An Ensemble Angle-Based Outlier Detection for Big Data

Raghda Al-taei[1] and Maryam Amir Haeri[1]

Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran
{raghdafaris,haeri}@aut.ac.ir

Abstract. Outlier detection is an important problem in machine learning and it has many applications such as fraud detection, network anomaly detection, medical diagnosis and data cleaning. However, traditional outlier detection techniques tend to fail in dealing with high-dimensional data especially when they encounter with big data. By the continues generating of data, we are facing massive datasets rather than just high dimensional data, in such conditions, the methods that depend on distance fail to provide correct results due to the vast distance between points which lead to similar distance between all points. Angle based outlier detection is a method proposed for outlier detection in high dimensional spaces. However, it is very time consuming and cannot be used for big data. In this paper, we introduce an angle based outlier detection ensemble method for big data. A dimensionality reduction technique called locality sensitive hash function (LSH) is used to reduce the complexity of big data. Moreover, the method utilizes an ensemble technique to improve angle based outlier detection and then a supervised combination method such as SVM is used to aggregate the ensemble subsets. The method is investigated with several real-world datasets and the results show that the proposed method can efficiently find the outliers.

Keywords: Angle based outlier detection, Big data, Outlier detection, Locality sensitive hash function, Ensemble learning.

1 Introduction

Big data can be analyzed for insights that lead to better decisions, but with a large number of data, the chance of having outlier data grows high. An outlier is a data object that deviates significantly from the normal objects as if it were generated by a different mechanism [1], [2]. Outliers are different from noise data. Noise is random error or variance in a measured variable. Outliers can be divided into two categories (global outliers and the local outliers). Global outliers are data objects that significantly deviate from the other samples of the dataset. Finding global outliers are much simpler than the local outliers. On the other hand, local outliers are a data instance which its density significantly deviates from its neighbors. It is important to note that, in a dataset, different types of

outliers may exist. Thus, we need different types of outlier detection methods[3].

Detecting outliers in big data could be very challenging [4], due to its large size(volume), the high speed of data changing according to time(complexity). The well-known methods may fail to deal with big data to detect outliers. Moreover, big data generated from different sources. Thus they contain various types of outliers. It is essential to pay attention to all types of outliers in big data. Using an ensemble learning method provide us with an opportunity to use different outlier detection strategies to find different types of outliers [5].

The concept of *ensemble model* can be constructed by integrating several learners t in-order to solve a specific problem [6]. The traditional learning methods tend to use one learning technique on the whole data which may contain unwanted data types like noise or outliers making it difficult to set parameters for these datasets, leading to uncertainty about the mod,el that should be used. Ensemble model address this problem by gathering various models together named base-learners they can be a decision tree, neural network, and others. Ensemble method is suitable for big data because ensemble learning provides the possibility of distributed and parallel implementation. Problems such as fraud detection and intrusion detection can be solved by choosing a proper method to detect outliers [7]. Thus, in this paper, we utilize the ensemble learning for detecting the outliers.

Most of the outlier detection methods rely on calculating the distance between points such as [8,9,10]. However, in case of high dimensional data defining a distance measure could be challenging, because the Euclidean distance will be the same for all points due to the sparsity of data [11,12]. Kriegle et al. [13]suggested a method called angle based outlier detection which is less sensitive to distance. This method is designed for high dimensional data and works appropriate in such space. However, this method is not efficient for big data and it is quit time-consuming. The aim of this paper is to improve this method for big data.

In this paper, we consider the high dimensional big data. In this case, bagging can be considered as a good approach for outlier detection in big data where the dataset is split into parts of subsets then, on each subset an outlier detection method is performed. By bagging several versions of the outlier detection methods are use a sample of data. These samples are generated with replacement; this can help to achieve class balance through sampling without losing much information. Moreover, in this paper in order to reduce the complexity of high dimensionality a dimension reduction technique proper for big data such as locality sensitive hashing (LSH) is used. Then an ensemble bagging method is applied to partition the data set into sub-samples, and on each sample, an angle-based outlier detection is applied, this can improve the ability to detect different types of outliers that exist in the data set. To aggregate all ensemble results an SVM combination method is performed.

The organization of this paper is as follows: next section devoted to the preliminary knowledge. Section 3 explains the proposed method in detail. In

section 4 the ensemble angel-based outlier detection method is evaluated. Finally, section 5 is the conclusion.

2 BACKGROUNDS

In this section, some of the most well-known algorithms that inspired our work will be explained starting with the locality sensitive hash function to reduce the dimensions then explaining the SVM method that we use to aggregate the final result of each bag then, an angle-based outlier detection approach which is suitable to detect outliers in big data.

2.1 Locality Sensitive Hash Function (LSH)

The locality sensitive hash function is used as a dimension reduction technique, which works by grouping the most similar data in one bucket without the need to examine every pair (in contrast to methods such as PCA). According to some conditions Fig.1 shows how the close points set to be in one bucket [14].

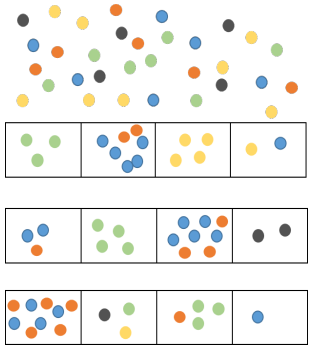

Fig. 1: Basic LSH Structure

LSH assign data points into the same buckets if they are similar. LSH is a proper method to use while working with big data because it works effectively in dealing with big data. LSH indicates that whether the two points s_1 and s_2 are candidate to be similar points or not. LSH supposes that the points are located in space S and a distance measure d such as Hamming, Cosine and Euclidean. Based on the distance measure the hash functions are defined. For each distance measure, a hash function family with random parameters should be define. The hash function family requires four parameters to be defined (d_1, d_2, p_1, p_2)-sensitive [12]. For each two points s_1 and s_2 in S in each LSH family the following two statements should be valid and based on them the two similar points belong to the same bucket with a high probability and the different points belong to the same bucket with low probability.

- If $d(s_1, s_2) < d$ then $h(s_1) = h(s_2)$ i.e. candidate (of being similar) with a p_1 probability at least.
- If $d(s_1, s_2) > d$ then $h(s_1) = h(s_2)$ i.e. candidate (of being similar) with a p_2 probability at most.

LSH Families can be used for many distance measures, here, we explain the LSH family for Euclidean distance measure. Each LSH family needs a random factor. Here the random factor is a random line in the space S. The random line is divided into buckets of size a. The hash function projects each point s_i onto the line and the bucket number is the results of hashing s_i, considering the random line. Two close points in the space S have a high chance of being projected on the same bucket [12],[14]. In order to increase the accuracy of the method several random lines are chosen and each data instance is projected to these lines. Thus, when this method is utilize to reduce the dimensionality of a dataset with d dimensions, if m lines chosen randomly and the data instances project on them and hashed to the bucket numbers, a dataset with m dimensions is obtained.

2.2 Angle base outlier detection method

Traditional methods tend to perform the worse while dealing with high-dimensional data because they depend on the data distances, angle base outlier detection (ABOD) method is less sensitive to distance. Consider Fig.2 the angle for a data point (O) that is in the cluster to any pair of points tend to differ extensively, but the angle for a point that is out of the cluster will be the same to other pairs of points. Thus, the points that have low variance are considered to be outliers.

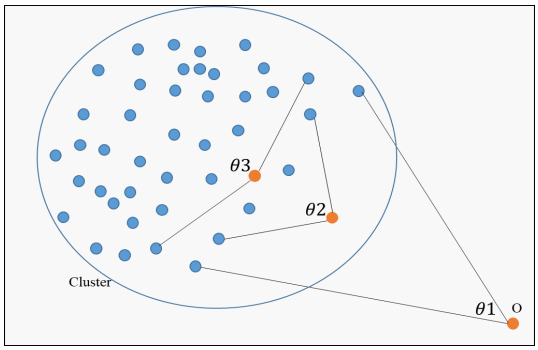

Fig. 2: ABOD for different data points

There are three types of points shown in Fig.2, points within the cluster, border points and outliers the spectrum angles for all three types are considered the variance angle for inlier point tends to be the highest and gets a little lower for a border point but, still not so low as the angle variance of an outlier point [13].

ABOD assign a degree of outlierness to all points by sorting them an outlier can be detected, assigning different degrees to different types of points is not possible with other outlier detectors. The most significant advantage of ABOD is that it does not require any additional parameter. This algorithm was based on the assumption that outliers are data point that are far from the normal data thus, the angles between inlier data are bigger than the angle between an outlier point and the other points, therefore a normal data tend to have a various different angle set between the surrounding points, while abnormal data angles set is a set of several small angles with less variance. The angle between three points can be calculated as in Eq.1 for data points s_1, s_2 and s_3 [15] [1].

$$cos(\overline{s_2} - \overline{s_1}, \overline{s_3} - \overline{s_1}) = \frac{< (\overline{s_2} - \overline{s_1}), (\overline{s_3} - \overline{s_1}) >}{\|\overline{s_2} - \overline{s_1}\|_2 . \|\overline{s_3} - \overline{s_1}\|_2} \tag{1}$$

The ABOF of data X is:

$$ABOF(\overline{s_1}) = Var_{s_2, s_3 \in S} cos(\overline{s_2} - \overline{s_1}, \overline{s_3} - \overline{s_1}) \tag{2}$$

However, performing the ABOD method for high dimensional big data it is very challenging. The dataset cannot be stored in the RAM. And the algorithm needs to read the dataset in many passes thus, this method is quite time-consuming. In the following, we describe that how we can use the ABOD for the high dimensional big data by utilizing the LSH method for the dimensionality reduction and the ensemble bagging for partitioning the data.

2.3 Support vector machine (SVM)

It is used in the case of classification and regression analysis. It can classify given data into two or more categories. By given a sample of the dataset as training data the SVM produce a leaner model that can classify new instances. The model and data points are built in different space separating the classes by a huge gap, SVM with the usage kernel can work for non-linear classification by moving the data into high dimensional space which makes the separation easier. Mainly the SVM was proposed to solve the problem of non-leaner separable classification[16]. In the case where we have a leaner separable data SVM aim to define two hyperplanes that each one lies closest to each class and having the maximum distance from the other class. The middle region between the two hyperplanes called margin and the hyperplane in the middle called the maximum margin hyperplane, in the case where the hyperplane for the first class with label 1 can be calculated by $\boldsymbol{wx} - b = 1$ and for the second class with the label -1 $\boldsymbol{wx} - b = -1$ the distance between the two hyperplanes is $\frac{2}{\|\boldsymbol{w}\|}$.

3 METHODOLOGY

In this section, an outlier detection method that we proposed will be explained. First, LSH which is a suitable dimensionality reduction method that works well

for big data angle base outlier detection method is explained with ABOD. After that ensemble LSH-angle base method which, ensemble method with SVM for combining the ensembles results is suggested.

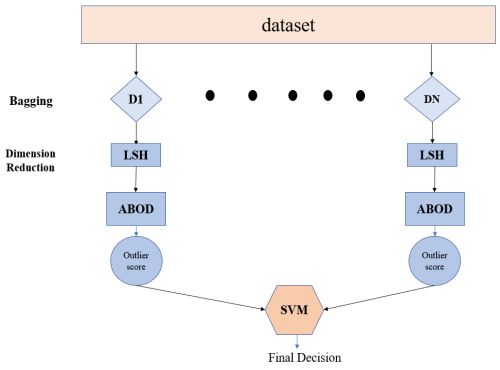

Fig. 3: Ensemble LSH-ABOD

Fig.3 shows the algorithm structure of the method that we are proposing. In the following subsections, we will describe the proposed algorithm in more details.

3.1 LSH angle base outlier detection

In the LSH angle-based outlier detection, at first, the dataset is hashed with a family of locality sensitive hash functions with a different random parameter. Using LSH results in a low dimension data and then the angle-based method is performed over the lower dimensional data. The angle is calculated for every other pair of points and using Eq.2 the variance of that point is measured assigning an outlier degree to all point the least variant points are outliers.

3.2 Ensemble LSH -ABOD method

In order to reduce the time complexity of the LSH based outlier detection method and to improve the results, we suggest an ensemble method based on the LSH-angle based outlier detection. By using an ensemble, the result of a single base learner can be advance this is because ensemble allows the algorithm to be trained on different subsets of the data, which makes it familiar with more data characters in different small spaces. The ensemble helped to produce many accurate results than just using a single outlier detector, in this paper after applying locality sensitive hash functions to reduce the dataset complexity. Afterward an ensemble bagging is performed which, divide the large dataset into chunks with replacement this helps us to obtain accurate result. For each produced bag, angle

base outlier detection is applied to provide each point with an outlierness score low scores indicate outliers and high scores belongs to normal data instances.

3.3 Aggregation of the outlier scores

Thus, in order to find a reasonable threshold for detecting outliers for each bag, a supervised learning method is used. The supervised learning method can be SVM. Moreover each date point might exist in several bags. The SVM can be trained over a minimal dataset, in order to detect the label (normal or anomaly) of the data from the outlier score. After training the SVM in each bag, all the data points of each bag are fed to the SVM, and it detects its label if a data points assigned to more than one bags a majority voting is performed for the final decision. It should be note that in order to learn the SVM the method only needs to very small set of labeled data.

3.4 Time Complexity

At first, the time complexity of the ensemble LSH-angle based method is investigated. The time complexity of the simple angle-based method when all the dataset can be stored in the RAM is $O(n_3)$, where is the size of the dataset. When it encountered with big data, the whole dataset cannot be stored in the RAM. Thus, it needs to access to the disk and read the data points (in each time we want to compute the angles between each data instance to the other ones) which need IO operations, and it can be very time to consume. Therefore, in this case, the time complexity of the angle-based method is $O(n_3 + T_{IO})$. For high dimensional big data, it can be assumed that by using the LSH the volume of data decreases significantly, and the whole dataset or significant part of it can be stored in the RAM. As in the ensemble LSH angle-based method, the dataset is divided into some bags, and the dimensions of the datasets decrease by the LSH. Thus it is possible to assume that the dataset can be stored in the RAM and the time of reading the data repetitively from the disk is saved in this case. Therefore, the time complexity of the ensemble LSH- angle based outlier detection is mostly related to the time complexity of angle-based method over each bag. In the ensemble model computing over each bag can be performed in a parallel manner. The time complexity of angle based method over each bag is $O(n_b^3)$, where n_b is the size of each bag. The final part is aggregating the results which we utilize SVM trained on a minimal training data which the training time is negligibly incomparable to the time complexity of the ABOD and the test part of SVM time complexity is $O(n)$.

4 EXPERIMENTAL RESULTS

In this section, we will discuss the effectiveness and efficiency of the outlier detection method that we use, in comparison with traditional outlier detection methods such as (iforest, LOF, Loop). The method is performed on real datasets.

First, the detection methods both suggested and traditional will be discussed. Second, we will represent their results on real datasets.

4.1 evaluation of outlier detection methods

Three methods such as (iforest [11], LOF [8], Loop [9]) are used to compare our outlier detection method with for the Euclidean space. By using LOF and LoOP methods local outliers can be detected in a dataset, they require distance calculation of all pair point in the dataset and save the resulted calculation in the RAM. Base on points density they tend to be accurate when normal size data is used while in the case of big data this will get more problematic due to the lack of space to save all calculations. iforest it is based on the fact that outliers are less than normal data it constructs small trees with outlier conditions which are fewer than the normal data conditions, it can perform well in parallel computation. Principal Component Analysis (PCA) is used to reduce the dimension before applying the outlier detection methods(iforest, LOF, Loop) in order to see the effect of dimensionality reduction on them.

4.2 Parameters

For the angle base ensemble outlier detection, the parameters that we use are the percentage of data points taken from the dataset. In order to structure the bag size of bagging ensemble model which is equal to 20% of the whole dataset drown with replacement into ten bags, then, by the usage of LSH for all the dataset, the dimension of our big data is decreased to 20 dimensions.

4.3 Results

In order to evaluate the method we used the most well-known datsets. All of these real dataset are available in [17] where preprocessed by [18] that remove the unnecessary features and assign the mode instead of the missing values this will help the outlier detection methods to provide much better results with the least error, the datasets were labeled by 1 referring to an outlier and 0 referring to normal data. Table 1 Shows the datasets in details the number of normal data is higher than the number outliers. To evaluate and analyze the methods the area under the curve (AUC) is used the model is accurate if the result is close to one AUC provides a value between [0,1] the area under the curve is equal to one in perfect conditions. By calculating the ratio between the true positive rate (TPR) and false positive rate (FPR) for various threshold values [0,1]. The true positive rate is $TPR = \frac{TP}{\#P}$ and false positive rate is $FPR = \frac{FP}{\#N}$, where TP is the instance that was classified correctly over the total number of the positive class and FP is the misclassified data as positive over the total number of the negative class.

The results are shown in Table 2 is the results of AUC according to deferent methods and different datasets (real datasets), in most cases provide better

results compared to iforest, LOF, LoOP. For the satimage-2 dataset, our method provided better results in detecting outliers both global and local compared to the well-known method which they failed to provide better detection results. Also for Musk, Mnist , optdigits and speech datasets, we get very accurate results in which none of the (PCA-iforest, PCA-LOF, PCA-LoOp) outlier detectors perform well, the portsweepnormal dataset also our method provided better results than both type detectors. Even when the well-known methods performed better, our result was very close to them, and their difference can also be ignored. Finally, the r2l dataset provided unignorable different between the results in comparison with PCA-LOF and PCA-LoOP.

However, the result of our method over Wbc dataset shows some limitation against detecting collective outliers. A clustering method like K-means can be used at first to cluster collective outlier in one cluster then another outlier detection method can be applied to the rest of the dataset in order to detect the other types of outliers.

The results show that although the time complexity of the ensemble angle-based method is low it can provide accurate results which is comparable with the most famous(and time-consuming) outlier detection methods.

Table 1: Real Data Information

Datasets	**Data**	**Dimension**	**Outliers**
Wbc	278	30	21
Speech	3686	400	61
satimage-2	5803	36	71
optdigits	5216	64	150
Musk	3062	166	97
Mnist	7603	100	700
arrhythmia	452	274	66
portsweepnormal	80793	39	3740
r2l	80247	39	3194
nmapnormal	78619	39	1566

5 Conclusion

This paper suggested an angle based ensemble method for outlier detection in big data. In this method at first the dataset is partitioned into N parts to construct the ensemble learning phase, then LSH function is applied to each data bag. After that SVM aggregates the results of each bag and label the data as inlier or outlier. The time complexity of the method is $O(n_b^3)$ where n_b is the size of each bag and this is quite smaller than the time complexity of ABOD which is equal to $O(n^3)$, where n is the size of the whole dataset. The method is evaluated by real-world data sets. The results demonstrated that the method can provide

Table 2: Real Dataset Results

Datasets	ABOD	Pca-iforest	Pca-LOF	Pca-LoOP
Wbc	0.590	0.6128	0.941	0.9377
Speech	0.682	0.4505	0.4202	0.4156
satimage-2	0.9284	0.6978	0.5849	0.4842
optdigits	0.771	0.4378	0.6518	0.5982
Musk	0.712	0.5345	0.3842	0.4076
Mnist	0.843	0.5328	0.6137	0.5859
arrhythmia	0.747	0.5679	0.7965	0.7964
portsweepnormal	0.831	0.8293	0.4936	0.4824
r2l	0.662	0.7577	0.5816	0.5361
nmapnormal	0.611	0.7934	0.443	0.4864

better results in most cases compared with the well-known methods. However, it was limited to detecting collective outliers and it should be improved in the future studies.

References

1. C. C. Aggarwal, "Outlier analysis," in *Data mining*. Springer, 2015, pp. 237–263.
2. S. Sreevidya *et al.*, "A survey on outlier detection methods," *IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 5, no. 6, 2014.
3. P. Carter, "Big data analytics: Future architectures, skills and roadmaps for the cio," *IDC white paper*, 2011.
4. T. Nasser and R. Tariq, "Big data challenges," *Journal of Computer Engineering and Information Technology*, vol. 9307, no. 2, 2015.
5. T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.
6. Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.
7. A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 5, pp. 363–387, 2012.
8. M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.
9. H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1649–1652.
10. J. Huang, Q. Zhu, L. Yang, and J. Feng, "A non-parameter outlier detection algorithm based on natural neighbor," *Knowledge-Based Systems*, vol. 92, pp. 71–77, 2016.
11. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.
12. J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014.

13. H.-P. Kriegel, A. Zimek *et al.*, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 444–452.
14. W. Hu, Y. Fan, J. Xing, L. Sun, Z. Cai, and S. Maybank, "Deep constrained siamese hash coding network and load-balanced locality-sensitive hashing for near duplicate image detection," *IEEE Transactions on Image Processing*, 2018.
15. G. Pang, L. Cao, L. Chen, D. Lian, and H. Liu, "Sparse modeling-based sequential ensemble learning for effective outlier detection in high-dimensional numeric data." AAAI, 2018.
16. C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.
17. S. Hettich and S. Bay, "The uci kdd archive [http://kdd. ics. uci. edu]. irvine, ca: University of california," *Department of Information and Computer Science*, vol. 152, 1999.
18. S. Rayana, "Odds library," *Stony Brook,-2016. NY: Stony Brook University, Department of Computer Science. URL: http://odds. cs. stonybrook.edu*, 2016.