ELSEVIER

Pattern Recognition Letters 17 (1996) 451–455

Pattern Recognition Letters

New methods for the initialisation of clusters

Moh'd B. Al-Daoud, Stuart A. Roberts *

School of Computer Studies, University of Leeds, Leeds, LS2 9JT, United Kingdom

Received 14 December 1994

Abstract

One of the most widely used clustering techniques is the *k*-means algorithm. Solutions obtained from this technique are dependent on the initialisation of cluster centres. In this article, two initialisation methods are developed. These methods are particularly suited to problems involving very large data sets. The methods have been applied to different data sets and good results are obtained.

Keywords: Clustering; Cluster initialisation; *k*-means algorithm

1. Introduction

Clustering techniques have received attention in many areas, such as engineering, medicine, biology and computer science. The purpose of clustering is to group together data points which are close to one another. The *k-means* algorithm (MacQueen, 1967) is one of the most widely used techniques for clustering. Most of the vector quantization techniques, for example, are based on the *k*-means method (Gersho and Gray, 1992), which being a heuristic method, yields a solution that is not guaranteed to be optimal, but which depends on the choice of the initial cluster centres (Babu and Murty, 1993; Nasrabadi and King, 1988).

Two simple approaches to cluster centre initialisation are either to select the initial values randomly, or to choose the first K samples of the data points (MacQueen, 1967). As an alternative, different sets of initial values are chosen (out of the data points) and the set which is closest to optimal is chosen. However, testing different initial sets is considered impracticable, especially for a large number of clusters (Ismail and Kamel, 1989). For example, the number of complete enumerations of every possible partition of 50 data points into five clusters is 7.4×10^{32} (Kaufman and Rousseeuw, 1990, p. 115).

Linde et al. (1980) proposed the Binary Splitting (BS) technique to initialise the clusters. The BS algorithm is as follows.

1. Calculate the value of the centroids of the data points, c_i, and assign this value as the initial centre for the first cluster;
2. Double the number of the initialised clusters by splitting each value c_i into two close data points according to the rule:

$$\underline{c}_i^+ = \underline{c}_i + \underline{\epsilon}, \qquad \underline{c}_i^- = \underline{c}_i - \underline{\epsilon},$$

* Corresponding author.

0167-8655/96/$12.00 © 1996 Elsevier Science B.V. All rights reserved
SSDI 0167-8655(95)00119-0

where ϵ is a small splitting parameter and i varies from 1 to the current number of the initialised clusters (which is 1 in the first case);

3. Calculate the centroids of the new clusters.

4. Repeat steps 2 and 3 until the desired number of clusters is achieved.

One problem with the binary splitting technique is that it requires there be at least 10 or 20 data points in each cluster (Buzo et al., 1980), which makes it impractical for applications that use fewer data points in each cluster. Another problem is the "empty cell" problem, which occurs when a new centroid is assigned to an empty cluster. This leads to an inefficient use of the cluster centres.

Huang and Harris (1993) proposed the Direct-Search Binary Splitting (DSBS) method, using the Principle Component Analysis (PCA) technique. However, it has been pointed out that there is no point in carrying on the PCA if the data points are approximately uncorrelated (Chatfield and Collins, 1980, p. 57).

Another initialisation method is the Simple Cluster-Seeking (SCS) method (Tou and Gonzales, 1974). It should be noted that this method was designed to be a clustering method rather than an initialisation method. However, it can be used to provide initialisation for other cluster methods (see Gersho and Gray, 1992, p. 362).

The SCS has the following steps: Initialise the first cluster centre with the value of the first data point. Next compute the distance between this (first) cluster centre and the next data point. If it is less than some threshold, continue, else add the new data point to the cluster centres. With each new point, find the nearest cluster centre. If the resulting distance is not within the threshold, add the data point to the cluster centres. Continue in this fashion until the desired number of clusters is achieved. If the number of cluster is not achieved, the threshold value must be reduced and the process is repeated. The above method depends on the order in which the data points are considered and on the threshold settings.

Babu and Murty (1993) used Genetic Algorithms (GAs) to initialise the clusters. GAs start with one population of a fixed size of candidate solutions (strings). During each iteration step, called a generation, the solutions of the current population are evaluated, and, on the basis of those evaluations, a new population of candidate solutions is formed.

Solutions in a population are evaluated based on their objective (fitness) value in order to evolve a new population. From the current population, the next population is generated using three probabilistic operators: reproduction, crossover and mutation operators. These operators play an important role in allowing solutions with high fitness to contribute in the next generation. To initialise the clusters, each solution in the population is mapped into a value. The k-means algorithm is then run using each of these values as an initial cluster value and the value with the least within-group sum of square distance is chosen as the final value to initialise the cluster. This process is continued until a maximum number of generations is reached. However, the complexity of this technique is very high, since the k-means is run as many times as the number of the values (solutions) in each population, which is an expensive process.

Most recently, Katsavounidis et al. (1994) proposed a method (it will be termed KKZ) which gave better results than the BS technique. The method works in the following way.

1. Initialise the first cluster with the data point at x, say, with maximum norm;

2. Initialise the second cluster with the data point which is farthest from x;

3. Compute the minimum distances between each of the remaining data points and all current initial cluster centre values, find the data point with the largest value of of these minimum distances and assign it as the next cluster initial value;

4. Repeat step 3 until the desired number of clusters is initialised.

In the following sections, two methods are presented for cluster initialisation. These methods require the division of the data space into a number of cells. Both methods initialise the clusters according to the density of data points, in method-1 so that the density of cluster centroids is proportional to density of data points; method-2 provides an optimal solution under simplifying assumptions.

Both methods are tested on different data sets and

compared with the SCS and KKZ methods. The results show that, for the data sets employed, the new methods lead to better solutions.

2. The new methods

In this section, two new methods for cluster initialisation are presented. The methods require the data space to be divided into a number of cells. The initialisation is done once and there is no need to try different initialisation sets. Further, there is no lower bound on the number of points per cluster for these methods to give good results. The idea of the methods is to initialise cluster centres according to the data distribution at a "macro" level and to rely on clustering to yield an improved solution at the detailed level.

2.1. Method-1

This method distributes cluster centres straightforwardly, according to the density of the data points. The algorithm is as follows.

Assume that the space is divided into equally sized cells;

let N be the number of data points and C the number of desired clusters.

let N_{C_i} be the number of data points in cell, i.

let M be the number of cells.

1. For each cell compute the integer part of $K_{C_i} = N_{C_i} * C/N$;

2. Select randomly K_{C_i} data points in cell i as initial cluster centres for data points in cell i.

3. If $(\sum_{i=1}^{M} K_{C_i} = C)$ STOP.

4. If $(\sum_{i=1}^{M} K_{C_i} < C)$ let $DEF = C - \sum_{i=1}^{M} K_{C_i}$; if DEF is large, say $> 10\%$ of C, let $M = M - (\sqrt{M} - 1)^2$; go to step 1; else select randomly DEF data points; STOP.

2.2. Method-2

The second method recognises that, if the objective is to minimise the sum of squares of distances from data points to cluster centroids, then the optimal distribution of cluster centroids will balance a few large distances in sparse regions with many small distances in dense regions.

We first suppose that the total volume, V containing the data points is divided into two volumes of size v and $(V - v)$.

Suppose p data points are randomly distributed throughout volume v, and $(N - p)$ data points are randomly distributed throughout volume $(V - v)$, where N is the number of data points in volume V.

We wish to place n cluster centres in volume v, $(C - n)$ in area $(V - v)$, such that

$$S = \sum (d^2)$$

is minimised; where d is the distance from each data point to the nearest cluster centre, and the sum is over all the data points.

Now, it can be shown that, in a k-dimensional space, if the data points are randomly distributed, and if the cluster centres are placed on a regular grid then, ignoring boundary effects, the expected square of the distance between any data point and its nearest cluster centre within volume v is proportional to $(v/n)^{2/k}$.

Similarly, the expectation for the square of the distance between data point and nearest cluster centre in volume $(V - v)$ is $[(V - v)/(C - n)]^{2/k}$.

So our best estimate for S is

$$S = P * \left(p * \left(\frac{v}{n} \right)^{2/k} + (N - p) * \left[\frac{(V - v)}{(C - n)} \right]^{2/k} \right),$$

where P is some proportionality constant.

Taking the first derivative of S with respect to n shows that S will be minimised when

$$\frac{n^{(2+k)}}{(p^k * v^2)} = \frac{(C - n)^{(2+k)}}{(N - p)^k * (V - v)^2}.$$

This is easily solved numerically for n which should be rounded to the nearest integer.

For the two-dimensional case it reduces to the quadratic equation:

$$-p * \frac{v}{n^2} + (N - p) * \frac{(V - v)}{(C - n)^2} = 0,$$

which can be solved in the usual way.

This second method can be modified by repeatedly subdividing volumes v and $(V - v)$ etc. into

Table 1
Final sum of square-distances for the generated data

Clusters	SCS	KKZ	Method-1	Method-2
600	1.818	0.822	0.809	0.777
700	1.535	0.700	0.669	0.627
800	1.125	0.605	0.570	0.510
900	1.003	0.434	0.437	0.326
1000	0.786	0.308	0.415	0.237

two, until some lower bound on the number of clusters in each subarea is reached. The method would be expected to work well if some intelligence were used in selecting how the division were made (i.e. corresponding to sparse and dense regions); note that v and $V - v$ need not be of equal size. However for our experiments, in order to present a fully-automated method, each binary division results in two equally-sized volumes. The experiments have also been restricted to the two-dimensional case.

3. Experimental results

As discussed in (Babu and Murty, 1993; Venkateswarlu and Raju, 1992), there is no general proof of convergence for the k-means clustering method. However, there exist some techniques for measuring clustering performance. One of these

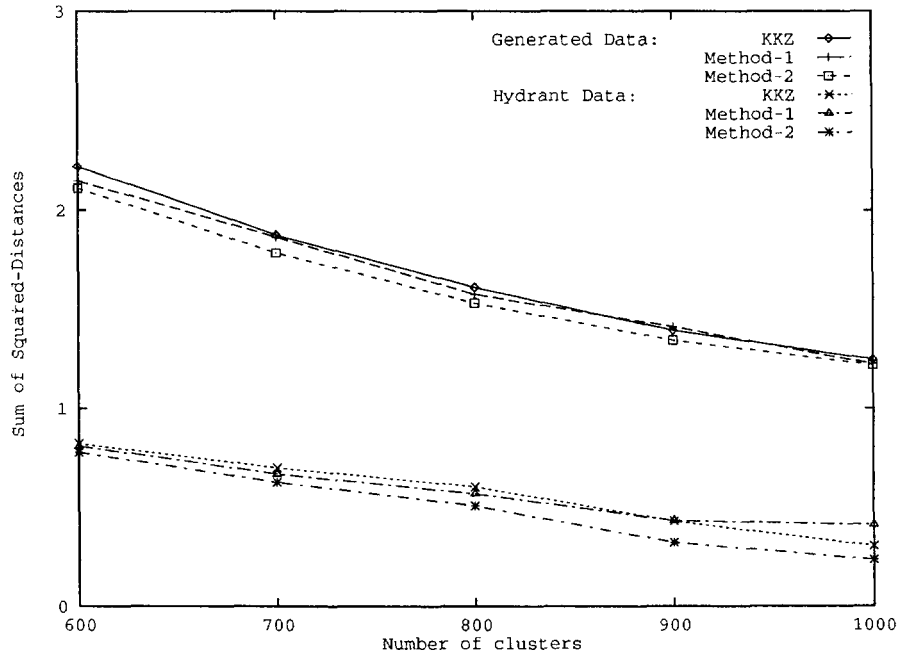


Fig. 1. Final sum of square-distances for the generated and hydrant data.

Table 2
Initial and final sum of square-distances for the hydrant data

Clusters	SCS		KKZ		Method-1		Method-2	
	Init	Fin	Init	Fin	Init	Fin	Init	Fin
600	1890.2	21.296	3.592	2.222	4.231	2.149	3.878	2.111
700	1884.4	20.265	3.023	1.877	3.714	1.866	3.258	1.786
800	1876.9	18.231	2.539	1.611	3.201	1.576	2.743	1.532
900	1789.4	16.134	2.164	1.395	3.101	1.415	2.493	1.345
1000	1663.1	14.976	1.907	1.249	2.331	1.226	2.212	1.221

techniques is the use of the sum of square-distances between data points and their cluster centre. That is

$$E = \sum_{i=1}^{N} d_{i,c(i)}^2,$$

where N is the number of data points and $d_{i,c(i)}$ is the distance from data point i, to the cluster centre to which it is assigned. This technique has been suggested in (Babu and Murty, 1993; Gersho and Gray, 1992; Wan et al., 1988)

The technique allows two solutions to be compared for a given data set, the smaller the value of E, the better the solution.

The new methods have been applied to two sets of data points to compute different sets of clusters. The first data set (which contained 10,000 data points) was artificially generated to simulate nine areas of dense population against a sparsely populated background, while in the second set (which contained 50,000 data points) the data points correspond to positions of water hydrants in an area including the city of Leeds. In Tables 1 and 2 are presented the final results (after applying the k-means algorithm) for all methods studied using the same stopping criteria and the best solution from different threshold values for the SCS method. Table 1 shows that for the generated data method-2 performed the best and the SCS performed the worst, while method-1 outperformed the KKZ in most cases. Table 2 shows similar results for the hydrant data where the new methods outperformed the other two, although the KKZ method obtained better starting initial values. Also in Table 2 is shown the sum of square-distances resulting from the initialisation. Fig. 1 shows the performance of the new methods.

Although our experiments did not show a significant change on the results by using different numbers of cells, however, the best results were obtained when the number of cells, M is approximately equal to $C/2\sqrt{C}$, where C is the number of desired clusters.

4. Summary

Two methods for cluster initialisation have been presented in this letter. Both methods gave better results than the SCS and ZZK methods and method-2 gave the best results. The SCS method gave the worst results, particularly for larger numbers of data points.

Although the KKZ gave the best initial results, the new methods gave better results after applying the k-means algorithm. This result is not unexpected as our new initialisation methods do not attempt to optimise at the detailed level.

Acknowledgements

The authors would like to thank Yorkshire Water for making available the water hydrant data used for this study.

References

Babu, G. and M. Murty (1993). A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm. *Pattern Recognition Lett.* 14, 763–769.

Buzo, A., A. Gray, R. Gray and J. Markel (1980). Speech coding based upon vector quantization. *IEEE Trans. Acoust. Speech Signal Process.* 28, 562–574.

Chatfield, C. and A. Collins (1980). *Introduction to Multivariate Analysis.* Chapman and Hall, London.

Gersho, A. and R. Gray (1992). *Vector Quantization and Signal Compression.* CAP.

Huang, C. and R. Harris (1993). A comparison of several vector quantization codebook generation approaches. *IEEE Trans. Image Process* 2 (1), 108–112.

Ismail, M. and M. Kamel (1989). Multidimensional data clustering utilization hybrid search strategies. *Pattern Recognition* 22 (1), 75–89.

Kaufman, L. and P. Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, New York.

Katsavounidis, I., C. Kuo and Z. Zhang (1994). A new initialization technique for generalized Lioyd iteration. *IEEE Signal Process Lett.* 1 (10), 144–146.

Linde, Y., A. Buzo and R. Gray (1980). An algorithm for vector quantizer design. *IEEE Trans. Comm.* 28 (1), 84–95.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symp. Math. Stat. and Prob.*, 281–297.

Nasrabadi, N. and R. King (1988). Image coding using vector quantization: a review. *IEEE Trans. Comm.* 36 (8), 957–970.

Tou, J. and R. Gonzales (1974). *Pattern Recognition Principles.* Addison-Wesley, Reading, MA.

Venkateswarlu, N. and P. Raju (1992). Fast isodata clustering algorithms. *Pattern Recognition* 25 (3), 335–342.

Wan, S., S. Wong and P. Prusinkiewicz (1988). An algorithm for multidimensional data clustering. *ACM Trans. Math. Software* 14 (2), 153–162.