Experiments for the Number of Clusters in K-Means

Mark Ming-Tso Chiang and Boris Mirkin

School of Computer Science & Information Systems, Birkbeck University of London, London, UK
{mingtsoc,mirkin}@dcs.bbk.ac.uk

Abstract. K-means is one of the most popular data mining and unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a pre-specified number of clusters K, therefore the problem of determining "the right number of clusters" has attracted considerable interest. However, to the authors' knowledge, no experimental results of their comparison have been reported so far. This paper presents results of such a comparison involving eight selection options presenting four approaches. We generate data according to a Gaussian-mixture distribution with clusters' spread and spatial sizes variant. Most consistent results are shown by the least squares and least modules version of an intelligent version of the method, iK-Means by Mirkin [14]. However, the right K is reproduced best by the Hartigan's [5] method. This leads us to propose an adjusted iK-Means method, which performs well in the current experiment setting.

1 Introduction

K-Means, in its model-free version, arguably is the most popular clustering method now and in predictable future. This is why studying its properties is of interest not only to the classification, data mining and machine learning communities, but also to an increasing numbers of practitioners in business intelligence, bioinformatics, customer management, engineering and other application areas. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters K, therefore the problem of determining "the right number of clusters" attracts considerable interest ([7], [14]). This paper focuses upon an experiment aiming at comparing of various options for selecting the number of clusters in K-Means (see [5], [7], [14]) and analysis of its results. The setting of our experiment is described in section 2. Section 3 presents all the clustering algorithms involved, including the Hartigan-adjusted iK-Means method. A Gaussian-mixture data generator is described in section 4. Our evaluation criteria are in section 5. The evaluation results are presented in section 6. The conclusion focuses on issues and future work.

2 Algorithm Descriptions

2.1 Generic K-Means

K-Means is an unsupervised clustering method that applies to a data set represented by the set of N entities, I, the set of M features, V, and the entity-to-feature matrix

J. Neves, M. Santos, and J. Machado (Eds.): EPIA 2007, LNAI 4874, pp. 395–405, 2007.

$Y=(y_{iv})$, where y_{iv} is the value of feature $v \in V$ for entity $i \in I$. The method produces a partition $S=\{S_1, S_2,\ldots, S_K\}$ of I in K non-overlapping classes S_k, referred to as clusters, each with a specified centroid $c_k=(c_{kv})$, an M-dimensional vector in the feature space (k=1,2,…K). Centroids form set $C=\{c_1, c_2,\ldots, c_K\}$.The criterion, minimised by the method, is the within-cluster summary distance to centroids:

$$W(S, C)=\sum_{k=1}^{K}\sum_{i\in S_k} d(i, c_k) \quad (1)$$

where d is typically the Euclidean distance squared or the Manhattan distance. In the first case criterion (1) is referred to as the square error criterion and in the second, the absolute error criterion.

Given K M-dimensional vectors c_k as cluster centroids, the algorithm updates cluster lists S_k according to the Minimum distance rule. For each entity i in the data table, its distances to all centroids are calculated and the entity is assigned to the nearest centroid. Then centroids are updated according to the criterion used. This process is reiterated until clusters do not change. Before running the algorithm, the original data are pre-processed (standardised) by subtracting the grand mean from each feature wand dividing it by its range. The above algorithm is referred to as *Straight K-Means*.

We use either of two methods for calculating the centroids: one by averaging the entries within clusters and another by taking the within-cluster median. The first corresponds to the least-squares criterion and the second to the least-moduli criterion [14].

2.2 Selection of the Number of Clusters with the Straight K-Means

In K-Means, the number K of clusters is pre-specified (see [10], [6]). Currently, a most popular approach to selection of K involves multiple running K-Means at different K with the follow-up analysis of results according to a criterion of correspondence between a partition and a cluster structure. Such, "internal", criteria have been developed using various probabilistic hypotheses of the cluster structure by Hartigan [5], Calinski and Harabasz [2], Tibshirani, Walther and Hastie [18] (Gap criterion), Sugar and James [17] (Jump statistic), and Krzanowski and Lai [9]. We have selected three of the internal indexes as a representative sample.

There are some other approaches to choosing K, such as that based on the silhouette width index [8]. Another one can be referred to as the consensus approach [15]. Other methods utilise a data based preliminary search for the number of clusters. Such is the method iK-Means [14]. We consider two versions of this method – one utilising the least squares approach and the other the least moduli approach in fitting the corresponding data model.

We use six different internal indexes for scoring the numbers of clusters. These are: Hartigan's index [5], Calinski and Harabasz's index [2], Jump Statistic [17], Silhouette width [8], Consensus distribution's index [15] and the DD index [14], which involves the mean and variance of the consensus distribution.

Before applying these indexes, we run the straight K-Means algorithm for different values of K in a range from START value (typically 4, in our experiments) to END

K-Means Results Generation
For K = START: END
For diff_init=1: number of different K-Means initialisations
- randomly select K entities as initial centroids
- run Straight K-Means algorithm
- calculate W_K, the value of W(S, C) (1) at the found clustering
- for each K , take the clustering corresponding to the smallest W_K among different initialisations

end diff_init
end K

value (typically, 14). Given K, the smallest W(S, C) among those found at different K-Means initialisations, is denoted by W_K. The algorithm is in the box above.

In the following subsections, we describe the statistics used for selecting "the right" K at the clustering results.

2.2.1 Variance Based Approach

Of many indexes based on W_K to estimate the number of clusters, we choose the following three: Hartigan [5], Calinski & Harabasz [2] and Jump Statistic [17], as a representative set for our experiments. Jump Statistic is based on the extended W, according to the Gaussian mixture model. The threshold 10 in Hartigan's index of estimating the number of clusters is "a crude rule of thumb" suggested by Hartigan [5], who advised that the index is proper to use only when the K-cluster partition is obtained from a (K-1)-cluster partition by splitting one of the clusters. The three indexes are described in the box below.

Hartigan (HK):
- calculate $HK=(W_K/W_{K+1}-1)(N-K-1)$, where N is the number of entities
- find the very first K at which HK is less than 10

Calinski and Harabasz (CH):
- calculate $CH=((T-W_K)/(K-1))/(W_K/(N-K))$, where $T=\sum_{i\in I}\sum_{v\in V} y_{iv}^2$ is the data scatter
- find the K which maximises CH

Jump Statistic (JS):
- for each entity i, clustering $S=\{S_1,S_2,\ldots,S_K\}$, and centroids $C=\{C_1,C_2,\ldots,C_K\}$
- calculate $d(i, S_k)=(y_i-C_k)^T\Gamma^{-1}(y_i-C_k)$ and $d_k=(\sum_k \sum_{i\in S_k} d(i, S_k))/M*N$, where

M is the number of features, N is the number of rows and Γ is the covariance matrix of Y
- select a transformation power, typically M/2
- calculate the jumps $JS= d_K^{-M/2} - d_{K-1}^{-M/2}$ and $d_0^{-M/2}\equiv 0$
- find K that maximises JS

2.2.2 Structural Approach

Instead of relying on the overall variance, the Silhouette Width index by Kaufman and Rousseeuw [8] evaluates the relative closeness of individual entities to their clusters. It calculates the silhouette width for each entity, the average silhouette width for each cluster and the overall average silhouette width for the total data set. Using this approach each cluster could be represented by the so-called silhouette, which is based on the comparison of its tightness and separation. The silhouette width s(i) for entity $i \in I$ is defined as

$$s(i)=\frac{b(i)-a(i)}{\max(a(i),b(i))} \tag{2}$$

where a(i) is the average dissimilarity between i and all other entities of the cluster to which i belongs and b(i) is the minimum of the average dissimilarity of i and all the entities in other clusters.

The silhouette width values lie in the range [–1, 1]. If the silhouette width value is close to 1, it means that sample is well clustered. If the silhouette width value for an entity is about zero, it means that that the entity could be assigned to another cluster as well. If the silhouette value is close to –1, it means that the entity is misclassified. The largest overall average silhouette width indicates the best number of clusters. Therefore, the number of clusters with the maximum overall average silhouette width is taken as the optimal number of the clusters.

2.2.3 Consensus Approach

We apply the following two consensus-based statistics for estimating the number of clusters: Consensus distribution area [15] and Average distance [14]. These two statistics represent the consensus over multiple runs of K-Means for different initialisations at a specified K. First of all, consensus matrix is calculated. The consensus matrix $C^{(K)}$ is an $N \times N$ matrix that stores, for each pair of entities, the proportion of clustering runs in which the two entities are clustered together.

An ideal situation is when the matrix contains 0's and 1's only: all runs lead to the same clustering. Consensus distribution is based on the assessment of how the entries in a consensus matrix are distributed within the 0-1 range. The cumulative distribution function (CDF) is defined over the range [0, 1] as follows:

$$\mathrm{CDF}(x)=\frac{\sum_{i<j} 1\{C^{(K)}(i,j) \le x\}}{N(N-1)/2} \tag{3}$$

where 1{cond} denotes the indicator function that is equal to 1 when cond is true, and 0 otherwise. The difference between two cumulative distribution functions can be partially summarized by measuring the area under the two curves. The area under the CDF corresponding to $C^{(K)}$ is calculated using the following formula:

$$A(K)=\sum_{i=2}^{m}(x_i-x_{i-1})CDF(x_i) \quad (4)$$

where set $\{x_1,x_2,\ldots,x_m\}$ is the sorted set of entries of $C^{(K)}$. We can calculate the proportion increase in the CDF area as K increases, computed as follows

$$\Delta(K+1)=\begin{cases} A(K), & K=1 \\ \dfrac{A(K+1)-A(K)}{A(K)}, & K\geq 2 \end{cases} \quad (5)$$

The number of clusters is selected when a large enough increase in the area under the corresponding CDF, which is to find the K which maximises Δ(K). The index average distancing is based on the entries of the consensus matrix $C^{(K)}(i,j)$ obtained from the consensus distribution algorithm. The mean and the variance of these entries $\mu_K$and σ_K^2 for each K can be calculated. We define avdis(K)= μ_K*(1- μ_K)- σ_K^2, which is proven to be equal to the average distance between the R clusterings $M(\{S^t\})=\frac{1}{m^2}\sum_{u,w=1}^{m}M(S^u,S^w)$ (Mirkin 2005, p.229), where $M=(|\Gamma s|+|\Gamma_T|-2a)/\binom{N}{2}$, $|\Gamma s|=(\sum_{t=1}^{K}N_{t+}^2-N)/2$, $|\Gamma_T|=(\sum_{u=1}^{L}N_{+u}^2-N)/2$ in the contingency table of the two partitions (see section 4.3). The index is DD(K)=(avdis(K) - avdis(K+1))/avdis(K+1). The number of clusters is decided by the maximum value of DD(K).

2.3 Choosing K with Intelligent K-Means

Another approach to selecting the number of clusters is proposed in Mirkin [14] as the so-called intelligent K-Means. It initialises K-Means with the so-called Anomalous pattern approach, which is described in the box below:

Anomalous Pattern (AP):

1. Find an entity in I, which is the farthest from the origin and put it as the AP centroid c.
2. Calculate distances $d(y_i,c)$ and $d(y_i,0)$ for each i in I, and if $d(y_i,c)<d(y_i,0)$, y_i is assigned to the AP cluster list S.
3. Calculate the centroid c' in the S. If c' differs from c, put c' as c, and go to step 2, otherwise go to step 4
4. Output S and its centroid as the Anomalous Pattern.

The distance and centroid in the AP with the Least Squares criterion are the Euclidean squared and the average of the within-cluster entries, respectively, whereas the iK-means with the Least Modules criterion are the Manhattan distance and the median of the within- cluster entries, respectively.

The intelligent K-Means algorithm iteratively applies the Anomalous Pattern procedure and after no unclustered entities remain, removes the singletons and takes the centroids of remaining clusters and their quantity to initialise K-Means. The algorithm is as follows:

Intelligent K-means:

0. Put t=1 and I_t the original entity set.
1. Apply AP to I_t to find S_t and C_t.
2. If there are unclustered entities left, put $I_t \leftarrow I_t - S_t$ and t=t+1 and go to step 1.
3. Remove all the found clusters with the cluster size 1. Denote the number of remaining clusters by K and their centroids by $c_1, c_2, \ldots, c_K$.
4. Do Straight K-means with $c_1, c_2, \ldots, c_K$ as initial centroids.

The iK-Means algorithm differs from those in the previous section by the following: (a) it uses just one run of the iterative AP algorithm, (b) it utilizes yet another parameter, the discarding threshold, which is taken to be DT=1 in the following up experiments.

To further enhance iK-Means methods LS and LM, one can make a notice that, according to Chiang and Mirkin [3], LS and, especially, LM may produce sometimes excessive numbers of clusters, which may contribute to their losing to other methods in these situations. We interpret this as an indication that the discarding threshold value DT, set to be always 1, can be overly restrictive. On the other hand, the tables show that HK results, on average, reasonably reproduce the number of clusters generated. This leads us to suggest that the HK number-of-cluster results should be taken as a reference to adjust the discarding threshold in iK-Means. An iterative procedure for such an adjustment is in the box below.

HK-adjusted iK-Means

0. HK-number: Find the number of clusters *K_h* by using R runs of Straight K-Means at each *K* with the Hartigan rule.
1. iK-Means number: Find the number of clusters by using iK-Means with the discarding threshold *DT=1*. Let it be *K_ls* for LS and *K_lm* for LM.
2. Adjust: If *K_ls* (or *K_lm*) is 1.15 times greater than *K_h*, increase the discarding threshold by 1 and go to step 1 with the updated *DT*. Otherwise, halt. (The adjustment factor value of 1.15 has been found experimentally.)

2.4 Selection

Here is the list of methods for finding the number of clusters in our experiment, with the acronyms assigned:

Table 1. Set of methods for selection of the number of clusters in K-Means under comparison

Method	Acronym
Hartigan	HK
Calinski & Harabasz	CH
Jump Statistic	JS
Silhouette Width	SW
Consensus Distribution area	CD
Davdis	DD
Least Squares	LS
Least Modules	LM
Adjusted Least Squares	ALS
Adjusted Least Modules	ALM

3 Data Generation for the Experiment

There is a popular distribution in the literature on computational intelligence, the mixture of Gaussian distributions, which can supply a great variability of cluster shapes, sizes and structures ([1] and [10]). Yet there is an intrinsic difficulty related to the huge number of parameters defining a Gaussian mixture distribution: (a) the cluster probabilities; (b) cluster centres; and (c) cluster covariance matrices, of which the latter involve $KM^2/2$ parameters, where M is the number of features, which is about a 1000 at K=10 and M=15 – by far too many for modelling in an experiment. However, there is a model involving the so-called Probabilistic Principal Components (PPCA) framework that uses an underlying simple structure covariance model ([16] and [19]).

The Gaussian mixture data are generated as implemented in a MATLAB Toolbox freely available on the web [4]. Our sampling functions are based on a modified version of that proposed in Wasito and Mirkin [20]. The mixture model type in the functions defines the covariance structure. We use either of two types: the spherical shape or the probabilistic principal component analysis (PPCA) shape [19]. The cluster spatial sizes are taken constant at the spherical shape, and variant at the PPCA shape. The cluster spatial size with the PPCA structure can be defined by multiplying its covariance matrix by a factor. We maintain two types of the cluster spatial size factors: the linear and quadratic distributions of the factors. To implement these, we take the factors to be proportional to the cluster's index k (the linear distribution being k-proportional) or k^2 (the quadratic distribution being k^2-proportional) (k=1,…,K).

Cluster centroids are generated randomly from a normal distribution with mean 0 and standard deviation 1 and then they are scaled by a factor expressing the spread of the centroids. Table 2 presents experimentally chosen spread values, which are used in the experiments. The PPCA model runs with the manifest number of features 15 and the dimension of the PPCA subspace equal to 6.

In the experiments, we generated Gaussian mixtures with 9 clusters. The cluster proportions (priors) we took were uniformly random.

Table 2. Cluster spread used in the experiments

Spread	Spherical	PPCA	
		k-proport.	k^2-proport.
Large	2	10	10
Small	0.2	0.5	2

4 Evaluation Criteria

4.1 Number of Clusters

This criterion is based on the difference between the number of generated clusters (7 or 9) and that in the selected clustering.

The number of clusters measure is rather rough; it does not take into account the clusters' content, that is, similarity between generated clusters and those found with the algorithms.

4.2 Distance Between Centroids

This is not quite an obvious criterion when the number of clusters in a resulting partition is greater than the number of clusters generated. In our procedure, we use three steps to score the similarity between the real and obtained centroids: (a) assignment, (b) distancing and (c) averaging. These steps are described in the box below for both the weighted and unweighted distance cases in terms of centroids $e_1,e_2,\ldots e_L$ of found clusters $Q_1,Q_2,\ldots Q_L$, and generated centroids $g_1,g_2,\ldots,g_K$ of generated clusters $P_1, P_2, \ldots, P_K$.

Distance between two sets of centroids

1. *Assignment:* For each $k=1,\ldots.K$, assign g_k with that e_l which is the closest to it.

If there remains any not assigned centroid e_i, find that g_k that is the nearest to it.

2. *Finding distances:* Denote by E_k the set of those e_l that have been assigned to g_K and take $\alpha_{lk} = q_l/|E_k|$ (weighted version) or $\alpha_{lk} = 1$ (unweighted version). Define, for each $k=1,\ldots,K$

$dis(k)= \sum_{e_1 \in E_k} d(g_k,e_l)* \alpha_{lk}$. (The distance d here is Euclidean squared distance.)

3. *Averaging:* Calculate $D= \sum_{k=1}^{K} p_k * dis(k)$ where $p_k=N_k= |N_k|$, in the weighted version, or $p_k= 1$, in the unweighted version.

4.3 Partition Confusion Measures

To measure the similarity between two partitions, their contingency (confusion) table is to be used. The entries in the contingency table are the co-occurrences of the generated partition clusters (row category) and the obtained clusters (column category), that is, counts of numbers of entities that fall simultaneously in both clusters. The generated cluster (row category) is denoted by $k \in T$, the obtained partition (column category) is denoted by $h \in U$ and the co-occurrences counts are denoted by N_{kh}. The frequencies of row and column categories usually are called marginals and denoted by N_{k+} and N_{+h} . The probabilities are defined accordingly: $p_{kh}=N_{kh}/N$, $p_{k+}=N_{k+}/N$, and $p_{+h}=N_{+h}/N$, where N is the total number of entities. Of the four used contingency-based measures (the relative distance, Tchouproff coefficient, the average overlap, and the adjusted Rand index), only the adjusted Rand index will be presented.

The adjusted Rand index ([6], [21]) is defined as follows:

$$Ari = \frac{\sum_{k\in T}\sum_{h\in U}\binom{N_{kh}}{2} - \left[\sum_{k\in T}\binom{N_{k+}}{2}\sum_{h\in U}\binom{N_{+h}}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_{k\in T}\binom{N_{k+}}{2} + \sum_{h\in U}\binom{N_{+h}}{2}\right] - \left[\sum_{k\in T}\binom{N_{k+}}{2}\sum_{h\in U}\binom{N_{+h}}{2}\right] / \binom{N}{2}} \tag{6}$$

where $\binom{N}{2} = \frac{N(N-1)}{2}$.

5 Evaluation Results

The evaluation results with the adjusted iK-means methods are shown in Table 3. The distance between centroids shown in Table 3 are rescaled according to Table 2 in such away that the distances at different spatial size distributions become comparable. Specifically, at the small spreads, the spread factor at k2-proportional distribution, 2, is four times greater than that at k-proportional, 0.5, and 10 times greater than that at the equal sizes, 0.2. By multiplying the distances between centroids at equal sizes by 100=102 and at k-proportional sizes by 16=42, we make them comparable with those at the k2-proportional distribution. (Note, the distance between centroids is Euclidean squared, which implies the need in the quadratic adjustment of the factors.) Similarly, at the large spreads, the spread factors at the variant size distributions are the same while that at the constant size is 5 times less. Multiplying the distances between centroids at equal sizes by 25, we make all the distances comparable. After having done this, we can see one more effect in the Table 3. All methods perform better when the cluster spatial sizes are less different: at the constant sizes the best and at the k2-proportional sizes the worst. This can be seen as conforming to the idea that K-Means best delivers at constant radius clusters indeed. The HK adjustment works too: ALS and ALM join HK to perform better on the number of clusters. On the other evaluation measures, the pattern supports the view that the iK-Means based methods perform well. LS, LM, ALS and ALM perform better over both the distance between centroids and cluster contents. Overall, the adjusted versions ALS and ALM win at most situations.

Table 3. The average values of evaluation criteria at 9-clusters data sets with NetLab Gaussian covariance matrix for the large and small spread values (LaS and SmS, respectively) in Table 2. The standard deviations are after slash, per cent. The three values in a cell refer to the three cluster structure models: the spherical on top, the PPCA with k-proportional cluster sizes in the middle, and the PPCA with k2-proportional cluster sizes in the bottom. Two winners of the ten methods (eight from Table 1 plus adjusted LS and LM which are denoted by ALS and ALM) and are highlighted using the bold font, for each of the options. Distances between centroids are rescaled according to Table 2, as explained in section 6.

	Estimated number of clusters		Adjusted distance between centroids		Adjusted Rand Index	
	LaS	SmS	LaS	SmS	LaS	SmS
HK	8.27/6	**7.6/10**	10310.00/13	38601.00/14	0.89 / 9	0.29/10
	8.55/7	**9.4 / 9**	11833.21/14*	**47448.96**/15	0.90 / 9	0.37/11
	9.35/7	9.12/10	12154.99/15	55286.55/14	0.84 / 9	0.28/12
CH	11.55/8	4.00 / 0	10096.25/12*	41927.00/12	0.82 / 9	0.25/12
	12.10/4	5.30 / 5	11788.38/14*	46924.64/19	0.81 / 8	0.21/12
	11.15/8	4.11 / 8	12146.83/13	53779.46/15	0.79 / 9	0.22/12
JS	12.12/8	4.50 / 0	10084.50/13*	41927.00/13	0.77/10	0.25/12
	12.75/9	6.15 / 8	11785.21/13*	46533.28/15	0.82 / 8	0.24/13
	12.10/8	4.45 / 5	12131.86/12	53699.24/14	0.80 / 8	0.22/11
SW	6.29/8	4.54/10	10456.50/12	41866.00/14	0.92/10	0.26/13
	6.95/7	4.95 / 4	11876.31/13*	45540.96/16	0.92 / 8	0.27/12
	7.15/8	4.28/11	12203.58/12	53583.12/16	0.85 / 6	0.22/13
CD	5.31/7	5.11 / 9	10749.00/12	37393.00/12	0.78/12	0.27/13
	5.30/6	5.10/10	11943.98/13	46361.76/18	0.78/12	0.28/14
	5.20/6	5.31 / 9	12265.98/12	55040.86/15	0.75/12	0.25/13
DD	5.67/3	6.42 / 8	10884.25/12	40997.00/13	0.75/12	0.27/12
	4.90/3	5.60 / 9	11979.30/13	47940.48/18	0.74/12	0.24/12
	5.3/3	5.83 / 8	12286.43/12	53912.13/13	0.71/12	0.27/10
LS	8.67/6	13.00/18	**10061.75/12**	**33591.00/23**	**0.99 / 9**	**0.48/12**
	8.80/6	10.80/16	**11771.70/12**	**42582.56/20**	**0.99/10**	**0.42/12**
	7.95/7	13.44/18	12031.13/11	54026.92/15	0.90 / 9	**0.45/12**
LM	**9.33/6**	25.00/18	**10004.50/12**	38112.00/25	0.92 / 9	0.38/12
	8.80/7	16.10/17	**11767.34/13**	**42377.60/20**	**0.99/10**	0.41/12
	10.00/6	23.11/18	12114.01/12	53507.21/16	0.84/10	**0.41/12**
ALS	8.50/5	**7.60 / 6**	10086.75/12*	33849.00/12	**0.99/11**	**0.50/11**
	8.70/7	**9.90 / 7**	11871.70/15*	43536.32/11	**0.99/11**	**0.42/12**
	8.70/9	**9.40 / 9**	**11031.13/12**	**52098.21/12**	**0.95/11**	0.38/12
ALM	**8.70/6**	7.50 / 6	10504.50/12	**30556.00/12**	**0.99/12**	0.44/10
	8.70/7	10.60 / 9	11867.34/15*	44298.88/11	**0.99/10**	0.38/11
	9.50/9	**9.60 / 9**	**10114.01/13**	**53057.21/11**	**0.92/13**	0.35 / 9

* within 1% of the best value.

6 Conclusions

Of the ten different procedures considered in these experiments, most consistent results are shown by the least-squares version of iK-Means LS [14] closely followed by its least-moduli counterpart LM. Rather unexpectedly, Hartigan's "rule of thumb" HK [5], appears to best reproduce the number of clusters, which can be used for the adjustment of the discarding coefficient in iK-Means methods as described above.

The future research should include, first of all, greater coverage of potential data distributions, in terms of greater freedom in covariance parameters of the Gaussians as well as involving other types of distributions.

References

1. Banfield, J.D., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics 49, 803–821 (1993)
2. Calinski, T., Harabasz, J.: A Dendrite method for cluster analysis. Communications in Statistics 3(1), 1–27 (1974)
3. Chiang Mark, M.T., Mirkin, B.: Determining the number of clusters in the Straight K-means: Experimental comparison of eight options. In: Proceeding of the 2006 UK workshop on Computational Intelligence, pp. 119–126 (2006)
4. Generation of Gaussian mixture distributed data, NETLAB neural network software (2006), http://www.ncrg.aston.ac.uk/netlab
5. Hartigan, J.A.: Clustering Algorithms. J. Wiley & Sons, New York (1975)
6. Hubert, L.J., Arabie, P.: Comparing partitions. Journal of Classification 2, 193–218 (1985)
7. Jain, A.K, Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
8. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. J. Wiley & Son, New York (1990)
9. Krzanowski, W., Lai, Y.: A criterion for determining the number of groups in a dataset using sum of squares clustering. Biometrics 44, 23–34 (1985)
10. McLachlan, G., Basford, K.: Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York (1988)
11. McQueen, J.: Some methods for classification and analysis of multivariate observations. In: Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. II, pp. 281–297 (1967)
12. Milligan, G.W., Cooper, M.C.: An examination of procedures for determining the number of clusters in a data set. Psychometrika 50, 159–179 (1985)
13. Mirkin, B.: Eleven ways to look at the Pearson chi squares coefficient at contingency tables. The American Statistician 55(2), 111–120 (2001)
14. Mirkin, B.: Clustering for Data Mining: A Data Recovery Approach. Chapman and Hall/CRC, Boca Raton Fl (2005)
15. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. Machine Learning 52, 91–118 (2003)
16. Roweis, S.: EM algorithms for PCA and SPCA. In: Jordan, M., Kearns, M., Solla, S. (eds.) Advances in Neural Information Processing Systems, vol. 10, pp. 626–632. MIT Press, Cambridge (1998)
17. Sugar, C.A., James, G.M.: Finding the number of clusters in a data set: An information-theoretic approach. Journal of American Statistical Association 98(463), 750–778 (2003)
18. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a dataset via the Gap statistics. Journal of the Royal Statistical Society B 63, 411–423 (2001)
19. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. J. Roy. Statist. Soc. Ser. B 61, 611–622 (1999)
20. Wasito, I., Mirkin, B.: Nearest neighbours in least-squares data imputation algorithms with different missing patterns. Computational Statistics & Data Analysis 50, 926–949 (2006)
21. Yeung, K.Y., Ruzzo, W.L.: Details of the Adjusted Rand index and clustering algorithms. Bioinformatics 17, 763–774 (2001)