Automatic PAM Clustering Algorithm for Outlier Detection

Dajiang Lei
College of Computer, Chongqing University, Chongqing, 400030, China
Email: leidj@cqupt.edu.cn

Qingsheng Zhu*
College of Computer, Chongqing University, Chongqing, 400030, China
*Corresponding Author, Email: qszhu@cqu.edu.cn

Jun Chen, Hai Lin and Peng Yang
College of Computer, Chongqing University, Chongqing, 400030, China
Email: chenjun@cqupt.edu.cn, woodosean@yahoo.com.cn, llylab@21cn.com

Abstract—**In this paper, we propose an automatic PAM (Partition Around Medoids) clustering algorithm for outlier detection. The proposed methodology comprises two phases, clustering and finding outlying score. During clustering phase we automatically determine the number of clusters by combining PAM clustering algorithm and a specific cluster validation metric, which is vital to find a clustering solution that best fits the given data set, especially for PAM clustering algorithm. During finding outlier scores phase we decide outlying score of data instance corresponding to the cluster structure. Experiments on different datasets show that the proposed algorithm has higher detection rate go with lower false alarm rate comparing with the state of art outlier detection techniques, and it can be an effective solution for detecting outliers.**

Index Terms—**outlier detection, PAM clustering algorithm, subtractive clustering, cluster validation**

I. Introduction

An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism [1]. It is concerned with discovering the exceptional behavior of certain objects [2]. Many data mining algorithms try to minimize the influence of outliers or eliminate them all together. However, it may result in the loss of important hidden information. Some data mining applications are focused on outlier detection, and it is the essential result of a data analysis. For example, while detecting fraudulent credit card transactions, the outliers are typical examples that may indicate fraudulent activity, and outlier detection is mainly process in the entire data mining [3], [4].

In the recent decades, many the state of art outlier detection techniques have been proposed, which can be mainly classified into several categories: distribution-based [5], [6], [7], depth-based [8], distance-based [9], [10], density-based [11], cluster-based [12], [13]. Distribution-based methods are mostly used in early studies, which is a statistic method. However, a large number of tests are often required to decide which distribution model fits the arbitrary dataset best. Fitting the data with standard distributions is costly, and may not produce satisfactory results. The second category of outlier studies in statistics is depth-based. Each data object is represented as a point in a k-d space, and is assigned a depth. With respect to outlier detection, outliers are more likely to be data objects with smaller depths. However, in practice, depth-based approaches become inefficient for large datasets for $k \geqslant 4$. Hence, many other categories of methods are proposed. In [9], Knorr and Ng firstly proposed the notion of distance-based outliers which is a non-local approach and can not find local outliers in complex structure data sets. Then, Breunig et al. propose density-based local outliers detection method (LOF) based on the distance of a point from its k nearest neighborhood and declare the top n points in this ranking to be outliers. Furthermore, many clustering algorithms, especially those developed in the context of KDD were extended to have capable of handling exceptions. The ordinary clustering based outlier detection methods find outliers as a side-product of clustering algorithm, which regard outliers as objects not located in clusters of dataset.

Even though clustering and anomaly detection appear to be fundamentally different from each other, there are numerous studies on clustering-based outlier detection methods. Many data-mining algorithms in literature detect outliers as a by-product of clustering algorithms themselves, which define outliers as points that do not lie in or located far apart from any clusters. Actually no clustering algorithm can precisely classify every data instance and some special data points in a certain cluster may be outliers. In this paper, we present one improved cluster based local outlier factor (***CBLOF***) [12] to tackle this problem. For clustering based outlier detection algorithms, the number of clusters is needed to choose. However, for unknown datasets, we choose the number of clusters arbitrarily and it would decrease the performance of the algorithm. In this paper, we combine PAM clustering algorithm with a cluster validation metric to propose an automatic PAM clustering algorithms for dividing data instances into k clusters. To our proposed algorithm, the accurate cluster structure (or approximate to the actual nature of data set) is crucial to improve detecting ratio and false alarm ration in the context of outlier detection.

The rest of this paper is organized as follows. Section 2 introduces subtractive clustering algorithm for estimating approximate number of clusters. Section 3 introduces the definition of the cluster validation metric used in our proposed algorithm and propose automatic PAM clustering algorithm. Section 4 proposes the improved ***CBLOF*** and outlier detection algorithm based automatic PAM clustering algorithm. Section 5 elaborates the experiments for demonstrating the effectiveness and efficiency of our proposed method. We conclude the paper in Section 6.

II. SUBSTRATIVE CLUSTERING

The Subtractive Clustering (***SC***) method is adopted in this paper to estimate approximate number of clusters of a given data set. Suppose that we don't have a clear idea how many clusters there should be for a given data set. Subtractive clustering [14] is a fast, one-pass algorithm for estimating the number of clusters and the cluster centers in a set of data. The subtractive clustering algorithm is described as follows.

Consider a group of n data points $\{x_1,x_2,...,x_N\}$, where x_i is a data point, denoted as a vector in the feature space. Without loss of generality, we assume that the feature space is normalized so that all data are bounded by a unit hypercube. Firstly, we consider each data point as a potential cluster center and define a measure of the point to serve as a cluster center. The potential of x_i, denoted as P_i, is computed as by Eq. 1.

$$P_i = \sum_{l=1}^{N} \exp(-\alpha \| x_i - x_l \|^2) \; . \qquad \textbf{(1)}$$

Where $\alpha=4/r_a^2$ and $r_a>0$ denotes the neighborhood radius for each cluster center. A data point with many neighboring data points will have a high potential value and the points outside r_a have little in influence on its potential. After calculating potential for each point, the one with the highest potential value will be selected as the first cluster center. Then the potential of each point is reduced to avoid closely spaced clusters. Selecting centers and revising potential is carried out iteratively until a stopping criteria satisfied. The SC algorithm can be described as follows.

Algorithm: Estimate k number of clusters based on SC algorithms (***SC_EstimateK***)
Input: A data set $D=\{x_1,x_2,...,x_N\}$
Output: k number of clusters
Step 1: Calculate the potential P_i for each point according to Eq. 1, $1\leq i\leq N$;
Step 2: Set the number of cluster center $k=1$ and select the data point with the highest potential value as the first cluster center. Let x_k^* be the location of the point and $P(x_k^*)$, its corresponding potential value;
Step 3: Revise the potential of each data point according to Eq. 2

$$P_i = P_i - P(x_k^*) \exp\left(-\beta \left\| x_i - x_k^* \right\|^2\right) . \qquad \textbf{(2)}$$

Step 4: If $\boldsymbol{max}_i(P_i)\leq\varepsilon * P(x_k^*)$, terminate the algorithm, return k number of clusters; otherwise, set $k=k+1$ and find the data point with the highest potential value. Let x_k^* be the location of the point and $P(x_k^*)$, its corresponding potential value, go to Step 3.

Note that in Eq. 2, $\beta=4/r_b^2$ and $r_b>0$ presents the radius of the neighborhood for which significant potential revising will occur. To avoid obtaining closely spaced cluster centers, r_b is chosen to be greater than r_a. Typically, $r_b=1.5r_a$. In step 4, the parameter ε should be selected within (0,1). If ε is selected to be close to 0, a large number of cluster centers will be generated. On the contrary, a value of ε close to 1 will render fewer cluster centers.

In this paper, we utilize subtractive clustering to estimate the approximate number k of clusters, which is a reference number of clusters in the next section. For it is approximate estimation, we can set the parameters ε, r_a and r_b to 0.5, 0.5 and 0.75 as default respectively.

III. CLUSTER VALIDATION METRIC AND AUTOMATIC PAM CLUSTERING ALGORITHM

A. Cluster Validation Metric

For PAM clustering algorithm, choosing the number of clusters (k) is crucial to the performance of clustering. With variant given the number of clusters, different clustering results may be acquired. In cluster analysis, we find the partitioning that best fits the underlying data through evaluating clustering results. The validity indices are simply and effective methodology of measuring the quality of clustering results. There are two kinds of validity indices: external indices and internal indices. In order to choose the optimal k value for PAM clustering algorithm, we choose an internal validity index. The principles of some widely-used internal indices for k-estimation and clustering quality evaluation are [15]: Silhouette index [16], Davies-Bouldin index, Calinski-Harabasz index, Dunn index, RMSSTD index. One may choose a validity index to estimate an optimal k value, where the optimal clustering solution is found from a series of clustering solutions under different k values.

We adapt Silhouette index as cluster validation metric for its simplicity. Silhouette index is a composite index reflecting the compactness and separation of the clusters, and can be applied to different distance metrics. For data instance x_i, its silhouette index $Sil(x_i)$ is definite as:

$$Sil(x_i)=(b(x_i)-a(x_i))/max\{a(x_i),b(x_i)\} \qquad \textbf{(3)}$$

where $a(x_i)$ is the average distance of data instance x_i to other data instances in the same cluster, $b(x_i)$ is the average distance of data instance x_i to instances in its nearest neighbor cluster. The average of $Sil(x_i)$ across all data instances reflects the overall quality of the clustering result. A larger averaged Silhouette index indicates a better overall quality of the clustering result.

B. Automatic PAM Clustering Algorithm

PAM clustering [17], standing for “partition around medoids”, is a well-known partitioning method. Objects are classified as belonging to one of K groups, K computed by the algorithm ***SC_EstimateK*** in section 2. Compared to the well-known k-means algorithm, PAM has the following features. Firstly, it operates on the dissimilarity matrix of the given data set or when it is presented with an $n\times m$ data matrix and the algorithm first computes a dissimilarity matrix. Secondly, it is more robust, because it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances. Finally, with the Silhouette index, it allows the user to select the optimal number of clusters.

In many clustering problems, one is interested in the characterization of the clusters by means of typical objects, which represent the various structural features of objects under investigation. The algorithm PAM first computes k representative objects, called medoids. A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal. In the classification literature, such representative objects are called centrotypes. After finding the set of medoids, each object of the data set is assigned to the nearest medoid. That is, object x_i is put into a certain cluster C_I, when medoid cm_I is nearer than any other medoid cm_w, denoted as follows:

$$d(x_i, cm_I) \le d(x_i, cm_k) \qquad \textbf{(4)}$$

where $k=1,...,K$. The K representative objects should minimize the objective function J, which is the sum of the dissimilarities of all objects to their nearest medoid, denoted as follows:

$$J = \sum_{k=1}^{K}\sum_{i=1}^{N} r_{ik} d(x_i, cm_k) \qquad \textbf{(5)}$$

where r_{ik} denotes 1 when x_i belonging to the cluster C_k, 0 when x_i not belonging to the cluster C_k; N denotes the number of data instances in data set; K denotes K clusters in data set; cm_k represents k^{th} cluster medoid.

The goal of the algorithm is to minimize the average dissimilarity of objects to their closest selected object. Equivalently, we can minimize the sum of the dissimilarities between object and their closest selected object.

The algorithm proceeds in two steps:

BUILD-step: This step sequentially selects k “centrally located” objects, to be used as initial medoids.

SWAP-step: If the objective function can be reduced by interchanging (swapping) a selected object with an unselected object, then the swap is carried out. This is continued till the objective function can no longer be decreased. In this paper, we refer to the PAM clustering algorithm as ***PAMClust*** in our proposed algorithm by brevity. We describe the algorithm as follows:

```
Algorithm: k-means clustering algorithm (PAMClust)
Input: A data set $D=\{x_1,x_2,...,x_N\}$, the number of clusters $K$;
Output: Clusters $\{C_1,C_2,\ldots,C_K\}$
Step 1. Initialize: randomly select the $K$ cluster centroids
        $CM=\{cm_1, cm_2,...,cm_K\}$;
Step 2. Associate each data point $x_i\in D$-$CM$ to the closet
        mediod $cm_k$ according to Eq. 4, and each centroid
        and associated data point form clusters, denoted as
        $\{C_1,C_2,\ldots,C_K\}$;
Step 3. For each medoid $cm_k$
        For each non-mediod data point $o\in D$-$CM$
        Swap $cm_k$ and $o$ and compute the function $J$ according to Eq. 5. If the function $J$ decrease, repalce $cm_k$ with $o$, else keep $cm_k$ as cluster mediod.
Step 4. Repeat Steps 2 and 3 until there is no mediods
        updating, viz. the function $J$ can not be decreased;
Step 5. Return $K$ clusters $\{C_1,C_2,\ldots,C_K\}$
```

Firstly, we acquire the proximate number of clusters by the algorithm ***SC_EstimateK*** in section 2. Then we run the algorithm ***PAMClust*** to find optimal cluster according to the Silhouette index value. The algorithm of finding optimal cluster is as follows:

```
Algorithm: Automatic PAM clustering (APAMClust) algorithm
Input: A data set $D=\{x_1,x_2,...,x_N\}$;
Output: Optimal clusters $Clust_{opt}=\{C_1,C_2,\ldots,C_{opt}\}$;
Step 1. $K$=SC_EstimateK ($D$)
Step 2. for i=$K$-λ:$K$+λ do loop
Step 3.   $Clust[i]$=PAMClust($D$,$i$)
Step 4.   $S[i]=Sil(Clust[i])$
Step 5.   $Array$[i]=$\{S[i], Clust[i]\}$
Step 6. end_loop
Step 7. Find maximal Silhouette index value and corresponding cluster in $Array$, and the found index is
        denoted as $maxindx$.
Step 8. Return $Clust_{opt}=Clust[maxindx]$;
```

In step 2, ***PAMClust*** algorithm run 2*λ times loop and produce the clusters of data set and corresponding Silhouette index value for every time, where λ is a desired value and can be a constant or adjusts to K value. In this paper, for the sake of simplicity, λ is set to $K/2$.

IV. Improved CBLOF and Outlier Detection Algorithm

In this section, we take use of the local outlier factor to identify the top n outliers in data set by analyzing the clustering structure obtained in section 3. It is reasonable to define the outliers based on the structure of clusters and identify those objects that do not lie in any large clusters as outliers. As the large clusters are often dominant in the data set, the outliers are less probably included in them. Before presenting our outlier detection algorithm, we describe the following definition about the local outlier factor [12].

Definition 1. (large and small cluster) Suppose that $C=\{C_1, \ldots, C_k\}$ is the set of clusters in the sequence that $|C_1|\ge \ldots \ge|C_k|$, where $|C_i|$ denotes the number of objects in C_i (i=1,…,k) and k is the number of clusters. Given two numeric parameters α and β, we define b as the boundary of large and small cluster if one of the following formulas holds.

$$|C_1|+\ldots+|C_b|\geq|D|*\alpha \qquad (6)$$

$$|C_b|/|C_{b+1}|\geq\beta \qquad (7)$$

Then, the set of large cluster is defined as $LC=\{C_i|i\leq b\}$ and the set of small cluster is defined as $SC=\{C_j|j>b\}$.

Definition 1 provides quantitative measure to distinguish large and small clusters. Formula 6 considers the fact that most data points in the data set are not outliers. Therefore, clusters that hold a large portion of data points should be treated as large clusters. Formula 7 considers the fact that large and small clusters should have significant differences in size.

To study clustering algorithm across different data set with outliers, we find that fact the point outliers will be classified as the nearest large clusters, since point outlier can not be clustered as one cluster. Therefore, the design of a new local outlier factor is desired in this situation. We designed an improved cluster-based local outlier factor based on the definition in the previous studies [12], [18].

Definition 2. (Cluster-Based Local Outlier Factor) Suppose $C=\{C_1,\ldots,C_k\}$ is the set of clusters in the sequence that $|C_1|\geq\ldots\geq|C_k|$ and the meanings of α, β, b, LC and SC are the same as they are formalized in Definition 1. For any record t, the cluster-based local outlier factor of t is defined as:

$$CBLOF(t)=dist(t,C_j)\big/|C_i|$$

$$\ldots\text{where } t\in C_i,\ C_i\in SC \text{ and } C_j\in LC$$

$$CBLOF(t)=dist(t,C_j)\bigg/\left(\sum_{C_j\in LC}|C_j|\bigg/|LC|\right) \qquad (8)$$

$$\ldots\text{where } t\in C_j, \text{ and } C_j\in LC$$

where $dist(t,C_j)$ is the Euclidean distance between t and the center of C_j, $|C_j|$ is the number of objects in C_j, $|LC|$ is the number of clusters in the large clusters set. ***CBLOF***(t) is simply to be calculated because we just need know the number of objects in certain large clusters as well as how far it close to t.

If $t\in SC$, C_j denotes the nearest large cluster which is neighboring to t. In this case, $\forall t\in SC$, C_j is almost identical. Thus, the object t which is more far away from the center of C_j will get a larger ***CBLOF***. Otherwise, if $t\in LC$, C_j denotes the large cluster which contains t. Assume that two objects are respectively within a large cluster and a small cluster, they have equal distance from their corresponding center of clusters. In this situation, the object t belong to the small cluster will get a larger ***CBLOF*** because we consider that data points in small clusters are outliers and provide more meaningful outlying information than point outliers in large clusters. In addition, in a sense, we can regard point outliers as noises and they provide less underlying outlying information about the data set.

Based on the clusters obtained in section 3, we can give an outlier detection algorithm.

Algorithm: Automatic PAM clustering algorithm for outlier detection (***APCOD***)

```
Input: A data set $D=\{x_1,x_2,...,x_N\}$; parameters $n$, $\alpha$ and $\beta$
Output: The top $n$ outliers with largest values of $CBLOF$ in data set
Step 1. Utilize APAMClust algorithm to produce optimal
        cluster $Clust_{opt}=\{C_1,C_2,\ldots,C_{opt}\}$;
Step 2. Get $LC$ and $SC$ according to Definition 2 based on
        parameters $\alpha$ and $\beta$;
Step 3. For each object $t$ in the data set, calculate
        $CBLOF(t)$ according to Eq. 8;
Step 4. Return the top $n$ outliers with largest CBLOF
```

V. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of our proposed algorithms. All algorithms are implemented through MATLAB on Pentium(R) Dual-Core CPU 2.0G PC with 2.0G main memory. We utilize the synthetic data set with outliers and real life data sets obtained from the UCI machine learning repository [21] to compare our algorithm against the original CBLOF algorithm (ORCBLOF) [18], KNN [10] and LOF [11] for outlier detection. For the results of KNN and LOF algorithm, we only present the best overall performance with the optimal number of nearest neighbors. For ORCBLOF algorithm, we choose the same parameters with our proposed algorithm.

We evaluate outlier detection techniques using two categorical different metrics. The first metric consists of detection rate (denoted as DR) and false alarm rate (denoted as FR) [19], which is the most commonly used in detection systems, defined as follows:

$$DR=|AO|/|CO| \qquad (9)$$

$$FR=(|BO|-|AO|)/(|DN|-|CO|) \qquad (10)$$

where $|AO|$ is the number of true outliers in the detected top n outliers, $|BO|$ is the number of the detected top n outliers and $|CO|$ is the number of true outliers in the entire data set; while $|DN|$ is the number of all objects in the data set.

One drawback of the above type metric is that it is highly dependent on the choice of n, which is used for top n outliers. An outlier detection technique might show 100% DR for a particular value of n but show 50% accuracy for 2n. To overcome this drawback we also use the following evaluation metric. The second categorical metric used to evaluate the outlier detection techniques is to obtain the ROC curve [20] obtained by varying n from 1 to $|DN|$. The advantage of ROC curve is that it is not dependent on the choice of n. The higher area under of the ROC curve (AUC) indicates higher performance corresponding competing outlier detection algorithm.

A. Effectiveness of Detecting Outliers on Synthetic Data Set

Figure 1 shows a synthetic 2-dimentional data set with three large clusters, two small clusters and four point outliers. Figure 2 shows estimation number of cluster corresponding to the data set depicted in Figure 1. Optimal number of cluster is indicated by a square symbol in Figure 2. Intuitively, all of 22 objects in the small clusters and 4 point outliers denoted with "*" (O_1 to O_6) can be regards as outliers. We apply ***APCOD***, ORCBLOF, KNN and LOF to find the top 26 outliers in the dataset. To KNN and LOF outlier detection algorithms, we choose 20 and 15 as optimal number of k nearest neighbors respectively. ***APCOD*** can successfully find the desirable outliers because it utilizes ***APAMClust*** algorithm to obtain stable clusters and effectively defines the local outlier factor in both large and small clusters for every object. ORCBLOF is not able to find outliers which are near to enormous clusters. According to the definition in [18], the skewed enormous clusters would decrease CBLOF value of outliers near to enormous clusters. For example, O_1 and O_2 will be assigned smaller CBLOF values relative to normal objects in C_3 and be treated as normal objects. However, LOF has some difficulty to distinguish outliers since the density of C_3 and O_6 are similar. So it identifies some normal objects in C_3 as outliers rather than detects the outlier cluster O_6. On the other hand, KNN fails to identify outlier cluster O_5 because it is much closed to C_3. In table I, we present the performance of competing outlier detection algorithms adapted in this paper.

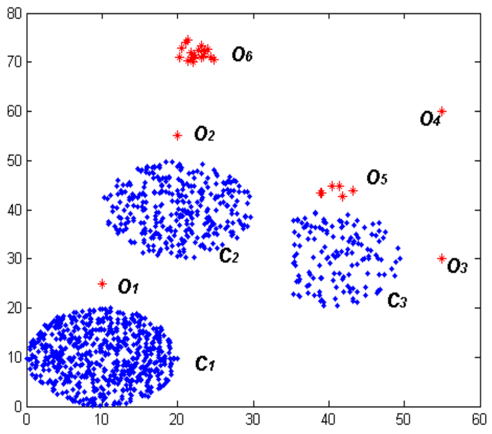

Figure 1. 2-dimensional data set with outliers

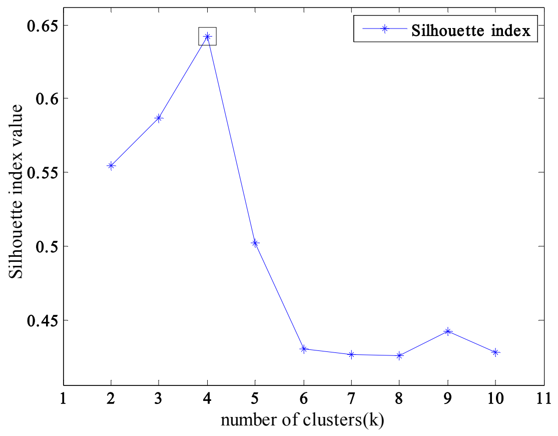

Figure 2. estimation number of cluster for synthetic 2-dimentional data

TABLE I.
RESULTS OF OUTLIER DETECTION ALGORITHM ON SYNTHETIC DATA SET.

	KNN	LOF	ORCBLOF	***APCOD***
DR	0.75	0.85	0.95	1
FR	0.04	0.02	0.009	0

B. Effectiveness of Detecting Outliers on Real Life Data Set

To verify the performance of our proposed algorithm on practical application domain, we apply all algorithms on several different real life data sets available at the UCI Machine Learning Repository. The details about the selected data sets are summarized in Table 2. For the sake of simplicity, we choose all data sets with purely continuous attributes. For the mixture of continuous and categorical attributes, a possible way is to compute the similarity for continuous and categorical attributes separately, and then do a weighted aggregation.

The "No. attributes" in table II denotes all attributes except the class attribute. Note that the "No. outliers" in the table is the total of objects in the small clusters. Each data set contains labeled instances belonging to multiple classes. We identify the last class or last two classes as the outlier class, and rest of the classes were grouped together and called normal. In order to produce the small clusters, we remove most objects within the last class and set the proportion of outliers to be not greater than 10% of total instances in data set. For Lymph data set, we choose two class containing least instances as outlier classes. In the experiments, the parameters α and β set to 0.75 and 5 respectively.

Tables III summarize the *DR*, *FR* results on the real life data sets. Figure 4 indicates performance of competing algorithms on Iris data set. Figure 5 indicates performance of competing algorithms on Lymph data set. Figure 6 indicates performance of competing algorithms on Optical data set. From table III, we can find that the detection ratio of all competing algorithms decreases along with higher dimensionality of data set. We can deduce that all competing outlier detection algorithm in this paper are more or less unsuitable for high-dimensional data due to the notorious "curse of dimensionality". This is primary drawback about our proposed algorithm. In order to improve performance, we recommend SNN distance as distance function to calculate similarity between two data instances. Though all competing algorithm suffer from the notorious "curse of dimensionality", our proposed algorithm, ***APCOD***, still performs munch better than ORCBLOF, LOF and KNN with a lower false alarm rate and higher AUC. While running on Optical data set, KNN performs extremely poorly because the data set is high dimensional and very spare and in turn identify the outliers simply based on Euclidean distance is not feasible. Though ***APCOD*** performs moderately on Optical data set, it still outperforms both LOF and KNN.

TABLE II.
CHARACTERISTICS OF REAL LIFE DATA SETS.

	No. attributes	No. clusters	No. outliers	No. instances
Iris	4	3	10	110
Lymph	18	4	6	148
Optical(training)	64	10	60	3121

TABLE III.
RESULTS OF OUTLIER DETECTION ALGORITHM ON REAL LIFE DATA SET.

	Iris		Lymph		Optical(training)	
	DR	*FR*	*DR*	*FR*	*DR*	*FR*
KNN	0.7500	0.06	0.5000	0.0211	0.0132	0.0279
LOF	0.7000	0.06	0.5000	0.0211	0.0789	0.0130
ORCBLOF	0.7000	0.08	0.6667	0.0141	0.3421	0.0186
APCOD	0.8000	0.04	0.8333	0.007	0.3684	0.0179

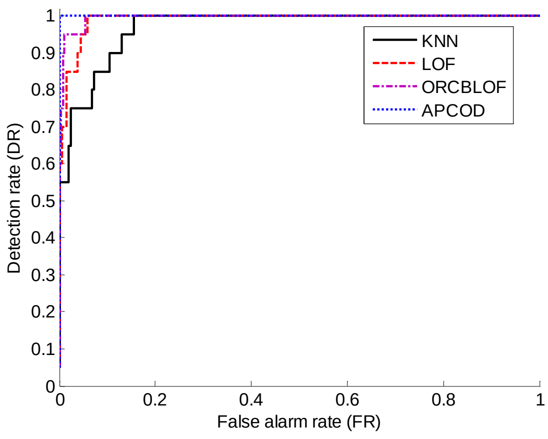

Figure 3. ROC of outlier detection algorithm on synthetic data set

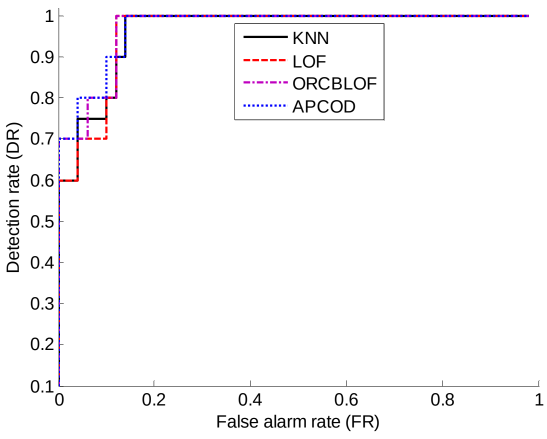

Figure 4. ROC of outlier detection algorithm on Iris

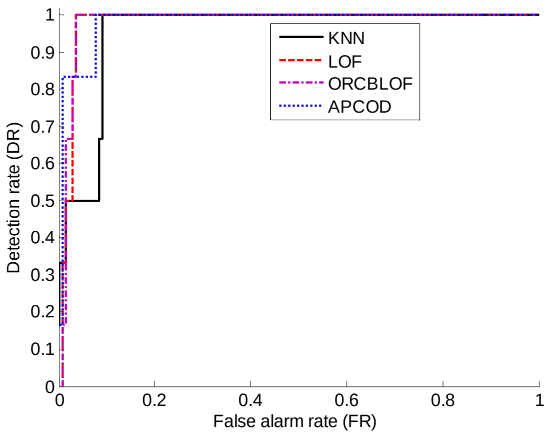

Figure 5. ROC of outlier detection algorithm on Lymph

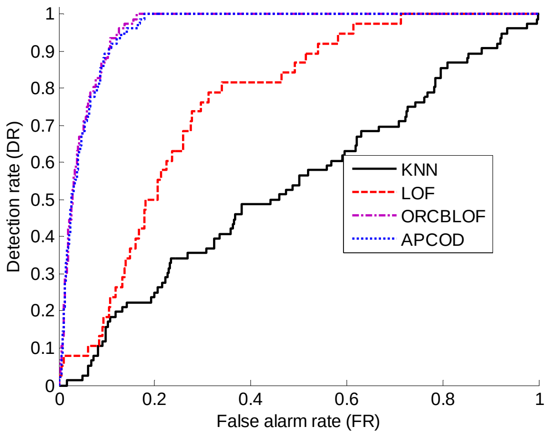

Figure 6. ROC of outlier detection algorithm on Optical

VI. CONCLUSION

In this paper, an efficient automatic PAM clustering outlier detection (***APCOD***) algorithm is proposed. The algorithm firstly estimates the number of clusters in data set by subtractive clustering without expert knowledge about the data set. Then it fixes the optimal number of clusters by combining Silhouette index and PAM clustering and the optimal number of clusters represents the nature of the underlying data set. Finally, we present the definition of improved ***CBLOF*** and propose our outlier detection algorithm. Our proposed algorithm treats the small clusters and points far away from the large clusters as outliers and can efficiently identify the top n outliers by defining improved local outlier factor (***CBLOF***) for each instances in all clusters. Experimental results demonstrate that our proposed algorithm can automatically produce desirable clusters and has superior performance for outlier detection with lower false alarm rate compared against ORCBLOF, LOF and KNN.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China ("Research on Structure Outlier Analysis and its Paraphrase Space Modeling", Grant No. 61073058).

REFERENCES

[1] D. M. Hawkins, Identification of outliers. New York: Chapman and Hall, USA, 1980.

[2] P. Yang and Q. S. Zhu, "Finding key attribute subset in dataset for outlier detection," *Knowledge-Based Systems*, vol. 24, pp. 269-274, Mar 2011.

[3] Y. Dianmin, W. Xiaodan, W. Yunfeng, L. Yue, and C. Chao-Hsien, "A Survey of Outlier Detection Methods in Network Anomaly Identification," *Computer Journal*, vol. 54, pp. 570-588, Apr 2011.

[4] K. Bhaduri, M. D. Stefanski, and A. N. Srivastava, "Privacy-Preserving Outlier Detection Through Random Nonlinear Data Distortion," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 41, pp. 260-272, Feb 2011.

[5] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," *Knowledge and Information Systems*, vol. 26, pp. 309-336, Feb 2011.

[6] S. G. Marroquin-Guerra, F. Velasco-Tapia, and L. Diaz-Gonzalez, "Statistical evaluation of geochemical reference materials from the Centre de Recherches Petrographiques et Geochimiques (France) by applying a schema for the detection and elimination of discordant outlier values," *Revista Mexicana De Ciencias Geologicas*, vol. 26, pp. 530-542, Aug 2009.

[7] Y. Zhang, S. Yang, and Y. Wang, "LDBOD: A novel local distribution based outlier detector," *Pattern Recogn. Lett.*, vol. 29, pp. 967-976, 2008.

[8] I. Ruts and P. J. Rousseeuw, "Computing depth contours of bivariate point clouds," *Computational Statistics & Data Analysis*, vol. 23, pp. 153-168, 1996.

[9] E. M. Knorr and R. T. Ng, "Algorithms for Mining Distance-Based Outliers in Large Datasets," *presented at the Proceedings of the 24rd International Conference on Very Large Data Bases*, New York, USA, 1998.

[10] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *Sigmod Record*, vol. 29, pp. 427-438, Jun 2000.

[11] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *Sigmod Record*, vol. 29, pp. 93-104, Jun 2000.

[12] Z. Y. He, X. F. Xu, and S. C. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, pp. 1641-1650, Jun 2003.

[13] M. F. Jaing, S. S. Tseng, and C. M. Su, "Two-phase clustering process for outliers detection," *Pattern Recogn. Lett.*, vol. 22, pp. 691-700, 2001.

[14] S. L. Chiu, "Extracting fuzzy rules for pattern classification by cluster estimation," *presented at the 6th Internat. Fuzzy Systems Association World Congress*, Taipai, Taiwan, 1995.

[15] K. Wang, B. Wang, and L. Peng, "CVAP: Validation for Cluster Analyses," *Data Science Journal*, vol. 8, pp. 88-93, 2009.

[16] G. Chen, S. A. Jaradat, N. Banerjee, T. S. Tanaka, M. S. H. Ko, and M. Q. Zhang, "Evaluation and Comparison of Clustering Algorithms in Anglyzing ES Cell Gene Expression Data," *Statistica Sinica*, vol.12, pp.241-262, 2002.

[17] L. Kaufman and P. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis," New York: John Wiley & Sons, USA, 1990.

[18] P. Yang, B. Huang, "An Outlier Detection Algorithm Based on Spectral Cluster," *presented at Proceedings of the 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Wuhan, China, 2008.

[19] A. Cerioli and A. Farcomeni, "Error rates for multivariate outlier detection," *Computational Statistics & Data Analysis*, vol. 55, pp. 544-553, Jan 2011.

[20] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," *presented at the Proceedings of the 23rd international conference on Machine learning*, Pittsburgh, Pennsylvania, 2006.

[21] A. Asuncion and D. J. Newman, "UCI machine learning repository," [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, 2007.

Dajiang Lei Born in 1979. Received his M.A's degree from the School of Computer Science, Wuhan University of Science & Technology in 2006. Ph. D. candidate in computing science from Chongqing University, China. His main research interests include data mining and intelligent computing.

Qingsheng Zhu Born in 1956. Received his M. A's degree and Ph. D. degree from Chongqing University, China. Professor, doctoral supervisor and senior member of China Computer Federation. His main research interests include business intelligence and image processing.