Sri Lankan Journal of Applied Statistics, Vol.9, 2008, p.109-120

Clustering of Large Data Sets Using Fuzzy Principal Component Analysis

S. SAMPATH[1] AND V.S.VAIDYANATHAN[2]

[1]Department of Statistics, University of Madras
Chennai, India
[2]Department of Statistics, Loyola College
Chennai, India

ABSTRACT

Cluster Analysis plays a vital role in various branches of data mining specifically in Image Processing, Pattern Recognition etc. The algorithms available in Cluster Analysis can be broadly categorized into "Hierarchical" and "Nonhierarchical" algorithms. Nonhierarchical algorithms can be further divided into "Partitioning ", "Density Based" and "Model Based" algorithms. The main problem associated with the partitioning algorithms like k-means, k-medoids etc is their inability to handle irregularly shaped data sets. As a result of this, the clusters created by those algorithms are not necessarily globally optimum clusters. To handle such situations, specialized algorithm like "CLARA" is available in the literature. Even though they partially take care of some limitations of k-means and k-medoids, the results produced by them are also not globally optimum. In this paper, a new clustering algorithm which makes use of "Fuzzy Principal Components" and the "Theory of Finite Population Sampling" has been developed. The proposed algorithm has been tested for some natural benchmark data sets available in the UCI data repository as well as for a variety of multivariate normal populations simulated with the help of R Statistical Computing Package. It is found to perform well with respect to many clustering validation measures like the value of objective function, purity and F-measure.

Key words: Clustering, Fuzzy Principal Components Analysis, Systematic Sampling, Purity, f –measure

CLUSTERING LARGE DATA SETS

Clustering algorithms play a vital role in various branches of data mining especially in Image Processing, Pattern Recognition, Codebook Optimization etc. Clustering algorithms can be broadly classified into "Hierarchical" and "Nonhierarchical" algorithms. Some approaches used under "Nonhierarchical" clustering are (i) Partitioning Algorithms (ii) Density Based Algorithms and (iii)

Model Based Algorithms. Among several algorithms available, two most frequently referred algorithms coming under the category of Partitioning algorithms are "k-means" and "k-medoids" algorithms. Details of these algorithms can be found in Tan *et al.*, 2006. These two algorithms induce a partition of the data set such that a given objective function is optimized. The objective function to be minimized is described below.

Let D be the given data set, $C_1, C_2, .., C_k$ be subsets of D such that $(i)\bigcup_{i=1}^{k} C_i = D$ and $(ii) C_i \cap C_j = \phi, i \neq j$. The k-medoids algorithm determines $C_1, C_2, .., C_k$ such that the objective function $J(C_1, C_2, .., C_k) = \sum_{i=1}^{k} \sum_{j \in C_i} \| X_j - M_i \|$ is minimized where X_j is the data vector corresponding to the j^{th} object of the data set and $M_i, i = 1,2,..,k$ is the medoid (a representative object of the data set) corresponding to the set C_i. In the case of k-means of algorithm, instead of a medoid, the group centroid computed using the members of the cluster C_i is used.

While attempting to minimize the above objective function, one frequently faces "hill climbing problem" and hence most of the times these algorithms end up with a local optimum solution rather than a global optimum. Hence the clusters formed are likely to be far away from the actual optimum solution. Situations like this are bound to arise, specifically, when these algorithms are applied for large databases. Special algorithms like CLARA (Kaufman and Rousseeuw, 1990) are available in the literature to address the above problem. CLARA is specifically used to cluster large data sets and it induces a partition of the underlying data set by making use of randomly selected objects from the data set.

The steps involved in obtaining clusters by CLARA are given below.

(1) Take 5 samples of sizes $40 + 2k$ randomly from the given data

(2) Perform k-medoids partitioning for each of the 5 sample data sets obtained in step 1

(3) Identify the best set of medoids among the 5 sets of medoids obtained in step 2

(4) Partition the data set using the best set of medoids identified in step 3

The partition obtained in step 4 is taken as the clustering solution for the given data set.

The following key points are to be observed in the above algorithm

(i) The number of samples to be selected in step 1 is 5.

(ii) The number of objects included in each sample is $40+2k$.

(iii) The manner in which objects are drawn to constitute the samples.

Hence, it will be of interest to analyze the effect of implementing alternative choices for CLARA by addressing each of the above mentioned points. Motivated by the above discussion, a new algorithm is presented in the sequel which uses the techniques of systematic sampling and fuzzy principal component analysis.

SYSTEMATIC SAMPLING FOR PARTITIONING DATA SETS

As mentioned in Section 1, in CLARA, 5 subsets of the given data set are sampled randomly. It is pertinent to note that random selection can be made either by replacing the data objects before making a new selection or one can follow a without replacement scheme. In either case one has large number of possible samples. To be specific, when there are 100 objects and 10 clusters are needed one has 100^{60} possible samples in the case of with replacement scheme and $\binom{100}{60}$ possible samples in without replacement scheme. It is well known from the Sampling Theory, that the above sampling procedure is likely to yield poorer results when compared to sampling schemes which use label information. This is bound to happen, specifically, for populations exhibiting a trend of given pattern. Sampling schemes which use such label information are christened as "Systematic Sampling Schemes". For more details, regarding such sampling schemes one can refer to Sampath, 2005. For efficiency issues related to such schemes one can refer to Bellhouse and Rao, 1975; Suresh Chandra *et al.*, 1992; Sampath and Uthayakumaran (1998). Hence, it makes sense to use systematic sampling rather

than random sampling to select samples which are to be considered for identifying the best set of medoids.

In data mining, mutivariate objects are encountered rather than univariate objects which are often found in the theory of sampling. Hence, ordering the objects becomes important to achieve overall efficiency. To sort out this issue of ordering objects, Fuzzy Principal Component Analysis (FPCA) is applied in this paper.

FUZZY PRINCIPAL COMPONENT ANALYSIS

It is known from the Theory of Multivariate Statistical Analysis, Principal Components (PC) are normalized linear combinations possessing special properties in terms of variances which retain the information contained in the given system and hence ordering the objects based on PC scores is more meaningful. For example, the first PC is the normalized linear combination having maximum variance and the second PC is the normalized linear combination having the maximum variance among all linear combinations uncorrelated with the first PC and so on. For the sake of brevity, detailed discussion on PCA is skipped. More about PCA can be found in Johnson and Wichern (2002).

In this paper we propose to use Fuzzy PCA to order the data objects and thereafter make use of systematic sampling to induce the partition of the data set. The advantage of using FPCA over PCA is that in PCA the inherent sub structure present in the data set is not taken into account thereby treating all objects as equally important where as in FPCA the relative importance of objects based on the sub structure of the data set is made use of. More details about the merits of FPCA over PCA and applications of FPCA can be found in Pop (2001), Cundari *et al.*, (2002) and Sarbu and Pop (2005). In the following paragraphs, we discuss the method of performing FPCA.

Let $X = \{x_1, x_2, ..., x_T\}$ denote the set of feature vectors in the p dimensional space. A fuzzy k clustering of these vectors specifies the degree of membership of each vector in each of the k clusters. Let $U = \lfloor u_{jt} \rfloor$ denote a k by T membership matrix where u_{jt} is the degree of membership of x_t in the j^{th} cluster, $t = 1,2.....T$, $j = 1,2...., k$. Note that each membership degree should lie between 0 and 1 and the sum of the membership degrees for each vector equals one.

The fuzzy covariance matrix of the j^{th} cluster is defined (Lee, 2004) as

$$F_j = \frac{\sum_{t=1}^{T} u_{jt}(x_t - c_j)(x_t - c_j)^t}{\sum_{t=1}^{T} u_{jt}}, j = 1,2,...,k,$$

where c_j is the centroid of the j^{th} cluster. The fuzzy membership matrix will be determined by minimizing the objective function

$$J_m(U,C;X) = \sum_{t=1}^{T}\sum_{j=1}^{k}(u_{jt})^m d^2(x_t,c_j)$$

where $m(\geq 2)$ is called the degree of fuzziness, $d^2(x_t,c_j)$ the distance between x_t and c_j is defined as

$$d^2(x_t,c_j) = \|x_t - c_j\|_F^2 = (x_t - c_j)^t F_j^{-1}(x_t - c_j).$$

The minimum value of the objective function is obtained by solving the following equations in an iterative manner:

$$u_{jt} = \frac{\left[\frac{1}{d^2(x_t,c_j)}\right]^{\frac{1}{(m-1)}}}{\sum_{j=1}^{k}\left[\frac{1}{d^2(x_t,c_j)}\right]^{\frac{1}{(m-1)}}}$$

and

$$c_j = \frac{\sum_{t=1}^{T}(u_{jt})^m x_t}{\sum_{t=1}^{T}(u_{jt})^m}$$

Thus, there will be k fuzzy covariance matrices $F_j, j = 1,2,...,k$ that would lead to the minimum value of the objective function. Combining the fuzzy covariance matrices, we define the overall fuzzy covariance as

$$F = \frac{\sum_{j=1}^{k} e_j F_j}{\sum_{j=1}^{k} ej}$$

where $e_j = \sum_{i=1}^{T} -u_{ji} \log u_{ji}$ is the entropy of the j^{th} cluster, $j = 1,2,...,k$. Since entropy is a measure that captures the relative importance of clusters, it is being used as a weight in finding the overall fuzzy covariance. FPCs are defined as Principal Components obtained with the help of the overall fuzzy covariance matrix F developed in the above manner.

NEW ALGORITHM

Based on the observations made in the Section 2 and the ideas presented in Section 3, we propose the new algorithm as follows.

Let N denote the number of objects in the given dataset and let $n \leq N$ be a positive number. Compute $k_1 = \frac{N}{n}$. Here we refer n as the sample size and k_1 as sampling interval.

Step 1: Find the fuzzy covariance matrices $F_j, j = 1,2,...,k$ that will minimize the objective function given in Section 3.

Step 2: Compute the overall fuzzy covariance matrix F.

Step 3: Perform FPCA by making use of F and obtain the FPC loadings.

Step 3: Use the loadings of the first FPC and compute the first FPC score for all the objects

Step 4: Arrange the objects in the data set either in ascending or descending order according to the first FPC scores

Step 5: Partition the data set into k_1 disjoint sets where the set $s_r = \{r + (j-1)k_1, j = 1,2,..,n\}, r = 1,2,..,k_1$

Step 6: Apply k-medoids algorithm for each of the k_1 sets and compute the objective function value corresponding to these sets of medoids

Step 7: Identify the medoids yielding the best value for the objective functions' obtained in Step 6.

Step 8: Partition the data set using the medoids identified in Step 7.

It is pertinent to note that in the above algorithm, the disjoint sets formed in Step 5 are nothing but all possible Linear Systematic Samples of size n. Further, the algorithm identifies the best set of medoids corresponding to each possible linear systematic sample and the final partition of the data set is based on the best out of them. Here the sample size can be decided at the time of actual implementation of the algorithm taking into account the size of the database rather than fixing it as $40+2k$. Apart from this, it makes use of a better cross section of the data set to identify the best set of medoids than those identified by CLARA, because the sets defined in Step 5 ensure that objects included in a sample are placed at a reasonably good label distance of k_1 units.

EXPERIMENTAL STUDY

In this section, the performance of the proposed algorithm has been assessed using three real life bench mark data sets available in R-Statistical Computing Package (iris data set) and in ftp://ftp.ics.uci.edu/pub/machine-learning-databases (wine and Yeast data sets).

Also three synthetic data sets, each comprising of 900, 900 and 1300 observations respectively and made up of 3 groups are generated randomly from Trivariate Normal Distributions using R-Statistical Computing Package. The first data set consisting of 900 observations is based on 300 observations each drawn from three different multivariate normal populations, namely $N_3(\mu^{(i)},\Sigma), i=1,2,3$ where $\mu^{(1)}=[5 \quad 17 \quad 27]$, $\mu^{(2)}=[14 \quad 24 \quad 32]$, $\mu^{(3)}=[25 \quad 39 \quad 47]$ and $\Sigma=\begin{bmatrix}1.3 & 0.2 & 2.1\\ 0.2 & 2.5 & 1.7\\ 2.1 & 1.7 & 5.0\end{bmatrix}$. The second data set consisting of 900 observations is based on 300 observations each drawn from three different multivariate normal populations, namely $N_3(\mu^{(i)},\Sigma), i=1,2,3$ where $\mu^{(1)}=[10 \quad 22 \quad 30]$ $,\mu^{(2)}=[60 \quad 30 \quad 88]$ $\mu^{(3)}=[125 \quad 138 \quad 149]$, and

$\Sigma = \begin{bmatrix} 4 & -5 & 1 \\ -5 & 7 & -2 \\ 1 & -2 & 8 \end{bmatrix}$. The third data set comprising of 1300 observations is generated from three trivariate normal distributions having different mean vectors and different covariance matrices. The following are choices of the mean vectors and the covariance matrices used in the generation of the three data sets.

$\mu^{(1)} = [1.39 \quad 2.33 \quad 4.25], \mu^{(2)} = [8.12 \quad 10.33 \quad 14.78],$

$\mu^{(3)} = [19.74 \quad 21.56 \quad 24.77]$ $\Sigma_1 = \begin{bmatrix} 2 & 3 & -5 \\ 3 & 8 & 12 \\ -5 & 12 & 12 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 2 & -1 & 3 \\ -1 & 5 & -3 \\ 3 & -3 & 5 \end{bmatrix}$ and

$\Sigma_3 = \begin{bmatrix} 4 & -5 & 1 \\ -5 & 7 & -2 \\ 1 & -2 & 8 \end{bmatrix}$

It is to be mentioned that 294, 465, 541 observations respectively are generated from the three distributions. In order to nullify the effect of grouping on the performance of the proposed algorithm, the observations within each of the synthetic data sets are shuffled.

The performance of the new algorithm is measured in terms of the value of the objective function, Purity and F-measure. The definitions of Purity and F-measure are given below:

Let there be k clusters $c_1, c_2, \ldots, c_k$ with sizes $m_1, m_2, \ldots, m_k$. Also let $m = \sum_{i=1}^{k} m_i$.

Purity is a measure to know the extent to which a cluster contains objects of a single class. The Purity of cluster i denoted by p_i is defined as $p_i = \max_j p_{ij}$

where $p_{ij} = \frac{m_{ij}}{m_i}$, $i = 1,2,\ldots,k$ and $j = 1,2,\ldots,s$, s being the number of classes in the data set, m_{ij} is the number of observations of the j^{th} class belonging to the i^{th} cluster. The overall purity of a clustering is given by Purity $= \sum_{i=1}^{k} \frac{m_i}{m} p_i$. F-measure measures the extent to which a cluster contains only objects of a particular class and all

objects of that class. The F-measure of cluster i with respect to class j is given by

$$F(i,j) = \frac{(2 * precision(i,j) * recall(i,j))}{(precision(i,j) + recall(i,j))}$$

where $precision(i,j) = p_{ij}$ and $recall(i,j) = \frac{m_{ij}}{m_j}$.

Tables 1, 2 and 3 give the objective functions' value together with Purity and F-measure for real-life and synthetic data sets (before and after shuffling) corresponding to the algorithm proposed in this paper as well as those of CLARA assuming $n = \frac{N}{10}$ and k_1 as the integer nearest to $\frac{N}{n}$. It is to be noted that wherever necessary (that is, when attributes had different units of measurement) standardization has been done.

Table 1: Objective function, Purity and F-measure values for real life data sets based on proposed Algorithm and CLARA

Data Set		**Iris**	**Wine**	**Yeast**
Data Size		150	170	1350
Number of Clusters		3	3	5
Objective function	Proposed Algorithm	0.6096	92.5839	0.1751
	CLARA	0.6260	93.0690	0.1856
Purity	Proposed Algorithm	0.8733	0.7235	0.4786
	CLARA	0.8800	0.7117	0.4622
F measure	Proposed Algorithm	0.9769	0.7093	0.4708
	CLARA	0.8801	0.6945	0.4052

Table 2: Objective function, Purity and F-measure values for synthetic data sets (before shuffling) based on proposed Algorithm and CLARA

Data Set		**1**	**2**	**3**
Data Size		900	900	1300
Number of Clusters		3	3	3
Objective function	Proposed Algorithm	2.6031	2.5529	3.3635
	CLARA	2.7226	2.6170	3.4063
Purity	Proposed Algorithm	1.0000	1.0000	1.0000
	CLARA	1.0000	1.0000	1.0000
F measure	Proposed Algorithm	1.0000	1.0000	1.0000
	CLARA	1.0000	1.0000	1.0000

Table 3: Objective function, Purity and F-measure values for synthetic data sets (after shuffling) based on proposed Algorithm and CLARA

Data Set		**1**	**2**	**3**
Data Size		900	900	1300
Number of Clusters		3	3	3
Objective function	Proposed Algorithm	2.5857	2.5559	3.3215
	CLARA	2.6531	2.6695	3.3836
Purity	Proposed Algorithm	0.3511	0.3566	0.4161
	CLARA	0.3511	0.3566	0.4161
F measure	Proposed Algorithm	0.3511	0.3566	0.3848
	CLARA	0.3511	0.3566	0.3848

A close scrutiny of the above table reveals that the objective functions' value in the case of the proposed algorithm is smaller than that of CLARA. Thus the proposed algorithm minimizes the value of the objective function better than CLARA for real life as well as synthetic data sets (both before and after shuffling). Also the Purity and F-measure values for the proposed algorithm are better or equal to that of CLARA in the case of real as well as synthetic data sets (both before and after shuffling).

CONCLUSION

Findings of the experimental studies based on both real life and the synthetic data sets clearly favor the usage of newly developed algorithm that makes use of systematic sampling based on FPCA ranks to do partitioning clustering for large data bases. Further work is being done to explore the possibilities of using different sampling schemes together with FPCA for other clustering methods.

REFERENCES

Bellhouse, D.R. and J.N.K. Rao (1975). Systematic sampling in the presence of a trend, *Biometrika*, 62(3): 694–697.

Cundari, T.R. C. Sarbu and H.F. Pop (2002). Robust Fuzzy Principal Component Analysis (FPCA). A comparative study concerning interaction of carbon – hydrogen bonds with molybdenum–oxobonds, *Journal of Chemical Information and Computer Sciences*, 42(6): 1363-1369.

Johnson, A.R. and D.W. Wichern (2002). *Applied Multivariate Statistical Analysis.* Prentice Hall, US.

Kaufman, L. and P.J. Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley and Sons, New York.

Lee, K.Y. (2004). Local Fuzzy PCA based GMM with dimension reduction on speaker identification. *Pattern Recognition Letters*, 25(16): 1811-1817.

Pop, H.F. (2001). Principal components analysis based on a Fuzzy Sets Approach. *Informatica*, XLV1(2): 45-52.

Sampath, S. (2005). *Sampling Theory and Methods, 2nd Edition*, Narosa Publishing House, New Delhi.

Sampath, S. and N. Uthayakumaran (1998). Markov Systematic Sampling, *Biometrical Journal*, 40(7): 883-895.

Sarbu, C. and H.F. Pop (2005). Principal component analysis versus Fuzzy PCA: A case study: the quality of Danube water (1985-1996). *Talanta*, 65(5): 1215-1220.

Suresh Chandra, K. S. Sampath and G.K. Balasubramani (1991). Markov sampling for finite populations. *Biometrika*, 79(1): 211-213.

Tan, T., M. Steinbach and V. Kumar (2006). *Introduction to Data Mining*. Addison Wesley, US.