

Clustering Large Datasets and Computation of Prototypes

Conference : MLM3037

Shobha.G, R.V.College of Engineering, Mysore road, Bangalore - 560059, India
Email shobhatilak@rediffmail.com

S C Sharma ,R.V.College of Engineering, Mysore road, Bangalore - 560059, India

Abstract

Cluster Analysis deals with finding groups in data. The groups should be such that the objects within a group are similar to each other whereas the objects in different groups are as dissimilar as possible. In clustering problems, one is particularly interested in the characterization of the clusters by means of typical or representative objects or prototypes. There exist many measures for measuring dissimilarity. The dissimilarity measure used in the present work is the Euclidean distance. The objective of the work reported in the current chapter is to find representative objects. This is useful in data reduction and characterization. In view of this, the number of objects should be as small as possible and still serve as good representatives of the data.

1. Introduction

The clustering methods are broadly classified into partitioning and hierarchical methods. A partition method constructs k clusters or k groups, satisfying following conditions.

1. Each group must contain at least one object, and
2. Each object must belong to exactly one group

The centroid based method is intended for interval-scaled measurements (in principle without missing measurements) [7] [8]. It attempts to minimize the average squared distance. Given a data consisting of interval scaled measurements; the “centroid” of cluster is the center of gravity of the cluster. In this method dissimilarity between two patterns is defined as the Euclidean distance between them. K-means algorithm, employing centroids as representatives, attempts to minimize average squared distance. The concepts of “medoids” is proposed by Kaufman and Rousseum [1] and are explained below:

In partitioning the data into, say k clusters, the representative objects for each of these clusters are chosen such that the average dissimilarity of representative objects with all other objects of the same cluster is minimized. Such optimal representative objects are called medoids. The medoids are preferred over centroids because of the following reasons:

- (1) medoids are some of the objects in the given data
- (2) medoids are more robust with respect to outliers

The main thrust of the current exercise is to examine whether representative objects of the data, in terms of medoids, are good enough to classify test patterns that would be presented

for classification. Once the optimal set of medoids is obtained, these medoids would represent all aspects of the data.

The second concept that has been examined for finding representative objects is “leaders”. The number of leaders depends on the dissimilarity threshold used in identifying leaders [4] [5] [6]. An initial leader is arbitrarily chosen. Subsequently, every pattern is compared with the leader. If the new pattern falls within close dissimilarity threshold, it becomes part of the cluster headed by the leader; otherwise a new leader is identified. As the threshold decreases, the number of leaders would increase. Leaders are computed for each class such that they provide good classification accuracy, after considering various thresholds.

The analysis in the current Chapter is based on k-medoid method [1], [2] and leader algorithm. The following sections describe the methods and provide a summary of various exercises carried out using these methods.

The following section discusses the important subject of clustering large data sets, bringing out the relative merits and demerits of various clustering algorithms in grouping large data sets. Sections containing details of the exercises carried out in computing prototypes along with result follow this. The summary section discusses comparison of results obtained in various exercises and provides conclusions

2. PAM (Partitioning Around Medoids)

This was one of the first k-medoids algorithms introduced attempts to determine k algorithm repeatedly tries to make a better choice of cluster representatives. All of the possible pairs of objects are analyzed, where one object in each pair is considered a representative object and the other is not. The quality of the resulting clustering is calculated for each such combination. An object, O_j , is replaced with the object causing the greatest reduction in error. The set of best objects for each cluster in iteration forms the representative objects for the next iteration. The final set of representative objects are the respective medoids of the clusters. The complexity of each iteration is $O(k(n-k)^2)$. For large values of n and k , such computation becomes very costly.

The following is the k-medoid algorithm for partitioning based on medoid. The input to the algorithm is the number of clusters (k) and a database (D) containing n objects. The output is a set of k clusters that minimizes the sum of the dissimilarities of all the objects to their nearest medoid. The output of the above algorithm is discussed in detail in Section 3.10.

- Step 1: Arbitrarily choose k objects in D as the initial representative objects or seeds;
 Step 2: Repeat
 Step 3: Assign each remaining object to the cluster with the nearest representative object;
 Step 4: Randomly select a nonrepresentative object, O_{random} ;
 Step 5: Compute the total cost, S , of swapping representative object, O_j , with O_{random} ;
 Step 6: If $S < 0$, then swap O_j with O_{random} to form the new set of k representative objects;
 Step 7: Until no change.

3. Computation of Medoids and Supplementing with Nearest Neighbors

The exercises in the section are aimed at computing minimum number of representative objects. It is observed that in practice in the large data set size of 3990 patterns, computation of medoids beyond 60 is expensive, both with PAM and CLARA methods. In view of this, only a small number of medoids are computed for a class of 3990 patterns. With 50 classes of 3990 patterns each, the total number of representative objects (medoids) amounts, including the medoids themselves, to 100. Each medoid is considered one at a time; dissimilarities with all the other patterns in the class are computed and ordered. First 10 neighbors are considered, including the medoid itself. Thus 10 neighbors to each of the 10 medoids of each class are identified, depending on the dissimilarity measure. At this stage, the total number of representative objects amount to 1000. Classification accuracy using an NNC (nearest neighbor classifier) is computed using a test data patterns. The exercise is repeated for different values of medoids, and nearest neighbors. The results are summarized in Table 1

Table 1: Classification Accuracy of Medoids after Including Neighbors

No. of medoids per class	No. of NN (nearest neighbor)	Total size of training data	Classification Accuracy
15	10	1000	81.4%
15	15	1500	82.2%
15	20	2000	84.6%

Subsequently, in addition to computing neighbors, an averaged pattern is considered as a representative pattern. Thus, in this case, total number of representative patterns amounts to 100 only. The exercise is repeated for different values of medoids and neighbors. The results are provided in Table 2.

Table 2: Classification Accuracy of Medoids with Averaged Neighbors

No. of medoids	No. of nearest neighbors for averaging	Total size of training data	Classification Accuracy using NNC (%)
25	5	400	81.2
	10	400	81.1
	15	400	82.1
	20	400	82.8
	25	400	82.2
	30	400	81.4
35	5	350	84.5
	10	350	83.2
	15	350	84.0
	20	350	84.4
	25	350	84.6

4. Computation of Medoids Using CLARA [11]

In the current set of exercises, the medoids are computed using CLARA. To understand how well the given set of medoids represents the entire data, classification accuracy is computed using NNC. As mentioned earlier, the computation cost both with PAM and CLARA is expensive as the required number of medoids increases. With CLARA, the number of training patterns for computing medoids is taken as $40 + 2k$ [11], where 'k' is the required number of medoids. It is observed that even though CLARANS is faster, it is found to be time consuming for medoids (say, k) greater than 50, for the current size and dimensionality of the data set. This is found by following the rule suggested by Ng and Han[12] for considering maximum neighbors for 'k' medoids in the data of size n, as the following:

“If $(n-k) \leq \text{minmaxneighbour}$, then $\text{maxneighbour} = k (n-k)$; otherwise, maxneighbour equals the larger of the values between $p\%$ of $k (n-k)$ and minmaxneighbour ”, where p is taken in the range between 1.0 and 2.0.

Table 3 provides the details of number of medoids, training data size obtained thereby and the corresponding classification accuracy.

Table 3: Classification Accuracy of Medoids Using CLARA

No. of medoids per class	Training Data size	Classification Accuracy
50	500	81.9%
80	800	86.1%
100	1000	88.5%
200	2000	90.8%

5. Computation of Medoids Using CLARA with Divide and Conquer Approach

Albeit, as mentioned earlier, the computation complexity increases with increasing number of medoids, it is necessary to obtain enough number of medoids to adequately represent the entire data. In other words, the number of medoids should be adequate in number to provide classification accuracy of 90% and above. In view of this points, divide and conquer approach is followed. In this case, the data of 200,000 patterns of each class is divided into 4 blocks of 50,000 each. Depending on the required number of medoids, one fourth of the medoids are computed from each of these partitions. At the end, all the medoids computed from the different blocks are combined to provide the required set of medoids. Table 4 provides statistics of the number medoids and classification accuracy.

Table 4: Classification Accuracy of Medoids Using Clara with Divide And conquer Approach

No. of medoids per class	Training data size	Classification Accuracy
40	400	86.6
60	600	87.5
100	1000	88.9
120	1200	89.6
160	1600	90.5
200	2000	90.2
240	2400	88.9
300	3000	91.5
400	4000	91.8

6. Computation of Representative Objects Using Leader Algorithm

The selected medoids would contain redundant medoids, since all patterns do not complexity of the pattern, its proximity to other patterns of other classes for possible misclassification, etc. Thus, there exists a possibility to reduce the number of medoids.

The current method is based on the assumption that if patterns were so chosen that they are uniformly distributed by a given dissimilarity measure, they would represent the pattern set adequately. In the current work, the dissimilarity measure is taken as Euclidean distance. This is a valid assumption because, the distance between closely resembling patterns is near zero and the distance between an ideal pattern and of a pattern having distinctly different appearance is large.

An outline of leader algorithm is provided in Section 1.2.2. For a given class of patterns, with a predefined distance threshold and a starting pattern, distance with each pattern is computed. If the distance falls within the threshold, a subsequent pattern is considered for comparison. In case the distance exceeds the threshold, a new leader is identified. At this stage, the comparison is extended to new leader too. The procedure is repeated until no new leader is found. In this method, first encountered pattern that exceeds the threshold is chosen as a representative pattern. It is not centrally located like the medoid. Figure 3.4 contains the flowchart. Table 5 summarizes Classification Accuracy of Leaders for Various Thresholds for 3444 test patterns.

Table 5: Classification Accuracy of Leaders for Various Thresholds

No of test patterns =3444

Threshold	No. of leaders	No. of validation patterns	Classification Accuracy using validation data(%)	Classification Accuracy using test data(%)
1.0	6324	456	100	92.74
1.5	5783	987	100	92.87
2.0	5358	1425	100	92.87
2.5	4434	2436	100	92.77
3.0	3718	3152	99.97	92.59
3.5	2476	4394	99.51	91.96
4.0	1799	5112	98.48	90.76
4.5	1020	5812	96.46	89.70
5.0	570	6189	92.68	85.67
5.5	335	6469	88.66	80.90
6.0	198	6598	82.88	75.78
6.5	99	6709	72.19	66.90
7.0	76	6509	66.50	60.89

7. Conclusions

This paper provides various options of computing representative patterns

- In case of medoid computation, supplemented by near-by patterns, the maximum classification accuracy obtained is 84.6% (Table 1).
- Averaging of neighbors has provided a maximum CA of 88.3% (Table2).
- CLARA provides CA of 90.8% with 2000 medoids (Table 3)
- CLARA with divide and conquer approach has provided a CA of 90.2% with 2000 medoids, 91.5% with 3000 medoids and 91.8% with 4000 medoids. The results with increasing medoids from 2000 to 4000 shows that increase of medoids do not significantly improve CA. It is only marginal.
- The results with leader show best CA of above 92% in many cases, although the number of representatives is large (Table 4).

From the above summary of results, the following conclusions may be drawn, considering CA of above 90% as a good result:

- Medoids above 2000 provide good CA
- Leader algorithm generally provides good results

References

1. Raymond T.Ng, Jiawei Han, Efficient and Effective Clustering Methods for Spatial Data Mining, Proc Int'l Conf. Very Large Data Bases, Sept. 1994, pp 144-155.
2. A.K.Jain and R.C Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.
3. Ming-Syan Chen, Jiawei Han and Philip S. Yu, Data Mining: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, 1996 pp 866-883.
4. Usama M. Fayyad, G. Piatetsky – Shapiro and P. Smyth, From Data Mining to Knowledge Discovery: An Overview in Advances in Knowledge Discovery and Data Mining Edited by U.M. Fayyad, G.P.Shapiro, P.Smyth and R. Uthurusamy AAA/MIT Press, 1996, pp.1 – 34.
5. Nikolai K. Kasabov, Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering, MIT Press, 1996 pp 74-89.
6. Christophere J. Mathews, Philip K. Chan and Gregory Piatetsky-Shapiro, Systems for Knowledge Discovery in Databases, IEEE Trans. On Knowledge and Data Engineering, Vol 5 No.6, 1993 pp 903-913.
7. R.O. Duda, P.E.Hart and D.Stork, pattern Classification and Scene Analysis, Wiley Interscience, V01 4 No5, 2000 pp 304 –322.
8. M.Prakash and M.Narasimha Murty, Growing Subspace Pattern Recognition methods and Their Neural Network Models, IEEE Tran. On NN, 1997 pp 161-168.