Volume 4, Issue 11, November 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Outlier Detection Using Spot

Madhav Bokare
Research Scholar
Priyadarshini Institute of Engineering & Technology,
Nagpur, India

V.M Thakare
Professor
P.G. Dept of Computer Science,
Sant Gadge Baba Amravati university, Amravati, India

Abstract— In today's cyber age there lot of data is generated and used in for multiple purposes in routine IT environment. Data mining deals with extracting meaningful information, finding interesting patterns that are hidden in the collected data. The question main arises in KDD (Knowledge Discovery Databases) which data gets more priority for knowledge extraction procedure. Data mining techniques are used fundamentally used for analysis purpose in today's market sector, Artificial Intelligence. For that purpose the term 'Outlier' was propose. This is mainly used for removal of noisy data (irrelevant, unnecessary data)but at the same end we cannot remove completely from the process. If we plotted such points on dimensional manner than a set of such points can be said as 'Outlier' .In this we will discuss all the SPOT mechanism of effective outlier detection.

Keywords— SST, SPOT, Outliers, MOGA, BCS,PCS.

I. INTRODUCTION

A Data mining, as a powerful knowledge discovery tool, aims at modeling relationships and discovering hidden patterns in large databases [1]. Among four typical data mining tasks, *outlier detection* is the closest to the initial motivation behind data mining than predictive modeling, cluster analysis and association analysis [2]. Outlier detection is a critical task in many safety critical environments as the outlier indicates abnormal running conditions from which significant performance degradation may well result.Outlier mining in d-dimensional point sets is a fundamental and well-studied data mining task due to its variety of applications. We can detect an outlier human with its behavior data as per situation in that human's life. As per today's literature view there is not at all an approximate definition of the term 'Outlier'. The two classical definition are as follows Hawkins[3] says "An outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" Another Barnett and Lewis[4] say's "an observation (or subset of observations) which appears to be inconsistent with the
Remainder of that set of data".

To make effective detection we need a good technique which will give accurate detection in it. In most of methods of outlier detection attributes in database plays a vital role in which that may be in numerical or in a different format. Another problem which can be found in outlier detection is mining of outlier is its size of datasets. This turn into a big problem when carrying outlier detection in large, parallel machine.Most existing outlier detection methods only deal with static data with relatively low dimensionality. Recently, outlier detection for high-dimensional stream data became a new emerging research problem. A key observation that motivates this research is that outliers in high-dimensional data are projected outliers, i.e., they are embedded in lower- dimensional subspace. There are also some demerits of outlier detection in large data sets.Detecting projected outliers from high-dimensional stream data is a very challenging task for several reasons. First, detecting projected outliers is difficult even for high-dimensional static data.. Second, the algorithms for handling data streams are constrained to take only one pass to process the streaming data with the conditions of space limitation and time criticality. The currently existing methods for outlier detection are found to be ineffective for detecting projected outliers in high-dimensional data streams.

II. BASIC TYPES OF OUTLIERS

First, outliers can be classified as point outliers and collective outliers based on the number of data instances involved in the concept of outliers.

A. Point outlier

In a given set of data instances, an individual outlying instance is termed as a point outlier. This is the simplest type of outliers and is the focus of majority of existing outlier detection schemes [3]. A data point is detected as a point outlier because it displays outlier-ness at its own right, rather than together with other data points. In most cases, data are represented in vectors as in the relational databases. Each tuple contains a specific number of attributes. The principled method for detecting point outliers from vector-type data sets is to quantify, through some outlier-ness metrics, the extent to which each single data is deviated from the other data in the data set[5].

B. Collective outliers

A collective outlier represents a collection of data instances that is outlying with respect to the entire data set. The individual data instance in a collective outlier may not be outlier by itself, but the joint occurrence as a collection is anomalous [5]. Usually, the data instances in a collective outlier are related to each other. Typical types of collective outliers are sequence outliers, where the data are in the format of an ordered sequence. Outliers can also be categorized into vector outliers, sequence outliers, trajectory outliers and graph outliers, etc, depending on the types of data from where outliers can be detected.

C. Vector outliers

Vector outliers are detected from vector-like representation of data such as the relational databases. The data are presented in tuples and each tuple has a set of associated attributes. The data set can contain only numeric attributes, or categorical attributes or both. Based on the number of attributes, the data set can be broadly classified as low-dimensional data and high-dimensional data, even though there is not a clear cutoff between these two types of data sets. As relational databases still represent the mainstream approaches for data storage, therefore, vector outliers are the most common type of outliers we are dealing with.

D. Sequence outliers

In many applications, data are presented as a sequence. A good example of a sequence database is the computer system call log where the computer commands executed, in a certain order, are stored. A sequenceof commands in this log may look like the following sequence: http-web, buffer-overflow, http-web, http-web, smtp-mail, ftp, http-web, ssh. Outlying sequence of commands may indicate a malicious behavior that potentially compromises system security. In order to detect abnormal command sequences, normal command sequences are maintained and those sequences that do not match any normal sequences are labeled sequence outliers. Sequence outliers are a form of collective outlier.

E. Trajectory outliers

Recent improvements in satellites and tracking facilities have made it possible to collect a huge amount of trajectory data of moving objects. Examples include vehicle positioning data, hurricane tracking data, and animal movement data [6]. Unlike a vector or a sequence, a trajectory is typically represented by a set of key features for its movement, including the coordinates of the starting and ending points; the average, minimum, and maximum values of the directional vector; and the average, minimum, and maximum velocities. Based on this representation, a weighted sum distance function can be defined to compute the difference of trajectory based on the key features for the trajectory [7]. A more recent work proposed a partition and detect framework for detecting trajectory outliers [6]. The idea of this method is that it partitions the whole trajectory into line segments and tries to detect outlying line segments, rather than the whole trajectory. Trajectory outliers can be point outliers if we consider each single trajectory as the basic data unit in the outlier detection. However, if the moving objects in the trajectory are considered, then an abnormal sequence of such moving objects (constituting the sub-trajectory) is a collective outlier.

F. Graph outliers

Graph outliers represent those graph entities that are abnormal when compared with their peers. The graph entities that can become outliers include nodes, edges and sub-graphs. For example, Sun et al. investigate the detection of anomalous nodes in a bipartite graph [8][9]. Auto part detects outlier edges in a general graph [10]. Noble et al. study anomaly detection on a general graph with labeled nodes and try to identify abnormal substructure in the graph [11]. Graph outliers can be either point outliers (e.g., node and edge outliers) or collective outliers (e.g., sub-graph outliers).

III. METHODOLOGY OF SPOT (STREAM PROJECTED OUTLIER DETECTOR)

Our technique for outlier detection in data streams, SPOT, can be broadly divided into two stages: the learning and detection stages. SPOT can further support two types of learning, namely offline learning and online learning. In the offline learning, Sparse Subspace Template (SST) is constructed using either the unlabeled training data (e.g., some available historic data) and/or the labeled outlier examples provided by domain experts. SST is a set of subspaces that features higher data sparsely /outlier-ness than other subspaces. SST consists of three groups of subspaces, i.e., Fixed SST Subspaces (FS), Unsupervised SST Subspaces (US) and Supervised SST Subspaces (SS), where FS is a compulsory component of SST while US and SS are optional components. SST casts light on where projected outliers are likely to be found in the high-dimensional space. SST is mainly constructed in an unsupervised manner where no labeled examples are required. However, it is possible to use the labeled outlier exemplars to further improve SST. As such, SPOT is very flexible and is able to cater for different practical applications that may or may not have available labeled exemplars. When SST is constructed, SPOT can start to screen projected outliers from constantly arriving data in the detection stage. The incoming data will be first used to update the data summaries (i.e., the PCSs) of the cell it belongs to in each subspace of SST.

This data will then be labeled as an outlier if the PCS values of the cell where it belongs to are lower than some pre-specified thresholds. The detected outliers are archived in the so-called Outlier Repository. Finally, all or only a specified number of the top outliers in Outlier Repository will be returned to users when the detection stage is finished. During the detection stage, SPOT can perform online training periodically. The online training involves updating SST with new

sparse subspaces SPOT finds based on the current data characteristics and the newly detected outliers. Online training improves SPOT's adaptability to dynamic of data streams.

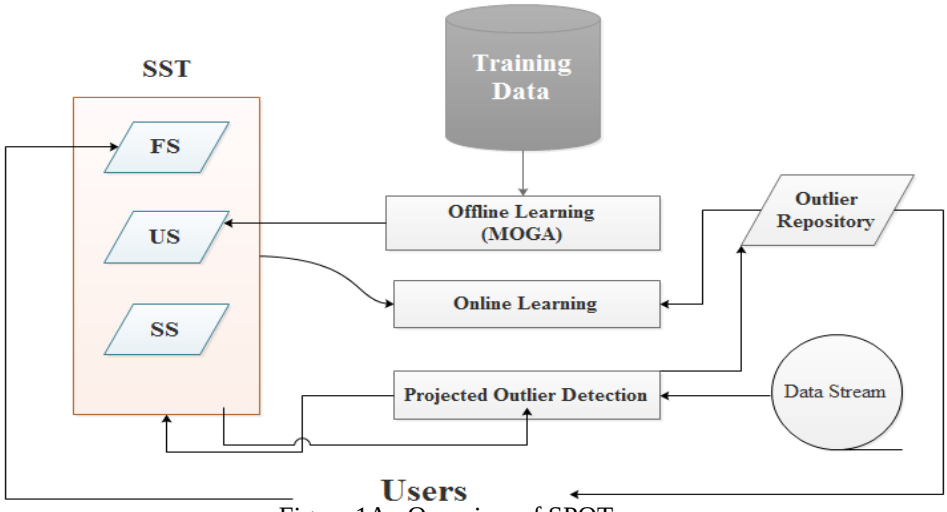

Figure 1An Overview of SPOT

The Learning Stage of SPOT since the number of subspaces grows exponentially with regard to the dimensionality of data streams, evaluating each streaming data point in each possible subspace becomes prohibitively expensive. As such, we only check each point in a few subspaces in the space lattice alternatively, in an effort to render projected outlier detection problem tractable. In SPOT, we evaluate each data point from the stream in the subspaces contained in the SST. We note that detecting outliers is more challenging and difficult than finding all their outlying subspaces. Once an outlier has been flagged in one or more subspaces, it is a fairly trivial task to find its other outlying subspaces by applying some appropriate search method, such as Multi-objective Genetic Algorithm (MOGA). The flagging of outliers is performed online while the search for the outliers' outlying subspaces can be done in an offline manner. Therefore, there is no need to require SST to contain all the outlying subspaces for any given projected outlier. The problem now becomes how to enable SST to contain one or more outlying subspaces for as many projected outliers in the streams as possible. In SPOT, SST consists of a few groups of subspaces that are generated by different underlying rationales. Different subspace groups supplement each other 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 groups, Fixed SST Subspaces (FS), Unsupervised SST Subspaces (US) and Supervised SST Subspaces (SS), respectively. Since the construction of FS does not require any learning process, the major task of the offline learning stage is to generate US and/or SS. SST is obtained through the offline learning process using a batch of training data. SPOT is mainly designed as unsupervised outlier detection method (by means of FS and US). However, a salient feature of SPOT is that it is not only able to deal with unlabeled data but also provides facility for learning from labeled outlier exemplars through SS.

The detection stage performs outlier detection for arriving stream data. As streaming data arrive continuously, the data synopses the PCS of the projected cell where the streaming data belongs to in each subspace of SST are first updated in order to capture new information of the arrived data. A hash function is employed here to quickly map a data into the cell it is located in any subspace. Then, the data is labeled as a projected outlier if the PCS of the cell it belongs to in one or more SST subspaces falls under certain pre-specified thresholds. These subspaces are the outlying subspaces of this outlier. All the outliers, together with their outlying subspaces and the PCS of the cell they belong to in these outlying subspaces, are output to the so-called Outlier Repository. All or a specified number of the top outliers in Outlier Repository are returned to the users in the end. Due to the speed of data streams and time criticality posed to the detection process, it is crucial that the aforementioned steps can be performed quickly. As we have shown earlier, BCS and the PCS can be updated incrementally and thus will be performed quickly. Also, the outlier-ness evaluation of each data in the stream is also efficient. It only involves mapping the data point into an appropriate cell and retrieving the PCS of this cell for outlier-ness checking. An essential issue to the effectiveness of SPOT is how to cope with dynamics of data streams and respond to possible concept drift.

IV. CONCLUSIONS

To solve the problem of projected outlier detection for multi and high-dimensional data streams, we present a new technique, called Stream Projected Outlier Detector (SPOT).SPOT utilizes compact data synopsis, including BCS and the PCS, to capture necessary data statistical information for outlier detection. Both of them can be computed and maintained efficiently, enabling SPOT to meet the one-pass constraint and time criticality posed by data stream applications. Another major feature of SPOT lies in the outlying search strategy it uses to construct SST, particularly US and SS. Unlike most of other outlier detection methods that measure outlierness of data points based on a single criterion, SPOT adopts a more flexible framework for using multiple measures for outlier detection. SPOT is designed to detect point outliers from vector-type data sets. It is only able to deal with well-structured data sets such as relational databases and data streams. It cannot be applied to other unstructured or semi-structured data sets such as TCP dump data or XML data.

ACKNOWLEDGMENT

We wish to thank Dr. S. B. Thorat, Director, ITM, Nanded. Dr.Pradeep Bute ,Professor, Nagpur. One of the author MadhavBokare Special Thanks to the honourable members of Shri Shardha Bhavan Education Society,Nanded and Mr. Pawale Satish R, Assistant Professor, CS dept, ITM, Nanded, Maharashtra, India.

REFERENCES

[1] Pang-Ninh Tan (2006) Knowledge Discovery from Sensor Data. Sensors
[2] Pei Sun (2006) Outlier detection in high dimensional, spatial and sequential data sets.
[3] D.M. Hawkins (1980) Identication of outliers. Chapman and Hall, Reading, London
[4] V. Barnett and T. Lewis (1994) Outliers in statistical data. John Wiley Sons, Reading, New York
[5] V. Chandola, A. Banerjee, and V. Kumar. Outlier Detection-A Survey,Technical Report, TR 07-017, Department of Computer Science and Engineering, University of Minnesota, 2007
[6] J. Lee, J. Han and X. Li. Trajectory Outlier Detection: A Partition-and-Detect Framework.ICDE'08, 140-149, 2008.
[7] E. M. Knorr, R. T. Ng and V. Tucakov. Distance-Based Outliers: Algorithms and Applications.VLDB Journal, 8(3-4): 237-253, 2000.
[8] J. Sun, H. Qu, D. Chakrabarti and C. Faloutsos. Neighborhood Formation and Anomaly Detection in Bipartite Graphs.ICDM'05, 418-425, 2005.[108]
[9] J. Sun, H. Qu, D. Chakrabarti and C. Faloutsos. Relevance search and anomaly detection in bipartite graphs. SIGKDD Explorations7(2): 48-55, 2005
[10] D. Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. InPKDD'04, pages 112-124, 2004
[11] C. C. Noble and D. J. Cook.Graph-based anomaly detection. In KDD'03, pages 631-636, 2003.
[12] Mahito Sugiyama, Karsten M. Borgwardt,Anomaly Detection on Data Streams with High Dimensional Data Environment, IJIRCCE,Vol.2, Special Issue 1, March 2014.

