Non-convex clustering using expectation maximization algorithm with rough set initialization

Pabitra Mitra [a,*], Sankar K. Pal [b], Md Aleemuddin Siddiqi [b]

[a] *Machine Intelligence Unit, Indian Statistical Institute, Calcutta 700035, India*
[b] *Department of Statistics and Applied Probability, University of California, Santa Barbara, CA 93106, USA*

Abstract

An integration of a minimal spanning tree (MST) based graph-theoretic technique and expectation maximization (EM) algorithm with rough set initialization is described for non-convex clustering. EM provides the statistical model of the data and handles the associated uncertainties. Rough set theory helps in faster convergence and avoidance of the local minima problem, thereby enhancing the performance of EM. MST helps in determining non-convex clusters. Since it is applied on Gaussians rather than the original data points, time required is very low. These features are demonstrated on real life datasets. Comparison with related methods is made in terms of a cluster quality measure and computation time.

Keywords: Mixture modelling; Minimal spanning tree; Rough knowledge encoding; Data mining; Pattern recognition

1. Introduction

The clustering problem has broad appeal and usefulness as one of the steps in exploratory data analysis (Jain et al., 1999). It is an important task in several data mining applications including document retrieval, image/spatial data segmentation, market analysis. Data mining applications place the following two primary requirements on clustering algorithms: scalability or speed of computation on large data sets (Bradley et al., 1998; Zhang et al., 1996) and non-presumption of any canonical data properties like convexity.

Non-hierarchical clustering algorithms, can be grouped broadly into two categories. One is based on iterative refinement of cluster parameters, optimizing some criterion function or likelihood of some probabilistic model (e.g., k-means (Jain et al., 1999), mixture of Gaussians (Dempster et al., 1977)). The second is graph-theoretic clustering, where each cluster represents a subgraph of a graph of the entire data. One of the well known graph-theoretic clustering is based on the construction of the minimal spanning tree (MST) of the data (Zahn, 1971). Both the approaches have their advantages and disadvantages and cannot directly be applied for data mining. While the iterative refinement schemes like k-means and expectation-maximization (EM) are fast and easily scalable to large databases (Bradley et al., 1998, 1999), they can only produce convex clusters and

* Corresponding author.
E-mail addresses: pabitra_r@isical.ac.in (P. Mitra), sankar@isical.ac.in (S.K. Pal), siddiqi@math.ucsb.edu (M.A. Siddiqi).

are sensitive to initialization of the parameters. The graph-theoretic methods can model arbitrary shaped clusters, but are slow and sensitive to noise. It may be noted that, the advantages of one are complimentary in overcoming the limitations of the other, and vice versa.

A general method of clustering using statistical principles is to represent the probability density function of the data as a *mixture model*, which asserts that the data is a combination of k individual component densities (commonly Gaussians), corresponding to k clusters. The task is to identify, given the data, a set of k populations in the data, and provide a model (density distribution) for each of the populations. The EM algorithm (Cherkassky and Mulier, 1998) is an effective and popular technique for estimating the mixture model parameters. It iteratively refines an initial cluster model to better fit the data and terminates at a solution which is locally optimal for the underlying clustering criterion (Dempster et al., 1977). Log-likelihood is used as the objective function which measures how well the model fits the data. Like other iterative refinement clustering methods, including the popular k-means algorithm, the EM algorithm is fast and its scalable versions are available (Bradley et al., 1999). An advantage of EM over k-means is that it provides a statistical model of the data and is capable of handling the associated uncertainties. However, a problem arising due to its iterative nature is convergence to a local rather than the global optima. It is sensitive to initial conditions and is not robust. To overcome the initialization problem, several methods for determining 'good' initial parameters for EM have been suggested, mainly based on subsampling, voting and two stage clustering (Meila and Heckerman, 1998). However, most of these methods have heavy computational requirement and/or are sensitive to noise.

Rough set theory (Pawlak, 1991; Komorowski et al., 1997) provides an effective means for classificatory analysis of data tables. A principal goal of rough set theoretic analysis is to synthesise or construct approximations (upper and lower) of sets concepts from the acquired data. The key concepts here are those of "information granule" and "reducts". Information granule formalises the concept of finite precision representation of objects in real life situations, and reducts represent the *core* of an information system (both in terms of objects and features) in a granular universe. An important use of rough set theory has been in generating logical rules for classification and association (Skowron and Rauszer, 1992). These logical rules correspond to different important regions of the feature space, which represent data clusters.

In this article we exploit the above capability of the rough set theoretic logical rules to obtain initial approximate mixture model parameters. The crude mixture model, after refinement through EM, leads to accurate clusters. Here, rough set theory offers a fast and robust (noise insensitive) solution to the initialization and local minima problem of iterative refinement clustering. Also the problem of choosing the number of mixtures is circumvented, since the number of Gaussian components to be used is automatically decided by rough set theory.

The problem of modelling non-convex clusters is addressed by constructing a MST with each Gaussian as nodes and Mahalanobis distance between them as edge weights. Since graph-theoretic clustering is performed on the Gaussian models rather than the individual data points and the number of models are much less than the data points, the computational time requirement is significantly small. A (non-convex) cluster obtained from the graph is a particular subset of all the Gaussians used to model the data.

Experiments were performed on some real life and artificial non-convex data sets. Comparison is made both in terms of a cluster quality index (Pal et al., 2000) and computational time. It is found that rough set enhances the performance of both k-means and EM based algorithms. It is also observed that EM performs better than k-means algorithm.

The organisation of the article is as follows: First we describe the EM algorithm for mixture modelling. Then we present some relevant concepts from rough set theory and the methodology for obtaining initial EM parameters. The method of obtaining non-convex clusters from the mixture model using MST is discussed next. Finally, experimental results are presented.

2. Mixture model estimation via the EM algorithm

The mixture model approximates the data distribution by fitting k component density functions f_h, $h = 1, \ldots, k$ to a data set D having m patterns and d features. Let $x \in D$ be a pattern, the mixture model probability density function evaluated at x is:

$$p(x) = \sum_{h=1}^{k} w_h f_h(x|\phi_h). \tag{1}$$

The weights w_h represent the fraction of data points belonging to model h, and they sum to one ($\sum_{h=1}^{k} w_h = 1$). The functions $f_h(x|\phi_h)$, $h = 1, \ldots, k$ are the component density functions modelling the points of the hth cluster. ϕ_h represents the specific parameters used to compute the value of f_h (e.g., for a Gaussian component density function, ϕ_h is the mean and covariance matrix).

For continuous data, Gaussian distribution is the most common choice for component density function. This is motivated by a result from density estimation theory stating that any distribution can be effectively approximated by a mixture of Gaussians (Scott, 1992). The multivariate Gaussian with d-dimensional mean vector μ_h and $d \times d$ covariance matrix Σ_h is:

$$f_h(x|\mu_h, \Sigma_h) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_h|}} \exp\left(-\frac{1}{2}(x-\mu_h)^{\mathrm{T}} \times (\Sigma_h)^{-1}(x-\mu_h) \right). \tag{2}$$

The quality of a given set of parameters $\Phi = \{(w_h, \mu_h, \Sigma_h), h = 1, \ldots, k\}$ is determined by how well the corresponding mixture model fits the data. This is quantified by the log-likelihood of the data, given the mixture model:

$$L(\Phi) = \sum_{x \in D} \log\left(\sum_{h=1}^{k} w_h f_h(x|\mu_h, \Sigma_h) \right). \tag{3}$$

The EM begins with an initial estimation of Φ and iteratively updates it such that $L(\Phi)$ is non-decreasing. We next outline the EM algorithm.

EM algorithm: Given a dataset D with m patterns and d continuous features, a stopping tolerance $\epsilon > 0$ and mixture parameters Φ^j at iteration j, compute Φ^{j+1} at iteration $j+1$ as follows:

Step 1 (E-Step): For pattern $x \in D$. Compute the membership probability of x in each cluster $h = 1, \ldots, k$:

$$w_h^j(x) = \frac{w_h^j f_h(x|\mu_h^j, \Sigma_h^j)}{\sum_i w_i^j f_i(x|\mu_i^j, \Sigma_i^j)}.$$

Step 2 (M-Step): Update mixture model parameters.

$$w_h^{j+1} = \sum_{x \in D} w_h^j(x),$$

$$\mu_h^{j+1} = \frac{\sum_{x \in D} w_h^j(x) x}{\sum_{x \in D} w_h^j(x)},$$

$$\Sigma_h^{j+1} = \frac{\sum_{x \in D} w_h^j(x)\left(x-\mu_h^{j+1}\right)\left(x-\mu_h^{j+1}\right)^{\mathrm{T}}}{\sum_{x \in D} w_h^j(x)},$$

$$h = 1, \ldots, k.$$

Stopping criteria: If $|L(\Phi^j) - L(\Phi^{j+1})| \leqslant \epsilon$, Stop. Else set $j \leftarrow j+1$ and Go To *Step 1*. $L(\Phi)$ is given in Eq. (3).

3. Rough set initialization of mixture parameters

In this section we describe the methodology for obtaining crude initial values of the parameters (Φ) of the mixture of Gaussians used to model the data. The parameters are refined further using EM algorithm described in the previous section. The methodology is based on the observation that 'reducts' obtained using rough set theory represent crude clusters in the feature space.

Let us first present some preliminaries of rough set theory which are relevant to this article. For details one may refer to Pawlak (1991) and Skowron and Rauszer (1992).

3.1. Definitions

An *information system* is a pair $\mathscr{S} = \langle U, A \rangle$, where U is a non-empty finite set called the *universe* and A a non-empty finite set of *attributes*. An attribute a can be regarded as a function from the domain U to some value set V_a.

An information system may be represented as an *attribute-value table*, in which rows are labeled by objects of the universe and columns by the attributes.

With every subset of attributes $B \subseteq A$, one can easily associate an equivalence relation I_B on U:

$$I_B = \{(x,y) \in U : \text{for every } a \in B, a(x) = a(y)\}.$$

Then $I_B = \cap_{a \in B} I_a$.

If $X \subseteq U$, the sets $\{x \in U : [x]_B \subseteq X\}$ and $\{x \in U : [x]_B \cap X \neq \emptyset\}$, where $[x]_B$ denotes the equivalence class of the object $x \in U$ relative to I_B, are called the *B-lower* and *B-upper approximation* of X in $\mathscr{S}$ and denoted by $\underline{B}X$, $\overline{B}X$ respectively.

$X(\subseteq U)$ is *B-exact* or *B-definable* in $\mathscr{S}$ if $\underline{B}X = \overline{B}X$. It may be observed that $\underline{B}X$ is the greatest B-definable set contained in X, and $\overline{B}X$ is the smallest B-definable set containing X.

We now define the notions relevant to knowledge reduction. The aim is to obtain irreducible but essential parts of the knowledge encoded by the given information system; these would constitute *reducts* of the system. So one is, in effect, looking for *maximal* sets of attributes taken from the initial set (A, say), which induce the *same* partition on the domain as A. In other words, the essence of the information remains intact, and superfluous attributes are removed. Reducts have been nicely characterized in (Skowron and Rauszer, 1992) by *discernibility matrices* and *discernibility functions*. Consider $U = \{x_1, \ldots, x_n\}$ and $A = \{a_1, \ldots, a_m\}$ in the information system $\mathscr{S} = \langle U, A \rangle$. By the discernibility matrix $\mathbf{M}(\mathscr{S})$, of $\mathscr{S}$ is meant an $n \times n$-matrix such that

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\}. \tag{4}$$

A discernibility function $f_{\mathscr{S}}$ is a function of m boolean variables $\bar{a}_1, \ldots, \bar{a}_m$ corresponding to the attributes $a_1, \ldots, a_m$ respectively and defined as follows:

$$f_{\mathscr{S}}(\bar{a}_1, \ldots, \bar{a}_m) = \wedge\{\vee(c_{ij}) : 1 \leqslant i, j \leqslant n, \; j < i, c_{ij} \neq \emptyset\}, \tag{5}$$

where $\vee(c_{ij})$ is the disjunction of all variables $\bar{a}$ with $a \in c_{ij}$. It is seen in (Skowron and Rauszer, 1992) that $\{a_{i_1}, \ldots, a_{i_p}\}$ is a reduct in $\mathscr{S}$ if and only if $a_{i_1} \wedge \cdots \wedge a_{i_p}$ is a prime implicant (constituent of the disjunctive normal form) of $f_{\mathscr{S}}$.

3.2. Indiscernibility of patterns and discretization of the feature space

A primary notion of rough set is of indiscernibility relation. For continuous valued attributes the feature space needs to be discretized for defining indiscernibility relations and equivalence classes. Discretization is a widely studied problem in rough set theory and in this article we use fuzzy set theory for effective discretization. Use of fuzzy sets has several advantages over 'hard' discretization, like modelling of overlapped clusters, linguistic representation of data. We discretize each feature into three levels low, medium and high, finer discretizations may lead to better accuracy at the cost of higher computational load.

Each feature of a pattern is described in terms their fuzzy membership values in the linguistic property sets *low* (L), *medium* (M) and *high* (H). Let these be represented by L_j, M_j and H_j respectively. The features for the ith pattern $\mathbf{F}_i$ are mapped to the corresponding three-dimensional feature space of $\mu_{\text{low}(F_{ij})}(\mathbf{F}_i)$, $\mu_{\text{medium}(F_{ij})}(\mathbf{F}_i)$ and $\mu_{\text{high}(F_{ij})}(\mathbf{F}_i)$, by Eq. (6). An n-dimensional pattern $\mathbf{F}_i = [F_{i1}, F_{i2}, \ldots, F_{in}]$ is represented as a $3n$-dimensional vector (Pal and Mitra, 1992, 1999)

$$\mathbf{F}_i = [\mu_{\text{low}(F_{i1})}(\mathbf{F}_i), \ldots, \mu_{\text{high}(F_{in})}(\mathbf{F}_i)], \tag{6}$$

where the μ values indicate the membership functions of the corresponding linguistic π-sets *low*, *medium* and *high* along each feature axis. This effectively discretizes each feature into three levels.

Then consider only those attributes which have a numerical value greater than some threshold Th ($=0.5$, say). This implies clamping only those features demonstrating high membership values with one, while the others are fixed at zero. An attribute-value table is constructed comprising of the above binary valued $3n$-dimensional feature vectors.

We use the π-fuzzy sets (in the one-dimensional form), with range $[0,1]$, represented as

$$\pi(F_j; c, \lambda) = \begin{cases} 2\left(1 - \frac{\|F_j - c\|}{\lambda}\right)^2, & \text{for } \frac{\lambda}{2} \leqslant \|F_j - c\| \leqslant \lambda, \\ 1 - 2\left(\frac{\|F_j - c\|}{\lambda}\right)^2, & \text{for } 0 \leqslant \|F_j - c\| \leqslant \frac{\lambda}{2}, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where λ (>0) is the radius of the π-function with c as the central point. The details of the above method may be found in (Pal and Mitra, 1999).

Let us now explain the procedure for selecting the centers (c) and radii (λ) of the overlapping π-sets. Let m_j be the mean of the pattern points along the jth axis. Then m_{j_l} and m_{j_h} are defined as the mean (along the jth axis) of the pattern points having co-ordinate values in the range $[F_{j_{\min}}, m_j)$ and $(m_j, F_{j_{\max}}]$ respectively, where $F_{j_{\max}}$ and $F_{j_{\min}}$ denote the upper and lower bounds of the dynamic range of feature F_j (for the training set) considering numerical values only. For the three linguistic property sets along the jth axis, the centers and the corresponding radii of the corresponding π-functions are defined as

$$\begin{aligned}
&c_{\text{low}(F_j)} = m_{j_l},\\
&c_{\text{medium}(F_j)} = m_j,\\
&c_{\text{high}(F_j)} = m_{j_h},\\
&\lambda_{\text{low}(F_j)} = c_{\text{medium}(F_j)} - c_{\text{low}(F_j)},\\
&\lambda_{\text{high}(F_j)} = c_{\text{high}(F_j)} - c_{\text{medium}(F_j)},\\
&\lambda_{\text{medium}(F_j)} = 0.5\left(c_{\text{high}(F_j)} - c_{\text{low}(F_j)}\right),
\end{aligned} \tag{8}$$

respectively. Here we take into account the distribution of the pattern points along each feature axis while choosing the corresponding centers and radii of the linguistic properties. The nature of membership functions are illustrated in Fig. 1.

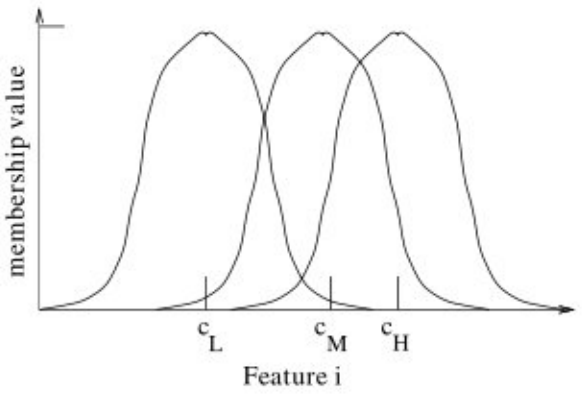

Fig. 1. π-Membership functions for linguistic property sets *low* (L), *medium* (M) and *high* (H) for each feature axis.

3.3. Methodology for generation of reducts

Let there be m sets $O_1, \ldots, O_m$ of objects in the attribute-value table (obtained using the procedure explained in the last section) having identical attribute values, and $\text{card}(O_i) = n_{k_i}$, $i = 1, \ldots, m$, such that $n_{k_1} \geqslant \cdots \geqslant n_{k_m}$ and $\sum_{i=1}^{m} n_{k_i} = n_k$. The attribute-value table can now be represented as an $m \times 3n$ array. Let $n_{k'_1}, n_{k'_2}, \ldots, n_{k'_m}$ denote the distinct elements among $n_{k_1}, \ldots, n_{k_m}$ such that $n_{k'_1} > n_{k'_2} > \cdots > n_{k'_m}$. Let a heuristic threshold function be defined as

$$\text{Tr} = \left\lceil \frac{\sum_{i=1}^{m} \frac{1}{n_{k'_i} - n_{k'_{i+1}}}}{\text{Th}} \right\rceil, \tag{9}$$

so that all entries having frequency less than Tr are eliminated from the table, resulting in the reduced attribute-value table $\mathscr{S}$. Note that the main motive of introducing this threshold function lies in reducing the size of the mixture model. One attempts to eliminate noisy pattern representatives (having lower values of n_{k_i}) from the reduced attribute-value table. From the reduced attribute-value table obtained, reducts are obtained using the methodology described below.

Let $\{x_{i_1}, \ldots, x_{i_p}\}$ be the set of those objects of U that occur in $\mathscr{S}$. Now a discernibility matrix (denoted $\mathbf{M}(B)$) is defined as follows:

$$c_{ij} = \{a \in B : a(x_i) \neq a(x_j)\}, \tag{10}$$

for $i, j = 1, \ldots, n$.

For each object $x_j \in x_{i_1}, \ldots, x_{i_p}$, the discernibility function f_{x_j} is defined as

$$f_{x_j} = \wedge\{\vee(c_{ij}) : 1 \leqslant i, j \leqslant n, j < i, c_{ij} \neq \emptyset\}, \tag{11}$$

where $\vee(c_{ij})$ is the disjunction of all members of c_{ij}. One thus obtain a rule r_i, viz. $P_i \rightarrow \text{cluster}_i$, where P_i is the disjunctive normal form (d.n.f) of f_{x_j}, $j \in i_1, \ldots, i_p$.

Support factor sf_i for the rule r_i is defined as

$$sf_i = \frac{n_{k_i}}{\sum_{i=1}^{p} n_{k_i}}, \tag{12}$$

where n_{k_i}, $i = 1, \ldots, p$ are the cardinality of the sets O_i of identical objects belonging to the reduced attribute value table.

3.4. Mapping reducts to mixture parameters

The mixture model parameters consists of the number of component Gaussian density functions (k) and weights (w_h), means (μ_h) and variances (Σ_h) of the components. We describe below the methodology for obtaining them.

(i) *Number of Gaussians (k):* Consider the antecedent part of a rule r_i; Split it into atomic formulas containing only conjunction of literals. For each such atomic formulae, assign a component Gaussian. Let the number of such formulae be k.

(ii) *Component weights (w_h):* Weight of a each Gaussian is set equal to the normalised support factor sf_i (obtained using Eq. (12)) of the rule (r_i) from which it is derived, $w_h = sf_i / \sum_{i=1}^{k} sf_i$.

(iii) *Means (μ_h):* A atomic formulae consists of conjunction of a number of literals. The literals are linguistic fuzzy sets low, medium and high along some feature axes. The component of the mean vector along that feature is set equal to the center (c) of the π-membership function of the corresponding fuzzy linguistic set. Note that all features do not appear in a formulae, implying those features are not necessary to characterise the corresponding cluster. The component of the mean vector along those features which do not appear are set to the mean of the entire data along those features.

(iv) *Variances (Σ_h):* A diagonal covariance matrix is considered for each component Gaussian. As in means, the variance for feature j is set equal to radius λ of the corresponding fuzzy linguistic set. For those features not appearing in a formulae the variance is set to small random value.

3.5. Example

Consider the following two reducts obtained from a reduced attribute value table of a data having two dimension F_1 and F_2. The example is illustrated in Fig. 2.

$$\text{cluster}_1 \leftarrow L_1 \wedge H_2, \quad sf_1 = 0.50,$$

$$\text{cluster}_2 \leftarrow H_1 \wedge L_2, \quad sf_2 = 0.40.$$

Let the parameters of the fuzzy linguistic sets 'low', 'medium' and 'high' be as follows:

Feature 1:

$$c_{\mathrm{L}} = 0.1, \quad \lambda_{\mathrm{L}} = 0.5, \quad c_{\mathrm{M}} = 0.5, \quad \lambda_{\mathrm{M}} = 0.7,$$
$$c_{\mathrm{H}} = 0.7, \quad \lambda_{\mathrm{H}} = 0.4.$$

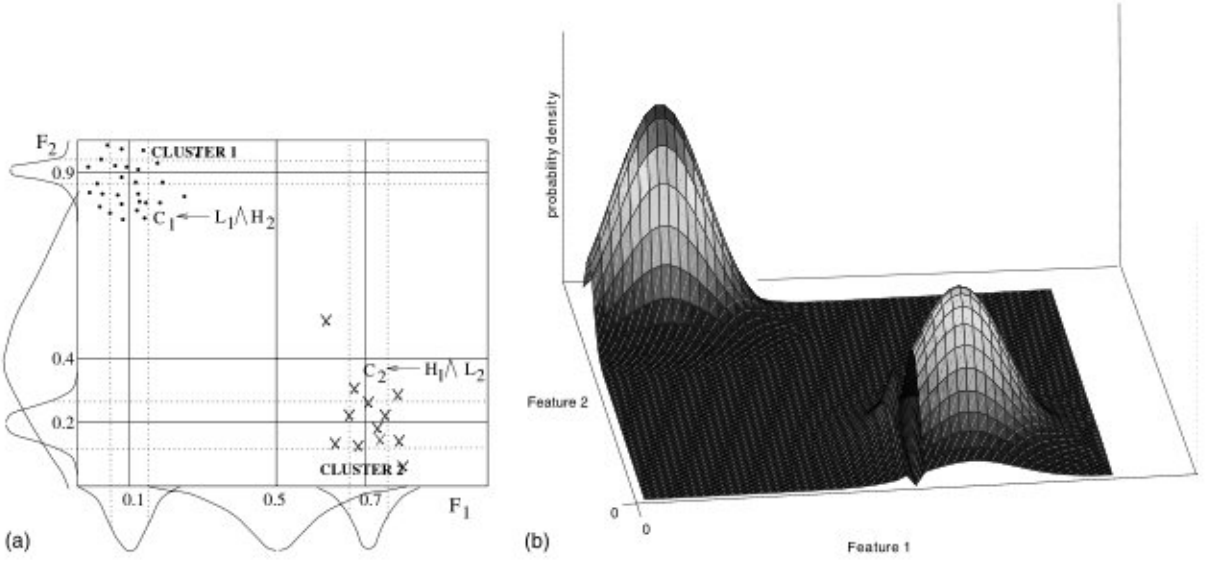

Fig. 2. Rough-fuzzy generation of crude clusters for a two-dimensional data, (a) data distribution and rough set rules, (b) probability density function for the initial mixture model.

Feature 2:

$c_L = 0.2, \quad \lambda_L = 0.5, \quad c_M = 0.4, \quad \lambda_M = 0.7,$
$c_H = 0.9, \quad \lambda_H = 0.5.$

Then we have two component Gaussians with parameters as follows:

$w_1 = 0.56, \quad \mu_1 = [0.1, 0.9] \quad \text{and}$
$\Sigma_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix},$

$w_2 = 0.44, \quad \mu_2 = [0.7, 0.2] \quad \text{and}$
$\Sigma_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}.$

We summarise below all the steps for rough set initialization of mixture models.

(i) Represent each pattern in terms of its membership to fuzzy linguistic sets low, medium and high along each axis. Thus a n-dimensional pattern is now represented by a $3n$-dimensional vector.
(ii) Threshold each $3n$-dimensional vector containing fuzzy membership values to obtain $3n$-dimensional binary vector. Retain only those vectors which are distinct and appear with frequency above a threshold.
(iii) Construct an attribute-value table from the reduced set of binary vectors.
(iv) Construct discernibility matrix from the attribute value table. Generate discernibility functions (rules) for each object in the matrix. Consider atomic formulae of the rules which are conjunction of literals (linguistic variables low, medium and high, in this case).
(v) Map each atomic formulae to parameters w_h, μ_h and Σ_h of corresponding component Gaussian density functions.

4. Graph-theoretic clustering of gaussian components

In this section we describe the methodology for obtaining the final clusters from the Gaussian components used to represent the data. A MST based approach is adopted for this purpose. The MST is a graph that connects a data set of N points so that a complete 'tree' of $N-1$ edges is built. (A tree is a connected graph without cycles.) The tree is 'minimal' when the total length of the edges is the minimum necessary to connect all the points. A MST may be constructed using either Kruskal's or Prim's algorithm. Desired number of clusters of points may be obtained from a MST by deleting the edges having highest weights. For example for the set of nine points $\{A, B, C, D, E, F, G, H, I\}$ illustrated in Fig. 3, two clusters can be obtained by deleting the edge CD having highest weight 6. The two subgraphs represent the clusters. It may be mentioned that arbitrary shaped clusters may be obtained using the above algorithm.

Instead of using individual points, we construct a MST whose vertices are the Gaussian components of the mixture model and the edge weights are the Mahalonbis distance (D) between them is defined as:

$$D^2 = (\mu_1 - \mu_2)^T (\Sigma_1 + \Sigma_2)^{-1} (\mu_1 - \mu_2), \tag{13}$$

where μ_1, μ_2 and Σ_1, Σ_2 are the means and variances of the pair of Gaussians. To obtain k clusters, $k-1$ edges having the highest weights are deleted, components belonging to a single connected subgraph after deletion are considered to represent a single cluster.

Note that each cluster obtained as above is a mixture model in itself. The number of its

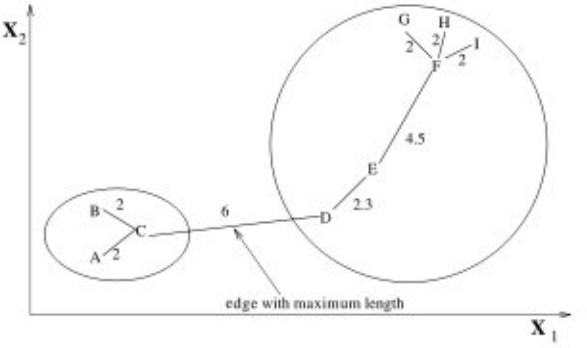

Fig. 3. Using MST to form clusters.

component Gaussians being equal to the number of vertices of the corresponding subgraph. For assigning a point (x) to a cluster, probability of belongingness of x to each of the clusters (submixture models) is computed using Eq. (1), and the cluster giving the highest probability $p(x)$ is assigned to x, i.e., we follow the Bayesian classification rule.

5. Experimental results

Experiments were performed on two real life data sets with large number of samples and dimension. Both the datasets are available in UCI Machine Learning Archive (Blake and Merz, 1998). An artificial non-convex dataset is also considered for the convenience of demonstrating some features of the algorithm along with visualization of the performance. The characteristics of the datasets are summarised below:

(i) *Forest covertype:* Contains 10 dimensions, 7 classes and 586,012 samples. It is an Geographical Information System data representing forest cover type (pine/fir etc.) of USA. The variables are cartographic and remote sensing measurements. All the variables are numeric.
(ii) *Multiple features:* This dataset consists of features of handwritten numerals (0–9) extracted from a collection of Dutch utility maps. There are total 2000 patterns, 649 features (all numeric) and 10 classes.
(iii) *Pat:* This is an artificial data with two dimensions and two horse-shoe shaped non-convex clusters with total 417 points.

The clustering results of the proposed methodology are compared with those obtained using

1. k-means algorithm with random initialization (KM).
2. k-means algorithm with rough set initialization (of centers) and graph-theoretic clustering (RKMG).
3. EM algorithm with random initialization and graph-theoretic clustering (EMG).
4. EM algorithm with means initialised with the output of k-means algorithm and with graph-theoretic clustering (KEMG).

Among the algorithms mentioned above, methods 2–4 have the capability for obtaining non-convex clusters, while method 1 can obtain convex clusters only. It may be mentioned that, in the proposed algorithm, we use EM algorithm with rough set initialization and graph-theoretic clustering. For the purpose of comparison, in addition to rough set theoretic initialization, we have also considered EM algorithms with random initialization (method 3) and another popular method for initialization (method 4). Besides these, to demonstrate the effect of rough set theoretic initialization on another hybrid iterative refinement-graph theoretic clustering method, we consider method 2, which is the k-means algorithm with graph theoretic clustering. We could not present the comparisons with purely graph-theoretic techniques (i.e., on the original data) as they require infeasibly large time for the datasets used.

Comparison is performed on the basis of cluster quality index β (Pal et al., 2000) and CPU time. CPU time is obtained on an Alpha 750 MHz workstation. β is defined as (Pal et al., 2000):

$$\beta = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(X_{ij}-\overline{X})^{\mathrm{T}}(X_{ij}-\overline{X})}{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(X_{ij}-\overline{X}_i)^{\mathrm{T}}(X_{ij}-\overline{X}_i)}, \tag{14}$$

where n_i is the number of points in the ith ($i = 1,\ldots,k$) cluster, X_{ij} is the feature vector of the jth pattern ($j = 1,\ldots,n_i$) in cluster i, $\overline{X}_i$ the mean of n_i patterns of the ith cluster, n is the total number of patterns, and $\overline{X}$ is the mean value of the entire set of patterns. Note that β is nothing but the ratio of the total variation and within-cluster variation. This type of measure is widely used for feature selection and cluster analysis (Pal et al., 2000). For a given data and k (number of clusters) value, the higher the homogeneity within the clustered regions, higher would be the β value.

For the purpose of visualization of the partitioning, and illustration of several characteristics of the algorithm, we first present the results on the artificial *Pat* data set which is of smaller dimension (=2). The non-convex character of the data is

shown in Fig. 4. The reducts obtained using rough set theory, and the parameters of the corresponding four Gaussians are as follows:

$$\text{cluster}_1 \leftarrow L_1 \wedge M_2; \quad w_1 = 0.15,$$
$$\mu_1 = [0.223, 0.511], \quad \Sigma_1 = \begin{bmatrix} 0.276 & 0 \\ 0 & 0.240 \end{bmatrix},$$

$$\text{cluster}_2 \leftarrow H_1 \wedge M_2; \quad w_2 = 0.16,$$
$$\mu_2 = [0.753, 0.511], \quad \Sigma_2 = \begin{bmatrix} 0.233 & 0 \\ 0 & 0.240 \end{bmatrix},$$

$$\text{cluster}_3 \leftarrow M_1 \wedge H_2; \quad w_3 = 0.35,$$
$$\mu_3 = [0.499, 0.744], \quad \Sigma_3 = \begin{bmatrix} 0.265 & 0 \\ 0 & 0.233 \end{bmatrix},$$

$$\text{cluster}_4 \leftarrow M_1 \wedge L_2; \quad w_4 = 0.34,$$
$$\mu_4 = [0.499, 0.263], \quad \Sigma_4 = \begin{bmatrix} 0.265 & 0 \\ 0 & 0.248 \end{bmatrix}.$$

The distribution of points belonging to each component Gaussian, obtained after refining the parameters using EM, is plotted in Fig. 5. These are indicated by symbols: +, ∘, ◇, and △. The variation of log-likelihood with EM iteration is presented in Fig. 6 for both random initialization and rough set initialization. It is seen that for rough set initialization log-likelihood attains a higher value at the start of EM. The final clusters (two in number) obtained by our method after

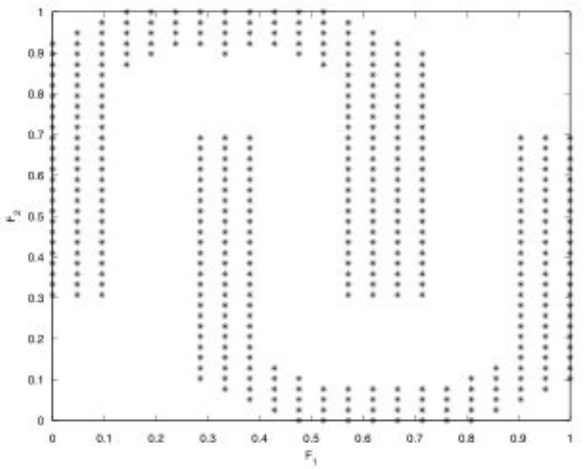

Fig. 4. Scatter plot of the artificial data *Pat*.

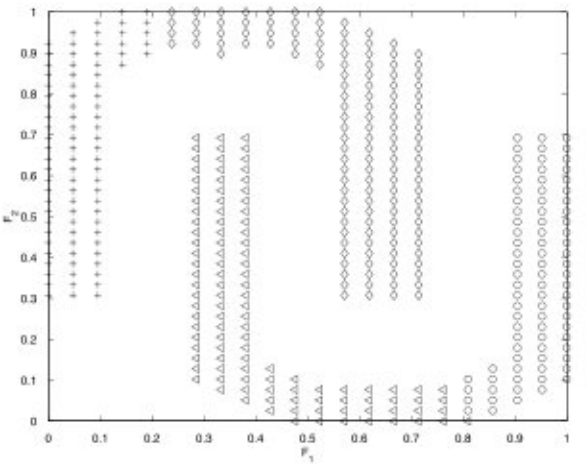

Fig. 5. Scatter plot of points belonging to four different component Gaussians for the *Pat* data. Each Gaussian is represented by a separate symbol (+, ∘, ◇ and △).

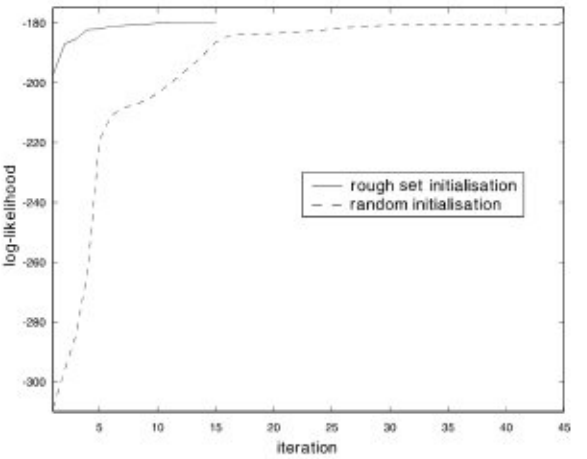

Fig. 6. Variation of log-likelihood with EM iterations for the *Pat* data.

graph-theoretic partitioning of the Gaussians are shown in Fig. 7(a). The algorithm is seen to produce the same natural non-convex partitions, as in the original data. It may be noted that the conventional k-means algorithm, which is capable of generating convex clusters efficiently; fails to do so (Fig. 7(b)), as expected.

Table 1 provides comparative results (in terms of β and CPU time) of the proposed algorithm with other four, as mentioned before, for three

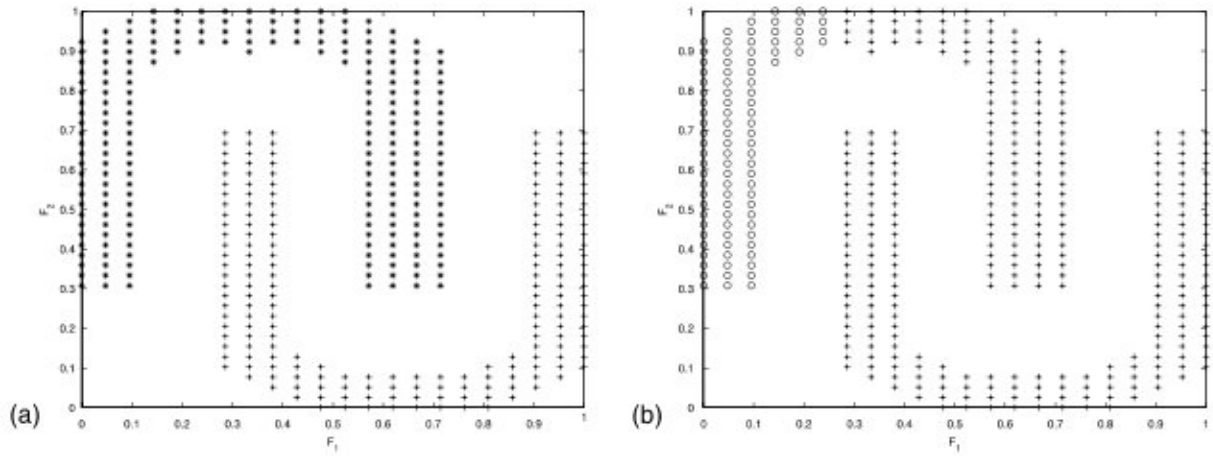

Fig. 7. Final clusters obtained using, (a) proposed algorithm, (b) k-means algorithm for the *Pat* data (clusters are marked by '+' and 'o').

Table 1
Comparative performance of clustering algorithms

Algorithm	Cluster quality (β)	CPU time (sec)
Forest data		
Proposed	7.10	1021
KEMG	6.21	2075
EMG	5.11	1555
RKMG	5.90	590
KM	3.88	550
Multiple features data		
Proposed	11.20	721
KEMG	10.90	881
EMG	10.40	810
RKMG	10.81	478
KM	7.02	404
Pat data		
Proposed	18.10	1.04
KEMG	15.40	2.10
EMG	10.90	1.80
RKMG	15.30	0.91
KM	8.10	0.80

different datasets. It is seen that the proposed methodology produces clusters having the highest β value for all the cases. The CPU time required is less than that of the other two EM based algorithms (EMG and KEMG). For the k-means algorithm (KM) although the CPU time requirement is the least, its performance is significantly poorer.

Rough set theoretic initialization is found to improve the β value as well as reduce the time requirement of both EM and k-means. It is also observed that k-means with rough set theoretic initialization (RKMG) performs better than EM with random initialization (EMG), though it is well known that EM is usually superior to k-means in partitioning.

6. Conclusions

The contribution of the article is twofold. Firstly rough set theory is used to effectively circumvent the initialization and local minima problems of iterative refinement clustering algorithms (like EM and k-means). This also improves the clustering performance, as measured by β value.

The second contribution lies in the development of a methodology integrating the merits of graph-theoretic clustering (e.g., capability of generating non-convex clusters) and iterative refinement clustering (such as low computational time requirement). At the local level the data is modelled by Gaussians, i.e., as combination of convex sets, while globally these Gaussians are partitioned using graph-theoretic technique; thereby enabling the efficient detection of the non-convex clusters present in the original data. Since the number of

Gaussians is much less than the total number of data points, the computational time requirement for this integrated method is much less than that required by a conventional graph theoretic clustering.

The number of clusters obtained in our algorithm is user specified. In case it is not available, the same can be automatically determined by computing the derivatives of the edge weight values of the minimal spanning tree, and deleting the edges corresponding to the maxima(s) of the derivatives. This will give rise to the natural grouping of the data.

It may be noted that the capability of rough set theory in extracting domain knowledge in the form of crude rules has been exploited here for clustering. Similar exploitation has been made earlier (Szczuka, 2000; Banerjee et al., 1998) for neural network architecture design.

References

Banerjee, M., Mitra, S., Pal, S.K., 1998. Rough fuzzy MLP: Knowledge encoding and classification. IEEE Trans. Neural Networks 9 (6), 1203–1216.

Blake, C.L., Merz, C.J., 1998. UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, Available from <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Bradley, P., Fayyad, U., Reina, C., 1998. Scaling clustering algorithms to large databases. In: The Fourth International Conference on Knowledge Discovery and Data Mining, AAAI, NY.

Bradley, P., Fayyad, U., Reina, C., 1999. Scaling EM (expectation maximization) algorithm to large databases, Microsoft Research Technical Report, MSR-TR-98-35, Available from <http://www.ece.nwu.edu/harsha/Clustering/tr-98-35.ps>.

Cherkassky, V., Mulier, F., 1998. Learning from Data: Concepts, Theories and Methods. John Wiley, NY.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc., Ser. B 39, 1–38.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Comput. Surveys 31 (3), 264–323.

Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A., 1997. Rough sets: A tutorial. In: Proceedings of PKDD'97, LNAI 1263, Springer-Verlag, Trondheim, Norway, pp. 103–114.

Meila, M., Heckerman, D, 1998. An experimental comparison of several clustering and initialization methods. Microsoft Research Technical Report, MSR-TR-98-06, Available from <ftp://ftp.research.microsoft.com/pub/tr/TR-98-06.PS>.

Pal, S.K., Ghosh, A., Uma Shankar, B., 2000. Segmentation of remotely sensed images with fuzzy thresholding, and quantitative evaluation. Internat. J. Remote Sensing 21 (11), 2269–2300.

Pal, S.K., Mitra, S., 1992. Multi-layer perceptron, fuzzy sets and classification. IEEE Trans. Neural Networks 3, 683–697.

Pal, S.K., Mitra, S., 1999. Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing. John Wiley, New York.

Pawlak, Z., 1991. Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic, Dordrecht.

Scott, D.W., 1992. Multivariate Density Estimation. John Wiley, New York.

Skowron, A., Rauszer, C., 1992. The discernibility matrices and functions in information systems. In: Słowiński, R. (Ed.), Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic, Dordrecht, pp. 331–362.

Szczuka, M., 2000. Rough sets and artificial neural networks. In: Polkowski, L., Skowron, A. (Eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems. Physica-Verlag, Heidelberg, pp. 449–470.

Zahn, C.T., 1971. Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans. Comput. 20, 68–86.

Zhang, T., Ramakrishnan, R., Livny, M., 1996. BIRCH: An efficient data clustering method for very large data bases. In: Proceedings ACM SIGMOD Conference on Management of Data, Montreal, Canada, pp. 103–114.