On the Approach of Automatic Adjustments for Gaussian-Mixture Clustering

Chin-Hwa Kuo[1], Tzu-Chuan Chou[2]* and Meng-Chang Chen[2]

[1]*Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C*
[2]*Institute of Information Science, Academia Sinica, Taipei, Taiwan 115, R.O.C.*

Abstract

In this paper, we discuss the dual-problem of adjusting the mixture number and avoiding local optima in the estimation of a Gaussian mixture. This estimation is widely used in unsupervised-classification applications; however, its results are serially sensitive to the initial setting, which is difficult to optimize. It is also difficult to automatically designate the mixture number in advance. In much of the literature, these two issues are discussed separately, meaning that one is considered at the expense of the other. To overcome this problem, we present some strategies that automatically and simultaneously adjust the mixture number and escape from local optima. The evaluation results are very encouraging and show that the proposed strategies are effective.

Key Words: Parameter Estimation of Gaussian Mixture, EM Algorithm, Clustering Algorithm, Local Optima

1. Introduction

The parameter estimation of mixture density is probably one of the most widely used techniques in computational pattern recognition. Multivariate Gaussian mixture density functions are the most popular continuous probability density functions used to describe an unordered statistically independent set of vectors (input data) because they can approximate any continuous probability density function. In this paper, we focus on a mixture of non-overlapped and non-adjacent multivariate Gaussian functions because it is very useful for solving unsupervised classification (clustering) problems. The Expectation Maximization (EM) algorithm is the method most frequently used for this purpose [1–3]. In this paper, we discuss two widespread problems that occur when we use the EM algorithm to estimate the parameters of the Gaussian mixture. The first is that improper initial data cause the local-optimum result, and the second is that the mixture number cannot be decided in advance.

Not only EM algorithm but also many conventional clustering algorithms, e.g., the *k*-means [4], the fuzzy *c*-means [5–7], and the varieties of fuzzy c-means [8,9], cannot designate the number of clusters in advance. If the number of clusters is incorrect, most clustering algorithms cannot yield a satisfactory result. Many researchers have tried to decide the number of clusters [10–14] by running a clustering algorithm several times with different assigned numbers of clusters. This leads to a sequence of clustering systems, whereby each system is tested by various objective functions to determine the best cluster number. In these methods, all possible cluster numbers must be tried and tested manually or by other techniques, e.g., MDL (Maximum Description Length). Unfortunately, every trial may fall into local optima if the initial setting is improper. This means that the trial with the correct number of clusters may produce a local optimum result that may be not good enough to compete with the results from wrong numbers of clusters. Moreover, as it is necessary to designate the range of possible cluster

*Corresponding author. E-mail: tzuchuan@iis.sinica.edu.tw

numbers in advance, the above approaches will still fail if the real cluster number is not in the designated range. Consequently, these kinds of trial mixture-number (number of clusters) discovery methods are not always reliable. If an algorithm can adjust the number of clusters automatically, trying to test all possible mixture-numbers is a redundant process.

Even if the correct number of clusters is determined in advance, conventional clustering algorithms can be trapped in local optima when the initial setting is improper. These algorithms are very sensitive to the chosen initialization; that is to say, if a poor initial setting is chosen, they fall into the local optima. However, good initial settings are difficult to determine. To overcome the above problem, Lawrence O. Hall et al. [15] proposed a genetically guided algorithm (GGA) that can ameliorate the difficulty of choosing an initialization for the fuzzy c-means clustering algorithm. The GGA algorithm attempts to achieve minimal sensitivity to initialization and avoidance of local optima. Nir Friedman et al. [16] enhanced the basic EM algorithm procedure by incorporating the technique of simulated annealing to escape from the local optima. Meanwhile, B. Schachter et al. [17] used the test statistic of distortion to estimate the goodness of fit for a hypothetical distribution of clusters.

Several researchers have devised supplementary enhancement strategies to improve conventional clustering algorithms. In the Split and Merge EM algorithm, (SMEM) [18], Naonori Ueda et al. used split and merge operations based on different justification criteria to escape from the local maxima. In the SMEM algorithm, on one hand the larger the number of samples that belong to two clusters simultaneously, the greater will be the opportunity to merge these two clusters, and on the other hand if the Kullback-Leibler divergence between the distribution of a cluster and the local density among the clusters is larger, the clusters have more opportunities to be split. With these two entirely different criteria, the full EM steps and the partial EM steps are performed iteratively until the objective function obtained by the SMEM is convergent. This algorithm holds the number of clusters (mixture number) by synchronized split and merge operations, and the number of clusters must be designated in advance, which is the same as the above-mentioned clustering methods. However, as stated previously, the number of clusters is usually unknown in practical applications, but the SMEM algorithm is powerless to find the correct number of clusters.

As previous discussion, it is clear that, for the parameter estimation of a Gaussian mixture using the EM algorithm, the problem of choosing the mixture number and the problem of local optima cannot be solved separately because they both influence the results. Therefore, if only one is solved, successful results cannot be guaranteed. Nikos, Vlassis et al. [19] proposed a greedy EM algorithm to improve the general EM algorithm and attempted to solve these two problems simultaneously. Under the assumption that maximum likelihood learning of a k+1 mixture of Gaussian clusters can be performed in a greedy manner by adding one new component to the maximized k mixture of the Gaussian clusters, they started with one component and successively added components to the mixture until the objective function of the EM algorithm was convergent. Although the greedy EM algorithm seems to outperform general EM algorithms, it has a problem in that the $(k+1)^{th}$ optimum solution can not always be derived from the k^{th} optimum solutions. Moreover, if one of the derivations is deviant, the result will be abnormal because the greedy algorithm cannot correct itself.

With regard to the issue of local optima, some researchers propose a simple scheme that tries many different random initial-settings to obtain many results and chooses the best one. However, we do not know how many times we need to try because the failure rate may be rather high in some cases. For example, the sample shows in Figure 1 is a tough case for using the pure EM algorithm. We can test and verify that most random initial-settings are improper. That is, when the failure rate is very high, we should not expect to obtain a correct result merely depending upon the trick of many trials. Thus this kind of strategy for avoiding local optima is not always reliable and we do not suggest using it.

We analyzed several results of local optima. Some of the final clusters contain more than one cluster, and some clusters overlap each other. These unreasonable phenomena must be corrected. Recall that Gaussian distribution governs clusters in the EM algorithms; therefore, to obtain a refined clustering result, we must ensure that every cluster fits with the hypothesis of Gaussian distribution. Henry C. Thode et al. [20] proposed a number of strategies to calculate the maximum likelihood estimates of a

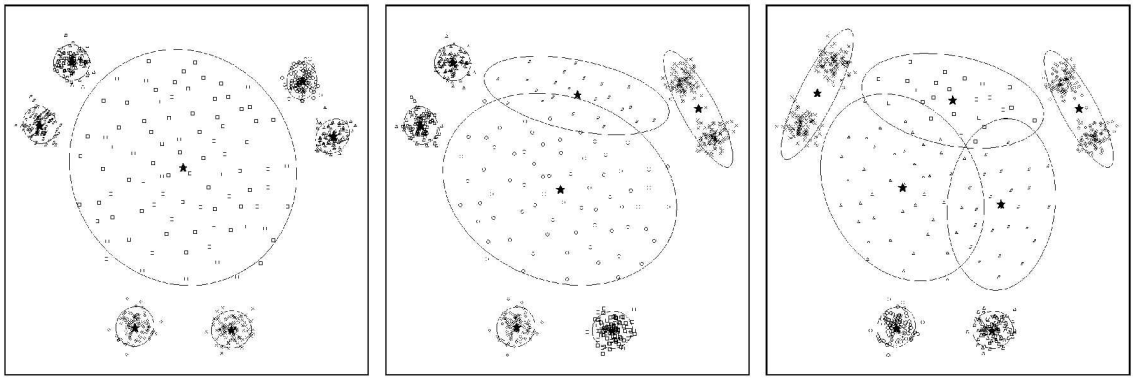

Figure 1. The tough satellite sample with one planet cluster and 6 satellite clusters. (a) is the expected result, but for most random initial settings, the pure EM algorithm produces the local-optimum results, e.g. (b), (c) and etc.

local, rather than global, maximum via the likelihood ratio test of the null hypothesis, which states that a sample is from only one normal distribution. This contrasts with the alternative hypothesis, which states that the sample is from a mixture of two distinct normal distributions, each with an equal variance. The latter approach is only useful for 1-dimensional samples with an equal variance for every cluster. Another kind of approach is to use the χ^2 test to calculate the goodness of fit for the hypothesis of Gaussian distributions. B. Schachter et al. [17] proposed that the goal of clustering is to obtain a sequence of hyper-ellipsoidal clusters starting with cluster centers positioned at maximum density locations in the pattern space, and growing clusters around these centers until an χ^2 test for goodness of fit is violated. However, in our experience, using the χ^2 test for the goodness of fit of a Gaussian distribution in the high-dimensional pattern space is not always accurate. This is because the fit of a high-dimensional Gaussian distribution is not only a matter of the number of samples in specific ranges. It is also a matter of fitting the strict definition of multivariate normal distribution. By definition, a random vector $\boldsymbol{X}$ having a multivariate normal distribution must guarantee at least the following two conditions: first, the linear combinations of the components of $\boldsymbol{X}$ must be distributed normally; second, all subsets of the components of $\boldsymbol{X}$ must have a (multivariate) normal distribution [21]. From the viewpoint of geometry, this is not only a matter of the number of samples in specific ranges; it is also a matter of testing for the symmetry and completeness of structures in the multi-dimensional space. The properties of symmetry and completeness in the high dimensional space cannot to be determined by the χ^2 test solely, so these methods are not suitable for solving general clustering problems especially not for high-dimensional problems. The hypotheses test for goodness of fit only has the ability to suspect or not suspect the distribution hypotheses, but does not provide unequivocal evidence to accept or reject the distribution hypotheses [21].

Because of the shortcomings of the above methods, we are motivated to develop a clustering strategy that enables clustering algorithms to escape from local optima and adjust the cluster number automatically and simultaneously. In this paper, we develop novel strategies to achieve our goal. Based on *delete*, *split*, and *merge* operations, we enhance the EM algorithms for parameter estimation of a Gaussian mixture and construct a refined algorithm that can escape from local optima automatically and adjust the mixture number simultaneously. In addition, we have designed a justification criterion that combines the Euclidean distance and the Mahalanobis distance. This enables us to determine the independent characters of two Gaussian distributions and choose either the *split* or the *merge* operation among the clusters. Differing from the SMEM algorithm, the proposed algorithm does not hold the number of clusters, and the *split* and the *merge* operations show their specific potencies in their own proper time. Therefore, the number of clusters is adjusted in the proper moment while the *split* and the *merge* operations executes. Besides *split* and *merge* operations, in order to deal with unreasonable parameters during the iterative process, we apply the *delete* operation to maintain the normality of the iterative processes and also reduce the mixture number. Therefore, our algori-

thm not only escapes from local optima, but also yields a precise cluster number automatically and simultaneously. Also, the algorithm always guarantees that the iteration can be processed normally, even if the middle parameters are unreasonable. Experimental results and thorough evaluations are given for our findings.

The remainder of the paper is organized as follows. The way to determine the boundary of a Gaussian cluster is discussed in Sec. 2. An overview of the proposed algorithm is presented in Sec. 3. Determination of five cases is described in Sec. 4. The partial EM algorithm before the split operation is shown in Sec. 5. Some experimental results and discussion are given in Sec. 6. Finally, the conclusions and the direction of future work are presented in Sec. 7.

2. The Boundary of a Gaussian Cluster

In this paper, we focus on a mixture of non-overlapped and non-adjacent multivariate Gaussian functions. In order to determine two Gaussian clusters are overlapped or not, we must first define the boundary of a Gaussian cluster. We define the boundary of a Gaussian cluster as the set of vectors that have a specific identity value of the specific Gaussian function as (1). We can calculate only the exponent part of a Gaussian function, the square of Mahalanobis distance namely m^2 in (2), for simplifying. In fact, the set of the vectors that have identity Mahalanobis distance, i.e. *m,* against a Gaussian cluster make up a hyper-ellipsoid [21,22].

$$N_d\left(x \mid [u,\Sigma]\right) = \frac{1}{\sqrt{|\Sigma|(2\pi)^d}} e^{\frac{-1}{2}(x-u)^T \Sigma^{-1}(x-u)} \tag{1}$$

$$m^2 = d^2{}_{Mahalanobis}\left(x,[u,\Sigma]\right) = (x-u)^T \Sigma^{-1}(x-u) \tag{2}$$

With focusing on the hyper-ellipsoid composed by the set of vectors that have a specific identity value of a Gaussian function, the value of *m* in (2) decides the size of the hyper-ellipsoid. In a Gaussian cluster, the large-size hyper-ellipsoids contain more samples, and vice versa. A proper-size hyper-ellipsoid that contains sufficient samples can be defined as the boundary of a Gaussian cluster. In fact, *m* must be set as infinity to support the hyper-ellipsoids to contain all the possible samples. If so, all clusters are overlapped with each other and those are not expected boundaries for us. In fact, to decide the ratio of samples inside the hyper-ellipsoid can help us determine proper *m*. For the proposed algorithm, users must assign this ratio. In fact, assigning the ratio of samples inside the hyper-ellipsoid is much more comprehensible than assigning suitable *m*.

It can be proved that the *d*-dimensional Mahalanobis distance has a χ^2-distribution with *d* degree of freedom. That is, the probability of $(x - u)^T\Sigma^{-1}(x-u) \le \chi_d{}^2(\alpha)$ is $1 - \alpha$ [21], and this probability is identical with the ratio of samples inside the hyper-ellipsoid if the quantity of samples is large enough. If the ratio $1 - \alpha$ is designated by users, we can obtain the proper Mahalanobis distance *m* as $\sqrt{\chi_d^2(\alpha)}$ with incomplete gamma function [24]. Or otherwise, if you can gain a complete χ^2-distribution critical points table, you can get the $\sqrt{\chi_d^2(\alpha)}$ simply by the table. If not so, we can also calculate $1 - \alpha$ under specific Mahalanobis distance *m* and specific dimension *d* to create the mapping between $1 - \alpha$ and *m* under different dimensions by Mote Carlo simulation [23,24]. If we choose the latest method, we can let the half axes (unit eigenvectors multiplied the square roots of corresponding eigenvalues) of the hyper-ellipsoid be the new coordinate axes and the center of the hyper-ellipsoid be the new origin. Thus, the Gaussian function (1) is simplified as (3), where $\Sigma' = diag(1,\ldots,1)_d$, $u' = (0,\ldots,0)_d{}^T$ and both can be left out. Thus the ratio $1 - \alpha$ can be gained by the integration of (4). The mapping between $1 - \alpha$ and *m* under different dimensions can be easily created by Monte Carlo simulations via (5) [24]. No matter which method we choose, once the ratio of samples inside the hyper-ellipsoids has assigned under a specific dimension, we can obtain the proper Mahalanobis distance *m* to determine the boundaries of Gaussian clusters.

$$N_d(x') = \frac{1}{\sqrt{(2\pi)^d}} e^{\frac{-1}{2}[(x_1')^2+(x_2')^2+\cdots+(x_d')^2]} = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2}(x_i')^2} \tag{3}$$

$$1-\alpha = \int\cdots\int_{(x_1')^2+(x_2')^2+\cdots+(x_d')^2 \le s^2} \left[\prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} e^{\frac{-1}{2}(x_i')^2}\right] dx_1' dx_2' \cdots dx_d' \tag{4}$$

$$1-\alpha = \frac{\sum_{j=1}^{100,000} \left\{ \begin{array}{l} 1, \ \sum_{i=1}^{d} (x_i')^2 \le m^2, \ x_i' \text{ is a standard Gaussian random variable} \\ 0, \ otherwise \end{array} \right\}}{100,000} \quad (5)$$

3. The Proposed Algorithm - Delete/Split/Merge Operations

We develop enhancement strategies for the ***EM*** algorithm to resolve the problems of local optima and mixture number simultaneously. As shown in Figure 2, the developed algorithm, which extends the EM algorithms, consists of three extra key operation rules; namely, ***Delete*** a Gaussian component when it is not reasonable, ***Split*** a Gaussian component into two Gaussian components when it contains separated blobs, and ***Merge*** two Gaussian components into one Gaussian component when they overlap each other. Hereafter, we call it the ***DSMEM*** algorithm. For each iterative process of the proposed algorithm, the first rule is appended to the EM algorithm. When all the parameters of the appended algorithm are convergent (stable), the last two rules are applied recursively. After the conditions of these two rules no longer exist, the appended algorithm continues the iteration repeatedly. The entire clustering process does not stop until the appended algorithm is convergent and the conditions of these three principles are non-existent.

Although these three rules appear simple and intuitive, determining when to delete, split, and merge is crucial. The fundamental principles of our derivations are based on the concept that the data in a Gaussian distribution forms a cluster, and if the Gaussian distribution cannot perfectly describe a cluster, that cluster must be deleted, split into two clusters, or merged with another cluster. The conditions of these three rules are as follows.

Rule 1, ***deleting***

A multivariate Gaussian distribution is obsolete if the determinant of its covariance matrix equals to or closes to zero, and thus the whole EM algorithm is no longer calculable. A simple solution is to ignore all the intermediate results, renew the initial setting, and execute the EM algorithm anew. However, this simple approach would be very frustrating if this obsolete situation

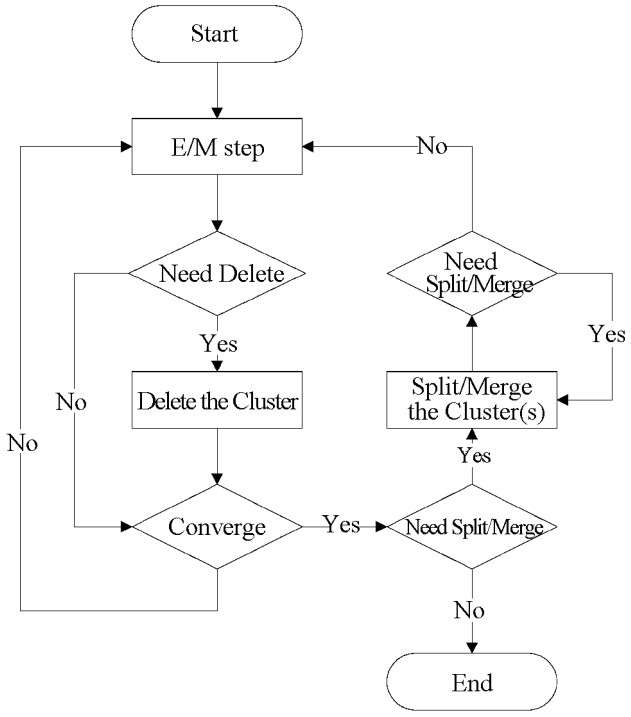

Figure 2. The flow chart of the proposed algorithm.

were to occur continually. We prefer, therefore, to delete the unreasonable components directly and then the EM algorithm is calculable again. We also delete a Gaussian component that has too few samples, and by this way we can speed up the convergence of the whole algorithm and avoid the effects of noisy samples. In practice, users can assign the threshold of the minimum sample number heuristically. If users do not determine the threshold, the clusters with fewer samples will not be deleted. However, deleting irregular and too-small clusters is reasonable in order to guarantee that the algorithm can perform normally and efficiently, even though the algorithm has no choice but to generate some unreasonable components.

Rules 2 and 3, ***splitting*** and ***merging***

If a cluster can be split into two non-adjacent clusters, it should be separated into two clusters. Conversely, if two clusters are adjacent or overlapped, they should be merged into one cluster. However, two clusters can have five kinds of relationship, which we need to consider, see Figure 3. Two clusters are distant in Figure 3(a) and 3(b). Two clusters are overlapped or adjacent in Figure 3(c), 3(d), and 3(e). The cases in Figure 3(b) and 3(c) are a bit ambiguous. The difference is that the tangent or intersection point of two ellipsoids is either outside of, or within, the lines connecting the centers of the two ellipsoids. Ba-

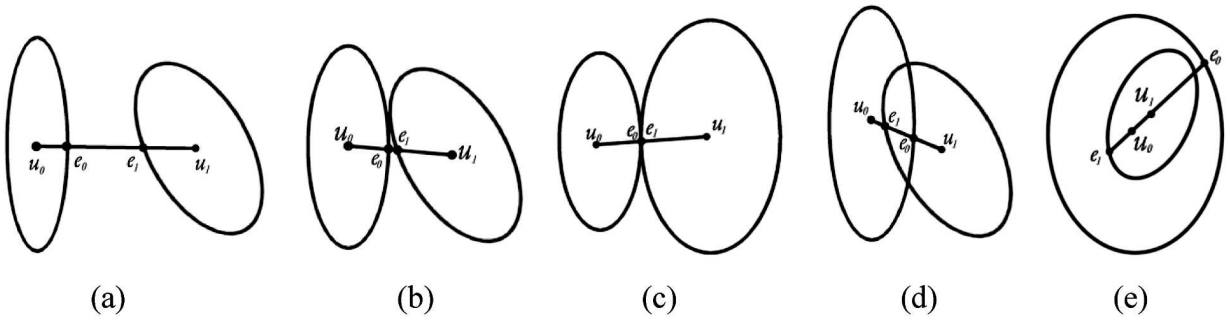

Figure 3. Five main kinds of relationships of two clusters.

sed on the shapes of these two ellipsoids, we subjectively consider that the two clusters are nonadjacent in Figure 3(b) and the two clusters are adjacent in Figure 3(c). Therefore, we need to separate one cluster into two clusters in the first two cases, and we need to combine two clusters into one cluster in the last three cases. In contrast to the SMEM algorithm [18] through split/merge conditions, we only use one principle to make decisions about split and merge, whereas SMEM uses two entirely distinct principles to make the same decisions. This is one of the significant differences between our algorithm and the SMEM algorithm.

When the EM algorithm is convergent, the proposed algorithm will check whether all Gaussian clusters need to be separated, or whether all pair Gaussian clusters need to be combined. If any split and merge operator is functioning, the EM algorithm will continue. Otherwise, the parameter estimation is complete.

There are still two problems that must be solved when the proposed algorithm is used. The first is how to determine which case of two clusters any pair of Gaussian clusters belongs to. The second problem is how to generate two suitable sub-clusters to help us decide whether a Gaussian cluster should be split or not. The solutions for these two problems will be presented in the next two sections.

4. The Determination for Split and Merge Operations

As bounded Mahalanobis distance m is decided as Sec. 2, we can look at Figure 3 more carefully. As the equations shown in (6), the vector from u_0 to e_0 is a multiple of the vector from u_0 to u_1, and the vector from u_1 to e_1 is a multiple of the vector from u_1 to u_0. The factors k_0 and k_1 in (6) are both positive. Moreover, the Mahalanobis distances between e_0 and the Gaussian cluster with the mean u_0 and the Mahalanobis distances between e_1 and the Gaussian cluster with the mean u_1 are both m exactly as (7). With the known values of m, u_0, u_1, Σ_0, and Σ_1, the values of k_0 and k_1 can be computed using (8).

$$e_0 - u_0 = k_0(u_1 - u_0),\ \ e_1 - u_1 = k_1(u_0 - u_1) \tag{6}$$

$$\begin{aligned} d^2_{Mahalanobis}(e_0, u_0) &= (e_0 - u_0)^T \Sigma_0^{-1}(e_0 - u_0) \\ &= k_0^2 (u_1 - u_0)^T \Sigma_0^{-1}(u_1 - u_0) = m^2 \\ d^2_{Mahalanobis}(e_1, u_1) &= (e_1 - u_1)^T \Sigma_1^{-1}(e_1 - u_1) \\ &= k_1^2 (u_0 - u_1)^T \Sigma_1^{-1}(u_0 - u_1) = m^2 \end{aligned} \tag{7}$$

$$\begin{aligned} k_0 &= \left[\frac{m^2}{(u_1 - u_0)^T \Sigma_0^{-1}(u_1 - u_0)}\right]^{1/2}, \\ k_1 &= \left[\frac{m^2}{(u_0 - u_1)^T \Sigma_1^{-1}(u_0 - u_1)}\right]^{1/2} \end{aligned} \tag{8}$$

In the five cases described in the previous section, if the Euclidean distance between u_1 and u_0 (9) is larger than the sum of the Euclidean distance between e_0 and u_0 and the Euclidean distance e_1 and u_1 (10), the two clusters are distant; otherwise, the two clusters are adjacent or overlapped. Comparing with (9) and (10), we can make a short conclusion: if the value of d_{mix} defined in (11) is less than 0, the two clusters are adjacent; otherwise, the two clusters are distant.

$$d_{Euclidean}(u_0, u_1) = [(u_1 - u_0)^T (u_1 - u_0)]^{1/2} \tag{9}$$

$$\begin{aligned} &d_{Euclidean}(e_0, u_0) + d_{Euclidean}(e_1, u_1) \\ &\quad = (k_0 + k_1)[(u_1 - u_0)^T (u_1 - u_0)]^{1/2} \end{aligned} \tag{10}$$

$$d_{mix}([u_0, \Sigma_0], [u_1, \Sigma_1]) = 1 - (k_0 + k_1) \tag{11}$$

5. The Partial EM Algorithm Before the Split Operation

As mentioned at the end of Sec. 3, before deciding whether a Gaussian cluster should be split or not, we should produce two sub-clusters from each cluster. For this purpose, it is intuitive to use the pure EM algorithm and set the number of clusters as 2. As a result, we will obtain two hypothetical clusters to determine whether the original cluster needs to split or not via the approach introduced in the Sec. 4. Nevertheless, we have to designate several initial parameters of these two sub-clusters, i.e., the means and the covariance matrices before further processing this partial EM algorithm. If the partial EM algorithm starts with random settings, it is still possible that the algorithm will be trapped in the local optima. Several researchers have investigated into the solutions of this elementary problem in order to avoid the local optima for only two clusters [18,19,25]. It seems much easier than the cases with more than two clusters, but if it is not satisfactorily arranged on a sound basis, the full algorithm will be completely annihilated because the *split* operation will be not executed correctly. To solve this problem, we designed a two-step approach that chooses the best one from d different candidates for d dimensional samples.

In Step 1, we designate d pairs of means and d covariance matrices to collocate as d different candidates. Because every covariance matrix of the original cluster is symmetrical, d different corresponding eigenvectors and eigenvalues can be derived to generate means and covariance matrices. Each pair of means is symmetric according to the mean of the original cluster and is generated by the mean of the original cluster adds and subtracts one of the eigenvectors multiplied the square root of the corresponding eigenvalue as (12); and further, each covariance matrix is generated by (14) to gain two half-sized sub-clusters because the original covariance matrix can be orthogonally diagonalized as (13) since it is symmetrical. Each initial candidate has two sub-clusters using one pair of means and identical corresponding covariance matrices. Therefore, these two sub-clusters are symmetric according to the original mean and have identical size and contour.

$$u_{p1} = u + \lambda_i^{1/2} v_i,\ u_{p2} = u - \lambda_i^{1/2} v_i,\ i = 1, 2 \cdots, d \tag{12}$$

$$\Sigma = \Phi \Lambda \Phi^T,\ \Lambda = diag(\lambda_1 \cdots \lambda_d),\ \Phi = \left[v_1 \cdots v_d \right] \tag{13}$$

$$\Sigma_{p1} = \Sigma_{p2} = \Phi \Lambda' \Phi^T,\ \Lambda' = diag(\cdots \lambda_i / 4 \cdots),\ i = 1, 2 \cdots, d \tag{14}$$

In Step 2, the best one of d different candidates generated in step 1 will be chosen. First, we must test and verify which sub-cluster all samples of the original cluster belong to. If the Mahalanobis distance between a sample and a Gaussian sub-cluster is shorter, the sample belongs to the sub-cluster. Therefore, two new groups of samples are composed. After that, we can calculate two new means and two new corresponding covariance matrices according to these two groups of samples for each candidate, and then we can calculate the d values of d_{mix} (11) for d candidates individually. If any d_{mix} are larger than zero, we choose the candidate that has maximum d_{mix} as the initial setting of the partial EM algorithm. If all d_{mix} are smaller than zero, d pure EM processes started with new means and covariance matrices are executed. With these d initial settings, the outcome of these pure EM processes that has maximum d_{mix} will be the result of the partial EM algorithm. Please notice that it is not necessary to execute the *split* operation when all d_{mix} are still smaller than zero after these pure EM processes because we find no non-adjacent outcome from the d candidates for this situation.

In our experience, above two-step approach is efficient for various cases. Two experiments in *2*-dimensional space, for example, are shown in Figure 4. In the case of Figure 4(a), we can decide the initial setting of the partial EM algorithm in Step 2. In the case of Figure 4(b), the two candidates generate the same results for partial EM algorithm. In these two cases the original cluster must be split to fit the practice.

6. Experimental Results and Evaluations

The proposed algorithm can escape from the local optimum. Theoretically, we do not prove that the proposed algorithm is guaranteed to converge to the global optimum. But, based on various test samples and thorough evaluations presented later in this section, the proposed algorithm yields very satisfactory results; and further, the proposed algorithm can automatically adjust the

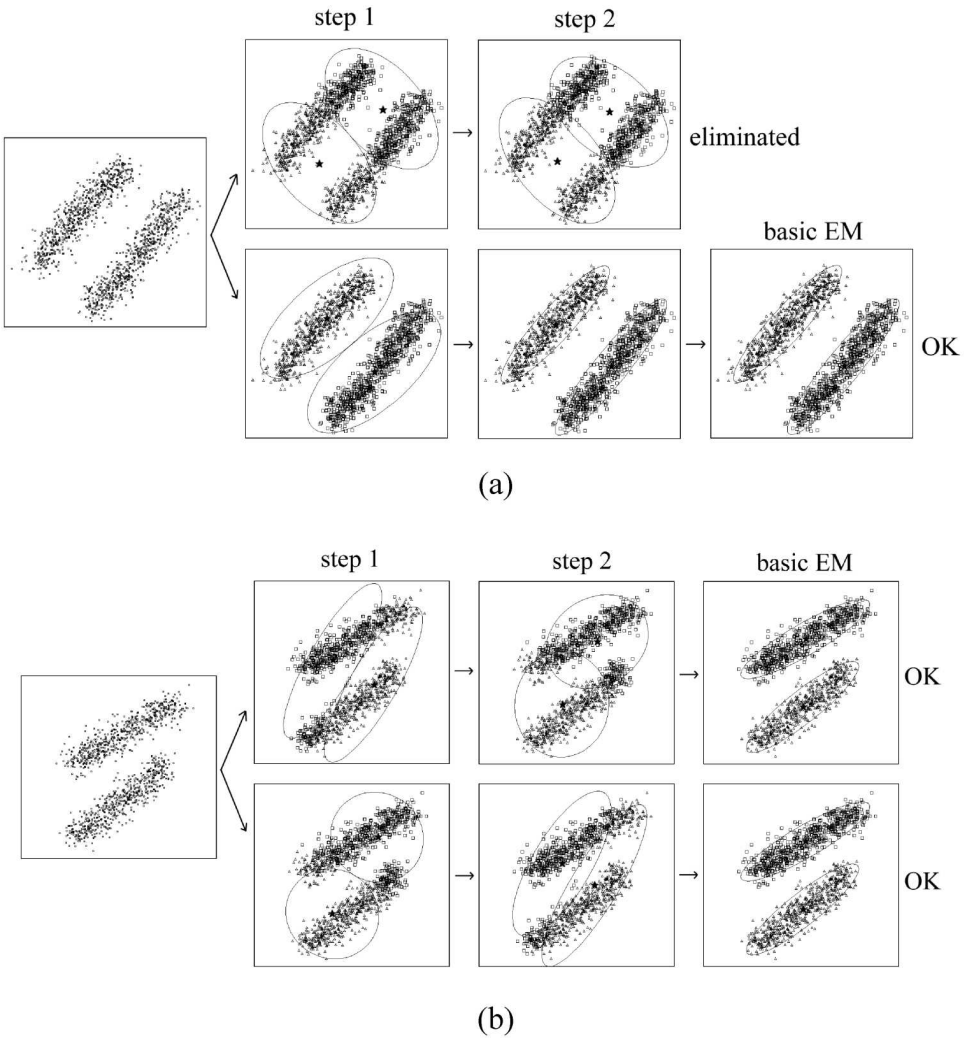

Figure 4. Demonstrations of the partial EM algorithm.

number of clusters. Of course, if the initial number of clusters is near the real one, it will be faster to converge. In our experiments, even if the initial number of clusters is far from the real one, the results are still correct after more iterative processes. We now present our experimental results and evaluations.

The first experimental data set is named as the satellite data set, which has 692 samples from 7 clusters including one big planet cluster and 6 satellite clusters as shown in Figure 5(a). For this data set, the demonstrated results of the fuzzy c-means algorithm with Euclidean distance are shown in Figure 5(b) and Figure 5(c). These two results start with different initial settings individually. Both of these results are unexpected. In Figure 5(b), the final result falls into the local optimum. Two pairs of satellite clusters are combined into one cluster, and the planet cluster is separated into three fragmented clusters. The best result of the fuzzy c-means algorithm with Euclidean distance is shown in Figure 5(c), but the planet cluster is unacceptably narrow. That is, the satellite clusters attract the surrounding samples of the planet cluster. This shows that the fuzzy c-means algorithm (or the k-means algorithm) with Euclidean distance frequently achieve poor results in this kind of data sets, even though the cluster number is correct, the initial setting is good, and so-called lobal optimization is gained.

With the same data set, the results in Figure 1(b) and Figure 1(c) are the results of the pure EM algorithm with different initial settings individually. These results show that the EM algorithm guarantees to converge toward the local optima, but does not guarantee to converge toward the correct results. After many tries and tests, we disco-

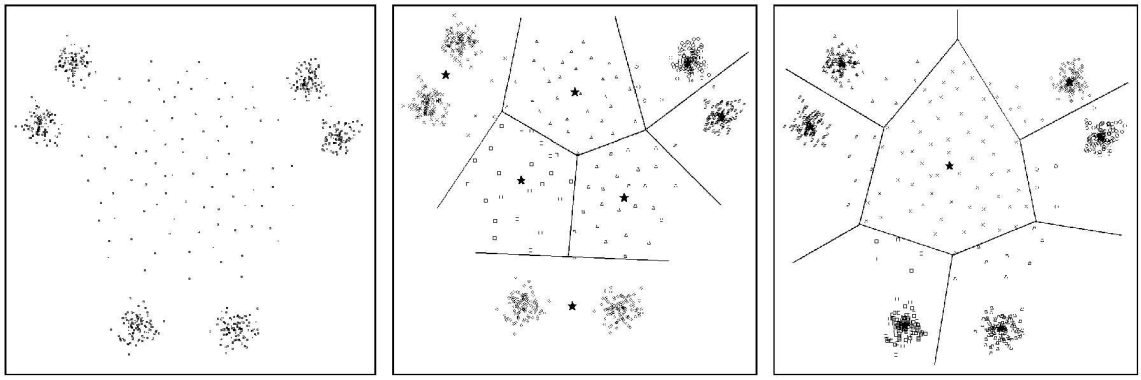

Figure 5. The satellite experiments.

ver that the EM algorithm hardly avoid locals optima in this kind of data sets; that is to say, the traditional EM algorithm usually falls into local optima with starting from random initial-settings in this kind of data sets. However, by using our proposed algorithm, even if the wrong numbers of clusters are assigned and the improper initial settings are given, the result is still exactly correct as Figure 1(a). This explains that the proposed algorithm is considerably robust.

Another data set is the wine recognition data from the UCI repository of machine learning databases [26]. The data contains one hundred and seventy-eight 13-dimensional data that are the results of a chemical analysis of three kinds of wines produced in the same region of Italy. This data set has been utilized in [27,28] to test the performance of classifiers. In our experiment, we use the technique of Latent Semantic Indexing [29] through singular value decomposition (SVD) to reconstruct representative properties, which contain only 6-dimensional data because the raw data have lots of redundant properties. We ignore the class label, i.e., the number of wines, when testing the proposed clustering algorithm. Even though we incorrectly set the initial number of clusters as 6, our proposed algorithm automatically adjusts the number of clusters to 3 correctly. The result of the proposed algorithm is compared with the original classification, and the precision rates are all above 92.96% as shown in Table 1. This demonstrates that the proposed algorithm can deal with high-dimensional samples efficiently.

In addition, in order to investigate the performance, we generated large sets of data to compare the proposed algorithm to the fuzzy c-means algorithm with Euclidean distance and the pure EM algorithm. The numbers of clusters are set from 2 to 10. For each number of clusters, we randomly generate 100 data sets in 2-dimensional space, and each cluster has 100 samples. Besides, all clusters are overlapped with each other. Each algorithm has only one chance to execute for every data set. We calculate the precision for all tests. One algorithm succeeds with a data set only if the numbers of samples of all final clusters are between 95 and 105, and at least 90 samples in every final cluster belong to the same original cluster. One algorithm fails if any final cluster does not fit the above two conditions. Therefore, success in this investigation is considerably difficult.

Table 1. Experimental results of wine recognition data by using the proposed algorithm

Cluster No.	Sample count	Correct count	Precision
1	59	58	98.31%
2	71	66	92.96%
3	48	47	97.92%

The evaluation results are given in Figure 6. We discover that the average precisions of all algorithms are comparatively high when the cluster number is small, but decreases significantly for the fuzzy c-means and the EM algorithms when the cluster number increases. In the case with 10 clusters, the average precision rates are lower than 60% and 70% for the fuzzy c-means and EM algorithms respectively. However, the proposed algorithm, DSMEM, has fairly high precision over a range of cluster numbers with an average precision rate of about 95%.

When using the fuzzy c-means algorithm and the EM algorithm, we must designate the number of clusters. All results stated in the previous paragraph are based on the given correct number of clusters, but it is not necessary

for the proposed algorithm in fact. In order to explain this property, we perform two distinct tests. One is with given correct numbers of clusters and denoted as DSMEM; another is without designated numbers of clusters and denoted as DSMEM2. The latter is executed with random numbers of clusters from 2 to 20 for each test. As shown in Figure 6, it is clear that the proposed algorithm has high precision without designated numbers of clusters. The average precision rates are always higher than 90% by using the proposed algorithm, DSMEM2. Although the precision rate is slightly lower than DSMEM, it is obviously better than the fuzzy c-means algorithm and pure EM algorithm.

In order to explain the optimizing process of the proposed algorithm, we demonstrate parts of an entire executive process in Figure 7. There are 10 clusters in this test sample. In the Figure 7(a), the EM algorithm is initiated by 10 random set parameters. After 7 steps, 3 unreasonable clusters are *deleted*, as shown in Figure 7(b). In the 31^{st} step, the EM algorithm is trapped at the local optimum, as shown in Figure 7(c). In the 32^{nd} step, 3 clusters are *split* into 3 pairs of clusters, as shown in Figure 7(d). In the 43^{rd} step, 1 cluster is *split* into 2 clusters again, as shown in Figure 7(e); therefore, there are 11 clusters at this point. Finally, in the 59^{th} step, 2 clusters are *merged* into 1 cluster and the correct result is obtained, as shown in Figure 7(f). We can see that these three operations, i.e., *delete*, *split*, and *merge*, are applied reciprocally and seamlessly to gain the correct result successfully.

7. Conclusions and Future Works

In this paper, we have discussed three problems of the EM algorithm in estimating the parameters of Gaussian mixtures. The first problem is that an improper initial setting will induce the EM algorithm towards local

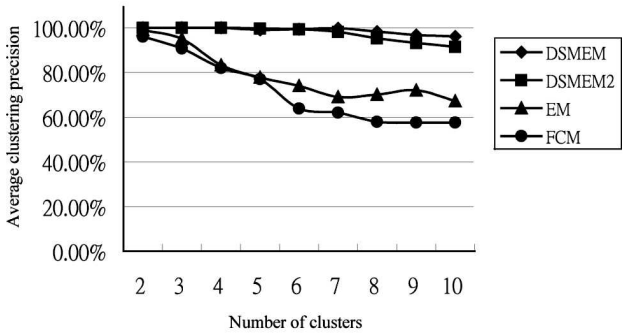

Figure 6. Evaluation of different clustering algorithms, fuzzy c-means (FCM), EM algorithm, and the proposed DSMEM algorithm with 100 randomly generated samples per cluster number. FCM, EM, and DSMEM are calculated with a known cluster number, but DSMEM 2 is calculated with a random cluster number from 2 to 20 for each test.

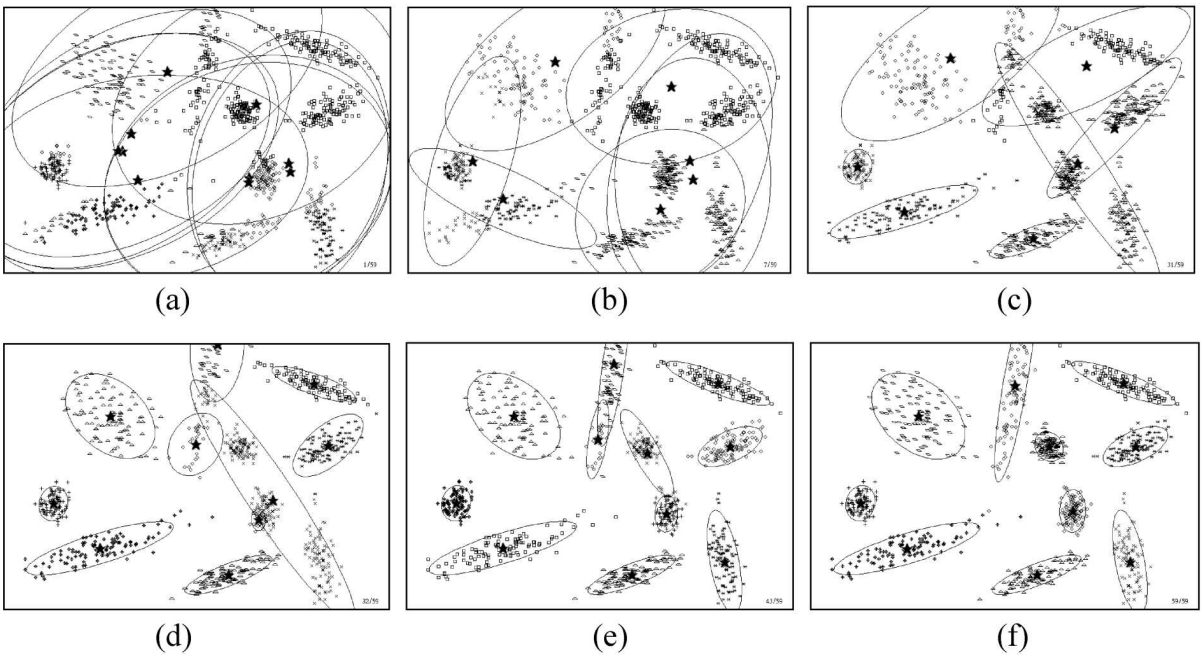

Figure 7. An optimizing process.

optimum results. The second problem is that the EM algorithm cannot decide the correct mixture number (the number of clusters). The third problem is that the EM algorithm cannot guarantee to produce the correct results if the above two problems are not solved simultaneously. We therefore propose the DSM (*Delete*, *Split*, and *Merge*) algorithm to solve these three problems. The experiment results show that our algorithm can escape from the recognized local optima and adjust the number of clusters automatically, simultaneously, and effectively. However, the idea of split and merge to solve local optima problem for clustering is not new. Several researchers have discussed it [18,30]. However, we discover that no one uses this idea and unites with the delete operation to solve the problems of unknown number of clusters and local optima simultaneously for parameter estimation of Gaussian mixture. This is the most important contribution of this paper.

The split and merge rules are not only useful for the purpose discussed in this paper. They can also be utilized by other situations that need to determine the factors of separation and combination, for example, the multi-center clustering algorithm [31], hierarchical clustering methods, and the SVM clustering algorithm [32]. In addition, if the number of clusters is already known and cannot be modified, the proposed algorithm could still work; however, the balance between decreasing a cluster (*delete* and *merge* rules) and increasing a cluster (*split* rule) requires further study. Also, there are cases where the Gaussian distribution may not govern directly, and the samples cannot be simply grouped by hyper-ellipsoids. In other words, there are irregular shapes, e.g., concave or doughnut shapes, in the hyperspace. In such cases, the criterion developed in this paper needs to be modified. We will also address these issues in our future work.

References

[1] Moon, Todd K., "The Expectation-Maximization Algorithm," *IEEE Signal Processing Magazine,* pp. 47–60 (1996).

[2] Dempster, A. P., Laird, N. M. and Rubin, D. B., "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society Series B,* Vol. 39, pp. 1–38 (1977).

[3] Bilmes, Jeff A. "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," Technical Report TR-97-021, Computer Science Division, Department of Electrical Engineering and Computer Science, U. C. Berkeley (1998).

[4] Ball, G. and Hall, D. "A Clustering Technique for Summarizing Multivariate Data," *Behavioral Science,* Vol. 12, pp. 153–155 (1967).

[5] Dunn, J. C., "A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-separated Clusters," *Journal Cybernetics,* Vol. 3, pp. 95–104 (1974).

[6] Frank Hoppner, Frank Klawonn, Rudolf Kruse and Thomas Runkler, *Fuzzy Cluster Analysis, Methods for Classification, Data Analysis and Image Recognition*, Wiley, New York, U.S.A. (2000).

[7] Bezdek, James C., Pattern Recognition with Fuzzy Objective Function Algorithm. Plenum, New York, U.S.A. (1981).

[8] Gustafson, Donald E. and Kessel, William C., "Fuzzy Clustering with a Fuzzy Covariance Matrix," *In Proc. of the IEEE Conference on Decision and Control,* pp. 761–766 (1979).

[9] Isak Gath and Geva, Amir. B., "Unsupervised Optimal Fuzzy Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol. 11, pp. 773–781 (1989).

[10] Jorma Rissanen, "Modeling by Shortest Data Description," *Automatica*, Vol. 14, pp. 465–471 (1978).

[11] Gideon Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics,* Vol. 6, pp. 461–464 (1978).

[12] Chris Fraley and Raftery, Adrian E., "How Many Clusters? Which Clustering Method? Answers via Model-based Cluster Analysis," *The Computer Journal,* Vol. 41, pp. 578–588 (1998).

[13] Peter Cheeseman and John Stutz, *Bayesian Classification (AutoClass): Theory and Results,* In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and U. Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press. 153–180 (1996).

[14] Pal, Nikhil R. and Bezdek, James C., "On Cluster Validity for the Fuzzy c-means Model," *IEEE Trans. Fuzzy Systems,* Vol. 3, pp. 370–379 (1995).

[15] Lawrence O. Hall, Ibrahim Burak Ozyurt, and Bezdek, James C., "Clustering with a Genetically Optimized Approach," *IEEE Transactions on Evolutionary Computation,* Vol. 3, pp. 103–112 (1999).

[16] Nir Friedman, Matan Ninio, Itsik Pe'er, and Tal Pupko, "A Structural EM Algorithm for Phylogenetic Inference," *Journal of Computational Biology.* Vol. 9, pp. 331–353 (2002).

[17] Schachter, B., Davis, L. and Rosenfeld, A., "Some Experiments in Image Segmentation by Clustering of Local Feature Values," *Pattern Recognition,* Vol. 11, pp. 19–28 (1979).

[18] Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani and Hinton, Geoffrey E. "SMEM Algorithm for Mixture Models," *Neural Computation,* Vol. 12, pp. 2109–2128 (2000).

[19] Nikos Vlassis and Aristidis Likas, "A Greedy EM Algorithm for Gaussian Mixture Learning," *Neural Processing Letters,* Vol. 15, pp. 77–87 (2002).

[20] Henry. C. Thode, Jr., Stephen J. Finch and Nancy R. Mendell, "Simulated Percentage Points for the Null Distribution of the Likelihood Ratio Test for a Mixture of Two Normals," *Biometrics,* Vol. 44, pp. 1195–1201 (1988).

[21] Richard A. Johnson and Dean W. Wichern, *Applied Multivariate Statistical Analysis, Third Edition,* Prentice-Hall (1992).

[22] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition,* Academic Press (1999).

[23] Gentle, James E. Random Number Generation and Monte Carlo Methods, Springer (1998).

[24] William H. Press, Brian P. Flannery, Saul A. Teukolsk and William T., Vetterling, *Numerical recipes in C,* Cambridge University Press (1988).

[25] Yoav Freund and Yishay Mansour, "Estimating a Mixture of Two Product Distributions," *ACM Conference on Computational Learning Theory,* pp. 53–59 (1999).

[26] Blake, C. L. and Merz, C. J., "UCI Repository of Machine Learning Databases," University of California, Irvine, Department of Information and Computer Sciences (1998).

[27] Aeberhard, S., Coomans, D. and de Vel, O., "Comparison of Classifiers in High Dimensional Settings," Technical Report No. 92–02, (1992), Department of Computer Science and Department of Mathematics and Statistics, James Cook University of North Queensland (1992).

[28] Aeberhard, S., Coomans, D. and de Vel, O., "The Classification Performance of RDA," Technical Report No. 92–01, (1992), Department of Computer Science and Department of Mathematics and Statistics, James Cook University of North Queensland (1992).

[29] Soumen Chakrabarti, *Mining the Web, Discovering Knowledge from Hypertext Data,* Morgan Kaufmann Publishers (2003).

[30] Zhihua Zhang, Chibiao Chen, Jian Sun and Kap Luk Chan, "EM Algorithms for Learning Gaussian Mixture Models with Split-and-Merge Operation," *Pattern Recognition,* Vol. 36, pp. 1973–1983 (2003).

[31] Tao, C. W. "Unsupervised Fuzzy Clustering with Multi-center Clusters," *Fuzzy Sets and Systems,* Vol. 128, pp. 305–322 (2002).

[32] Asa Ben-Hur, David Horn, Hava T. Siegelmann and Vladimir Vapnik. "Support Vector Clustering," *Journal of Machine Learning Research,* Vol. 2, pp. 125–137 (2002).

Manuscript Received: Dec. 12, 2005
Accepted: Feb. 27, 2006