Detecting Changes in Rare Patterns from Data Streams

David Tse Jung Huang\textsuperscript{1}, Yun Sing Koh\textsuperscript{1}, Gillian Dobbie\textsuperscript{1}, and Russel Pears\textsuperscript{2}

\textsuperscript{1} Department of Computer Science, University of Auckland, New Zealand
\{dtjh,ykoh,gill\}@cs.auckland.ac.nz
\textsuperscript{2} School of Computing and Mathematical Sciences, AUT University, New Zealand
rpears@aut.ac.nz

Abstract. Current drift detection techniques in data streams focus on finding changes in streams with labeled data intended for supervised machine learning methods. Up to now there has been no research that considers drift detection on item based data streams with unlabeled data intended for unsupervised association rule mining. In this paper we address and discuss the current issues in performing drift detection of rare patterns in data streams and present a working approach that enables the detection of rare pattern changes. We propose a novel measure, called the $M$ measure, that facilitates pattern change detection and through our experiments we show that this measure can be used to detect changes in rare patterns in data streams efficiently and accurately.

Keywords: Data Stream, Drift Detection, Rare Pattern.

1 Introduction

Mining data streams for knowledge discovery, using techniques such as clustering, classification, and frequent pattern discovery, has become increasingly important. A data stream is an ordered sequence of instances that arrive at a high rate. This characteristic imposes additional constraints on the mining algorithms to be efficient enough to keep up with the fast rate of arrival and also requires an efficient memory usage as not all data instances can be stored in memory. Many techniques that find frequent patterns from data streams have been proposed, such as CPS-Tree \cite{17} and FPStream \cite{6}. Frequent patterns have been widely considered to be informative and useful but in some domains and scenarios rare patterns may be more interesting. Rare patterns are patterns that do not occur frequently and can sometimes be considered as exceptions. Rare patterns often represent irregular behaviors such as frauds. The detection of rare patterns can benefit a wide range of domains such as fraud detection in credit card transactions and auctions. The mining of rare patterns from data streams has been considered in some previous research, such as SRP-Tree \cite{9}.

An important characteristic of data streams is that changes in the underlying distribution can signal important changes in the data stream. Many drift detection techniques have been proposed to detect these changes. However,
these techniques are designed with the focus of detecting drift in data streams that contain class labels which are intended for supervised machine learning methods such as classification. These drift detection techniques (e.g., ADWIN2 [3]) take in binary inputs that are derived from the error rates of a classifier run on the labeled data stream. Because these techniques are designed for use in labeled data streams, they cannot be applied directly onto unlabeled data streams that frequent and rare pattern mining techniques take in as input.

A naive method for detecting changes in patterns would be to mine the stream for a set of patterns at given intervals and then compare the sets of patterns. This is not the suitable approach, especially in the case of rare patterns, where it is often harder and more costly to discover rare patterns from data streams. A more enlightened scheme would be to apply drift detection techniques at an item level. Therefore, instead of running one instance of the drift detection technique (e.g., ADWIN2) on the stream, multiple instances of the technique is run on each separate item (or a subset of the items) found in the unlabeled stream. The binary inputs into ADWIN2 can be derived from the presence or absence of the item in a series of transactions where the binary input 1 would represent that the item occurs in the transaction and the binary 0 would represent the item did not occur in the transaction. For example, consider three transactions: $T_1 : \{a, b, c\}, T_2 : \{a\}, T_3 : \{a, b\}$. The binary inputs used for item $a$ would be $\{111\}$ as item $a$ appears in all three transactions and the input for item $b$ would be $\{101\}$ as it does not occur in transaction $T_2$. Essentially this monitors the support change of the individual items in the stream and a detected change in this case would represent that the item is either occurring in more transactions or occurring in fewer transactions than it did previously. The issue with this method is that only pattern changes caused by support variations in items will be detected. If there is a change in pattern, but without an accompanying change in support of items, the change will not be detected. For example, consider 4 items $\{a, b, c, d\}$ where $\{a, b\}$ always occur together forming a pattern and $\{c, d\}$ always occur together forming another pattern. If the support of these items do not change but now item $a$ occurs with item $c$ and item $b$ occurs with item $d$, then this change will not be picked up by simply monitoring the support of items using current drift detection techniques. In this paper, we describe this form of pattern change as a change in item association.

Motivated by the difficulties and the lack of methods for detecting rare pattern changes from unlabeled data streams, the aim of this paper is to address this problem by proposing a novel approach that enables such detection. We propose a novel $M$ measure that enable the detection of both rare pattern changes caused by support change and item association change. The $M$ measure consolidates the state of association of items into one numerical value and in our approach, instead of monitoring the support of items as described earlier, we monitor the $M$ measure. Through our evaluations, we demonstrate that this overall approach is capable of detecting rare pattern changes.

There are several scenarios where detecting a change in item associations can be useful. For example, consider a stream of data for recording a series of