Knowledge-Based Systems 000 (2017) 1–13

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A multiple-instance stream learning framework for adaptive document categorization

Yanshan Xiao[a], Bo Liu[b,*], Jie Yin[c], Zhifeng Hao[d]

[a] School of Computers, Guangdong University of Technology, PR China
[b] School of Automation, Guangdong University of Technology, PR China
[c] Information Engineering Laboratory, CSIRO ICT Centre, Australia
[d] School of Mathematics and Big Data, Foshan University, PR China

ARTICLE INFO

Article history:
Received 3 March 2016
Revised 27 December 2016
Accepted 1 January 2017
Available online xxx

Keywords:
Data stream
Multiple-instance learning

ABSTRACT

The task of document categorization is to classify documents from a stream as relevant or non-relevant to a particular user interest so as to reduce information overload. Existing solutions typically perform classification at the document level, i.e., a document is returned as relevant if at least a part of the document is of interest of the user. In this paper, we propose a novel multiple-instance stream learning framework for adaptive document categorization, named MIS-DC. Our proposed approach has the ability of making accuracy prediction at both the document level and the block level, while only requires labeling the training documents at the document level. In addition, our proposed approach can also provide adaptive document categorization by detecting and handling concept drift at a finer granularity when data streams evolve over time, thereby yielding higher prediction accuracy than existing data stream algorithms. Experiments on benchmark and real-world datasets have demonstrated the effectiveness of our proposed approach.

1. Introduction

With the tremendous growth of resources on the Internet, the need to provide useful information to end users has become increasingly critical in the development of document-based recommendation systems. Nowadays, users find themselves confronted with huge amounts of information such as real-time news, blogs and documents in a streaming manner, which are refereed to as document data streams [1–3]. However, only a small fraction of this is actually relevant to the interests of a particular user. In order to reduce the effort a user has to put into determining which information is relevant to his interests, the primary task of *document categorization* is to automatically classify all non-relevant documents from an incoming stream, such that only relevant documents are presented to the end user.

Despite much progress in traditional data mining, classifying time-evolving document data streams still remains a difficult task [4–6]. One reason is that the content on documents may contain information about multiple topics, and possibly other undesirable parts, e.g., text advertisements. This is especially the case among long and multiple-topic documents, with each block corresponding to a different topic [7,8]. Therefore, it is most likely that only some parts on the document are relevant to a particular user interest. If we treat each document as a whole and build a classifier using the features extracted from the entire document, it would inevitably incur a lot of noise into the learning process, which degrades the prediction accuracy of the classifier. Another source of difficulty is that both user interests and the information space are complex and dynamic. On the one hand, a user is typically interested in a variety of topics, which are fluid and interrelated. Such interests may vary rapidly over time, with new topics of interest emerging, and previously interesting topics waning and even becoming obsolete. For example, in a news categorization application, a user might be interested in news articles about “death of Michael Jackson” in July 2009, and then his interest might drift to “Avatar” when the new movie was released in late 2009. On the other hand, the information space itself also dynamically changes over time, filled with new material such as new combinations of concepts, even new concepts, and the occurrences of new events. Therefore, it is a challenging task to design an adaptive classifier that not only provides prediction to identify targeted content, but also matches to changing concepts in the stream [9–11].

In this paper, we propose a novel multiple-instance stream learning framework for document categorization, termed as

[*] Corresponding author.
E-mail addresses: xiaoyanshan@189.cn (Y. Xiao), csbliu@189.cn, csboliu@163.com (B. Liu), Jie.Yin@csiro.au (J. Yin), zfhao@fosu.edu.cn (Z. Hao).

http://dx.doi.org/10.1016/j.knosys.2017.01.001
0950-7051/© 2017 Published by Elsevier B.V.

MIS-DC. To the best of our knowledge, our work is the first to propose a multiple-instance learning framework for classifying time-evolving document data streams. In document data stream categorization, a document is classified as relevant to a particular user interest as long as a certain part of the document is on that topic. It does not require every piece of the document is about that topic. Therefore, it is more appropriate to formulate this task from a multiple-instance angle. Following the description of multiple-instance learning, we consider a document as a bag, and the paragraph blocks in the document as instances in the bag. A document is classified as relevant (positive) if it contains at least one block of content related to the targeted subject of interest. Otherwise, if a document contains all negative (non-relevant) blocks, it is classified as non-relevant. Based on this formulation, the classifier cannot only classify whether a document contains some relevant content, but also label which part of the document is of interest to the user at a finer granularity.

In our proposed framework, we also design a new mechanism to handle concept drift in user interests. Our method builds on chunk-based approaches [12,13], which typically divide a data stream into a number of chunks and combine the base classifiers learnt on individual data chunks to form an ensemble classifier for prediction. The basic assumption made by these works is that the data within a same chunk shares an identical distribution, so that concept drift only happens at the boundaries between chunks. However, this assumption may not hold in real-world applications, where we have no prior knowledge of when the drift would happen. Therefore, we propose a novel approach to detect and handle concept drift that could happen within a data chunk. Specifically, our proposed approach works in three steps. Firstly, we decompose each data chunk in the stream into a sequence of small portions at a lower granularity, and for each portion, an instance-level multiple-instance learning algorithm is utilized to remove irrelevant paragraphs from positive documents so that the remaining paragraphs are of interest to the user. Secondly, we perform core vocabulary analysis to extract positive features from relevant paragraphs in positive documents and detect the occurrence of concept drift between portions. Finally, when concept drift is detected between portions, we construct a transfer learning model to make prediction for incoming data chunks at both the document level and at the block level.

The advantage of our proposed approach can be summarized as follows.

- Our proposed approach can effectively detect and handle concept drift that occurs within a data chunk, thereby yielding higher prediction accuracy than existing data stream algorithms, when data streams evolve over time.
- Our proposed approach enables accuracy predication at both the document level and block level, while only requires the labelling of training documents at the document level.

The rest of the paper is organized as follows. Section 2 discusses the previous works related to our study. Section 3 gives a formal definition of the problem addressed in this paper. Section 4 presents the details of our proposed approach. Section 5 reports the experimental results on real-world datasets. Section 6 concludes the paper and discusses possible directions for future work. To be clear, the basic notations in this paper are defined in Table 1.

2. Related work

In this section, we review the related work in two branches of research areas, including multiple-instance learning and learning concept drift from data streams.

2.1. Multiple-instance learning

Multiple-instance learning [14,15] is a new paradigm in data mining and machine learning that addresses the classification of bags. In multiple-instance learning, the training set contains a series of positive bags B_i^+ and negative bags B_i^-. Each bag consists of a set of instances. A bag is labeled as positive if it contains at least one *positive instance*. Otherwise, it is labeled as negative[1]. The set of positive bags is called the *positive class*, and the set of negative bags is the *negative class*. The objective of multiple-instance learning is to train a distinctive classifier from the training set and utilize the obtained classifier to predict the labels of unknown bags. Multiple-instance learning has been applied to many real-world applications, ranging from drug activity prediction [16], image categorization [17], to text categorization [18,19].

Based on different classification levels, existing methods on multiple-instance learning can be roughly classified into two categories: bag-level approaches and instance-level approaches. For bag-level approaches [20,21], the whole bag is considered as a training unit and the bag-level classifier is built to predict the bag label directly. A typical example of bag-level approaches is DD-SVM [20]. DD-SVM learns a collection of instance prototypes according to a Diverse Density (DD) function and nonlinear mapping is defined to map every bag to a point by using the instance prototypes. Then, SVM is trained to classify these points in the new feature space. For instance-level approaches [22,23], they attempt to distinguish the instance label and utilize the predicted instance label to obtain the bag label. For example, mi-SVM [22] utilizes SVM to classify the instances in the training bags and as a result, each positive bag has at least one instance classified as positive by the classifier. An unknown bag is predicted to be positive if it has at least one instance classified as positive by the obtained classifier.

Most of the existing approaches for multiple-instance learning focuses on building classifiers on static data which assumes that the concept underneath the data is static and unchanged. However, in real-world applications, the concept underneath the data can change over time and how to construct an appropriate multiple-instance learning classifier to classify this time-evolving data becomes a challenging issue. This motivates the work in this paper.

2.2. Learning concept drift from data streams

Addressing changes in concept can be broken down into two subtasks: *concept drift detection* and *concept drift handling*. Concept drift detection deals with the problem of detecting when a significant change in concept has taken place. Concept drift handling is concerned with how a classifier is updated to take account of changes in concept.

There are two main categories of solutions for classifying data streams: incremental learning [1,9,24] and chunk-based learning [2,10,13,25,26]. Incremental learning aims at making use of new data to update the models trained from historical streaming data, so that the learning process can adapt to the changing concepts. Chunk-based learning, on the other hand, divides the data stream into a sequence of data chunks and combines the base classifiers learned on individual data chunks to form an ensemble classifier for prediction. As a result, various ensemble methods, such as weighted voting, dynamic voting, and dynamic selection, have been proposed to determine the weight for each base classifier to construct the ensemble classifier. By combining a group of individual classifiers, ensemble learning has demonstrated the capability of yielding higher prediction accuracy than just using a single classifier.

[1] For the sake of simplicity, we will omit the +/− sign when there is no need for disambiguation.