T \mathbf{x}_{j,t} + b_t)]}, \quad (1)$$

so that $f(1 | \mathbf{x}_{j,t}, \mathbf{a}_t, b_t) + f(-1 | \mathbf{x}_{j,t}, \mathbf{a}_t, b_t) = 1$. For the entire batch of training instances $\mathbf{X}_t = \{\mathbf{x}_{t,j}, j = 1, \dots, N_t\}$ and their class labels $\mathbf{Y}_t = \{y_{t,j}, j = 1, \dots, N_t\}$ the joint probability function is the product:

$$\Phi(\mathbf{Y}_t | \mathbf{X}_t, \mathbf{a}_t, b_t) = \prod_{j=1}^{N_t} f(y_{j,t} | \mathbf{x}_{j,t}, \mathbf{a}_t, b_t).$$

The key element of our Bayesian approach to the concept drift problem is treating the time-varying parameters of the hyperplane $(\mathbf{a}_t, b_t)$ as a hidden random processes that possesses the Markov property:

$$\begin{aligned} \mathbf{a}_t &= \mathbf{a}_{t-1} + \boldsymbol{\xi}_t, M(\boldsymbol{\xi}_t) = \mathbf{0}, M(\boldsymbol{\xi}_t \boldsymbol{\xi}_t^T) = d\mathbf{I}, \\ b_t &= b_{t-1} + \nu_t, M(\nu_t) = 0, M(\nu_t^2) = d'. \end{aligned} \quad (2)$$

Here ξ_t and ν_t are independent white noises with zero mathematical expectations, whose variances d and d' determine the assumed hidden dynamics of the concept.

The online learning protocol suggests that when the current training batch $(\mathbf{X}_t, \mathbf{Y}_t)$ comes, the observer retains only the last estimate of the discriminant hyperplane $(\hat{\mathbf{a}}_{t-1}, \hat{b}_{t-1})$ inferred from the precedent part of the data stream $(\mathbf{X}_s, \mathbf{Y}_s)_{s=1}^{t-1}$ that is no longer stored in the memory, and his/her task is to immediately recompute the current estimate with respect only to the new information $(\hat{\mathbf{a}}_t, \hat{b}_t) = \mathcal{F}[(\hat{\mathbf{a}}_{t-1}, \hat{b}_{t-1}), (\mathbf{X}_t, \mathbf{Y}_t)]$. In order to maximally save the statistical advantages of off-line learning $(\hat{\mathbf{a}}_t, \hat{b}_t) = \mathcal{F}[(\mathbf{X}_s, \mathbf{Y}_s)_{s=1}^t]$, let us temporarily assume that the entire prehistory of the data stream is still available.

The a priori distribution density of the hidden sequence of hyperplane parameters will have the form (2):

$$\begin{aligned} \Psi((\mathbf{a}_s, b_s)_{s=2}^t | \mathbf{a}_1, b_1) &= \prod_{s=2}^t \psi_s(\mathbf{a}_s, b_s | \mathbf{a}_{s-1}, b_{s-1}), \\ \psi_s(\mathbf{a}_s, b_s | \mathbf{a}_{s-1}, b_{s-1}) &\propto \mathcal{N}(\underbrace{\mathbf{a}_s}_{\text{math. exp.}} | \underbrace{\mathbf{a}_{s-1}}_{\text{math. exp.}}, \underbrace{d\mathbf{I}}_{\text{covar. matr.}}) \mathcal{N}(\underbrace{b_s}_{\text{math. exp.}} | \underbrace{b_{s-1}}_{\text{math. exp.}}, \underbrace{d'}_{\text{var.}}) = \\ &\frac{1}{d^{n/2}(2\pi)^{n/2}} \exp\left(-\frac{1}{2d}(\mathbf{a}_s - \mathbf{a}_{s-1})^T(\mathbf{a}_s - \mathbf{a}_{s-1})\right) \times \frac{1}{2\pi d'} \exp\left(-\frac{1}{2d'}(b_s - b_{s-1})^2\right). \end{aligned}$$

If we assume that there is no a priori information on the first value of the parameter vector $(\mathbf{a}_1, b_1)$, then the a posteriori distribution density of the entire hidden sequence of hyperplane parameters will be proportional to the product $P((\mathbf{a}_s, b_s)_{s=1}^t | (\mathbf{X}_t, \mathbf{Y}_t)_{s=1}^t) \propto \Phi((\mathbf{Y}_s)_{s=1}^t | (\mathbf{X}_s, \mathbf{a}_s, b_s)_{s=1}^t) \Psi((\mathbf{a}_s, b_s)_{s=1}^t)$.

The sought-for estimate of time-varying parameters $(\hat{\mathbf{a}}_s, \hat{b}_s)_{s=1}^t$ is the maximum point of the joint distribution of the parameters and the training set:

$$(\hat{\mathbf{a}}_s, \hat{b}_s)_{s=1}^t = \arg \max_{\mathbf{a}_s, b_s} P((\mathbf{a}_s, b_s)_{s=1}^t | (\mathbf{X}_s, \mathbf{Y}_s)_{s=1}^t) = \arg \min_{\mathbf{z}_s} J_t((\mathbf{z}_s)_{s=1}^t), \quad (3)$$

$$\begin{aligned} J_t((\mathbf{z}_s)_{s=1}^t) &= \underbrace{\sum_{s=1}^t \sum_{j=1}^{N_s} \ln(1 + \exp(-C \mathbf{g}_{j,s}^T \mathbf{z}_s))}_{\eta_s(\mathbf{z}_s)_t} + \\ &+ \underbrace{\sum_{s=2}^t (\mathbf{z}_s - \mathbf{z}_{s-1})^T \mathbf{U} (\mathbf{z}_s - \mathbf{z}_{s-1})}_{\gamma_s(\mathbf{z}_{s-1}, \mathbf{z}_s)} = \sum_{s=1}^t \eta_s(\mathbf{z}_s) + \sum_{s=2}^t \gamma_s(\mathbf{z}_{s-1}, \mathbf{z}_s) \rightarrow \min_{\mathbf{z}_s}, \end{aligned} \quad (4)$$

$$\mathbf{z}_s = \begin{bmatrix} \mathbf{a}_s \\ b_s \end{bmatrix} \in \mathbb{R}^{n+1}, \quad \mathbf{g}_{j,s} = \begin{bmatrix} y_{j,s} \mathbf{x}_{j,s} \\ y_{j,s} \end{bmatrix} \in \mathbb{R}^{n+1}, \quad C = \frac{2}{\sigma^2}, \quad \mathbf{U} = \text{Diag}\left\{\frac{1}{d} \cdots \frac{1}{d}, \frac{1}{d'}\right\}.$$

The criterion (4) is pair-wise separable, i.e., has the structure of a sum of functions each of which is associated with one time point (s) or two immediately adjacent time points in their increasing order ($s-1, s$). This computational problem is, generally speaking, that of dynamic programming.

In the online learning mode, only the last element $\hat{\mathbf{z}}_t = (\hat{\mathbf{a}}_t, \hat{b}_t)$ of the estimated sequence $(\hat{\mathbf{z}}_s)_{s=1}^t$ is of interest at each current time point t . The main notion of dynamic programming is that of the sequence of Bellman functions $\tilde{J}_t(\mathbf{z}_t)$ whose minimum points yield just the required online estimates:

$$\tilde{J}_t(\mathbf{z}_t) = \min_{\mathbf{z}_1, \dots, \mathbf{z}_{t-1}} J_t\left((\mathbf{z}_s)_{s=1}^t\right), \quad \hat{\mathbf{z}}_t = \arg \min_{\mathbf{z}_t} \tilde{J}_t(\mathbf{z}_t). \quad (5)$$

The fundamental property of Bellman functions is the almost evident equality

$$\tilde{J}_t(\mathbf{z}_t) = \eta_t(\mathbf{z}_t) + \min_{\mathbf{z}_{t-1}} \left[\gamma_t(\mathbf{z}_{t-1}, \mathbf{z}_t) + \tilde{J}_{t-1}(\mathbf{z}_{t-1}) \right], \quad (6)$$

which would immediately suggest an algorithm of their sequential computation if only there existed a simple way of solving the optimization problem in (6) with real-valued vectors $\mathbf{z}_{t-1}$.

If all the items of the pair-wise separable learning criterion (4) were quadratic, each of the Bellman functions (5)-(6) would be also quadratic

$$\tilde{J}_t(\mathbf{z}_t) = (\mathbf{z}_t - \tilde{\mathbf{z}}_t)^T \tilde{\mathbf{Q}}_t (\mathbf{z}_t - \tilde{\mathbf{z}}_t) + \tilde{c}_t. \quad (7)$$

and easily computable in terms of their parameters $[\tilde{\mathbf{z}}_t \in \mathbb{R}^{n+1}, \tilde{\mathbf{Q}}_t((n+1) \times (n+1)), \tilde{c}_t \in \mathbb{R}]$ by the classical Kalman-Bucy filter [11,12]. But only the second sum in (4) is quadratic, whereas the first one is formed by logarithmic items. Therefore, an approximation of Bellman functions is required before immediate application of the exceptionally effective Kalman-Bucy filter.

3 Approximate Dynamic Programming for Online Estimating the Time-Varying Discriminant Hyperplane

A method of overcoming the obstacle of non-quadratic Bellman functions (5)-(6) is proposed in [13]. However, that way is based on the assumption that the size of the training set t is much greater than the number of features n , and that both classes are equally represented in the training set. The former of these assumptions is simply satisfiable in the case of a data stream, but the latter one can be mistaken in very many cases. We suggest here a more universal approach to quadratic approximation of logarithmic summands in (4).

Our procedure is based on the assumption that there exists an approximate compact representation of Bellman functions, which permits storing them in the memory. The previous Bellman function $\tilde{J}_{t-1}(\mathbf{z}_{t-1})$ in (6) is non-quadratic because of the logarithmic term in (4). The idea of approximate implementation of the dynamic programming procedure consists in the substitution of the function

$$F_t(\mathbf{z}_t) = \min_{\mathbf{z}_{t-1}} \left[\gamma_t(\mathbf{z}_{t-1}, \mathbf{z}_t) + \tilde{J}_{t-1}(\mathbf{z}_{t-1}) \right] \quad (8)$$

by an appropriate quadratic function

$$\bar{F}_t(\mathbf{z}_t) = (\mathbf{z}_t - \bar{\mathbf{z}}_t)^T \bar{\mathbf{Q}}_t (\mathbf{z}_t - \bar{\mathbf{z}}_t), \quad (9)$$

then the following approximations of the next Bellman function $\tilde{J}_t(\mathbf{z}_t)$ will be quadratic too. It remains to choose appropriate values of the parameters $(\bar{\mathbf{z}}_t, \bar{\mathbf{Q}}_t)$ of the quadratic function $\bar{F}_t(\mathbf{z}_t)$, which would ensure conservation of the main aspects of, generally speaking, non-quadratic original Bellman function. We propose to retain the minimum point of the function $\bar{\mathbf{z}}_t = \arg \min F_t(\mathbf{z}_t)$ and the Hessian at the minimum point $\bar{\mathbf{Q}}_t = \nabla^2 F_t(\bar{\mathbf{z}}_t)$.

Theorem 1. *Let function (8) have the form*

$$F_t(\mathbf{z}_t) = \min_{\mathbf{z}_{t-1}} \left[(\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-1})^T \mathbf{U}(\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-1}) + \sum_{j=1}^{N_{t-1}} \ln(1 + \exp(-C\mathbf{g}_j^T \mathbf{z}_{t-1})) \right].$$

Then the parameters $(\bar{\mathbf{z}}_t, \bar{\mathbf{Q}}_t)$ of approximation (9) are determined as

$$\bar{\mathbf{Q}}_t = \tilde{\mathbf{Q}}_{t-1} (\mathbf{A}\mathbf{U}\mathbf{A} + \tilde{\mathbf{Q}}_{t-1})^{-1} \mathbf{U}, \quad \bar{\mathbf{z}}_t = \bar{\mathbf{Q}}_t \left[(\mathbf{A}\mathbf{U}\mathbf{A} + \tilde{\mathbf{Q}}_{t-1})^{-1} \mathbf{A}\mathbf{U}\tilde{\mathbf{Q}}_{t-1} \bar{\mathbf{z}}_{t-1} \right].$$

4 Experimental Results

In order to test our algorithm, we got the Luxembourg dataset by Zliobaite [16], which is constructed using the European Social Survey on internet usage [15]. The dataset is set of 1901 answers to the survey questionnaire, each represented by 31 real-valued features transformed from the original categorical variables in the survey format, and supplied by a time stamp over a 5 year period. The task is to classify the instances with respect to the internet usage high or low under the assumption of concept drift, i.e., internet usage changing in time.

For the experiment, we took the data part from the 851th instance to 1901th one, within which the time increment is relatively constant, because our a priori model of concept drift (2) does not observe time stamps. The entire time series was separated into 105 batches of 10 instances each.

In the process of online classifier evaluation, before training on the next batch, we computed the error rate on it with the current decision rule. The total error was calculated as the average value of error rates in all the successive batches.

Table 1 shows the comparative results of our algorithm abbreviated as **LogitClassifierDrift** with two algorithms of concept drift realized in the software environment Massive Online Analysis (MOA), see [17] for detail.

Table 1. The experimental results: Luxembourg dataset

Algorithm	Classification error, %
OzaBagAdwin (16 models)	14.519
SingleClassifierDrift	16.36
LogitClassifierDrift	14.381

5 Conclusions

In this paper we present the mechanism for the problem under concept drift with data streams. This mechanism is based on the Bayesian approach to the logistic regression in the finite-dimensional feature space under Markov assumption on the drifting discriminant hyperplane. The error rate indicates that the proposed algorithm outperforms the benchmark ones in the dataset of the European Social Survey on internet usage.

References

1. Polikar, R., Upda, L., Upda, S.S., Honavar, V.: Learn++: an incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 31(4), 497–508 (2001)
2. Mongillo, G., Deneve, S.: Online learning with hidden Markov models. *Neural Computation* 20, 1706–1716 (2008)
3. Florez-Larrahondo, G., Bridges, S., Hansen, E.A.: Incremental estimation of discrete hidden Markov models based on a new backward procedure. In: *Proceedings of the 20th National Conference on Artificial intelligence*, vol. 2, pp. 758–763 (2005)
4. Ulas, A., Semerci, M., Yildiz, O.T., Alpaydin, E.: Incremental construction of classifier and discriminant ensembles. *Information Sciences* 179, 1298–1318 (2009)
5. Kapp, M., Sabourin, R.R., Maupin, P.: Adaptive incremental learning with an ensemble of support vector machines. In: *Proceedings of the 20th International Conference on Pattern Recognition*, Istanbul, Turkey, pp. 4048–4051 (2010)
6. Elwell, R., Polikar, R.: Incremental Learning of Concept Drift in Nonstationary Environments. *IEEE Transactions on Neural Networks* 22(10), 1517–1531 (2011)
7. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. *Machine Learning* 23, 69–101 (1996)
8. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: *SIAM International Conference on Data Mining* (2007)
9. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: *Proceedings of the Ninth ACM SIGKDD International Conference KDD 2003*, pp. 226–235. ACM Press (2003)
10. Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus (2006)
11. Markov, M., Krasotkina, O., Mottl, V.: Time-varying regression model with unknown time-volatility for nonstationary signal analysis. In: *Proceedings of the 8th IASTED International Conference on Signal and Image Processing*, Honolulu, Hawaii, August 14–16, pp. 534–196 (2006)
12. Grewal, M.S., Andrews, A.P.: *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley (2008)
13. Turkov, P., Krasotkina, O., Mottl, V.: Bayesian Approach to the Concept Drift in the Pattern Recognition Problems. In: Perner, P. (ed.) *MLDM 2012. LNCS (LNAI)*, vol. 7376, pp. 1–10. Springer, Heidelberg (2012)
14. Street, W., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: *Proc. 7th Int. Conf. on Knowledge Discovery and Data Mining KDD 2001*, pp. 377–382 (2001)
15. Jowell, R., The Central Coordinating Team.: *European social survey 2002/2003; 2004/2005; 2006/2007*. Technical Reports, London: Centre for Comparative Social Surveys, City University (2003, 2005, 2007)
16. Žliobaitė, I.: Combining time and space similarity for small size learning under concept drift. In: Rauch, J., Raś, Z.W., Berka, P., Elomaa, T. (eds.) *ISMIS 2009. LNCS (LNAI)*, vol. 5722, pp. 412–421. Springer, Heidelberg (2009)
17. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive Online Analysis. *Journal of Machine Learning Research* 11, 1601–1604 (2010), <http://sourceforge.net/projects/moa-datastream/>