Modified global k-means algorithm for clustering in gene expression data sets

Adil M. Bagirov **Karim Mardaneh**

Centre for Informatics and Applied Optimization,
School of Information Technology and Mathematical Sciences,
University of Ballarat, Victoria, 3353, Australia,
Email: a.bagirov@ballarat.edu.au

Abstract

Clustering in gene expression data sets is a challenging problem. Different algorithms for clustering of genes have been proposed. However due to the large number of genes only a few algorithms can be applied for the clustering of samples. k-means algorithm and its different variations are among those algorithms. But these algorithms in general can converge only to local minima and these local minima are significantly different from global solutions as the number of clusters increases. Over the last several years different approaches have been proposed to improve global search properties of k-means algorithm and its performance on large data sets. One of them is the global k-means algorithm. In this paper we develop a new version of the global k-means algorithm: the modified global k-means algorithm which is effective for solving clustering problems in gene expression data sets. We present preliminary computational results using gene expression data sets which demonstrate that the modified k-means algorithm improves and sometimes significantly results by k-means and global k-means algorithms.

1 Introduction

This paper develops an incremental algorithm for solving sum-of-squares clustering problems in gene expression data sets. Clustering in gene expression data sets is a challenging problem. Different algorithms for clustering of genes have been proposed (see, for example, (Medvedovic & Sivaganesan 2002, Yeung et al. 2001, Yeung et al. 2003)). However due to the large number of genes only a few algorithms can be applied for the clustering of samples ((Bagirov et al. 2003)). As the number of clusters increases the number of variables in the clustering problem increases drastically and most of clustering algorithms become inefficient for solving such problems. k-means algorithm and its different variations are among those algorithms which still applicable to clustering of samples in gene expression data sets. But k-means algorithms in general can converge only to local minima and these local minima may be significantly different from global solutions as the number of clusters increases. Recently the global k-means algorithm has been proposed to improve global search properties of k-means algorithms ((Likas et al. 2003)). In this paper we develop a new version of the global k-means algorithm: the modified global k-means algorithm which is effective for solving clustering problems in gene expression data sets.

The cluster analysis deals with the problems of organization of a collection of patterns into clusters based on similarity. It is also known as the *unsupervised* classification of patterns and has found many applications in different areas. In cluster analysis we assume that we have been given a finite set of points A in the n-dimensional space $\mathbb{R}^n$, that is

$$A = \{a^1, \dots, a^m\}, \text{ where } a^i \in \mathbb{R}^n, \ i = 1, \dots, m.$$

There are different types of clustering. In this paper we consider the hard unconstrained partition clustering problem, that is the distribution of the points of the set A into a given number k of disjoint subsets $A^j, \ j = 1, \dots, k$ with respect to predefined criteria such that:

1) $A^j \neq \emptyset, \ j = 1, \dots, k;$

2) $A^j \bigcap A^l = \emptyset, \ j, l = 1, \dots, k, \ j \neq l;$

3) $A = \bigcup\limits_{j=1}^{k} A^j.$

4) no constraints are imposed on clusters $A^j, \ j = 1, \dots, k.$

The sets $A^j, \ j = 1, \dots, k$ are called clusters. We assume that each cluster A^j can be identified by its center (or centroid) $x^j \in \mathbb{R}^n, \ j = 1, \dots, k$. Then the clustering problem can be reduced to the following optimization problem (see (Bock 1998, Spath 1980)):

$$\text{minimize} \quad \psi(x, w) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} w_{ij} \|x^j - a^i\|^2 \qquad (1)$$

subject to

$$x = (x^1, \dots, x^k) \in \mathbb{R}^{n \times k}, \qquad (2)$$

$$\sum_{j=1}^{k} w_{ij} = 1, \ i = 1, \dots, m, \qquad (3)$$

and

$$w_{ij} = 0 \text{ or } 1, \ i = 1, \dots, m, \ j = 1, \dots, k \qquad (4)$$

where w_{ij} is the association weight of pattern a^i with cluster j, given by

$$w_{ij} = \begin{cases} 1 & \text{if pattern } a^i \text{ is allocated to cluster } j, \\ 0 & \text{otherwise} \end{cases}$$

Copyright ©2006, Australian Computer Society, Inc. This paper appeared at The 2006 Workshop on Intelligent Systems for Bioinformatics (WISB2006), Hobart, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 73. Mikael Bodén and Timothy L. Bailey, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

and

$$x^j = \frac{\sum_{i=1}^m w_{ij} a^i}{\sum_{i=1}^m w_{ij}}, \quad j = 1, \dots, k.$$

Here $\| \cdot \|$ is an Euclidean norm and w is an $m \times k$ matrix. The problem (1)-(4) is also known as minimum sum-of-squares clustering problem.

Different algorithms have been proposed to solve the clustering problem. The paper (Jain et al. 1999) provides survey of most of existing algorithms. We mention among them heuristics like k-means algorithms and their variations (h-means, j-means etc.), mathematical programming techniques including dynamic programming, branch and bound, cutting plane, interior point methods, the variable neighborhood search algorithm and metaheuristics like simulated annealing, tabu search, genetic algorithms (see (Al-Sultan 1995, Brown & Entail 1992, de Merle et al. 2001, Diehr 1985, Dubes & Jain 1976, Hanjoul & Peeters 1985, Hansen & Jaumard 1997, Hansen & Mladenovic 2001a, Hansen & Mladenovic 2001b, Koontz et al. 1975, Selim & Al-Sultan 1991, Spath 1980, Sun et al. 1994)). Since the number of genes in gene expression data sets are very large most of these algorithms cannot be applied for clustering of samples in such data sets.

The problem (1)-(4) is a global optimization problem and the objective function ψ in this problem has many local minima. However clustering algorithms based on global optimization techniques are not applicable to even relatively large data sets. Algorithms which are applicable to such data sets can locate only local minima of the function ψ and these local minima can differ from global solutions significantly as the number of clusters increases. Another difficulty is that the number of clusters, as a rule, is not known a priori. Over the last several years different incremental algorithms have been proposed to address these difficulties. Results of numerical experiments show that an incremental approach allows one, as a rule, to locate a local solution close to global one. Consequently it can produce a better cluster structure of a data set. The paper (Bagirov & Yearwood, 2006) develops an incremental algorithm based on nonsmooth optimization approach to clustering. The global k-means algorithm was developed in (Likas et al. 2003). The incremental approach is also discussed in (Hansen et al. 2004).

In this paper we propose a new version of the global k-means algorithm for solving clustering problems in gene expression data sets. In this algorithm a starting point for the k-th cluster center is computed by minimizing so-called auxiliary cluster function. We present the results of numerical experiments with 6 gene expression data sets. These results demonstrate that the proposed algorithm improves solutions obtained by the global k-means algorithm and for some data sets this improvement is substantial.

The rest part of the paper is organized as follows: Section 2 gives a brief description of k-means and the global k-means algorithms. The nonsmooth optimization approach to clustering and an algorithm for the computation of a starting point is described in Section 3. Section 4 presents an algorithm for solving clustering problems. The results of numerical experiments are given in Section 5 and Section 6 concludes the paper.

2 k-means and the global k-means algorithms

In this section we give a brief description of k-means and the global k-means algorithms.

The k-means algorithm proceeds as follows:

1. choose a seed solution consisting of k centers (not necessarily belonging to A);
2. allocate data points $a^i \in A$ to its closest center and obtain k-partition of A;
3. recompute centers for this new partition and go to Step 2 until no more data points change cluster.

The effectiveness of this algorithm highly depends on a starting point. It converges only to a local solution which can significantly differ from the global solution in many large data sets.

The global k-means algorithm proposed in (Likas et al. 2003) computes clusters successively. At the first iteration of this algorithm the centroid of the set A is computed and in order to compute k-partition at the k-th iteration this algorithm uses centers of $k-1$ clusters from the previous iteration. The global k-means algorithm for the computation of $q \leq m$ clusters in a data set A can be described as follows.

Algorithm 1 The global k-means algorithm.

Step 1. (Initialization) Compute the centroid x^1 of the set A:

$$x^1 = \frac{1}{m} \sum_{i=1}^m a^i, \quad a^i \in A, \ i = 1, \dots, m$$

and set $k = 1$.

Step 2. Set $k = k + 1$ and consider the centers $x^1, x^2, \dots, x^{k-1}$ from the previous iteration.

Step 3. Consider each point a of A as a starting point for the k-th cluster center, thus obtaining m initial solutions with k points $(x^1, \dots, x^{k-1}, a)$; apply k-means algorithm to each of them; keep the best k-partition obtained and its centers $x^1, x^2, \dots, x^k$.

Step 4. (Stopping criterion) If $k = q$ then stop, otherwise go to Step 2.

This version of the algorithm is not applicable for clustering on middle sized and large data sets. Two procedures were introduced to reduce its complexity (see (Likas et al. 2003)). We mention here only one of them because the second procedure is applicable to low dimensional data sets. Let d^i_{k-1} be a squared distance between $a^i \in A$ and the closest cluster center among the $k-1$ cluster centers obtained so far. For each $a^i \in A$ we calculate the following:

$$r_i = \sum_{j=1}^m \min\{0, \|a^i - a^j\|^2 - d^j_{k-1}\}$$

and we take the data point $a^l \in A$ for which

$$l = \arg\min_{i=1,\dots,m} r_i$$

as a starting point for the k-th cluster center. Then k-means algorithm is applied starting from the point $x^1, x^2, \dots, x^{k-1}, a^l$ to find k cluster centers. In our numerical experiments we use this procedure.

It should be noted that k-means algorithm and its variants tend to produce only spherical clusters and they are not always appropriate for solving clustering problems. However applying k-means algorithms we assume that clusters in a data set can be approximated by n-dimensional balls.

3 Computation of starting points

The clustering problem (1)-(4) can be reformulated in terms of nonsmooth, nonconvex optimization as follows (see (Bagirov et al. 2002, Bagirov et al. 2003)):

$$\text{minimize} \quad f(x) \tag{5}$$

subject to

$$x = (x^1, \dots, x^k) \in \mathrm{IR}^{n \times k}, \tag{6}$$

where

$$f(x^1, \dots, x^k) = \frac{1}{m} \sum_{i=1}^{m} \min_{j=1,\dots,k} \|x^j - a^i\|^2. \tag{7}$$

We call f a *cluster function.* If $k > 1$, the function f is nonconvex and nonsmooth. The number of variables in problem (1)-(4) is $(m+n) \times k$ whereas in problem (5)-(6) this number is only $n \times k$ and the number of variables does not depend on the number of instances. It should be noted that in many real-world data sets the number of instances m is substantially greater than the number of features n. On the other hand in the hard clustering problems the coefficients w_{ij} are integer, that is the problem (1)-(4) contains both integer and continuous variables. In the nonsmooth optimization formulation of the clustering problem variables are continuous only. All these circumstances can be considered as advantages of the nonsmooth optimization formulation (5)-(6) of the clustering problem.

Let us consider the problem of finding k-th cluster center assuming that the centers $x^1, \dots, x^{k-1}$ for $k-1$ clusters are known. Then we introduce the following function:

$$\bar{f}^k(y) = \frac{1}{m} \sum_{i=1}^{m} \min \left\{ d^i_{k-1}, \|y - a^i\|^2 \right\} \tag{8}$$

where $y \in \mathrm{IR}^n$ stands for k-th cluster center and

$$d^i_{k-1} = \min \left\{ \|x^1 - a^i\|^2, \dots, \|x^{k-1} - a^i\|^2 \right\}.$$

The function $\bar{f}^k$ is called an *auxiliary cluster function.* It has only n variables.

Consider the set

$$\overline{D} = \left\{ y \in \mathrm{IR}^n : \|y - a^i\|^2 \geq d^i_{k-1} \right\}.$$

$\bar{D}$ is the set where the distance between any its point y and any data point $a^i \in A$ is no less than the distance between this data point and its cluster center. We also consider the following set

$$D_0 = \mathrm{IR}^n \setminus \overline{D} \equiv \{ y \in \mathrm{IR}^n :$$

$$\exists I \subset \{1, \dots, m\}, \ I \neq \emptyset : \|y - a^i\| < d^i_{k-1} \ \ \forall i \in I \}.$$

The function $\bar{f}^k$ is a constant on the set $\overline{D}$ and its value in this set is

$$\bar{f}^k(y) = d_0 \equiv \sum_{i=1}^{m} d^i_{k-1}, \quad \forall y \in \overline{D}.$$

It is clear that $x^j \in \overline{D}$ for all $j = 1, \dots, k-1$ and $a^i \in D_0$ for all $a^i \in A, \ \ a^i \neq x^j, \ j = 1, \dots, k-1$. It is also clear that $\bar{f}^k(y) < d_0$ for all $y \in D_0$.

Any point $y \in D_0$ can be taken as a starting point for the k-th cluster center. The function $\bar{f}^k$ is nonconvex function with many local minima and one can assume that the global minimum of this function can be a good candidate to be the starting point for the k-th cluster center. However it is not always possible to find the global minimum of $\bar{f}^k$ in a reasonable time. Therefore we propose an algorithm for finding a local minimum of the function $\bar{f}^k$.

For any $y \in D_0$ we consider the following sets:

$$S_1(y) = \left\{ a^i \in A : \|y - a^i\|^2 = d^i_{k-1} \right\},$$

$$S_2(y) = \left\{ a^i \in A : \|y - a^i\|^2 < d^i_{k-1} \right\},$$

$$S_3(y) = \left\{ a^i \in A : \|y - a^i\|^2 > d^i_{k-1} \right\}.$$

The set $S_2(y) \neq \emptyset$ for any $y \in D_0$.

The the following algorithm is proposed to find a starting point for the k-th cluster center.

Algorithm 2 An algorithm for finding the starting point.

Step 1. For each $a^i \in D_0 \bigcap A$ compute the set $S_2(a^i)$, its center c^i and the value $\bar{f}^k_{a^i} = \bar{f}^k(c^i)$ of the function $\bar{f}^k$ at the point c^i.

Step 2. Compute

$$\bar{f}^k_{min} = \min_{a^i \in D_0 \bigcap A} \bar{f}^k_{a^i},$$

$$a^j = \arg\min{}_{a^i \in D_0 \bigcap A} \bar{f}^k_{a^i},$$

the corresponding center c^j and the set $S_2(c^j)$.

Step 3. Recompute the set $S_2(c^j)$ and its center until no more data points escape or return to this cluster.

Let $\bar{x}$ be a cluster center generated by Algorithm 2. Then the point $\bar{x}$ is a local minimum of the function $\bar{f}^k$.

4 An incremental clustering algorithm

In this section we describe an incremental algorithm for solving cluster analysis problems.

Algorithm 3 An incremental algorithm for clustering problems.

Step 1. (Initialization). Select a tolerance $\epsilon > 0$. Compute the center $x^{1*} \in \mathrm{IR}^n$ of the set A. Let f^{1*} be the corresponding value of the objective function (7). Set $k = 1$.

Step 2. (Computation of the next cluster center). Let $x^{1*}, \dots, x^{k*}$ be the cluster centers for k-partition problem. Apply Algorithm 2 to find a starting point $y^{k+1,0} \in \mathrm{IR}^n$ for the $(k+1)$-st cluster center.

Step 3. (Refinement of all cluster centers). Take $x^{k+1,0} = (x^{1*}, \dots, x^{k*}, y^{k+1,0})$ as a new starting point, apply k-means algorithm to solve $(k+1)$-partition problem. Let $x^{1*}, \dots, x^{k+1,*}$ be a solution to this problem and $f^{k+1,*}$ be the corresponding value of the objective function (7).

Step 4. (Stopping criterion). If

$$\frac{f^{k*} - f^{k+1,*}}{f^{1*}} < \epsilon$$

then stop, otherwise set $k = k + 1$ and go to Step 2.

It is clear that $f^{k*} \geq 0$ for all $k \geq 1$ and the sequence $\{f^{k*}\}$ is decreasing, that is,

$$f^{k+1,*} \leq f^{k,*} \quad \text{for all} \quad k \geq 1.$$

The latter implies that after $\bar{k} > 0$ iterations the stopping criterion in Step 4 will be satisfied. Thus Algorithm 3 computes as many clusters as the data set A contains with respect to the tolerance $\varepsilon > 0$.

The choice of the tolerance $\varepsilon > 0$ is crucial for Algorithm 3. Large values of ϵ can result in the appearance of large clusters whereas small values can produce small and artificial clusters.

5 Results of numerical experiments

To verify the effectiveness of the proposed algorithm and to compare it with similar algorithms a number of numerical experiments with six gene expression data sets have been carried out on a Pentium-4, 2.0 GHz, PC. We also use multi-start k-means (MSKM) and global k-means (GKM) algorithms for comparison. 100 randomly generated starting points are used in MSKM. In tables below MGKM stands for the modified global k-means algorithm. In tables we present the number of clusters (N), values f of the clustering function obtained by different algorithms and CPU time (t). We used the following gene expression data sets.

5.1 Data set 1

This data set is Boston Lung Cancer data set and was generated at the Dana Farber Cancer Institute. The data set consists of 12484 genes, 185 lung tumor samples and 17 normal lung samples. Of these, there were 138 lung adenocarcinoma, 6 small-cell lung cancer, 20 carcinoid lung cancer and 21 squamous cell. Expression profiles were generated using the Affymetrix GeneChip HG_U95Av2. This data set can be accessed from Cancer Genomics expression database at the Broad Institute of MIT and Harvard. Results for this data set are presented in Table 1.

Table 1: Results for Data set 1

N	MSKM		GKM		MGKM	
	$f \times 10^{10}$	t	$f \times 10^{10}$	t	$f \times 10^{10}$	t
2	8.441	542.81	8.441	59.31	8.441	102.47
5	6.644	1652.08	6.769	240.39	6.712	415.58
10	5.703	2714.59	6.094	545.19	5.696	962.94
15	5.467	4086.98	5.556	862.45	5.177	1543.30
20	4.900	5016.28	5.041	1199.98	4.812	2150.46

Results presented in Table 1 demonstrate that MSKM algorithm produces better results when the number of clusters $N \leq 10$. However MGKM outperforms two other algorithms as the number of clusters increases. GKM requires less CPU time however its solutions are not good. MGKM requires significantly less CPU time than MSKM.

5.2 Data set 2

This is the Novartis multi-tissue data set. The data set includes tissue samples of four cancer types with 26 breast,26 prostate, 28 lung, and 23 colon samples. There are 103 samples all together and 1000 genes. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Results for this data set are presented in Table 2.

One can see from Table 2 that algorithms repform similar when the number of clusters $N \leq 5$. However GKM requires significantly less CPU time. MGKM produces better solutions than two other algorithms as the number of clusters increases. Again MGKM requires less CPU time than MSKM.

Table 2: Results for Data set 2

N	MSKM		GKM		MGKM	
	$f \times 10^{10}$	t	$f \times 10^{10}$	t	$f \times 10^{10}$	t
2	9.212	0.81	9.212	0.19	9.212	0.30
5	5.024	3.30	5.032	0.61	5.032	1.03
10	3.424	6.70	3.408	1.36	3.351	2.88
15	2.849	10.13	2.897	2.16	2.812	5.98
20	2.470	11.42	2.556	3.00	2.422	10.23

5.3 Data set 3

This is a leukemia data set with 5000 genes and 38 samples including 11 acute myeloid leukemia (AML)and 27 acute lymphoblastic leukemia (ALL) samples. The original data set is retrievable from: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Results are presented in Table 3. We calculate maximum 10 clusters because this data set contains only 38 samples.

Table 3: Results for Data set 3

N	MSKM		GKM		MGKM	
	$f \times 10^{10}$	t	$f \times 10^{10}$	t	$f \times 10^{10}$	t
2	7.880	3.06	8.137	0.58	7.880	0.67
5	5.537	8.17	5.837	2.02	5.729	2.64
10	4.104	10.47	4.399	4.59	4.271	8.19

Results from Table 3 show MSKM produces better solutions than two other algorithms, however it requires more computational time. MGKM produces better solutions than the GKM algorithm.

5.4 Data set 4

This data set includes 248 samples and 985 genes. Diagnostic bone narrow samples from pediatric acute leukemia patients corresponding to 6 prognostically important leukemia subtypes: 43 T-lineage ALL, 27 E2A-PBX1, 15 BCR-ABL, 79 TEL-AML1, 20 MLL rearrangements and 64 “hyperdiploid>50” chromosomes. The data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Computational results for this data set are presented in Table 4.

Table 4: Results for Data set 4

N	MSKM		GKM		MGKM	
	$f \times 10^{13}$	t	$f \times 10^{13}$	t	$f \times 10^{13}$	t
2	2.777	7.47	2.777	0.97	2.777	1.81
5	1.939	20.44	1.939	3.55	1.939	6.81
10	1.671	36.44	1.685	7.86	1.626	15.20
15	1.570	51.67	1.555	12.34	1.480	25.14
20	1.534	60.36	1.473	17.02	1.364	36.09

For data set 4 all three algorithms give the same solutions when the number of clusters $N \leq 5$. However, for larger number of clusters MGKM outperforms other two algorithms. GKM requires the least CPU time and MGKM requires less CPU time than MSKM.

5.5 Data set 5

This is a lung cancer data set which includes 2000 genes and 139 adenocarcinomas, 21 squamous cell carcinomas, 20 carcinoids and 17 normal lung samples. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Results are given in Table 5.

Table 5: Results for Data set 5

N	MSKM		GKM		MGKM	
	$f \times 10^{10}$	t	$f \times 10^{10}$	t	$f \times 10^{10}$	t
2	1.588	5.28	1.589	0.70	1.589	1.23
5	1.068	24.30	1.067	2.33	1.067	4.47
10	0.870	39.94	0.880	5.27	0.862	10.05
15	0.860	50.67	0.819	8.23	0.781	15.61
20	0.824	53.45	0.766	11.23	0.726	22.47

Results presented in Table 5 demonstrate that algorithms produce almost the same solutions when the number of clusters $N \leq 5$. The algorithm MGKM significantly outperforms other algorithms as the number of clusters increases. GKM requires the least CPU time and MGKM requires less CPU time than the algorithm MSKM.

5.6 Data set 6

This data set has 90 samples and 1277 genes. It contains 13 distinct tissue types: 5 breast cancer, 9 prostate, 7 lung, 11 colon, 6 germinal center cells, 7 bladder, 6 uterus, 5 peripheral blood monocytes, 12 kidney, 10 pancreas, 4 ovary, 5 whole brain and 3 cerebellum. This data set is available at: http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. Computational results for this data set are presented in Table 6.

Table 6: Results Data set 6

N	MSKM		GKM		MGKM	
	$f \times 10^{11}$	t	$f \times 10^{11}$	t	$f \times 10^{11}$	t
2	1.554	2.16	1.589	0.20	1.582	0.36
5	1.040	7.06	1.064	0.69	1.065	1.23
10	0.655	14.28	0.651	1.52	0.633	2.69
15	0.526	23.58	0.461	2.44	0.453	4.86
20	0.476	29.78	0.352	3.38	0.349	8.27

Results from Table 6 demonstrate that for small number clusters MSKM works better than other algorithms, however GKM and MGKM produce better solutions as the number of clusters increases. MGKM is best for larger number clusters. MSKM is computationally more expensive and GKM use the least CPU time.

5.7 Content of clusters

In this subsection we demonstrate the content of clusters produced by different algorithms and we use the notion of cluster purity to compare clusters. The notion of cluster purity is defined as follows:

$$P(A^i) = 100 \frac{1}{n_{A^i}} \max_{j=1,\dots,l} \quad n^j_{A^i},$$

where $n_{A^i} = |A^i|$ is the cardinality of the cluster A^i, $n^j_{A^i}$ is the number of instances in the cluster A^i that belong to the true class j and l is the number of true classes. Then the total purity $P(A)$ for the data set A can be calculated as:

$$P(A) = \frac{n_{A^i} P(A^i)}{m}.$$

We used the data set 6 and calculated 30 clusters. Results are as follows.

- MSKM algorithm produced 13 empty, 6 mixed and 11 pure clusters with total purity $P(A) = 64.44$;
- GKM algorithm produced 27 pure and 3 mixed clusters with the total purity $P(A) = 83.33$. In three mixed clusters the results were as follows:
  - Cluster 1 - 17 tumors: breast(1), lung(2), colon(2), germinal center cells (1), bladder(1), uterus(2), kidney(3), pancreas(5);
  - Cluster 2 - 4 tumors: bladder(1), uterus(3);
  - Cluster 3 - 5 tumors: whole brain(2), cerebellum(3).
- MGKM algorithm produced 27 pure and 3 mixed clusters with the total purity $P(A) = 85.56$. In three mixed clusters the results were as follows:
  - Cluster 1 - 14 tumors: breast(1), lung(2), colon(1), bladder(2), kidney(3), pancreas(5);
  - Cluster 2 - 3 tumors: colon(1), germinal center cells (1), bladder(1).
  - Cluster 3 - 5 tumors: bladder(1), uterus(3), whole brain(1);

One can see that MGKM algorithm produces better clusters than two other algorithms.

6 Conclusions

In this paper we have developed the new version of the global k-means algorithm, the modified global k-means algorithm. This algorithm computes clusters incrementally and to compute k-partition of a data set it uses $k-1$ cluster centers from the previous iteration. An important step in this algorithm is the computation of a starting point for the k-th cluster center. This starting point is computed by minimizing so-called auxiliary cluster function. The proposed algorithm computes as many clusters as a data set contains with respect to a given tolerance.

We have presented the results of numerical experiments on 6 gene expression data sets. These results clearly demonstrate that the modified global k-means algorithm proposed in this paper is efficient for solving clustering problems in gene expression data sets. It outperforms both the multi-start and global k-means algorithms as the number of clusters increases. However the proposed algorithm requires more computational efforts than the global k-means algorithm.

Acknowledgement

This research was supported by the Australian research Council.

References

Al-Sultan, K.S. (1995), A tabu search approach to the clustering problem, *Pattern Recognition*, **28(9)**, 1443-1451.

Bagirov, A.M., Rubinov, A.M. & Yearwood, J. (2002), A global optimisation approach to classification, *Optimization and Engineering,* **3(2)**, 129-155.

Bagirov, A.M., Rubinov, A.M, Soukhoroukova, N.V. & Yearwood, J. (2003), Supervised and unsupervised data classification via nonsmooth and global optimisation, *TOP: Spanish Operations Research Journal,* **11(1)**, 1-93.

Bagirov, A.M. & Yearwood, J. (2006), A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, *European Journal of Operational Research,* **170(2)**, 578-596.

Bagirov, A.M., Ferguson, B., Ivkovic, S., Saunders, G. & Yearwood, J. (2003), New algorithms for multi-class cancer diagnosis using tumor gene expression signatures, *Bioinformatics,* **19(14)**, 1800-1807.

Bock, H.H. (1998), Clustering and neural networks, In: Rizzi, A., Vichi, M. & Bock, H.H. (eds), *Advances in Data Science and Classification*, Springer-Verlag, Berlin, pp. 265-277.

Brown, D.E. & Entail, C.L. (2001), A practical application of simulated annealing to the clustering problem, *Pattern Recognition*, **25(4)**, 401-412.

de Merle, O., Hansen, P., Jaumard, B. & Mladenovic, N. (2001), An interior point method for minimum sum-of-squares clustering, *SIAM J. on Scientific Computing,* **21,** 1485-1505.

Diehr, G. (1985), Evaluation of a branch and bound algorithm for clustering, *SIAM J. Scientific and Statistical Computing*, **6,** 268-284.

Dubes, R. & Jain, A.K. (1976), Clustering techniques: the user's dilemma, *Pattern Recognition*, **8,** 247-260.

Hanjoul, P. & Peeters, D. (1985), A comparison of two dual-based procedures for solving the p-median problem, *European Journal of Operational Research,* **20,** 387-396.

Hansen, P. & Jaumard, B. (1997), Cluster analysis and mathematical programming, *Mathematical Programming,* **79(1-3),** 191-215.

Hansen, P. & Mladenovic, N. (2001a), J-means: a new heuristic for minimum sum-of-squares clustering, *Pattern Recognition*, **4,** 405-413.

Hansen, P. & Mladenovic, N. (2001b), Variable neighborhood decomposition search, *Journal of Heuristic,* **7,** 335-350.

Hansen, P., Ngai, E., Cheung, B.K. & Mladenovic, N. (2001b), Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering, submitted.

Houkins, D.M. , Muller, M.W. & ten Krooden, J.A., (2001b), Cluster analysis, In: *Topics in Applied Multivariate Analysis*, Cambridge University press, Cambridge.

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

Jensen, R.E. (1969), A dynamic programming algorithm for cluster analysis, *Operations Research,* **17,** 1034-1057.

Koontz, W.L.G., Narendra, P.M. & Fukunaga, K. (1975), A branch and bound clustering algorithm, *IEEE Transactions on Computers*, **24,** 908-915.

Likas, A., Vlassis, M. & Verbeek, J. (2003), The global k-means clustering algorithm, *Pattern Recognition*, **36,** 451-461.

Medvedovic, M. & Sivaganesan, S. (2002), Bayesian infinite mixture model based clustering gene expression profiles, *Bioinformatics*, **18,** 1194-1206.

Selim, S.Z. & Al-Sultan, K.S. (1991), A simulated annealing algorithm for the clustering, *Pattern Recognition*, **24(10)**, 1003-1008.

Spath, H. (1991), *Cluster Analysis Algorithms*, Ellis Horwood Limited, Chichester.

Sun, L.X., Xie, Y.L., Song, X.H., Wang, J.H. & Yu, R.Q. (1994), Cluster analysis by simulated annealing, *Computers and Chemistry*, **18,** 103-108.

Yeung, K.Y., Fraley, C., Murua, A., Raftery, A.E. & Ruzzo, W.L. (2001), Model-based clustering and data transformations for gene expression data, *Bioinformatics,* **17,** 977-987.

Yeung, K.Y. , Medvedovic, M. & Bumgarner, R.E., (2003), Clustering gene expression data with repeated measurements, *Genome Biol.,* **4,** R34.