Intelligent choice of the number of clusters in K-Means clustering: an experimental study with different cluster spreads[1]

Mark Ming-Tso Chiang and Boris Mirkin

School of Computer Science & Information Systems, Birkbeck University of London, London, UK

Email: {mingtsoc, mirkin}@dcs.bbk.ac.uk

Abstract

The issue of determining "the right number of clusters" in K-Means has attracted considerable interest, especially in the recent years. Cluster overlap appears to be a factor most affecting the clustering results. This paper proposes an experimental setting for comparison of different approaches at data generated from Gaussian clusters with the controlled parameters of between- and within-cluster spread to model different cluster overlaps. The setting allows for evaluating the centroid recovery on par with conventional evaluation of the cluster recovery. The subjects of our interest are two versions of the "intelligent" K-Means method, ik-Means, that find the right number of clusters one-by-one extracting "anomalous patterns" from the data. We compare them with seven other methods, including Hartigan's rule, averaged Silhouette width and Gap statistic, under six different between- and within- cluster spread-shape conditions. There are several consistent patterns in the results of our experiments, such as that the right K is reproduced best by Hartigan's rule – but not clusters or their centroids. This leads us to propose an adjusted version of iK-Means, which performs well in the current experiment setting.

1 Introduction

The problem of determining "the right number of clusters" attracts considerable interest in the literature (for reviews, see Jain and Dubes (1988), Dudoit and Fridlyand (2002), Mirkin (2005), Steinley (2006) and section 3 below). Most papers propose a procedure for estimating the number of clusters and experimentally compare it to some other methods. Some authors do more comprehensive experiments and either arrive at some winning procedures, like Milligan and Cooper (1985) in their seminal study of 30 indexes for cutting cluster hierarchies, or obtain inconclusive results like Hardy (1996) and Dimitraidou, Dolnicar and Weingessel (2002). Steinley and Henson (2005) pointed out that it is very important, in experiments with simulated data, to maintain a degree of cluster overlap to be able to derive any realistic conclusions, which was not the case in previously published experimental studies. They propose a model for data generation with overlapping clusters, which however contains too many parameters and can model only one-

---

dimensional overlaps. In a follow-up experimental study of different initialization strategies, Steinley and Brusco (2007) come to the conclusion that cluster overlap is the property of generated data that most affects the cluster recovery.

This paper focuses on experimental comparison of various options for selecting the number of clusters in the most popular partitioning method, K-Means, at different cluster overlapping conditions. Specifically, we analyze the performance of an "intelligent" version of K-Means, iK-Means (Mirkin 2005), which initializes K-Means with the so-called Anomalous pattern (AP) clusters that are furthest away from the origin of the feature space. This method is compared with a number of other methods, each attempting at selection of the right number of clusters from a range of numbers, by using a specially designed index. To choose these other methods, we undertake a systematic review of more than a dozen recently published methods for identifying the right number of clusters in K-Means, as well as some earlier experimentally supported recipes.

We utilize the conventional Gaussian distribution as the cluster generation model, with its parameters, the mean point and covariance matrix, being naturally interpretable in terms of location and shape, respectively. The cluster overlap, in this setting, can be modelled by combining two factors, the within-cluster spread and between-cluster spread, rather than as an independent property. This allows us for easily modelling such situations when the centers of two clusters are close to each other (a small between-cluster spread) but are well separated because of small (co)variances, while another cluster, with its center being much further away, may overlap either or both of them, because of its large (co)variances as illustrated on Figure 1.
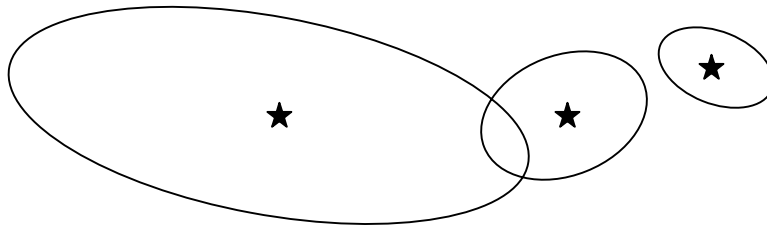

Figure 1. An illustration of the cluster overlap depending on the distance between cluster centroids (represented by pentagrams), and their geometric sizes (represented by ellipses): two clusters on the right are close to each other but well separated, whereas the cluster on the left is further away but not separated because of its larger spread.

Because of both the usage of Gaussians in the data generation and specifics of K-Means, which finds both clusters and their centroids, we can compare methods' performances not only in terms of the recovery of the number of clusters or cluster themselves, as usual, but also in terms of their capabilities in recovering cluster centroids. This is not a trivial matter, because cluster centroids may be as, or even more, important as clusters themselves: they may represent the

---

our revision of the paper.

conceptual, or intensional, meaning of clusters. An important issue emerging in this respect is whether the cluster sizes should be taken into account or not, the latter under the assumption that the centroid of a small cluster, containing just a few entities, can be as important as the centroid of a large cluster. We suggest that this issue can be addressed by comparing the clustering methods' centroid recovery performances with their cluster recovery performances. Since these two performances should go in line, that method of measuring the centroid recovery performance that makes it more similar to the cluster recovery performance should be preferred. In our experiments, the un-weighted centroids win indeed.

Another experimental finding is that the number of clusters is best reproduced by Hartigan's (1975) method, though the method's performance regarding the cluster or centroid recovery is not that impressive. On the other hand, iK-Means performs rather well in terms of the cluster and centroid recovery, but may drastically overestimate the number of clusters, especially at the small between-cluster spreads. This leads us to propose an adjusted version of iK-Means, which performs rather well on all three counts – the number of clusters, the cluster recovery and the centroid recovery.

The paper is organized as follows. Generic versions of K-Means and intelligent K-Means are described in Section 2. Section 3 contains a review of methods for finding the right K in K-Means in the published literature. We distinguish between five approaches as based primarily on: cluster variance, within-cluster cohesion versus between-cluster separation, consensus distribution, hierarchical clustering, and resampling. The setting of our experiments at the comparison of nine selected methods for finding the "right clustering" – the data sizes, the cluster shapes, the within- and between-cluster spread parameters, and evaluation criteria - is described in section 4. Basically, we deal with six different cluster overlap settings resulting from combining two types of the between-cluster spread and three models of the within-cluster shape and spread. Section 5 presents results of our experiments in tables containing the evaluation criteria values, averaged over multiple data generations at each of the six settings, along with issues raised before the experiments and answers to them coming from the results. Section 6 concludes the paper.

2. K-Means and intelligent K-Means.

2.1. Generic K-Means

K-Means clustering method, conventionally, applies to a dataset involving a set of N entities, I, set of M features, V, and an entity-to-feature matrix $Y=(y_{iv})$, where y_{iv} is the value of feature $v \in V$ at entity $i \in I$. The method produces a partition $S=\{S_1, S_2,..., S_K\}$ of I in K non-empty non-overlapping

classes S_k, referred to as clusters, each with a centroid $c_k=(c_{kv})$, an M-dimensional vector in the feature space $(k=1,2,...K)$. Centroids form the set $C=\{c_1, c_2,..., c_K\}$. The criterion, alternatingly minimized by the method, is the sum of within-cluster distances to centroids:

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

where d is a distance measure, typically the squared Euclidean distance or Manhattan distance. In the former case criterion (1) is referred to as the squared error criterion and in the latter, the absolute error criterion.

Given K M-dimensional vectors c_k as cluster centroids, the algorithm updates clusters 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 its nearest centroid. Given clusters S_k, centroids c_k are updated according to the distance d in criterion (1), $k=1, 2, ..., K$. Specifically, c_k is calculated as the vector of within-cluster averages if d in (1) is Euclidean distance squared and as of within-cluster medians if d is Manhattan distance. This process is reiterated until clusters S_k stabilize. This algorithm is sometimes referred to as Batch K-Means or Straight K-Means.

When the distance d in (1) is indeed the squared Euclidean distance, K-Means can be seen as an implementation of the alternating optimization procedure for maximization of the maximum likelihood under the assumed mixture of "spherical" Gaussian distributions model, in which all covariance matrices are equal to a diagonal matrix $\sigma^2 I$ where I is the identity matrix and σ^2 the variance value (Hartigan 1975, Banfield and Raftery 1993, McLachlan and Peel 2000). Another, somewhat lighter interpretation comes from the data mining paradigm, in which (1) is but the least-squares criterion for approximation of the data with a data recovery clustering model (Mirkin 1990, 2005) that states that every entry y_{iv} in the data matrix (i denotes an entity and v a feature), can be presented as approximated by the "hidden" set of clusters $S=\{S_1, S_2,..., S_K\}$ and their centers $C=\{c_1, c_2,..., c_K\}$ through equations

$$y_{iv} = \sum_{k=1}^{K} c_{kv} s_{ik} + e_{iv}, \qquad (2)$$

where $s_k=(s_{ik})$ is S_k membership vector in which $s_{ik}=1$ if $i \in S_k$ and $s_{ik}=0$ otherwise, and e_{iv} are residuals to be minimized over unknown c_k and s_k ($k=1,2,...,K$). Criterion (1) is the least-squares or least-moduli fitting criterion for model (2) if d in (1) is the squared Euclidean distance or Manhattan distance, respectively.

More on K-Means and its history can be found in reviews by Steinley (2006) and Bock (2007).

What is important in this is that, both K and initial centroids are to be pre-specified to initialize the method. The algorithm converges to a local minimum of criterion (1) rather fast, and the goodness of the stationary solution much depends on the initialization.

2.2. Choosing K with the Intelligent K-Means

A version of K-Means in which the number of clusters and initial centroids are determined with a procedure targeting "anomalous patterns" as the candidates for the initial centroids has been recently described as "intelligent K-Means" algorithm, iK-Means in Mirkin (2005). It initializes K-Means by standardizing the data in such a way that the origin is put into a "normal" point, usually the gravity centre of all the data points, and iterating then the so-called Anomalous Pattern (AP) algorithm which builds clusters one by one, starting from that which is the furthest away from the origin, and reapplying the process to the entities remaining not clustered yet. This is a version of the so-called Principal Cluster Analysis approach that emulates the one-by-one strategy of the Principal component analysis applied to model (2): an AP pattern is a cluster derived from model (2) at $K=1$ in such a way that it maximally contributes to the data scatter (Mirkin 1990). The fact that AP cluster is far away from the normal point conforms to the notion of interestingness in data mining: the farther from normal, the more interesting (Fayyad et al 1996). Those of the found clusters that are too small, that is, singletons, are removed from the set, and the centroids of the rest are taken as the initial setting for K-Means.

The AP algorithm starts from that entity, which is the farthest from the origin, as the initial centroid c. After that, a one-cluster version of the generic K-Means is utilized. The current AP cluster S is defined as the set of all those entities that are closer to c than to the origin, and the next centroid c is defined as the center of gravity of S. This process is iterated until convergence. The convergence is guaranteed because the process alternatingly minimizes criterion (1) at K=2 with $S_1=S$, $S_2=I-S$, and centroids $c_1=c$ and $c_2=0$, the origin which is kept unchanged through the iterations. The final S, along with its centroid c and its contribution to the data scatter, is the output AP cluster. After it is removed from the data set, the process of extracting of AP clusters is reiterated without ever changing the origin, until no entity remains. Centroids of those of AP clusters that have more than one entity are used as c set at the initialization of K-Means.

We implemented the intelligent K-Means procedure in two versions depending on the criterion behind formula (1): least squares (LS) and least moduli (LM). The LM version has some advantages over LS at data with skewed feature distributions according to our experiments (not presented here), which goes in line with the conventional wisdom regarding the least-moduli

estimates.

The intelligent K-Means procedure seems appealing both intuitively and computationally, and it leads to interpretable solutions in real-world problems. Therefore, it seems reasonable to put it to empirical testing. A version of the method, with a pre-specified K and with no removal of singletons, has been tested by Steinley and Brusco (2007), leading to rather mediocre results in their experiments. Here we intend to test the original version of the iK-means as a device for identifying both the number K and initial centroids.

3. Approaches to choosing K in K-Means

There have been a number of different proposals in the literature for choosing the right *K* after multiple runs of K-Means, which we categorize in five broad approaches:

(i)   Variance based approach: using intuitive or model based functions of criterion (1) which should get extreme values at a correct *K*;

(ii)   Structural approach: comparing within-cluster cohesion versus between-cluster separation at different *K*;

(iii)   Consensus distribution approach: choosing *K* according to the distribution of the consensus matrix for sets of K-Means clusterings at different *K*;

(iv)   Hierarchical approach: choosing *K* by using results of a divisive or agglomerative clustering procedure;

(v)   Resampling approach: choosing *K* according to the similarity of K-Means clustering results on randomly perturbed or sampled data.

We describe them in the following subsections. Let us denote the minimum of (1) at a specified K by W_K. Empirically, one can run K-Means R times starting using random subsets of K entities for initialization and use the minimum value of (1) at obtained clusterings as a W_K estimate.

3.1. Variance based approach

There have been several different W_K based indices proposed to estimate the number of clusters *K* (see Calinski and Harabasz (1974), Hartigan (1975), Krzhanowski and Lai (1985), Tibshirani, Walther and Hastie (2001), Sugar and James (2003)). The issue is that W_K itself cannot be used for the purpose since it monotonically decreases when *K* increases. Thus, various "more sensitive" characteristics of the function have been utilised based on intuitive or statistical modelling of the situation. Of these, we take four –two heuristic measures that have been experimentally approved by Milligan and Cooper (1985) and two model-based more recent indexes:

(A) A Fisher-wise criterion by Calinski and Harabasz (1974) finds *K* maximizing *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. This criterion showed the best performance

in the experiments by Milligan and Cooper (1985), and was subsequently utilized by some authors for choosing the number of clusters (for example, Casillas et al 2003).

(B) A heuristic rule by Hartigan (Hartigan 1975) utilizes the intuition that when clusters are well separated, then for $K<K^*$, where K^* is the "right number" of clusters, a $(K+1)$-cluster partition should be the K-cluster partition with one of its clusters split in two. This would drastically decrease W_K. On the other hand, at $K>K^*$, both K- and $(K+1)$-cluster partitions will be equal to the "right" cluster partition with some of the "right" clusters split randomly, so that W_K and W_{K+1} are not that different. Therefore, as "a crude rule of thumb", Hartigan (1975, p. 91) proposed calculating $HK=(W_K/W_{K+1} - 1)(N-K-1)$, where N is the number of entities, while increasing K so that the very first K at which HK becomes less than 10 is taken as the estimate of K^*. The Hartigan's rule can be considered a partition-based analogue to the Duda and Hart (1973) criterion involving the ratio of the criterion (1) at a cluster and at its two-cluster split, which came very close second-best winner in the experiments of Milligan and Cooper (1985). It should be noted that, in our experiments, the threshold 10 in the rule is not very sensitive to 10-20% changes.

(C) The Gap Statistic introduced by Tibshirani, Walther and Hastie (2001) has become rather popular, especially, in the bioinformatics community. This method, in the authors-recommended version, compares the value of (1) with its expectation under the uniform distribution. Analogously to the previously described methods, it takes a range of K values and finds W_K for each K. To model the reference values, a number, B, of uniform random reference datasets over the range of the observed data are generated so that criterion (1) values W_{Kb} for each $b=1,...,B$ are obtained. The Gap statistic is defined as $Gap(K)=1/B \sum_b log(W_{Kb})-log(W_K)$. Then the average $GK = 1/B \sum_b log(W_{Kb})$ and its standard deviation $sd_K=[1/B \sum_b (log(W_{Kb})-GK)^2]^{1/2}$ are computed leading to $s_K=sd_K \sqrt{1+1/B}$. The estimate of K^* is the smallest K such that $Gap(K) \geqq Gap(K+1)-s_{K+1}$ (Tibshirani, Walther and Hastie 2001).

(D) The Jump Statistic (Sugar and James 2003) utilizes the criterion W in (1) extended according to the Gaussian distribution model. Specifically, the distance between an entity and centroid in (1) is calculated as $d(i, S_k)=(y_i-C_k)^T \Gamma_k^{-1}(y_i-C_k)$, where Γ_k is the within cluster covariance matrix. The jump is defined as $JS(K)= W_K^{-M/2} - W_{K-1}^{-M/2}$ assuming that $d_0^{-M/2} \equiv 0$. The maximum jump $JS(K)$ corresponds to the right number of clusters. This is supported with a mathematical derivation stating that if the data can be considered a standard sample from a mixture of Gaussian distributions

at which distances between centroids are great enough, then the maximum jump would indeed occur at *K* equal to the number of Gaussian components in the mixture (Sugar and James 2003).

3.2. Within-cluster cohesion versus between-cluster separation

A number of approaches utilize indexes comparing within-cluster distances with between cluster distances: the greater the difference the better the fit; many of them are mentioned in Milligan and Cooper (1985). Two of the indexes are the point-biserial correlation, that is, the correlation coefficient between the entity-to-entity distance matrix and the binary partition matrix assigning each pairs of the entities 1, if they belong to the same cluster, and 0, if not, and its ordinal version proposed by Hubert and Levin (1976). These two show a very good performance in the Milligan and Cooper's tests. This, however, perhaps can be an artefact of the very special type of cluster structure utilized by Milligan and Cooper (1985): almost equal sizes of the generated clusters. Indeed, a mathematical investigation described in Mirkin (1996, pp. 254-257) shows that the point-biserial correlation expresses the so-called "uniform partitioning" criterion, which tends to produce equal-sized clusters.

More recent efforts in using indexes relating within- and between-cluster distances are described in Shen et al. (2005) and Bel Mufti et al. (2006).

A well-balanced coefficient, the silhouette width, which has shown good performance in experiments (Pollard and van der Laan 2002), was introduced by Kaufman and Rousseeuw (1990). The concept of silhouette width involves the difference between the within-cluster tightness and separation from the rest. Specifically, the silhouette width *s(i)* for entity $i \in I$ is defined as:

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

where *a(i)* is the average distance between *i* and all other entities of the cluster to which *i* belongs and *b(i)* is the minimum of the average distances between *i* and all the entities in each other cluster. The silhouette width values lie in the range from –1 to 1. 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 width value is close to –1, it means that the entity is misclassified. If all the silhouette width values are close to 1, it means that the set I is well clustered.

A clustering can be characterized by the average silhouette width of individual entities. The largest average silhouette width, over different K, indicates the best number of clusters.

3.3. Consensus distribution approach

The consensus distribution approach relies on the entire set of all R clusterings produced at multiple runs of K-Means, at a given K, rather than just the best of them. The intuition is that the clusterings should be more similar to each other at the right K because a "wrong" K introduces more arbitrariness into the process of partitioning. Thus, a measure of similarity between clusterings should be introduced and utilized. We consider two such measures. One is the Consensus distribution area introduced by Monti et al. (2003). To define it, the consensus matrix is calculated first. The consensus matrix $C^{(K)}$ for R partitions is an $N \times N$ matrix whose (i,j)-th entry is the proportion of those clustering runs in which the entities $i,j \in I$ are in the same cluster. An ideal situation is when the matrix contains 0's and 1's only: this is the case when all the R runs lead to the same clustering. The cumulative distribution function (CDF) of entries in the consensus matrix is:

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

where 1{cond} denotes the indicator function that is equal to 1 when cond is true, and 0 otherwise. The area under the CDF corresponding to $C^{(K)}$ is calculated using the conventional formula:

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

where $\{x_1, x_2, ..., x_m\}$ is the sorted set of different entries of $C^{(K)}$.

We suggest that the average distance between the R partitions can be utilized as another criterion: the smaller, the better. This equals $avdis(K) = \frac{1}{R^2} \sum_{u,w=1}^{R} M(S^u, S^w)$, where distance M is defined as the squared Euclidean distance between binary matrices of partitions S^u and S^w. A binary partition matrix is an entity-to-entity similarity matrix; its (i,j)-th entry is 1 if i and j belong to the same cluster, and 0, otherwise, so that consensus matrix $C^{(K)}$ is the average of all R binary partition matrices. Denote the mean and the variance of matrix $C^{(K)}$ by μ_K and σ_K^2, respectively. Then the average distance can be expressed as $avdis(K) = \mu_K*(1-\mu_K) - \sigma_K^2$ (see Mirkin 2005, p. 229), which also shows how close $C^{(K)}$ to being binary.

To estimate "the right number of clusters", the relative change of the indexes is utilized. Specifically, the relative change in the CDF area in (4) is defined as

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

The average between partitions distance based index is defined similarly except that it decreases rather than increases with the growth of *K*, so that *DD(K)=(avdis(K) - avdis(K+1))/avdis(K+1)*. The number of clusters is decided by the maximum value of *Δ(K)* or *DD(K)*, respectively.

A slightly different approach relating the average distance/Rand measure and the entropy of the consensus distribution on real and artificial data sets has been utilized by Kuncheva and Vetrov (2005).

3.4. Hierarchical clustering approaches

A number of approaches rely on the hierarchy of clustering solutions found by consecutive merging of smaller clusters into larger ones (agglomerative clustering) or by splitting larger clusters into smaller ones (divisive clustering). Duda and Hart (1973) proposed using cluster-related items in the summary criterion (1): the ratio of the summary distance to the centroid in a node of the hierarchical cluster tree over the summary distances to the centroids in the children of the node expresses the "local" drop in distances due to the split. The ratio should be greater than a threshold, to stop the splitting. This criterion, with an adjusted threshold value, showed very good performance in experiments by Milligan and Cooper (1985). Comparing the values of criterion (1) at each split with their average was the base of proposals made by Mojena (1977), leading, though, to rather mediocre results in the experiments by Milligan and Cooper (1985). More recently, the idea of testing individual splits with more advanced statistical tools, namely BIC criterion, was picked up by Pelleg and Moore (2000) and extended by Ishioka (2005) and Feng and Hamerly (2006); these employ a divisive approach with splitting clusters by using 2-Means method.

Some authors propose versions involving combining several techniques. For example, Casillas et al. (2003) utilize the Minimum spanning tree which is split into a number of clusters with a genetic algorithm to meet an arbitrary stopping condition. Six different agglomerative algorithms are applied to the same data by Chae et al. (2006), and the number of clusters at which these partitions are most similar is selected.

3.5. Resampling methods

Resampling, in its wide sense, means using many randomly produced "copies" of the data for assessing statistical properties of a utilized method (see, for instance, Mirkin 2005). Among methods for producing the random copies are: (a) random sub-sampling in the data set; (b) random splitting the data set into "training" and "testing" subsets, (c) bootstrapping, that is, randomly sampling entities with replacement, usually to their original numbers, and (d) adding random noise

to the data entries. All four have been tried for finding the right numbers of clusters based on the intuition that different copies should lead to more similar results at the right number of clusters: see, for example, Minaei-Bidgoli, Topchy and Punch (2004) for (a), Dudoit and Fridland (2002) for (b), McLachlan and Khan for (c), and Bel Mufti, Bertrand and Moubarki (2005) for (d).

Let us describe in brief the popular approach taken by Dudoit and Fridland (2002) following the pioneering work by Breckenridge (1989). For each K, a number B of the following operations is performed: the set is split into non-overlapping training and testing sets, after which the training part is partitioned into K parts; then a classifier is trained on the training set clusters and applied for predicting clusters on the testing set entities. The predicted partition of the testing set is compared with that found, with the same procedure, on the testing set. The result of this B iterations is the median value t(K) of the index of similarity between two partitions of the testing set, that predicted from the training set and that found directly. After that a number of data sets of the same size is generated randomly and the same procedure applies to each of them producing the average value of the index t'(K) under the null hypothesis. The estimated K is that maximizing the difference t(K)-t'(K) under some additional conditions. This procedure, as well as other resampling schemes, involves a number of important parameters such as the type of classifier (taken to be the linear discriminant analysis with the diagonal covariance matrix in Dudoit and Fridlyand 2002), the training-testing split proportion (taken to be 2:1), numbers of iterations and reference sets generated (taken to be 20), the threshold on K values (taken to be 5 or 10), the index of similarity between partitions, etc. The choice of these parameters, which is rather arbitrary, may affect the results. On the same data generating mechanisms, the Dudoit and Fridlyand (2002) setting was outperformed by a model-based statistic as reported by MacLaclan and Khan (2004).

4. Choosing parameters of the experiment in K-Means clustering

The data for experimental comparisons can be taken from real-world applications or generated artificially. Clustering experiments have been conducted, in the published literature, either way or both: over real-world data sets only by Casillas et al. (2003), Minael-Bidgoli et al (2005), Shen et al. (2005), over generated data only by Hand and Krzhanowski (2005), Hardy (2005), Ishioka (2005), Milligan and Cooper (1985), Steinley and Brusco (2007), and over both by Chae et al. 2006, Dudoit and Fridland (2002), Feng and Hamerly (2005), Kuncheva and Vetrov (2005), Maulik and Bandyopadhyay (2000). In this paper, we consider generated data only, to allow us to control the parameters of the experiments. Having the set of parameter values specified, we generate a number of datasets so that the results reported further on are averaged over these datasets. Initially we generated 20 random datasets for each parameter setting (as did Dudoit and Fridlyand 2002) – these are reflected in Tables 2 and 3, but then for the sake of time, we reduced the number of generated

datasets to 10 (in Tables 4 and 5). In the remainder of the section we discuss and set parameters related to the data generation and pre-processing, running algorithms, and evaluation of the results.

4.1. Data and cluster structure parameters

These concern the following.

A. Sizes. First of all, the quantitative parameters of the generated data and cluster structure are specified: the number of entities N, the number of generated clusters K^*, and the number of variables M. In most publications, these are kept relatively small: N ranges from about 50 to 200, M is in many cases 2 and, anyway, not greater than 10, and K^* is of the order of 3, 4 or 5 (see, for example, Casillas et al. 2003, Chae et al. 2006, Hand and Krzhanowski 2005, Hardy 1996, Kuncheva and Petrov 2005, McLachlan and Khan 2004, Milligan and Cooper 1985). Larger sizes appear in Feng and Hamerly (2006) (N= 4000, M is up to 16 and K^*=20) and Steinley and Brusco (2007) (N is up to 5000, M=25, 50 and 125, and K^* =5, 10, 20). Our choice of these parameters is based on the idea that the data should imitate the conditions of real-world data analysis, under the timing constraints of the computational capacity. That means than N should be in thousands while limiting M within one or two dozens, to mimic the situation in which the data analysts select only features relevant to the problem at hand ("tall" data table cases) rather than using all features or key words available ("wide" data table case); the latter should be treated in a different experiment. Another consideration taken into account is that, according to our real-world clustering experiences, it is not the absolute values of M and K^* but rather their ratios, the average cluster sizes, that affect the clustering results. As the major focus of our experiment is the effects of within and between cluster spreads on the clustering results, we decided to keep the ratio restricted, while maintaining two rather distinct values of K^*. Therefore, two settings for the sizes are: (i) N=1000, M=15, K^*=9 – about 110 entities in a cluster on average, and (ii) N=3000, M=20, K^*=21 – about 145 entities in a cluster on average. These are obviously at the upper end of the sizes in the published reports. It should be noted that in the setting (i), we also used K^*=7; this case is not reported, because the results are similar.

It is probably worth mentioning that we do not consider the so-called irrelevant, or noisy, features: The presence of features that have nothing to do with the cluster structure was considered by Milligan and Cooper (1985); see also Dudoit and Fridlyand (2002) and Kuncheva and Vetrova (2005). K-Means partitioning can be and has been applied when no visible cluster structure is present, just to dissect the domain into manageable chunks (Spaeth 1985). The issue of noisy features, in this perspective, deserves a separate consideration.

B. Cluster sizes. The term "size" is ambiguous in the clustering context, because it may refer to

both the number of entities and spatial volume taken by a cluster; we use it here for the number only, in accordance with the practice of Computer Sciences, while utilizing the term "spread" for the geometric size. (Steinley and Brusco 2007 term the cluster size as the "cluster density" –we prefer to utilize this regarding a probabilistic density function.) The difference in cluster sizes can affect the outcome of a clustering process if it is driven by a criterion, such as the point-biserial correlation, that depends on them in a non-linear way. As mentioned in section 3.2, this may affected some of experimental results in Milligan and Cooper (1985) because of the relatively equal cluster sizes utilized by them. However, criterion (1) always involves the same number N of distances, whichever cluster sizes are, so that cluster sizes should not much matter. Steinley and Brusco (2007), who maintained three different patterns for cluster size distributions, report no differences in their results regarding the patterns. Therefore, we decided to disregard this aspect of the cluster structure: our generated clusters have uniformly random size distributions. To generate a random distribution of the cluster size proportions $p=(p_1,...,p_{K*})$ under the condition that elements of p are positive and sum up to 1, one can randomly generate $K*-1$ real numbers $r_1, r_2, ...,r_{K*-1}$ in interval (0,1), sort them in the ascending order so that $r_1< r_2< ...< r_{K*-1}$, and put $r_0=0$ and $r_{K*}=1$, after which the uniformly random proportions are computed as $p_k = r_k - r_{k-1}$ $(k=1,...,K*)$.

C. Cluster shapes. This property is not typically taken into account as a variable to control, because K-Means is conventionally seen as a method for fitting the Gaussian mixture model with spherical Gaussians – and this, in fact, is a property which is directly associated with the Minimum distance rule. However, in real-world applications clusters may have more complex and elongated shapes, which can be, to an extent, be caught by the ellipsoidal shape of the Gaussian clusters (see also McLachlan and Khan 2004, p. 92). Thus, we generate data entities in each cluster by independently sampling from a Gaussian distribution. We take the conventional spherical shape of Gaussian clusters versus another one, much more elongated. Since the number of parameters needed to define the covariance matrix of a Gaussian distribution is in hundreds for our size settings, we utilize a version of the covariance matrix defined with a smaller number of control variables in a MatLab toolbox NetLab (see Generation of Gaussian mixture distributed data 2006). According to the so-called Probabilistic Principal Component Analysis (PPCA) model (Tipping and Bishop 1999), the $M{\times}M$ covariance matrix of a Gaussian distribution in this toolbox is defined by selecting the hidden dimension q as:

$$Cov(\sigma)=W_q*W_q{}'+\sigma^2 I_{M\times M} \qquad (6)$$

where $W_q = \begin{pmatrix} I_{q \times q} \\ 1_{(M-q) \times q} \end{pmatrix}$, $\boldsymbol{I_{n \times n}}$ is an $n \times n$ identity matrix, and $\boldsymbol{1_{n \times m}}$ a $n \times m$ matrix whose all entries are equal to 1. The PPCA model runs with the manifest number of features M and the hidden dimension q. The hidden factor structure is also advocated in Maclachlan and Peel (2000).

It is easy to show that $Cov(0) = \begin{pmatrix} I_{q \times q} & 1_{q \times (m-q)} \\ 1_{(m-q) \times q} & q1_{(m-q) \times (m-q)} \end{pmatrix}$. Obviously, the eigen-values of

$Cov(\sigma)$ are the same as those of $Cov(0)$ with σ^2 added to each; the eigen vectors are the same as well.

The structure of eigenvalues of $Cov(0)$ has been investigated by Wasito and Mirkin (2006) who found that, of q nonzero eigenvalues, the maximal one is $\lambda = 1 + (M-q)q$ whereas all the other $q-1$ eigen-values are equal to unity. This provides for really elongated shapes, so that we could check whether this change of the shape indeed affects the clustering results.

The actual data generation process is based on the spectral decomposition of matrix $Cov(0)$ such as described in Murtagh and Raftery (1984) and Fraley and Raftery (2002). In our experiments q is set to be 6. The variance σ^2 is taken to be 0.1, which is not much important because, anyway, it is multiplied by the within-cluster spread factor described in the following item **D**.

D. Clusters overlap. The possibility of controlling cluster overlap is a much-desired property in clustering experiments. Steinley and Henson (2005) noted that this issue had never been satisfactorily addressed in the literature and proposed a mechanism for generating clusters with an explicitly formalized degree of overlap. Specifically, their model involves a value of the overlap for each pair of clusters over each single feature, thus having a disadvantage of "restricting the generation of the joint distribution clusters to be the product of the marginal distributions" (Steinley and Henson 2005, p. 245). Another problem with this mechanism is by far too many parameters which are not necessarily directly related to parameters of the generated clusters themselves. We consider that the cluster overlap should be modelled in terms of the parameters of distributions used for modelling individual clusters. Since we utilize Gaussian clusters, the overlap should be modelled by using the Gaussian characteristics of location, centers, and cluster shape and spread, covariance matrices. In this way, the overlap between Gaussian clusters can be captured not as an independent property, but a consequence of the two not necessarily related aspects: the distance between cluster centroids ("between-cluster spread") and the magnitude of their variance/covariance values ("within-cluster spread"), as illustrated on Figure 1 in Introduction. By combining these, one can easily model overlaps not only between individual clusters, but also within any subset of clusters.

To control the within-cluster spread, one can multiply the cluster's covariance matrix by a factor. The number of these factors is equal to the number of generated clusters K*. To keep things simple, one should try to define such a distribution of the within-cluster spreads that can be controlled by a single parameter. One obvious definition comes for the model of spherical clusters – all the spreads are equal to each other, that is, all clusters are represented by spheres with a constant radius. This pattern fits well into the theoretical perspective of K-Means as a maximum likelihood method for fitting a Gaussian distribution mixture model in which all individual distributions are spherical with the same variance. However, within the data-mining framework, clusters to be found may have different geometrical sizes. To fit into this perspective, one may use different settings such as several, two or three or four, different within-cluster spread factor values – which would lead then to the task of defining the proportions for each of these types, for which we could find no guidance in the literature or our personal experiences. Therefore, we decided to go along a less challenging path by designing two types of the variant within-cluster spread factors: the "linear" and "quadratic" ones. Specifically, we take the within-cluster spread factor to be proportional to the cluster's index k (the linear, or k-proportional distribution) or k^2 (the quadratic, or k^2-proportional distribution), $k=1, 2, ..., K*$. That is, with the variable within-cluster spreads, the greater the generated cluster index, the greater its spatial size. For example, the within cluster-spread of cluster 7 will be greater than the that of cluster 1, by the factor of 7 in k-proportional model and by the factor of 49 in k^2-proportional model. Since the clusters are generated independently, the within-cluster spread factors can be considered as assigned to clusters randomly. Hence, three different models for the within-cluster spread factors utilized in our experiments are: (i) constant, (ii) k-proportional, and (iii) k^2-proportional. To keep the number of parameters small, we assign a specific cluster shape with each of these models: the spherical shape for the constant spread factor (i), and the PPSA shape (6) for the k- and k^2-proportional factors, (ii) and (iii).

To control the distance between clusters with a single parameter, we utilize a special two-step mechanism for the generation cluster locations. On the first step, all cluster centroids are generated randomly around the origin, so that each centroid entry is independently sampled from a normal distribution $N(0,1)$ with mean 0 and standard deviation 1. On the second step, each of these centroids is shifted away from 0, and from the others, along the line passing through the centroid and space origin, by multiplying it with a positive factor: the greater the factor, the greater the shift, and the greater the distances between centroids. The cluster shift factor is taken the same for all centroids. For our experiments, we consider two types of the between-cluster spread, "small" and "large" ones. These have been experimentally chosen in such a way that the clustering algorithms recover the generated clusters well – at the large spreads, and not so well at the small spreads; see

15

Table 2 for the between-cluster spread factor values at different within-cluster spread-shape models. Typical configurations of datasets with *K**=9 clusters generated as explained above are illustrated on Figure 2.
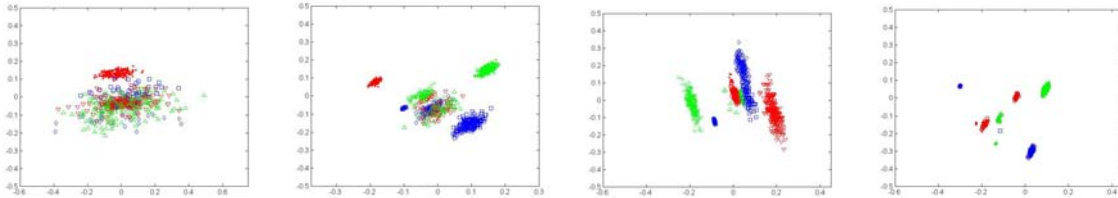

Figure 2. Examples of datasets generated at different data models on a plane defined by the two largest principal components, from the most confusing pattern on the left (PPCA clusters with the quadratic within-cluster spread and the between-cluster spread factor equal to 2) to a clear-cut pattern on the right (the same cluster model, but the between-cluster spread factor grows to 28). The nine clusters are shown with symbols: *,., +, o, x, □, ✳, ▽, △.

E. Feature standardization: In many publications, starting from Milligan and Cooper (1985), the data are generated in such a way that features are comparable and no data standardization is needed, which is very far from the real case scenario. In real-world data, features are usually incomparable so that some form of data standardization is needed. Conventionally, data standardization is conducted as an independent transformation of each individual feature by shifting its origin with the follow-up rescaling.

In statistics, the most popular standardization is the so-called z-scoring which shifts the origin of each feature to its grand mean and then rescales the feature into the units of its standard deviation. This standardization is rooted in the invariance properties of the one-dimensional Gaussian distribution. In the neural network and support vector machine learning literature, the standardization is conventionally performed in a distribution-free way – by shifting the origin to the midrange and relating the result to the half-range so that the boundary values become -1 and +1, which is very convenient for working with target features that tend to have exactly these values (Vapnik 2006).

In cluster analysis, however, more favoured is a mixed standardization in which the origin is shifted to the grand mean, whereas rescaling is conducted according to the range. We can contribute to the debate with the following argument. Dividing the feature scale over the standard deviation is counter-intuitive in the following example that involves two features of the same ranges, so that one of them is uni-modal and the other is bi-modal, as shown on Figure 3, (a) and (b), respectively. The standard deviation of the former is much smaller than that of the latter so that after dividing by the standard deviations the uni-modal feature's range and, thus, contribution to the distances, will be by

far greater than that of the multimodal feature. But intuition tells us that it is rather the bi-modal feature which is more useful for clustering, because the two modes lead to natural subgroups while the uni-modal feature tends to put all, except for the outliers, into the same group.

(a)   **(b)**

Figure 3. Uni-modal distribution shape on (a) versus a bi-modal distribution shape on (b): the standard deviation of the latter is greater, thus making the latter less significant under the z-scoring standardization, which is odd in the clustering context.

Published clustering experiments have demonstrated that the normalization by the range is better for cluster recovery than that by the standard deviation (Milligan and Cooper 1988, Steinley 2004). Shifting to the grand mean has the advantage of mimicking the concept of mechanical inertia (Benzecri 1992) and, also, explicitly relates to established statistics concepts when using mixed scale data (Mirkin 2005). The mixed standardization is adopted in our experiments.

4.2. Algorithmic issues

In this section, we first select K-identifying procedures for our experiments, and then discuss the setting of their parameters.

4.2.1   Selection of procedures for estimation of $K*$

Five different approaches to estimating the "right" number of clusters $K*$ in K-Means are described in the previous section: (i) Variance based, (ii) Structural, (ii) Consensus distribution, (iv) Hierarchical, and (v) Resampling. Of these, we take only three, (i), (ii), and (iii), for our experiments. Each of the other two approaches, both (iv) Hierarchical and (v) Resampling, involves too many diverse ideas and parameters that are absent from the other approaches – these deserve to be explored in separate experiments. As the (i) Variance based approach relates to the criterion of K-Means and has received most theoretical support, we take all four procedures referred to in section 3.2 – Hartigan's "rule of thumb", Calinski and Harabash criterion, Gap statistic and Jump statistic; the latter two in the author-recommended formats. We also take in the Silhouette width statistic, as the most versatile procedure, from (ii) Structural approaches, and two procedures from the (iii) Consensus distribution approach. Table 1 presents the selection of $K*$ estimating methods that participate in our experiments, along with their acronyms used in the remainder.

It is probably worth noting that almost all the methods utilize Euclidean square distance throughout, except for two cases: (a) a version of intelligent K-Means LM is based on Manhattan metric, and (b) the Jump-statistic utilizes Mahalanobis distance within clusters.

The seven methods from the three selected approaches utilize the same format of computations:

they run K-Means at different *K* and then choose "the best" fitting value among the Ks as the estimate of *K**. Thus, we need to specify the range of *K* values for the experiments. Since the data are generated many times for each of the chosen values *K**=9 and *K**=21, and the between-cluster spread factors are large enough to have several of the clusters well separated, we decided, to keep the computations within a reasonable time limit, that the range of tested *K* values should be within an interval of about a dozen with *K** in the middle; thus, the range of tested *K* values is from 4 to 14 at *K**=9 and from 15 to 25 at *K**=21.

As is well known, the clustering and criterion value produced by K-Means much depend on the initialization. In most cases, the user is unable to give any advice on the initial centroids because of insufficient knowledge of the domain. Experimental evidence suggests that the multiple random initialization strategy outperforms other initialization methods in real-world conditions (see Pena et al. 1999, Hand and Krzanowski 2005, Steinley and Brusco 2007). Therefore, at each *K* from the ranges defined above, the Batch K-Means is run *R* times, each time from a random set of entities taken as initial centroids. Of the *R* resulting clusterings, that one minimizing the value of criterion (1) is chosen, and the value of criterion (1) at it is denoted by W_K. We accept *R*=100. This choice is consistent with, first, Hand and Krzanowski (2005) recommended *R*=20 for *N* of the order of 200 in their experiments, and, second, our desire to simulate the constraints of real-world computations.

It should be noted that there have been suggested many improvements over the Straight K-Means version, leading to deeper minima of the criterion (1) for the same initializations, such as the adaptable change of centroids after each entity's Minimum distance assignment (McQueen 1967) or shifting the neighbourhoods (Hansen and Mladenovich 2001) or using simultaneously a population of solutions along with its evolutionary improvements (Maulik and Bandyopadhyay 2000, 2002, Krink and Paterlini 2005). Different distances were explored in Leisch 2006. Modified criteria have been utilized by many (see, for reviews, Steinley 2006 and Bock 2007). These all are left outside of our experiments: only Straight K-Means is being tested.

4.3. Evaluation criteria

Since the generated data is a collection of entities from K* Gaussian clusters, the results of a K-Means run can be evaluated by the quality of recovery of the following components of the generated clusters: (1) the number K*, (2) the cluster centroids, and (3) the clusters themselves. This leads us to using three types of criteria based on comparison of each of these characteristics as produced by the algorithm with those in the generated data. The cluster recovery conventionally is considered of greater importance than the other two.

The recovery of K^* can be evaluated by the difference between K^* and the number of clusters K in the clustering produced with a procedure under consideration. The other two are considered in the subsequent subsections.

4.3.1. Distance between centroids

Measuring the distance between found and generated centroids is not quite straightforward even when $K=K^*$. Some would argue that this should be done based on a one-to-one correspondence between centroids in the two sets, hence the best pair-wise distance matching between two sets. The others may consider that such a matching would not necessarily be suitable because of the asymmetry of the situation – one should care only of how well the generated centroids are reproduced by those found ones, so that if two of the found centroids are close to the same generated centroids, both should be considered its empirical representations. We adhere to the latter view, the more so that this becomes even more relevant, both conceptually and computationally, when K differs from K^*.

Another issue that should be taken into account is of the difference in cluster sizes: should the centroid of a smaller cluster bear the same weight as the centroid of a larger cluster? Or, on the contrary, should the relative cluster sizes be involved so that the smaller clusters less affect the total? To address this issue, we use both weighting schemes in the experiments conducted, to find out which of them is more consistent with cluster recovery than the other.

According to the "asymmetric" perspective above, to score the similarity between the generated centroids, $g_1, g_2, \ldots, g_{K^*}$, and those obtained using one of the chosen algorithms in Table 1, $e_1, e_2, \ldots, e_K$, we utilize a procedure consisting of the following three steps:

(a) pair-wise matching of the obtained centroids to those generated,

(b) calculating distances between matching centroids, and

(c) averaging the distances.

1. *Pair-wise matching centroids:* For each $k=1,\ldots.K^*$, assign g_k with that e_j ($j=1,\ldots,K$) which is the nearest to it. Any not yet assigned centroid e_i then is matched to its nearest g_k.

2. *Computing distances:* Let E_k denotes the set of those e_j that have been assigned to g_k; and $\alpha_{jk} = q_j/|E_k|$, where q_j is the proportion of entities in j-th found cluster (weighted version) or $\alpha_{jk} = 1$ (unweighted version). Define, for each $k=1,\ldots,K,\ dis(k) = \Sigma_{e_j \in Ek} d(g_k,e_j)^* \alpha_{jk}$. The weighted distance is the average weighted distance between the generated and the set of matching centroids in the computed clusters; the unweighted distance is just the summary distance between all matching pairs of clusters. (The distance d here is Euclidean squared distance.)

3. *Averaging distances:* Calculate $D=\sum_{k=1}^{K*} p_k * dis(k)$ where $p_k=N_k= |N_k|$, is the number of entities

in the generated k-th cluster (in the weighted version), or $p_k= 1/K*$ (in the unweighted version).

4.3.2. Confusion between partitions

To measure similarity between two partitions, the contingency (confusion) table of the corresponding partitions of *I* is used. Entries in the contingency table are the co-occurrence frequencies of the generated partition clusters (row categories) and the obtained clusters (column categories): they are the counts of entities that fall simultaneously in both. Denote the generated clusters (rows) by *k*, the obtained partition clusters (columns) by *j* and the co-occurrence counts by N_{kj}. The frequencies of row and column categories (cluster sizes) are denoted by N_{k+} and N_{+j}. The relative frequencies are defined accordingly as $p_{kj}=N_{kj}/N$, $p_{k+}=N_{k+}/N$, and $p_{+j}=N_{+j}/N$, where *N* is the total number of entities. We use a conventional similarity measure, the adjusted Rand index *ARI* defined by the following formula (Hubert and Arabie 1985, Yeung and Ruzzo 2001):

$$ARI = \frac{\sum_{k=1}^{K}\sum_{l=1}^{L}\binom{N_{kl}}{2} - \left[\sum_{k=1}^{K}\binom{N_{k+}}{2}\sum_{l=1}^{L}\binom{N_{+l}}{2}\right]/\binom{N}{2}}{\frac{1}{2}\left[\sum_{k=1}^{K}\binom{N_{k+}}{2}+\sum_{l=1}^{L}\binom{N_{+l}}{2}\right] - \left[\sum_{k=1}^{K}\binom{N_{k+}}{2}\sum_{l=1}^{L}\binom{N_{+l}}{2}\right]/\binom{N}{2}} \tag{10}$$

where $\binom{N}{2}=\frac{N(N-1)}{2}$. The greater the ARI, the more similar are the partitions.

In fact, we have used three more between-partition similarity measures: the relative distance *M*, Tchouproff's coefficient *T*, and the average overlap *A* (Mirkin 2005) The four coefficients capture different structural properties of partitions and do expose different behaviour in our experiments, but regarding our main conclusions they tend to show the same. This is why in the follow-up tables we present only values of *ARI* coefficient.

5. Experimental results

5.1. First series.

The major parameters of the experiment are six spread-shape models that are the result of combining two types of models: (a) either of the three cluster models according to the distribution of the within-cluster spreads and associated shape formats (the spherical shape for the constant spreads, and the elongated NetLib covariance for the variant within-cluster spreads), and (b) either of two modes of the between-cluster spreads, the "large" and "small", according to Table 2.

The experiment is conducted in two instalments. The first instalment is, basically, to see whether our assumptions are right, specifically:

(i)   If one of the two distance formulas, weighted and unweighted, is any better then the other;

(ii)   If the randomness in the generated cluster sizes or initialization of centroids makes a difference, and if it does, what to do about it;

(iii)   Are there any patterns in the recovery of the number of generated clusters K*, that go across the lines of the within- and between-cluster spread models accepted for the experiment? If there are, can they be used for enhancing the clustering procedures?

(iv)   Are there any patterns in the cluster recovery within or across the within- and between-cluster spread models?

The results of the experiments are presented in Table 3, for the case of 9 Gaussian clusters. The entries are averages of the respective evaluation values taken over 20 data sets generated, along with their standard deviations expressed per cent, for the sake of space. The cluster shape, spread and spatial sizes are taken according to Table 2 in section 4.1. In Table 3, we highlight two winners among the nine algorithms under comparison, at each of the six spread patterns (three cluster spread-shape models times two between-cluster spreads), by using the bold font. The two different between-cluster spreads are presented in different columns while the three cluster spread-shape models are reflected within the cells by three rows, as explained in the captions.

With respect to the issues (i)-(iv) raised for this batch of experiments, one can notice the following:

(i)   The orderings of estimates according to the weighted and unweighted distances between centroids differ considerably. The winners with respect to the centroid recovery closely match the winners with respect to the cluster recovery when the unweighted distance is used, and do not match at all, when the weighted distance is used. This goes in line with the view that K-Means clustering results can be interpreted as a form of typology at which centroids represent its so-called intensional, that is, conceptual, part. According to this view, the cluster sizes have nothing to do with their intensions so that the distances should not be weighted by the cluster sizes. The results in the table support this view and make us use only the unweighted distance in the further experiments.

(ii)   The averages reported in Table 3 are rather stable: all the standard deviations lie within 15% of the average values (except for LM and LS at the small between-cluster spread

associated with very high numbers of clusters found – these two will be modified later on). That means that the randomness of the choice of initial centroids and the randomness in cluster sizes do not affect the results that much, and can be considered justifiable.

(iii)   With regard to the number K^* recovery, one can easily notice that the differences in within-cluster shape/spread do not appear to affect the outcomes. However, with respect to between-cluster spread differences, there can be discerned four different patterns: (a) HK consistently chooses K values that are very close to $K^*=9$; (b) LM and LS closely follow $K^*=9$ at the large spread and lead to much larger Ks at the small spread – this especially concerns LM; (c) both CH and JS overestimate K* at the large spread and underestimate it at the small spread, and (d) GS, SW, CD, and DD underestimate K* at both between-cluster spreads, though SW is close at the large spread and DD at the small spread.

(iv)   With respect to the cluster recovery, the common pattern is that the larger spreads lead to better reproduction of clusters than the small spreads for all of the algorithms. The algorithms LS and LM dominate in five of the six spread models, especially at the small between-cluster spread, so that only SW and HK can rival them, at the large between-cluster spreads.

5.2   HK-adjustment of the iK-Means

According to the experiment, iK-Means methods LS and LM may lead to excessive numbers of clusters, while HK, on the other hand, makes a very good recovery of the number of clusters. This leads us to suggest that the HK number-of-cluster results should be taken as a reference to adjust the threshold for removing small AP clusters for the initial setting in iK-Means. So far, only AP singletons are removed from the initial setting. If other "smaller" AP clusters are removed, the chosen K will be smaller and, thus, closer to K^*. A straightforward option would just remove all AP clusters whose sizes are less than or equal to a pre-specified discarding threshold D. Given K_h, found with the Hartigan rule, a suitable discarding threshold D can be found in such a way that the number of clusters K_D identified with D, taken as the discarding threshold, is close enough to K_h. This can be done by gradually increasing D from the default value $D=1$. A typical sequence of steps, at a given K_h, say $K_h=9$, could be like this: at $D=1$, the number of AP clusters is $K_D=32$; at D=2, still $K_D=32$, that is, no doubletons among the AP clusters; then $K_3=29$, $K_4=24$, $K_8=20$, $K_{11}=14$, $K_{12}=11$, and $K_{14}=8$ (the omitted D values give no reduction in K_D values). Therefore, D should be taken as $D=14$. Since K_h value is not necessarily correct but rather indicative, $D=12$,

leading to 11 clusters, is also acceptable, especially if K*=10 or 11. Thus, one can use a computational routine of increasing D one by one until K_D becomes less than θK_h. When we put $\theta=1.1$, the next K_D value is typically less than K_h, whereas $\theta=1.2$ leaves K_D rather large, but $\theta=1.15$ produces reasonable approximations of K_h. We refer to thus HK conditioned versions of LS and LM as ALS and ALM.

5.3 Second experimental series

The second series of our experiments differs from the first one in three aspects: (1) the adjusted versions of iK-Means clustering, ALS and ALM, are included in the list of methods and run on newly generated 10 data sets; (2) the data are generated according to both, not just one, size patterns: (i) N=1000, M=15, K*=9, and (ii) N=3000, M=20, K*=21; (3) only unweighted distance between centroids is used as the measure of centroid recovery.

The issues to be addressed in these experiments, in addition to those (ii)-(iv) above, are as follows:

(v)   Is there any pattern of (dis)similarity between the two data size formats;

(vi)   Are the HK-adjusted iK-Means methods better than the original ones;

(vii)   Are the algorithms' recovery properties at the constant spherical within-cluster-spread model any better than those at the elongated not-constant spread clusters?

The averaged, over ten data sets generated at the greater data size pattern, evaluation criteria values are presented in Table 5. Table 4 presents results reported in Table 3 with the following changes: (a) the weighted distance case is removed; (b) results for ALS and ALM on ten different data sets generated for the former size pattern are added; (c) highlights are changed accordingly; (d) the unweighted distances are rescaled to achieve comparability across the between-cluster spread models, according to Table 2, to address issue (vii). Indeed, at the small within-cluster spreads, the spread factor at k^2-proportional model, 2, is four times greater than that at k-proportional model, 0.5, and 10 times greater than that at the equal spread model, 0.2. By multiplying the distances between centroids at the equal spread model by $100=10^2$ and at the k-proportional model by $16=4^2$, we make them comparable with those at the k^2-proportional model. (Note that the distance between centroids is squared Euclidean, which implies the quadratic adjustment of the factors.) Similarly, at the large spreads, the within-cluster spread factors at the variant spread models are the same while that at the constant spread model is 5 times smaller. Multiplying the distances between centroids at the equal spread model by 25 makes all the distances in the tables comparable.

Here are the findings related to each of the three items above:

(v)   Tables 4 and 5 show a remarkable degree of similarity regarding the main findings of the first series of experiments:

a. The relatively small standard deviations;

b. The same four groupings of the procedures with regard to the number of clusters K* recovery, with the obvious ALS and ALM effects;

c. The same winners over a bulk of the experimental conditions, though HK at K*=21 shows winning performances over some of the conditions too.

(vi)   The HK-adjusted iK-Means methods are not better than the original ik-Means with respect to the cluster recovery; they, however, are better with respect to the number of clusters. It is somewhat surprising that the least-moduli based method LM is on a par with the least-squares based method LS, in spite of the fact that the data are generated according to Gaussian distributions favouring squared distances.

(vii)   Indeed, the algorithms' recovery properties at the equal within-cluster-spread model are better than those at the elongated not-constant spread clusters, if measured by the distances between the generated and original centroids. Moreover, all methods perform better when the cluster spatial sizes are less different: at the constant sizes the best, and at the k^2-proportional sizes the worst. This can be seen as conforming to the idea that K-Means best delivers at the constant spread clusters indeed. The cluster recovery also slightly improves, though not always but rather as a tendency that may fail sometimes (as, for example, for HK at the small between-cluster spreads and for JS and LM at the large between-cluster spreads). However, the effects of differences in within-cluster spread-shape patterns are rather minor.

6. Conclusion

K-Means arguably is the most popular clustering method. This is why studying its properties is of interest not only to the classification, data mining and machine learning communities, but also to the increasing numbers of practitioners in marketing research, bioinformatics, customer management, engineering and other application areas.

This paper addresses one of the most controversial issues in clustering: the right number of clusters, which some may view as baseless because in many cases, "clusters are not in data but in the viewing eye." In our experiments, we try to maintain the case when clusters are in data. The data are generated as sets of entities randomly drawn from Gaussian clusters, with the cluster sizes (proportions) drawn randomly as well. Using Gaussian clusters allows us to address the issue of modelling the cluster overlap in an intuitively appealing way in terms of within- and between-cluster spreads. This also enables us to conduct experiments by confronting two types of situations:

well separated clusters (large between-cluster spread) and overlapping clusters (small between-cluster spread). We combine these with three different models of within-cluster spread and shape. One of the models is of a conventional spherical cluster with a constant variance; the other two involve elongated clusters and different cluster variances. The six combined data settings provide rather different cluster structures. To be closer to the real-world data analyses, we maintain relatively large data sizes (one or three thousand entities) and cluster numbers (9 and 21). Another feature of our data generation approach is that we can utilize the centroid recovery performance of a clustering method in addition to the conventional cluster recovery performance.

The subject of our interest is the intelligent K-Means method, iK-Means (Mirkin 2005) that finds the right number of clusters by sequentially extracting "anomalous patterns", in two versions differing by the distance and centroid definitions. We confront them with seven other procedures that have either experimental support or theoretical meaning. Our experimental results indicate that

(a) Centroids should not be weighted by the cluster sizes in evaluation of the centroid recovery of a clustering method;

(b) There are some consistent patterns across the six spread-shape conditions and different data sizes:

   (b1) the methods are not sensitive to the relative proportions of entities in different clusters;

   (b2) both, the cluster recovery and centroid recovery, are better at the large between-cluster spreads;

   (b3) the centroid recovery of all methods slightly improves when moving from elongated clusters of different variances to spherical clusters of a constant variance; the cluster recovery also typically improves, but the effects are minor;

   (b4) Hartigan's rule "of thumb" HK outperforms the others, in most cases, in terms of the number of clusters, and is good, at the large between-cluster spreads, in terms of cluster recovery; the other methods form consistent patterns of, typically, under-estimating the number of clusters;

   (b5) ik-Means, in most cases, outperforms the others in terms of centroid and data recovery, but overestimates the number of clusters, especially at the small between-cluster spreads, which can be cured by using an HK-adjusted version.

It should be pointed out that these conclusions are drawn on the base of variant between- and within-cluster spread conditions, which have been never analyzed before. This should be taken into account if any of our conclusions are seen as being at odds with others' findings. Overall, the accepted data generation model leads to a set of consistent patterns, whereas models utilised in

some other works frequently lead to inconclusive results (Hardy 1996, Dimitraidou, Dolnicar and Weingessel 2002, Kuncheva and Vetrov 2005).

Further work should include the two approaches to choosing K* that have been not covered in these experiments: those resampling based and those utilizing hierarchical clustering approaches. Some more search-intensive versions of K-Means could be involved, too. A greater coverage of potential data distributions should be considered as well.

References

Banfield, J.D. and Raftery, A.E. (1993) Model-based Gaussian and non-Gaussian clustering, *Biometrics*, 49, 803-821.

Bel Mufti, G, Bertrand, P., and El Moubarki, L. (2005) Determining the number of groups from measures of cluster stability, In: *Proceedings of International Symposium on Applied Stochastic Models and Data Analysi,* 404-412.

Benzecri, J.P. (1992) *Correspondence Analysis Handbook*, New York, Marcel Dekker.

Bock, H.-H. (2007)

Breckenridge, J. (1989) Replicating cluster analysis: method, consistency and validity, *Multivariate Behavioral Research*, 24, 147-161.

Calinski, T. and Harabasz, J. (1974) A dendrite method for cluster analysis, *Communications in Statistics*, 3(1), 1-27.

Casillas, A., Gonzales de Lena, M.T. and Martinez, H. (2003) Document clustering into an unknown number of clusters using a Genetic algorithm, *Text, Speech and Dialogue: 6th International Conference*, 43-49.

Chae, S.S., DuBien, J.L. and Warde, W.D. (2006) A method of predicting the number of clusters using Rand's statistic, *Computational Statistics and Data Analysis*, 50 (12), 3531-3546.

Dimitriadou, E., Dolnicar, S. and Weingassel, A. (2002), An examination of indexes for determining the number of clusters in binary data sets, *Psychometrika*, 67(1), 137-160.

Duda, R.O. and Hart, P.E. (1973) *Pattern Classification and Scene Analysis*, New York, Wiley.

Dudoit, S. and Fridlyand, J. (2002) A prediction-based resampling method for estimating the number of clusters in a dataset, *Genome Biology*, 3(7): research0036.1-0036.21.

Efron B. and Tibshirani R. J. (1993) *An Introduction to the Bootstrap*, Chapman and Hall.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (Eds.) (1996) *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press, Menlo Park, Ca.

Feng, Y. and Hamerly, G. (2006) PG-means: learning the number of clusters in data, *Advances in Neural Information Processing Systems, 19 (NIPS Proceeding)*, MIT Press, 393-400.

Fraley, C. and Raftery, A.E. (2002) Model-based clustering, discriminant analysis, and density estimation, Journal of the American Statistical Association, 97 (458), 611-631.

Generation of Gaussian mixture distributed data (2006), *NETLAB neural network software*, http://www.ncrg.aston.ac.uk/netlab.

Hand, D.J. and Krzhanowski, W.J. (2005) Optimising k-means clustering results with standard software packages, *Computational Statistics and Data Analysis*, 49, 969-973.

Hansen, P. and Mladenovic, N. (2001) J-MEANS: a new local search heuristic for minimum sum of squares clustering, *Pattern Recognition*, 34, 405-413.

Hardy A. (1996), On the number of clusters, *Computational Statistics & Data Analysis* 23, 83-96

Hartigan, J. A. (1975). *Clustering Algorithms*, New York: J. Wiley & Sons.

Hubert, L.J. and Arabie, P. (1985), Comparing partitions, *Journal of Classification*, 2, 193-218.

Hubert, L.J. and Levin, J.R. (1976) A general statistical framework for assessing categorical clustering in free recall, *Psychological Bulletin*, 83, 1072-1080.

Ishioka, T. (2005) An expansion of X-means for automatically determining the optimal number of clusters, *Proceedings of International Conference on Computational Intelligence* (Calgary, Canada), 91-96.

Jain, A.K and Dubes, R.C. (1988). *Algorithms for Clustering Data*, Prentice Hall.

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

Krzanowski W. and Lai Y. (1985), A criterion for determining the number of groups in a dataset using sum of squares clustering, *Biometrics*, 44, 23-34.

Kuncheva, L.I. and Vetrov, D. P. (2005) Evaluation of stability of k-means cluster ensembles with respect to random initialization, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 28, n. 11, 1798-1808.

Leisch, F. (2006) A toolbox for K-centroids cluster analysis, *Computational Statistics and Data Analysis*, 51, 526-544.

Maulik,U., Bandyopadhyay, S. (2000) Genetic algorithm-based clustering technique, *Pattern Recognition*, 33, 1455-1465.

McLachlan, G.J. and Khan, N. (2004) On a resampling approach for tests on the number of clusters with mixture model-based clustering of tissue samples, *Journal of Multivariate Analysis*, 90, 90-1005.

McLachlan, G.J. and Peel, D. (2000), *Finite Mixture Models*, New York: Wiley.

McQueen, J. (1967) Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. II*, pages 281–297.

Milligan, G.W. (1981) A Monte-Carlo study of thirty internal criterion measures for cluster analysis, *Psychometrika*, 46, 187-199.

Milligan, G.W. and Cooper, M.C. (1985), An examination of procedures for determining the number of clusters in a data set, *Psychometrika*, 50, 159-179.

Milligan, G. W. and Cooper, M. C. (1988) A study of standardization of variables in cluster analysis, *Journal of Classification*, 5, 181-204.

Minaei-Bidgoli, B., Topchy, A. and Punch, W.F. (2004) A comparison of resampling methods for clustering ensembles, *International conference on Machine Learning; Models, Technologies and Application (MLMTA04)*, Las Vegas, Nevada, pp. 939-945.

Mirkin, B. (1990) Sequential fitting procedures for linear data aggregation model, *Journal of Classification*, 7, 167-195.

Mirkin, B. (1996) *Mathematical Classification and Clustering*, Kluwer, New York.

Mirkin B. (2005) *Clustering for Data Mining: A Data Recovery Approach*, Boca Raton Fl., Chapman and Hall/CRC, Boca Raton, Fl.

Monti S., Tamayo P., Mesirov J. and, Golub T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data, *Machine Learning*, 52, 91-118.

Mojena, R. (1977) Hierarchical grouping methods and stopping rules: an evaluation, *The Computer Journal*, 20, 359-363.

Murtagh, F. and Raftery, A.E. (1984) Fitting straight lines to point patterns, *Pattern Recognition*, 17, 479-483.

Pelleg, D. and Moore, A. (2000) X-means: extending k-means with efficient estimation of the number of clusters, *Proceedings of 17th International Conference on Machine Learning*, San-Francisco, Morgan Kaufmann, 727-734.

Pena, J. M., Lozano, J. A., Larranaga P., (1999), An empirical comparison of four initialization methods for K-Means algorithm, *Pattern Recognition Letters,* 20(10), 1027-1040.

Pollard, K.S. and van der Laan, M.J. (2002) A method to identify significant clusters in gene expression data, U.C. Berkeley Division of Biostatistics Working Paper Series, 107.

Shen, J., Chang, S.I., Lee, E.S., Deng, Y., Brown, S.J. (2005) Determination of cluster number in clustering microarray data, *Applied Mathematics and Computation*, 169, 1172-1185.

Spaeth, H. (1985) *Cluster Dissection and Analysis*, Ellis Horwood, Chichester.

Steinley, D. (2004) Standardizing variables in K-means clustering. In D. Banks, L. House, F.R. McMorris, P. Arabie and W. Gaul (Eds.) *Classification, Clustering, and Data Mining Applications*, New York, Springer, 53-60.

Steinley, D. (2006), K-means clustering: A half-century synthesis, *British Journal of Mathematical and Statistical Psychology*, 59, 1-34.

Steinley, D., Brusco M., (2007), Initializing K-Means batch clustering: A critical evaluation of several techniques, *Journal of Classification*, 24, pp. 99-121.

Steinley, D. and Henson, R. (2005) OCLUS: An analytic method for generating clusters with known overlap, *Journal of Classification*, 22, 221-250.

Sugar C.A. and James G.M. (2003), Finding the number of clusters in a data set: An information-theoretic approach, *Journal of American Statistical Association*, 98, n. 463, 750-778.

Sun, H., Wang, S. and Jiang, Q. (2004) FCM-based model selection algorithms for determining the number of clusters, *Pattern Recognition*, 37, 2027-2037.

Tibshirani R., Walther G. and Hastie T. (2001), Estimating the number of clusters in a dataset via the Gap statistics, *Journal of the Royal Statistical Society B*, 63, 411-423.

Tipping M.E. and Bishop C.M. (1999) Probabilistic principal component analysis, *Journal of the Royal Statistics Society, Ser*. B 61, 611–622.

Vapnik, V. (2006) *Estimation of Dependences Based on Empirical Data*, Springer Science + Business Media Inc., 2d edition.

Wasito I., Mirkin B. (2006) Nearest neighbours in least-squares data imputation algorithms with different missing patterns, *Computational Statistics & Data Analysis,* 50, 926-949.

Yeung K. Y. and Ruzzo W. L. (2001), Details of the Adjusted Rand index and clustering algorithms, *Bioinformatics*, 17, 763--774.

Method	Acronym
Calinski and Harabasz index	CH
Hartigan rule	HK
Gap statistic	GS
Jump statistic	JS
Silhouette width	SW
Consensus distribution area	CD
Average distance between partitions	DD
Square error iK-Means	LS
Absolute error iK-Means	LM

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

Between-cluster spread	Within-cluster spread type		
	Spherical clusters	PPSA clusters	
	Constant	k-proportional	k^2-proportional
Large	2	10	10
Small	0.2	0.5	2

Table 2. Between-cluster spread factors depending on the within-cluster spread-shape types in the experiments.

	Estimated number of clusters		Weighted distance between centroids		Unweighted distance between centroids		Adjusted Rand Index	
	LaS	SmS	LaS	SmS	LaS	SmS	LaS	SmS
CH	11.55/8	4.00 / 0	53057.85/13	832.87/15	403.85/12	419.27/12	0.82 / 9	0.25/12
	12.10/4	5.30 / 5	1462774.95/11	465599.77/14	11788.38/14*	2932.79/19	0.81 / 8	0.21/12
	11.15/8	4.11 / 8	1560337.21/11	50703.90/12	12146.83/13*	53779.46/15	0.79 / 9	0.22/12
HK	8.27/6	**7.60/10**	47293.32/13	**742.47/13**	412.40/13	386.01/14	0.89 / 9	0.29/10
	8.55/7	**9.40 / 9**	1332058.56/15	409831.54/14	11833.21/14*	2965.56/15	0.90 / 9	0.37/11
	9.35/7	**9.12/10**	1495325.18/14	51941.10/15	12154.99/15	55286.55/14	0.84 / 9	0.28/12
GS	6.25/7	5.75 / 8	47295.85/11	795.52/11	438.33/12	385.25/12	0.77/11	0.28/13
	6.75/8	5.95/10	1305125.52/10	394596.52/11	11758.62/12	2991.15/12	0.77/12	0.28/12
	5.95/8	6.25 / 9	1395568.25/11	51845.25/11	12185.62/13	54258.63/13	0.76/12	0.29/12
JS	12.12/8	4.50 / 0	55417.22/15	798.96/13	403.38/13	419.27/13	0.77/10	0.25/12
	12.75/9	6.15 / 8	1548757.47/12	510687.27/15	11785.21/13*	2908.33/15	0.82 / 8	0.24/13
	12.10/8	4.45 / 5	1570361.91/12	50716.82/12	12131.86/12*	53699.24/14	0.80 / 8	0.22/11
SW	6.29/8	4.54/10	**46046.56/15**	805.30/15	418.26/12	418.66/14	0.92/10	0.26/13
	6.95/8	4.95 / 4	**1299190.70/15**	**393227.66/14**	11876.31/13*	2846.31/16	0.92 / 8	0.27/12
	7.15/8	4.28/11	**1462999.91/12**	50383.53/13	12203.58/12	53583.12/16	**0.85 / 6**	0.22/13
CD	5.31/7	5.11 / 9	**47122.13/14**	791.76/12	429.96/12	**373.93/12**	0.78/12	0.27/13
	5.30/6	5.10/10	**1305051.80/14**	394572.84/13	11943.98/13	2897.61/18	0.78/12	0.28/14
	5.20/6	5.31 / 9	**1350841.29/13**	51968.86/12	12265.98/12	55040.86/15	0.75/12	0.25/13
DD	5.67/3	**6.42 / 8**	47190.83/15	792.15/15	435.37/12	409.97/13	0.75/12	0.27/12
	4.90/3	5.60 / 9	1306014.88/13	395524.66/12	11979.30/13	2996.28/18	0.74/12	0.24/12
	5.30/3	**5.83 / 8**	1394892.59/14	50813.28/15	12286.43/12	53912.13/13	0.71/12	0.27/10
LS	**8.67/6**	13.00/18	49095.21/15	1110.88/13	**402.47/12**	**335.91/23**	**0.99 / 9**	**0.48/12**
	8.80/6	**10.80/16**	1485719.73/12	486979.24/14	**11771.70/12**	**2661.41/20**	**0.99/10**	**0.42/12**
	7.95/7	13.44/18	1444645.99/15	51226.10/12	**12031.13/11**	54026.92/15	**0.90 / 9**	**0.45/12**
LM	**9.33/6**	25.00/18	54478.33/13	**705.61/15**	**400.18/12**	381.12/25	**0.92 / 9**	**0.38/12**
	8.80/7	16.10/17	1487335.77/13	487940.63/13	**11767.34/13**	**2648.60/20**	**0.99/10**	**0.41/12**
	10.00/6	23.11/18	2092537.57/12	**50506.80/12**	12114.01/12	**53507.21/16**	0.84/10	**0.41/12**

* within 1% of the best value

Table 3. The average values of evaluation criteria at 9-clusters data sets with NetLab Gaussian covariance matrix for the large and small between-cluster spreads (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 spread-shape models: the spherical on top, the PPCA with k-proportional cluster sizes in the middle, and the PPCA with k^2-proportional cluster sizes in the bottom. Two winners among the nine methods are highlighted using the bold font, for each of the options.

	Estimated number of clusters		Adjusted distance between centroids		Adjusted Rand Index	
	LaS	SmS	LaS	SmS	LaS	SmS
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
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
GS	6.25/7	5.75 / 8	10958.25/12	38526.96/12	0.77/11	0.28/13
	6.75/8	5.95/10	11758.62/12	47857.52/12	0.77/12	0.28/12
	5.95/8	6.25 / 9	12185.62/13	54258.63/13	0.76/12	0.29/12
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
	5.30/3	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

Table 4. The average values of evaluation criteria at 9-clusters data sets with NetLab Gaussian covariance matrix for the large and small between-cluster spreads (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 spread-shape models: the spherical on top, the PPCA with k-proportional cluster sizes in the middle, and the PPCA with k^2-proportional cluster sizes in the bottom. Two winners among the eleven methods (nine 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 5.2.

	Estimated number of clusters		Adjusted distance between centroids		Adjusted Rand Index	
	LaS	SmS	LaS	SmS	LaS	SmS
CH	24.56/9	15.00 / 0	14598.62/10*	24583.26/10	0.79 / 8	0.24/11
	24.96/8	17.35 / 9	18789.25/11*	37856.25/19	0.80 / 8	0.23/12
	23.45/9	16.45 / 9	20984.62/10	55145.89/12	0.82/10	0.21/12
HK	20.45/8	18.50 / 8	14378.52/11*	26164.25/13	0.90 / 9	0.21/11
	21.12/8	**21.45/10**	18457.52/10*	38152.52/15	0.89 / 8	**0.45/12**
	21.52/7	**21.12/11**	**20761.95/10**	59254.56/11	0.87/10	0.35/12
GS	18.32/9	15.32/10	15489.65/10	24354.25/11	0.81/11	0.25/11
	17.56/9	16.52/11	21278.32/11	37524.21/11	0.77/10	0.27/11
	17.52/8	18.32/10	21859.32/11	55328.45/11	0.79/11	0.26/10
JS	25.58/7	15.00 / 0	14478.96/12*	24583.26/10	0.76/10	0.24/11
	24.65/8	18.75 / 9	18546.32/11*	37526.25/15	0.82 / 8	0.25/13
	25.25/7	15.85 / 7	21254.74/10	56254.85/14	0.81 / 9	0.25/11
SW	19.35/8	17.50/10	15895.52/11	22267.25/12*	0.93/10	0.26/12
	18.35/8	16.85 / 7	21587.85/10	37859.26/16	0.91 / 8	0.26/12
	18.52/8	17.38 / 7	22459.45/12	56859.25/16	0.85 / 8	0.23/13
CD	17.52/9	17.00 / 0	15254.95/11	27154.26/12	0.79/12	0.30/11
	18.52/7	17.25 / 8	21148.52/11	37152.56/18	0.78/11	0.27/14
	16.45/9	18.52 / 9	22984.52/11	55492.17/15	0.77/10	0.29/13
DD	17.84/9	17.25 / 8	15269.52/11	26458.25/10	0.79/12	0.35/11
	15.95/8	16.52 / 9	20365.14/11	38185.54/18	0.76/10	0.25/12
	17.52/9	17.25 / 9	21523.65/11	56874.82/13	0.76/12	0.27/10
LS	**20.85/7**	25.85 / 8	**14254.85/11**	26954.23/12	**0.99 / 9**	0.36/10
	20.75/7	27.65 / 9	**18254.65/11**	**31459.25/18**	**0.99/10**	0.40/12*
	18.96/7	30.45 / 9	22351.85/11	53462.52/15	0.90/10	**0.42/12**
LM	21.56/8	37.45/18	15254.85/11	24586.23/12	0.96 / 9*	0.40/11
	20.65/9	36.25/16	18754.25/11	**29025.52/17**	**0.99/10**	0.42/12*
	22.45/9	38.12/17	22145.88/11	52854.21/16	0.86/10	**0.40/12**
ALS	20.32/8	**19.85 / 6**	**14358.95/11**	**22145.85/11**	**0.99/11**	**0.50/11**
	21.85/8	**21.87 / 8**	19658.52/11	38452.95/10	**0.99/11**	**0.45/12**
	21.42/8	**21.85 / 9**	**20542.65/11**	51954.65/12	0.97/11	0.39/12*
ALM	**21.25/9**	**22.52 / 6**	15254.95/11	**21856.32/12**	**0.99/12**	**0.45/11**
	20.12/8	23.45 / 9	**18236.12/11**	37529.52/10	**0.99/10**	0.38/11
	21.85/7*	21.45 / 8*	22956.25/11	**52018.85/11**	**0.95/13**	0.38 / 9*

* within 1% of the best value

Table 5. The average values of evaluation criteria at 21-cluster data sets with NetLab Gaussian covariance matrix for the large and small between-cluster spreads (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 spread-shape models: the spherical on top, the PPCA with k-proportional cluster sizes in the middle, and the PPCA with k^2-proportional cluster sizes in the bottom. Two winners of the eleven methods (nine from Table 1 plus adjusted LS and LM denoted as 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 5.2.