2008

Detecting Clusters and Outliers for Multi-dimensional Data

Yong Shi
Kennesaw State University, yshi5@kennesaw.edu

Detecting Clusters and Outliers for Multi-Dimensional Data

Yong Shi
Department of Computer Science and Information Systems
Kennesaw State University
Kennesaw, GA 30144
yshi5@kennesaw.edu

Abstract

Nowadays many data mining algorithms focus on clustering methods. There are also a lot of approaches designed for outlier detection. We observe that, in many situations, clusters and outliers are concepts whose meanings are inseparable to each other, especially for those data sets with noise. Thus, it is necessary to treat clusters and outliers as concepts of the same importance in data analysis. In this paper, we present a cluster-outlier iterative detection *algorithm, tending to detect the clusters and outliers in another perspective for noisy data sets. In this algorithm, clusters are detected and adjusted according to the intra-relationship within clusters and the inter-relationship between clusters and outliers, and vice versa. The adjustment and modification of the clusters and outliers are performed iteratively until a certain termination condition is reached. This data processing algorithm can be applied in many fields such as pattern recognition, data clustering and signal processing.*

1 Introduction

The generation of multi-dimensional data has proceeded at an explosive rate in many disciplines with the advance of modern technology. Many new clustering, outlier detection and cluster evaluation approaches are presented in the last a few years. Nowadays a lot of real data sets are noisy, which makes it more difficult to design algorithms to process them efficiently and effectively.

We observe that, in many situations, clusters and outliers are concepts whose meanings are inseparable to each other, especially for those data sets with noise. Thus, it is necessary to treat clusters and outliers as concepts of the same importance in data analysis.

Another fundamental problem in data analysis field is that clusters and outliers are detected mostly based on the information of the features of data sets, and the results are compared to ground truth of natural clusters and outliers. However, in many cases in the real world, such as for some gene expression data, the ground truth and the information of features of the real data sets do not match each other very well, and good results are hard to achieve even using dimension reduction approaches. It is another motive for the development of our algorithm. We tend to detect the clusters and outliers in another perspective, not only relying on the features of the data sets, but also exploiting the relationship between clusters and outliers in a computable way.

In this paper, we present a *cluster-outlier iterative detection* algorithm for noisy multi-dimensional data set. In this algorithm, clusters are detected and adjusted according to the intra-relationship among clusters and the inter-relationship between clusters and outliers, and vice versa. The adjustment and modification of the clusters and outliers are performed iteratively until a certain termination condition is reached.

The remainder of this paper is organized as follows. Section 2 presents the formalization and definitions of the problem, Section 3 describes the cluster-outlier iterative detection algorithm, and concluding remarks are offered in Section 4.

2 Problem Definition

The concepts of cluster and outlier are related to each other. Real world data don't necessarily have natural clusters at all. And for those which do have clusters, there are seldom the cases in reality that the data objects (data points) in the data all belong to some natural cluster. In other words, there are normally outliers existing in the data. One of the aspects of the qualities of clusters and outliers is reflected by how much *diversity* they have inside and have to each other. Clusters and outliers are concepts whose meanings are inseparable to each other. Thus, it is necessary to treat clusters and outliers as the same important concepts in the data processing. Equal treatment to clusters and outliers can benefit applications in many fields.

978-0-7695-3134-2/08 $25.00 © 2008 IEEE
DOI 10.1109/MUE.2008.19

The cluster-outlier iterative detection problem is formalized as follows. In order to describe our approach we shall introduce a few notation and definitions. Let *n* denote the total number of data points and *d* be the dimensionality of the data space. Let the input *d*-dimensional dataset be **X**

$$\mathbf{X} = \{\vec{X}_1, \vec{X}_2, ..., \vec{X}_n\}, \quad (1)$$

which is normalized to be within the hypercube $[0,1]^d \subset R^d$. Each data point $\vec{X}_i$ is a *d*-dimensional vector:

$$\vec{X}_i = [x_{i1}, x_{i2}, ..., x_{id}]. \quad (2)$$

Our main goal is to refine and improve the clustering and outlier detection results of clustering algorithms. According to the input of the initial cluster-outlier division of a data set, we perform the algorithm in an iterative way. In a given iteration step, we assume the current number of clusters is k_c, and the current number of outliers is k_o. The set of clusters is $\mathcal{C} = \{C_1, C_2, ..., C_{k_c}\}$, and the set of outliers is $\mathcal{O} = \{O_1, O_2, ..., O_{k_o}\}$. Here we use the term *compactness* to measure the quality of a cluster on the basis of the closeness of data points to the centroid of the cluster.

2.1 The compactness of a cluster

A cluster in a data set is a subset in which the included points have a closer relationship to each other than to points outside the cluster. In the literature [1, 4], the intra-cluster relationship is measured by *compactness* and the inter-cluster relationship is measured by *separation*. Compactness is a relative term; an object is compact in comparison to a looser surrounding environment.

Definition 1: *Given the current set of clusters* $\mathcal{C} = \{C_1, C_2, ..., C_{k_c}\}$ *and the current set of outliers* $\mathcal{O} = \{O_1, O_2, ..., O_{k_o}\}$*, the* **compactness** *(CPT) of a cluster* C_i *is the closeness measurement of the data points in* C_i *to the centroid of* C_i*:*

$$CPT(C_i) = \frac{\sum_{p \in C_i} d(p, m_{c_i})}{|C_i|}, \quad (3)$$

where m_{c_i} *is the centroid of Cluster* C_i*, p is any data point in Cluster* C_i*,* $|C_i|$ *is the number of data points in* C_i*, and* $d(p, m_{c_i})$ *is the distance between p and* m_{c_i}*. The centroid* m_{c_i} *of the cluster is the algebraic average of all the points in the cluster:* $m_{c_i} = \sum_{p \in C_i} p/|C_i|$.

2.2 The diversities of data groups

We use the term *diversity* to describe the difference between two clusters, the difference between two outliers, and the one between a cluster and an outlier.

The way to denote the diversity between an outlier and a cluster is similar to the problem of the distance measurement between a query point and a cluster which is well studied in the data mining field. Here we use our *compactness* (CPT) concept of the cluster to set up the weights for the distance measurement.

Definition 2: *The diversity between a cluster C and an outlier O is defined as:*

$$D_1(C, O) = w_1 \cdot d_{min}(O, C) + w_2 \cdot d_{avr}(O, C) \quad (4)$$

where $w_1 = \frac{1}{CPT(C)+1}$*,* $w_2 = \frac{CPT(C)}{CPT(C)+1}$ *,* $d_{avr}(O, C) = d(O, m_c)$ *and* $d_{min}(O, C) = max(d(O, m_c) - r_{max}, 0)$ *where* r_{max} *is the maximum distance of the data points in C from its centroid. The criteria for setting the weights* w_1 *and* w_2 *are similar to those in [2].*

We apply a simplified criterion which integrates the *compactness* concept into the diversity measurement of two clusters which represents how compact the data points inside the cluster is.

Definition 3: *The diversity between two clusters* C_1 *and* C_2 *is defined as:*

$$D_2(C_1, C_2) = \frac{d(C_1, C_2)}{CPT(C_1) + CPT(C_2)} \quad (5)$$

where $d(C_1, C_2)$ *can be either the average distance between the two clusters or the minimum distance between them. Here we just simply apply the former one* $d(m_{C_1}, m_{C_2})$*. The larger the value of* $D_2(C_1, C_2)$ *is, the larger diversity the clusters* C_1 *and* C_2 *have to each other.*

Definition 4: *The diversity between two outliers* O_1 *and* O_2 *is defined as:*

$$D_3(O_1, O_2) = d(O_1, O_2) \quad (6)$$

2.3 The qualities of data groups

In this subsection we define the quality of a cluster and the quality of an outlier.

We propose a novel way to define the quality of a cluster C. The quality of C is reflected not only by the diversity between it and other clusters (how far away and different they are from each other), but also by the diversity between it and outliers. In other words, if C is near some outliers, its quality should certainly be impacted, because outliers are supposed to be far away from any cluster. So we take consideration of both the diversity between clusters and the diversity between a cluster and an outlier to define the quality of a cluster.

Definition 5: *We measure the quality of a cluster C as:*

$$Q_c(C) = \frac{\frac{\sum_{C\prime\in\mathcal{C},C\prime\neq C} D_2(C,C\prime)}{k_c-1} + \frac{\sum_{O\in\mathcal{O}} D_1(C,O)}{k_o}}{CPT(C)} \quad (7)$$

The larger $Q_c(C)$ is, the better quality cluster C has.

Similarly, the quality of an outlier O is reflected not only by the diversity between it and clusters, but also by the diversity between it and other outliers. The farther distances it has from other outliers and clusters, the better quality it should obtain.

Definition 6: *We measure the quality of an outlier O as:*

$$Q_o(O) = \frac{\sum_{O\prime\in\mathcal{O},O\prime\neq O} D_3(O,O\prime)}{k_o-1} + \frac{\sum_{C\in\mathcal{C}} D_1(C,O)}{k_c} \quad (8)$$

The larger $Q_o(O)$ is, the better quality outlier O has.

3 Algorithm

The main goal of the algorithm is to mine the optimal set of clusters and outliers for the input data set. As we mentioned in the previous sections, in our approach, clusters and outliers of multi-dimensional data are detected, adjusted and improved iteratively. Clusters and outliers are closely related and they affect each other in a certain way. The basic idea of our algorithm is that clusters are detected and adjusted according to the intra-relationship within clusters and the inter-relationship between clusters and outliers, and vice versa. The adjustment and modification of the clusters and outliers are performed iteratively until a certain termination condition is reached. This analysis approach for multi-dimensional data can be applied in many fields such as pattern recognition, data clustering and signal processing.

The algorithm proceeds in two phases: an *initialization* phase and an *iterative* phase. In the *initialization* phase, we find the centers of clusters and locations of outliers. In the *iterative* phase, we refine the set of clusters and outliers gradually by optimally exchanging some outliers and some boundary data points of the clusters. Each phase is detailed in the following.

3.1 Initialization phase

In the initialization phase, we first find the initial set of medoids. In the next step we dispatch data points to their nearest medoids, forming data subsets associated with medoids. Then we exploit some approaches to determine whether a data subset is a cluster or a group of outliers. Following is each detailed step.

3.1.1 Acquirement of medoids

It is critical to find medoids which can approximate the centers of different clusters for our approach. We choose a random set RS_1 of data points from the original data set with the size of RandomSize1 which is proportional to the required cluster number k. Then we apply the greedy algorithm in [3] to find another random set RS_2 from RS_1 with the size of RandomSize2 which is also proportional to k, and RandomSize1 $>$ RandomSize2. By applying the greedy algorithm on RS_2, the efficiency of the algorithm is greatly improved, and the number of outliers generated by the algorithm is largely reduced.

3.1.2 Dispatching data points

Once we get the smaller random set RS_2 of medoids, we shall find a way to determine which medoids are in some clusters, and which ones are actually outliers.

We first assign each data point dp to a certain medoid $\in RS_2$ which is the nearest one to dp. After this step, each $medoid_i \in RS_2$ is associated with a set of data points.

3.1.3 Initial division of the data set

Cluster or outlier? Now that we get the set E of the initial division of the input data set **X**, we check the size of each medoid-associated data subset $\in E$, and apply some strategies to determine if a medoid-associated data subset is a cluster, or should be regarded as a group of outliers. After the process of $ClusterOrOutlier$, it should be very unlikely the case that the size of cluster set $\mathcal{C}$ is less than k if the initial sizes RandomSize1 and RandomSize2 are large enough. If it does happen, we can just run the initial step again to make sure the size of the cluster set $\mathcal{C}$ is at least k.

3.2 Iterative phase

In the iterative phase, we first merge the initial set of clusters into k clusters. In the second step, we sort clusters and outliers based on their qualities and select the worst clusters and outliers. The quality of each cluster is calculated according to the intra-relationship within clusters and the inter-relationship between clusters and outliers, and vice versa. In the third step, for clusters of the worst qualities, we exploit some methods to select the boundary data points for each of them. In the fourth step, we refine the set of clusters and outliers gradually by optimally exchanging the selected boundary data points and the worst outliers. Steps two, three and four are performed iteratively until a certain termination condition is reached. We detail each step in the following.

3.2.1 Merging clusters

Before we perform the cluster-outlier iterative detection process, we should first merge the current cluster set $\mathcal{C}$ to k clusters. It's an iterative process. In each iteration, two nearest clusters are found in $\mathcal{C}$ and they are merged. The distance between two clusters C_1 and C_2 is based on the diversity measurement $D_2(C_1, C_2)$ of two clusters defined in Section 2. The iteration step is performed until the total number of clusters in $\mathcal{C}$ is k. We compute the centroid of each cluster $C_i \in \mathcal{C}$ (denoted as c_i).

3.2.2 Sorting clusters and outliers

For each outlier $\in \mathcal{O}$, its nearest cluster $\in \mathcal{C}$ is found. The distance between the cluster C and the outlier O is based on the diversity measurement $D_1(C, O)$ defined in Section 2. Also its quality $Q_o(O)$ (defined in Section 2) is calculated according to both the information of outliers in $\mathcal{O}$ and the information of clusters in $\mathcal{C}$. Outliers with the worst qualities are put into set $\mathcal{O}'$.

Similarly, for each cluster C $\in \mathcal{C}$, its quality $Q_c(C)$ (defined in section 2) is calculated according to not only the information of clusters in $\mathcal{C}$, but also the information of outliers in $\mathcal{O}$. Clusters with the worst qualities are put into set $\mathcal{C}'$.

The worse quality a cluster has, the more likely it contains some data points which are better to be outliers. Similarly, the worse quality an outlier has, the more likely it should be included in a certain cluster.

3.2.3 Finding boundary data points

We call those data points in clusters which not only are farthest from the centroids of the clusters, but also have the least number of neighboring data points as boundary data points of the clusters. The latter factor ensures that this method does not only favor clusters of standard geometries such as hyper-spherical ones.

3.2.4 Exchanging data points in clusters and outliers

Next step is to exchange the characteristics of outliers and boundary data points. For each outlier O in $|\mathcal{O}'|$, we add it into its nearest cluster. For each boundary data point bdp , we change it into a new outlier.

The reason that we don't exchange boundary data points *between* clusters is that the whole quality of the data division will be deteriorated if it is conducted.

3.3 Time and space analysis

Suppose the size of the data set is n. Throughout the process, we need to keep track of the information of all points, which collectively occupies $O(n)$ space. For the iteration step, we need space for the information of current set $\mathcal{C}$ of clusters, the current set $\mathcal{O}$ of outliers, the boundary data points of each cluster, the worst outliers and worst clusters in each iteration. The total space needed is $O(n)$. The time required for each iteration is $O(n+|\mathcal{C}|\log|\mathcal{C}|+|\mathcal{O}|\log|\mathcal{O}|)$ mainly for computation of the various of qualities and sort process. $\mathcal{C}$ and $\mathcal{O}$. So the total time required for the algorithm is $O(\Im * (n + |\mathcal{C}|\log|\mathcal{C}| + |\mathcal{O}|\log|\mathcal{O}|))$ in which $\Im$ is the threshold of the iteration number.

4 Conclusion and discussion

In this paper, we presented a novel cluster and outlier iterative detection approach. The method can effectively and efficiently improve the qualities of clusters as well as outliers in a noisy data set of multi-dimensions. Clusters are detected and adjusted according to the intra-relationship within clusters and the inter-relationship between clusters and outliers, and vice versa. The adjustment and modification of the clusters and outliers are performed iteratively until a certain termination condition is reached. Besides treating the clusters and outliers as concepts of the same importance, we also hope to smooth the problem of the lack of match between the ground truth of the real data sets and their available features.

References

[1] Chi-Farn Chen, Jyh-Ming Lee . The Validity Measurement of Fuzzy C-Means Classifier for Remotely Sensed Images. In *Proc. ACRS 2001 - 22nd Asian Conference on Remote Sensing*, 2001.

[2] Dantong Yu and Aidong Zhang. *ClusterTree*: Integration of Cluster Representation and Nearest Neighbor Search for Large Datasets with High Dimensionality. *IEEE Transactions on Knowledge and Data Engineering(TKDE)*, 14(3), May/June 2003.

[3] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:311–322, 1985.

[4] Maria Halkidi, Michalis Vazirgiannis. A Data Set Oriented Approach for Clustering Algorithm Selection. In *PKDD*, 2001.