The Journal of Systems and Software 102 (2015) 158–166

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Countering the concept-drift problems in big data by an incrementally optimized stream mining model

Hang Yang[a,1], Simon Fong[b,*]

[a] Electric Power Research Institute, China Southern Power Grid, China
[b] Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau

ARTICLE INFO

Article history:
Received 7 March 2014
Revised 29 May 2014
Accepted 6 July 2014
Available online 22 July 2014

Keywords:
Concept drift
Data stream mining
Very fast decision tree

ABSTRACT

Mining the potential value hidden behind big data has been a popular research topic around the world. For an infinite big data scenario, the underlying data distribution of newly arrived data may be appeared differently from the old one in the real world. This phenomenon is so-called the concept-drift problem that exists commonly in the scenario of big data mining. In the past decade, decision tree inductions use multi-tree learning to detect the drift using alternative trees as a solution. However, multi-tree algorithms consume more computing resources than the singletree. This paper proposes a singletree with an optimized node-splitting mechanism to detect the drift in a test-then-training tree-building process. In the experiment, we compare the performance of the new method to some state-of-art singletree and multi-tree algorithms. Result shows that the new algorithm performs with good accuracy while a more compact model size and less use of memory than the others.

1. Introduction

Big data has become a hot research topic, and how to mine valuable information from such huge volumes of data remains an open problem. Many research institutes worldwide have dedicated themselves to solving this problem. The solutions differ from traditional data mining methods, where the mining process must be efficient and incremental.

Processing big data presents a challenge to existing computation platforms and hardware. However, according to Moore's Law, CPU hardware may no longer present a bottleneck in mining big data due to the rapid development of the integrated circuit (IC) industry. A well-designed algorithm is crucial in solving the problems associated with big data.

Parallel random access and distributed computing has been a mainstream research topic in the big data era. These techniques generally need high-bandwidth communication within a network environment. For instance, the MapReduce model was proposed by Google to solve the large-scale computing problem. The mechanism of key-value pairs makes it possible to divide the task across distributed nodes using the Map function and then reconstruct the result using the Reduce function. The bridge between Map and Reduce is the Shuffle process, which automatically processes the intermediate results. However, this intermediate process may become a bottleneck of computational efficiency due to the uncertainty of network communication.

A data stream model is usually defined as a model in which data move continuously at high-speed. Most big data can be considered as data streams, in which new data are generated continuously. Data streams contain very large volumes of data, which cannot be stored in either internal or external memory. A one-pass algorithm therefore forms the basis of data stream mining, which briefly stores a sufficient statistical matrix when new data passes, but does not require the full dataset to be scanned repeatedly. A data stream also depicts an infinite big data scenario in which the underlying data distribution of newly arriving data may differ from older data in the real world: the so-called concept-drift problem. For example, click-streams of users' navigation patterns on an e-commerce website may reflect their purchase preferences as analyzed by the system. However, as people's preferences for products change over time, the old model is no longer applicable, resulting in concept drift.

Decision trees are one of the most important data classification techniques. These techniques are widely used because of their ability to interpret knowledge in different domains and present it as a tree-like graph. Decision trees can be distinguished into two categories according to their components: single-tree algorithms and multi-tree algorithms. A single-tree algorithm is lightweight and

* Corresponding author. Tel.: +853 62208922.
E-mail addresses: henry.yh@gmail.com (H. Yang), ccfong@umac.mo, fong_simon@yahoo.com (S. Fong).
[1] Tel.: +86 20 38124519.

http://dx.doi.org/10.1016/j.jss.2014.07.010
0164-1212/

easy to implement and thus favored for data stream environments, although in some cases, a multi-tree algorithm may achieve slightly higher accuracy.

In this paper, we investigate the performance of single-tree learning for concept-drift data streams. Three representative tree inductions are used in this evaluation: VFDT (Domingos and Hulten, 2000), a classic algorithm that pioneered the use of Hoeffding bound to build an incremental decision tree; ADWIN (Bifet and Gavalda, 2007), a start-of-the-art tree model that uses an adaptive-window technique to handle concept drift; and iOVFDT (Yang and Fong, 2012, 2013), a model previously developed by the present authors that balances accuracy, tree size and learning speed. The results show that iOVFDT has good performance for both synthetic and real-world concept-drift data streams. The advantage of the adaptive tie threshold makes iOVFDT suitable for real-world applications.

The paper is organized as follows: Section 1 introduces the research topic; Section 2 reviews some background; Section 3 illustrates different methods of handling concept-drift; Section 4 shows the experiment, analyzes the evaluation results and discusses the comparison; and Section 5 concludes the paper.

2. Background

Data stream is also an infinite big data scenario that the underlying data distribution of newly arrival data may be appeared differently from the old one in the real world, so called concept-drift problem. For example, click-streams of user's navigating e-commerce website may reflect the preferences of purchase through the analysis systems. When people's preferences of product change, however, the old user's behavior model is not applicable any more that the drifting of concepts appears.

The hidden changes in the attributes of data streams will cause a drift of target concept. In terms of the occurring frequency, commonly it can be distinguished in two kinds: abrupt drift and gradual drift. For data streams, the data arrive continuously that the concept-drift is local, for instance, only particular types of attribute may change with time while the others remain the same.

Generally, there are three ways to deal with concept drift: instance selection, instance weighting and ensemble learning (Tsymbal, 2004). Sliding-window is a commonly used technique to handle the drift by moving over recently arrived instances and used the learnt concepts for prediction in the future. Window size is an important parameter for this technique, influencing the reliable of detection of abrupt or gradual drift. FLORA (Widmer and Kubat, 1996), FRANN (Kubat and Widmer, 1994), and TMF (Salganicoff, 1997) are the early window-based systems that use a window of fixed size. Adaptive-Size (Klinkenberg, 2004) and FLORA2 (Widmer and Kubat, 1993) use heuristics to adjust the size of window to the current extent of concept drift. Instance can be weighted in terms of their age and the competence of current concept. However, the paper (Klinkenberg, 2004) shows instance weighting handle concept drift worse than instance selection probably due to over-fitting the data. Ensemble learning maintains a set of concept descriptions and combines them using voting technique to select the most relevant concept from the set.

For concept-drift data stream experiment, the manual of Massive Online Analysis (MOA) (Bifet et al., 2010) introduces some real-world sample data, one of which is the Cover Type data. However, there is not any document that explains clearly when drift appears in the data. The concept-drift problem exists in many real world data streams that how to visualize it is difficult in practical. In order to study the change of relationship between attributes and class, a feature selection of information gain ratio is applied in this test firstly in Fig. 1. The *x*-axis is the number of instances passed. The ranked ratio of the 54 attributes to the class is shown according to the attribute index in *y*-axis. We find the ratios of 300–400 and 400–500 are distinguished that the relation between attributes and class may drift to some extent.

On the other hand, we visualize the characteristic of concept-drift data. The benchmark is a synthetic Hyper-plane dataset, which is a typical experimental data used for concept-drift problem (Bifet and Gavalda, 2007). This data is generated from MOA (Bifet et al., 2010) and the drift width is 1000. Comparing to the other two real world data from UCI, Cover Type and Person Activity data, the result is shown in Fig. 2. The data are trained and tested sequentially with a frequency of 1000 instances. Because we defined the drift zone

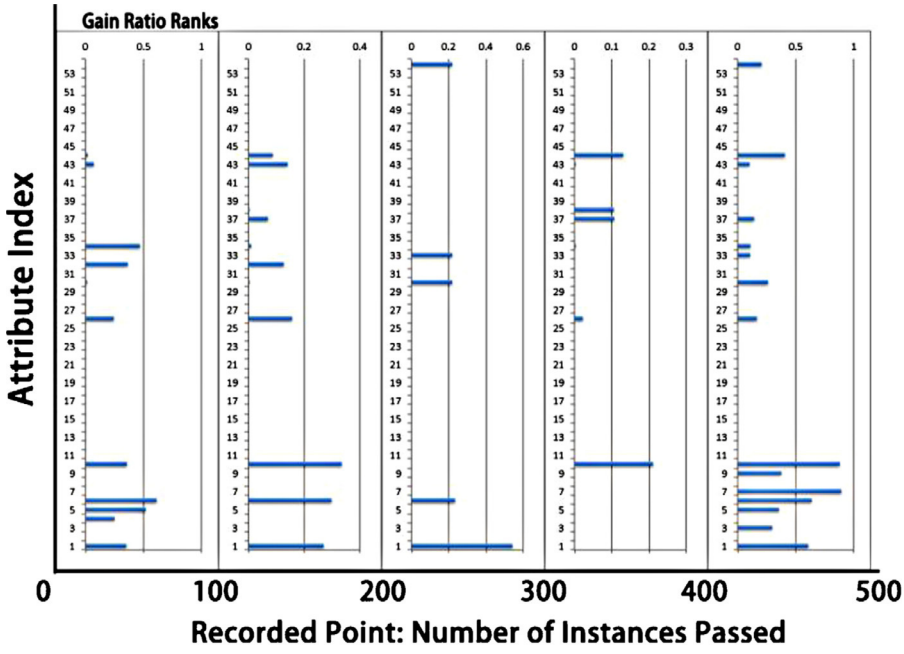

Fig. 1. Visualizing concept-drifts problem in cover type data.