An Incremental K-means algorithm

D T Pham, S S Dimov and C D Nguyen

Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 2004 218: 783
DOI: 10.1243/0954406041319509

An Incremental *K*-means algorithm

D T Pham*, **S S Dimov** and **C D Nguyen**
Manufacturing Engineering Centre, Cardiff University, Cardiff, Wales, UK

Abstract: Data clustering is an important data exploration technique with many applications in engineering, including parts family formation in group technology and segmentation in image processing. One of the most popular data clustering methods is *K*-means clustering because of its simplicity and computational efficiency. The main problem with this clustering method is its tendency to converge at a local minimum. In this paper, the cause of this problem is explained and an existing solution involving a cluster centre jumping operation is examined. The jumping technique alleviates the problem with local minima by enabling cluster centres to move in such a radical way as to reduce the overall cluster distortion. However, the method is very sensitive to errors in estimating distortion. A clustering scheme that is also based on distortion reduction through cluster centre movement but is not so sensitive to inaccuracies in distortion estimation is proposed in this paper. The scheme, which is an incremental version of the *K*-means algorithm, involves adding cluster centres one by one as clusters are being formed. The paper presents test results to demonstrate the efficacy of the proposed algorithm.

Keywords: clustering, *K*-means method, incremental clustering

NOTATION

C	a cluster
d	length of one side of the calculated hyper-cube
$d(w, x)$	distance between the cluster's centre w and a position x in the Euclidean space
ΔD	estimated decrease of the total distortion error when the centre of a cluster is moved to a new position
I_z	distortion error of cluster z
ΔI	estimated increase of the total distortion error when the centre of a cluster is removed
ΔM	estimated change of the total distortion error when the jumping operation has occurred
n	number of objects of the dataset
N	number of objects belonging to the cluster (cluster's capacity)
N_d	dimension of the Euclidean space
N_i, N_j, N_k, N_z	number of objects belonging to clusters C_i, C_j, C_k, C_z respectively
R	number of disjoint regions
S	sum of the squared distances between the objects in the cluster and the centre of the Euclidean space
w	centre of a cluster
x_0	centre of the Euclidean space

The MS was received on 20 August 2003 and was accepted after revision for publication on 26 March 2004.
* *Corresponding author: Manufacturing Engineering Centre, Cardiff University, The Cardiff School of Engineering, PO Box 925, Newport Road, Cardiff CF24 0YF, Wales, UK.*

1 INTRODUCTION

Data clustering (DC) is an important data exploration technique for grouping similar physical or abstract objects. The technique allows objects with common characteristics to be lumped together in order to facilitate their further processing. DC is an unsupervised technique that generates hypotheses based on the provided unlabelled objects. This makes this method a very attractive data processing technique for a wide range of applications [**1**].

K-means clustering (vector quantization) is one of the most popular data clustering methods because of its simplicity and computational efficiency. The computational efforts required to form the clusters grow linearly with the increase of the dataset size. When applied to small or medium sized datasets, *K*-means clustering gives better results than other methods in terms of clustering performance and computational time [**2**].

There are a number of different implementations of the *K*-means method. For example, Linde–Buze–Gray

(LBG) is one version of this method in which a batch update mode is applied [3]. Other implementations of the method, ISODATA [4] and MAXNET [5], restrict the cluster diameters and introduce flexibility in specifying the number of clusters. Another version of the *K*-means method [6] employs a contiguity characteristic to improve the algorithm performance in some specific applications.

K-means clustering has been used as a clustering method in many application areas. For example, this method could be employed for:

(a) image segmentation and compression [5, 6],
(b) grouping image voxels [7],
(c) initial clustering before applying more sophisticated iterative methods [8],
(d) analysing a robot's trajectory [9],
(e) speech and handwriting feature vectors analysis [10, 11],
(f) grouping machined parts into families in cellular manufacturing system design [12, 13].

Although the *K*-means method has demonstrated a number of advantages over other DC techniques, it also has drawbacks. In particular, it often converges at a local optimum and, therefore, acceptable results can be found only after several iterations. The local optimum problem has been studied extensively by a number of researchers [3, 14–16].

In recent years, many improvements have been proposed and implemented in the *K*-means method. A number of researchers have proposed different techniques to improve its convergence speed [15, 17–21]. The effect of finite sample size on the *K*-means method was studied [22]. To obtain better results, other researchers [23–25] modified the initialization procedure by presenting the algorithm with data collected using a density-based approach. Again, to improve performance, Fritzke [3] suggested a new jumping operation to facilitate the algorithm's convergence and assist it in escaping from local minima. In the same direction as Fritzke's work, the utility index is used in reference [26]. Chinrungrueng and Sequin [27] proposed a new updating method introducing a restriction hypothesis about the problem's underlying object distribution. The stochastic relaxation scheme was applied to the *K*-means method to improve its performance [28].

In this paper, a new version of the *K*-means algorithm called Incremental *K*-means is proposed. In section 2, the original *K*-means algorithm is described. Section 3 explains why the original algorithm converges to a local minimum and suggests a way to avoid this. Incremental *K*-means is presented in section 4. Section 5 discusses two approaches to speed-up Incremental *K*-means. The effect of the proposed modifications on the performance of the algorithm is analysed in section 6. Conclusions are given in section 7.

2 THE ORIGINAL K-MEANS ALGORITHM

The *K*-means method is applicable only to datasets with numerical attributes. The Euclidean distance is employed to measure the distance between objects. The main steps in the original *K*-means algorithm are shown as follows:

Step 1. Choose arbitrary *K* objects for *K* cluster centres.
Step 2. Assign each object in the training set to the closest cluster and update the centres of the clusters.
Step 3. If the clustering criterion is satisfied (the cluster centres do not move), the algorithm stops. Otherwise, go to step 2.

For convenience, in this paper the information in a cluster is represented by a triple $\langle w, N, S\rangle$ where w is the centre of a cluster, N is the number of objects belonging to the cluster (cluster's capacity) and S is the sum of the squared distances between the objects in the cluster and the centre of the Euclidean space. The distortion error I of a cluster is calculated using the following equation:

$$I = S - N[d(w, x_0)]^2 \tag{1}$$

where $d(w, x_0)$ is the distance between the cluster's centre w and the centre of the Euclidean space x_0.

3 MOTIVATION

The performance of the *K*-means algorithm can be measured by considering the movements of the centres of the clusters. When a centre is initiated in an inappropriate position, it cannot move to an optimum location. For example, in Fig. 1a the dataset is split into two disjoint regions, R1 and R2, with the same uniform distribution. Suppose that the number of clusters is chosen to be 4. In this example, the hypothesis about the smooth underlying distribution [27] is not satisfied. Because of the random initialization, after step 1 of the *K*-means algorithm, the centres might be located as shown in Fig. 1b. There is not any object in region R2 which can belong to any cluster in region R1 due to the distance between the two regions. Thus, no cluster centre in region R1 can move to region R2. Therefore, the clustering obtained by *K*-means (Fig. 1c) differs from the optimal results for this dataset (Fig. 1d).

To overcome the problem of cluster centres being trapped in inappropriate locations, Fritzke [3] suggested a modified *K*-means algorithm incorporating a jumping operation to move the cluster centre with the least distortion error to the cluster with the most distortion error:

Step 1. Choose arbitrary *K* objects for *K* cluster centres.
Step 2. Assign each object in the training set to the closest cluster and update the centres of the clusters.

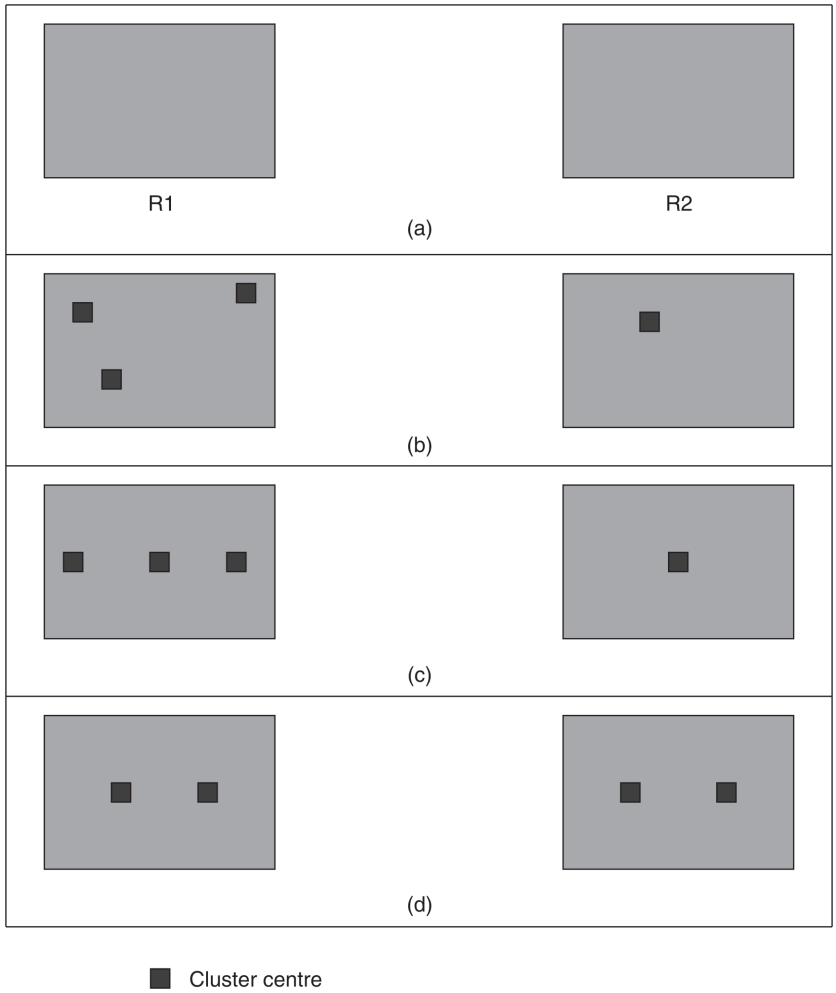

Fig. 1 The results of applying *K*-means ($K = 4$) on two split regions

Step 3. If the clustering criterion is satisfied (the cluster centres do not move), go to step 4.
Else, go to step 2.

Step 4. If there is a cluster that can be moved to a better position to reduce the total sum of the distortion errors, move it to the new position and then go to step 2.
Else, stop.

When the centre of a cluster is taken away from an inappropriate position, the sum of distortion errors of all clusters increases by a value equal to the sum of the squared distances between objects of the removed cluster and the second nearest cluster centre. However, this calculation does not take into account the fact that the centre of the second nearest cluster centre will be moved when the objects of the removed cluster are added to it. Thus, the increase of this sum will be smaller than it would otherwise be. Moreover, in the proposed operation, the removed cluster centre will be inserted at a random position into the cluster with the largest distortion. There is no estimation of the effect of this operation on the sum of distortion errors of all clusters.

Pelleg and Moore [**29**] proposed to start the algorithm with a small number of clusters, *K*, then double it by inserting new cluster centres in suitable positions. There are two problems with the criterion used to evaluate the performance of this operation. Firstly, each cluster is divided independently into two without taking into account the influence of neighbouring clusters. Secondly, the BIC scoring that Pelleg and Moore adopt does not guarantee that the distortion errors of all clusters will be minimized.

In this paper, a new criterion is proposed to assess the performance of the jumping operation suggested by Fritzke [**3**]. During the learning process, as already mentioned, the operation deals with the local minimum problem by removing a cluster from an inappropriate position and inserting it into a more promising position. The increase in the sum of distortion errors of all clusters when one cluster centre is removed and the decrease in the same sum when a new cluster centre is inserted into a new position are two parameters used to evaluate performance. Because it is infeasible to calculate the values of these parameters precisely in the general case, two procedures are described in the following section to estimate them.

Evaluation of distortion of the clusters

Suppose that the centre of cluster C_i is taken out. In the worst case, all objects belonging to C_i will be allocated to the second nearest cluster C_j without affecting any other neighbouring clusters. The triples (w_i, N_i, S_i) and (w_j, N_j, S_j) characterize C_i and C_j. The triple (w_k, N_k, S_k) of the new cluster C_k is calculated from the following equations:

$$N_k = N_i + N_j \tag{2}$$

$$w_k = \frac{1}{N_k}\sum_{i=1}^{N_k} x_i^k = \frac{1}{N_k}(N_i w_i + N_j w_j) \tag{3}$$

$$S_k = S_i + S_j \tag{4}$$

and the increase of the distortion ΔI in the worst case is calculated using

$$\begin{aligned}\Delta I &= I_k - I_i - I_j \\ &= S_k - N_k[d(w_k, x_0)]^2 - \left\{S_i - N_i[d(w_i, x_0)]^2\right\} \\ &\quad - \left\{S_j - N_j[d(w_j, x_0)]^2\right\} \\ &= N_i[d(w_i, x_0)]^2 + N_j[d(w_j, x_0)]^2 - N_k[d(w_k, x_0)]^2 \\ &= \frac{N_i N_j}{N_i + N_j}[d(w_i, w_j)]^2\end{aligned} \tag{5}$$

where I_k, I_i and I_j are the distortion of C_k, C_i and C_j respectively.

When the centre of a cluster is moved to a new position, it will cause a decrease in the sum of cluster distortion errors. This decrease cannot be calculated in the general case. In this paper, it is assumed that a cluster C_z is a hyper-cube with a uniform distribution density p of objects belonging to it (Fig. 2). When a new cluster centre is inserted, C_z will be split into two clusters C_{z1} and C_{z2}. The triples, (w_z, N_z, S_z), (w_{z1}, N_{z1}, S_{z1}) and (w_{z2}, N_{z2}, S_{z2}), represent these clusters. All objects of C_z are assumed to belong to C_{z1} or C_{z2}. After training, the centres of the two new clusters will be positioned as shown in Fig. 2. Without loss of generality, the centre of C_z is considered to be the origin of the coordinate system.

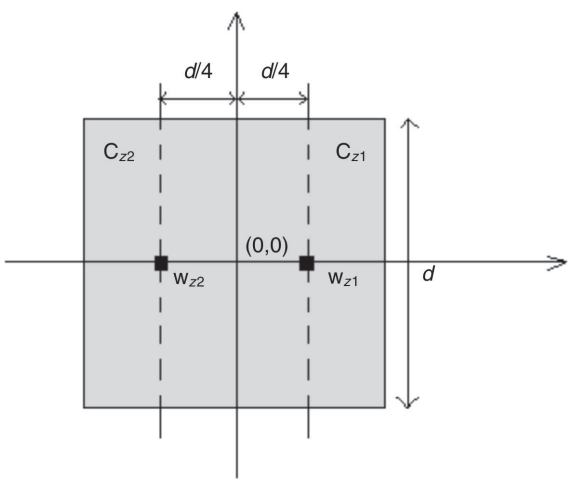

Fig. 2 The splitting of C_z into C_{z1} and C_{z2} after training

The distortion error I_z of cluster C_z is calculated as follows:

$$\begin{aligned}I_z &= \int_{-d/2}^{d/2}\int_{-d/2}^{d/2}\cdots\int_{-d/2}^{d/2}[d(x, x_0)]^2 p\ \mathrm{d}x^{(1)}\mathrm{d}x^{(2)}\cdots\mathrm{d}x^{(N_\mathrm{d})} \\ &= \int_{-d/2}^{d/2}\int_{-d/2}^{d/2}\cdots\int_{-d/2}^{d/2}\left[\sum_{t=1}^{N_\mathrm{d}}(x^{(t)})^2\right] p\ \mathrm{d}x^{(1)}\mathrm{d}x^{(2)}\cdots\mathrm{d}x^{(N_\mathrm{d})} \\ &= p\sum_{t=1}^{N_\mathrm{d}}\left\{\int_{-d/2}^{d/2}\int_{-d/2}^{d/2}\cdots\int_{-d/2}^{d/2}[(x^{(t)})^2]\mathrm{d}x^{(1)}\mathrm{d}x^{(2)}\cdots\mathrm{d}x^{(N_\mathrm{d})}\right\} \\ &= p\sum_{t=1}^{N_\mathrm{d}}\left\{\left(\prod_{\substack{j=1\\ j\neq t}}^{N_\mathrm{d}}\left|x^{(j)}\right|_{-d/2}^{d/2}\right)\left[\left|\frac{(x^{(t)})^3}{3}\right|_{-d/2}^{d/2}\right]\right\} \\ &= \frac{pN_\mathrm{d}d^{N_\mathrm{d}+2}}{12} \\ &= \frac{N_z N_\mathrm{d}\ d^2}{12}\end{aligned} \tag{6}$$

where N_d is the dimension of the Euclidean space. Because C_z is a cube with a uniform distribution, the two new clusters C_{z1} and C_{z2} contain the same number of objects $N_{z1} = N_{z2} = N_z/2$. Using equation (6), the decrease in distortion errors is calculated as follows:

$$\begin{aligned}\Delta D &= \frac{N_{z1}N_{z2}}{N_z}[d(w_{z1}, w_{z2})]^2 \\ &= \frac{N_z}{4}\left(\frac{d}{2}\right)^2 \\ &= \frac{3I_z}{4N_\mathrm{d}}\end{aligned} \tag{7}$$

By applying the jumping operation, the sum of the distortion errors will be changed by a value $\Delta M = \Delta I - \Delta D$. If ΔM is smaller than 0, the operation could lead to better clustering.

As the performance of the jumping operation is evaluated based on two estimated parameters in the cluster centre removal and insertion operators, this may introduce additional errors. An incremental strategy can

be used to eliminate the removal of a cluster centre and the dependence on the initial positions of cluster centres. An incremental algorithm starts with the number of clusters K being set equal to 1 and increasing by 1 in each step. With each increase of K, a new cluster centre is inserted into the cluster with the most distortion and then objects are reassigned to clusters until the centres do not move. The process is repeated until K reaches the specified number of clusters. A new improved *K*-means algorithm with this incremental strategy will be described in the next section. The proposed algorithm has the advantage of determining near-optimal cluster centre positions.

To the authors' knowledge, there is another *K*-means clustering algorithm [**30**] with a similar incremental strategy. In each step of the incremental process, that algorithm uses a local search procedure to calculate the position of the new cluster centre, assuming that the positions of the current cluster centres are optimal and can remain fixed. Because of the dynamic nature of clusters in a *K*-means operation, this calculation will not yield the optimal position for the new cluster centre for each step. The position error accumulated over the clustering process can affect the final performance of the algorithm.

4 INCREMENTAL *K*-MEANS ALGORITHM

The Incremental K-means algorithm is summarized as follows:

Assign $K = 1$.

Phase 1. *Normal training*

Step 1. If $K = 1$, choose an arbitrary point for a cluster centre.
If $K > 1$, insert the centre of the new cluster in the cluster with the greatest distortion.

Step 2. Assign each object in the training set to the closest cluster and update its centre.

Step 3. If the cluster centre does not move, go to phase 2.
Else, go to phase 1, step 2.

Phase 2. *Increasing the number of clusters*

If K is smaller than a specified value, increase K by 1 and go to phase 1, step 1.
Else, stop.

Phase 1 includes steps that are similar to the steps of the conventional *K*-means algorithm, except in its restriction on where the new cluster centre can be placed. The centres of all existing clusters do not change their positions, which makes the algorithm less dependent on the random placement of the new centres.

The complexity of the new algorithm can be assessed using the formula

$$O(K^2 * n * \mathit{num\_of\_iterations})$$

where n is the number of objects and *num_of_iterations* is the largest possible number of iterations in phase 1. Compared with the complexity $O(K * n * \mathit{num\_of\_iterations})$ of the *K*-means algorithm, the Incremental *K*-means algorithm requires K times more iterations.

When there are K clusters, the new algorithm needs to run phase 1 K times, each iteration being equivalent to one execution of the traditional *K*-means algorithm. Of those K times, $(K - 1)$ are considered intermediate steps that prepare the data for the next iteration. Therefore, only the last iteration of phase 1 has to satisfy the strict end condition defined in step 3. In this paper, the end condition for each intermediate iteration is relaxed and tested separately.

In the initial step of each run of phase 1 of the Incremental *K*-means algorithm, a new cluster centre is inserted in the cluster with the largest distortion error. The insertion of the new centre mostly affects the objects belonging to this cluster. The performance of the algorithm can be improved further by organizing the indexing of centres to reduce the computational effort in finding the nearest cluster [**19**].

5 PERFORMANCE

Six artificial datasets and six real datasets from the UCI Repository [**31**] were used to test the proposed new algorithm. The characteristics of these datasets are represented in Table 1. The object distribution of the six artificial datasets is shown in Fig. 3.

The research carried out by Bottou and Bengio [**15**] and Bilmes *et al*. [**2**] showed that it takes on average 15 iterations for the *K*-means algorithm to reach a local minimum. The clustering process could be stopped by specifying termination conditions such as a predefined number of iterations and the percentage reduction of the distortion errors in one iteration being smaller than a given value ε. In this work, these two termination criteria were used. In particular, the maximum number of iterations was empirically set to 20 and ε to 10^{-7}. The algorithm stops when one of these conditions is satisfied.

Due to the random nature of the *K*-means algorithm, it is important to conduct a large number of tests to demonstrate its performance in a statistically significant way. When the problem has R disjoint and distant regions and K clusters should be formed, an extremely large number of possibilities exist to allocate the K

Table 1 Characteristics of data sets

(a) Real data sets	Balance-Scale	Ionosphere	Iris	Pima	Wine	Zoo
Number of attributes	4	34	4	8	13	17
Number of objects	635	351	150	768	178	101

(b) Artificial data sets	Uniform1	Uniform2	Uniform3	Gauss1	Gauss2	Gauss3
Number of attributes	2	2	2	2	2	2
Number of objects	421	1084	800	848	1220	800

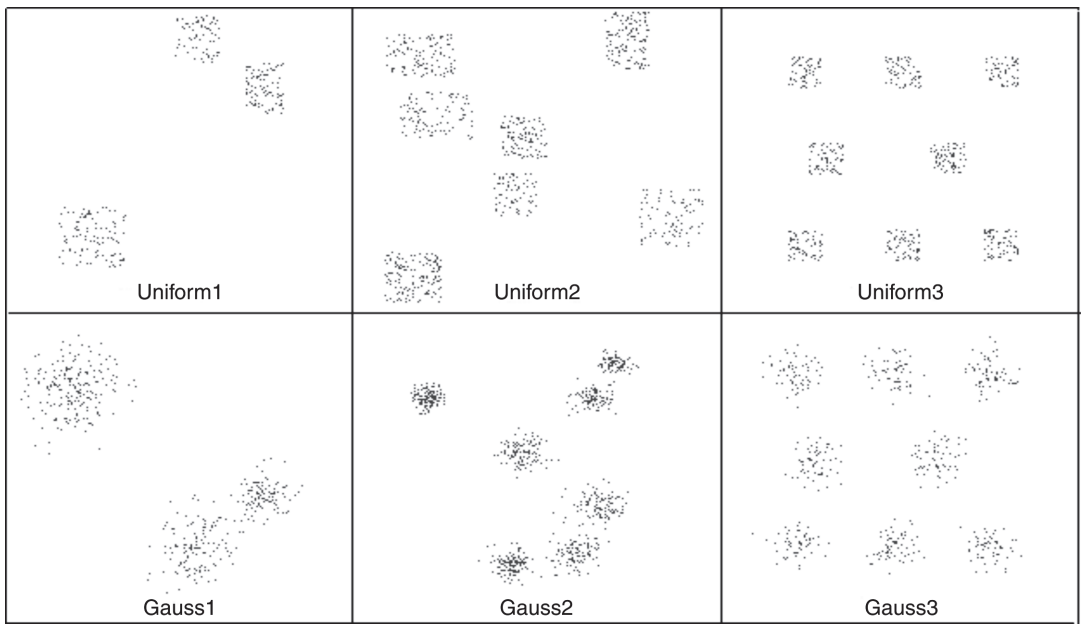

Fig. 3 The object distribution of the contrived datasets

cluster centres to different regions. Each particular allocation will lead to different distortion errors. Unfortunately, R is not known in real problems. Many researchers select a large K and a small number of tests, which may not lead to optimal clustering results. In this work, K was selected in the range of 1–15 and the number of tests for each dataset was taken as 500.

Figure 4 shows the results obtained by applying four different versions of the K-means algorithm (original K-means, K-means with the jumping operation, Incremental K-means and Incremental K-means with predefined termination conditions) to the 12 datasets. On all datasets, except the Balance-Scale dataset, the K-means algorithm with the jumping operation outperforms the original K-means algorithm in spite of

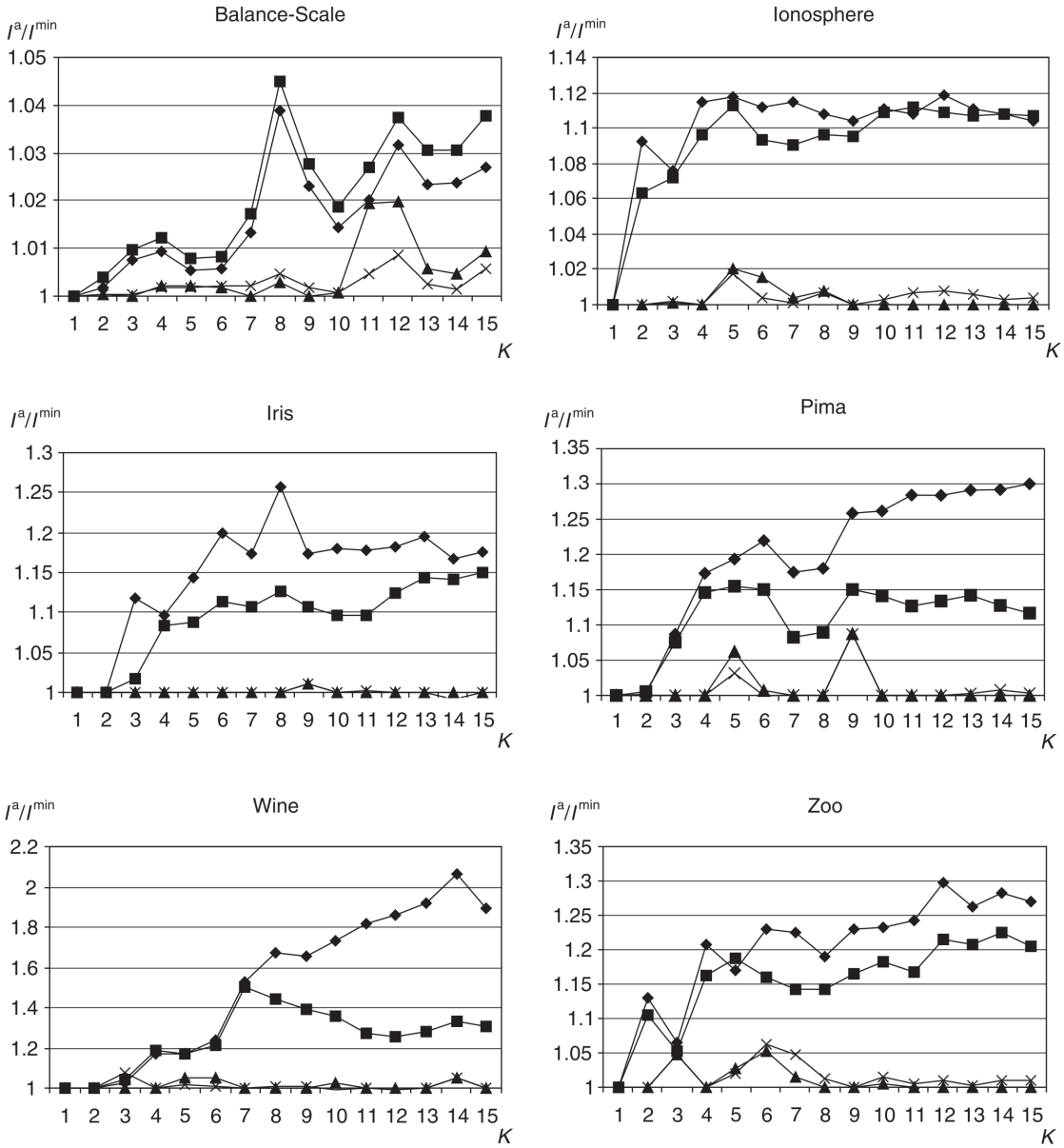

Fig. 4 (continued over)

the fact that the results obtained are far from the optimal solution. Also, on all datasets, the Incremental *K*-means algorithm groups objects in clusters whose average distortion error is very close to the smallest distortion error of any of those clusters. This means that the Incremental *K*-means algorithm does not depend on the specific characteristics of the datasets and the value of *K*, and produces reliable and optimal clustering of objects.

Figure 5 gives the running time of the *K*-means algorithm, Incremental *K*-means algorithm and Incremental *K*-means algorithm with predefined termination conditions. All algorithms were implemented in C++ and executed on a Pentium II 300 MHz PC. Although the theoretical complexity of Incremental *K*-means is a function of K^2, the experiments carried out show that the running time depends linearly on *K* (see the Appendix). By specifying termination conditions, the running time is reduced without sacrificing the quality of the clustering results (Fig. 4).

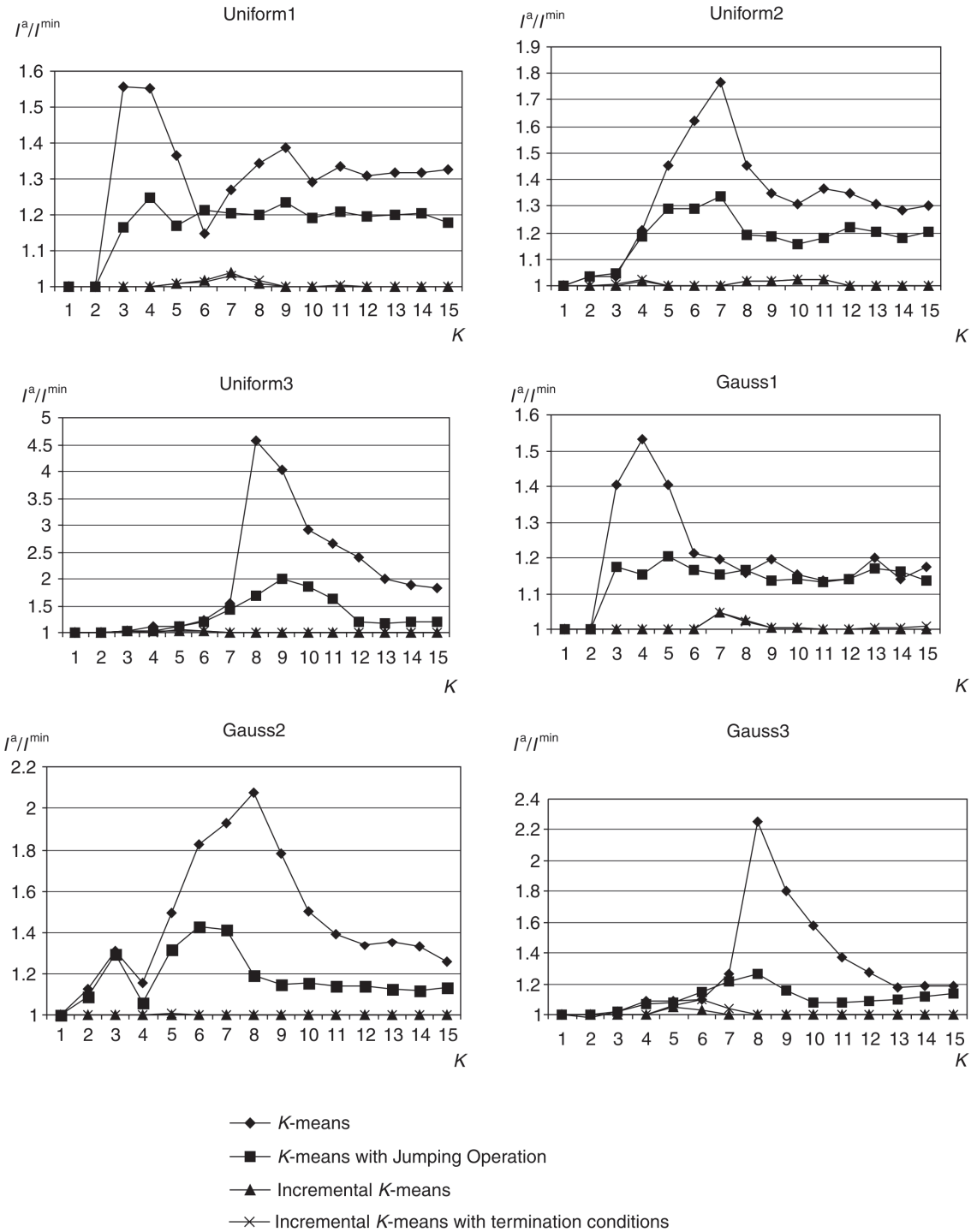

Fig. 4 Clustering results of *K*-means, *K*-means with the jumping operation, Incremental *K*-means and Incremental *K*-means with termination conditions (I^a and I^{min} are the average and the minimum values of cluster distortion errors)

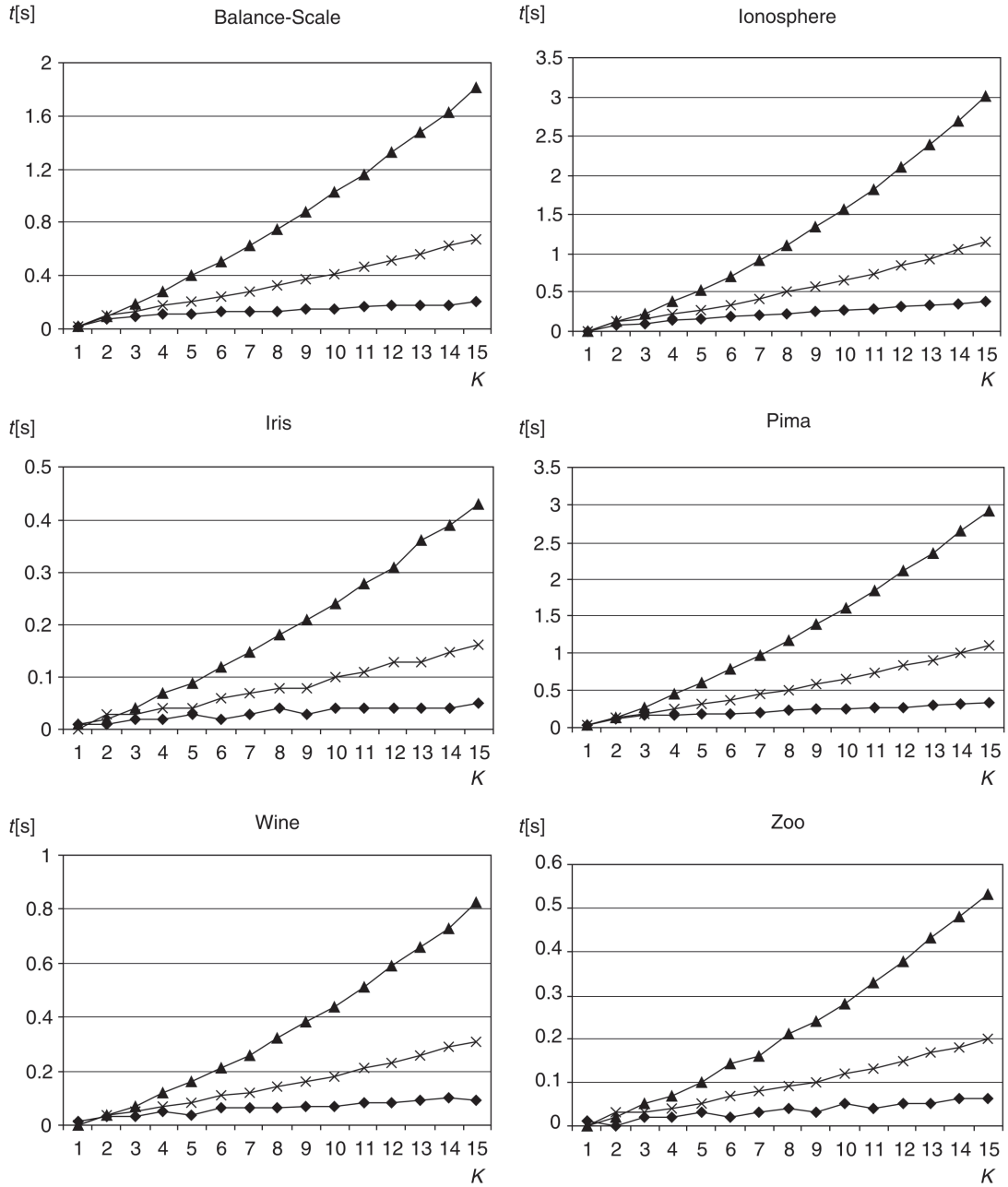

Fig. 5 (continued over)

6 FURTHER IMPROVEMENT

With some values of K, the results of the Incremental K-means algorithm on real datasets are not close to the optimal, e.g. the Balance-Scale dataset with $K = 11$ or 12 and the Ionosphere dataset with $K = 5$ or 6. The reason for this problem is the heuristical insertion of a new centre into the cluster, with the largest distortion when increasing K by 1. After insertion, the total distortion is decreased by a value smaller than or equal to the distortion of the split cluster. If a different cluster has its distortion larger than this amount of decrease, it may be a better choice for splitting up. Thus, the algorithm has to investigate all possibilities in order to find the most beneficial place to insert the new cluster. However, this searching slows down the algorithm for large values of

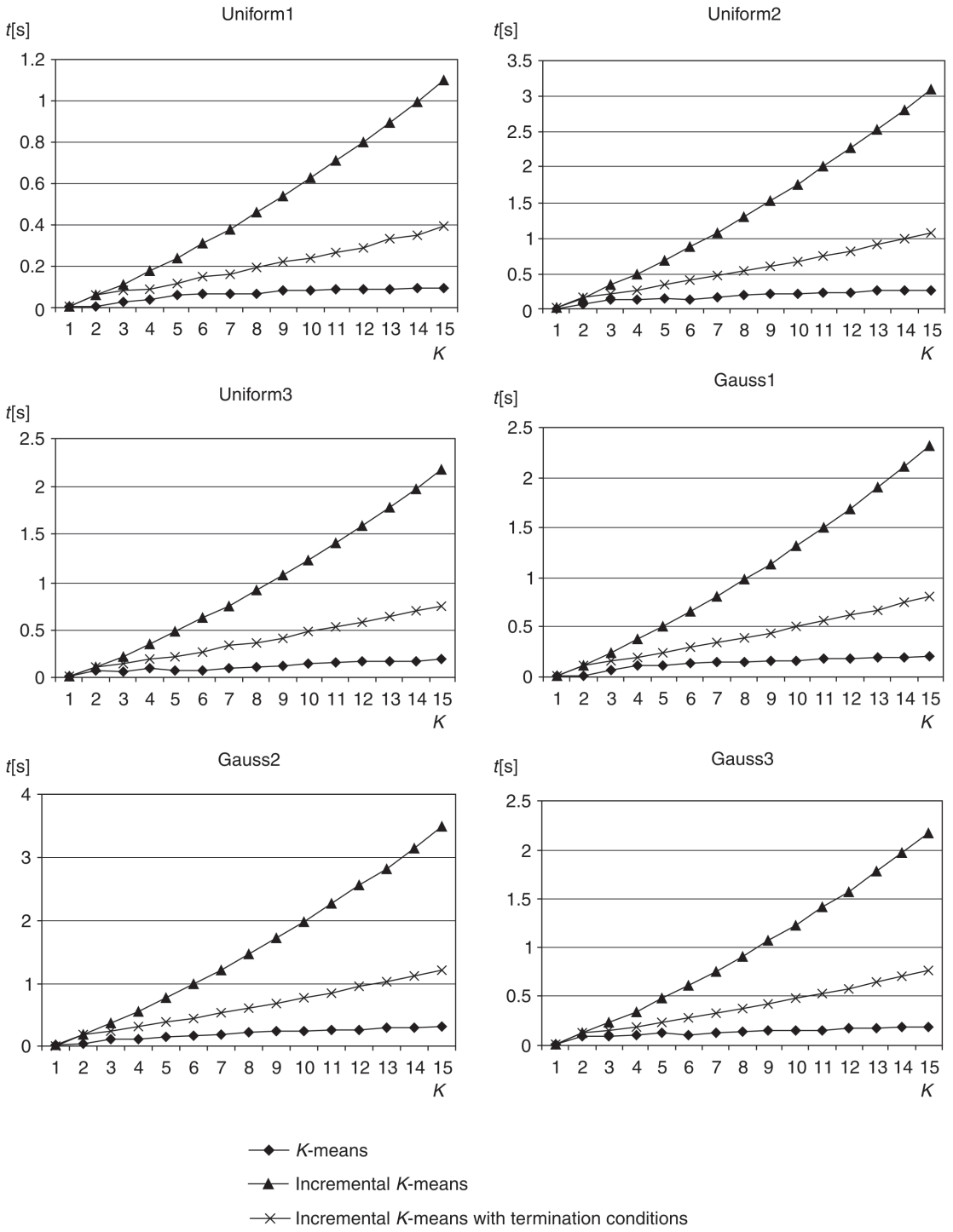

Fig. 5 Comparison of the running time of *K*-means, Incremental *K*-means and Incremental *K*-means with termination conditions

K, so that, for any given insertion, the search only tries clusters with a distortion at least 1.5 times larger than the amount of decrease achieved until that point.

Figure 6 shows the results obtained by applying three different versions of the *K*-means algorithm, original *K*-means, Incremental *K*-means and Incremental *K*-means with clusters search to the six real datasets. The third version has all its results close to 1 for all values of *K*, demonstrating that the search strategy helped it to handle cases that the plain incremental version had difficulties with.

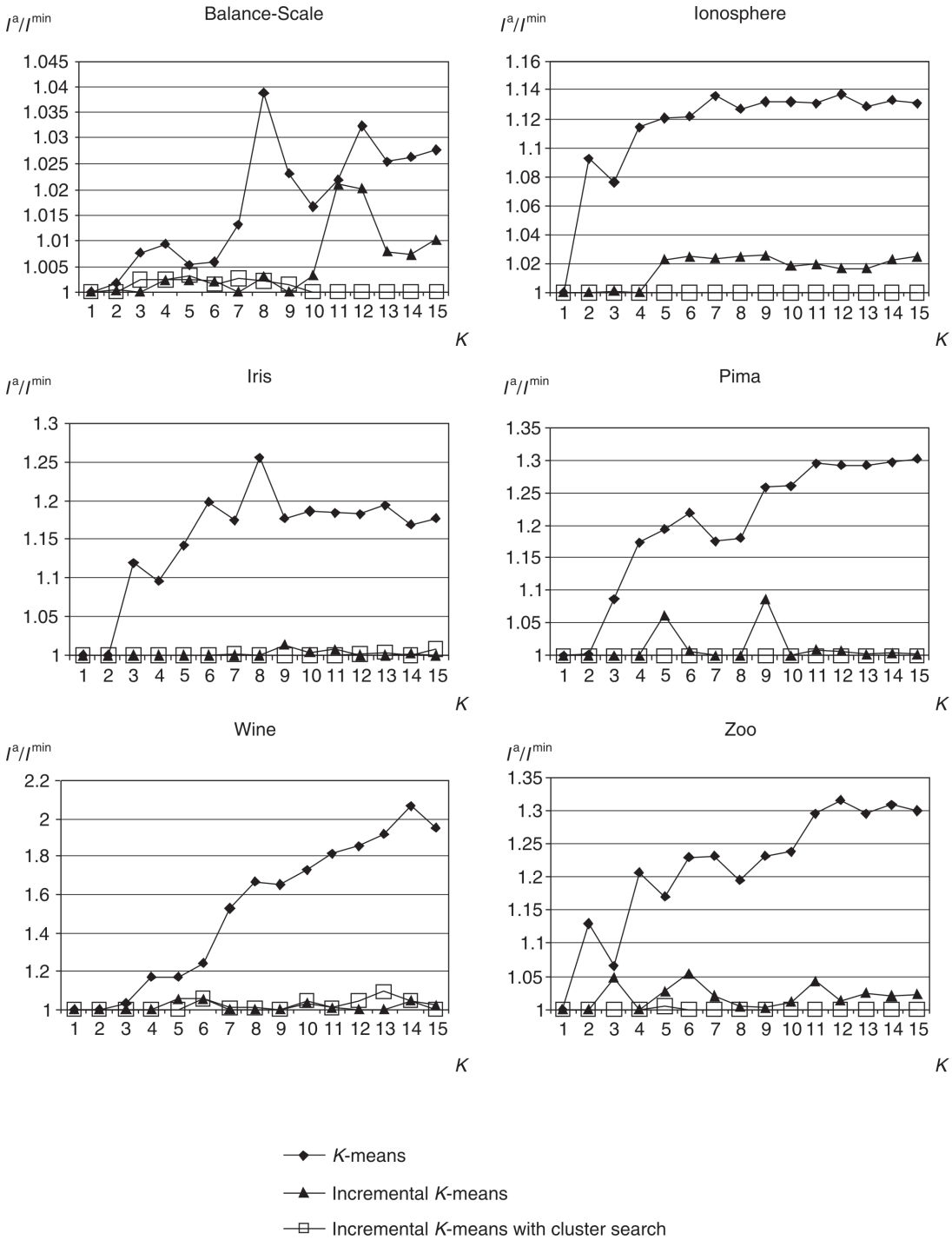

Fig. 6 Clustering results of *K*-means, Incremental *K*-means and Incremental *K*-means with cluster search (I^a and I^{min} are the average and the minimum values of cluster distortion errors)

7 CONCLUSION

This paper has described a new clustering algorithm, Incremental *K*-means. The algorithm has been tested on a number of artificial and real datasets. The algorithm consistently outperforms the original *K*-means algorithm. The proposed search strategy decreases the dependence of the algorithm on the initialization of cluster centres. In addition, the new algorithm only needs to be applied once to achieve almost optimal results. Further experiments will be carried out to test the new algorithm on both nominal and mixed data.

REFERENCES

1 **Romesburg, H. C.** *Cluster Analysis for Researchers*, 1990 (Krieger Publishing, Malabar, Florida).
2 **Bilmes, J., Vahdat, A., Hsu, W.** and **Im, E. J.** Empirical observations of probabilistic heuristics for the clustering problem. Technical Report TR-97-018, International Computer Science Institute, Berkeley, California, 1997.
3 **Fritzke, B.** The LBG-U method for vector quantization—an improvement over LBG inspired from neural networks. Neural Processing Letters 5, No. 1, 1997, pp. 35–45.
4 **Kaufman, L.** and **Rousseeuw, P. J.** *Finding Groups in Data: An Introduction to Cluster Analysis*, 1990 (John Wiley, New York).
5 **Pao, Y.-H.** *Adaptive Pattern Recognition and Neural Networks*, 1989 (Addison-Wesley, New York).
6 **Theiler, J.** and **Gisler, G.** A contiguity-enhanced *K*-means clustering algorithm for unsupervised multispectral image segmentation. *Proc. SPIE*, 1997, **3159**, 108–118.
7 **Gee, C. J., Fabella, A. B., Fernandes, I. B., Turetsky, I. B., Gur, C. R.** and **Gur, E.** New experimental results in atlas-based brain morphometry. In Proceedings of SPIE Medical Imaging 1999: *Image Processing* (Ed. K. M. Hanson), Bellingham, Washington, 1999, pp. 604–611.
8 **Hansen, L. K.** and **Larsen, J.** Unsupervised learning and generalisation. In Proceedings of the IEEE International Conference on *Neural Network*, Washington DC, June 1996, pp. 25–30.
9 **McGovern, A.** An algorithm for automatically learning macro-actions. In NIPS'98 Workshop on *Abstraction and Hierarchy in Reinforcement Learning*, Denver, Colorado, 1998.
10 **Cook, G. D.** and **Robinson, A. J.** Utterance clustering for large vocabulary continuous speech recognition. In Proceedings of the European Conference on *Speech Technology*, Madrid, Spain, Vol. I, 1995, pp. 219–222.
11 **Kosmala, A., Rottland, J.** and **Rigoll, G.** Improved online handwriting recognition using context dependent hidden Markov models. In Proceedings of the International Conference on *Document Analysis and Recognition*, Ulm, Germany, 1997, pp. 641–644.
12 **Josien, K.** and **Liao, T. W.** Simultaneous grouping of parts and machines with an integrated fuzzy clustering method. *Fuzzy Sets and Systems*, 2002, **126**(1), 1–21.
13 **Lozano, S., Dobado, D., Larraneta, J.** and **Onieva, L.** Modified fuzzy C-means algorithm for cellular manufacturing. *Fuzzy Sets and Systems*, 2002, **126**(1), 23–32.
14 **MacQueen, J. B.** Some methods for classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on *Mathematical Statistics and Probability*, Vol. I, *Statistics*, Berkeley, California, 1967, pp. 281–297 (University of California Press, Los Angeles, California).
15 **Bottou, L.** and **Bengio, Y.** Convergence properties of the *K*-means algorithm. In *Advances in Neural Information Processing Systems*, Vol. 7, 1995 (MIT Press, Cambridge, Massachusetts).
16 **Pena, J. M., Lazano, J. A.** and **Larranaga, P.** An empirical comparison of four initialisation methods for the *K*-means algorithm. *Pattern Recognition Lett.*, 1999, **20**, 1027–1040.
17 **Al-Daoud, M. B., Venkateswarlu, N. B.** and **Roberts, S. A.** Fast *K*-means clustering algorithms. Report 95.18, School of Computer Studies, University of Leeds, June 1995.
18 **Alsabti, K., Ranka, S.** and **Singh, V.** An efficient *K*-means clustering algorithm. In Proceedings of the First Workshop on *High-Performance Data Mining*, Orlando, Florida, 1998; ftp://ftp.cise.ufl.edu/pub/faculty/ranka/Proceedings.
19 **Pelleg, D.** and **Moore, A.** Accelerating exact K-means algorithms with geometric reasoning. In Proceedings of the Conference on *Knowledge Discovery in Databases 1999* (*KDD99*), San Diego, California, 1999, pp. 277–281.
20 **Castro, V. E.** and **Yang, J.** A fast and robust general purpose clustering algorithm. In Fourth European Workshop on *Principles of Knowledge Discovery in Databases and Data Mining 2000* (*PKDD00*), Lyon, France, 2000.
21 **Kanungo, T., Mount, D. M., Netanyahu, N., Piatko, C., Silverman, R.** and **Wu, A.** The efficient *K*-means clustering algorithm: analysis and implementation. *IEEE Trans. Pattern Analysis and Mach. Intell.*, 2002, **24**(7), 881–892.
22 **Bermejo, S.** and **Cabestany, J.** The effect of finite sample size on on-line *K*-means. *Neurocomputing*, 2002, **48**, 511–539.
23 **Al-Daoud, M. B., Venkateswarlu, N. B.** and **Roberts, S. A.** New methods for the initialisation of clusters. *Pattern Recognition Lett.*, 1996, **17**, 451–455.
24 **Epter, S., Krishnamoorthy, M.** and **Zaki, M.** Clusterability detection and initial seed selection in large data sets. Technical Report 99-6, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York, 1999.
25 **Bradley, S.** and **Fayyad, U. M.** Refining initial points for *K*-means clustering. In Proceedings of the Fifteenth International Conference on *Machine Learning* (*ICML '98*) (Ed. J. Shavlik), Madison, Wisconsin, 1998, pp. 91–99 (Morgan Kaufmann, San Francisco, California).
26 **Patane, G.** and **Russo, M.** The enhanced LBG algorithm. *Neural Networks*, 2001, **14**, 1219–1237.
27 **Chinrungrueng, C.** and **Sequin, C. H.** Optimal adaptive *K*-means algorithm with dynamic adjustment of learning rate. *IEEE Trans. Neural Networks*, January 1995, **6**(1), 157–169.
28 **Kovesi, B., Boucher, J.-M.** and **Saoudi, S.** Stochastic *K*-means algorithm for vector quantisation. *Pattern Recognition Lett.*, 2001, **22**, 603–610.
29 **Pelleg, D.** and **Moore, A.** X-means: extending *K*-means with efficient estimation of the number of clusters. In Proceedings of the Seventeenth International Conference on *Machine Learning* (*ICML2000*), Stanford, California, 2000.
30 **Likas, A., Vlassis, N.** and **Verbeek, J. J.** The global *K*-means clustering algorithm. *Pattern Recognition*, 2003, **36**, 451–461.
31 **Blake, C., Keogh, E.** and **Merz, C. J.** UCI Repository of machine learning databases. Department of Information and Computer Science, University of California Irvine, California, 1998.

APPENDIX

The complexity of the Incremental *K*-means algorithm over *K* insertions is

$$O(K) = \sum_{t=1}^{K} tnR_t \tag{8}$$

where R_t is the number of iterations when the number of clusters is t and n is the number of objects. In the worst case,

$$R_1 = R_2 = \cdots = R_K = \text{maximum number of iterations } R \tag{9}$$

Thus,

$$O(K) = nR\left(\sum_{t=1}^{K} t\right) = \tfrac{1}{2}nRK(K+1) \tag{10}$$

However, when the dataset has well-separated regions, as in the case of the chosen datasets, the insertion of a new cluster may affect only one of the regions. With such a dataset, when t increases, the reduction in the sum of distortions also decreases, as discussed in section 3. This decrease can lower the value of R_t when t increases. The decrease in R_t could compensate for the increase in t. This means the factor (tR_t) could almost be a constant C, which means the complexity of the algorithm becomes

$$O(K) = \sum_{t=1}^{K} nC = KnC \tag{11}$$