ARTICLE IN PRESS
JID: NEUCOM [m5G;February 21, 2018;17:43]
Neurocomputing 000 (2018) 1–12
Contents lists available at ScienceDirect
Neurocomputing
journal homepage: www.elsevier.com/locate/neucom

An effective pattern-based Bayesian classifier for evolving data stream

Jidong Yuan, Zhihai Wang*, Yange Sun, Wei Zhang, Jingjing Jiang

School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China

ARTICLE INFO

Article history:
Received 1 August 2017
Revised 7 December 2017
Accepted 2 January 2018
Available online xxx

Communicated by Dr. Haiqin Yang

Keywords:
Data stream
Frequent pattern
Bayesian
Lazy learning

ABSTRACT

One of the hot topics in graph-based machine learning is to build Bayesian classifier from large-scale dataset. An advanced approach to Bayesian classification is based on exploited patterns. However, traditional pattern-based Bayesian classifiers cannot adapt to the evolving data stream environment. For that, an effective Pattern-based Bayesian classifier for Data Stream (PBDS) is proposed. First, a data-driven lazy learning strategy is employed to discover local frequent patterns for each test record. Furthermore, we propose a summary data structure for compact representation of data, and to find patterns more efficiently for each class. Greedy search and minimum description length combined with Bayesian network are applied to evaluating extracted patterns. Experimental studies on real-world and synthetic data streams show that PBDS outperforms most state-of-the-art data stream classifiers.

1. Introduction

Classification based on patterns has attracted significant attention and research effort in recent years [1–7]. Pattern is a subset of data or a set of items, where an item refers to a pair of attribute-value. Frequent patterns (or itemsets) are generated w.r.t. minimum support. Since frequent pattern combines the set of single features non-linearly and indicates more underlying semantics of data [4], except associative classifier [7], it has also been successfully employed to approximate the joint probability of Bayesian classifier [1,2,8]. However, those previous pattern-based classifiers built on traditional datasets cannot adapt to the highly dynamic and complex data stream environment.

A data stream is an infinite sequence of records, generated in a non-stationary environment continuously. Building a pattern-based classifier on data streams differs from traditional pattern mining of static datasets in the following aspects: (1) Each record should be processed quickly in a limited time and memory consumption due to the characteristics of high speed and infinity. (2) Each record can only be examined once and only a small part of the datasets can be stored since of time and space constraints. Therefore the frequency and completeness of mined patterns in data stream cannot be promised. (3) When a classification request occurs, the algorithm needs to respond it in a timely manner and to avoid misclassification. (4) Concept drift may happen in streaming data, and the algorithm must be able to adapt it.

For mining patterns over data stream, state-of-the-art approaches mainly focus on discovering frequent pattern [9] or its variants, such as closed frequent pattern [10,11], maximal frequent pattern [12], high utility pattern [13] and even emerging pattern [14]. Most of the exiting classifiers for non-stationary data stream pay more attention to adaptive tree models [15,16], decision rules [17], ensemble algorithms [18–21] and *k*NN [22]. However, the above algorithms ignore the potential of incorporating pattern in graphical models.

As is well known, Bayesian classification is a graphical model based on Bayes theorem.[1] The major challenge of Bayesian classification lies in the computation of joint probability $P(\mathbf{x}, y)$. The simplest way to address this issue is the Naive Bayes model (as shown in Fig. 1 (*left*)). It assumes that all the attributes x_i are conditional independent given class y_i:

$$\begin{aligned} P(\mathbf{x}, y_i) &= P(x_1, x_2, \ldots, x_m, y_i) \\ &= P(y_i)\cdot P(x_1|y_i)\cdot P(x_2|y_i)\cdots P(x_m|y_i) \end{aligned} \quad (1)$$

Note that although Naive Bayes classifier could be applied on evolving data stream, it cannot adapt to concept drift directly. To alleviate the strong assumption, researchers proposed the conditional dependence model, *e.g.* Bayesian networks [23,24]. The joint probability of Bayesian network shown in Fig. 1 (*middle*) is:

$$\begin{aligned} P(\mathbf{x}, y_i) &= P(x_1, x_2, x_3, x_4, y_i) \\ &= P(y_i)\cdot P(x_1|y_i)\cdot P(x_2|x_1 y_i)\cdot P(x_3|x_1 y_i)\cdot P(x_4|x_3 y_i) \end{aligned} \quad (2)$$

Besides the low-order dependencies among variables, some higher order dependency models, *e.g.* frequent pattern based Bayesian

* Corresponding author.
E-mail addresses: yuanjd@bjtu.edu.cn (J. Yuan), zhhwang@bjtu.edu.cn (Z. Wang).

[1] $P(y_i|\mathbf{x}) = \frac{P(\mathbf{x}, y_i)}{P(\mathbf{x})} = \frac{P(y_i)\cdot P(\mathbf{x}|y_i)}{P(\mathbf{x})}$

https://doi.org/10.1016/j.neucom.2018.01.016
0925-2312/

Please cite this article as: J. Yuan et al., An effective pattern-based Bayesian classifier for evolving data stream, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.01.016

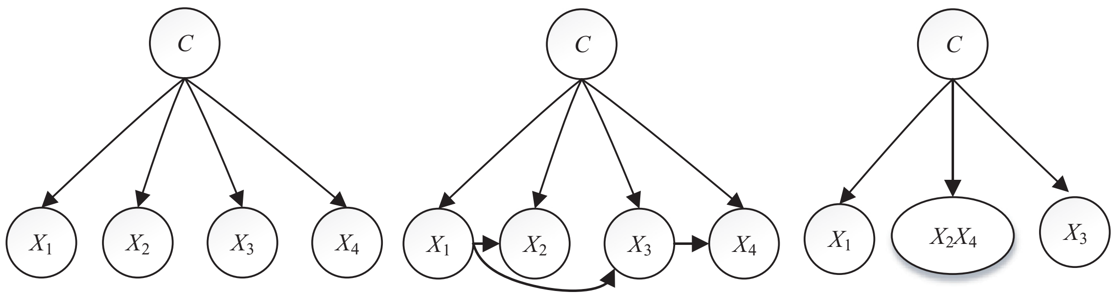

Fig. 1. (*Left*) The structure of Naive Bayes classifier. (*Middle*) The structure of Bayesian Network classifier. (*Right*) The structure of pattern-based Bayesian classifier.

[2,5,8], have also been proposed. Long and overlapped patterns are used to build Bayesian model previously [5,8], while independent pattern based Bayesian classifier presents its effectiveness and flexibility recently [2]. For pattern-based model, the joint probability is (as shown in Fig. 1 (*right*)):

$$P(\mathbf{x}, y_i) = P(x_1, x_2, x_3, x_4, y_i) = P(y_i) \cdot P(x_1|y_i) \cdot P(x_2 x_4|y_i) \cdot P(x_3|y_i) \quad (3)$$

Previous works on streaming data mining either focus on finding interesting patterns efficiently [10–13,25] or creating classifiers effectively [15–17,19,20,22]. However, none of them combines pattern with classifier practically. Unlike these listed methods, our objective is to learn an efficient and effective **P**attern-based **B**ayesian classifier for **D**ata **S**tream (PBDS) that adapts to concept drift. Meanwhile, several challenges remain.

First, due to the space limitation, a window-based model for frequent pattern mining should be selected. To our knowledge, there are three basic stream processing models: landmark window model [9], sliding window model [11,12,25] and damped window model [26]. The sliding window considers a fixed number of stream records, compared with landmark window model, it detects changes in the properties of stream records faster (*e.g.* concept drift), and does not have to assign different weights to stream records (as damped window model), so the sliding window model is employed for efficient pattern mining in this paper.

Second, traditional pattern mining methods may generate excessive number of patterns that are useless in the classification, lacking of expressive power. In contrast, a lazy classifier can provide a more complete description for each record and avoid the extra computation adopted by eager algorithms. Since the computing of lazy classifier is performed on a demand driven basis, only the "useful" portion of the training data (in a sliding window) is mined for generating patterns applicable to the test instance, which increases the chance of achieving the most significant patterns that are useful for classifying the test case.

Third, unlike traditional methods that scan the whole database multiple times to get patterns, frequent pattern mining in a data stream can just check each instance for a single time. In order to store records adequately and mine frequent patterns rapidly, a compact representation of samples is crucial and helpful. Hence a simple but effective summary data structure for each class based on the sliding window model is proposed, which means the probability approximation for PBDS is separately tailored to each class.

Fourth, for a pattern-based Bayesian classifier, a set of long and not overlapped patterns that fully covers the given test case should be found, so it could be considered as a set covering problem essentially. As set covering problems are NP-hard in general, a heuristic pattern extraction mechanism is adopted, which is based on greedy search and the minimum description length (MDL) for the Bayesian classifier to reduce the generation of candidate itemsets, and to ensure the fitness between extracted patterns and original data records. It is also obvious that Naive Bayes is a special case of PBDS.

Table 1
Symbol table.

Symbol	Explanation
$\mathbf{U}$	Data stream
$\mathbf{x}$, T, T_i	A record
y, y_i	Class label of a record
C	Class attribute
X, X_i	Attribute of a record
Ω	Discrete domain of attribute X
$pa(X)$, $pa(X_i)$	parent node of attribute X or X_i
(X_i, x_i), x_i	Item
$\mathbf{z}$	Itemset or pattern
B	Bayesian network
D, D_1, D_2, D'_1	Sliding window
N	Number of records in a sliding window
min_sup	Minimum (class) support
t	Time stamp
m	Length of a record
k	Number of parent nodes
r	Cardinality of attribute X
H, $H(\cdot\|\cdot)$	Entropy, conditional entropy
I	Mutual information
P, P_B, P_D	Probability distribution

Our contributions can be summarized as follows:

- A novel pattern-based Bayesian classifier for data stream is proposed;
- We propose an effective summary data structure for processing records over sliding windows;
- A lazy learning strategy is proposed to find local frequent patterns for each test record;
- A variant of MDL is formulated for pattern-based Bayesian classifier;
- Through an experimental study on real-world and synthetic datasets, the potential of proposed pattern-based Bayesian classifier is shown.

The remaining of this paper is organized as follows. Section 2 introduces preliminary concepts and notations. MDL for pattern-based Bayesian network is illustrated in Section 3. Section 4 describes our summary data structure, establishing the principle of extracting and updating local frequent patterns. Section 5 discusses the approach to estimate the Bayesian joint probabilities. In Section 6, an extensive experimental evaluation of the proposed approach is studied. Threat to works is discussed in Section 7. Finally, Section 8 concludes our work.

2. Definition and notation

In this section, some relevant definitions of this study will be introduced. Table 1 summarizes the notation of this paper, we expand on the definitions below.