Knowledge-Based Systems 157 (2018) 34–51

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

High utility drift detection in quantitative data streams

Quang-Huy Duong[*,a], Heri Ramampiaro[a], Kjetil Nørvåg[a], Philippe Fournier-Viger[b], Thu-Lan Dam[a]

[a] Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway
[b] School of Humanities and Social Sciences, Harbin Institute of Technology (Shenzhen), Shenzhen, China

ARTICLE INFO

Keywords:
High utility pattern mining
Data stream
Drift detection
Change detection

ABSTRACT

This paper presents an efficient algorithm for detecting changes (drifts) in the utility distributions of patterns, named High Utility Drift Detection in Transactional Data Stream (HUDD-TDS). The algorithm is specifically suitable for quantitative data streams, where each item has a unit profit, and non-binary purchase quantities are allowed. We propose a method that enables the HUDD-TDS algorithm to be used in an online setting to detect drifts. An important property of HUDD-TDS is that it can quickly adapt to changes in streams, while considering older transactions to be less important than new ones. Furthermore, the proposed method applies statistical testing based on Hoeffding bound with Bonferroni correction in order to ensure that only significant changes are reported to the user. This test allows identifying a change (drift) if the difference between current and the previous time window is significant in terms of utility distribution. In this work, we focus on both local and global utility drifts. A local utility drift is a drift in the utility distribution of a single pattern, whereas a global utility drift is a change in the utilities of all high utility itemsets. In order to be able to compute the similarity of different high utility itemsets to detect drifts, we propose a new distance measure function. The results of our experiments on both real world and synthetic datasets show the feasibility and efficiency of the proposed HUDD-TDS algorithm.

1. Introduction

Frequent Itemset Mining (FIM) is a fundamental research topic in data mining [1]. The task of FIM is to discover all itemsets in a transactional database so that the frequency of the itemsets is no less than a user-specified minimum support threshold. FIM has attracted a lot of attention from researchers and it has been applied in many applications [2,3]. However, in FIM, the unit profits of items are not considered, and the purchase quantities are assumed to be binary in each transaction. This assumption often does not hold in real life. To address the limitation of these studies, the FIM problem has been generalized as the problem of High Utility Itemset Mining (HUIM) [4]. The goal of HUIM is to discover patterns that generate a high profit in static customer transaction databases. The key differences between HUIM and FIM are that each item has a unit profit, and non-binary purchase quantities are allowed. Several studies on HUIM have been conducted [5,6], but most of existing approaches are suitable for pattern discovery in a static database. With more and more data, including customer transactions, being generated in streams, HUIM must also support pattern discovery in dynamic databases.

As with general data streams, streaming transactional data is generally infinite and changes continuously. This combined with the high data generation speed makes mining of a stream of transactional data to discover patterns more challenging than mining a static database. Thus, developing efficient methods and algorithms for analyzing transaction streams is an important research problem [7,8]. Nevertheless, most studies on this topic, including [9,10], have focused on adapting traditional data mining techniques to streams and improving their efficiency to deal with streaming data. Note, however, that the underlying distribution of data objects in a stream generally changes over time [11], thus making such approaches unsuitable. At the same time, detecting changes, called *concept drifts*, is crucial because it allows to discover the latest trends in a stream. A concept drift mainly refers to a significant decrease or increase in the distribution of data objects in a data stream with respect to a given measure [12].

In recent years, incremental and online learning have attracted the attention of many researchers to detect changes due to their numerous real-life applications, including market basket analysis, image processing, outlier detection, and climate monitoring [13,14]. An important challenge of analyzing data streams is that trends may emerge, or

* Corresponding author.
E-mail addresses: huydqyb@gmail.com (Q.-H. Duong), heri@idi.ntnu.no (H. Ramampiaro), noervaag@ntnu.no (K. Nørvåg), philfv8@yahoo.com (P. Fournier-Viger), lanfict@gmail.com (T.-L. Dam).

https://doi.org/10.1016/j.knosys.2018.05.014
Received 11 December 2017; Received in revised form 9 May 2018; Accepted 12 May 2018
Available online 22 May 2018

remain steady over time, and that the streams often contain noise. In other words, to allow decision-makers to quickly react to changes, it is necessary to design efficient algorithms that can detect and monitor these changes in real-time. Nevertheless, although monitoring changes in data streams is widely recognized as important, most existing algorithms have mainly focused on discovering frequent patterns with changing frequencies, rather than considering changes in terms of other meaningful measures, such as the profit generated by the sale of items. To the best of our knowledge, only few approaches have been proposed to detect changes in the utility (profit) distribution of itemsets, where transactions are treated as streaming data. Monitoring such fluctuations in profit is necessary and important in many real-life applications including online retail stores and monitoring stock exchanges.

The work presented in this paper is motivated by the need to address the limitations due to the lack of approaches that fully study the issues with concept drifts in high utility itemsets in data streams. We propose an efficient algorithm called HUDD-TDS (High Utility Drift Detection in Transactional Data Streams), with which we introduce several novel ideas to detect drifts efficiently. We propose a new distance measure function to measure the similarity of different high utility itemsets. In order to quickly adapt to changes in streams, the HUDD-TDS considers weighting factor of older transactions and utilizes statistical testing based on Hoeffding bound with Bonferroni correction. The proposed method detects changes in utility distribution of patterns in quantitative data streams which including both changes in utility distribution of single itemsets and changes in structure of utility distribution of itemsets.

Overall, the main contributions of this work are as follows:

1. We introduce the task of detecting both local and global drifts by considering the utility measure and the recency of transactions in streams, defined as follows:
   - A *local utility drift* is a change in the utility distribution of an itemset (e.g., the utility of an itemset has recently considerably increased or decreased).
   - A *global utility drift* is a change in the total utility distribution of all itemsets (e.g., the sales of products in a retail store have globally considerably increased or decreased).
2. We propose an efficient algorithm for the drift detection task in (1). The proposed algorithm relies on probability theory and statistical testing to identify changes in the utility distribution of itemsets in a quantitative data stream. Moreover, our approach takes into account the evolving behavior of the streams, and we employ a fading function to identify recent trends. Such a fading function is specifically useful in weighing the importance of transactions according to their age.
3. We introduce a new distance measure function called *Dmo* to compare high utility itemsets for detecting drifts. Although *Dmo* is based on the cosine similarity, which is a standard measure for calculating the similarity between vectors, it is more general in that *Dmo* not only considers the difference of vectors in terms of orientation (i.e., vector angles) but also magnitude. As discussed in this paper, this is necessary to address our problem.
4. We conduct an extensive experiment to evaluate the proposed method HUDD-TDS, showing the feasibility, effectiveness, and efficiency of our HUDD-TDS algorithm.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 defines the problem of drift detection and introduces necessary preliminaries. Section 4 presents the proposed approach for detecting changes in the utility distribution of itemsets in a quantitative transaction data stream. Section 5 presents results from an extensive experimental evaluation to evaluate the performance of the proposed algorithm. Finally, Section 6 concludes the paper and outlines our plans for future work.

2. Related work

Detecting concept drifts is an important research problem that has applications in many domains such as flow prediction in industrial systems [15] and information filtering [16]. Numerous approaches have been proposed to detect changes in the distribution of data objects in data streams. Techniques for drift detection [17–19] are generally based on one of the following approaches: sequential analysis [20], statistical process control [21], comparison of two consecutive time windows [22], and contextual approaches [16]. The Hoeffding's Inequality has been used to design several approaches for determining the upper bounds for drift detection. Such upper bounds have been used in algorithms such as the Fast Hoeffding Drift Detection Method for Evolving Data Streams (FHDDM) [23], Hoeffding Adaptive Tree (HAT) [24], and the HAT + DDM + ADWIN [25] algorithm which extends ADaptive sliding WINdow (ADWIN) algorithm [26] and the Drift Detection Method (DDM) [20]. ADWIN is one of the most popular change detection, and it uses sliding windows to maintain the distribution and detect changes, whilst the DDM uses an online learning model to control the online error-rate and detect changes. Frías-Blanco et al. [27] proposed online and non-parametric drift detection methods using several bounds based on Hoeffding's Inequality. The algorithm can detect concept drifts based on the movements of distribution averages in streaming data. The algorithm uses counters to maintain information for detecting drifts. The time complexity of this approach is constant ($\mathcal{O}(1)$ for processing each data point in the stream).

In HUIM, several algorithms have been proposed for discovering high utility itemsets [4,5,28,29] in static databases. Early HUIM algorithms adopt a two-phase approach to discover patterns in customer transaction databases. For example, Two-Phase [5], IHUP [28] and UPGrowth [30] are two-phase algorithms. Although a two-phase approach is useful and guarantees completeness in mining of high utility itemsets, a drawback is that the two-phase approach generates a huge amount of candidates, requiring a significant amount of memory to maintain the candidates. This greatly degrades the performance of the two-phase algorithms, thus making them unsuitable for streaming data. To address this issue, recent HUIM algorithms have adopted a one-phase approach using the utility-list structure. Liu et al. Liu and Qu [31] first introduced and utilized this structure in the HUI-Miner algorithm. The utility-list structure can be used to mine high utility itemsets in a single phase, i.e., without maintaining candidates in memory. The utility of itemsets can be directly calculated using their utility-lists without scanning the database again. The simplicity of the utility-list structure has led to the development of numerous utility-list-based algorithms [31–33], which generally outperform other algorithms. To discover high-utility patterns in data streams, some algorithms have been proposed [10,34,35]. These studies generally extend traditional HUIM methods to increase their efficiency in a streaming context. Nevertheless, they are not designed to detect changes or drifts.

As mentioned earlier, customer transactions in retail stores can be seen as a stream of data, because customers continuously purchase products in the stores. This also means that the data is not static and is often impossible to store in memory due to its large volume. Moreover, in a streaming context, the distribution of data and the transactional behavior of customers can change and evolve over time. To the best of our knowledge, no work has been proposed to detect drifts for high utility itemset mining in streams of quantitative transactions, while considering the importance of utility over time. On the other hand, in traditional frequent itemset mining, several algorithms have been proposed to identify concept drifts in data streams [36,37]. Ng and Dash [36] proposed a test paradigm named Algorithm for change detection (ACD) for detecting changes in transactional data streams by considering the support of itemsets for reservoir sampling. ACD evaluates drifts by performing reservoir sampling and applying three statistical tests. The ACD method employs a bound based on Hoeffding's Inequality to determine the number of transactions to be kept in each