Efficient global clustering using the Greedy Elimination Method

Z.S.H. Chan and N. Kasabov

A novel global clustering method called the Greedy Elimination Method is presented. Experiments show that the proposed method scores significantly lower clustering errors than the standard *K*-means over two benchmark and two application datasets, and it is efficient for handling large datasets.

Introduction: The *K*-means algorithm is used widely either as a stand-alone clustering method, or as a fast method for computing the optimal initial cluster centres for more expensive clustering methods. It employs a simple iterative scheme that performs hill climbing from initial centres, whose values are usually randomly picked from the training data. Although the algorithm is very efficient, it suffers two well-known problems: (i) the solutions are only locally optimal, and (ii) their qualities are sensitive to the initial conditions (i.e. the values of the initial centres). This Letter presents an efficient global clustering method called the Greedy Elimination Method (GEM) for alleviating these problems.

Problem definition and Greedy Elimination (GEM) algorithm: With the conventional *K*-means algorithm, the clustering task is to cluster N samples of training data $\mathbf{X}=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ into K Voronoi partitions defined by the cluster centres $\mathbf{M}=\{\mathbf{m}_1,\ldots,\mathbf{m}_K\}$. The most common clustering criterion $E(\mathbf{X},\mathbf{M})$ is the minimisation of the clustering error, which is defined as the sum of squared Euclidean distances between each data point to its nearest cluster centre. Let C_k, $k=[1, 2, \ldots, K]$ represent K disjoint subsets such that $(\mathbf{x}_n \in C_k)$ if $k=\arg\min_i(\|\mathbf{x}_n-\mathbf{m}_i\|^2)$. $E(\mathbf{X},\mathbf{M})$ is given by

$$E(\mathbf{X},\mathbf{M})=\sum_{n=1}^{N}\sum_{k=1}^{K} I(\mathbf{x}_n \in C_k)\|\mathbf{x}_n-\mathbf{m}_k\|^2 \qquad (1)$$

where $I(X)=1$ if X is true and 0 otherwise.

Let $\alpha>1$ represent the enlargement factor for the desired number of centres K. GEM begins by obtaining a solution of αK centres using the standard *K*-means (with random initialisation), and then eliminates them one-by-one until K centres remain in the solution. Let $\mathbf{M}^*(J)$ denote the optimal solution for J centres. The kernel operation of GEM is the greedy elimination of the centres for obtaining $\mathbf{M}^*(J-1)$ given $\mathbf{M}^*(J)$, and it proceeds as follows. From $\mathbf{M}^*(J)$ we extract J sets of reduced solutions $\mathbf{M}_{-j}$ $j=[1,2,\ldots,J]$, where $\mathbf{M}_{-j}$ is given as $\mathbf{M}^*(J)$ minus the jth centre. Next we perform *K*-means on each reduced solution $\mathbf{M}_{-j}$ to obtain the corresponding optimal solutions $\mathbf{M}^*_{-j}$ and clustering errors. The solution that yields the lowest clustering error is regarded as $\mathbf{M}^*(J-1)$, which is the optimal solution for $(J-1)$ centres. Thus for GEM to compute the optimal solution for K centres, we set the initial solution to $\mathbf{M}^*(\alpha K)$ and then compute $\mathbf{M}^*(\alpha k-1)$ from $\mathbf{M}^*(\alpha K)$ using the greedy method. Next, we compute $\mathbf{M}^*(\alpha K-2)$ from $\mathbf{M}^*(\alpha K-1)$ and so on until $\mathbf{M}^*(K)$ is obtained. An illustration of GEM is shown in Fig. 1.

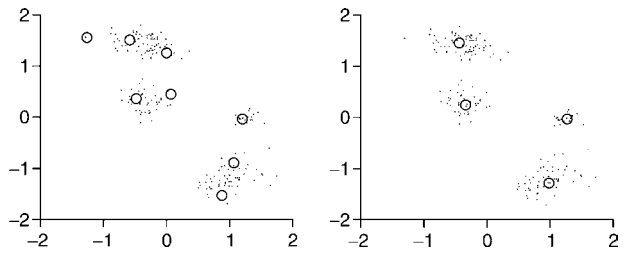

Fig. 1 *Demonstration of GEM on clustering an artificial dataset into four clusters*

a We use an enlargement factor of α=2 to begin with eight initial centres, which provide knowledge on the global distribution of the data
b The centres are eliminated one-by-one until only four centres remain

At each stage of eliminating one centre from $\mathbf{M}^*(J)$, GEM requires J runs of *K*-means to evaluate the clustering errors of the reduced solutions $\mathbf{M}_{-j}$, $j=[1,2,\ldots,J]$. It must be noted that the computational intensity of each *K*-means run varies according to the number of centres used. In the efficiency comparisons, we ignore this factor because it is insignificant relative to the total number of *K*-means run incurred by different algorithms. This operation constitutes the main computational bottleneck, costing GEM a total of $(K_{start}+K_{start}-1+\ldots+K)\approx(\alpha^2-1)K^2/2$ runs of *K*-means. A much faster version of GEM can be achieved by using the upper bound of clustering errors instead of the *K*-means optimised clustering errors, which can be computed efficiently by: (i) storing the table that records the Euclidean distance between the data points and the centres from computing $\mathbf{M}^*(j)$, and (ii) summing the minimum distance between each data point to all except the jth centre in the table. Let $U_{-j}(\mathbf{X},\mathbf{M}^*(J))$ denote the upper bound clustering error for the reduced solution $\mathbf{M}_{-j}$. It is formally given as

$$U_{-j}(\mathbf{X},\mathbf{M}^*(J))=\sum_{n=1}^{N}\sum_{k=1,k\neq j}^{K} I(\mathbf{x}_n \in C_k)\|\mathbf{x}_n-\mathbf{m}^*_k\|^2 \qquad (2)$$

We apply *K*-means to the reduced solution that yields the least upper-bound error to retrieve $\mathbf{M}^*(J-1)$. The fast GEM requires only one run of *K*-means for each stage of centre reduction, and therefore a total of $(\alpha K-K)=(\alpha-1)K$ runs of *K*-means. Empirical tests show that the fast GEM yields very competitive results to the standard GEM, yet requires much shorter computational time.

GEM achieves two important objectives: global clustering and efficiency. It achieves global clustering because, by using an initial solution of more than K centres, GEM covers a larger portion of the search space, and therefore gains more knowledge on the global distribution of the data in the beginning of the clustering process. By comparing the optimality of different centre solutions and eliminating the sub-optimal centres during the greedy elimination process, GEM uses this knowledge to guide the search towards the globally optimal region. One can draw an analogy between the use of more than K centres in GEM for global clustering, and the use of multiple search points in evolutionary algorithms (such as genetic algorithms) for global optimisation.

GEM is also more efficient than many global clustering methods proposed in the literature because of two factors. First, empirical tests show that a small enlargement factor of $\alpha=2$ is often sufficient. Second, the required number of *K*-means runs scale only as K^2 for the standard version and as K for the fast version. Since K is usually a small number lying in the range [2, 20] and is much smaller than the number of data points N, this requirement is much smaller than other global clustering methods, like the Greedy method by Likas [1] that scales as NK, and the genetic algorithm by Maulik [2] that scales as (population size × max.generation), where the values of the population size and maximum generation are [10,100] and 1000, respectively.

In addition to its global clustering property and efficiency, GEM has two other advantages: first, GEM generates the clustering error of the solutions for the αK to K centres during the greedy elimination process. We use this information to determine the optimal number of centres using criteria like Akaike or Bayesian Information Criteria. Second, the algorithmic structure of the standard version of GEM is suitable for parallel computing and we can easily achieve significant speed-up through distributing the task of evaluating the clustering error of each reduced solution on multiple nodes.

Experiments: We compare the performance between the standard *K*-means, the fast GEM and the standard GEM over clustering (unsupervised) two benchmark datasets and two application datasets into 2 to 10 clusters. Performance is measured in sum square clustering error. For the standard *K*-means, we perform 20 runs of clustering for each of the 2 to 10 clusters. For both the fast GEM and standard GEM, we initialise with $K_{start}=20$ centres (which gives the scaling factor $\alpha=2$ for $K=10$ centres) and then obtain the optimal solutions and clustering errors for $(10, 9, \ldots, 2)$ centres progressively in the same run. Results from 20 runs are obtained. The stopping criterion for the *K*-means iterations is if the clustering error decreases by less than 0.01%. All algorithms initialise the centre values by sampling randomly from the training data. They are coded in Matlab and tested on a P4 2.4 GHz PC.

The sizes and the dimensions of the datasets span a large range of [58, 2000] and [9, 256], respectively. The two benchmark datasets are the Breast Cancer dataset, which consists of 683 samples of nine features, and the Glass Identification dataset, which consists of 214 samples of nine features. Both datasets are obtained from the UCI

database. The first application dataset comes from a gene expression classification problem and it consists of the microarray data of 58 diffuse large B-cell lymphoma (DLBCL) patients [3]. The values of 11 selected genes are used as the feature variables. The second application dataset is a texture segmentation problem and it consists of 2000 patches of 16×16 pixel images randomly sampled from ten 256×256 pixel Brodatz texture images [4]. Each 16×16 pixel images is expressed as a 256×1 vector of feature variables.

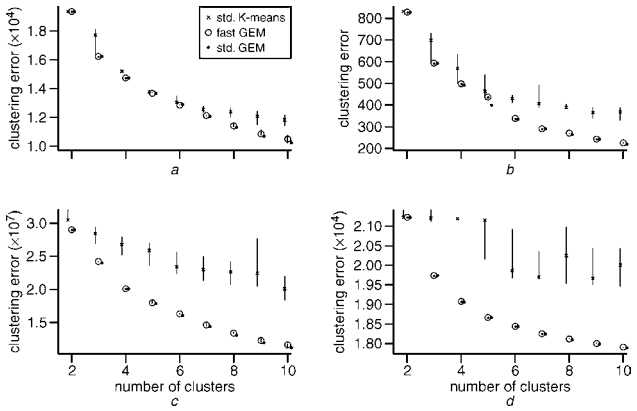

Fig. 2 *Boxplot of clustering errors scored by standard K-means, fast GEM and standard GEM for clustering datasets of 2–10 clusters*

The markers indicate the median value and the lines indicate the quartiles.
a Breast cancer *b* Glass *c* DLBCL *d* Brodatz

The comparisons of the clustering errors for 2–10 clusters are shown in Fig. 2, and the comparisons of the computational times for clustering the data into ten clusters are shown in Table 1. For all four datasets, the GEM methods score much lower clustering errors, and much smaller variance (the quartile marker lines are much shorter) than the standard *K*-means. The improvements are bigger with more cluster centres because the number of sub-optimal solutions increases, making the global clustering property of GEM more prominent over the local clustering property of standard *K*-means. In all cases, the fast GEM performs as well as or only slightly worse than the standard GEM, but requires much shorter computational time. It requires only $\sim K$ times more than that of the standard *K*-means, which is much lesser the greedy method by Likas [1] and the genetic algorithm by Maulik [2] mentioned earlier. Considering that K is a small integer in most applications, the fast GEM is hence a very cost-effective solution. Its feasibility for large datasets is demonstrated in the task of clustering the Brodatz dataset that consists of 2000 samples of 256 variables into ten clusters, for which it only spends 5.4 s.

Table 1: Computation time required by standard *K*-means, fast GEM and standard GEM to cluster the Breast Cancer, Glass, DLBCL and Brodatz datasets into ten clusters

	No. data	No. dim	Computational time, s		
			std. *K*-means	std. GEM	fast GEM
Breast Cancer	683	9	0.059	5.11	0.49
Glass	214	9	0.026	1.78	0.18
DLBCL	58	11	0.0085	0.66	0.066
Brodatz	2000	256	0.8	104	5.4

Acknowledgment: This research is supported by the KEDRI research fund.

4 October 2004
Electronics Letters online no: 20046785
doi: 10.1049/el:20046785

Z.S.H. Chan and N. Kasabov (*Knowledge Engineering Discovery and Research Institute, Auckland University of Technology, New Zealand*)

E-mail: shun.chan@aut.ac.nz

References

1 Likas, A., Vlassis, N., and Verbeek, J.: 'The global *k*-means clustering algorithm', *Pattern Recognit.*, 2003, **36**, pp. 451–461
2 Maulik, J., and Bandyopadhyay, S.: 'Genetic algorithm-based clustering technique', *Pattern Recognit.*, 2000, **33**, pp. 1455–1465
3 Shipp, M.A., Ross, K.N., Tamayo, P., and Weng, A.P.: 'Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning', *Nature Medicine*, 2002, **8**, pp. 68–74
4 Brodatz, P.: 'Textures: a photographic album for artists and designers' (Dover, New York, 1966)