Statistical Approach to Clustering in Pattern Recognition

Yujing Zeng
Department of Electrical Engineering & Comp. Science
OhioUniversity, Athens, OH 45701 USA

Janusz Starzyk
Department of Electrical Engineering & Comp. Science
OhioUniversity, Athens, OH 45701 USA

Abstract-- Clustering is a typical method of grouping data points in an unsupervised learning environment. The performance of most clustering algorithms is dependent on the accurate estimate of the cluster number, which is always unknown in the real applications. In this paper, we propose a new parametric approach, which starts with an estimate of the local distribution and efficiently avoids pre-assuming the cluster number. This clustering program is applied to both artificial and benchmark data classification and its performance is proven better than the well-known k-means algorithm.

I. *INTRODUCTION*

The clustering problem is defined as a problem of classifying a group of data points into a number of clusters without any prior knowledge about data structure, to produce a concise representation of the data. It is a fundamental means for multivariate data analysis widely used in numerous applications, especially in pattern recognition.

Clustering techniques have been investigated extensively for decades. The existing approaches to data clustering include statistical approach (e.g., the K-means algorithm, [Yub 95]), optimization approach (e.g. branch and bound method [Che95A], simulated annealing technique [Sel91]), and neural network approach (e.g., HEC [Mao96]). Some special techniques, such as fuzzy clustering [Kar94], [Che95B] and classification based on mixtures [Che88], are hot topics of study.

According to [Fun 90], the existing clustering methods can be divided into two approaches. One is the parametric approach, and the other one is the nonparametric one.

The widely used K-means method is one example of the parametric approach. In this method, a criterion is given, and data is arranged into a pre-assigned number of groups to optimize the criterion. Another kind of parametric approach assumes some mathematical form to express the data distribution, such as summation of normal distributions [Tit85]. Both approaches are developed based on the view of the global data structure and are ready to be improved in combination with other optimization methods. However, the performance of all this kind of methods depends on the assumption of the cluster number, which is hard to estimate beforehand in real applications.

In the nonparametric approach, data are grouped according to the valleys of the density function, such as in the valley-seeking method [Sal93]. This method does not require knowledge of the number of clusters beforehand. But since its performance is, in general, very sensitive to the control parameters and the data distribution, its application is limited. Furthermore, the resulting clusters of nonparametric clustering procedures contain the tails of other distributions and do not contain their own tails.

In our study, a new approach is derived to express data distribution with a Multi-Gaussian method. This approach is a parametric one, but it does not pre-assume the number of clusters, making it more suitable for many applications. In the next section, this approach is described in detail. Applications of the resulting algorithm to some artificial data sets and to the well-known IRIS data are provided in Section 3. Section 4 gives the conclusions.

II. *CLUSTERING PROGRAM*

As discussed before, the fatal shortcoming of the parametric approach is its dependence on the pre-assumed cluster number. To resolve this problem in our method, clustering starts with an estimate of the local distribution. We construct small clusters, called seed clusters, according to the local distribution, and then merge those whose distributions are consistent with each other.

The problem of clustering the signal data can be decomposed into two subproblems:

- Extraction of the seed clusters – Cluster Growth program
- Merging of clusters with a similar distribution of data – Cluster Merge program

The clustering process moves from small, local clusters that capture information about the density of local data points towards bigger clusters with a specified probability density function. Their merging is performed in order to reduce the complexity of data representation as well as to provide statistically supported generalization ability for classification.

2.1 *CLUSTER GROWTH PROGRAM*

In the Cluster Growth program, seed clusters are constructed in the following way. First, one point is chosen randomly to initiate the cluster, and then the cluster absorbs the external nearest neighbors. Here the nearest neighbor of the cluster is defined as the point whose distance to the cluster is shortest. The distance from a point to a cluster is defined as its distance to the nearest point in this cluster. This distance will be compared with the threshold, Thd_Growth. If it is less than the threshold, this point will be absorbed and the process will be repeated for the next nearest neighbor; otherwise, the cluster growth is terminated. Fig.1 shows the cluster growth process.

In Fig.1, the point represented by the circle is the initial point. The arrows show the process of cluster growth. In this

0-7803-6661-1/01/$10.00 c2001 IEEE

way, the seed clusters are constructed based on the local distribution learned by the Cluster Growth program. Since no prior knowledge is needed, any structure of data can be followed, and the cluster growth can be terminated automatically according to the local distribution.

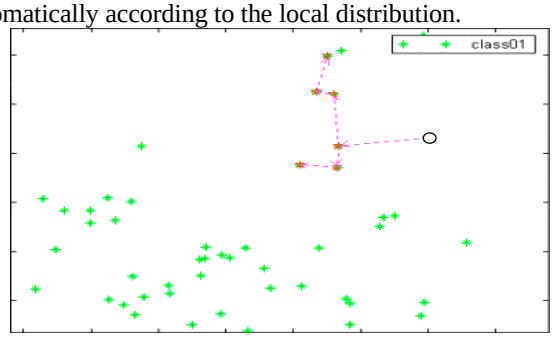
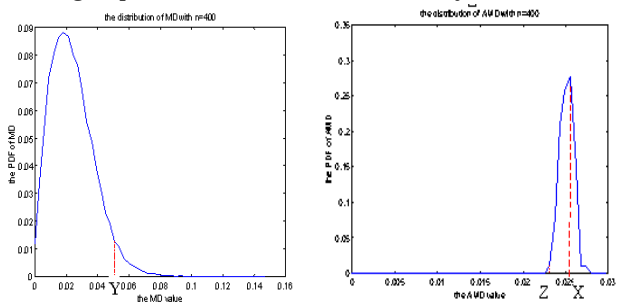


Fig. 1 Cluster growth process

2.1.2 *THRESHOLD ESTIMATION*

Based on the above description, threshold selection is critical to the Cluster Growth program. To find its value, statistical properties of the minimum distance between every two data points, called MD, is studied. These statistical properties are decided by the local distribution of data points. Throughout the clustering process, the exact distributions of data points are unknown and need to be approximated. Considering the complexity of high dimensional distributions, practical selection of approximating functions is limited to either normal or uniform model. Gaussian distribution is often an attractive choice based on its limited number of parameters. However, since a cluster constructed by the seed cluster growth program often contains very few samples, usually the concerned region is too small to obtain a statistical estimate of normal parameters, such as covariance matrix. Therefore, it is practical to assume that data samples in a local neighborhood follow a uniform, rather than Gaussian distribution.

In our study, the statistics of MD are studied for N data points which are uniformly distributed in the D-dimensional cube with a side size equal to A. The experimental results show that the distributions of MD and its average value for each group, called AMD are determined by N, D and A.



(a) MD distribution (b) AMD distribution

Fig. 2 A distribution of MD and AMD(where N=400,D=2, A=1)

Their distributions have the following features: (Fig.2 gives an example distribution of MD and AMD)

1) The distributions of both MD and AMD are asymmetric, with the lower bound closer to the mean value than the upper bound.
2) The distributions of both MD and AMD are dependent on the values of N and D. When N or D increases, the curves of the distributions become narrower.
3) The AMD has a definite lower bound which is larger than 0.

The upper bound of MD (represented as Y, on Fig. 2(a)) defines values of Thd_growth with normalized A. As analyzed, the Thd_growth is in a positive proportion to the real value of the scale factor A_r. That is,

$$Thd\_growth = Y * A_r \quad (1)$$

So the value of A_r is necessary in the estimation of the Thd_growth, but it cannot be obtained directly from the input data because the cluster distribution is unknown. However, we can calculate AMD based on the selected samples in the current cluster. Fortunately, the AMD value is in a positive proportion to the scale factor A as well. Therefore, if we can get a specific value Z of AMD with normalized A, and given N and D, A_r can be estimated using the value of AMD observed in the current cluster, AMD_r, from the following equation

$$A_r = AMD_r / Z \quad (2)$$

Since AMD is not a constant but a random variable, there is a problem of choosing a Z value from the distribution of AMD. Since we consider Thd_growth as the maximum distance to absorb the samples, it is reasonable to take the lower bound of AMD as a specific value for Z.

By plugging equation (2) into (1), we get:

$$Thd\_growth = \frac{Y}{Z} * AMD_r = Spread * AMD_r \quad (3)$$

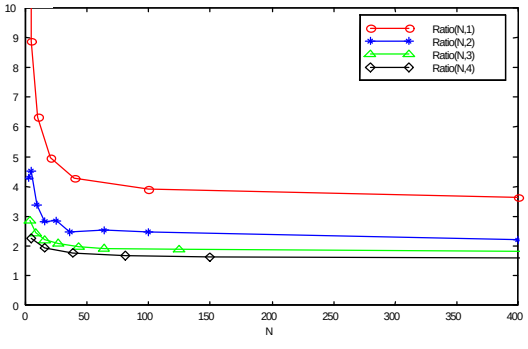


Fig. 3 *Spread* as a function of N with D=1,2,3,4

Since Y and Z can be estimated from the experiment results, then the *Spread*, which is defined as Y/Z, is easy to obtain. Fig. 3 shows the Spread values for different N and D. Thus

by combining the observed average mean value, AMD_r, and *Spread* we can estimate dynamically the threshold value by (3). The threshold value will control the cluster growing termination.

2.2 *CLUSTER MERGING*

After the cluster growth program, we get a lot of small seed clusters, some of which overlap each other. Next, the clusters coming from a consistent distribution should be selected, and merged into a single cluster. The problem is choosing which clusters to merge. The Cluster Merge program proposed in this section is based on a similarity between the pdf (Possibility Density Function) estimates before and after merging. In the following discussion, a selection rule is defined, and then the program is described.

2.2.1 *THE SELECTION RULE --NOA*

For each cluster, we approximate its distribution by Gaussian pdf. The process of merging is driven by the replacement of the multi-Gaussian by the single Gaussian distribution.

Consider two seed clusters, A and B. Their approximate Gaussian distributions are *pdfA* ~ $N(m_A, \Sigma_A)$ and *pdfB* ~ $N(m_A, \Sigma_A)$. The two distributions form a multi-Gaussian approximation, represented by *pdfW*. Suppose that the points in two clusters are from one Gaussian distribution and should be merged. The distribution after merging is another Gaussian, *pdfM*. The resulting functioins, *pdfW* and *pdfM* approximate an unknown distribution of the same group of points. They overlap each other and produce two areas: the overlapped area and non-overlapped area. Fig. 4 shows the relationship between *pdfA*, *pdfB*, *pdfW* and *pdfM*. The non-overlapped area is marked with a shadow. When *pdfA* and *pdfB* are obtained form the same Gaussian distribution, *pdfW* and *pdfM* are closer, and the non-overlapped area is smaller (theoretically approaches 0) .

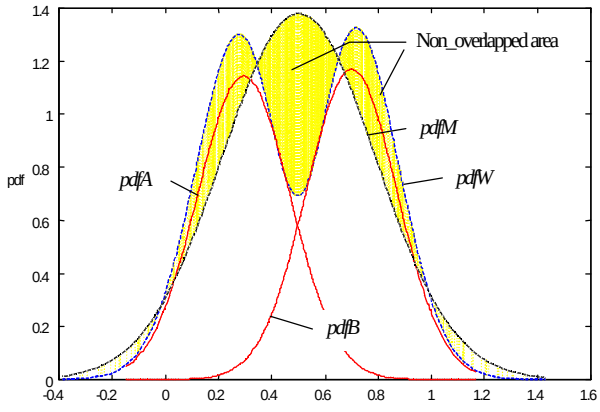


Fig. 4 An illustration demonstrating the non-overlapped area

The cluster merging process is set up to replace *pdfW* with *pdfM* and to simplify the pdf approximation. Assuming that *pdfW* is close to the true distribution, we want to reduce the error created by the merging process through selecting clusters carefully. So the similarity between *pdfW* and *pdfM* is used as a direct rule for the selection. The size of the non-overlapped area, represented as *NOA*, can be used to estimate this similarity.

In our algorithm, we randomly generated two groups of points based on distributions of the seed clusters. The calculation of *NOA* is done based on those points. This way, we can control the number of points concerned and reduce the influence of any insufficiency of the points in the seed clusters used for calculations required in the Mote Carlo integration used to obtain the *NOA*. The obtained *NOA* will be compared to the threshold Thd_Merge set empirically to determine if the two clusters should be merged.

Thd_Merge is estimated based on a sequence of experiments, which calculates the NOA on two groups of randomly generated data. For simplicity, our experiments are done on the data groups in one-dimensional uniform distribution. Since NOA shows the relationship of two groups and is not affected by the dimension and distribution of the data groups, this assumption does not lose its generality in higher dimensions. Fig. 5 shows NOA values with different distance between two means of the groups, (represented as Distance) and different size ratio of two groups (represented as R_size). Based on results of statistical analysis, we make the following observations:

- When Distance increases, NOA values increase monotonically. This agrees with the real situation: when the distance between two means increases, the distribution of two clusters have a greater separation, and the possibility of merging declines.

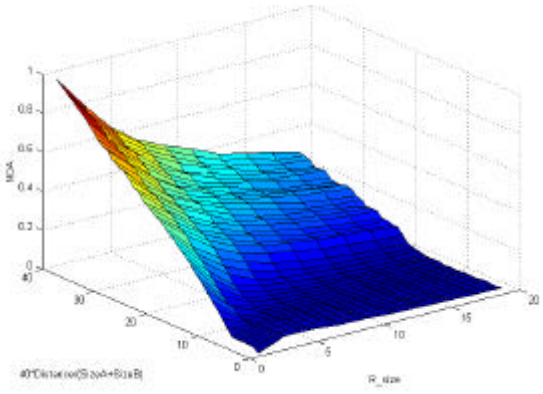


Fig. 5 NOA as a function of Distance and R_size

- When R_size increases, the NOA value decreases and its slope vs. Distance is smaller. This is expected as well, since when the ratio of two cluster sizes increases, the smaller one has less of an impact on the approximations of pdfW and pdfM. Consider the extreme situation when cluster B compared to A is so small that it can be ignored.

PdfM and pdfW are both close to pdfA, and NOA will be small, independent of cluster B's location. In this situation, the slope of NOA vs. Distance is 0, and no threshold can be found. Fortunately, we need not consider an R_size with a large value (i.e. larger than 25) since we can set its limits in the cluster growth program. Fig. 6 gives the curve of NOA vs. R_size with Distance equal to (SizeA+SizeB)/2, where two clusters barely touch each other (which is a cut in the middle of Fig.5). The threshold value of NOA that can be used for cluster merging (Thd_Merge) can be read directly from this curve.

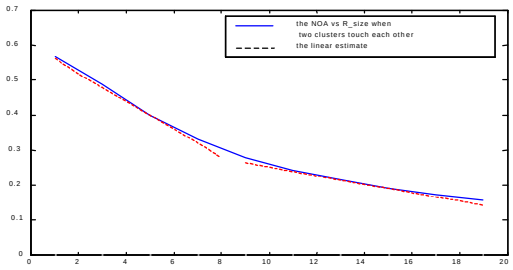


Fig. 6 NOA vs. R_size when two clusters touch each other(Distance = (SizeA+SizeB)/2)

2.2.2 *MERGING PROCESS*

Suppose that we have a number of seed clusters from different classes. The procedure used to apply the NOA threshold to cluster merging has the following steps:

Step 1: Calculate the NOA values between every two clusters. For each cluster I, find its partner J, whose NOA is the smallest one.

Step 2: If the partner cluster J is from the same class as the current cluster I, their NOA is compared with Thd_Merge. If the NOA value is less than Thd_Merge, proceed to step 4; otherwise, change the current cluster and return to Step 1.

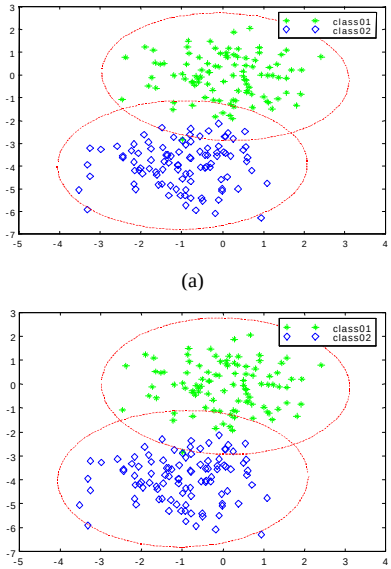
Step 3: If the partner cluster J is from a different class than the current cluster I, cluster I can not be merged with any other clusters. Change the current cluster and go back to Step 1.

Step 4: Merge Cluster J and I and Return to Step 1.

III. *CLUSTERING RESULTS OF EXAMPLES*

In this section, we investigate the performance of this clustering algorithm on several data sets. Comparisons are made with the well-known k-means algorithm. Since k-means algorithm needs additional process to estimate the number of clusters beforehand, for simplicity, we show its performance with the same number of clusters which is detected by our clustering algorithm. Most time, this assumption gives the best clustering result.

1. The first artificial data set includes two clusters with Gaussian distributions as shown in Fig.7. The distance between two clusters, that is the distance between two means of clusters, changes from $2\bullet(s_1+s_2)$ to $3\bullet(s_1+s_2)$, where s_1 and s_2 are the standard deviation of two clusters. Experimental results show that our clustering algorithm produces the same results as k-means algorithm. Fig.7(a),7(b) show the clustering results of both algorithms when the distance between two clusters is $2\bullet(s_1+s_2)$.



(a)

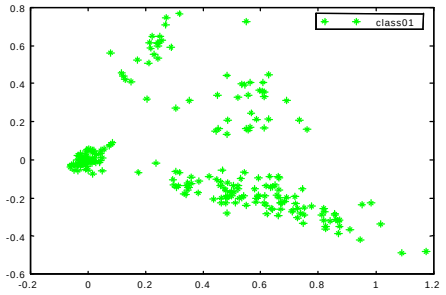
(b)

Fig.7 clustering results on the artificial data I
(a) the clustering result of our clustering algorithm;
(b) the clustering result of the k-mean algorithm

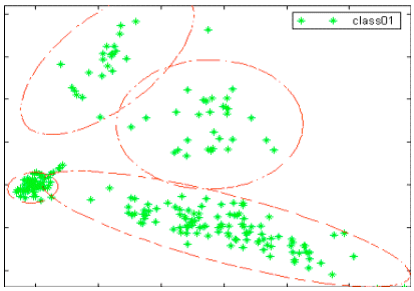
2. The ability of our algorithm to find the optimal position of a cluster's centroid is shown also in the following examples. This artificial data set is closer to the real world data, as shown in figure 8(a). There are four obvious clusters, whose distributions are not strictly Gaussian. In this case,k-means algorithm cannot find the 'correct' centroid's positions of these four clusters. As shown in Fig. 8(b). The partitioning result of our algorithm is shown in Fig.8(c).
3. We have also applied our algorithm to Fisher's "iris" data [Jam85]. The data set consists of two sepal and two petal measurements from 150 irises, 50 from each species (1, Setosa, 2, Versicolor, 3, Virginica). From [Jam85], we know that group 1 is well separated from groups 2 and 3, but 2 and 3 overlap. Both our algorithm and k-means method did a good job to separate the first group, but misclassify several points of group2 to group3. For the partitioning with these three clusters, our algorithm has correctly classified 145 out of 150, or 95% of the irises, while the performance of the k-means algorithm classified 90% of the iris data correctly. So that the algorithm developed in this study is better than the K-mean algorithm in performance.

The computational cost of the algorithm presented in this paper is heavy compared to that of k-means algorithm. To

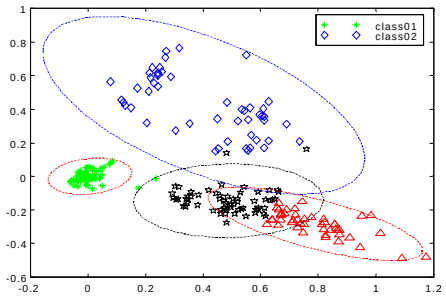
cluster the iris data, our algorithm needs 20 seconds while k-means only needs about 1.5 seconds on a PC with a speed of 233MHz. With the additional process to find the number of clusters, which is required by the k-means algorithm, this difference can be reduced. However, it is still an important problem, especially in high dimension. This larger computational cost is related to the calculation of NOA. A more efficient computational method with a similar accuracy is currently under investigation.



(a)



(b)



(c)

Fig. 8 clustering results on the artificial data II
(a) the original data set; (b) the clustering result of our clustering algorithm; (c) the clustering result of the k-mean algorithm;

VI. *CONCLUSION*

Many algorithms have been devised for clustering. They are divided into two categories: the parametric approach and the nonparametric approach. The clustering method described in this paper is a parametric approach. It starts with an estimate of the local distribution, which efficiently avoids pre-assuming the cluster number. Then the seed clusters that come from a similar distribution are merged by this algorithm. This clustering program was applied to both artificial and benchmark data classification and its performance is proven better than the well-known k-means algorithm.

REFERENCE:

[Che88]P. Cheesman, J. Kelly, J. Self, M. Stutz, W. Taylor, and D. Freeman, "AutoClass: a Bayesian classification system". *Proc. 5th Int. Workshop on Machine Learning*, Ann Arbor, pp. 54-64. San Mateo, CA, Morgan Kaufmann, 1988.

[Che95A] C.H.Cheng, " A branch and bound clustering algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, pp. 895-898, 1995

[Che95B] Tai Wai Cheng, Dmitry B. Goldgolf, and Lawrence O. Hall "Fast Clustering with Application to Fuzzy Rule Generation", IEEE Transactions on…,1995

[Fun83] K. Fukunaga and T. E. Flick, "Estimation of the parameters of a Gaussian mixture using the method of moments", *Tran. IEEE Pattern Anal. And Machine Intell*. PAMI-5, pp. 410-416, 1983

[Fun90] Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, INC, 1990

[Jai88] A. K. Jain, R.C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ. Prentice Hall, 1988.

[Jam85] M.James, *Classification Algorithm. NewYork*: John Wiley,1985,ch.7

[Kar94] Mohaned S. Kamel and Shokri Z. Selim, "A relaxation approach to the fuzzy clustering problem", *Fuzzy Sets and Systems* 61 177-188, 1994

[Mao 96] Jianchang Mao and Anil. K. Jain, "A self-Organizing Network for Hyperellipsoidal Clustering (HEC)", *IEEE Transactions on Neural Networks*, vol.7 , no. 1, pp. 16-29, Jan. 1996

[Sal93] Sarle, W.S and Kuo, An-Hsiang, "The MODECLUS Procedure", *SAS Technical Report* P-256, Cary, NC: SAS Institute Inc. 1993

[Sel91] S. Z. Selim and K. S. Al-Sultan, " A simulated annealing algorithm for the clustering problem," *Pattern Recognition* 24, pp. 1003-1008, 1991.

[Sta 99] Janusz A. Starzyk, "*Clustering for Classification of HRR Signals*", Progress Report to Air Fore Research Laboratory, Feb. 1999.

[Tit85] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*, Wiley, New York, 1985

[Yub 95] B. Yu and B. Yuan. "A global optimum clustering *Engineering applications of artificial inteligence*. 8(2):223-227, April 1995