International Journal of Pattern Recognition and Artificial Intelligence
Vol. 15, No. 2 (2001) 269–285

CLUSTERING USING SIMULATED ANNEALING WITH PROBABILISTIC REDISTRIBUTION

SANGHAMITRA BANDYOPADHYAY
Machine Intelligence Unit, Indian Statistical Institute,
203 B.T. Road, Calcutta 700 035, India
E-mail: sanghami@isical.ac.in

UJJWAL MAULIK and MALAY KUMAR PAKHIRA
Department of Computer Science and Technology,
Kalyani Government Engineering College, Kalyani 741 235, India
E-mail: {ujjwal_maulik,mkp}@kucse.wb.nic.in

An efficient partitional clustering technique, called SAKM-clustering, that integrates the power of simulated annealing for obtaining minimum energy configuration, and the searching capability of K-means algorithm is proposed in this article. The clustering methodology is used to search for appropriate clusters in multidimensional feature space such that a similarity metric of the resulting clusters is optimized. Data points are redistributed among the clusters probabilistically, so that points that are farther away from the cluster center have higher probabilities of migrating to other clusters than those which are closer to it. The superiority of the SAKM-clustering algorithm over the widely used K-means algorithm is extensively demonstrated for artificial and real life data sets.

Keywords: Euclidean distance; K-means algorithm; pattern recognition; partitional clustering; simulated annealing; temperature scheduling.

1. Introduction

Simulated annealing (SA)[2,13,18,21,22] is a recent technique for finding good solutions to a wide variety of combinatorial optimization problems. It mimics the principles of the annealing procedure which is a physical process where a crystal is cooled down from the liquid to the solid phase. If the cooling is done slowly enough, the energy state of the crystal at the end will be very close to its minimum value. Simulation of this physical cooling may be done with the Metropolis algorithm. It generates sequences of configurations in the following way: given a current configuration C_i with energy E_i, the next configuration C_j (with energy E_j) is generated by applying a small perturbation in C_i. If $(E_j - E_i)$ is less than or equal to 0, then C_j is accepted as the current configuration. Otherwise, it is accepted with a probability $\exp(-\frac{E_j - E_i}{k_\beta T})$, where T and k_β represent the temperature and Boltzmann's constant respectively. If the lowering of the temperature is done slowly enough, the crystal reaches thermal equilibrium at each temperature. In the Metropolis al-

gorithm this is achieved by applying sufficiently large number of perturbations at each temperature.

The SA process may be viewed as a graph with an energy E assigned to each node. Here, the nodes are called *states*, the arcs represent *moves* from one state to a neighbouring state, and the energy is equivalent to *cost*. The algorithm starts from a random initial configuration at high temperature. It then proceeds by generating new candidate states and accepting/rejecting them according to a probability which is a function of the current temperature and energy difference. The temperature is gradually decreased towards a minimum value, while the system settles down to a stable low energy state.

Clustering[1,5,9,10,23] is an important unsupervised classification technique where a set of patterns, usually vectors in a multidimensional space, are grouped into clusters in such a way that patterns in the same cluster are similar in some sense and patterns in different clusters are dissimilar in the same sense. For this, it is necessary to first define a measure of similarity which establishes a rule for assigning patterns to the domain of a particular cluster centre. One such measure of similarity, that has been adopted in this article, is the Euclidean distance $\mathbf{D}$ between two patterns $\mathbf{x}$ and $\mathbf{z}$ defined by $\mathbf{D} = ||\mathbf{x} - \mathbf{z}||$. Smaller the distance between $\mathbf{x}$ and $\mathbf{z}$, greater is the similarity between the two and vice versa. Another well-known measure that can be found in the literature is Mahalanobis distance.[11,25] One may note that an exhaustive search in the feature space in order to provide the optimal clusters has combinatorial complexity. Therefore, the application of SA, which is known to be effective in large and multimodal search space, seems appropriate and natural.

Some clustering techniques that are available in the literature are branch and bound procedure,[14] maximum likelihood estimate technique,[24] graph theoretic approaches.[15] Branch and bound procedure uses a tree search technique for searching the entire solution space in order to classify a given set of points into a fixed number of clusters, along with a criterion for eliminating subtrees which do not contain the optimum result. Here the number of nodes to be searched increases rapidly with the size of the data. Therefore, a proper choice of the criterion for eliminating subtrees becomes crucial.[7] The maximum likelihood estimate technique performs clustering by computing the posterior probabilities of the classes after assuming a particular distribution of the data set. In the graph theoretic approach, a directed tree is formed among the data set by estimating the density gradient at each point. The clustering is realized by finding the valley of the density function. It is known that the quality of the result depends wholly on the performance of the estimation technique for the density gradient, particularly in the low density area of the valley. One recent attempt regarding the application of genetic algorithms (GAs)[4,8] to the clustering problem may be found in Ref. 16. Here a partition of the data set is encoded as a string whose length is equal to the size of the data set. Consequently the method will fail to attain good results quickly for large data set since the string length to be handled by GA becomes large. Extensive studies dealing with comparative analysis of different clustering methods suggest that there is no general

strategy which works equally well in different problem domains. However, it has been found that it is usually beneficial to run schemes that are simpler, and execute them several times, rather than using schemes that are very complex but need to be run only once.

One such widely used and intuitively simple clustering technique is the K-means algorithm,[23] which optimizes the distance criterion by minimizing the intra cluster spread. Since our aim is to propose a clustering technique based on SA, a criterion is required whose optimization would provide the final clusters. An intuitively simple criterion is within cluster spread, which, as in the K-means algorithm, needs to be minimized for good clustering. However, unlike the K-means algorithm which may get stuck at values which are not optimal,[19] the proposed technique should be able to provide good results irrespective of the starting configuration. It is towards this goal that we have integrated the simplicity of the K-means algorithm with the capability of SA in avoiding local optima for developing a clustering technique called SAKM-clustering algorithm (for simulated annealing based clustering using K-means).

Experimental results comparing the SAKM-clustering algorithm with the K-means algorithm are provided for several artificial and real life data sets. Since our purpose is to demonstrate the effectiveness of the proposed technique for a wide variety of data sets, we have chosen artificial and real life data sets with both overlapping and nonoverlapping class boundaries, where the number of dimensions ranges from two to ten and the number of clusters ranges from two to six.

2. Partitional Clustering

The problem of partitional clustering is formally stated as follows. Given n patterns in an N-dimensional Euclidean space, $\mathbb{R}^N$, determine a partition of the patterns into K groups, or clusters, such that the patterns in a cluster are more similar to each other than to patterns in different clusters. There are two issues that need to be addressed in the definition of the clustering strategy. The first problem to be solved is how to translate in some mathematical formula the criterion that represents the intuitive notions about cluster. One such mathematical formulation of the clustering criterion is as follows.

Let the n points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be represented by S and the K clusters by $\{C_1, C_2, C_3, \ldots, C_K\}$. Then

$$C_i \neq \emptyset \qquad \text{for } i = 1, \ldots, K$$

$$C_i \bigcap C_j = \emptyset \quad \text{for } i = 1, \ldots, K, \qquad j = 1, \ldots, K \text{ and } i \neq j, \text{ and}$$

$$\bigcup_{i=1}^{K} C_i = S.$$

The second problem which needs to be focussed on is how to check, in reasonable time, all possible partitions in order to find those that optimize the defined criterion. It is shown in Refs. 1 and 20 that if exhaustive enumeration is used to solve a

clustering problem with n points and K clusters, then one requires to evaluate

$$\frac{1}{K}\sum_{j=1}^{K}(-1)^{K-j}j^n$$

partitions. For a data set of size 10 with 2 clusters, the value is $2^9 - 1$ (= 511), while that of size 50 with 2 clusters is $2^{49} - 1$ (i.e. of the order of 10^{15}).

Since enumerating all possible partitions is a combinatorial problem, heuristics are applied for getting good solutions in polynomial time. In this article our aim is to design a clustering technique which is simple and able to provide good solution fast enough while being independent of the distribution of the data set. The principles of the K-means algorithm are utilized for devising such a technique, along with the capability of an annealing procedure for providing the requisite perturbation to force the system towards a state with minimum energy. The steps of the K-means algorithm[23] are therefore first described in brief.

Step 1. Choose K initial cluster centers $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$ randomly from the n points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$.

Step 2. Assign point $\mathbf{x}_i$, $i = 1, 2, \ldots, n$ to cluster C_j, $j \in \{1, 2, \ldots, K\}$ iff

$$||\mathbf{x}_i - \mathbf{z}_j|| < ||\mathbf{x}_i - \mathbf{z}_p||, \qquad p = 1, 2, \ldots, K, \text{ and } j \neq p.$$

Ties are resolved arbitrarily.

Step 3. Compute new cluster centres $\mathbf{z}_1^*, \mathbf{z}_2^*, \ldots, \mathbf{z}_K^*$ as follows:

$$\mathbf{z}_i^* = \frac{1}{n}\sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j, \qquad i = 1, 2, \ldots, K$$

where n_i is the number of elements belonging to cluster C_i.

Step 4. If $\mathbf{z}_i^* = \mathbf{z}_i$, $i = 1, 2, \ldots, K$, then terminate. Otherwise go to step 2.

Note that in case the process does not terminate at Step 4 normally, then it is executed for a maximum fixed number of generations.

It has been shown in Ref. 19 that K-means algorithm may converge to values that are not optimal. Also global solutions of large problems cannot be found with a reasonable amount of computational effort.[20] It is because of these factors that several approximate methods have been developed to solve the underlying optimization problem. One such method using SA is described in the next section.

3. Clustering Using Simulated Annealing

3.1. *Basic principle*

The searching capability of simulated annealing has been used in this article for the purpose of appropriately determining a fixed number K of cluster centers in $\mathbb{R}^N$; thereby suitably clustering the set of n unlabeled points. The clustering metric that has been adopted is the sum of the Euclidean distances of the points from their

respective cluster centers. Thus like K-means algorithm the simulated annealing process tries to minimize the intra cluster spread.

Mathematically, the clustering metric $\mathcal{M}$ for the K clusters $C_1, C_2, \ldots, C_k$ is given by

$$\mathcal{M}(C_1, C_2, ..., C_k) = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} ||\mathbf{x}_j - \mathbf{z}_i|| .$$

The goal of the method is to search for the appropriate cluster centers $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$ such that the clustering metric $\mathcal{M}$ is minimized. In this article, we have used $\mathcal{M}$ as the energy E associated with a particular configuration i.e. a partition of the n points in K clusters.

3.2. *SAKM-clustering algorithm*

In the SAKM-clustering algorithm, the data points are initially assigned to K clusters known *a priori*, randomly, and the cluster centers are computed as in K-means algorithm. Each element in a cluster has a certain degree of membership to it which is inversely proportional to its Euclidean distance from the cluster center. So elements that are at a larger distance from the centre are the fittest candidates for redistribution to another cluster. We have redistributed an element x_i in cluster C_j to cluster C_k with probability

$$\exp\left(\frac{-[D_{ik} - D_{ij}]+}{T_t}\right) \tag{1}$$

where $[x]+ = \max(x, 0)$, and $D_{ik} = ||x_i - \mathbf{z}_k||$ and $k \neq j$. T is a temperature schedule, which is a sequence of strictly positive numbers such that $T_1 \geq T_2 \geq \ldots T_t = 0$, $(\lim t \to \infty)$. The suffix t of T indicates the number of generations through the annealing process. The simulated annealing algorithm is shown in Fig. 1.

1. $T = T_{\max}$
2. Generate initial configuration (C) with energy E by randomly distributing the points to K clusters.
3. while$(T > T_{\min})$
4. for $i = 1$ to N_T do /* N_T is the number of generations a temperature T */
5. Evolve C' with energy E' from C by redistributing points in C following equation 1
6. If $(E' - E \leq 0)$ $C \leftarrow C'$
7. Else $C \leftarrow C'$, with probability $\exp(-\frac{E'-E}{T})$
8. end for
9. Decrement T
10. end while

Fig. 1. Basic steps in SA.

3.2.1. *The temperature schedule*

The asymptotic convergence (i.e. at $t \to \infty$) of the SA is guaranteed for a logarithmic annealing schedule of the form $T_t = T_1/(1 + \ln t)$, where $t \geq 1$ and T_1 is the initial temperature. However, in practice, the logarithmic annealing is far too slow and hence we have used a geometric schedule of the form $T_t = (1 - \alpha)^t * T_1$, where α is a positive real number close to zero. As $T_t \to 0$, no more perturbation of the cluster configuration is possible and hence termination condition is assumed to be reached. In practice, the state of the system configuration is found to be frozen well before this.

4. Implementation Results

The experimental results comparing the SAKM-clustering algorithm with the K-means algorithm are provided for three artificial data sets (*Data* 1, *Data* 2 and *Data* 3) and three real life data sets (*Vowel, Iris* and *Crude Oil*) respectively. These are first described below:

Artificial Data Sets:

Data 1. This is a nonoverlapping two-dimensional data set with two clusters. It has 10 points. The value of K is chosen to be 2 for this data set.

Data 2. This is a nonoverlapping two-dimensional data set with three clusters. It has 76 points. The value of K is chosen to be 3 for this data set.

Data 3. This is an overlapping ten-dimensional data set generated using a triangular distribution of the form shown in Fig. 2 for two classes, 1 and 2. It has 1000 data points. The value of K is chosen to be 2 for this data set. The range for class 1 is

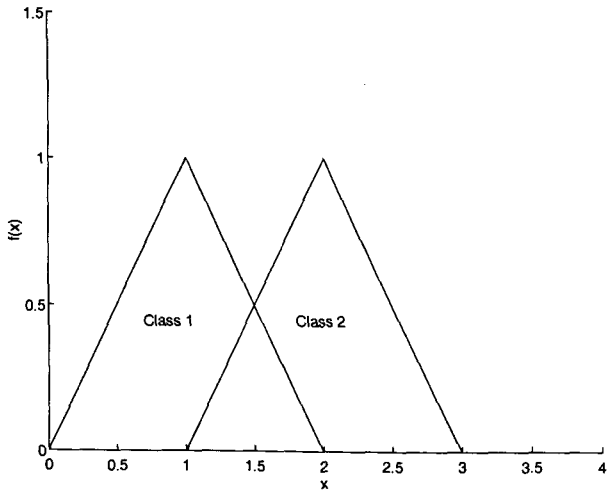


Fig. 2. Triangular distribution along the X axis.

$[0,2] \times [0,2] \times [0,2] \ldots 10$ times, and that for class 2 is $[1,3] \times [0,2] \times [0,2] \ldots 9$ times, with the corresponding peaks at $(1,1)$ and $(2,1)$. The distribution along the first axis (X) for class 1 may be formally quantified as

$$\begin{aligned} f_1(x) &= 0 \quad \text{for} \quad x \leq 0 \\ f_1(x) &= x \quad \text{for} \quad 0 < x \leq 1 \\ f_1(x) &= 2 - x \quad \text{for} \quad 1 < x \leq 2 \\ f_1(x) &= 0 \quad \text{for} \quad x > 2 . \end{aligned}$$

Similarly for class 2

$$\begin{aligned} f_2(x) &= 0 \quad \text{for} \quad x \leq 1 \\ f_2(x) &= x - 1 \quad \text{for} \quad 1 < x \leq 2 \\ f_2(x) &= 3 - x \quad \text{for} \quad 2 < x \leq 3 \\ f_2(x) &= 0 \quad \text{for} \quad x > 3 . \end{aligned}$$

The distribution along the other nine axes (Y_i, $i = 1, 2, \ldots, 9$) for both the classes is

$$\begin{aligned} f(y_i) &= 0 \quad \text{for} \quad y_i \leq 0 \\ f(y_i) &= y_i \quad \text{for} \quad 0 < y_i \leq 1 \\ f(y_i) &= 2 - y_i \quad \text{for} \quad 1 < y_i \leq 2 \\ f(y_i) &= 0 \quad \text{for} \quad y_i > 2 . \end{aligned}$$

Real Life Data Sets:

Vowel Data. This data consists of 871 Indian Telugu vowel sounds.[17] These were uttered in a consonant-vowel-consonant context by three male speakers in the age group of 30–35 years. The data set has three features F_1, F_2 and F_3, corresponding to the first, second and third vowel format frequencies, and six overlapping classes $\{\delta, a, i, u, e, o\}$. The value of K is therefore chosen to be 6 for this data. Figure 3 shows the distribution of the six classes in the $F_1 - F_2$ plane.

Iris Data. This data represents different categories of irises having four feature values. The four feature values represent the sepal length, sepal width, petal length and the petal width in centimeters.[6] It has three classes (with some overlap between classes 2 and 3) with 50 samples per class. The value of K is therefore chosen to be 3 for this data.

Crude Oil Data. This overlapping data[12] has 56 data points, 5 features and 3 classes. Hence the value of K is chosen to be 3 for this data set.

For K-means algorithm maximum of 1000 generations are allowed although it was observed that in all the experiments the algorithm terminated much before

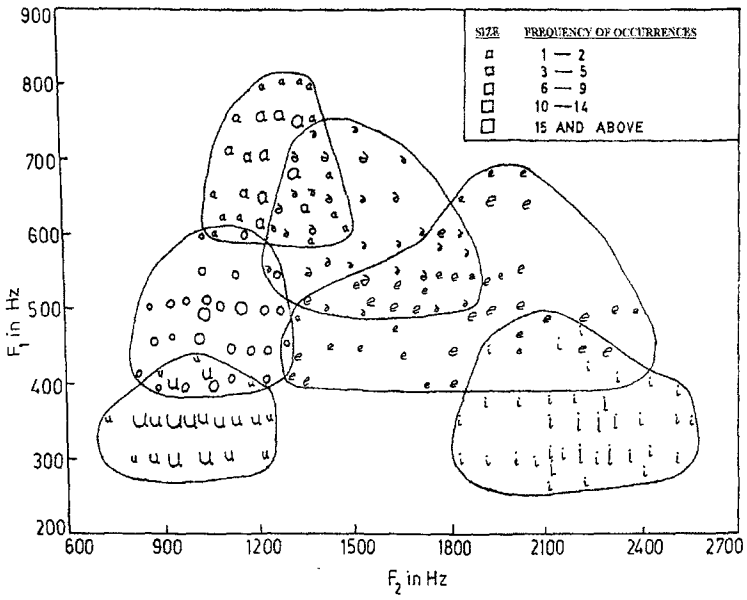


Fig. 3. *Vowel* data in the $F_1 - F_2$ plane.

that. The SAKM-clustering algorithm is executed for a maximum 200 generations. The values of α and T_1 are chosen to be 0.05 and 100 respectively. In a part of the investigation, we have demonstrated the variation of energy with the number of generations for several values of α for one data. The results of implementation of the K-means algorithm and SAKM-clustering algorithm are shown in Tables 1–6 for *Data* 1, *Data* 2, *Data* 3, *Vowel*, *Iris* and *Crude Oil* respectively. Although both the algorithms were run for 100 simulations, for the purpose of demonstration, results are provided for only five different initial configurations.

For *Data* 1 (Table 1), it is found that the SAKM-clustering algorithm provides the optimal value of 2.225498 in all the runs. K-means algorithm also attains this value most of the times (67%). However in the other cases, it gets stuck at a suboptimal value 5.383132. For *Data* 2 (Table 2), SAKM-clustering attains the best value of 47.616294 in all the runs. K-means, on the other hand, attains this value in 51% of the total runs, while in other runs it gets stuck at different suboptimal values. For *Data* 3 (Table 3), the best values obtained by SAKM-clustering algorithm and K-means algorithm are 1246.226685 and 1246.236680 obtained in 63% and 18% of total runs respectively.

For *Vowel* (Table 4), the K-means algorithm attains the best value of 149912.625831 only once (out of 100 runs). The best value obtained by SAKM-clustering algorithm is 149409.250128 (which is obtained in 18% of the total runs). Notably, the latter always obtains values of $\mathcal{M}$ that are better than the best obtained by the former. For *Iris* (Table 5) and *Crude Oil* (Table 6) data sets, the

Table 1. $\mathcal{M}$ obtained by K-means and SAKM-clustering algorithm for five different initial configurations for *Data* 1 when $K = 2$.

Initial configuration	K-means	SAKM-clustering
1	5.383132	2.225498
2	2.225498	2.225498
3	2.225498	2.225498
4	5.383132	2.225498
5	2.225498	2.225498

Table 2. $\mathcal{M}$ obtained by K-means and SAKM-clustering algorithm for five different initial configurations for *Data* 2 when $K = 3$.

Initial configuration	K-means	SAKM-clustering
1	47.616294	47.616294
2	61.613329	47.616294
3	47.616294	47.616294
4	61.613329	47.616294
5	47.616294	47.616294

Table 3. $\mathcal{M}$ obtained by K-means and SAKM-clustering algorithm for five different initial configurations for *Data* 3 when $K = 2$.

Initial configuration	K-means	SAKM-clustering
1	1246.239153	1246.235596
2	1246.239153	1246.226685
3	1246.236680	1246.231934
4	1246.239153	1246.226685
5	1246.237127	1246.226685

Table 4. $\mathcal{M}$ obtained by K-means and SAKM-clustering algorithm for five different initial configurations for *Vowel* when $K = 6$.

Initial configuration	K-means	SAKM-clustering
1	149912.625831	149409.250128
2	150466.266983	149409.312288
3	149913.078096	149429.391189
4	150488.406180	149443.484103
5	150462.450107	149409.312998

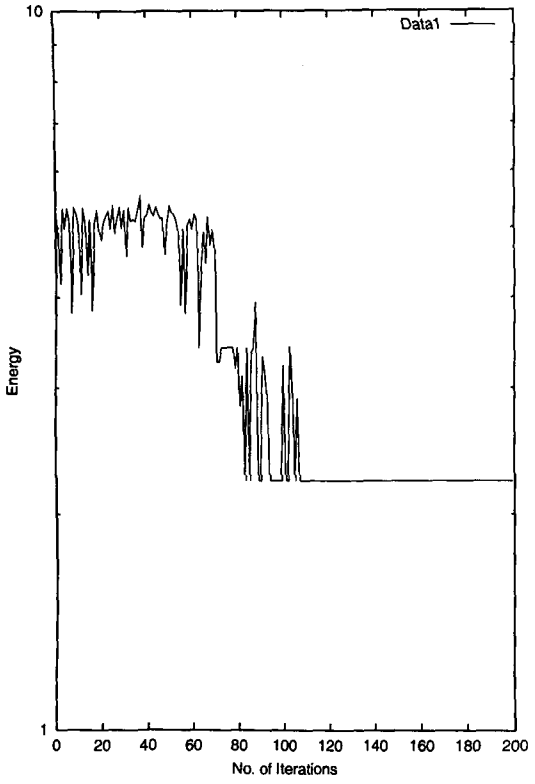


Fig. 4. Variation of energy with generations for *Data* 1.

Table 5. $\mathcal{M}$ obtained by K-means and SAKM-clustering algorithm for five different initial configurations for *Iris* when $K = 3$.

Initial Configuration	K-means	SAKM-clustering
1	97.621869	97.453773
2	98.406574	97.571774
3	98.177353	97.571774
4	97.896373	97.453773
5	98.219574	97.352772

SAKM-clustering algorithm attains the best values of 97.352772 and 279.45815C respectively. The K-means algorithm, on the other hand, fails to attain these value: in any of its runs. The best that K-means algorithm achieved are 97.621869 (reached 25% of the times) and 279.743216 (reached 100% of the times) respectively.

Figures 4–9 demonstrate the variation of the energy values (in logarithmic scale) with the number of generations for SAKM-algorithm for the above mentioned six data sets respectively. All the figures show that in the initial stages the system

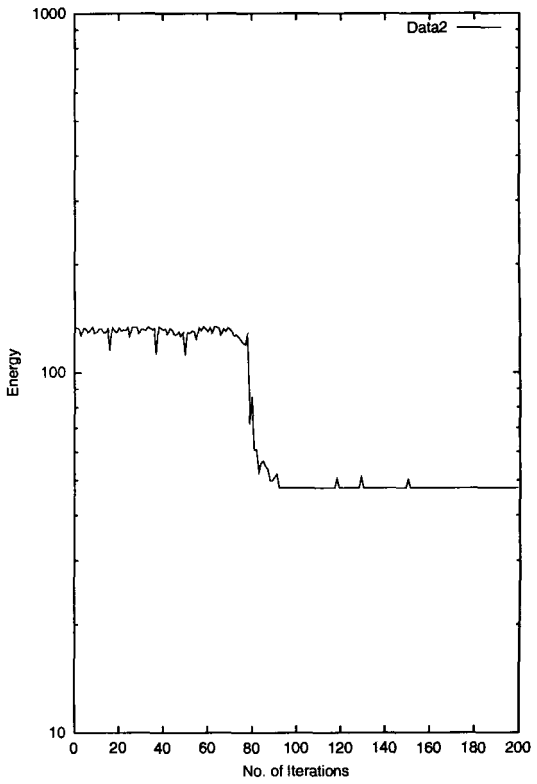


Fig. 5. Variation of energy with generations for *Data* 2.

Table 6. $\mathcal{M}$ obtained by K-means and SAKM-clustering algorithm for five different initial configurations for *Crude Oil* when $K = 3$.

Initial Configuration	K-means	SAKM-clustering
1	279.743216	279.458150
2	279.743216	279.458150
3	279.743216	279.458150
4	279.743216	279.458150
5	279.743216	279.458150

is unstable indicated by the significant changes in the energy values. But after some generations, the system slowly settles down to a more stable state, providing small improvement in steps. Figure 10 shows the variation of energy values with the number of generations for *Data* 2 for five different values of α respectively. As is evident from the figure, the convergence of SAKM algorithm is dependent on the value of α, which needs to be set appropriately for good performance of the algorithm.

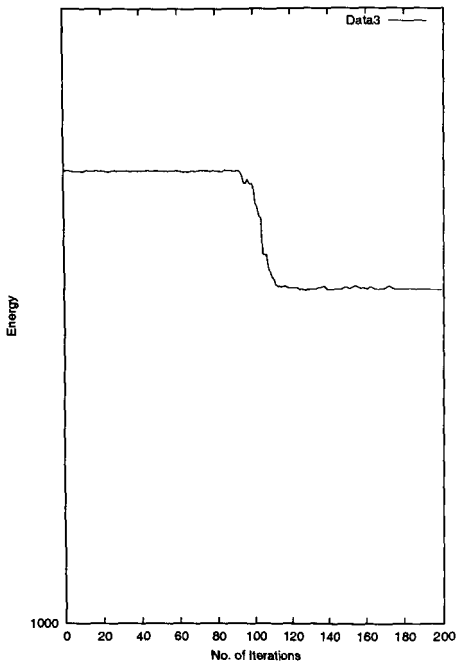


Fig. 6. Variation of energy with generations for *Data* 3.

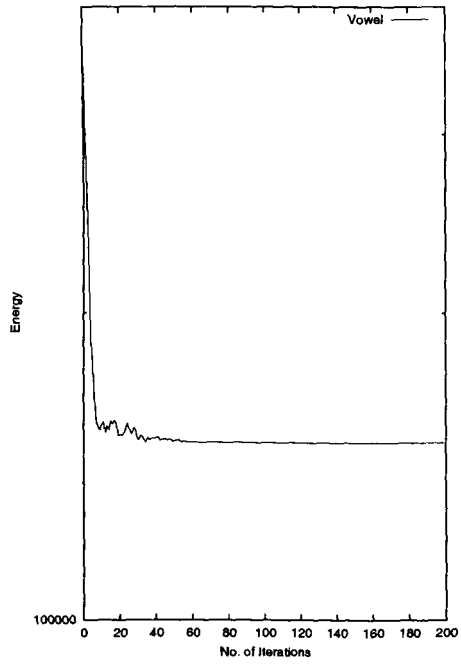


Fig. 7. Variation of energy with generations for *Vowel*.

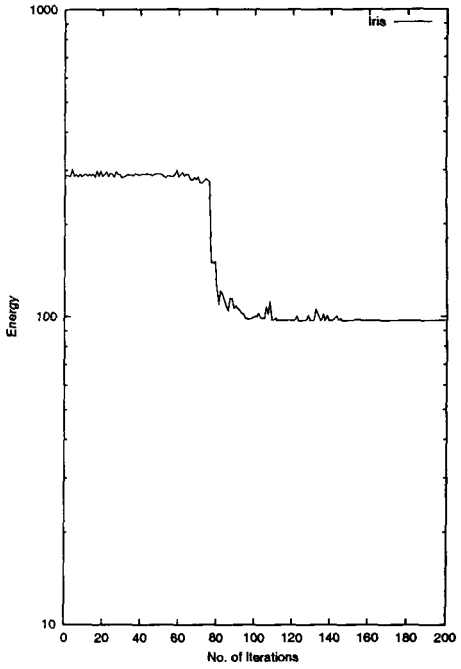


Fig. 8. Variation of energy with generations for *Iris*.

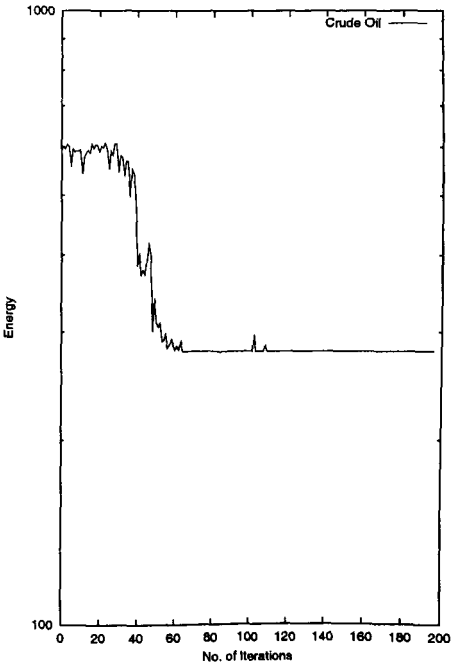


Fig. 9. Variation of energy with generations for *Crude Oil*.

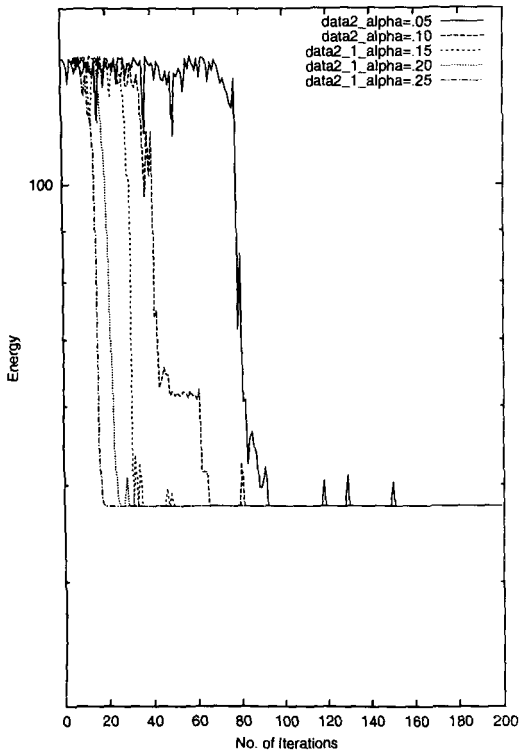


Fig. 10. Variation of energy with generations for *Data* 2 with different alpha values.

5. Discussion and Conclusions

A simulated annealing based clustering algorithm, called SAKM-clustering, has been developed in this article. The power of annealing procedure for providing stable minimum energy configuration has been integrated with the principal of K-means algorithm to search for appropriate clusters which minimizes the clustering metric $\mathcal{M}$. In order to demonstrate the effectiveness of the SAKM-clustering algorithm, several artificial and real life data sets with the number of dimensions ranging from two to ten and the number of clusters ranging from two to six have been considered.

The results show that SAKM-clustering algorithm provides a performance that is significantly superior to that of the K-means algorithm for the overlapping and complex data sets like *Vowel*. On the other hand, for simple data sets some marginal improvement in performance is obtained over the K-means algorithm. Interestingly, the K-means algorithm is sometimes found to get stuck at suboptimal values, even for very simple data, while SAKM-clustering technique has the ability to overcome this. Moreover since the proposed method is inherently very simple, it may be used as an initial analysis tool for a wide variety of data sets in different domains.

The principal of SAKM-clustering algorithm may be extended to the case where the number of cluster are not known *a priori*. In this case cluster indices like Davis

Bouldin index, Dunn's index and some recent generalized indices[3] may be used, which can evolve the number of clusters automatically, while optimizing the clustering metric. Research in this direction is currently being carried out by the authors.

References

1. M. R. Anderberg, *Cluster Analysis for Application*, Academic Press, 1973.
2. S. Bandyopadhyay, S. K. Pal and C. A. Murthy, "Simulated annealing based pattern classification," *Inform. Sci.* **109** (1998) 165–184.
3. J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst. Man Cybern.* **28** (1998) 301–315.
4. L. Davis (ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, NY, 1991.
5. P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall, London, 1982.
6. R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eug.* **3** (1936) 179–188.
7. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, NY, 1990.
8. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, NY, 1989.
9. J. A. Hartigan, *Clustering Algorithms*, Wiley, 1975.
10. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Englewood Cliffs, Prentice-Hall, NJ, 1988.
11. M. Jianchang and A. K. Jain, "A self organizing network for hyperellipsoidal clustering (HEC)," *IEEE World Congress Computat. Intell.* **5** (1994) 2967–2972.
12. R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice-Hall, 1982.
13. S. Kirkpatrik, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," *Science* **220** (1983) 671–680.
14. W. L. G. Koontz, P. M. Narendra and K. Fukunaga, "A branch and bound clustering algorithm," *IEEE Trans. Comput.* **C-24** (1975) 908–915.
15. W. L. G. Koontz, P. M. Narendra and K. Fukunaga, "A graph theoretic approach to nonparametric cluster analysis," *IEEE Trans. Comput.* **C-25** (1975) 936–944.
16. C. A. Murthy and N. Chowdhury, "In search of optimal clusters using genetic algorithms," *Patt. Recog. Lett.* **17** (1996) 825–832.
17. S. K. Pal and D. D. Majumder, "Fuzzy sets and decision making approaches in vowel and speaker recognition," *IEEE Trans. Syst. Man Cybern.* **SMC-7** (1977) 625–629.
18. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes, The Art of Scientific Computing*, Cambridge University Press, 1986.
19. S. Z. Selim and M. A. Ismail, "K-means type algorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Trans. Patt. Anal. Mach. Intell.* **6** (1984) 81–87.
20. H. Spath, *Cluster Analysis Algorithms*, Chichester, Ellis Horwood, UK, 1989.
21. H. Szu and R. Hartley, "Fast simulated annealing," *Phys. Lett.* **A122** (1987) 157–162.
22. H. L. Tan, S. B. Gelfand and E. J. Delp, "A cost minimization approach to edge detection using simulated annealing," *IEEE Trans. Patt. Anal. Mach. Intell.* **14**, 1 (1992) 3–18.
23. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, Reading, 1974.

24. J. H. Wolfe, "Pattern clustering by multivariate mixture analysis," *Multivar. Behav. Res.* **5** (1970) 329–350.
25. K. Younis, M. Karim, R. Hardie, J. Loomis and S. Rogers, "Clustering merging based on weighted mahalanobis distance with application in digital mamograph," *Proc. IEEE 1998 National Aerospace and Electronics Conf. NAECON*, 1998, pp. 525–530.

Sanghamitra Bandyopadhyay did her Bachelors in physics and computer science in 1988 and 1991, respectively from the University of Calcutta, India. Subsequently, she did her Masters in computer science from Indian Institute of Technology, Kharagpur, India in 1993 and Ph.D. in computer science from Indian Statistical Institute, Calcutta, India, in 1998. Since 1999, she has been working at the Indian Statistical Institute as a lecturer.

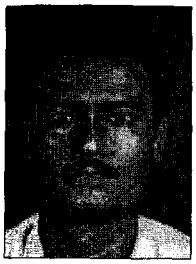
She is the first recipient of *Dr. Shanker Dayal Sharma Gold Medal* and *Institute Silver Medal* for the best all round post graduate performer in 1994. She has worked in Los Alamos National Laboratory in 1997 as a graduate research assistant and in the University of New South Wales, Sydney, Australia, as a post doctoral fellow. Dr. Bandyopadhyay received the Indian National Science Academy (INSA) and the Indian Science Congress Association (ISCA) *Young Scientist Awards* in 2000.

Her research interests include evolutionary and soft computing, pattern recognition, parallel processing and distributed computing.

Ujjwal Maulik did his Bachelors in physics and computer science in 1986 and 1989 respectively from the University of Calcutta, India. Subsequently, he did his Masters and Ph.D. in computer science in 1991 and 1997, respectively from Jadavpur University, India. Dr. Maulik has worked as a scientist in Center for Adaptive Systems Application, Los Alamos, and Los Alamos National Laboratory, New Mexico, USA, in 1997. In 1999, he went on a post doctoral assignment to University of New South Wales, Sydney, Australia.

He is the recipient of the Government of India BOYSCAST fellowship for doing research in the University of Texas at Arlington, USA in 2001. Dr. Maulik has been elected Fellow of the Institute of Electronics and Telecommunication Engineers (IETE). He is currently an assistant professor in the Department of Computer Science, Kalyani Government Engineering College, India.

His research interests include parallel and distributed processing, natural language processing, evolutionary computation and pattern recognition.



Malay K. Pakhira received his B.Sc. degree in physics and B.Tech. degree in computer science and technology from the University of Calcutta, India in 1987 and 1991, respectively. He received his Master of computer science and engineering degree from Jadavpur University, Calcutta, India in 1993. Currently he is a lecturer in computer science and technology at Kalyani Government Engineering College, West Bengal, India. He is doing his Ph.D. research work on pattern classification.

His research interest includes image processing, pattern recognition, genetic algorithm, fuzzy logic and artificial neural networks.