SPOT: A System for Detecting Projected Outliers From High-dimensional Data Streams

Ji Zhang #1, Qigang Gao #2, Hai Wang *3

# *Faculty of Computer Science*
Dalhousie University
Halifax, Nova Scotia, Canada
1 jiz@cs.dal.ca
2 qggao@cs.dal.ca

* *Sobey School of Business*
Saint Mary's University
Halifax, Nova Scotia, Canada
3 hwang@smu.ca

***Abstract*— In this paper, we present a new technique, called Stream Projected Ouliter deTector (SPOT), to deal with outlier detection problem in high-dimensional data streams. SPOT is unique in a number of aspects. First, SPOT employs a novel window-based time model and decaying cell summaries to capture statistics from the data stream. Second, Sparse Subspace Template (SST), a set of top sparse subspaces obtained by unsupervised and/or supervised learning processes, is constructed in SPOT to detect projected outliers effectively. Multi-Objective Genetic Algorithm (MOGA) is employed as an effective search method in unsupervised learning for finding outlying subspaces from training data. Finally, SST is able to carry out online self-evolution to cope with dynamics of data streams. This paper provides details on the motivation and technical challenges of detecting outliers from high-dimensional data streams, present an overview of SPOT, and give the plans for system demonstration of SPOT.**

I. Introduction

Outlier detection is an important research problem in data mining that aims to find objects that are considerably dissimilar, exceptional and inconsistent with respect to the majority data in an input database [4]. In recent years, we have witnessed a tremendous research interest sparked by the explosion of data collected and transferred in the format of streams. The research advancements in outlier detection in data streams can contribute to a wide range of applications in analysis and monitoring of network traffic data, web log, sensor networks and financial transactions, etc. In these applications, we may deal with data streams that contain dozens of, even hundred of, attributes, thus a technique for supporting outlier detection in high-dimensional data streams are desired to develop.

We observe that the overwhelmingly majority of outliers existing in high-dimensional data streams are embedded in relatively low-dimensional subspaces (spaces consisting of a subset of attributes). These outliers are termed *projected outliers* in the high-dimensional space context. This is because that, as dimensionality of data goes up, data tend to be equally distant from each other. As a result, the difference of data points' outlier-ness will become increasingly weak and thus undistinguishable. Only in moderate or low dimensional subspaces can significant outlier-ness of data be observed.

Formally, the problem of detecting projected outliers from high-dimensional data streams can be formulated as follows: given a data stream $\mathcal{D}$ with a potentially unbounded size of φ-dimensional data points, each data point $p_i = \{p_{i1}, p_{i2}, \ldots, p_{i\varphi}\}$ in $\mathcal{D}$ will be labeled as either a projected outlier or a regular data point. If p_i is a projected outlier, its associated outlying subspace(s) will be given as well. The results to be returned will be a set of projected outliers and their associated outlying subspace(s) to indicate the context where these projected outliers exist.

Technically speaking, detecting projected outliers in high-dimensional data streams is a nontrivial problem. The challenges mainly come from two aspects. First, finding outlying subspace of the data is a NP problem, and the exhaustively search (brute force) of the space lattice is rather computationally demanding and totally infeasible when the dimensionality of data is high. Another aspect of the challenge originates from the characteristics of streaming data themselves. Outlier detection algorithms are constrained to take only one pass to process the streaming data with the conditions of space limitation and time criticality.

Recently, there are some emerging work in dealing with outlier detection either in high-dimensional data or data streams. However, there have not been any reported research work so far for exploring the intersection of these two research fields. For those methods in projected outlier detection in high-dimensional space [1][3][9][6][7][8], their measurements used for evaluating points' outlier-ness are not incrementally updatable and many of the methods involve multiple scans of data, making them incapable of handling fast data streams. For instance, [3][9] use the Spasity Coefficient to measure data sparsity. Spasity Coefficient is based on an equi-depth data partition that has to be updated frequently in data stream. This will be expensive and such updates will require multiple scan of data. [6][7][8] use data sparsity metrics that are based on distance involving the concept of kNN. This is not suitable

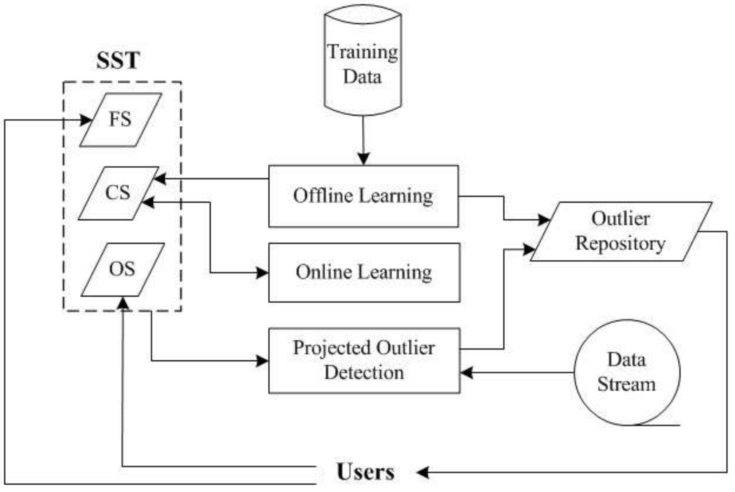

Fig. 1. An overview of SPOT

for data streams as one scan of data is not sufficient for retaining kNN information of data points. The techniques for tackling outlier detection in data streams [2][5] rely on full data space to detect outliers and thus projected outliers cannot be discovered by these techniques.

II. An Overview of SPOT

In this section, an overview of SPOT will be presented. We will focus on discussing the time model, data synapses, learning processes employed in SPOT. The system architecture diagram of SPOT is given in Figure 1.

A. Time Model

We use a novel window-based time model, called (ω, ϵ)-model, in SPOT for discriminating data arriving at different times in the stream. Each data in the window will be assigned a weight, indicating its importance or influence to the data synapses at the current time. The basic idea of (ω, ϵ)-model is that, for the points that has slided out of the slide window with a size of ω, the sum of their weights will not exceed ϵ. (ω, ϵ)-model is an approximation of conventional window-based model of a window size ω with an approximation factor of ϵ. Unlike the conventional window-based model, (ω, ϵ)-model does not need to keep trace of detailed data in the window. Moreover, instead of maintaining a large number of historical snapshots of data synapses as in the tilted time models, only the latest snapshot needs to be kept in (ω, ϵ)-model.

B. Data Synapses

In SPOT, we employ *Base Cell Summary* (BCS) and *Projected Cell Summary* (PCS), two compact structures that are able to capture the major underlying characteristics of the data stream for detecting outliers. Quantitization of BCS and PCS entails an equi-width partition of domain space. A *base cell* is a cell in hypercube with the finest granularity with dimensions of φ, while a *projected cell* is a cell that exists in a particular subspace s.

Definition 1. *Base Cell Summary* (BCS): The Base Cell summary of a base cell c in the hypercube is a triplet defined as $BCS(c) = \{D_c, \overrightarrow{LS}_c, \overrightarrow{SS}_c)$, where D_c, $\overrightarrow{LS}_c$ and $\overrightarrow{SS}_c$ are the number of points in c, the sum and squared sum of each single dimension of points in c, respectively, i.e. $\overrightarrow{LS}= \sum \overrightarrow{p}_i$ and $\overrightarrow{SS}= \sum \overrightarrow{p}_i^{\,2}$, for p_i located in c, $1 \leq i \leq \varphi$, n_c is a scalar while both $\overrightarrow{LS}_c$ and $\overrightarrow{SS}_c$ are φ-dimensional vectors.

Definition 2. *Projected Cell Summary* (PCS): The Projected Cell Summary of a cell c in a subspace s is a pair of scalars defined as $PCS(c, s) = (RD, IRSD)$, where RD and $IRSD$ are the Relative Density and Inverse Relative Standard Deviation of data points in c of s.

The advantages of BCS and PCS lie in that they can both be computed and maintained incrementally thanks to their additive and incremental properties, which enables SPOT to deal with fast data streams.

C. Stages of SPOT

SPOT can be divided into two stages: learning and detection stages. In learning stage, Sparse Subspace Template (SST) is constructed by supervised and/or unsupervised learning process. SST casts light on where projected outliers are likely to be found in the high-dimensional space. Based upon SST, SPOT screens projected outliers from constantly arriving data in the detection stage.

1) Learning Stage: Since the number of subspaces grows exponentially with regard to dimensions of the data set, evaluating each streaming data point in each possible subspace thus becomes prohibitively expensive. As such, we only alternatively check each point in a few subspaces, called *Sparse Subspace Template* (SST), in the space lattice in an effort to render this problem tractable. SST consists of a few subsets of subspaces generated by different underlying rationales. These subspace subsets supplement each other in terms of towards capturing the right subspaces where projected outliers are hidden. This helps enable SPOT to detect projected outliers more effectively. Specifically, SST contains the following three subspace subsets, *Fixed SST Subspaces (FS), Clustering-based SST Subspaces (CS)* and *Outlier-driven SST Subspaces (OS)*, respectively.

- **Fixed SST Subspaces (FS)**

Fixed SST Subspaces (FS) contains all the subspaces in the full lattice whose maximum dimension is $MaxDimention$, where $MaxDimention$ is a user-specified parameter. In other words, FS contains all the subspaces with dimensions of 1, 2, $\cdots$, $MaxDimention$.

- **Clustering-based SST Subspaces (CS)**

Clustering-based SST Subspaces (CS) consists of the sparse subspaces of the top training data that have the highest overall outlying degree. The selected training data are more likely to be considered as outliers that can be potentially used to detect more subsequent outliers in the stream. The overall outlying degree of training data is computed by employing clustering method.

- **Outlier-driven SST Subspaces (OS)**

A few outlier examples may be provided by domain experts. MOGA is applied on each of these outliers to find their top sparse subspaces. There subspaces are called Outlier-driven SST Subspaces (OS). Based on OS, exampled-based outlier detection can be performed that effectively detects more outliers that are similar to these outlier examples.

SST is obtained by offline learning processes using a batch of training data. A salient feature of SPOT is that it provides flexibility to allow for both *unsupervised* and *supervised* learning. Since the construction of FS does not require any learning process, thus the major task of learning stage is to generate CS and OS.

Unsupervised learning

In unsupervised learning, SPOT takes in unlabeled training data from the data stream and *automatically* find the set of subspaces in which data exhibit a high level of overall sparsity, indicated by PCS with small RD and IRSD. Intuitively, these subspaces are where projected outliers are likely to exist. We assume that a set of historical data is available for unsupervised learning at the beginning of SPOT. The training dataset should fit into main memory for minimizing possible I/O overhead. Multi-objective Genetic Algorithm (MOGA) is employed to search space lattice to find those top subspaces in which training data exhibit highest sparsity. The general steps of unsupervised learning are as follows. First, we perform MOGA on the whole training data to find their top sparse subspaces. Then, we cluster training data using lead clustering method under different data order based upon. Finally, the outlying degree of all the training data are computed and MOGA is applied again on the top training data to find their top sparse subspaces, which will become the CS of SST.

Supervised Learning

Supervised learning in SPOT attempts to incorporate the prior domain knowledge, if any, into SPOT to assist SST construction. The knowledge that will help in the learning includes, but not necessarily restricted to, the relevancy of attributes with respect to the outlier detection task under study and the identified/labeled projected outliers provided by domain experts, as did in [9]. The former can contribute to removal of irrelevant attributes to speed up the learning process, while the later will be applied MOGA whose top sparse subspaces will become OS.

2) Detection Stage: The detection stage performs outlier detection for incoming stream data. As streaming data arrive continuously, data synapses (BCS and PCS) are first updated dynamically in order to capture new information of arrived data. Then, we retrieve PCS of the projected cell to which each data belongs in subspace of SST, and label the point as a projected outlier if PCS of the cell it belongs to in one or more subspaces fall under certain pre-specified thresholds. Due to the speed of data streams and time criticality posed to the detection process, it is crucial that the abovementioned steps can be performed quickly. BCS and PCS can be updated incrementally and thus will be very quickly. Also, the outlier-ness check of each data in the stream is also very efficient. It only involves mapping the data point into an appropriate cell and retrieving PCS of this cell for outlier-ness checking.

In addition to performing fast online detection, SPOT is also equipped with the ability to cope with dynamics of data streams and respond to the possible concept drift, which are presented as follows.

First, both BCSs of populated base cells and PCSs of populated projected cells in subspaces of SST will be efficiently maintained as data flow in. This ensures SST to be able to capture the latest data characteristics of the stream and response quickly to data changes;

Second, any outliers detected will be stored. Their top sparse subspaces produced by MOGA will be added into OS of SST to detect outliers from streaming data arriving later. As a consequence, the detecting ability of SST will be enhanced constantly as an increasing number of outliers are detected along the detection process;

Finally, CS in SST is equipped with an unique ability of *online self-evolution* (referred to as the online learning in Figure 1). The basic idea of self-evolution of CS in SST is that, as the detection stage proceeds, a number of new subspaces are periodically generated online by crossovering and mutating the top subspaces in the current CS. These newly generated subspaces represent the new evolution of CS. Once the new subspaces join SST, the whole CS, including the new subspaces, will be re-ranked and the new top sparse subspaces will be chosen to create a new CS.

III. Innovative Features and Contributions of SPOT

Based on the descriptions of SPOT presented in section 2, we can now summarize the innovative features and contributions of SPOT as follows:

- In SPOT, we employ a new window-based time model and decaying cell summaries to capture statistics from the data streams for outlier detection. The time model is able to approximate the conventional window-based model without maintaining the detailed data in the window or keeping multiple snapshots of data synapses. The decaying cell summaries can be efficiently computed and incrementally maintained, enabling SPOT to handle fast data streams;
- SPOT constructs Sparse Subspace Template (SST) to detect projected outliers. SST consists of a number of mutually supplemented subspace subsets that contribute collectively to an effective detection of projected outliers. SPOT is able to perform supervised and/or unsupervised learning to construct SST, providing a maximum level of flexibility to users. A number of strategies, such as self-evolution of SST and concept drift detection, have also been incorporated into SPOT to greatly enhance its adaptability to dynamics of data streams;
- Unlike most of other outlier detection methods that measure outlier-ness of data points based on a single criterion, SPOT adopts a more flexible framework of using multiple measurements for this purpose. SPOT utilizes the Multi-Objective Genetic algorithm (MOGA) as an effective search method to find subspaces that are able to optimize all the criteria;
- Last but not the least, we show that SPOT outperforms the existing method in terms of efficiency and effectiveness through experiments on both synthetic and real-life streaming data sets.

IV. Main Features of Demonstration

Our demonstration of SPOT will consist of the following four parts:

- First, we introduce the problem of projected outlier detection in high-dimensional data streams. To better illustrate the motivation of this research, we pictorially show the distribution of outliers and the existence of projected outliers in high-dimensional data streams. We also show to the audience some real-life applications to which SPOT can be potentially applied. These examples provide the audience with insight into the interesting notion of projected outliers in high-dimensional space context, the valuable abnormality patterns that can be explored from them and finally the inherent technical challenges associated with this problem.
- Second, we showcase the system architecture of SPOT. Emphasis of architecture demonstration will be the learning and detection process of SPOT. The major modules in SPOT such as *Offline unsupervised/supervised learning*, *online detection*, *Multi-Objective Genetic Algorithm (MOGA)*, and *concept drift detection*, etc, will be shown to audience by means of the System Diagram.
- Third, by using synthetic and real-life data sets, we illustrate to the audience the experimental evaluation of SPOT and the comparative study between SPOT and the latest stream outlier/anomaly detection method, in terms of efficiency and effectiveness under a wide spectrum of settings.
- Finally, we showcase the prototype of SPOT and the audience will be encouraged the play the demo interactively themselves. We will provide on-site assistance to the audience to use the prototype upon request.

Acknowledgment

The research and development of SPOT are supported in part by grant of Natural Sciences and Engineering Research Council of Canada (Grant #:312423) and Killam Foundation.

References

[1] C. C. Aggarwal and P. S. Yu. An effective and efficient algorithm for high-dimensional outlier detection. *VLDB Journal*, 14: 211-221, 2005.

[2] C. C. Aggarwal. On Abnormality Detection in Spuriously Populated Data Streams. *SDM'05*, Newport Beach, CA, 2005.

[3] C. C. Aggarwal and P. S. Yu. Outlier Detection in High Dimensional Data. *SIGMOD'01*, 37-46, Santa Barbara, California, USA, 2001.

[4] J. Han and M Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufman Publishers, 2000.

[5] T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos. Distributed deviation detection in sensor networks. *SIGMOD Record* 32(4): 77-82, 2003.

[6] J. Zhang, M. Lou, T. W. Ling and H. Wang. HOS-Miner: A System for Detecting Outlying Subspaces of High-dimensional Data. *VLDB'04*, 1265-1268, Toronto, Canada, 2004.

[7] J. Zhang, Q. Gao and H. Wang. A Novel Method for Detecting Outlying Subspaces in High-dimensional Databases Using Genetic Algorithm. *ICDM'06* 731-740, Hong Kong, China, 2006.

[8] J. Zhang and H. Wang. 2006. Detecting Outlying Subspaces for High-dimensional Data: the New Task, Algorithms and Performance. *Knowledge and Information Systems (KAIS)*, 33-355, 2006.

[9] C. Zhu, H. Kitagawa and C. Faloutsos. Example-Based Robust Outlier Detection in High Dimensional Datasets. *ICDM'05*, 829-832, 2005.