A NEW MODIFIED GLOBAL K-MEANS ALGORITHM FOR CLUSTERING LARGE DATA SETS

3 authors, including:

A.M. Bagirov
Federation University Australia
184 PUBLICATIONS **2,019** CITATIONS
SEE PROFILE

Julien Ugon
Deakin University
71 PUBLICATIONS **532** CITATIONS
SEE PROFILE

The XIII International Conference
"Applied Stochastic Models and Data Analysis"
(ASMDA-2009)
June 30-July 3, 2009, Vilnius, LITHUANIA

ISBN 978-9955-28-463-5
L. Sakalauskas, C. Skiadas and
E. K. Zavadskas (Eds.): ASMDA-2009
Selected papers. Vilnius, 2009, pp. 1–5

A NEW MODIFIED GLOBAL K-MEANS ALGORITHM FOR CLUSTERING LARGE DATA SETS

Adil M. Bagirov[1], Julien Ugon[2] and Dean Webb[3]

Centre for Informatics and Applied Optimization,
School of Information Technology and Mathematical Sciences,
University of Ballarat, Victoria, 3353, Australia
E-mail: [1]a.bagirov@ballarat.edu.au

Abstract: The k-means algorithm and its variations are known to be fast clustering algorithms. However, they are sensitive to the choice of starting points and inefficient for solving clustering problems in large data sets. Recently, in order to resolve difficulties with the choice of starting points, incremental approaches have been developed. The modified global *k*-means algorithm is based on such an approach. It iteratively adds one cluster center at a time. Numerical experiments show that this algorithm considerably improve the *k*-means algorithm. However, this algorithm is not suitable for clustering very large data sets. In this paper, a new version of the modified global k-means algorithm is proposed. We introduce an auxiliary cluster function to generate a set of starting points spanning different parts of the data set. We exploit information gathered in previous iterations of the incremental algorithm to reduce its complexity.

Keywords: clustering, nonsmooth optimization, *k*-means, global *k*-means.

1. Introduction

Cluster analysis, also known as unsupervised data classification, is an important subject in data mining. Its aim is to partition a collection of patterns into clusters of similar data points. There are different types of clustering and in this paper we consider the unconstrained hard clustering problem. The *k*-means algorithm is a fast algorithm for solving such problems, which makes it applicable to very large data sets. The main drawback of the *k*-means algorithm is that it is very sensitive to the choice of starting points. One common way of avoiding this problem is to use the multi restarting *k*-means algorithm. However, as the size of the data set and the number of clusters increase, more and more starting points are needed to get a near global solution to the clustering problem. Consequently the *k*-means algorithm becomes very time consuming and inefficient for solving clustering problems, even in moderately large data sets.

Different approaches to improve the efficiency of the *k*-means algorithm have been proposed, of which incremental ones are among the most successful. In these approaches clusters are computed incrementally by solving all intermediate clustering problems (Bagirov, 2008, Bagirov and Yearwood, 2006, Hansen, 2005, Likas et al. 2003). Results of numerical experiments on real world data sets show that algorithms based on an incremental approach allow one to find a near global minimizer of the cluster (or error) function. However, these algorithms are time consuming on very large data sets and the solution they find may differ from the global one. This difference may increase as the number of clusters increases.

In this paper, a new incremental algorithm is proposed for clustering large data sets. We introduce an auxiliary cluster function to generate a set of starting points lying in different parts of the data set at each iteration. The *k*-means algorithm is applied starting from these points and the best solution is selected as a starting point for the next cluster center. We exploit information from previous iterations of the algorithm to reduce its complexity.

2. Cluster and auxiliary cluster functions

In cluster analysis we assume that we have been given a finite set A of points in the n-dimensional space R^n, that is $A = \{a^1,...,a^m\} \subset R^n$. 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 $A^j, j=1,...,k$ called clusters. We assume that each cluster A^j can be identified by its center (or centroid). There are different reformula-

ions of clustering as an optimization problem. A nonsmooth, nonconvex optimization formulation is as follows (Bagirov et al., 2003):

$$\min f(x^1,...,x^k) \text{ subject to } x^1,...,x^k \in R^n,$$

$$\text{where } f(x^1,...,x^k) = \frac{1}{m}\sum_{i=1}^{m} \min\{\|x^1 - a^i\|^2,...,\|x^k - a^i\|^2\}. \quad (1)$$

Here $\|\cdot\|$ is an Euclidean norm. Assume that the solution $x^1,...,x^{k-1}$ to the *(k-1)*-clustering problem is known. We introduce the auxiliary cluster function (Bagirov, 2008):

$$\bar{f}_k(x) = \frac{1}{m}\sum_{i=1}^{m} \min\{d_{k-1}^i, \|x - a^i\|^2\}, x \in R^n. \quad (2)$$

Here d_{k-1}^i is the squared distance between a^i and the closest center among *k-1* cluster centers: $d_{k-1}^i = \min\{\|x^1 - a^i\|^2,...,\|x^{k-1} - a^i\|^2\}$.

3. Selection of starting points

Both the global *k*-means and modified global *k*-means algorithms involve schemes to generate good starting points for the *k*-means algorithm. They use an incremental approach and a special procedure to find the next cluster center. In the following we discuss new algorithms for finding the set of starting points and for reducing the complexity of the modified global *k*-means algorithm. Let U be a finite set of positive numbers and $u \in U$. We modify the auxiliary function $\bar{f}_k$ as $\bar{f}_k^u(x) = \frac{1}{m}\sum_{i=1}^{m} \min\{d_{k-1}^i, u\|x - a^i\|^2\}, x \in R^n$.

For a given $y \in R^n$ consider the set $S^u(y) = \{a^i \in A : u\|y - a^i\|^2 < d_{k-1}^i\}$.

Algorithm 1. An algorithm to find a set of starting points.

Step 1. For each $u \in U$ and $a^i \in A$ compute the set $S^u(a^i)$, its center c^i and the value $\bar{f}_{k,a^i}^u = \bar{f}_k^u(c^i)$ of the function $\bar{f}_k^u$ at the point c^i.

Step 2. Compute $\bar{f}_{k,a^i}^u = \min_{a^i \in A} \bar{f}_{k,a^i}^u$, $a^j = \arg\min_{a^i \in A} \bar{f}_{k,a^i}^u$, the corresponding center c^j and the set $S^u(c^j)$.

Step 3. Recompute the set $S^u(c^j)$ and its center until no more data points escape or return to this cluster. The final solution *c(u)* is accepted as a starting point.

Algorithm 1 generates a set Q of starting points $Q = \{c(u), u \in U\}$.

Algorithm 2 Multi-start modified global *k*-means algorithm.

Step 1. Select a tolerance $\varepsilon > 0$. Compute the center $x^1 \in R^n$ of the set A. Let f^1 be the corresponding value of the objective function (1). Set $k=1$.

Step 2. Set $k=k+1$. Let $x^1,...,x^{k-1}$ be the cluster centers for the *(k-1)*-partition problem. Apply Algorithm 1 to find a set of starting points Q for the *k*-th cluster center.

Step 3. For all $u \in U$ select $(x^1,...,x^{k-1},c(u))$ as a new starting point, apply the *k*-means algorithm to solve the *k*-partition problem. Let $(y^1(u),...,y^k(u))$ be a solution to this problem and $\bar{f}_k^u$ be the corresponding value of (2).

Step 4. Compute $\bar{f}_k = \min_{u \in U} \bar{f}_k^u$ and $u \in U$ such that $\bar{f}_k^u = \bar{f}_k$.

Step 5. If $(f^{k-1} - f^k)/f < \varepsilon$ then stop, otherwise set $x^i = y^i(u), i = 1,...,k$ and go to Step 2.

4. Reduction of complexity

Both the global and the modified global algorithms use an affinity (or distance) matrix at each iteration. In large data sets this matrix cannot be stored in memory. An incremental approach provides information to further decrease the complexity of the *k*-means algorithm and to avoid to compute the whole affinity matrix.

In the proposed algorithm the most time consuming step is Step 2, where we apply the reduced *k*-means algorithm (Algorithm 1) to minimize the auxiliary function for different $u \in U$. Since for large data

sets it is not possible to store the affinity matrix in memory one needs to repeatedly compute it at each iteration of the incremental algorithm. As a result this algorithm is very time consuming for large data sets.

We consider two schemes to reduce the complexity of Algorithm 1. Both schemes exploit the incremental nature of Algorithm 2. We suggest to use the distances between data points and cluster centers instead of the affinity matrix. Since the number of clusters is significantly less than the number of data points the former matrix is much smaller than the latter one. Let $d_{il} = \left\|x^l - a^i\right\|^2$ be the squared distance between the data point a^i and the cluster center x^l. We can consider a matrix $D_{k-1} = (d_{il}), i = 1,...,m, l = 1,...,k-1$. We also consider the vector $\overline{D}_{k-1} = (d^1_{k-1},...,d^m_{k-1})$ of m components where $d^i_{k-1}, i = 1,...,m$ is defined above. The matrix D_{k-1} and the vector $\overline{D}_{k-1}$ are known after the *(k-1)*-st iteration.

We use two ways to reduce the complexity of Algorithm 1. The first one is very simple. For a given $u \in U$ and data point a^i if $d_{ij} \geq (1+1/u)^2 d^j_{k-1}$ then $a^j \notin S^u(a^i)$ and we do not compute the distance between a^i and a^j.

The second approach is based on the fact that data points which are very close to previous cluster centers cannot be considered as candidates to be starting point for the next cluster center. At the *(k-1)*-st iteration we can compute a squared radius of each cluster $A_l, l = 1,...,k-1$ as follows:

$$r_l = \max\{\left\|x^l - a\right\|^2, a \in A_l\}.$$

Let ε>0 be a given tolerance. Then we can consider the following subset of the cluster A_l:

$\overline{A}_l = \{a \in A_l : \left\|x^l - a\right\|^2 \geq \varepsilon r_l\}$. In other words we remove from the cluster A_l all points *a,* for which: $\left\|x^l - a\right\|^2 < \varepsilon r_l$. Consider the set $\overline{A} = \bigcup_{l=1}^{k-1} \overline{A}_l$. Then we can rewrite Step 1 of Algorithm 1 as follows:

Step 1. For each $u \in U$ and $a^i \in \overline{A}$ compute the set $S^u(a^i)$, its center c^i and the value $\bar{f}^u_{k,a^i} = \bar{f}^u_k(c^i)$ of the function $\bar{f}^u_k$ at the point c^i.

5. Numerical experiments

To verify the efficiency of the proposed algorithm numerical experiments with 4 real-world large data sets have been carried out on a Pentium 4 1.83 GHz CPU and 1 GB RAM. A brief description of these data sets is given in Table 1 (see also Asuncion and Newman, 2007).

Table 1. Brief description of data sets

Data sets	No. instances	No. attributes
Shuttle control	58000	9
Letter recognition	20000	16
Magic telescope	19020	10
Pendigit	10992	16

We apply the proposed algorithm with $|U|=2$ and compare our results with the global *k*-means algorithm (GKM). We computed up to 100 clusters in all data sets. Results of numerical experiments are presented in Table 2. In these tables we use the following notation:

- *k* is the number of clusters;
- fopt is the best known value of the cluster function (1) (multiplied by *m*) for the corresponding number of clusters. We take as *fopt* the best value obtained by the GKM and the proposed algorithm. We use its representation in the form $fopt = d \cdot 10^l$ and present *d* in Table 2. l=6 for Letters data set and l=8 for all other data sets.

- E is the error in % and it is calculated as follows: $E = 100(\bar{f} - fopt)/fopt$. Here $\bar{f}$ is the best value (multiplied by m) of the objective function (1) obtained by an algorithm. $E=0$ implies that an algorithm finds the best known solution.
- N is the number of Euclidean norm evaluations for the computation of the corresponding number of clusters. To avoid big numbers being in the tables we express them in the form $N = \alpha \cdot 10^l$ and present the values of α in Table 2. $l=8$ for all data sets.
- t is the CPU time (in seconds).

Results from Table 2 demonstrate that the proposed algorithm produces more accurate solutions and uses significantly less CPU time and norm evaluations than the GKM algorithm. Moreover the proposed algorithm produces better results than the GKM algorithm in 16 cases, whereas the GKM algorithm finds better solutions in 4 cases. In 12 cases both algorithms find the same solutions.

6. Conclusions

In this paper, we have developed a new version of the modified global *k*-means algorithm. This algorithm computes clusters incrementally, using the cluster centers from the previous iteration to solve the *k*-partition problem. 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 the so-called auxiliary cluster function. One can expect that the global minimizer of this function is a good candidate as a starting point for the *k*-th cluster center. In order to find global or near global solutions, we use more than one starting point to minimize the auxiliary function. We introduce weights within the auxiliary function in order to generate starting points from different parts of the data set. We apply the *k*-means algorithm to minimize the auxiliary function using those starting points. Two schemes to reduce the complexity of the algorithm are introduced. The results of numerical experiments demonstrate that the proposed algorithm is faster and more accurate than the global *k*-means algorithm for clustering large data sets. Furthermore this improvement becomes more substantial as the size of the data set increases.

Dr. Adil Bagirov is the recipient of an Australian Research Council Australian Research Fellowship (Project number: DP 0666061).

Table 2. Results of numerical experiments

K	*fopt*	GKM			New MGKM		
		E	α	t	E	α	t
Pendigit							
2	1.28120	0.39	1.21	9.30	0.00	1.92	18.84
10	0.49302	0.00	10.97	81.08	0.00	9.23	111.00
20	0.34123	0.00	23.29	166.58	0.17	13.18	196.78
40	0.23472	0.00	48.52	339.50	0.01	18.40	341.31
50	0.21131	0.37	61.27	426.14	0.00	20.35	400.88
60	0.19399	0.00	74.16	513.39	0.16	22.12	454.09
80	0.16982	0.10	100.30	689.23	0.00	25.81	547.39
100	0.15316	0.07	126.94	867.17	0.00	28.98	622.47
Magic Telescope							
2	2.00960	0.00	3.62	24.59	0.00	6.24	51.30
10	0.84035	0.00	32.74	202.00	0.00	13.70	128.33
20	0.59799	0.09	70.30	422.67	0.00	21.54	238.73
40	0.44466	0.00	145.71	820.02	0.00	32.79	450.77
50	0.40398	0.00	184.56	1022.19	0.00	38.80	553.30
60	0.37402	0.09	224.17	1226.36	0.00	45.17	649.16

K	*fopt*	GKM			New MGKM		
		E	α	t	E	α	t
80	0.33217	0.00	302.47	1624.84	0.20	55.29	808.25
100	0.30260	0.09	383.54	2033.98	0.00	65.78	947.53
Letters							
2	1.38190	0.00	4.01	40.17	0.00	6.34	115.44
10	0.85752	0.00	36.47	363.78	0.00	21.45	400.81
20	0.67620	0.19	77.62	732.23	0.00	36.33	725.72
40	0.51925	0.78	161.25	1453.70	0.00	58.16	1264.69
50	0.47837	0.06	203.30	1812.06	0.00	66.67	1479.50
60	0.44274	0.12	246.19	2173.91	0.00	74.27	1668.38
80	0.39285	0.23	331.58	2892.00	0.00	88.24	1985.80
100	0.35671	0.21	417.47	3611.05	0.00	98.73	2222.48
Shuttle control							
2	21.34300	0.00	33.64	212.19	0.00	0.40	13.77
10	2.83170	0.02	302.86	1980.22	0.00	5.64	147.63
20	1.06010	0.00	639.75	4286.66	0.00	42.44	765.09
40	0.37540	0.85	1317.37	8644.75	0.00	152.39	2793.89
50	0.25937	0.93	1655.04	10436.01	0.00	184.47	3582.50
60	0.20725	0.00	1995.10	12281.36	1.21	217.19	4382.47
80	0.14348	0.16	2679.18	15930.31	0.00	278.00	5903.14
100	0.10591	0.00	3365.34	19538.72	0.00	327.20	7109.78

References

Bagirov, A. M. 2008. Modified global -means algorithm for sum-of-squares clustering problem, *Pattern Recognition*, 41, 3192–3199.

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

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

Hansen, P., Ngai, E., Cheung, B. K., Mladenovic, N. 2005. Analysis of global -means, an incremental heuristic for minimum sum-of-squares clustering, *J. of Classification* 22(2): 287–310.

Likas, A., Vlassis, M. and Verbeek, J. 2003. The global -means clustering algorithm, *Pattern Recognition* 36: 451–461.

Asuncion, A. and Newman, D. J. 2007. *UCI Machine Learning Repository*. Irvine, CA: Uni. of California. Available from Internet: http://www.ics.uci.edu/ mlearn/MLRepository.html.