Vol-2, Issue-8 PP. 1269-1274 ISSN: 2394-5788

GLOBAL JOURNAL OF ADVANCED RESEARCH
(Scholarly Peer Review Publishing System)

RECURSIVE ANTIHUB^2 OUTLIER DETECTION IN HIGH DIMENSIONAL DATA

J.Michael Antony Sylvia
Research Scholar,
Dept. of Computer Science,
St. Xavier's College,
Palayamkottai, Tirunelveli- 627007.
India

Dr.T.C.Rajakumar
Associate Professor,
Dept. of Computer Science,
St.Xavier's College,
Palayamkottai, Tirunelveli- 627002,
India

ABSTRACT

Unsupervised outlier detection is done in a raw data collected from system. To identify unsupervised anomalies in high dimensional data is more complex. Therefore, the main objective of this thesis is to propose the unsupervised anomaly detection in high dimensional data. Anomaly detection in high dimensional data exhibits that as dimensionality increases there exists hubs and antihubs. Hubs are points that frequently occur in k nearest neighbor lists. Antihubs are points that infrequently occur in kNN lists. Outlier detection using AntiHub method is reformulated as Antihub^2 to refine the outlier scores of a point produced by the AntiHub method by considering Nk scores of the neighbors of x in addition to Nk(x) itself. Discrimination of outlier scores produced by Antihub^2 acquires longer period of time with larger number of iterations. Therefore Recursive AntiHub^2 method was introduced to improve the computational complexity of discriminating the outlier scores with reduced number of iterations to detect the more prominent outlier in high dimensional data.

General Terms

Nearest Neighbour, Outlier Detection, Discrimination

Keywords

kNN, AntiHub, AntiHub^2, Recursive AntiHub^2

1. INTRODUCTION

Outliers are data objects that are different or inconsistent with the remaining set of data. Outlier detection is identifying data objects that are deviating from the rest of the objects. Outlier detection methods that use sample data are categorized as supervised methods, semi-supervised methods and unsupervised methods. Supervised outlier detection determines the class of an observation from the classifiers. In the absence of training data for the outliers, the semi supervised methods can be used. The unsupervised methods are used to detect outliers where sample data are not there for outliers or for all the normal observations. Clustering methods can be used as unsupervised outlier detection methods.

Approaches based on nearest neighbors assume that outliers appear far from their closest neighbors. Such methods rely on a distance or similarity measure to find the neighbors, with Euclidean distance being the most popular option. Variants of neighbor-based methods include defining the outlier score of a point as the distance to its k^{th} nearest neighbor

GLOBAL JOURNAL OF ADVANCED RESEARCH
(Scholarly Peer Review Publishing System)

(henceforth referred to as the k- NN method) or as the sum of distances to the k nearest neighbors. An unsupervised outlier detection method in high-dimensional data identifies seven issues in addition to distance concentration: noisy attributes definition of reference sets, bias (comparability) of scores, interpretation and contrast of scores, exponential search space, data-snooping bias, and hubness.

The main objective is to increase the speed of computation and check the performance accuracy in discriminating the outlier scores with a threshold value. The computation speed is increased by reducing the number of iterations filtering the search area using binary search. Recursive AntiHub^2 method is introduced to do this process.

2. OUTLIER DETECTION

Various methods and techniques for outlier detection and the difference of outliers in uniform variate, multivariate techniques and in parametric, non-parametric procedures [1]. The paper highlights the combination of supervised, semi-supervised and unsupervised techniques used for outlier detections and their merits and demerits. The different ways to mine the outliers by ranking the points of the data set with the sum of the distances from its k nearest neighbours[2]. A quick and efficient way to identify k nearest neighbors of each point by linearizing the search space through the Hilbert space filling curve is used. Partition based Algorithm [3] is used to mine top n points as outliers. In partition based algorithm it first partitions the input points using some clustering algorithm and computes lower and upper bounds on the distance of a point from the kth nearest neighbor in each partition. This information is then used to prune the partitions that do not contain the top n outliers and to compute the outliers from the points in the remaining partitions. Since n is small, this algorithm saves the time of computation. Hence this algorithm results better for high dimensional data.

Reverse nearest neighbor search [4] is used on high dimensional and multimedia data. Search algorithm [5] do not require detailed descriptions of objects until their mutual distances can be computed and the distance metric satisfies the triangle inequality. Efficient RNN search is done in generic metric spaces.

3. METHODOLOGY OF ANTIHUB DETECTION

Clustering and outlier detection is one of the major tasks in high dimensional data. Clustering approaches are supported by outlier detections for new optimistic approaches. The thrust of the new optimistic approach applies nearest neighbor based clustering method and detect outliers in high dimensional data. The present method uses AntiHub^2 and Recursive AntiHub^2 algorithm to cluster the nearest neighbor data items and detect outliers in high dimensional data.

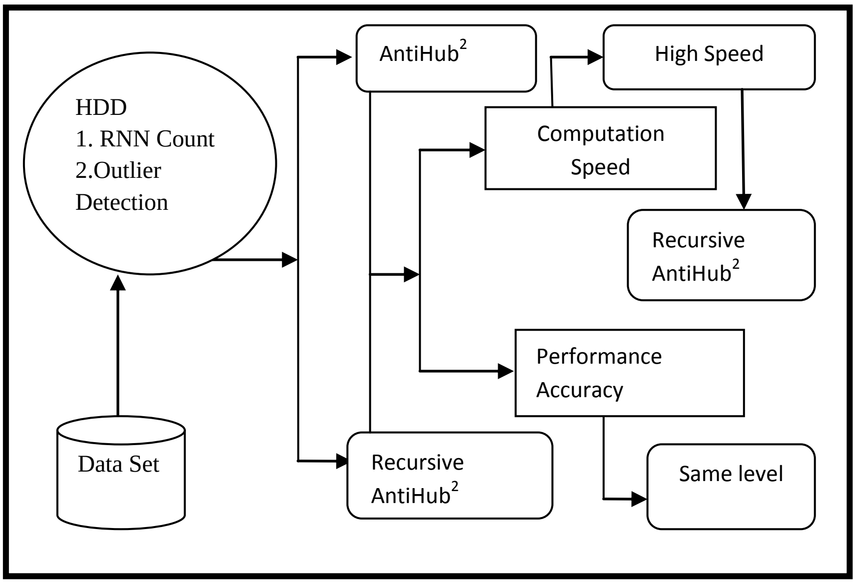

Figure 1: System Architecture of Recursive AntiHub^2

3.1 Data Set

AntiHub^2 and Recursive AntiHub^2 algorithms are compared for outlier detection. In order to perform this comparison, the data sets were taken from UCI machine learning repository [6], from which wilt data set is used for this work. It comprises some training and testing data from a study of remote sensing that performed detecting diseased trees in Quickbird imagery.

3.2 Nearest Neighbor Based Outlier Detection

Given two points *p* and *q*, we use dist (*p*,*q*) to denote the Euclidean distance between *p* and *q*. A k-Nearest-Neighbors query *k*NN(*q*,*k*,*P*) on a dataset P finds the set of *k* objects that are nearest to the query location *q*. Formally, an object *p 2 P* is in the result of *kNN*(*q; k; P*) if and only if it satisfies the following condition:

$$| \{o \in P | dist(o,q) < dist(p,q)\} | < k \quad (1)$$

The notion of distance based outliers is extended by using the distance to the *k*-nearest neighbor to rank the outliers. Nearest Neighbor (NN) is finding the point closer to the query point in a given data set and a query point in k dimensional space. As dimensionality increases, the distance to the nearest data point approaches the distance to the farthest data point.

3.3 Definition Of Hubs And Antihubs

Definition 1: For $q \in (0, 1)$, hubs are the nq points $x \in D$ with the highest values of $N_k(x)$.
Definition 2: For $p \in (0, 1)$, $p < 1 - q$, antihubs are the np points $x \in D$ with the lowest values of $N_k(x)$.

Increase in dimensionality of data causes the k-occurrence distribution to become skewed with a high variance where in some points [hubs] [7] become very frequent members of k-NN list and at the same time, some other points (antihubs) become in frequent neighbors of k-NN lists.

4. EXPERIMENTAL RESULTS

The experimental evaluation was performed on PC Intel Pentium processor, 2GB RAM, OS Windows 7 Ultimate 32-bit. The algorithms are implemented in MATLAB 8.10. Binary search technique is used for the evaluation of computation complexity and Statistical Measures of accuracy is used for performance evaluation. The accuracy is calculated using the true positive rate, false positive rate, false negative rate and true negative rate. Wilt data set is used in this thesis. Wilt data set comprises few training samples for the 'diseased trees' class (74) and many training samples for 'other land cover' class (4265). The data set also contains testing samples for both the classes (500).

Table 1: The Computation Complexity for Antihub^2 and Recursive Antihub^2 when K=100

Dataset	AntiHub^2	Recursive AntiHub^2
wilt	91119	39051

Figure 2 shows, that the proposed system Recursive AntiHub^2 has 50% of reduction in computation complexity when compared with the existing system AntiHub^2.

GLOBAL JOURNAL OF ADVANCED RESEARCH
(Scholarly Peer Review Publishing System)

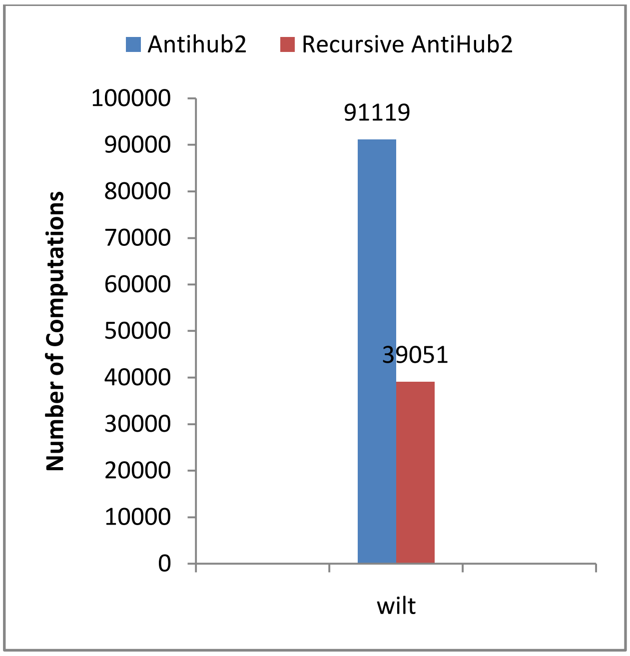

Figure 2: The computation complexity for AntiHub^2 and Recursive AntiHub^2

Table 2: The Performance Accuracy for Antihub^2 and Recursive Antihub^2 when k=10,100 and 120

k value	**AntiHub^2**	**Recursive AntiHub^2**
10	0.9813	0.9825
100	0.9827	0.9827
120	0.9827	0.9827

Figure 3, shows the performance accuracy between AntiHub^2 and Recursive AntiHub^2 algorithms when using the dataset wilt. From the analysis, Recursive AntiHub^2 shows 0.12% of improvement when compared with AntiHub^2 when the value of k is 10.When k=100 and above, we obtain the same level of accuracy. Therefore while comparing the reduction in computation complexity, the result of accuracy is negligible one.

GLOBAL JOURNAL OF ADVANCED RESEARCH
(Scholarly Peer Review Publishing System)

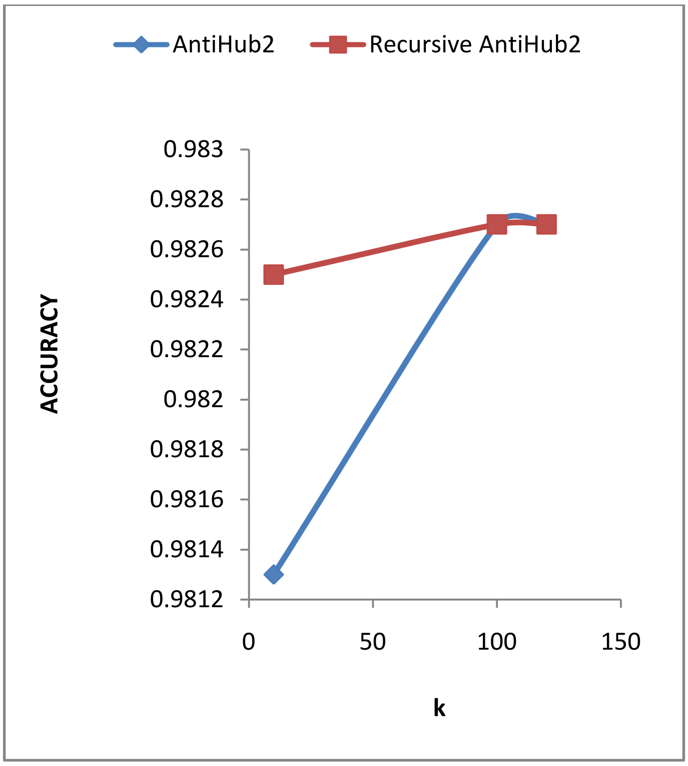

Figure 3: The performance accuracy for AntiHub^2 and Recursive AntiHub^2

5. CONCLUSION

Outliers are filtered by their discriminating score using Recursive AntiHub^2 approach. Prominent outliers are identified on high dimensional data. Search region of k nearest neighbor is defined and an algorithm Recursive AntiHub^2 is used to compute the region based on the data objects. Recursive AntiHub^2 then filters and reduces the search space. Experimental evaluation shows that the search region computed has a strong computation speed. These factors make the Recursive AntiHub^2 more efficient than the existing Antihub^2 method. The experimental evaluation shows that the proposed system Recursive AntiHub^2 has strong computation complexity when compared with the existing system AntiHub^2.

6. REFERENCES

[1] Irad Ben-Gal, 2005. “Outlier Detection”, Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer, Academic Publishers.

[2] Angiulli and C. Pizzuti. 2002. Fast outlier detection in high dimensional spaces. In Proceedings of the 6th European Conference on Principles of Knowledge Discovery and Data Mining (PKDD), Helsinki, Finland.

[3] S.Ramaswamy, R. Rastogi, and K. Shim, 2000. “Efficient algorithms for mining outliers from large data sets,” SIGMOD Rec, vol. 29, no. 2, pp. 427–438,

[4] J. Lin, D. Etter, and D. DeBarr, 2008 “Exact and approximate reverse nearest neighbor search for multimedia data,” in Proc 8th SIAM Int Conf on Data Mining (SDM),pp. 656–667.

GLOBAL JOURNAL OF ADVANCED RESEARCH
(Scholarly Peer Review Publishing System)

[5] Y. Tao, M. L. Yiu, and N. Mamoulis, Reverse nearest neighbor search in metric spaces", IEEE T Knowl Data Engineering,vol. 18, no. 9, pp. 1239–1252, 2006.

[6] K.Bache and M. Lichman, "UCI machine learning repository,"2014. [Online]. Available: http://archive.ics.uci.edu/ml

[7] Milos Radovanovic, Alexandros Nanopoulos and Mirjana Ivanovic, 2014. "Reverse Nearest Neighbors in Unsupervised Distance Based Outlier Detection" IEEE Transactions on Knowledge and Data Engineering,

[8] Karanjit Singh and Dr.Shuchita Upadhyaya 2012. "Outlier Detection: Applications and Techniques" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 3, January.

[9] Dasgupta, D. and Majumdar, N. 2002 "Outlier detection in multidimensional data using negative selection algorithm" In Proceedings of the IEEE Conference on Evolutionary Computation. Hawaii, 1039 - 1044.

[10] K.S.Beyer,J.Goldstein,R.Ramakrishnan and U. Shaft, 1999 "When is "nearest neighbor" meaningful?" in Proc 7th Int Conf on Database Theory (ICDT), pp. 217–235.