Research issues in Outlier mining: High-Dimensional Stream Data

Vijay Kumar Janga
Research Scholar (JNTU-HyD) & Asst.Prof
Balaji Institute of Engineering & Sciences, Narsampet, Warangal (A.P)-506132

Ajay Kumar Mamindla
Asst.Prof
Balaji Institute of Engineering & Sciences, Narsampet, Warangal (A.P)-506132
Narsampet, Warangal (A.P)-506132

Abstract - Outlier detection is an important research problem in data mining that aims to discover useful abnormal and irregular patterns in large data sets. There exist emerging applications of data streams that require Outlier mining, such as network traffic monitoring, web click analysis, weather forecasting and geo sensor data .Different from data in traditional static databases, data streams are huge in nature and arrive continuously with changing data distribution. This raises new issues that need to be considered when developing Outlier mining techniques for stream data. This paper discusses those issues and how they are addressed in the existing literature.

Key words: Stream data, Multi-dimensional, Kernal estimators.

I. INTRODUCTION

A data stream is an ordered sequence of items that arrives in timely order. Different from data in traditional static databases, data streams are continuous, unbounded, usually come with high speed and have a data distribution that often changes with time. One example application of data stream outlier mining is to estimate outlier in sensor networks [Subramaniam, T.Palpanas, 2006].

There exists numerous set of outlier mining mechanisms for traditional static data they include distribution-based, clustering, distance-based, or density-based. The distribution-based approach [2,3] assumes the data following a distribution model (e.g. Normal) and flags as outliers those objects which deviate from the model. Thus, such approaches do not work well in moderately high dimensional (multivariate) spaces, and have difficult to find a right model to fit the evolving data stream.

To overcome these limitations, researchers have turned to various non-parametric approaches (clustering, distance-based, and density-based). Most clustering algorithms (e.g. CLARANS [4], DBSCAN [5], BIRCH [6], WaveCluster [7], CLIQUE [8]), are to some extent capable of handling exceptions. However, since the main objective is to find cluster, they are developedto optimize clustering, and regard outliers as "by-products".

Distance based approaches [9,10,11,12,13,14,15], first proposed by E.M. Knorr and R.T. Ng [9], attempt to overcome limitations of distribution-based approach and they detect outliers by computing distances among points. A point p in a data set T is a distance-based outlier (DB (ρ, r)-outlier) if at most a fraction ρ of the points in T lie within distance r from p. It has an intuitive explanation that an outlier is an observation that is sufficiently far from most other observations in the data set. However, it will be no effect when the data points exhibit different densities in different regions of the data space or across time, because this outlier definition is based on a single, global criterion determined by the parameters ρ and r, so more robust density-based techniques were provided.

Density-based approaches proposed originally by M. Breunig, et al. [16] which defines a local outlier factor (LOF) for each point depending on the local density of its neighborhood. In general, points with a high LOF are flagged as outliers. It has attracted considerable attention [17,18,19], and a large number of algorithms have been developed which concern how to define the density or accelerate theefficiency, such as, LOCI [17] method. In the distributed data stream environments, Babcock and Olston presented an original algorithm for distributed top-k monitoring [20]. AmitManjhi, etal. [21] thought of finding recently frequent items. S. Subramaniam, et al. [22] cared about the outlier over sensor stream data. Graham Cormode, et al. [23] considered continuous clustering.

II. STREAM DATA MODELS

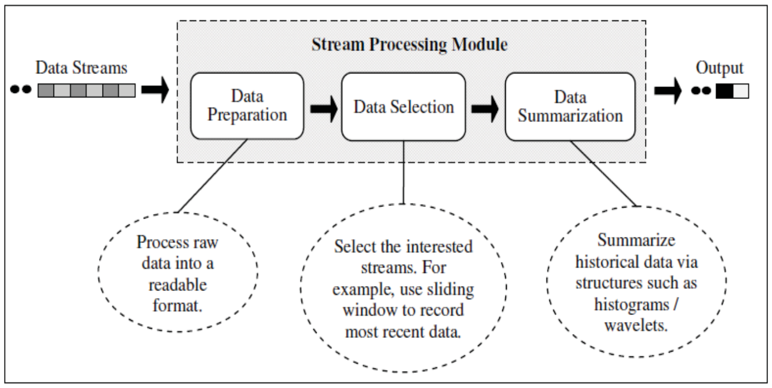
2.1 Stream Data Processing Architecture (SDPA)

SDPA works in three phases

1. Data Preparation
2. Data Selection
3. Data Summarization

2.1.1. Data Preparation

Data preparation includes data formatting and cleaning. Incoming data arrive in raw formats in which many cases cannot be efficiently handled by the DSMS. Hence, an efficient method must be used to transform the data into a format that can be rapidly processed by the remaining subsystems.
Data cleaning is necessary to reduce errors and computational overhead of the results. A typical data cleaning task would be to remove invalid data items (e.g., remove negative values when non-negative values are expected) using a simple filtering based method.


Stream Processing Module
Data Streams
Data Preparation
Data Selection
Data Summarization
Output
Process raw data into a readable format.
Select the interested streams. For example, use sliding window to record most recent data.
Summarize historical data via structures such as histograms / wavelets.

SDPA Architecture

2.1.2. Data Selection

The primary goal of data selection is to reduce the size of the total data. Previous works have proposed the following three main methods to address this issue:
i) Sliding window– It's a window of size *L* (can be dynamic) to store *L* most recent data elements. The motivation behind this data structure is the assumption that recent data carry greater significance over old data. There are several applications which can fulfill the assumptions such as monitoring and surveillance systems. Extensions of this sliding window structure have been proposed to store data at varying time granularity and length.

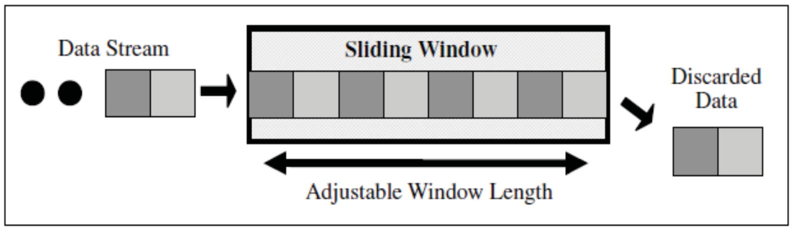


Diagram showing sliding window model

ii) Sampling– It's a process of assigning a probabilistic value to each data element under selection. The advantages of the sampling technique are that it can dramatically reduce the data size and provide a probabilistic error bound (Hoeffding bound [27]) for a given sampling rate. The disadvantage is that potentially important data items (e.g., outliers) may be missed and hence sampling may not be directly applicable to outlier detection tasks.
iii) Load shedding– It utilizes a similar scheme to sampling except that certain sequence sets of data elements are discarded. This approach has demonstrated its applicability in stream environment where the incoming data rate leads to bursts. Load shedding will selectively drop high rate streams and divert resources to higher priority tasks [28]. Load shedding shares the same drawback as sampling as it may remove some important anomalous data elements.

Kernel Estimators: The simplest statistical estimator for estimating the probability density function is random sampling. The kernel estimator [40, 8] is a generalized form of sampling, whose basic step is to produce a uniform random sample. As in random sampling, each sample point has
a weight of one. In kernel estimation however, each point distributes its weight in the space around it. A kernel function describes the form of this weight distribution, generally distributing most of the weight in the area near the point. Summing up all the kernel functions we obtain a density function
for the dataset.
More formally, assume that we have a static relation, T, that stores the d-dimensional values t,
T = (t_1. . . t_d), whose distribution we want to approximate. The recorded values must fall in the interval $[0, 1]^d$. This requirement is not restrictive, since we can map the domain of the input values
to the interval $[0, 1]^d$. Let R be a random sample of T, and k(x) a d-dimensional function of x = R (x_1, . . . ,x_d), such that ∫ [0,1] k(x)dx = 1, for all tuples in R. We call k(x) the kernel function. We can now approximate the underlying distribution f(x), according to which the values in T were generated,
using the following function

$$\boxed{f(\mathbf{x}) = \frac{1}{|T|}\sum_{\mathbf{t_i}\in R} k(x_1 - t_{i_1}, \ldots, x_d - t_{i_d}).}$$

The choice of the kernel function is not significant for the results of the approximation [40]. Hence, we choose the Epanechnikov kernel that is easy to integrate:

$$\boxed{k(\mathbf{x}) = \begin{cases} \left(\frac{3}{4}\right)^d \frac{1}{B_1 \ldots B_d} \prod_{1\le i\le d}\left(1 - \left(\frac{x_i}{B_i}\right)^2\right) \\ \qquad\qquad\qquad , \text{if } \forall i,\ 1 \le i \le d,\ \left|\frac{x_i}{B_i}\right| < 1 \end{cases}}$$

Where B = (B_1, . . . ,B_d) is the bandwidth of the kernel function. We use Scott's rule to set B [40, 8]: Bi = √5σi|R|− 1d+4, where σ_i is the standard deviation of the values in T in dimension i.

III. Outlier Detection Methods for Low Dimensional Data

3.1 Statistical method

The statistical outlier detection methods can be broadly classified into two categories, i.e., the parametric methods and the non-parametric methods. The major differences between these two classes of methods lie in that the parametric methods assume the underlying distribution of the given data and estimate the parameters of the distribution model from the given data [41] while the non-parametric methods do not assume any knowledge of distribution characteristics Statistical outlier detection methods (parametric and non-parametric) typically take two stages for detecting outliers, i.e., the training stage and test stage.
•*Training stage*The training stage mainly involves fitting a statistical model or building data profiles based on the given data. Statistical techniques can be performed in a supervised, semi-supervised, and unsupervised manner. Supervised techniques estimate the probability density for normal instances and outliers. Semi-supervised techniques estimate the probability density for either 10 normal instances, or outliers, depending on the availability of labels. Unsupervised techniques determine a statistical model or profile which fits all or the majority of the instances in the given data set;
• *Test stage*. Once the probabilistic model or profile is constructed, the next step is to determine if a given data instance is an outlier with respect to the model/profile or not. This involves computing the posterior probability of the test instance to be generated by the constructed model or the deviation from the constructed data profile. For example, we can find the distance of the data instance from the estimated mean and declare any point above a threshold to be an outlier.

3.2 Parametric Method

Parametric statistical outlier detection methods explicitly assume the probabilistic or distribution model(s) for the given data set. Model parameters can be estimated using the training data based upon the distribution assumption. The major parametric
outlier detection methods include Gaussian model-based and regression model-based methods.

3.2.1 Gaussian Models

Detecting outliers based on Gaussian distribution models have been intensively studied. The training stage typically performs estimation of the mean and variance (or standard deviation) of the Gaussian distribution using Maximum Likelihood Estimates (MLE). To ensure that the distribution assumed by human users is the optimal or close-to-optima underlying distribution the data fit, statistical discordany tests are normally conducted in the test stage. So far, over one hundred discordancy/outlier tests have been developed for different circumstances, depending on the parameter of dataset (such as the assumed data distribution) and parameter of distribution (such as mean and variance), and the expected number of outliers .

The rationale is that some small portions of points that have small probability of occurrence in the population are identified as outliers. The commonly used outlier tests for normal distributions are the mean-variance test and box-plot test .In the mean-variance test for a Gaussian distribution N(μ, σ2), where the population has a mean μ and variance σ, outliers can be considered to be points that lie 3 orMore standard deviations (i.e., ≥ 3σ) away from the mean. This test is general and can be applied to some other commonly used distributions such as Student t distribution and Poisson distribution, which feature a fatter tail and a longer right tail than a normal distribution, respectively. The box-plot test draws on the box plot to graphically depict the distribution of data using five major attributes.Smallest non-outlier observation (min), lower quartile (Q1), median, upper quartile (Q3), and largest non-outlier observation (max). The quantity Q3-Q1 is called the Inter Quartile Range (IQR). IQR provides a means to indicate the boundary beyond which the data will be labelled as outliers; a data instance will be labelled as an outlier if it is located 1.5*IQR times lower than Q1 or 1.5*IQR times higher than Q3. In some cases, a mixture of probabilistic models may be used if a single model is not sufficient for the purpose of data modelling. If labelled data are available, two separate models can be constructed, one for the normal data and another for the outliers. The membership probability of the new instances can be quantified and they are labelled as outliers if their membership probability of outlier probability model is higher than that of the model of the normal data. The mixture of probabilistic models can also be applied to unlabelled data, that is, the whole training data are modelled using a mixture of models. A test instance is considered to be an outlier if it is found that it does not belong to any of the constructed models.

3.2.2. Regression

If the probabilistic model is unknown regression can be employed for model con struction. The regression analysis aims to find a dependence of one/more random variable(s) Y on another one/more variable(s) X. This involves examining the conditional probability distribution Y|X. Outlier detection using regression techniques are intensively applied to time-series data. The training stage involves constructing a regression model that fits the data. The regression model can either be a linear or non-linear model, depending on the choice from users. The test stage tests the regression model by evaluating each data instance against the model. More specifically, such test involves comparing the actual instance value and its projected value produced by the regression model. A data point is labelled as an outlier if a remarkable deviation occurs between the actual value and its expected value produced by the regression model. Basically speaking, there are two ways to use the data in the dataset for building the regression model for outlier detection, namely the reverse search and direct search methods. The reverse search method constructs the regression model by using all data available and then the data with the greatest error are considered as outliers and excluded from the model. The direct search approach constructs a model based on a portion of data and then adds new data points incrementally when the preliminary model construction has been finished. Then, the model is extended by adding most fitting data, which are those objects in the rest of the population that have the least deviations from the model constructed thus far. The data added to the model in the last round, considered to be the least fitting data, are regarded to be outliers.

Non-parametric Methods The outlier detection techniques in this category do not make any assumptions about the statistical distribution of the data. The most popular approaches for outlier detection in this category are histograms and Kernel density function methods. A. HistogramsThe most popular non-parametric statistical technique is to use histograms to maintain a profile of data. Histogram techniques by nature are based on the frequency or counting of data. The histogram based outlier detection approach is typically applied when the data has a single feature. Mathematically, a histogram for a feature of data consists of a number of disjoint bins (or buckets) and the data are mapped into one (and only one)bin. Represented graphically by the histogram graph, the height of bins corresponds to the number of observations that fall into the bin. Thus, if we let n be the total number of instances, k be the total number of bins and mi be the number of data pointin the ith bin ($1 \leq i \leq k$), the histogram satisfies the following condition n = P$_k$i=1to 13.The training stage involves building histograms based on the different values taken by that feature in the training data. The histogram techniques typically define a measure between a new test instance and the histogram based profile to determine if it is an outlier or not. The measureis defined based on how the histogram is constructed in the first place. Specifically,there are three possible ways for building a histogram: 1. The histogram can be constructed only based on normal data. In this case, the histogram only represents the profile for normal data. The test stage evaluates whether the feature value in the test instance falls in any of the populated bins of the constructed histogram. If not, the test instance is labelled as an outlier.

The histogram can be constructed based on a mixture of normal data and outliers. This is the typical case where histogram is constructed. Since normal data typically dominate the whole data set, thus the histogram represents an approximated profile of normal data. The sparsity of a bin in the histogram can be defined as the ratio of frequency of this bin against the average frequency of all the bins in the histogram. A bin is considered as sparse if such ratio is lower than a user-specified threshold. All the data instance falling into the sparse bins are labelled

as outliers. The first and second ways for constructing histogram, as presented above, rely on the availability of labelled instances, while the third one does not. For multivariate data, a common approach is to construct feature-wise histograms. In the test stage, the probability for each feature value of the test data is calculated and then aggregated to generate the so-called outlier score. A low probability value
Corresponds a higher outlier score of that test instance.

IV. How to select the window size in stream data processing

Dealing with data whose nature changes over time is one of the core problems in data mining and machine learning. In this chapter we discuss ADWIN, an adaptive sliding window algorithm, as an estimator with memory and change detector with the main properties of optimality. We study and develop also the combination of ADWIN with Kalman filters.
Most strategies in the literature use variations of the sliding window idea: a window is maintained that keeps the most recently read examples, and from which older examples are dropped according to some set of rules. The contents of the window can be used for the three tasks: 1) to detect change (e.g., by using some statistical test on different sub windows), 2) obviously, to obtain updated statistics from the recent examples, and 3) to have data to rebuild or revise the model(s) after data has changed. The simplest rule is to keep a window of some fixed size, usually determined *a priori* by the user. This can work well if information on the time-scale of change is available, but this is rarely the case. Normally, the user is caught in a trade off without solution: choosing a small size (so that the window reflects accurately the current distribution) and choosing a large size (so that many examples are available to work on, increasing accuracy in periods of stability).
A different strategy uses a *decay function* to weight the importance of examples according to their age (see e.g. [39]): the relative contribution of each dataitem is scaled down by a factor that depends on elapsed time. In this case, the trade-off shows up in the choice of decay constant that should match the unknown rate of change. Less often, it has been proposed to use windows of variable size. In general, one tries to keep examples as long as possible, i.e., while not proven stale. This delivers the users from having to guess *a priori* an unknown parameter such as the time scale of change. However, most works along these lines that we know the heuristics and have no rigorous guarantees of performance. Some works in computational learning theory describe strategies with rigorous performance bounds, but to our knowledge they have never been tried in real learning/mining contexts and often assume a known bound on the rate of change.

V. Conclusions

In this paper we discussed the issues that need to beConsidered when designing a stream data association rule mining technique. We reviewed how these issues are handled in the existing literature. We also discussed issues that are application-dependent the current stream data mining methods require users to define one or more parameters before their execution. However, most of them do not mention how users can adjust these parameters online while they are running. It is not desirable/feasible for users to wait until a mining algorithm to stop before they can reset the parameters. This is because it may take a long time for the algorithm to finish due to the continuous arrival and huge amount of data streams. Some proposed methods let users adjust only certain parameters online, but these parameters may not bethe key ones to the mining algorithms, and thus are not enough for a user friendly mining environment. For example, in [Ghoting, 2004], the authors proposed a method to mine distributed data streams which allows the users, to modify online only one of the mining parameters, the response time, to trade-off between the query response time and accuracy of the mining results. For further improvement, we may consider to either let users adjust online or let the mining algorithm auto-adjust most of the key parameters in association rule mining, such as support, confidence and error rate.
Research in data stream association rule mining is still in its early stage. To fully address the issues discussed in this paper would accelerate the process of developing association rule mining applications in data stream systems. As more of these problems are solved and more efficient and user-friendly mining techniques are developed for the end users, it is quite likely that in the near future data stream association rule mining will play a key role in thebusiness world.

VI. Acknowledgement

This material is based upon work submitted by Dr.ZhiZhong Professor in Queensland University and the recommendations expressed in this material are not copied directly but taken as a base.

References

[1] Towards Outlier Detection for High-dimensional Data Streams Using Projected Outlier Analysis strategy
[2] Dr Ji Zhang Dalhousie University Halifax, Nova Scotia December 2008
[3] Barnett, V., Lewis, T.: Outliers in statistical data, 3rd edn. Wiley, Chichester (2001)
[4] Eskin, E.: Anomaly detection over noisy data using learned probability distributions. In: ICML (2000)

[5] Ng, R.T., Han, J.: Efficient and effective clustering methods for spatial data mining. In: VLDB (1994)
[6] Ester, M., Kriegel, H.P., Xu, X.: A database interface for clustering in large spatial databases. In: KDD (1995)
[7] Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. In: SIGMOD (1996)
[8] Sheikholeslami, G., Chatterjee, S., Zhang, A.:Wavecluster: A multi-resolution clustering approach for very large spatial databases. In: VLDB (1998)
[9] Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: SIGMOD (1998)
[10] Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: VLDB (1998)
[11] Knorr, E.M., Ng, R.T.: Finding intensional knowledge of distance-based outliers. In: VLDB (1999)
[12] Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: SIGMOD (2000)
[13] Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: KDD (2003)
[14] Ghoting, A., Parthasarathy, S., Otey, M.E.: Fast mining of distance-based outliers in high-dimensional datasets. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, Springer, Heidelberg (2006)
[15] Wu, M., Jermaine, C.: Outlier detection by sampling with accuracy guarantees. In: KDD (2006)
[16] Tao, Y., Xiao, X., Zhou, S.: Mining distance-based outliers from large databases in any metric space. In: KDD (2006)
[17] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: SIGMOD (2000)
[18] Papadimitriou, S., Kitagawa, H., Gibbons, P.B., Faloutsos, C.: Loci: Fast outlier detection using the local correlation integral. In: ICDE (2003)
[19] Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: KDD (2005)
[20] Jin, W., Tung, A.K.H., Han, J., Wang, W.: Ranking outliers using symmetric neighborhood relationship. In: Ng,W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.)
[21] PAKDD 2006. LNCS (LNAI), vol. 3918, Springer, Heidelberg (2006)
[22] Babcock, B., Olston, C.: Distributed top-k monitoring. In: SIGMOD (2003)
[23] Manjhi, A., Shkapenyuk, V., Dhamdhere, K., Olston, C.: Finding (recently) frequent items in distributed data streams. In: ICDE (2005)
[24] Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., Gunopulos, D.: Online outlier detection in sensor data using non-parametric models. In: VLDB (2006)
[25] Cormode, G., Muthukrishnan, S., Zhuang, W.: Conquering the divide: Continuous clustering of distributed data streams. In: ICDE (2007)
[26] Charu C. Aggarwal. Data Streams: Models and Algorithms. Springer Berlin Heidelberg, 2007.
[27] Brian Babcock, ShivnathBabu, MayurDatar, Rajeev Motwani, and Jennifer Widom.Models and issues in data stream systems. In 21st ACM SIGMOD-SIGACT-SIGARTSymposium on Principles of Database Systems (PODS), New York, NY,USA, 2002. ACM Press.
[28] Mohamed MedhatGaber, ArkadyZaslavsky, and ShonaliKrishnaswamy.Miningdata streams: a review. ACM SIGMOD Record, 34(2): 2005.
[29] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association,* vol. 58, pp. 13-30, 1963.
[30] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "Mining data streams: a review," *ACMSIGMOD Record,* vol. 34, pp. 18-26, 2005.