Unification of Subspace Clustering and Outliers Detection
On High Dimensional Data

Sarala.R\textsuperscript{1}, Prakruthi.V\textsuperscript{2}, Prathibha Annapurna.P\textsuperscript{3}, Saranya.S\textsuperscript{4}
\textsuperscript{1}Department of CSE, Pondicherry Engineering College
Pondicherry, India
1sarala@pec.edu
2prakruthi_30@pec.edu
3prathibha_32@pec.edu
4saranyas_40@pec.edu

Abstract

With the unanticipated requisites springing up in the data mining sector, it has become essential to group and classify patterns optimally in different objects based on their attributes, and detect the abnormalities in the object dataset. The grouping of similar objects can be best done with clustering based on the different dimensional attributes. When clustering high dimensional objects, the accuracy and efficiency of traditional clustering algorithms have been very poor, because objects may belong to different clusters in different subspaces comprised of different combinations of dimensions. By utilizing the subspace clustering as a method to initialize the centroids, and combine with fuzzy logic, this paper offers a fuzzy subtractive subspace clustering algorithm for automatically determining the optimal number of clusters. By our new Fuzzy Outlier detection and ranking approach, we detect and rank the outliers in heterogeneous high dimensional data. The experiment results show that the proposed clustering algorithm can give better cluster validation performance than the existing techniques.

1. Introduction

Clustering is perceived as an unsupervised process, “the process of organizing objects into groups whose members are similar in some way”. The validity of clustering results have to be evaluated by finding the optimal number of clusters that best fits the given data set. Clustering objects in high dimensional spaces may confine to clustering the objects in subspaces which may be of different dimensions. The trial-and-error approach may fail because of the following difficulties:

1) Predefining the number of clusters initially is not easy; 2) Re-initialization at each phase increases the computational cost; and 3) The sparsity and the so-called “curse of dimensionality” as in [8]. In view of the above, we have presented a new fuzzy subspace clustering algorithm for clustering high-dimensional datasets, and an algorithm for detecting the outliers based on Mahalanobis distance.

Initialization of centroids is very sensitive in clustering algorithms. For good initialization, we use subtractive subspace clustering method (subspaces must be selected in the entire high dimensional space and data points must be clustered with their influence) as the initialization technique, and combine fuzzy logic for determining the clusters in subspaces and entire space. The ability to detect outliers can be improved using a combined perspective of outlier detection and cluster identification for multidimensional data.

2. Related work

Clustering is the unsupervised classification of patterns into groups (clusters)[1]. There are different approaches to clustering data which can be described with the help of the hierarchy. At the top level of the hierarchy, there is a distinction between hierarchical and partitional approaches (hierarchical methods produce a nested series of partitions, while partitional methods produce only one).

Instead of identifying exact subspaces for clusters, soft subspace clustering is used to cluster data objects in the entire data space but assign different weighting values to different dimensions of clusters in the clustering process, based on the importance of the dimensions in identifying the corresponding clusters[9]. A noisy data is also called as an outlier. An outlier is a data point which is very dissimilar from the rest of the
data objects based on some attribute. There are univariate and multivariate outliers which can be detected using various algorithms [11].

3. Fuzzy subspace clustering and outlier detection

In this paper, we select the Subtractive Subspace Clustering algorithm as the basic clustering algorithm for initialization, which inherits the advantages of Fuzzy type clustering algorithms such as ease of computation, simplicity, and can deal with noise and overlap clusters. Our proposed system consists of two modules namely: (1) Clustering module; and (2) Outlier Detection module.

Figure 1. Architecture diagram of the proposed system

The clustering module is further divided into two submodules, which are
(1) Initialization using Subspace clustering
(2) Clustering using fuzzy logic

3.1. Subtractive subspace clustering

Subspace clustering seeks to find clusters in different subspaces within a dataset. This means that a data point might belong to multiple clusters, each existing in a different subspace. Subspace algorithm determines the potential of each cluster center and then determines all the centroids. Often in high dimensional data, many dimensions may be irrelevant and can mask existing clusters in noisy data. Subspace clustering algorithms generally localize the search for relevant dimensions allowing them to find clusters that exist in multiple, possibly overlapping subspaces.

The steps include:
(1) Select the data object with the highest potential to be the first cluster center;
(2) Remove all data objects in the vicinity of the first cluster center;
(3) Repeat steps (1) and (2) until the data points are within the radii of a cluster center.

Algorithm for initialization using subtractive subspace clustering:

Input: Dataset \( X = \{ x_1, x_2, \ldots, x_n \} \subseteq \mathbb{R}_d \)

(Initialize the centres \( C^{(0)} \) and set \( C_{\text{max}} = k \))

1. For each \( x_i \in X \), compute the density index

\[
D_i = \sum_{j=1}^{n} \exp \left[ -\frac{\|x_i - x_j\|^2}{(0.5r_a)^2} \right]
\]

2. Let \( D_{c1} = \max \{ D_i : i = 1, 2, \ldots, n \} \), then select \( x_{c1} \) as the first cluster center;

REPEAT
Let \( x_{ck} \) be the \( k \)th cluster center, and the density index be \( D_{ck} \).
For each \( x_i \in X \), update the density index,

\[
D_i = D_i - D_{ck} \sum_{j=1}^{n} \exp \left[ -\frac{\|x_i - x_{ck}\|^2}{(0.5r_b)^2} \right]
\]

UNTIL \( \frac{D_{ck+1}}{D_{c1}} < \delta \)

Where \( r_a, r_b \) and \( \delta \) need preassignment.

\[
r_a = r_b = \frac{1}{2} \min \left( \max \left\{ \frac{\max_i \|x_i - x_k\|}{k} \right\} \right)
\]

And in a general way, at each phase in testing the number of clusters \( c \) between \( C_{\text{min}} \) and \( C_{\text{max}} \), the cluster centers should be initialized at the beginning of running the subtractive algorithm. In subtractive clustering algorithm, the generation order of the cluster centers is determined by the density index. The larger the density index is, the earlier the cluster center generated. Thus, at each stage the top \( c \) cluster centers can be selected as the new initialization cluster centers, and there is no need to re-initialize the cluster centers. After initialization of the centroids, the next step is to apply the fuzzy algorithm to obtain the membership degree for each data point with respect to each cluster.

3.2. Fuzzy means

Fuzzy method of clustering is a data clustering technique in which a dataset is grouped into \( n \) clusters with every data point in the dataset belonging to every cluster to a certain degree. For example, a certain data point that lies close to the center of a cluster will have a high degree of belonging or membership to that cluster and another datapoint that lies far away from the center of a cluster will have a low degree of belonging or membership to that cluster.
With fuzzy method, the centroid of a cluster is computed as being the mean of all points, weighted by their degree of belonging to the cluster. The performance depends on initial centroids. For a robust approach there are two ways.

1. Using an algorithm to determine all of the centroids (In our approach, Subspace clustering algorithm is used to determine the potential of each cluster center and then determine all of the centroids).

2. Run Fuzzy algorithm several times each starting with different initial centroids.

The fuzzy steps include

1. Update membership matrix(U);
2. Determine membership for each point;
3. Repeat steps (1) and (2) until the centroids are stabilized.

**Algorithm for fuzzy clustering after centroids initialization:**

1. At k-step: calculate the centers vectors $C^{(k)} = \{c_j\}$ with $U^{(k)}$.

   $$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

2. Update $U^{(k)}, U^{(k+1)}$.

   $$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\|x_i - c_k\|\right)^{m-1}}$$

3. If $\|U^{(k+1)} - U^{(k)}\| < \epsilon$ then STOP; otherwise return to step 2.

   $\epsilon$ is a predefined value, specified as input.

Generally $\epsilon$ is taken as 0.0001.

By adopting the subtractive clustering as a part of FCM algorithm, the problem of initialization and the maximal number of clusters in traditional “trial-and-error” algorithm is resolved. Subspace clustering is used as the basic clustering algorithm, and combined with the proposed indices which are specially defined for subspace clusters, to determine the optimal number of clusters in high dimensional spaces. Thus, in our proposed system, every dimension contributes to the discovery of clusters, but the dimensions with larger weights form the subsets of dimensions of cluster.

After subspace clustering of the multivariate data points based on their attributes using fuzzy logic, clusters with their centroids and data points are obtained. Since the shape of the clusters as specified is circular (the radii being specified), there may be unclustered points or abnormal data points, which are called outliers. Detection of outliers is very important to obtain good clustering results.

### 3.3. Robust statistical based outlier detection

When analyzing data in datasets, sometimes outlying observations cause problems. Robust statistics\[15\] aims at detecting the outliers by searching for the model fitted by the majority of the data. In real data sets, often some observations are different from the majority. Such observations are called outliers\[15\]. Outlying observations may be different (abnormal) from the majority of points or may be errors, or they could have been recorded under exceptional circumstances, or belong to another population. Consequently, they do not fit the model well. To avoid masking or swamping effects, robust statistics finds a fit that is close to the fit found without the outliers. The outliers can then be identified by their large deviation from that robust fit.

**Multivariate location and covariance estimation:**

Assume that the dataset containd $n$ data objects, which are $p$ dimensional and stored in an $n \times p$ data matrix, $X = (x_1, \ldots, x_n)^T$ with $x_i = (x_{i1}, \ldots, x_{ip})^T$ the $i^{th}$ observation. Empirical mean, $\bar{x}$ is

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

and the empirical covariance matrix $S_x$ is obtained using,

$$S_x = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T / (n-1)$$

**Algorithm for robust statistical based outlier detection (using mahalanobis distance):**

Let $X$ be the matrix of the original dataset containing the data points, with columns centered by their means.

1. Obtain the covariance matrix of the p features using,

   $$S = 1 / (n-1) \ X^T \ X$$

2. The multivariate version of equation to find the distance is

   $$D^2 (x,u) = (x-u)^T S^{-1} (x-u) > k$$

where $D^2$ is the Mahalanobis square distance from $x$ to the centroid of the cluster(u).

An observation (data point) with a large Mahalanobis distance can be considered as an outlier.

3. Obtain the outliers for each of the clusters.

4. Rank the coinciding outliers of the whole dataset in descending order based on their mahalanobis distance.
4. Implementation

4.1. Input specification
The dataset is chosen from the UCI repository. We have chosen the iris plant dataset which contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two and the latter are NOT linearly separable from each other[16]. In total, the dataset contains 150 instances of iris plants which are to be clustered into their corresponding classes based on their multidimensional attributes.

Iris plant dataset
The dataset contains 150 instances (50 in each of three classes) with 4 attributes describing them (4 numeric, predictive attributes). The class Distribution is 33.3% for each of 3 classes.

![Figure 2. Iris plant dataset description](image)

The attributes describing/giving information about the plants in the dataset[16] are:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class: -- Iris Setosa
   -- Iris Versicolour
   -- Iris Virginica

4.2. Implementation details
The dataset is fed to the Matlab program and subspaces are obtained based on the similarity measure. The number of clusters are determined based on the radii specified for each cluster. This is fed as input to the fuzz algorithm. Hence the process of initializing centroids is done using the subspace algorithm. The membership matrix is initialized and the objective function is computed. Fuzziness is based on the minimization of the objective function. Finally, this minimization process is repeated until the clusters obtain stability i.e until the objective function value reaches a minimum threshold (specified by the user). The clusters are obtained with their centroids and their respective multivariate data points.

4.3. Output specification
After subspace clustering of the multivariate data points based on their attributes using fuzzy logic, clusters with their centroids and data points are obtained. There may be unclustered points or abnormal data points called multivariate outliers. The outliers of each cluster are then detected using robust statistical outlier detection method—using Mahalanobis distance. The outliers of each cluster are considered and the overall outliers of the entire multidimensional dataset are detected and ranked based on their mahalanobis distance from the centroids of all the clusters. Accuracy and Optimality of the clusters is determined based on the cluster validity indices (CVI) which includes compactness (Intra-cluster distance), separability (Inter-cluster distance) and exclusiveness (Probability density function—to measure irregularity) as parameters. The CVI is compared with the existing system to arrive at the consensus.

5. Results and discussions
The evaluation of any clustering system is based on some performance parameters. The performance parameters for the clustering process are an objective function and cluster validity index.

5.1. Objective function
Clustering algorithms aim at minimizing an objective function, a squared error function whose value can be calculated using,

\[ J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2 \]

where \( \| x_i^{(j)} - c_j \| \) is a chosen distance measure between a data point, and \( x_i^{(j)} \) and the cluster centre \( c_j \), is an indicator of the distance of the \( n \) data points from their respective cluster centers.

5.2. Cluster validity index
Considering Separability (Inter-Cluster Distance), Compactness (Intra-cluster distance) and Exclusiveness (Probability density function—to measure irregularity) as parameters in assessing the cluster validity indices, an objective function is defined for each cluster. Inter-cluster distance measures clusters “separability”, intra-cluster distance measures cluster “compactness”, while exclusiveness measure is to measure irregularity i.e finding outliers in the dataset. A clustering algorithm is considered best if it has minimum compactness, maximum separability and minimum outlier distance from the centroids of the clusters(if the dataset is good).
The objective function is generalized and defined for a data set of ‘n’ number of dimensions.

Objective Function (OBF) = Min(Compactness) + Max (Separability) + Max (Exclusiveness)

Mathematically, OBF defined for a cluster is based on compactness, separability and exclusiveness, is as follows,

\[ OBF = \min \left[ C = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \| x_{ij} - y_{ij} \|^2 \right] + \max \left[ S = \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \| e_{i} - c_{j} \|^2 \right] + \max \left[ \frac{1}{2\pi^{k/2}} \frac{1}{|\Sigma|^{1/2}} \sum_{i=1}^{n} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right) \right] \]

The clusters obtained using the existing system and the proposed system, are compared based on the objective function and the cluster validity indices. Optimality of clusters is obtained by detecting the outliers and eliminating them, if necessary.

5.2.1. Separability. Table 1 lists the separability measure of the clusters, which are obtained by using the existing system (fuzzy clustering) and the proposed system (fuzzy subspace clustering) on the multi-dimensional dataset. The fuzzy subspace method gets the radii of the clusters as input and obtains the optimal number of clusters.

<table>
<thead>
<tr>
<th>Radii of the cluster (No.of clusters)</th>
<th>Existing System (Fuzzy clustering)</th>
<th>Proposed System (FuzzySubspace clustering)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6(3)</td>
<td>81.4160</td>
<td>82.5230</td>
</tr>
<tr>
<td>0.3(8)</td>
<td>227.7040</td>
<td>245.9924</td>
</tr>
<tr>
<td>0.22(11)</td>
<td>353.1556</td>
<td>330.0207</td>
</tr>
</tbody>
</table>

5.2.2. Compactness. Table 2 lists the compactness measure of the clusters, which are obtained by using the existing system (fuzzy clustering) and the proposed system (fuzzy subspace clustering) on the multi-dimensional dataset. The fuzzy subspace method gets the radii of the clusters as input and obtains the optimal number of clusters.

<table>
<thead>
<tr>
<th>Radii of the cluster (No.of clusters)</th>
<th>Existing System (Fuzzy clustering)</th>
<th>Proposed System (FuzzySubspace clustering)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6(3)</td>
<td>8618.23</td>
<td>8144.01</td>
</tr>
<tr>
<td>0.3(8)</td>
<td>6753.99</td>
<td>6514.85</td>
</tr>
<tr>
<td>0.22(11)</td>
<td>3978.4</td>
<td>3524.568</td>
</tr>
<tr>
<td>0.2(15)</td>
<td>1893.463</td>
<td>1575.901</td>
</tr>
</tbody>
</table>

5.2.3. Exclusiveness and outliers. Fig.3 graphically represents the outliers’ Mahalanobis distances with respect to the centroids of the clusters. Minimum distance of the outliers indicates a good dataset and indicates a good clustering algorithm. Here, x-axis indicates the outliers and y-axis represents their total distance from the centroids.

6. Conclusions and future enhancements

In high dimensions, data becomes very sparse and distance measures become increasingly meaningless and difficult to obtain similarities. A novel architecture and algorithm for unifying subspace clustering and detection of outliers, has been proposed and implemented. This work obtains the influence of the clusters on data dimensions, which indicates the influence on subspaces. Our proposed system overcomes the existing system drawbacks by providing good quality clusters. Optimal clusters are obtained by detecting the outliers and eliminating them, if necessary.

There are many potential applications with high dimensional data where subspace clustering approaches
could help to uncover patterns missed by current clustering approaches. Applications in bioinformatics and text mining are particularly relevant and present unique challenges to subspace clustering. For future work, a stronger focus can be put on the problem of multivariate data and considering their hierarchies. The issue of time complexity and optimality of the algorithm can be considered and worked on. Furthermore, heuristic approaches to improve the quality of the clustering results can be introduced. Subspace clustering over fast changing environments like data streams where access to raw data is not allowed, can be undertaken.

References


[8] Lance Parsons, Ehtesham Haque and Huan Liu, Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter Volume 6, Issue 1 (June 2004).


