Robust Outlier Detection Using Commute Time and Eigenspace Embedding

Nguyen Lu Dang Khoa and Sanjay Chawla

School of Information Technologies, University of Sydney
Sydney NSW 2006, Australia
khoa@it.usyd.edu.au, sanjay.chawla@sydney.edu.au

Abstract. We present a method to find outliers using 'commute distance' computed from a random walk on graph. Unlike Euclidean distance, commute distance between two nodes captures both the distance between them and their local neighborhood densities. Indeed commute distance is the Euclidean distance in the space spanned by eigenvectors of the graph Laplacian matrix. We show by analysis and experiments that using this measure, we can capture both global and local outliers effectively with just a distance based method. Moreover, the method can detect outlying clusters which other traditional methods often fail to capture and also shows a high resistance to noise than local outlier detection method. Moreover, to avoid the $O(n^3)$ direct computation of commute distance, a graph component sampling and an eigenspace approximation combined with pruning technique reduce the time to $O(nlogn)$ while preserving the outlier ranking.

Keywords: outlier detection, commute distance, eigenspace embedding, random walk, nearest neighbor graph.

1 Introduction

Unlike other data mining techniques which extract common or frequent patterns, the focus of outlier detection is on finding abnormal or rare observations in the data. Standard techniques for outlier detection include statistical [7,14], distance based [2,10] and density based [3] approaches. However, standard statistical and distance based approaches can only find global outliers which are extremes with respect to all observations in the dataset. On the other hand local outliers are extremes with respect to their neighborhood observations, but may not be extremes with respect to all other observations in the dataset [16]. A well-known method for detecting local outliers is the Local Outlier Factor (LOF), which is a density based approach [3]. The downside of LOF is the outlier score of each observation only considers its local neighborhood and does not have the global view over all the dataset. Recently, Moonesinghe and Tan [13] proposed a method called OutRank to detect outlier using a random walk on graph. The outlier score is the connectivity of each node which is computed from a stationary random walk. This method cannot find outlying clusters where the node

M.J. Zaki et al. (Eds.): PAKDD 2010, Part II, LNAI 6119, pp. 422–434, 2010.

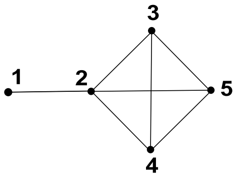


Fig. 1. Example of CD. Edge e_{12} has a larger CD than all edges in the cluster while its Euclidean distance is the same or smaller than their Euclidean distances.

connectivities are still high. An excellent survey by Chandola et. al [4] provides a more detailed view on outlier detection techniques.

In this paper, we present a new method to find outliers using a measure called commute time distance, or commute distance for short (CD)[1]. CD is a well-known measure derived from a random walk on graph [11]. The CD between two nodes i and j in the graph is the number of steps that a random walk, starting from i will take to visit j and then come back to i for the first time. Indeed CD is a Mahalanobis distance in the space spanned by eigenvectors of the graph Laplacian matrix. Unlike traditional Mahalanobis distance, CD between two nodes can capture both the distance between them and their local neighborhood densities so that we can capture both global and local outliers using distance based methods such as methods in [2,10]. Moreover, the method can be applied directly to graph data.

To illustrate, consider a graph of five data points shown in Figure 1, which is built from a dataset of five observations. Denote $d_{ED}(i,j)$ and $d_{CD}(i,j)$ as an Euclidean distance and a CD between observations i and j, respectively. The distances between all pairs of observations are in Table 1.

Table 1. The Euclidean distance and CD for the graph in Figure 1

	Euclidean Distance					Commute Distance				
Index	1	2	3	4	5	1	2	3	4	5
1	0	**1.00**	1.85	1.85	2.41	0	**12.83**	19.79	19.79	20.34
2	1.00	0	1.00	1.00	1.41	12.83	0	6.96	6.96	7.51
3	1.85	1.00	0	1.41	1.00	19.79	6.96	0	7.51	6.96
4	1.85	1.00	1.41	0	1.00	19.79	6.96	7.51	0	6.96
5	2.41	1.41	1.00	1.00	0	20.34	7.51	6.96	6.96	0

It can be seen that $d_{CD}(1,2)$ is much larger than $d_{CD}(i,j)$ $((i,j) \in \{2,3,4,5\}$ $, i \neq j)$ even though $d_{ED}(i,j)$ have the same or larger Euclidean distances than $d_{ED}(1,2)$. The CD from an observation outside the cluster to an observation inside the cluster is significantly larger than the CDs of observations inside the cluster. Since CD is a metric, a distance based method can be used to realize that point 1 is far away from other points using CD. Therefore, the use of CD is promising for identifying outliers. The contributions of this paper are as follows:

[1] A preliminary version of this work appeared as a technical report [8].

- We prove that CD can naturally capture the local neighborhood density and establish a relationship between CD and local outlier detection.
- We propose an outlier detection method using the CD metric to detect global and local outliers. The method can also detect outlying clusters which traditional methods often fail to capture. Moreover, the method is shown to be more resistant to noise than other local outlier detection methods.
- We accelerate the computation of CD using a graph component sampling and an eigenspace approximation to avoid $O(n^3)$ computation. Furthermore, pruning technique is used to calculate the CD 'on demand'. All of them speed up the method significantly to $O(nlogn)$ while preserving the outlier ranking.

The remainder of the paper is organized as follows. Section 2 reviews the theory of random walk on graph and CD. In Section 3, we introduce the method to detect outliers with the CD measure. Section 4 presents a way to approximate CD and accelerate the algorithm. In Section 5, we evaluate our approach using experiments on real and synthetic datasets. Section 6 is the conclusion.

2 Background

2.1 Random Walk on Graph and Stationary Distribution

The random walk on a graph is a sequence of nodes described by a finite Markov chain which is time-reversible [11]. The probability that the random walk on node i at time t selects node j at time $t+1$ is determined by the edge weight on the graph: $p_{ij} = P(s(t+1) = j|s(t) = i) = w_{ij}/d_{ii}$ where $d_{ii} = \sum_{j \in adj(i)} w_{ij}$ and $adj(i)$ is a set of neighbors of node i.

Let P be the transition matrix with entry p_{ij}, A is the graph adjacency matrix, and D is the diagonal matrix with entries d_{ii}. Then $P = D^{-1}A$. Denote $\pi_i(t)$ as the probability of reaching node i at time t, $\pi(t) = [\pi_1(t), \pi_2(t), ..., \pi_n(t)]^{\mathrm{T}}$ as the state probability distribution at time t, the state on transforming is $\pi(t+1) = P^{\mathrm{T}}\pi(t)$ and thus $\pi(t) = (P^{\mathrm{T}})^t\pi(0)$ where $\pi(0)$ is an initial state distribution. The distribution $\pi(t)$ is stationary if $\pi(t) = \pi(0)$ for all $t > 0$.

2.2 Commute Distance

This section reviews two measures of a random walk called hitting time $h(i,j)$ and commute time $c(i,j)$ [11]. The hitting time $h(i,j)$ is the expected number of steps a random walk starting at i will take to reach j for the first time:

$$h(i,j) = \begin{cases} 0 & \text{if } i = j \\ 1 + \sum_{k \in adj(i)} p_{ik} h(k,j) & \text{otherwise.} \end{cases} \tag{1}$$

The commute time, which is known to be a metric and that is the reason for the term 'commute distance' [6], is the expected number of steps that a random walk starting at i will take to reach j once and go back to i for the first time:

$$c(i,j) = h(i,j) + h(j,i). \tag{2}$$

The CD can be computed from the Moore-Penrose pseudoinverse of the graph Laplacian matrix [9,6]. Denote $L = D - A$ and L^+ as the graph Laplacian matrix and its pseudoinverse respectively, the CD is:

$$c(i, j) = V_G(l^+_{ii} + l^+_{jj} - 2l^+_{ij}), \tag{3}$$

where $V_G = \sum_{i=1}^{n} d_{ii}$ is the volume of the graph and l^+_{ij} is the (i, j) element of L^+. Equation 3 can be written as

$$c(i, j) = V_G(e_i - e_j)^{\mathrm{T}} L^+(e_i - e_j), \tag{4}$$

where e_i is the i-th column of an $(n \times n)$ identity matrix I [15]. Consequently, $c(i, j)^{1/2}$ is a distance in the Euclidean space spanned by the e_i's.

3 Commute Distance Based Outlier Detection

3.1 A Proof of Commute Distance Property for Outlier Detection

We now show that CD is a good metric for local outlier detection.

Lemma 1. *The expected number of steps that a random walk which has just visited node i will take before returning back to i is V_G/d_{ii}.*

Proof. For the proof of this Lemma, see [11].

Theorem 1. *Given a cluster C and a point s outside C connected to a point t on the boundary of C (Fig. 2a). If C becomes denser (by adding more points or edges), the CD between s and t increases.*

Proof. From Lemma 1, the expected number of steps that a random walk which has just visited node s will take before returning back to s is $V_G/d_{ss} = V_G/w_{st}$. Since the random walk can only move from s to t, $V_G/w_{st} = h(s, t) + h(t, s) = c(s, t)$ (Fig. 2b). If cluster C becomes denser, there are more edges in cluster C. As a result, V_G increases while w_{st} is unchanged. So the CD between s and t (i.e $c(s, t)$) increases. □

As shown in Theorem 1, the denser the cluster, the larger the CD between a point s outside the cluster to a point t in the cluster. That is the reason why we can effectively detect local outliers using CD.

3.2 Outlier Detection Using Commute Distance

This section introduces a method based on CD to detect outliers. As CD is a metric and captures both the distance between nodes and their local neighborhood densities, we can use a CD based method to find global and local outliers.

First, a mutual k_1-nearest neighbor graph is constructed from the dataset. The mutual k_1-nearest neighbor graph connects nodes u and v if u belongs to the k_1 nearest neighbors of v and v belongs to the k_1 nearest neighbors of u. The

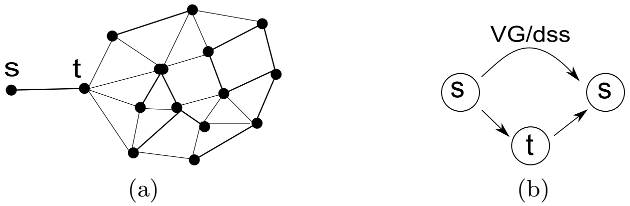


Fig. 2. The CD from an outlier to an observation in a cluster increases when the cluster is denser

reason for choosing mutual k_1-nearest neighbor graph is that this graph tends to connect nodes within cluster of similar densities, but does not connect nodes from clusters of different densities [12]. Therefore, outliers are isolated and data clusters form graph components in mutual k_1-nearest neighbor graph. Moreover, the mutual k_1-nearest neighbor graph with n nodes ($k_1 \ll n$) is usually sparse, which has an advantage in computation. If the data has coordinates, we can use kd-tree to avoid $O(n^2)$ searching of nearest neighbors. The edge weights are inversely proportional to their Euclidean distances. However, it is possible that the mutual k_1-nearest neighbor graph is not connected so that we cannot apply random walk on the whole graph. One approach to make the graph connected is to find its minimum spanning tree and add the edges of the tree to the graph.

Then the graph Laplacian matrix L and its pseudoinverse L^+ are computed. After that the pairwise CDs between any two observations are calculated from L^+. Finally, the distance based outlier detection using CD with pruning technique proposed by Bay and Schwabacher [2] is used to find the top N outliers. The main idea of pruning is that an observation is not an outlier if its average distance to k_2 current nearest neighbors is less than the score of the weakest outlier among top N found so far. Using this approach, a large number of non-outliers can be pruned without carrying out a full database scan. The outlier score used is the average distance of an observation to its k_2 nearest neighbors. Suitable values for k_1 (for building the nearest neighbor graph) and k_2 (for estimating the outlier score) will be presented in the experiments.

4 Graph Component Sampling and Eigenspace Approximation

While CD is a robust measure for detecting both global and local outliers, its main drawback is its computational time. The direct computation of CD from L^+ is proportional to $O(n^3)$ which is not feasible for large graphs (n is the number of nodes). In this work, the graph components are sampled to reduce the graph size and then eigenspace approximation in [15] is applied to approximate the CD on the sampled graph.

4.1 Graph Sampling

An easy way to sample a graph is selecting nodes from it uniformly at random. However, sampling in this way can break the graph geometry structure and outliers may not be chosen in sampling. To resolve this, we propose a sampling strategy called component sampling. After creating the mutual k_1-nearest neighbor graph, the graph tends to have many connected components corresponding to different data clusters. Outliers are either isolated nodes or nodes in very small components. For nodes in normal components (we have a threshold to distinguish between normal and outlying components), they are uniformly sampled with the same ratio $p = 50k_1/n$, which is chosen from experimental results. For nodes in outlying components, we sample all of them. Then we rebuild a mutual k_1-nearest neighbor graph for the sampled data. Sampling in this way will maintain the geometry of the original graph and the relative densities of normal clusters. Outliers are also not sampled in this sampling strategy.

4.2 Eigenspace Approximation

Because the Laplacian matrix L $(n \times n)$ is symmetric and has rank $n-1$ [5], it can be decomposed as $L = VSV^{\mathrm{T}}$, where V is the matrix containing eigenvectors of L as columns and S is the diagonal matrix with the corresponding eigenvalues $\lambda_1 = 0 < \lambda_2 < ... < \lambda_n$ on the diagonal. Then $L^+ = VS^+V^{\mathrm{T}}$ where S^+ is the diagonal matrix with entries $\lambda_1^+ = 1/\lambda_2 > \lambda_2^+ = 1/\lambda_3 > ... > \lambda_{n-1}^+ = 1/\lambda_n > \lambda_n^+ = 0$. Equation 4 can be written as $c(i,j) = V_G(x_i - x_j)^{\mathrm{T}}(x_i - x_j)$ where $x_i = S^{+1/2}V^{\mathrm{T}}e_i$ [15]. Therefore, the CD between nodes on the graph can be viewed as the Euclidean distance in the space spanned by eigenvectors of the graph Laplacian matrix.

Denote $\tilde{V}$, $\tilde{S}$ as a matrix containing m largest eigenvectors of L^+ and its corresponding diagonal matrix, and $\tilde{x}_i = \tilde{S}^{+1/2}\tilde{V}^{\mathrm{T}}e_i$, the approximate CD is

$$\tilde{c}(i,j) = V_G(\tilde{x}_i - \tilde{x}_j)^{\mathrm{T}}(\tilde{x}_i - \tilde{x}_j), \tag{5}$$

The CD $c(i,j)$ in an n dimensional space is transformed to the CD $\tilde{c}(i,j)$ in an m dimensional space. Therefore, we just need to compute the m smallest eigenvectors with nonzero eigenvalues of L (i.e the largest eigenvectors of L^+) to approximate the CD. This approximation is bounded by $\|c(i,j) - \tilde{c}(i,j)\| \leq V_G \sum_{i=1}^{m} \lambda_i^+$ [15].

4.3 Algorithm

The proposed method is outlined in Algorithm 1. We create the sampled graph from the data using graph components sampling. Then the graph Laplacian L of the sampled graph and matrix $\tilde{V}$ (m smallest eigenvectors with nonzero eigenvalues of L) are computed. Since we use the pruning technique, we do not need to compute the approximate CD for all pairs of points. Instead, we compute it 'on demand' using the formula in equation 3 where $\tilde{l}_{ij}^+ = \sum_{k=1}^{m} \lambda_k^+ v_{ik} v_{jk}$, v_{jk} and v_{jk} are entries of matrix $\tilde{V}$.

4.4 The Complexity of the Algorithm

The k-nearest neighbor graph with n nodes is built in $O(nlogn)$ using kd-tree with the assumption that the dimensionality of the data is not very high. The average degree of each node is $O(k)$ ($k \ll n$). So the graph is sparse and thus finding connected components take $O(kn)$. After sampling, the size of graph is $O(n_s)$ ($n_s \ll n$). The standard method for eigen decomposition of L is $O(n_s^3)$. Since L is sparse, it would take $O(Nn_s) = O(kn_s^2)$ where N is the number of nonzeros. The computation of just the m smallest eigenvectors ($m < n_s$) is less expensive than that.

The typical distance based outlier detection takes $O(n_s^2)$ for the neighborhood search. Pruning can scale it nearly linear. We only need to compute the CDs $O(n_s)$ times each of which takes $O(m)$.

So the time needed for two steps is proportional to $O(nlogn + kn + kn_s^2 + mn_s) = O(nlogn)$ as $n_s \ll n$.

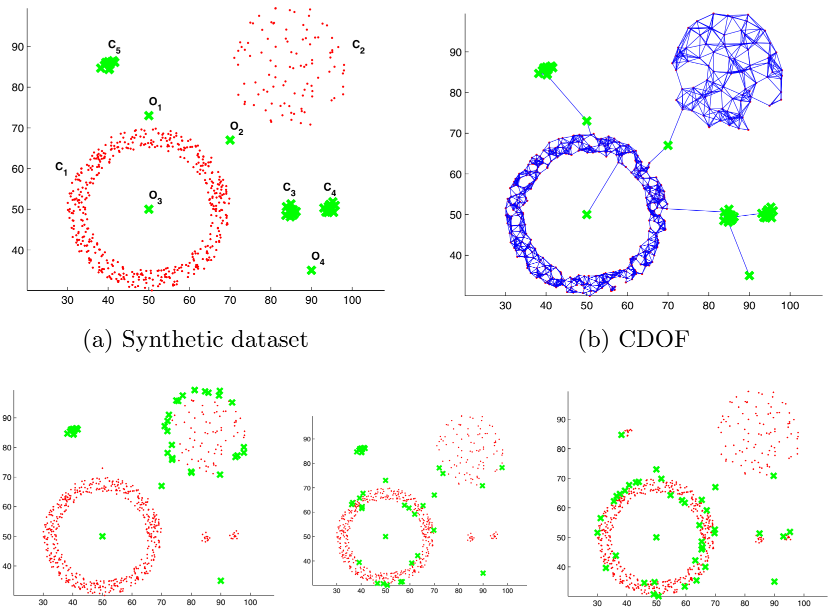
Algorithm 1. Commute Distance Based Outlier Detection with Graph Component Sampling and Eigenspace Approximation.

Input: Data matrix X, the numbers of nearest neighbors k_1 and k_2, the numbers of outliers to return N
Output: Top N outliers

```
Construct the mutual $k_1$-nearest neighbor graph from the dataset
Do the graph component sampling
Reconstruct the mutual $k_1$-nearest neighbor graph from sampled data
Compute the Laplacian matrix of the sampled graph and its $m$ smallest eigenvectors
Find top $N$ outliers using the CD based technique with pruning rule (using $k_2$)
Return top $N$ outliers
```

5 Experiments and Analysis

In this section, the effectiveness of CD as a measure for outlier detection is evaluated. Firstly, the ability of the distance based technique using CD (denoted as CDOF) in finding global, local outliers, and outlying clusters was tested in a synthetic dataset. The distance based technique using Euclidean distance [2] (denoted as EDOF), LOF [3], and OutRank [13] (denoted as ROF and the same graph of CDOF was used) were also used to compare with CDOF. Secondly, the effectiveness of CDOF was evaluated in a real dataset. Thirdly, we have shown that CDOF is more resistant to small perturbations to data than LOF. Finally the performance and effectiveness of approximate CDOF were evaluated. The experiments were conducted on a workstation with an 3GHz Intel Core2 Duo processor and 2GB of main memory in Windows XP.



(a) Synthetic dataset

(b) CDOF

(c) EDOF

(d) LOF

(e) ROF

Fig. 3. Comparison of the results of EDOF, LOF, ROF, and CDOF. CDOF can detect all global, local outliers, and outlying clusters effectively.

5.1 Synthetic Dataset

Figure 3a shows a 2-dimensional synthetic dataset. It contains one dense cluster of 500 observations (C_1) and one sparse cluster of 100 observations (C_2). Moreover, there are three small outlying clusters with 12 observations each (C_{3-5}) and four outliers (O_{1-4}). All the clusters were generated from a Normal distribution. O_2, O_3, O_4 are global outliers which are far away from other clusters. O_1 is a local outlier of dense cluster C_1.

In the following experiments for this dataset, the numbers of nearest neighbors are $k_1 = 10$ (for building the graph), $k_2 = 15$ (for estimating the outlier score. Since the size of outlying clusters is twelve, fifteen is a reasonable number to estimate the outlier scores), and the number of top outliers is $N = 40$ (the total observations in three outlying clusters and four outliers). The results are shown in Figure 3. The 'x' signs mark the top outliers found by each method. The figure shows that EDOF cannot detect local outlier O_1. Both EDOF and LOF cannot find two outlying cluster C_3 and C_4. The reason is those two clusters are near each other with similar densities and consequently for each point in the two clusters the average distance to its nearest neighbors is small and the relative density is similar to that of its neighbors. Moreover, ROF outlier score is actually the node probability distribution when the random walk is stationary

[13]. Therefore, it is d_{ii}/V_G [11], which is small for outliers[2]. Therefore, it cannot capture nodes in the outlying clusters where d_{ii} is large. For degree one outlying nodes, ROF and CDOF have similar scores. The result in Figure 3b shows that CDOF can identify all the outliers and outlying clusters. The key point is in CD, inter-cluster distance is significantly larger then intra-cluster distance even if the two clusters are near in the Euclidean distance.

5.2 Real Dataset

In this experiment, CDOF was used to find outliers in an NBA dataset. The dataset contains information of all the players in the famous basketball league in the US in year 1997-1998. There were 547 players and six attributes were used: position, points per game, rebounds per game, assists per game, steals per game and blocks per game. Point and assist reflect the offensive ability of a player while steal and block show how good a player is in defending. Rebound can be either offensive or defensive but total rebound is usually an indicator of defensive ability. The results are shown in Table 2 with the ranking and statistics of top five outliers. The table also shows the maximums, averages, and standard deviations for each attribute over all players.

Table 2. The outlying NBA players

Rank	Player	Position	Points	Rebounds	Assists	Steals	Blocks
1	Dikembe Mutombo	Center	13.43	11.37	1.00	0.41	3.38
2	Dennis Rodman	Forward	4.69	15.01	2.88	0.59	0.23
3	Michael Jordan	Guard	28.74	5.79	3.45	1.72	0.55
4	Shaquille O'neal	Center	28.32	11.35	2.37	0.65	2.40
5	Jayson Williams	Forward	12.88	13.58	1.03	0.69	0.75
	Max		28.74	15.01	10.54	2.59	3.65
	Average		7.69	3.39	1.78	0.70	0.40
	Standard deviation		5.65	2.55	1.77	0.48	0.53

Dikembe Mutombo was ranked as the top outlier. He had the second highest blocks (5.6 times of standard deviation away from mean), the highest rebounds for center players, and high points. It is rare to have good scores in three or more different statistics and he was one of the most productive players. Dennis Rodman and Michael Jordan took the second and third positions because of their highest scores in rebound and point (4.6 and 3.7 times of standard deviation away from mean, respectively). Dennis Rodman was a rare case because his points were quite low among high rebound players as well. The next was Shaquille O'Neal who had the second highest points and high rebounds. He was actually the best scoring center and is likely a local outlier among center players. Finally, Jayson Williams had the second highest rebounds. It is interesting to note that except for Dennis Rodman because of his bad behaviour in the league, the other four players were listed in that year as members of All-Stars team [1].

[2] This score has not been explicitly stated in the ROF paper.

5.3 Sensitivity to Data Perturbation

In this section LOF and CDOF were compared on their ability to handle 'noise' perturbations in data. Recall that $LOF(p)$ is the ratio between the average relative density of the nearest neighbors q of p over the relative density of p. $LOF(p)$ is high (i.e p is outlier) if p's neighborhood area is sparse and q's neighborhood area is dense. Suppose that noise is uniformly distributed in the data space, it is obvious that the noise will have more effect on outliers than points in clusters. The noise data can be neighbors of outliers and their neighborhood are also sparse. Thus the numerator in $LOF(p)$ formula where p is outlier reduces considerably while the denominator increases. As a result, LOFs of outliers may reduce significantly but they does not change much for points inside the clusters. Therefore, the relative rankings of data may not be preserved. On the other hand, uniform noise changes the nearest neighbor graph for CDOF in the way that degrees of outliers will increase but are still much smaller than degree of points inside the clusters. Thus there will still be a big difference between inter-cluster and intra-cluster CDs. That maintains the higher scores for outliers than the points inside the cluster.

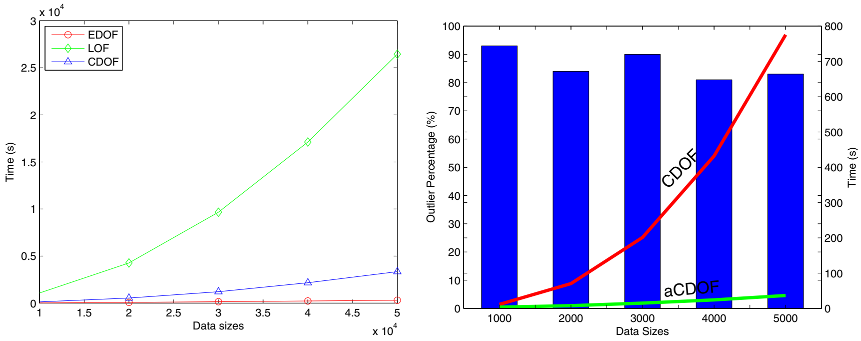
To show this, we randomly added 10% noise from a uniform distribution to the synthetic dataset in Section 5.1 and applied LOF and CDOF in the new dataset. Then noise data was removed from the ranking. Two criteria were used to compare two methods: Spearman rank test for the ranking of the whole dataset and the similarity between the sets of top outliers before and after adding noise. The results were averaged over ten trials. Spearman rank test in LOF was 0.01 while the it was 0.48 for CDOF. It shows the relative ranking by LOF changes significantly due to noise effect. After adding noise, there were 62% of the original outliers still in the top outlier list for LOF while it was 92% for CDOF. CDOF is less sensitive as it combines local and global views of the data.

Since noise is a kind of outlier, we cannot distinguish between outliers and noise but the preliminary results for noise effect show that the proposed method is more resistance to noise than the local outlier detection method.

5.4 Performances of the Proposed Method

In the following experiments, we compared the performances of EDOF, LOF, and approximate CDOF mentioned in Section 4. The experiment was performed using five synthetic datasets, each of which contained different clusters generated from Normal distributions and a number of random points. The number of clusters, the sizes, and the locations of the clusters were also chosen randomly. The results are shown in Figure 4a where the horizontal axis represents the dataset sizes in the ascending order and the vertical axis is the corresponding computational time. The result of approximate CDOF was averaged over ten trials of data sampling. It is shown that approximate CDOF is faster than LOF and slower than EDOF. This reflects the complexities of $O(n)$, $O(nlogn)$, and $O(n^2)$ for EDOF, approximate CDOF, and LOF, respectively.

In order to validate the effectiveness of approximate CD, we used CD and approximate CD to find outliers in five synthetic datasets generated in the same



(a) Performances of EDOF, LOF, and approximate CDOF

(b) Effectiveness of approximate CDOF

Fig. 4. Performances of the method using approximate commute distance

way as the experiment noted above with smaller sizes due to the high computation of CDOF. The results were averaged over ten trials. The results in Figure 4b shows approximate CDOF (aCDOF) is much faster than CDOF but still preserves a high percentage (86.2% on average) of top outliers found by CDOF.

5.5 Impact of Parameter k

In this section, we investigate how the number of nearest neighbors affects CDOF. Denote k_{min} as the maximum number of nodes that a cluster is an outlying cluster and k_{max} as the minimum number of nodes that a cluster is a normal cluster. There are two situations. If we choose the number of nearest neighbors $k_2 < k_{min}$, nodes in an outlying cluster do not have neighbors outside the cluster. As a result, their outlier factors are small and we will miss them as outliers. On the other hand, if we choose $k_2 > k_{max}$, nodes in a normal cluster have neighbors outside the cluster. And it is possible that some nodes in the cluster will be falsely recognized as outliers. The value of k_{min} and k_{max} can be considered as the lower and upper bounds for the number of nearest neighbors. They can be different depending on the application domains. In the experiment in Section 5.1, we chose $k_2 = 15$, which is just greater than the sizes of all outlying clusters (i.e 12) and is less than the size of the smallest normal cluster (i.e 100). The same result can be obtained with $15 < k_2 < 100$ but it requires longer computational time. k_2 is also chosen as a threshold to distinguish between normal and outlying clusters.

Note that k_2 mentioned in this section is the number of nearest neighbors for estimating the outlier scores. For building mutual k_1-nearest neighbor graph, if k_1 is too small, the graph is very sparse and may not represent the dataset densities properly. In the experiment in Section 5.1, if $k_1 = 5$, the algorithm misclassifies some nodes in the smaller normal cluster as outliers. If k_1 is too large, the graph tend to connect together clusters whose sizes are less than k_1

and are close to each other. Then some outlying clusters may not be detected if they connect to each other and form a normal cluster. $k_1 = 10$ is found suitable for many synthetic and real datasets.

6 Conclusions

We have proposed a method for outlier detection using 'commute distance' as a metric to capture global, local outliers, and outlying clusters. The CD captures both distances between observations and their local neighborhood densities. We observed and proved a property of CD which is useful in capturing local neighborhood density. The experiments have shown the effectiveness of the proposed method in both synthetic and real datasets. Moreover, graph component sampling and eigenspace approximation used to approximate CD and the use of pruning rule can accelerate the algorithm significantly while still maintaining the accuracy of the detection. Furthermore preliminary experiments suggest that CDOF is less sensitive to perturbations in data than other measures.

Acknowledgement

The authors of this paper acknowledge the financial support of the Capital Markets CRC.

References

1. Database basketball, `http://www.databasebasketball.com`
2. Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: KDD '03: Proc. of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 29–38. ACM, New York (2003)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: Chen, W., Naughton, J.F., Bernstein, P.A. (eds.) Proc. of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, May 16-18, pp. 93–104. ACM, New York (2000)
4. Chandola, V., Banerjee, A., Kumar, V.: Outlier detection: A survey. Tech. Rep. TR 07-017, Department of Computer Science and Engineering, University of Minnesota, Twin Cities (2007)
5. Chung, F.: Spectral Graph Theory. In: Conference Board of the Mathematical Sciences, Washington. CBMS Regional Conference Series, vol. 92 (1997)
6. Fouss, F., Renders, J.M.: Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Transaction on Knowledge and Data Engineering 19(3), 355–369 (2007)
7. Hawkins, D.: Identification of Outliers. Chapman and Hall, London (1980)
8. Khoa, N.L.D., Chawla, S.: Unifying global and local outlier detection using commute time distance. Tech. Rep. 638, School of IT, University of Sydney (2009)
9. Klein, D.J., Randic, M.: Resistance distance. Journal of Mathematical Chemistry 12, 81–95 (1993), `http://dx.doi.org/10.1007/BF01164627`

10. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: The 24rd International Conference on Very Large Data Bases, pp. 392–403 (1998)
11. Lovász, L.: Random walks on graphs: a survey. Combinatorics, Paul Erdös is Eighty 2, 1–46 (1993)
12. Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
13. Moonesinghe, H.D.K., Tan, P.N.: Outrank: a graph-based outlier detection framework using random walk. International Journal on Artificial Intelligence Tools 17(1), 19–36 (2008)
14. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. John Wiley and Sons, Chichester (2003)
15. Saerens, M., Fouss, F., Yen, L., Dupont, P.: The principal components analysis of a graph, and its relationships to spectral clustering. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 371–383. Springer, Heidelberg (2004)
16. Sun, P.: Outlier Detection In High Dimensional, Spatial And Sequential Data Sets. Ph.D. thesis, The University of Sydney (2006)