Pattern Recognition 81 (2018) 280–293

Contents lists available at ScienceDirect

Pattern Recognition

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

Variational inference based bayes online classifiers with concept drift adaptation

Thi Thu Thuy Nguyen[a], Tien Thanh Nguyen[a,b], Alan Wee-Chung Liew[a,*], Shi-Lin Wang[c]

[a] School of Information and Communication Technology, Griffith University, Gold Coast Campus, QLD 4222, Australia
[b] School of Applied Mathematics and Informatics, Hanoi University of Science and Technology, Vietnam
[c] School of Information Security Engineering, Shanghai Jiaotong University, Shanghai, China

ARTICLE INFO

Article history:
Received 15 November 2017
Revised 7 March 2018
Accepted 8 April 2018
Available online 9 April 2018

Keywords:
Online learning
Variational inference
Bayesian classifier
Data stream
Concept drift

ABSTRACT

We present VIGO, a novel online Bayesian classifier for both binary and multiclass problems. In our model, variational inference for multivariate distribution technique is exploited to approximate the class conditional probability density functions of data in an online manner. To handle concept drift that could arise in streaming data, we develop 2 new adaptive methods based on VIGO, which we called VIGOw and VIGOd. While VIGOw naturally adapts to any kind of changing environments, VIGOd maximises the benefit of a static environment as long as it does not detect any change. Extensive experiments on big/medium real-world/synthetic datasets demonstrate the superior performance of our algorithms over many state-of-the-art methods in the literature.

1. Introduction

Nowadays very often data come in the form of streams. Examples of such data can be easily seen in many real-world applications like network traffic, sensor networks, web searches, stock market systems and others. Storing large volumes of streaming data in the machine's main memory is often infeasible and traditional offline method where the prediction is made based on learning the entire training dataset at once becomes impractical. Moreover, offline algorithms are not applicable in real-time learning scenarios where a stream of data is arriving, and predictions must be made before all the data is seen. Therefore, online learning is emerging as an efficient machine learning method for large-scale applications, especially those with streaming data. In an online learning process, predictive models can be updated after the arrival of every new data point (one-by-one) in a sequential fashion or defer until a group of points has arrived (minibatch-by-minibatch) to reduce the effect of noise in the data. They do not require the whole data set to be stored or loaded into memory, but just make use of a single/set of observations and then discard them before the next observations are used.

Online learning can deal with many tasks such as classification, regression and clustering. In this paper, we focus on online classification algorithms with full feedback (i.e. supervised learning). An online classification task usually involves the three main steps:

- *Predict*: When a new instance $\mathbf{x}_t$ arrives, a prediction $\hat{y}_t$ is made using the current model L_t.
- *Calculate the suffered loss:* After making the prediction, the true label y_t is revealed, and the loss $l(y_t, \hat{y}_t)$ can be estimated to measure the difference between the learner's prediction and the revealed true label y_t.
- *Update:* Based on the result of the loss, the learner can use the sample $(\mathbf{x}_t, y_t)$ to update the classification model $(L_t \rightarrow L_{t+1})$.

From this framework, we can see that online learning algorithms avoid re-training when adding new data. Besides the requirement of accurate and rapid learning and prediction on-the-fly without storing past instances, another challenge to any online method is that it does not know in advance if the data stream is stationary with stable concepts or evolving over time with changing concepts.

In this paper, we introduce a novel online classifier (VIGO) based on the variational inference (VI) technique. In our framework, two learning phases (learning from past instances (prior information) and from recent instances (through sufficient statistics)) flexibly support each other. They are also naturally separated which offer us the opportunity to focus more on recent information or to detect concept drifts. This resulted in 2 new adaptive

* Corresponding author.
E-mail addresses: thithuthuy.nguyen@griffithuni.edu.au (T.T.T. Nguyen), tienthanh.nguyen2@griffithuni.edu.au (T.T. Nguyen), a.liew@griffith.edu.au (A.W.-C. Liew), wsl@sjtu.edu.cn (S.-L. Wang).

https://doi.org/10.1016/j.patcog.2018.04.007
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

methods named VIGOw and VIGOd, respectively. Our algorithms are *second-order* generative models, where distributions are placed not only on the data of each class but also on the model parameters. They do not require storing more than a single instance in the main memory. We evaluate the performance of our proposed methods by comparing them with recent or well-known online methods including kernel-based DualSGD (Dual space gradient descent) [1], FOGD (Fourier online gradient descent) and NOGD (Nystrom ONLINE GRADIENT DEScent) [2]; ensemble-based BLAST [3]; state-of-the-art second-order linear AROW (adaptive regularisation of weights) [4]; widely used first-order linear PA (passive aggressive learning) [5], the most used decision tree HT (Hoeffding tree) [6]. We also compare our proposed methods with ONBG (online Naïve Bayes for Gaussians) [7]—a first-order generative method. Over and above that, the experiment is extended to estimate the adaptability of VIGOw/d in mining evolving data streams with concept drifts. A number of recent or commonly-used adaptive stream learning methods such as SAM–kNN (kNN classifier with Self Adjusting Memory) [8,9], kNN–PAW (kNN with probabilistic adaptive windowing) [10], ensemble-based DACC (dynamic adaptation to concept changes) [11], and HAT (Hoeffding adaptive tree) [12] are used as benchmark algorithms.

The remainder of this paper is organised as follows. In Section 2, we have a brief review of online methods, especially the ones we used as benchmark algorithms in this paper. After that, the background about Bayesian methods and variational inference techniques is summarised in Section 3. Online variational inference for Gaussian (VIGO) is introduced in Section 4. VIGO with built-in concept drift detector (VIGOd) is the topic of Section 5. Section 6 introduces online variational inference weighted for multivariate Gaussian (VIGOw). Experimental results are provided in Section 7. The final section contains conclusion and suggestions for future work.

2. Related work

2.1. Online classifiers

In this section, we discuss online classifiers in general and about the one we use as benchmark algorithms in our experiments in more details. From the three main steps of online classification, we can see that different online algorithms are mainly distinguished in terms of the different type of loss function $l(y_t, \hat{y}_t)$ and different way of updating $L_t \rightarrow L_{t+1}$. Given a problem instance to be classified represented by a vector $\mathbf{x} = (x_1, x_{2,} \ldots x_D) \in \mathbb{R}^D$, linear methods use the predictive rule: $\hat{y}_t = \text{argmax}_{i\in\{1,\cdots,K\}} \mathbf{w}_i \cdot \mathbf{x}_t$, where K is the number of classes and $\mathbf{w}_i$ is the weight vector of class i $(i = 1, \ldots, K)$. Many popular first-order linear methods such as Perceptron [13], OGD (online gradient descent) [14] and PA (passive aggressive learning) [5] are additive algorithms, i.e., when an instance $\mathbf{x}_t$ is misclassified, the weight vector $\mathbf{w}$ is usually updated by shifting along the direction of $\mathbf{x}_t$: $\mathbf{w} + \alpha_t \mathbf{x}_t \rightarrow \mathbf{w}$, where α_t weighs the misclassified instance. These methods only utilise the first-order information of the received instances, thus maintaining a single point solution for the classification model at any trials. Later on, to better exploit the underlying structures between features, second-order online learning algorithms such as SOP (second-order perceptron) [15], SCW (soft confidence weighted learning) [16], and AROW (adaptive regularisation of weight vectors) [4] have been proposed. Most of the second-order learning algorithms typically assume the weight vector follows a Gaussian distribution $\mathbf{w} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$. They not only find the most likely solution for $\mathbf{w}$ but also the distribution of all possible solutions, hence taking advantage of the training data more efficiently. Although linear methods have high time efficiency, they learn linear prediction models which are not flexible enough for many real-world applications.

The limitation of online linear methods in classifying data with nonlinear dependency has motivated the research in online kernel-based methods, which apply linear models in the kernelized feature space to handle the nonlinear separation of data. Conventionally, for the kernel-based predictive model, a set of support vectors (SV) is maintained in main memory and any misclassified new incoming instances are kept. This results in an unbounded SV set during the online learning process. One notable research direction to tackle this key challenge of kernel online learning is to use a fixed-size budget with different budget maintenance strategies (e.g., removal, projection, or merging) [17]. In another direction, a recent method [2] transforms data from the input space to the random-feature space, and then performed stochastic gradient descent in the feature space to create Fourier online gradient descent (FOGD) and Nystrom online gradient descent (NOGD). Recently, to make online kernel methods more scalable, Dual space gradient descent (DualSGD) [1] utilises random features as an auxiliary space to maintain information from data points removed during budget maintenance.

Another widely used approach to deal with the online classification problems is to apply tree-based models. Among incremental trees, Hoeffding tree (HT) [6] is used the most (especially as base learners of ensemble methods, see e.g. [18]) because it has a good performance guaranteed by Hoeffding bound. However, this guarantee does not work for small datasets.

Being one of the most powerful methods, Bayesian classifiers are very flexible generative algorithms which give access to the posterior class probabilities as well as the full data distribution. This information is especially valuable in an online setting, where samples are discarded after use and cannot be retrieved later. Given a problem instance to be classified represented by a vector $\mathbf{x} = (x_1, x_{2,} \ldots x_D) \in \mathbb{R}^D$, a Bayesian classifier based on Bayes' theorem predicts the label y of $\mathbf{x}$ from the label set $\{1, 2, \cdots, K\}$ as

$$y = \text{argmax}_{k\in\{1,2,\cdots,K\}} p(y = k|\mathbf{x})$$
$$\sim \text{argmax}_{k\in\{1,2,\cdots,K\}} p(y = k) p(\mathbf{x}|y = k)$$

where $p(y = k|\mathbf{x})$ is the posterior probability that $\mathbf{x}$ belongs to the class k, $p(y = k)$ is the prior probability of class k, and $p(\mathbf{x}|y)$ is the class conditional probability density function, respectively. Different Bayesian methods are distinguished by the way they approximate $p(\mathbf{x}|y)$. Naïve Bayes is the simplest Bayesian classifier which is called 'naive' because it assumes independence of the attributes given the label. In ONBG (online Naïve Bayes for Gaussians) [7], every attribute x_i of $\mathbf{x}$ follows a univariate Gaussian distribution $p(x_i|y = k) = \mathcal{N}(x_i|\mu_i, \sigma_i^2)$, $i \in \{1, \cdots, D\}$, and the maximum likelihood estimates of parameters μ_i, σ_i^2 are updated on-the-fly based on the training set coming so far. To estimate the parameters of the approximating distributions for $p(\mathbf{x}|y)$, optimisation techniques using latent variables like expectation-maximisation (EM) and variational inference (VI) (see e.g. [19]) are employed. expectation-maximisation (EM) is a popular two-stage iterative technique for finding the parameters of flexible but complicated distributions like mixture of Gaussians (see e.g. [19]) (where all mixing coefficients are positive), or linear combination of continuous or discrete Gaussians [20,21] where the mixing coefficients can be both negative and positive. While also using the coordinate ascent update like EM, VI is a second-order approach where the distribution of each parameter is estimated instead of just the point estimate. To deal with streaming data and big data applications, some algorithms based on incremental EM (see e.g. [19]) and stochastic VI were developed [22–24]. Variational inference often uses the variational lower bound on the marginal likelihood as an objective function, and stochastic variational inference (SVI) [22–24] applies a variant of stochastic gradient descent to this objective for