IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

An Event based Micro Continuous Outlier Detection

M. Obulakshmi

B.E, Computer Science and Engineering (Pre Final Year), PSG College of Technology, Coimbatore, India

Abstract: Anomaly detection is an important data mining task, aiming at the discovery of elements that show significant diversion from the expected behaviour; such elements are termed as outliers. In this paper studies the problem of outlier detection on continuous data streams. The proposed system EMCOD (Event based Micro Cluster-Based Continuous Outlier Detection)algorithms for continuous outlier monitoring on deterministic data streams based on the sliding window. In this paper, design efficient algorithms for continuous monitoring of distance-based outliers, in sliding windows over data streams, aiming at the elimination of the limitations of previously proposed SVDD algorithms. The primary concerns are efficiency improvement and storage consumption reduction. The proposed algorithms are based on an event-based framework that takes advantage of the expiration time of objects to avoid unnecessary computations. The EMCOD algorithm is an outlier detection method based on micro-cluster. Thesis technique is able to reduce the required storage overhead, run faster than previously proposed SVDD technique and offers significant flexibility. Experiments performed on real-life as well as synthetic data sets.

Keywords: Anomaly Detection, Outlier Detection, Distance Calculation, Cluster Outlier.

I. INTRODUCTION

Intrusion detection is a very important topic of network security that has received much attention since potential cyber threats are making the organizations vulnerable. Intrusion Detection Systems (IDS) are intended to protect information systems against intrusions and attacks and are traditionally based on signatures of known attacks. The main drawback is that in case of an emerging attack, based on the recent discovery of a new security hole for instance, the IDS will ignore it since this new attack has not yet been listed in the base of signatures.

In many data analysis tasks a large number of variables are being recorded or sampled. One of the first steps towards obtaining a coherent analysis is the detection of outlaying observations. Although outliers are often considered as an error or noise, they may carry important information. Detected outliers are candidates for aberrant data that may otherwise adversely lead to model misspecification, biased parameter estimation and incorrect results. It is therefore important to identify them prior to modelling and analysis. Outliers are also referred to as abnormalities, discordant, deviants, or anomalies in the data mining and statistics literature. In most applications , the data is created by one or more generating processes, which could either reflect activity in the system or observations collected about entities. When the generating process behaves in an unusual way, it results in the creation of outliers. Therefore, an outlier often contains useful information about abnormal characteristics of the systems and entities, which impact the data generation process. The recognition of such unusual characteristics provides useful application-specific insights.

In principle, an outlier detection technique can be considered as a mapping function f that can be expressed as $f(p) \rightarrow q$, where $q \in RE^{+}$. Giving a data point p in the given dataset, a corresponding outlierness score is generated by applying the mapping function f to quantitatively reflect the strength of outlier-ness of p. Based on the mapping function, there are typically two major tasks for outlier detection problem to accomplish, which leads to two corresponding problem formulations. From the given dataset that is under study, one may want to find the top k outliers that have the highest outlier-ness scores or all the outliers whose outlier-ness score exceeding a user specified threshold.

The exact techniques or algorithms used in different outlier methods may vary significantly, which are largely dependent on the characteristic of the datasets to be dealt with. The datasets could be static with a small number of attributes where outlier detection is relatively easy. Nevertheless, the datasets could also be dynamic, such as data streams, and at the same time have a large number of attributes. Dealing with this kind of datasets is more complex by nature and requires special attentions to the detection performance(including speed and accuracy) of the methods to be developed.

In this paper proposed a new algorithm Event based Micro Continuous Uncertain Outlier Detection(EMCOD) is designed for outlier detection on uncertain data streams. The algorithm is able to quickly determine the nature of an uncertain element by probabilistic pruning, to further improve the efficiency. The propose a pruning approach Probability Pruning for Continuous Uncertain Outlier Detection (EMCOD) based on EMCOD by estimating the outlier probability. In this way, the amount of calculation is effectively reduced, thus meeting with the real-time criteria of processing uncertain data streams.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

II. RELATED WORK

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. Outlier detection, also known as anomaly detection in some literatures, has become the enabling underlying technology for a wide range of practical applications in industry, business, security and engineering, etc. For example, outlier detection can help identify suspicious fraudulent transaction for credit card companies. It can also be utilized to identify abnormal brain signals that may indicate the early development of brain cancers. Due to its inherent importance in various areas, considerable research efforts in outlier detection have been conducted in the past decade. A number of outlier detection techniques have been proposed that use different mechanisms and algorithms. This paper presents a comprehensive review on the major state-of-the-art outlier detection methods. We will cover different major categories of outlier detection approaches and critically evaluate their respective advantages and disadvantages.

A. Statistical Detection Methods

Statistical outlier detection methods rely on the statistical approaches that assume a distribution or probability model to fit the given dataset. Under the distribution assumed to fit the dataset, the outliers are those points that do not agree with or conform to the underlying model of the data. 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 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.

B. Distance-based Methods

There have already been a number of different ways for defining outliers from the perspective of distance related metrics. Most existing metrics used for distance based outlier detection techniques are defined based upon the concepts of local neighbourhood or k nearest neighbours (kNN) of the data points. The notion of distance-based outliers does not assume any underlying data distributions and generalizes many concepts from distribution-based methods. Moreover, distance-based methods scale better to multi-dimensional space and can be computed much more efficiently than the statistical-based methods.

In distance-based methods, distance between data points is needed to be computed. We can use any of the Lp metrics like the Manhattan distance or Euclidean distance metrics for measuring the distance between a pair of points. Alternately, for some other application domains with presence of categorical data (e.g., text documents), non-metric distance functions can also be used, making the distance-based definition of outliers very general. Data normalization is normally carried out in order to normalize the different scales of data features before outlier detection is performed.

C. Density-based Methods

Density-based methods use more complex mechanisms to model the outlier-ness of data points than distance based methods. It usually involves investigating not only the local density of the point being studied but also the local densities of its nearest neighbours. Thus, the outlier-ness metric of a data point is relative in the sense that it is normally a ratio of density of this point against the averaged densities of its nearest neighbours. Density-based methods feature a stronger modelling capability of outliers but require more expensive computation at the same time. What will be discussed in this subsection are the major density-based methods called LOF method, COF method, INFLO method and MDEF method.

D. Clustering-based Methods

The final category of outlier detection algorithm for relatively low dimensional static data is clustering based. Many data-mining algorithms in literature find outliers as a by-product of clustering algorithms themselves and define outliers as points that do not lie in or located far apart from any clusters. Thus, the clustering techniques implicitly define outliers as the background noise of clusters. So far, there are numerous studies on clustering, and some of them are equipped with some mechanisms to reduce the adverse effect of outliers, such as CLARANS, DBSCAN, BIRCH, Wave Cluster .More recently, we have seen quite a few clustering techniques tailored towards subspace clustering for high-dimensional data including CLIQUE and HP-Stream.

III.PROPOSED APPROACH

Data stream is a special data model in data processing, characterized by high data throughput rate and potentially large amounts of data. In this model, data does not take the form of persistent relations, but rather arrives in multiple continuous rapidly time-varying data streams. In this case, traditional algorithms fail to meet the needs of processing. Since the stream is continuously updated, it is impossible to maintain all tuples in local memory. A sliding window is used to track the most recent data.

In data stream applications, data volumes are huge, meaning that it is not possible to keep all data memory resident. Instead, a sliding window is used, keeping a percentage of the data set in memory. The data objects maintained by the sliding window are termed active objects. When an object leaves the window we say that the object expires, and it is deleted from the set of active objects. There are two basic types of sliding windows:(i) the count-based window which always maintains the n most recent objects and (ii) the time-based window which maintains all objects arrived the last t time instances. In

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

both cases, the expiration time of each seen object is known. The challenge is to design efficient algorithms for outlier monitoring, considering the expiration time of objects. Another important factor of stream-based algorithms is the memory space required for auxiliary information. Storage consumption must be kept low, enabling the possible enlargement of the sliding window, to accommodate more objects.

A. Synthetic Data Generation

The datasets to test the training speed of RBF network based on SC. The variables were scaled in[0,1] so that all the input and output values were of the same order of magnitude. The parameters ε, α and β were set to 0.4, 0.7 and 0.4 respectively. We compared this training model with the conventional RBF model which randomly selects an instance in training set as the hidden node centre. The dataset can select more desirable hidden node centres, thereby leading to a proper network structure. For each data set, the mean squared error of SC based RBF network is lower. Especially, its training time is remarkably reduced. In RBF network, most of the methods employ unsupervised learning utilizing only the normal class during training since abnormal patterns are not usually available. Although outliers are not sufficient to construct a binary classifier, they can help refine the boundary around the normal patterns determined by the unsupervised method [15]. Generalization is to include the normal patterns within the boundary while specialization is to exclude patterns from all other classes. A balance between the two concepts is critical to classification accuracy. Even though outlier detection model is able to generalize from the normal data, most of them cannot specialize from data but only from a particular internal bias since they do not take outliers into consideration. In this sense, utilizing abnormal data helps a classification model specialize from data. It has been experimentally shown that one can achieve a higher accuracy by utilizing abnormal data. As mentioned above ,while training the RBF neural network, we added a regularization term to the network error function so as to prevent reconstruction errors of abnormal patterns from being lower than a threshold. Then, the unseen data will be input into the trained network and those with higher degree of outlier, which is defined below, will be candidate outliers.

B. The Event-Based Approach

The proposed approach is interested in tracking the outliers in a set of objects of a stream defined by a sliding window. In particular, a set of outliers is maintained subject to arrivals of new objects from the stream and departures of existing objects due to the restricted window size (either restricted with respect to time or with respect to number of objects). The arrival and departure of objects has the effect of a continuously evolving set of outliers. At only certain discrete moments, however, this set may change and an inlier becomes an outlier or vice-versa. Between these discrete moments, the set of outliers remains as is. The idea is to focus on the temporal and geometric relations between objects to guarantee the correctness of the set of outliers for a period of time.

The effect of arrivals of objects is to turn existing outliers into inliers. On the other hand, the potential effect of departures is to turn inliers into outliers. However, the exact time of the departure of each object is pre-specified (due to the sliding window) and thus we can plan in the future the exact moments in which one needs to check whether an inlier has turned into outlier. Unfortunately, this is not the case for arrivals since it has no information about them (unless probabilistic or of similar flavour assumptions are made).

Henceforth, an event is the process of checking whether an inlier becomes an outlier due to departure of objects from the window. The expiration time of the objects is known whether we talk about time-based windows (in this case a new object p has expiration time now + d WSlide e) or for count-based windows (in this case p expires after a predefined number of new objects have arrived). Thus, the time stamp of an event depends on the expiration time of objects. This forces a total order on the events which can be organized in an event queue.

An event queue is a data structure that supports efficiently the following operations:

- Find min: returns the event with the most recent timestamp (the most recent event).
- Extract min: invokes a call to find min and deletes this event from the event queue.
- Increase time(p, t): increases the time stamp of the event associated to object p by t. It is assumed that we are provided with a pointer to p and there is no need to search for it.
- Insert(p, t): inserts an event for object p into the queue with time stamp t.

These operations can be supported efficiently by a disordered priority queue. Employing a Fibonacci heap allows us to support these operations in O(1) worst-case time as well a sin O(log n), O(1) and O(1) amortized time respectively. Note that due to the min-order of the heap, these structures support the operation of decrease time which, however, can be trivially changed to support the operation of increase time. The event-based method for outliers employs an event queue to efficiently schedule the necessary checks that have to be made when objects depart. Thus, in the event queue there are only stored inliers since only these can be affected by the departure of an object. Arrival of new objects results in potential updates of the keys of some objects in the event queue. Additionally, existing outliers are checked as to whether they have become inliers and thus they should be inserted in the event queue. The proposed framework to describe efficient algorithms for continuous outlier detection.

C. Event Based Micro Cluster

Let us assume that, initially, the R and k parameters for outlier detection are fixed. We set the radius of MCi, which is the maximum distance of any object belonging to MCi from mcci, to R/2, and the minimum size of a micro-cluster to k + 1. An object can belong to at most a single micro cluster .As such, there are at most bn/(k +1)c micro-

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

clusters at any window. In general, an object may have neighbours that belong to other micro-clusters. However, the centres of such micro-clusters are within a range of 2R from that object .micro-clusters have been employed in several works to assist clustering in streamed data. Such works tend to build upon the cluster feature vector introduced, to attain a more compact representation of the objects with a view to improving clustering efficiency without sacrificing cluster quality. The actual clustering is performed by a subsequent offline stage. However, in our case micro-clustering serves a different purpose, i.e., outlier detection, and micro clusters are fully tailored to online processing.

The information kept for each object in the current window differs on the basis of the set it belongs to. More specifically, for objects $p \in I^{mc}$, we only keep p.mc. For each object $p \in PD$, we keep the expiration time of the k most recentpreceding neighbors and the number of succeeding neighbours, as described in the previous sections. In addition, we keep a list containing the identifiers of the micro-clusters, whose centres are less than 3/2R far. The reason we keep this information derives from the lemma above. The assignment of objects to those micro-clusters may lead to a change in the status of the potential outliers; in other words, the micro-clusters of this type may affect the objects in PD. The list of identifiers is stored in p. Rmc. Also, we employ a hash data structure so that we can find (i) the objects in each micro-cluster, (ii) the objects deemed as potential outliers, and (iii) the objects in PD referring to a particular micro-cluster in O(1) time. The main rationale behind our approach is to drastically reduce the number of objects that are considered during the range queries when these are performed.

D. Algorithm Implantation

The detailed steps of the modified algorithm after each window slide are as follows:

Step 1: The expired objects are purged after having updated the counters mcn of corresponding micro-clusters (if any),accordingly. Subsequently, steps 2 and 3 are performed for each new data object p; new objects are processed in the order of their arrival.

Step 2: For each p, we detect (i) the micro-cluster, the centre of which is closest to that object, and (ii) all micro-clusters, the centres of which are within a 32R range. Conflicts (i.e., when there are two centres with equal distance) are resolved arbitrarily. Note that we can employ a specific structure to store the micro-cluster centres, such as an M-tree, to perform this task efficiently.

Step 3: If the distance from the closest centre is not greater than R/2, then:(3a-i) the new object is assigned to the corresponding micro cluster and the value of p.mc is updated;(3a-ii) the size of the corresponding micro-cluster is increased by one;(3a-iii) let MCi be the micro-cluster where the new object is inserted. We evaluate the distance between the new object and all objects in PD that contain MCi in their Rmc lists ,to check (i) whether the number of succeeding neighbours of the latter should be increased and (ii) whether any previous reported outliers have become inliers; Otherwise, i.e., if the distance from the closest centre is greater than R/2, no assignment takes place and the following process is applied: (3b-i) For the new object p that has not been assigned to a micro-cluster, we perform a range query taking into account only (i) the objects in PD and (ii) the objects in the micro clusters for which the distance from their centres is not greater than 3/2R (the relevant micro-clusters have been detected in Step 2).(3b-ii) If the number of neighbours from the PD set within R/2 distance exceeds μk, μ Π 12, then a new micro-cluster is created, with the new object as its centre. All the corresponding objects are moved from PD to Imc. All objects still in PD that are less than 3/2R apart update their Rmc lists with the identifier of the new micro-cluster.(3b-iii) Otherwise, the event-based algorithm described in the previous sections (i.e., creation of the list of the expiration number of succeeding neighbours) is applied. The objects in p.Rmc are the cluster identifiers for which the distance from their centres is not greater than 3/2R.

Step 4: If the size of a micro-cluster shrinks below k + 1,then this micro-cluster is dissolved, and its former objects are treated in a way similar to that described in Step 3b.At the end of these steps, additional outliers are reported with the help of the event queue, which in MCOD, does not include any object p ∈Imc. The main advantage compared to the algorithms in the previous sections is that the number of distance computations is reduced significantly. The efficiency of this algorithm is expected to increase with the proportional size of Imc. In other words, if the size of PD is small, and close to the size of the actual outliers, then the performance improvements are expected to be higher. This is the case when the (average) density of the objects is higher than the density threshold implied by the R and k parameters by several factors.

Finally, this methodology can easily support multiple values for k, if the minimum size of a micro-cluster is set to kmax+1.However, for the rest of the values of k, the number of inliers regarded as potential outliers would increase thus leading to performance degradation.

IV. EXPERIMENTAL RESULTS

The two real-life and one synthetic data sets. The real data sets are (i) FC (Forest Cover), available at the UCI KDD Archive (url:kdd.ics.uci.edu), containing 581,012records with quantitative attributes such as elevation, slope etc. and (ii) ZIL (Zillow), extracted from www.zillow.com, containing 1,252,208 records with attributes such as price and number of bedrooms. The synthetic one (IND) contains5M objects with independent attributes that follow a uniform distribution. We study the performance of the proposed methods by varying several of the most important parameters such as the window size W, the distance R, the number of required neighbors k and the number of queries measure the CPU cost, the memory requirements, the number of distance computations and other qualitative measurements .Count-based windows

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

have been used, whereas time-based ones are supported, without significant changes in the results.
Here examines the behavior of the event-based technique. For each method, the following measurements are taken: a) the average number of events that exist in the system ,b) the average number of events triggered by the arrival of new objects, and c) the average number of events processed after each arrival .the total number of events is similar but less events are processed, due to the fact that the average number of neighbors for an object is higher and therefore more objects are safe inliers.

Table 1 Outlier Detection Accuracy with different algorithm

Dataset	STROM	ABOD	KNN	EMCOD
airlines	0:17	0:12	0:25	0:47
Cov type Norm	0:19	0:15	0:35	0:71
kddcup99	0:20	0:14	0:36	0:69
Reuters	0:26	0:16	0:51	0:94

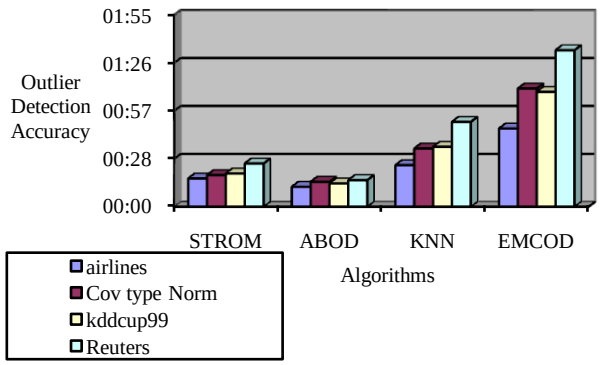

Fig 1. Outlier Detection accuracy

A. CPU Times
First, we study the performance of the methods for varying values of W in the range [10K; 1000K]. Depicts the results. For Slide = 1, the memory requirements for Abstract- C are very high. More specifically, Abstract-C stores W¢(W¡1) 2 counters, which corresponds to 74GB for W = 200K and 465GB for W = 500, assuming integers need 4 bytes. Because of that, in Figure 7, Slide = 1 only for STROM, LOF, SVDD and EMCOD, while we choose Slide = 0:001W for Abstract- C. Despite that favourable configuration, Abstract-C performs significantly worse than our algorithms in terms of running time.

Table2 CPU Time vs different algorithms

Window Size (K)	**STROM**	**ABOD**	**KNN**	**EMCOD**
10	10	0.8	0.7	0.1
100	11	0.9	0.74	0.3
200	12.5	0.95	0.8	0.4
300	12.8	1.5	1	0.8
400	13	1.8	1.3	1
500	13.5	1.9	1.45	1.1

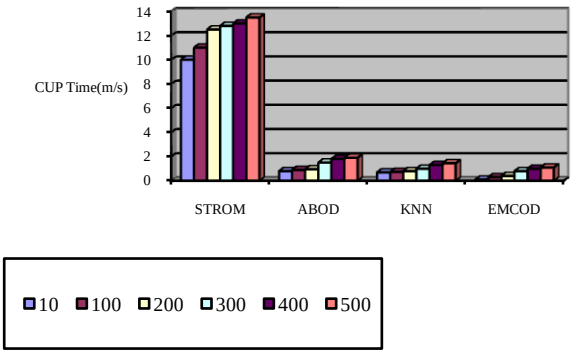

Fig 2 CPU Time for outlier detection

B. Outliers (% W)
The memory requirements, the number of distance computations and other qualitative measurements. Sliding windows have been used, whereas time-based ones are supported, without significant changes in the results. The default values for the parameters (unless explicitly specified otherwise) are: W = n = 200K, jQj = 1, i.e., there is a single query, k = 10 and the parameter R is set in a way that the number of outliers jDj = (0:01 0:001) n. Since we want to investigate the most demanding form of continuous queries, we set Slide = 1. All measurements correspond to 1000 slides, i.e., 1000 insertions/deletions in P.

Table 3(%) Outlier Vs different algorithms

outliers (% W)	**STROM**	**ABOD**	**KNN**	**EMCOD**
0.1	**0.8**	**0.8**	**1**	**2**
0.5	**1**	**1.5**	**1.2**	**2.5**
1	4	3	3.4	4.5
1.5	5	4.6	3.6	5.3
2	8	6.4	5.3	6.65
2.5	9	7.4	6.3	8.4
3	10	8.5	7.8	9.8

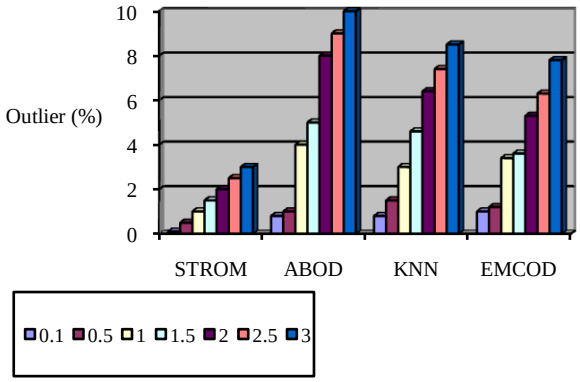

Fig 3 Percentage of outlier detection

C. Memory Consumption
The memory consumption of the two real data sets for the experiment. The consumed memory corresponds to the

DOI 10.17148/IJARCCE.2016.5603

memory needed to store the information for each active object (i.e., preceding and succeeding neighbours), the heap size used for the events prioritization, the outliers of all the queries and the micro-cluster information for KNN. As can be seen, the required amount of memory is only a small fraction of the total memory available in modern machines, even for the COD method.

Table 4 Memory usage vs different algorithm

W	**STROM**	**ABOD**	**KNN**	**EMCOD**
10,000	0.48	2.95	4.29	0.27
200,000	9.60	100.45	111.85	4.94
300,000	14.47	133.27	194.28	11.04
400,000	19.32	178.30	280.37	15.40
500,000	24.23	232.72	377.51	20.23

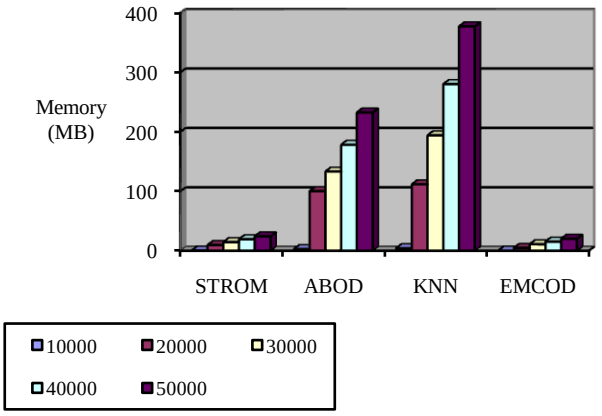

Fig 4 Memory Consumption

V. CONCLUSION

Anomaly detection is an important data mining task aiming at the selection of some interesting objects, called outliers that show significantly different characteristics than the rest of the data set. In this project the problem of continuous outlier detection over data streams, by using sliding windows. More specifically, EMCOD algorithms are design, aiming at efficient outlier monitoring with reduced storage requirements. This method do not make any assumptions regarding the nature of the data, excepts from the fact that objects are assumed to live in a metric space. As it is shown in the performance evaluation results, based on real-life and synthetic data sets, the proposed techniques are by factors more efficient than previously proposed algorithms. An interesting direction for future work is the design of randomized algorithms for detection, aiming at significant improvement of efficiency by sacrificing the accuracy of results.

REFERENCES

[1] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in SIGMOD Conference, 2000, pp. 427–438.
[2] E. Knorr and R. Ng, "Algorithms for mining distance-based outliers in large data sets," in VLDB Conference, 1998.
[3] E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," The VLDB Journal, vol. 8, no. 3-4, pp. 237–253, 2000.
[4] D. Yang, E. Rundensteiner, and M. Ward, "Neighbor-based pattern detection for windows over streaming data," in EDBT, 2009, pp. 529–540.
[5] Y. Zhu and D. Shasha, "Statstream: statistical monitoring of thousands of data streams in real time," in VLDB Conference, 2002, pp. 358–369.
[6] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in VLDB Conference, 1997, pp. 426–435.
[7] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in SDM, 2006.
[8] Aggarwal C C. On density based transforms for uncertain data mining. In Proc. the 23rd International Conference on Data Engineering, April 2007, pp.866-875.
[9] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in Proc. ACM SIGMOD Int. Conf. Manage. Data, New York, NY, USA, 2000, pp. 93–104.
[10] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," Knowl. Inform. Syst., vol. 26, no. 2, pp. 309–336, 2011.
[11] C. C. Aggarwal and P. S. Yu, "A survey of uncertain data algorithms and applications," IEEE Trans. Knowl. Data Eng., vol. 21, no. 5, pp. 609–623, May 2009.
[12] Y. Shi and L. Zhang, "COID: A cluster-outlier iterative detection approach to multi-dimensional data analysis," Knowl. Inform. Syst., vol. 28, no. 3, pp. 709–733, 2011.

BIOGRAPHY

AREA OF INTEREST: Data mining, Artificial neural networks, big data, android, weka, database management system and image processing. Basics of game programming.

DOI 10.17148/IJARCCE.2016.5603