IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

# Accurate Detection of Unusual Event in Network Data Stream (Outlier)

**Priti G. Manekar[1], Prof. Pravin G. Kulurkar[2]**

M.Tech CSE, Vidarbha Institute of Engineering, Nagpur[1]

H.O.D, CSE, Vidarbha Institute of Engineering, Nagpur[2]

**Abstract:** Outlier Mining is an important task of discovering the data records which have an exceptional behavior comparing with other records in the remaining dataset. Outliers do not follow with other data objects in the dataset. There are many effective approaches to detect outliers in numerical data. Most of the earliest work on outlier detection was performed by the statistics community on numeric data. But for categorical dataset there are limited approaches By using memory efficient incremental local outlier (MiLOF) detection algorithm and ROAD (Ranking-based Outlier Analysis and Detection algorithm).

**Keywords:** Outlier detection, Stream data mining, Local outlier, Memory efficiency.

## 1. INTRODUCTION

Outlier detection is the process of detecting instances with unusual behavior that occurs in a system. Effective detection of outliers can lead to the discovery of valuable information in the data. Over the years, mining for outliers has received significant attention due to its wide applicability in areas such as detecting fraudulent usage of credit cards, unauthorized access in computer networks, weather prediction and environmental monitoring.

A number of existing methods are designed for detecting outliers in continuous data. Most of these methods use distances between data points to detect outliers. In the case of data with categorical attributes, attempts are often made to map categorical features to numerical values. Such mappings impose arbitrary ordering of categorical values and may cause unreliable result.

Another issue is related to the big data phenomenon. Many systems today are able to generate and capture real-time data continuously. Some examples include real-time

An outlier is a data point which is significantly different from the remaining data. Hawkins formally defined the concept of an outlier as follows:

"An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."

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.

Some examples are as follows:

Intrusion Detection Systems: In many host-based or networked computer systems, different kinds of data are collected about the operating system calls, network traffic, or other activity in the system. This data may show unusual behavior because of malicious activity. The detection of such activity is referred to as intrusion detection. OUTLIER detection (also known as rare event or anomaly detection) has received considerable attention in the field of data mining because of the need to detect unusual events in a variety of applications, such as fraud detection, human gait analysis and intrusion detection. A variety of outlier detection algorithms has been proposed for use on static data sets which have a finite number of samples. However, outlier detection on streaming data is particularly challenging, since the volume of data to be analyzed is effectively unbounded and cannot be stored indefinitely in memory for processing . Data streams are also generated at a high data rate, hence making the task of outlier detection even more challenging. This is a significant problem that arises in many real applications.

For example, an outlier detection system in wireless sensor networks must work with the limited memory in each sensor node in order to detect rare events in near real time. The cost of communication is an important factor in such systems, which limits the scope for downloading data to a central server for archiving and analysis. Hence a memory efficient outlier detection algorithm is needed for streaming data, as it satisfies the memory constraint in the sensor nodes and avoids the communication cost of having to transfer data to external storage.

## 2. LITERATURE SURVEY

Outlier detection algorithms attempt to find data points that are different from the rest of the data points in a given

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

data set. The problem is of considerable importance, arising frequently in many real-world applications, for data mining researchers. Many practical applications concerning outlier detection occur in different domains such as fraud detection, cyber-intrusion detection, medical anomaly detection, image processing and textual anomaly detection.

Statistics-based approaches were first used for outlier detection based on an assumption that the distributions of datasets are known. A data point was defined as an outlier if it deviates from the existing distribution. With sufficient knowledge about the dataset, statistics-based methods work effectively. But in real-world, unfortunately, distributions of datasets are unknown, signify all points that belong to clusters. The effectiveness of this approach depends on the clustering algorithm. Knorr and Ng propose to detect an outlier based on its distances from neighboring data points, many other variations of distance-based approaches have been discussed in the literature .
Breunig et al. proposed that each data point of the given data set should be assigned a degree of outlierness. In their view, as in other recent studies, a data point's degree of outlierness should be measured relative to its neighbors; hence they refer to it as the Local Outlier Factor (LOF) of the data point. Tang et al.argued that an outlier doesn't always have to be of lower density and lower density is not a necessary condition to be an outlier. They modified LOF to obtain the connectivity-based outlier factor (COF) which they argued is more effective when a cluster and a neighboring outlier have similar neighborhood densities. Local density of a is generally measured in terms of k-nearest neighbors; LOF and COF both exploit properties associated with k-nearest neighbors of a given object in the data set. However, it is possible that an outlier lies in a location between objects from a sparse and a denser cluster. To account for such possibilities, Jin etal.proposed another modification, called INFLO, which is based on a symmetric neighborhood relationship, i.e., the proposed modification considers neighbors and `reverse neighbors' of a data point when estimating its density distribution impacting the performances of these methods.
To overcome this obstacle clustering-based algorithms have been proposed to detect outliers. The basic idea is that a data point is an outlier if it does not belong to any cluster. An outlier is a data object that deviates significantly from the rest of the data. Detection of outliers is aimed at identifying such rare objects and exceptions in a given data set. It is a popular data mining task with applications in various domains such as fraud detection and intrusion detection in computer networks. Consequently, many methods have been developed for outlier detection employing various detection strategies. According to the distance-based methods, a data object is considered unusual if it has very few neighbors in its proximity. On the other hand, the clustering-based methods try to identify various groups of objects based on their intrinsic similarity there by isolating objects with outlier characteristics.
Additionally, many application settings like fraud detection and anomaly detection lend themselves to be posed as unsupervised problems due to lack of prior knowledge about the nature of various outlier objects. This emphasizes the need for developing efficient unsupervised methods for outlier detection.
In many data mining applications, the data objects are described using qualitative (categorical) attributes. The acceptable values of such a qualitative attribute are represented by various categories. The information on the occurrence frequencies of various categories of a categorical attribute in a given data set is very useful for many data-dependent tasks such as outlier detection.
Though there exist a number of methods for outlier detection in numerical data, the problem of outlier detection in categorical data is still evolving. The fundamental challenge in solving this problem is the difficulty in defining a suitable similarity measure over the categorical values. This is due to the fact that the various values that a categorical variable can assume are not inherently ordered. As a result, many data mining related tasks such as determining the nearest neighbor of a categorical object turn out to be non-trivial. Some research efforts in this direction are indicative of the importance of this issue.

## 3.PROPOSED METHODOLGY

Proposed work includes new hybrid approach for outlier detection analysis for Categorical dataset by merging NAVF (Normally distributed attribute value frequency) and Ranking algorithm.
If any dataset consists outliers then it deviates from its from its original behavior and this dataset gives wrong results in any analysis. The algorithm proposed the idea of finding a small subset of the data records that contribute to eliminate the disturbance of the dataset. This disturbance is also called entropy or uncertainty. We can also define it formally as ‘let us take a dataset D with m at-tributes A1, A2--- Am and d(Ai) is the do-main of distinct values in the variable Ai, then the entropy of single attribute Aj is

Because of all attributes are independent to each other, Entropy of the entire dataset
D={ A1, A2-------- Am} is equal to the sum of the entropies of each one of the m attrib-utes, and is defined as follows

When we want to find entropy the algorithm takes k outliers as input . All re-cords in the set are initially designated as non-outliers. Initially all attribute value’s frequencies are computed and using these frequencies the initial entropy of the dataset is calculated.
Then, Granular algorithm scans k times over the data to determine the top k outliers keeping aside one non-outlier each time. While scanning each time every single non-outlier is temporarily removed from the dataset once and the total entropy is recalculated for the remaining dataset. For any non-outlier point that results in the maximum de-

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

crease for the entropy of the remaining data-set is the outlier data-point removed by the algorithm. The Granular algorithm complexity is O(k *n*m*d), where k is the required number of outliers, n is the number of objects in the dataset D, m is the number of attributes in D, and d is the number of distinct attribute values, per attribute.

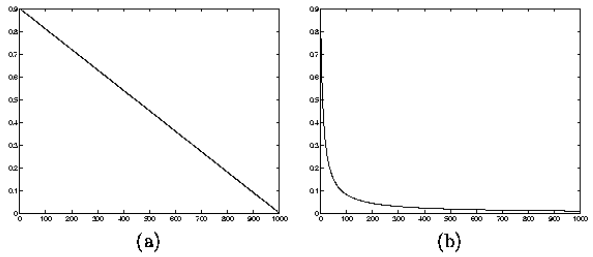

**Figure 2.7:** Learning rates as functions of time

A random seed (or seed state, or just seed) is a number (or vector) used to initialize a pseudorandom number generator. For a seed to be used in a pseudorandom number generator, it does not need to be random.

Because of the nature of number generating algorithms, so long as the original seed is ignored, the rest of the values that the algorithm generates will follow probability distribution in a pseudorandom manner.

**Learning rate Learning rate function**
Learning rate is a decreasing function of time. Two forms that are commonly used are a linear function of time and a function that is inversely proportional to the time t. These are illustrated in the Figure 2.7. Linear alpha function (a) decreases to zero linearly during the learning from its initial value whereas the inverse alpha function (b) decreases rapidly from the initial value Both the functions in the Figure have the initial value of 0.9. The initial values for must be determined. Usually, when using a rapidly decreasing inverse alpha function, the initial values can be larger than in the linear case. The learning is usually performed in two phases. On the first round relatively large initial alpha values are used whereas small initial alpha values are used during the other round.

**Constant bias input**
Input which is not biased to any result.Training transaction, Training modeThe file from user were algorithm will learn outlinear. In our project we are having training and testing, so when we say Training mode it is to train the file and when we say Testing mode, it is to test the file.

**Predictions on training**
SetWhen the algorithm will work it will predict certain pattern on training file like 00001 mean brain,00002 mean eye like that.

**Correctly outlinear detection**
Correctly outlinear detection means ,The actual value in brain and predicted is also eye, then the pattern of eye is different from brain .so hear outlinear is detected. If actual value is brain and predicted is also brain ,then both pattern are same so incorrectly classified.

**Kappa statistic**
kappa coefficient is a statistical measure of inter-rater agreement or inter-annotator agreement for qualitative (categorical) items. It is generally thought to be a more robust measure than simple percent agreement calculation since κ takes into account the agreement occurring by chance.

**Class complexity order 0? Class complexity scheme?**
In computational complexity theory, a complexity class is a set of problems of related resource-based complexity. A typical complexity class has a definition of the form: the set of problems that can be solved by an abstract machine M using O(f(n)) of resource R, where n is the size of the input.

**Mean absolute error (MAE)**
The MAE measures the average magnitude of the errors in a set of forecasts, without considering their direction. It measures accuracy for continuous variables. The equation is given in the library references. Expressed in words, the MAE is the average over the verification sample of the absolute values of the differences between forecast and the corresponding observation. The MAE is a linear score which means that all the individual differences are weighted equally in the average.

**Root mean squared error (RMSE)**
The RMSE is a quadratic scoring rule which measures the average magnitude of the error. The equation for the RMSE is given in both of the references. Expressing the formula in words, the difference between forecast and corresponding observed values are each squared and then averaged over the sample. Finally, the square root of the average is taken. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable.

The MAE and the RMSE can be used together to diagnose the variation in the errors in a set of forecasts. The RMSE will always be larger or equal to the MAE; the greater difference between them, the greater the variance in the individual errors in the sample. If the RMSE=MAE, then all the errors are of the same magnitude.

**Relative absolute error. Root relative square error**
The relative absolute error is very similar to the relative squared error in the sense that it is also relative to a simple predictor, which is just the average of the actual values. In this case, though, the error is just the total absolute error instead of the total squared error. Thus, the relative absolute error takes the total absolute error and normalizes it by dividing by the total absolute error of the simple predictor.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 2, February 2017

**Total number of instance**
Total number of rows or attribute present in data.

**Detailed accuracy by class**
Detail description of how much outlinear has be detected respective each attribute(class) like for brain how much ,for eyes how etc. means 10 for brain,20 for eye.

## 4. IMPLEMENTATION DETAILS

**1. Data**
A large problem when evaluating outlier detection methods is that there are very few real world data sets where it is exactly known which objects are really behaving differently due to belonging to a different mechanism. Though there exist multiple case studies on outlier detection, the question whether an object is an outlier or not is often depending on the point of view. Another problem is that the list of possible outliers is often incomplete making it hard to evaluate whether the algorithm ranked all outliers in the database properly. Therefore, we decided to evaluate our methods on artificially generated data. Thus, we can generate outliers and ordinary data points with respect to the initial definition, i.e. an outlier is a point being generated by a different mechanism than the majority of data objects. To exactly evaluate the behavior of our new method for different dimensionalities and database sizes, we generated multiple data sets having 25, 50 and 100 dimensions. As database sizes (dbsize) we selected 500, 1,000, 5,000 and 10,000 data objects.
In order to find data sets having well-defined but not obvious outliers, we proceeded as follows. First of all, we randomly generated a Gaussian mixture model consisting of ve equally weighted processes having random mean and variance values. This mixture model now describes the ordinary data points, i.e. the none-outlier data points. To build the outliers corresponding to another mechanism that does not assign all the outliers to an additional cluster, we employed a uniform distribution on the complete data space.

This way we generated 10 outliers for each data set which are totally independent on the mixture model describing the general data distribution. Let us note that it is possible that some outliers might be generated in an area being populated by none-outlier objects drawn from the Gaussian mixture model. Thus, even if an outlier detection mechanism works well, it does not necessarily have to rank all outliers into top positions.

**2. Data Processing**
1) Extract nominal feature attribute values from data files
Protocol_type=icmp,udp,tcp
Attack=phf,buffer_overflow,teardrop,guess_passwd,multihop,loadmodule,smurf,spy,normal,land,back,portsweep,warezclient,ftp_write,nmap,satan,rootkit,perl,imap,neptune,warezmaster,ipsweep,pod
Flag=RSTR,S3,SF,RSTO,SH,OTH,S2,RSTOS0,S1,S0,REJService=vmnet,smtp,ntp_u,shell,kshell,aol,imap4,urh_i,netbios_ssn,tftp_u,mtp,uucp,nnsp,echo,tim_i,ssh,iso_tsap,time,netbios_ns,systat,hostnames,login,efs,supdup,http_8001,courier,ctf,finger,nntp,ftp_data,red_i,ldap,http,ftp,pm_dump,exec,klogin,auth,netbios_dgm,other,link,X11,discard,private,remote_job,IRC,daytime,pop_3,pop_2,gopher,sunrpc,name,rje,domain,uucp_path,http_2784,Z39_50,domain_u,csnet_ns,whois,eco_i,bgp,sql_net,printer,telnet,ecr_i,urp_i,netstat,http_443,harvest
This has been done once and wont be required to be done again.

2) Transformations of the nominal values to the numeric value according to as explained in paper A Practical Guide to Support Vector Classification
"We recommend using m numbers to represent an m-category attribute. Only one of the m numbers is one, and others are zero. For example, a three-category attribute such as fred, green, blueg can be represented as (0,0,1), (0,1,0), and (1,0,0).

Our experience indicates that if the number of values in an attribute is not too large, this coding might be more stable than using a single number."
If input file name is kddcup.data_10_percent_corrected-1000 then result of this step would be saved in kddcup.data_10_percent_corrected-1000-transformed
3) Scaling -as explained in same paper, we scale between 0.0 to 1.0
Result of this step would be saved in kddcup.data_10_percent_corrected-1000-transformed-scaled

**Confusion matrix**
In the field of machine learning, a confusion matrix, also known as a contingency table or an error matrix , is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted outlinear, while each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

## 5. CONCULSION

In this project, we introduced a novel, parameter-free approach to outlier detection based on the variance of angles between pairs of data points. This idea alleviates the effects of the curse of dimensionality on mining high-dimensional data where distance-based approaches often fail to offer high quality results. In addition to the basic approach memory efficient incremental local outlier (MiLOF) detection algorithm, we proposed two variants: RANK as an acceleration suitable for low-dimensional but big data sets, and Hybrid,alter-refinement approach as an acceleration suitable also for high-dimensional data. In a

thorough evaluation, we demonstrate the ability of our new approach to rank the best candidates for being an outlier with high precision and recall. Furthermore, the evaluation discusses efficiency issues and explains the influence of the sample size to the runtime of the introduced methods.

## REFERENCES

[1] M. E. Otey, A. Ghoting, and and A. Parthasarathy, "Fast Distributed Outlier Detection in Mixed-Attribute Data Sets," Data Mining and Knowledge Discovery
[2] He, Z., Deng, S., Xu, X., "A Fast Greedy algorithm for outlier mining", Proc. of PAKDD, 2006.
[3] S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining: Pearson Addison-Wesley, 2005
[5] E. Knorr, R. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," VLDB Journal, 2000.
[6] M. M. Breunig, H.-P.Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density based local outliers," presented at ACM SIGMOD International Conference on Management of Data, 2000
[7] S. Papadimitriou, H. Kitawaga, P. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," presented at International Conference on Data Engineering, 2003
[8] Z. He, X. Xu, J. Huang, and S. Deng, "FP-Outlier: Frequent Pattern Based Outlier Detection", Computer Science and Information System (ComSIS'05)," 2005
[9] S. Wu and S. Wang, "Information-Theoretic Outlier Detection for Large-Scale Categorical Data, IEEE Transactions on Knowledge Engineering and Data Engineering,2011
[10] Frank, & A. Asuncion, (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
[11] E. Muller, I. Assent, U. Steinhausen, and T. Seidl, "Outrank: ranking outliers in high dimensional data," in IEEE ICDE Workshop, Cancun, Mexico, 2008, pp. 600–603.
[12] K. Das and J. Schneider, "Detecting anomalous records in categorical datasets," in ACM KDD, San Jose, California, 2007, pp. 220–229.
[13] Z. He, X. Xu, and S. Deng, "A fast greedy algorithm for outlier mining," in PAKDD, Singapore, 2006, pp. 567–576.
[14] Koufakou, E. Ortiz, and M. Georgiopoulos, "A scalable and efficient outlier detection strategy for categorical data," in IEEE ICTAI, Patras, Greece, 2007, pp. 210–217.
[15] S. Guha, R. Rastogi, and S. Kyuseok, "ROCK: A robust clustering algorithm for categorical attributes," in ICDE, Sydney, Australia, 1999, pp. 512–521.
[16] Z. Huang, "A fast clustering algorithm to cluster very large categorical data sets in data mining," in SIGMOD DMKD Workshop, 1997, pp. 1–8.
[17] K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, vol. 31, pp. 651–666, 2010.
[18] F. Cao, J. Liang, and L. Bai, "A new initialization method for categorical data clustering," Expert Systems with Applications, vol. 36, pp. 10 223–10 228, 2009.
[19] Asuncion and D. J. Newman.(2007) UCI machine learning repository.[Online]. Available: http://archive.ics.uci.edu/ml
[20] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets.In Proc. VLDB, 1998.