BIOINFORMATICS ORIGINAL PAPER

Vol. 26 no. 22 2010, pages 2841–2848
doi:10.1093/bioinformatics/btq534

Gene expression

Advance Access publication September 29, 2010

A simple and fast method to determine the parameters for fuzzy c–means cluster analysis

Veit Schwämmle* and Ole Nørregaard Jensen

Department of Biochemistry and Molecular Biology, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark

Associate Editor: John Quackenbush

ABSTRACT

Motivation: Fuzzy c-means clustering is widely used to identify cluster structures in high-dimensional datasets, such as those obtained in DNA microarray and quantitative proteomics experiments. One of its main limitations is the lack of a computationally fast method to set optimal values of algorithm parameters. Wrong parameter values may either lead to the inclusion of purely random fluctuations in the results or ignore potentially important data. The optimal solution has parameter values for which the clustering does not yield any results for a purely random dataset but which detects cluster formation with maximum resolution on the edge of randomness.

Results: Estimation of the optimal parameter values is achieved by evaluation of the results of the clustering procedure applied to randomized datasets. In this case, the optimal value of the fuzzifier follows common rules that depend only on the main properties of the dataset. Taking the dimension of the set and the number of objects as input values instead of evaluating the entire dataset allows us to propose a functional relationship determining the fuzzifier directly. This result speaks strongly against using a predefined fuzzifier as typically done in many previous studies. Validation indices are generally used for the estimation of the optimal number of clusters. A comparison shows that the minimum distance between the centroids provides results that are at least equivalent or better than those obtained by other computationally more expensive indices.

Contact: veits@bmb.sdu.dk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on April 7, 2010; revised on September 6, 2010; accepted on September 15, 2010

1 INTRODUCTION

New experimental techniques and protocols allow experiments with high resolution and thus lead to the production of large amounts of data. In turn, these datasets demand effective machine-learning techniques for extraction of information. Among them, the recognition of patterns in noisy data still remains a challenge. The aim is to merge the outstanding ability of the human brain to detect patterns in extremely noisy data with the power of computer-based automation. Cluster analysis allows to group high-dimensional data points that exhibit similar properties and so to discover a possible functional relationship within subsets of data.

Nowadays, cluster analysis is in widespread use for the analysis of microarray data to discover genes with similar expression changes. Recently, large datasets from quantitative proteomics, for instance measuring the peptide/protein expression by means of mass spectrometry, became available. For this data, cluster analysis is a powerful tool to identify or to confirm pathways of interacting proteins.

Different approaches to the problem of cluster analysis exist, such as hierarchical clustering (Eisen *et al.*, 1998), *k*-means clustering (Tavazoie *et al.*, 1999) and self-organizing maps (Tamayo *et al.*, 1999). Noise or background signals in collected data normally come from different sources, such as intrinsic noise from variation within the sample and noise coming from the experimental equipment. An appropriate method to find clusters in this kind of data is based on fuzzy c-means clustering (Bezdek, 1981; Dunn, 1973) due to its robustness to noise (Hanai *et al.*, 2006). Although this method has been modified and extended many times [for an overview see Döring *et al.* (2006)], the original procedure (Bezdek, 1981) remains the most commonly used to date.

In contrast to *k*-means clustering, the fuzzy c-means procedure involves an additional parameter, generally called the fuzzifier. A data point (e.g. a gene or protein, from now on called an object) is not directly assigned to a cluster but is allowed to obtain *fuzzy* memberships to all clusters. This makes it possible to decrease the effect of data objects that do not belong to one particular cluster, for example objects located between overlapping clusters or objects resulting from background noise. These objects, by having rather distributed membership values, now have a low influence in the calculation of the cluster center positions. Hence, with the introduction of this new parameter, the cluster analysis becomes much more efficient in dealing with noisy data. The value of the fuzzifier defines the maximum fuzziness or noise in the dataset. Whereas the *k*-means clustering procedure always finds clusters independently on the extent of noise in the data, the fuzzy method allows first to adapt the method to the present amount of noise and second to avoid erroneous detection of clusters generated by random patterns. Therefore, the challenge consists in determining an appropriate value of the fuzzifier.

To our knowledge, only few methods exist to determine an optimal value of the fuzzifier. In Dembélé and Kastner (2003), the fuzzifier is obtained with an empirical method calculating the coefficient of variation of a function of the distances between all objects of the entire dataset. Another approach searches for a minimal

*To whom correspondence should be addressed.

fuzzifier value for which the cluster analysis of the randomized dataset produces no meaningful results, by comparing a modified partition coefficient for different values of both parameters (Futschik and Carlisle, 2005). The calculations in these two methods imply operations on the entire dataset and become computationally expensive for large datasets.

Here, we introduce a method to determine the value of the fuzzifier without using the current dataset. Our study also shows that the optimal fuzzifier generally takes values far from the frequently used value 2. The present method can be applied to any dataset for which one wants to detect clusters of non-random origin. Its advantages are to provide an optimal parameter set and to save computational time when processing large datasets.

In the following section, the algorithm of fuzzy c-means clustering is introduced and the concept to avoid random cluster detections is explained. We present a simplified model showing a strong dependence of the fuzziness on the main properties of the dataset and confirm this result by evaluating randomized artificial datasets. We distinguish between valid and invalid cluster analysis by looking at the minimal distances between the found centroids. This relationship is quantified by fitting a mathematical function to the results for the minimum centroid distance. Finally, we determine the second parameter of the cluster analysis, the number of clusters. Different validation indices are compared for artificial and real datasets.

2 DATASET AND ALGORITHM

Clustering algorithms are often used to analyze a large number of objects, for example genes in microarray data, each containing a number of values obtained at different experimental conditions. In other terms, the dataset consists of N object vectors of D dimensions (experimental conditions), and thus an optimal framework contains $N\times D$ experimental values. The aim is to group these objects into clusters with similar behaviors.

In gene expression data and in quantitative proteomics data, the values of each object represent only a relative quantity to be compared with the other values of the object. Therefore, the focus is on fold changes and not on absolute value changes (a 2-fold, i.e. 200%, increase has the same weight as a 2-fold decrease, 50%). In this case, the values are transformed by taking their logarithm before the data are to be evaluated.

In fuzzy c-means clustering (Bezdek, 1981), the data are prepared by normalization of each object to have values with mean 0 and SD 1. Then, for a given parameter set c,m — the number of clusters and the fuzzifier — the clustering corresponds to minimizing the objective function,

$$J(c,m)=\sum_{k=1}^{c}\sum_{i=1}^{N}(u_{ki})^m|\mathbf{x_i}-\mathbf{c_k}|^2 \ , \tag{1}$$

where we used Euclidean metrics for the distances between centroids $\mathbf{c_k}$ and objects $\mathbf{x_i}$. Here, u_{ki} denotes the membership value of object i to the cluster k, satisfying the following criteria,

$$\sum_{k=1}^{c}u_{ki}=1 \ ; \ 0\leq u_{ki}\leq 1 \ . \tag{2}$$

The following iteration scheme allows the calculation of the centroids and the membership values by solving

$$\mathbf{c}_k=\frac{\sum_{i=1}^{N}(u_{ki})^m\mathbf{x}_i}{\sum_{i=1}^{N}(u_{ki})^m} \tag{3}$$

for all k and afterwards obtaining the membership values through

$$u_{ki}=\frac{1}{\sum_{s=1}^{c}\left(\frac{|\mathbf{x}_i-\mathbf{c}_k|^2}{|\mathbf{x}_i-\mathbf{c}_s|^2}\right)^{\frac{1}{m-1}}} \ . \tag{4}$$

A large fuzzifier value suppresses outliers in datasets, i.e. the larger m, the more clusters share their objects and vice versa. At the limit $m\to 1$, the method becomes equivalent to k-means clustering, whereas for $m\to\infty$ all data objects have identical membership to each cluster.

Usually, the value of the fuzzifier is set equal to 2 (Babuska, 1998; Höppner *et al.*, 1999; Pal and Bezdek, 1995). This may be considered a compromise between an a priori assumption of a certain amount of fuzziness in the dataset and the advantage of avoiding a time consuming calculation of its value. However, by carefully adjusting the fuzzifier, it should be possible to optimize the algorithm to take into account the characteristic noise present in the dataset. We are interested in having maximal sensitivity to observe barely detectable cluster structures combined with a low probability of assigning clusters originating from random fluctuations.

We minimize the objective function $J(c,m)$ by carrying out 100 iterations of Equations (3 and 4). The application of Equations (3–4) converges to a solution that might be trapped in a local minimum, requiring the user to repeat the minimization procedure several times with different initial conditions. In order to be able to carry out a vast parameter study, we limited the evaluation to 5–10 performances per dataset and parameter set, taking the performance corresponding to the best clustering result, i.e. the one with the smallest final value of the objective function.

The final classification of a dataset into different clusters in fuzzy clustering is not as clear as in the case of k-means clustering where each object is assigned to exactly one cluster. In fuzzy c-means clustering, each object belongs to each cluster, to the degree given by the membership value. The centroid, i.e. the center of a cluster, corresponds therefore to the center of all objects of the dataset, each contributing with its own membership value. As a consequence, we need to define a threshold that defines whether an object belongs to a certain cluster. Ideally, this threshold is set to $1/2$. Hence, due to the limitation of Equation (2), each object belongs to maximally one cluster.

A cluster with at least one object having a membership value greater than $1/2$ is called a *non-empty* cluster. The number of non-empty clusters c_{final} found in the cluster analysis can be smaller than the number of previously defined clusters, c. Therefore, we can define the case $c_{\text{final}}<c$ to be a case of *no solution* for the application of the cluster analysis. In other words, a cluster analysis leading to at least one empty cluster will not be considered as a valid performance.

By distinguishing cases for which the cluster analysis gives a valid result and cases of invalid results, it is possible to identify parameter regions where the algorithm identifies clusters that may result from random fluctuations. As example, take a dataset and its randomized counterpart. We now fix c and compare the results of the clustering for increasing fuzzifier values, m. At $m \rightarrow 1$, the cluster analysis is equivalent to k-means clustering, assigning exactly one cluster to each object and the no-solution case does not exist. The clustering of both the original and the randomized dataset will give c valid clusters. By increasing the value of the fuzzifier, the membership values of outliers become more distributed between the clusters whereas objects pertaining to real clusters get their largest membership value decreased only slightly. Each cluster looses object members with membership values larger than $1/2$ and the total number of objects that are assigned to a cluster as hard members decreases. As the objects of a randomized dataset are distributed almost homogeneously in cluster space, the clustering algorithm stops to detect a total of c non-empty clusters above a certain threshold of the fuzzifier. When further increasing m, also the objects in the original dataset will have their largest membership values fall below $1/2$ and so the clustering of the original data will stop to produce valid results above another threshold of m. The parameter region between these two thresholds is of particular interest. Within this region, only the clustering of the original dataset produces valid results and thus the found clusters can be understood to correspond to non-random object groupings. Precisely, we prefer to take an as low as possible value of the fuzzifier, combining minimal fuzziness and maximal cluster recognition. The procedure presented in the next sections shows how to obtain a minimal value of m that still does not give a valid solution for the clustering of the randomized dataset. A dataset having the same threshold for both the clustering of the original set and the randomized one should be discarded as it is too noisy. However, we will see that the value of the fuzziness increases strongly for low-dimensional datasets and thus a compromise between accepting clusters with members of noisy origin and low detection of patterns must be found.

3 ARGUMENTS FOR A FUNCTIONAL RELATIONSHIP BETWEEN THE FUZZIFIER AND THE DATASET STRUCTURE

A strong relationship between the fuzzifier and the basic properties of the dataset can be demonstrated by means of a simplified model system. With increasing dimension, clusters are less likely to be found in a completely random dataset. In order to illustrate this dependency mathematically, one might reduce the system to a binary D-dimensional object space, i.e. $x_{id} \in \{-1, 1\}$. Let us now look at a cluster that contains an accumulation of objects at a given point in object space. For example, for a purely random object, the probability to have $\mathbf{x}_i = (1, 1, ..., 1)$ is given by 2^{-D}. Furthermore, the probability to have half or more of all objects of the dataset with this particular value, for $2^{-D} \ll 1$, might be approximated by $2^{N(1-\frac{D}{2})}\sqrt{2/(\pi N)}$ (for details see Supplementary Materials). Hence, the probability for a well-defined cluster decreases exponentially with respect to the dimension of the dataset, and slightly slower for an increasing number of objects in the set. As a consequence, the clustering parameter value m being a measure for the fuzziness of the system should follow these tendencies at least qualitatively. This finding argues strongly against an application of the fuzzy algorithm by merely using $m = 2$. We will show that the simplified model predicts the dependencies on both quantities in the right way.

Table 1. Summary of the parameters

Parameters of the clustering	Parameters of the artificial dataset
m: fuzzifier	N: number of objects
	D: number of dimensions of an object
c: number of clusters	M: number of Gaussian-distributed clusters
	N_O: number of data points per cluster
	w: SD of Gaussian

An extensive evaluation of the clustering procedure is carried out using artificially generated datasets as input. Each object corresponds to a random point generated out of D-dimensional Gaussian distributions with SD w. The dataset consists of M Gaussian-distributed clusters with each having N_O objects, leading to a total of $N = N_O \times M$ objects in the set. Each Gaussian is centered at a random position in object space, having coordinates between 0 and 10 for each dimension. An optimal cluster analysis should identify $c = M$ as best solution. The parameters of the fuzzy c-means algorithm and the parameters of the artificial dataset are summarized in Table 1.

A first step to find an optimal value of the fuzzifier consists in applying the clustering procedure to randomized datasets. We generate these sets by random permutations of the values of each object. A threshold for the fuzzifier value m is reached as soon as the clustering procedure does not provide any valid solution for the randomized set. This corresponds to the case where the number of non-empty clusters is smaller than the value of the parameter c. However, another criterion allows a more accurate estimation. We will refuse a clustering solution having two centroids that coincide, i.e. their mutual distance falls below some predefined value.

Figure 1 shows both the number of non-empty clusters as well as the minimum centroid distance for different realizations of artificial datasets. There is a sudden decay to zero of both quantities when increasing the fuzzifier. Three important conclusions can be made from the results depicted in Figure 1: first, the position of the decay of the minimal centroid distances coincides with the one where the number of non-empty clusters changes from c to $c-1$. Apparently, a cluster without any membership values over the $1/2$ limit (an empty cluster) has always its centroid coincide with the centroid of one of the non-empty clusters. We could not find any mathematical explanation for this behavior, but our analysis shows that this relation seems to be a general characteristics of the fuzzy c-means algorithm. Taking a threshold other than $1/2$ would break this equivalence. By using the minimum centroid distance as main criterion, the results now are independent of this threshold. Secondly, the minimum centroid distance decay occurs at almost exactly the same value of m over the entire range of c. This seems to be a typical behavior in randomized datasets. Thirdly, the m-position of the decay decreases for higher dimensions of the dataset. Datasets of higher dimension have a structure where random clusters are less likely as already illustrated with the simplified model presented above. We will take the minimum centroid distance to measure the m-value of the threshold in the following analysis, which is from now on denoted m_t.

Figure 2 compares the minimum centroid distance for differently distributed datasets, each randomized before applying the cluster analysis. The picture remains mainly the same, with exception of the case $M=1$, where the threshold m_t lies at a slightly higher value. The reason is that the threshold still varies within some range for randomized datasets of equal dimension and number of objects. The magnitude of this variation decreases for datasets of higher dimension.

Despite the normalization of each object to have SD 1 and mean 0, a strong bias of the values toward certain dimensions may occur. This bias leads to different results for the clustering of the randomized dataset. By processing different datasets with the same parameters but different positions of the artificial Gaussian-distributed objects' center, we try to capture the effects of both symmetric as well as biased datasets. The case $M=1$ in Figure 2 corresponds to the clustering results of strongly biased data. The bias becomes large the more the center of the Gaussian deviates from the origin of the coordinate system. For $M>1$, this bias becomes smoothed out by randomization and therefore m_t varies much less. For example, a biased dataset would be gene expression data where most of the genes are upregulated at one of the experimental stages (dimensions). A more detailed description of the effect of biased datasets can be found in the Supplementary Materials.

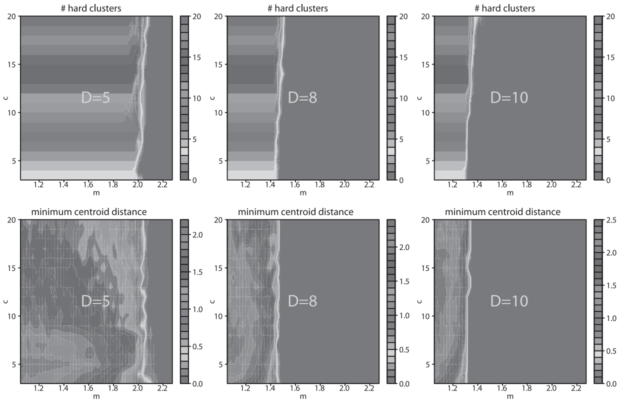

Fig. 1. Showing the number of non-empty clusters and the minimum centroid distances of randomized datasets for different values of m (horizontal axis) and c (vertical axis). Each panel shows these quantities calculated over the given range of m and c for a randomized dataset. The picture is nearly identical for different datasets with the same properties. The object points are Gaussian distributed with SD $w=1$, dimensions 5, 8 and 10, and were randomized afterwards. There are 500 objects per dataset. The threshold of m where the number of non-empty clusters becomes smaller than c and the minimum centroid distance approaches zero does not vary significantly for different c. Moreover, the m-position of the threshold is the same for both measures within the same dimension.

No changes of the minimum centroid distance landscape were found for datasets with different data ranges or datasets with a varying amount of cluster members inside the same set (see Supplementary Materials).

The analysis of the simplified model showed also a dependency of the fuzziness in the dataset on the number of objects, N, although weaker than the one on the dimension of the dataset. Figure 3 confirms this result, showing that m_t increases for smaller N and saturates at a certain level for large $N \gtrsim 1000$.

4 ESTIMATING THE OPTIMAL VALUE OF THE FUZZIFIER

We now focus on the estimation of the dependency of the threshold on both N and D, i.e. we neglect the effect of biased datasets. This threshold will then be taken as the optimal value. A rule of thumb for the maximum number of clusters in a dataset is that it does not exceed the square root of the number of objects (Zadeh, 1965). As the threshold of the minimum of centroid distances m_t does not vary with c, we determine the threshold in the following analysis by carrying out cluster analysis with different m for $c=\sqrt{N}$. Precisely, the threshold m_t corresponds to the value of the fuzzifier at which the minimum centroid distance falls <0.1 for the first time. For small dimensions and object numbers and $m>m_t$, the minimum centroid distance may vary to values slightly larger than 0. A threshold of 0.1 allows to include even extreme cases like $D=4$ and $N=50$. However, a threshold of 0.01 would already be sufficient for $N \geq 100$ and $D \geq 5$. Note, that we hereby exclude the situation that the centroids of two clusters locate at mutually small distances of less than 0.1. However, this limitation did not affect the results.

The clustering is carried out 5–10 times, each analysis for a different randomized artificial dataset having the same parameters. From these different runs we take the largest value of m_t.

The usage of $m=m_t$ in the cluster analysis of the original dataset has two advantages. First, a dataset lacking non-random clusters does not provide any reasonable results, i.e. the number of detected non-empty clusters is lower than the parameter c. This means that the value of the minimum centroid distance is around 0 for all c. Secondly, this smallest allowed value of m guarantees an optimal estimation of a maximal number of clusters which is in general better than for larger m and so still ensures the recognition of barely detectable clusters. The dependency of m_t on the dimension of the dataset is shown in Figure 4a and compared with the values calculated by the method introduced in Dembélé and Kastner (2003). The curves from the latter method exhibit the same tendency but an overestimation of the fuzzifier.

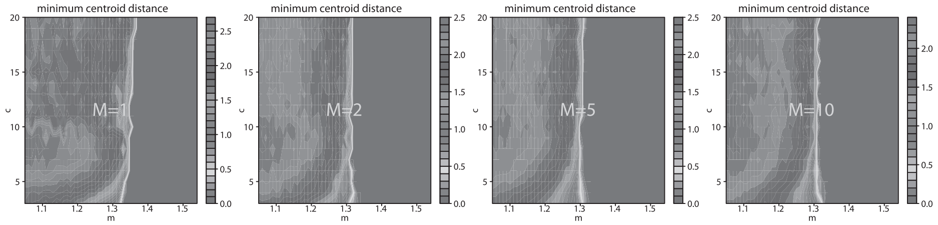

Fig. 2. Landscape of the minimum centroid distance for randomized data sets with Gaussian-distributed clusters. The numbers of previously set different clusters are $M=1,2,5$ and 10. The data sets have the same total number of objects, $N=1000$, the dimension of the sets is 10 and we have $w=1$. No significant differences can be observed except for the panel with $M=1$ where the threshold m_t seems to have a slightly larger value.

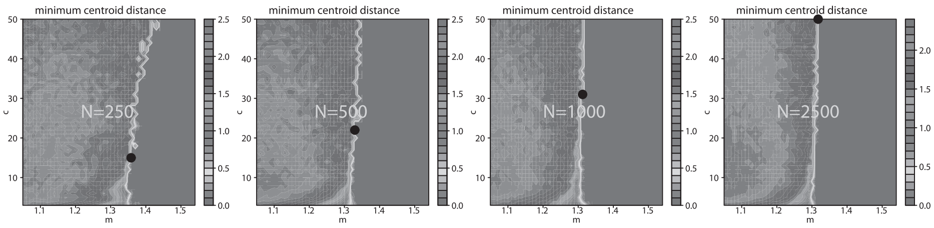

Fig. 3. Landscape of the minimum centroid distance from randomized datasets with different numbers of objects, $N = 250, 500, 1000, 2500$. The threshold m_t decreases for increasing N and seems to saturate for very large numbers. We took $D = 10$, $w = 1$ and $M = 5$. The black points indicate the position where we take the fuzzifier threshold m_t.

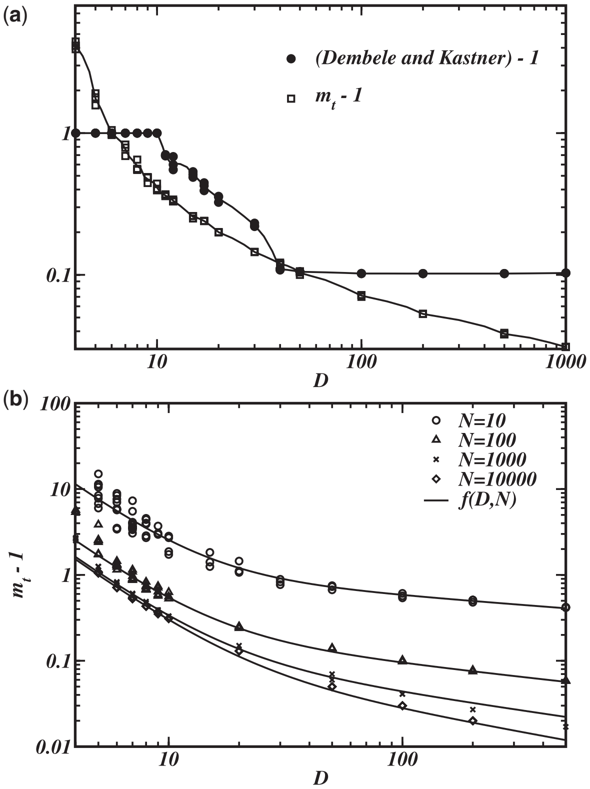

Fig. 4. **(a)** A comparison of our method for the estimation of the fuzzifier to the one presented in Dembélé and Kastner (2003) shows that the value of m is mostly overestimated in the latter. In addition, our method allows to cope with a larger dimensional range. **(b)** Comparing the threshold of the minimal centroid distance for randomized datasets with different numbers of objects, N. The threshold increases for larger N and the curve seems to approach a limiting shape for very large N. Fluctuations become large for $D < 10$. The lines show the values of the fitting function, Equation (5).

A thorough analysis, calculating m_t for randomized datasets of different dimensions and object numbers shows a general functional relation between m_t and the dataset properties. The following function provides a good fit of the curves for all combinations of N and D,

$$f(D,N) = 1 + \left(\frac{1418}{N} + 22.05\right) D^{-2} + \left(\frac{12.33}{N} + 0.243\right) D^{-0.0406 \ln(N) - 0.1134}. \quad (5)$$

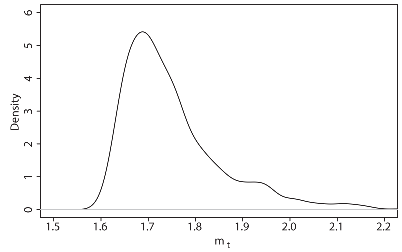

Fig. 5. The density distribution of threshold values for different implementations of randomized artificial datasets with $M = 1$, $D = 7$, $w = 0.1$ and $N = 200$.

Both the data points of m_t and their empirical fit with Equation (5) are depicted in Figure 4b. The prediction with the empirical formula improves for large N and large D. For smaller values of these input values, the m_t obtained from the artificial sets may deviate from the predicted value due to their dependency on the individual dataset.

We calculated the density distribution of m_t for artificial sets with the same parameters, setting $M = 1$, $D = 7$ and $N = 200$ (Fig. 5). The corresponding prediction for m_t is given by $f(7,200) = 1.75$. The only difference between the datasets consists in the position of the mean of the Gaussian, and thus the bias of the data. The maximum of the distribution lies at a slightly smaller value than the one predicted in Equation (5). The figure shows also that the lower limit of m_t is rather well defined, whereas high values are possible, even far away from the maximum. Consequently, for datasets with small N and D, Equation (5) may be more useful for the estimation of the lower limit of m_t. However, the prediction works much better for larger values of D and N.

Equation (5) accounts also for randomized real datasets where the distribution within a cluster may be non-Gaussian. For the analysis, we tested datasets from different origin including biological data from protein research (Horton and Nakai, 1996; Olsen *et al.*, 2006; Pierce *et al.*, 2008; Wolf-Yadlin *et al.*, 2007), microarray data (Cho *et al.*, 1998; Iyer *et al.*, 1999; Tavazoie *et al.*, 1999) and data gathered from non-biological experiments (Nash *et al.*, 1994; Sigillito *et al.*, 1989).

Table 2 compares the minimum centroid threshold calculated from the randomized datasets to the empirical value obtained from Equation (5). We find a deviation for the iTRAQ3 dataset having a small $D = 7$ and $N = 222$. From Figure 5, we see that the higher value of $m_t = 1.81$ is still within the range of the distribution. Note, that the optimal fuzzifier value for the yeast2 dataset was estimated to be $m = 1.15$ in Futschik and Carlisle (2005), identical with our estimation.

5 DETERMINING THE NUMBER OF CLUSTERS

After calculating the optimal value of the fuzzifier by either using Equation (5) or determining m_t directly as done above, the final step consists in estimating the number of clusters in the dataset. Various validity indices for the quality of the clustering are present in the literature. They in general are a function of the membership values, the centroid coordinates and the dataset. The results for the indices summarized in Table 3 will be compared for artificial and real datasets.

First we take another look on the minimum centroid distance, V_{MCD}, now taken from the cluster analysis of artificial (not randomized) datasets (Fig. 6). The panels show V_{MCD} for datasets with 10 Gaussian-distributed clusters, each panel for a set of Gaussians with different SDs. For datasets with clearly separated clusters (small SDs), the picture is completely different to the one of a randomized dataset (Figs 1–3). A strong decay, this time not necessarily to zero, occurs at $c=c_t$ independent of the value of the used fuzzifier m. Note that in the randomized case the decay was at m_t for all c. The position of the sudden decrease coincides with the number of clusters $M=10$ of the artificial dataset, and thus the minimum centroid distance provides a reasonable measure also to determine the optimal number of clusters. For more mixed clusters, the landscape transforms gradually into the picture observed for randomized sets.

The parameter landscapes of real datasets will exhibit a combination of two extremes, a plateau below the threshold c_t for a dataset with clearly distinguishable cluster and a plateau below m_t for a completely noisy dataset. We can also observe that the number of found clusters decreases with increasing m (cases $w=2$, $w=3$ and $w=4$ in Fig. 6) as would be expected.

Equation (5) gives $m_t=1.47$ for the parameters of the artificial datasets in Figure 6. The figure shows that some of the clusters may be recognized even for $w=3$ and $w=4$, when using $m=m_t$ for the

Table 2. Comparing estimated values of m_t to their predictions from Equation (5) and estimating the optimal cluster number with the minimum centroid distance

Dataset	D	N	m_t	$f(D,N)$	c_{opt}
PhosPept (Olsen *et al.*, 2006)	5	1050	2.15	2.07	4
iTRAQ1, Table 1 (Pierce *et al.*, 2008)	7	1775	1.6	1.58	9
iTRAQ2, Table 2 (Pierce *et al.*, 2008)	7	829	1.56	1.59	13
iTRAQ3, T. 4 (Wolf-Yadlin *et al.*, 2007)	7	222	1.81	1.74	2
Ecoli (Horton and Nakai, 1996)	7	335	1.64	1.68	5
Abalone (Nash *et al.*, 1994)	8	4174	1.41	1.45	2
Serum (Iyer *et al.*, 1999)	13	517	1.27	1.25	5
Yeast1 (Tavazoie *et al.*, 1999)	16	2885	1.18	1.16	6
Yeast2 (Cho *et al.*, 1998)	17	2951	1.17	1.15	10
Ionosphere (Sigillito *et al.*, 1989)	34	351	1.13	1.1	4

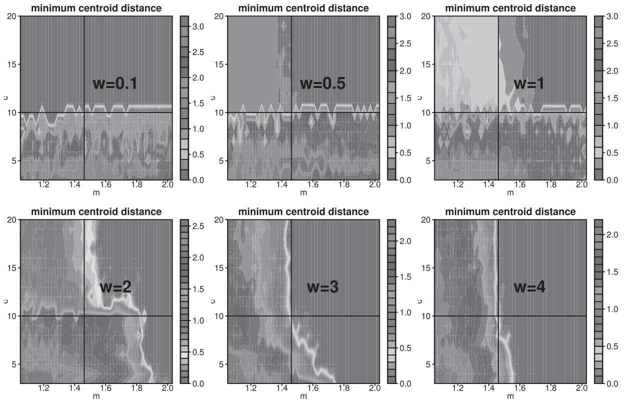

Fig. 6. Landscape of the minimum centroid distances for a set of 10 clusters with each having 100 objects of $D=8$. The clusters were produced to have a Gaussian distribution with different SDs, being $w=0.1, 0.5, 1$ (upper panels left, middle and right), and $w=2, 3, 4$ (lower panels left, middle and right). The black lines indicate m_t and c_t.

Table 3. Summary of the validation indices

Partition coefficient (Bezdek, 1975)	$V_{PC}=\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^2$
Modified partition coefficient (Dave, 1996)	$V_{MPC}=1-\frac{c}{c-1}(1-V_{PC})$
Partition entropy (Bezdek, 1974)	$V_{PE}=-\frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}u_{ij}\log(u_{ij})$
Average within–cluster distance (Krishnapuram and Freg, 1992)	$V_{\mathrm{AVCD}}=\frac{1}{c}\frac{1}{N}\sum_{i=1}^{c}\frac{\sum_{j=1}^{N}(u_{ij})^m\lvert\mathbf{x_j}-\mathbf{c_i}\rvert^2}{\sum_{j=1}^{N}(u_{ij})^m}$
Fukuyama-Sugeno index (Fukuyama and Sugeno, 1989)	$V_{FS}=\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^m\left(\lvert\mathbf{x}_j-\mathbf{c}_i\rvert^2-\left\lvert\left(\frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k\right)-\mathbf{c}_i\right\rvert^2\right)$
Xie-Beni index (Xie and Beni, 1991)	$V_{XB}=\frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^m\lvert\mathbf{x}_j-\mathbf{c}_i\rvert^2}{N\min_{i\neq j}\lvert\mathbf{c}_i-\mathbf{c}_j\rvert^2}$
PCAES (Wu *et al.*, 2005)	$V_{\mathrm{PCAES}}=\sum_{i=1}^{c}\sum_{j=1}^{N}\frac{(u_{ij})^2}{\min_{1\leq i\leq c}\left(\sum_{k=1}^{N}(u_{ij})^2\right)}-\sum_{i=1}^{c}\exp\left(-\min_{k\neq i}\left(\frac{c\lvert\mathbf{x}_i-\mathbf{x}_k\rvert^2}{\sum_{l=1}^{c}\left\lvert\mathbf{x}_l-\left(\sum_{s=1}^{N}\mathbf{x}_s/N\right)\right\rvert^2}\right)\right)$
Minimum centroid distance	$V_{MCD}=\min_{i\neq j}\lvert\mathbf{c}_i-\mathbf{c}_j\rvert^2$

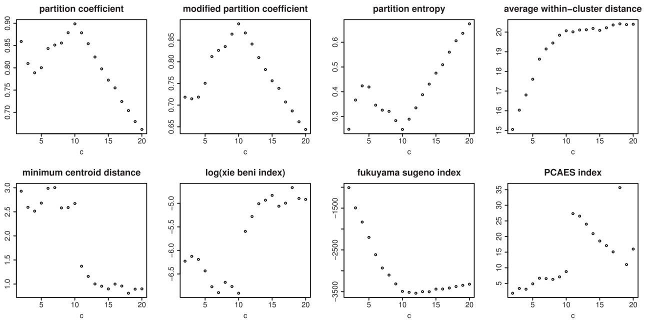

Fig. 7. Comparison of the different validity indices for an artificial system of 500 10-dimensional data points, with 10 clusters each with 50 points. All indices show that $c=10$ is an optimal solution given by a maximum, maximum, minimum, knee, jump, minimum, minimum and a maximum for the indices beginning from the upper left to the lower right, respectively.

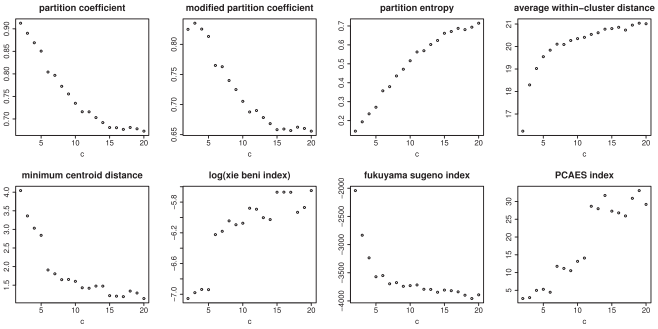

Fig. 8. Comparison of the different validity indices for the serum dataset. For real data, it is obviously more difficult to estimate the number of clusters. However, some indices have a jump at or near $c=5$.

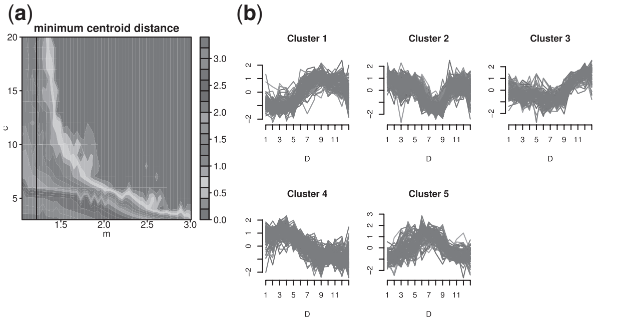

Fig. 9. **(a)** Landscape of the minimum centroid distances for the serum dataset. The strongest decay is found for $c=5$ around $m_t=1.25$. The black line denotes $m_t=1.25$. **(b)** Patterns of the objects in all five clusters depicting only the ones with membership values larger than 1/2. Different colors correspond to different membership values. A cluster analysis of the serum set with $m=2$ and $c=5$ (see Supplementary Materials) shows similar patterns as the ones found for $m=m_t$. However, the validation indices did not provide enough information and only half of the objects could be assigned to the clusters for $m=2$.

clustering (i.e. we find a decay of the minimum centroid distance at $c=7$ for $m=1.47$). For larger values of the fuzzifier, no clusters can be detected, whereas the decay begins to become less accentuated for smaller m-values. Hence, the minimum centroid distances may be considered as a powerful validity index for the case that the appropriate $m=m_t$ is chosen. Another advantage of using V_{MCD} is that its calculation is faster than the one of the other validity indices. Figure 6 proofs how a carefully chosen fuzzifier allows optimal cluster detection. When taking the popular value $m=2$, the cluster analysis finds a minimum centroid distance equal to zero for the cases $w=2$, $w=3$ and $w=4$ and thus cannot be considered as valid as there are at least two identical cluster centers.

For a comparison of the different validation indices, we generated a dataset with $D=13$, $N=500$, $M=10$ and $w=2$ for which Equation (5) gives $m_t=1.25$. Figure 7 shows the validation indices versus the cluster number c using $m=m_t$. All methods clearly indicate $c=10$ as the optimal solution. Note, that there is also a strong decay of V_{MCD} at $c=10$.

Real data normally is more complex than the artificial sets analyzed here. Not only the kind of noise may be different but also the clusters may not have normal distributed values and the clusters might have different sizes. As a consequence, often an optimal parameter set does not exist, and the most appropriate solution must be chosen manually out of the best candidates. As a test dataset we used the serum set (Iyer *et al.*, 1999) that has the same number of dimensions and a similar number of objects as the artificial dataset analyzed in Figure 7. The validation indices now do not agree in giving a clear indication for the number of clusters in the system (Fig. 8). However, most of them yield $c=5$ as the optimal solution. The abrupt decay of the minimum centroid distance at the same $c=5$ is remarkable. Figure 9a depicts the landscape of the minimum centroid distance for the serum dataset over a large range of m and c. We observe a similarity between Figure 9a and the case $w=3$ in Figure 6, suggesting that the dataset consists of overlapping but distinguishable clusters. The minimum centroid distance has a plateau for $c \leq 5$ and $m < 2$ with a decay at $c=5$ over a considerable range of m-values around $m_t=1.25$ indicating $c=5$ as the optimal choice.

Figure 9b shows the patterns of all clusters for the cluster analysis on the serum dataset taking $c=5$ and $m=1.25$. The lines correspond to the coordinates of the centroids. Only objects with membership values over 1/2 for the corresponding cluster are shown.

The optimal cluster numbers for the other real datasets calculated by inspection of the minimum centroid distance are shown in Table 2.

6 CONCLUSIONS

In fuzzy c-means cluster analysis, it is crucial to choose the optimal parameters since a large fuzzifier value leads to the loss of information and a low one leads to the inclusion of false observations originating from random noise. The value of the fuzzifier was frequently set to 2 in many studies without specification of the amount of noise in the system. We show here that the strong dependence of the optimal fuzzifier value on the dimension of the system requires fine-tuning of this parameter.

To our knowledge, two methods exist to obtain the fuzzifier by processing the dataset (Dembélé and Kastner, 2003; Futschik and Carlisle, 2005). In the former method, a distance matrix between all objects of the dataset needs to be calculated which makes the method cumbersome for large datasets as the number of distances increases with N^2. In Futschik and Carlisle (2005), they compare the values of a modified partition coefficient for a large range of the parameters m and c. As the clustering procedure also slows down considerably for larger datasets, this second method may require large computation times.

We present here a new, fast and simple method to estimate the fuzzifier being calculated from only two main properties of the dataset, its dimension and the number of objects. Using this method, we obtained not only an optimal balance between maximal cluster detection and maximal suppression of random effects but it also allows us to process larger datasets. Instead of using computationally expensive measures applying the clustering algorithm many times, the fuzzifier is simply calculated from Equation (5) and thus there are no slowdowns for large datasets. The estimated values of the fuzzifier are similar in all three methods, although they might be overestimated in Dembélé and Kastner (2003). Our method allows a larger range of dataset properties, like very high dimensions ($D > 40$) and very large numbers of objects. The results suggest that biased data lead to an increase of the value of the fuzzifier in low-dimensional datasets with a small number of objects (for instance, $N < 200$ and $D < 8$) and thus the parameters should be chosen carefully for this type of data. The estimation is based on the evaluation of the minimal distance between the centroids of the clusters found by the cluster analysis. The minimum centroid distance provides sufficient information for the estimation of the other parameter necessary for the clustering procedure, the number of clusters, and eliminates the need for calculation of computationally intensive validation indices.

In data from proteomic studies, especially labeled mass spectrometry data, protein expressions are compared over a generally smaller number of stages (for instance, less or equal to 8 in iTRAQ data). As our study shows, the optimal value of the fuzzifier increases strongly at low dimensions to values larger than $m = 2$ making it difficult to obtain well-defined clusters. Therefore, a compromise needs to be made, by allowing lower fuzzifier values, $m < m_t$, admitting the influence of random fluctuations to the results. A quantification of the confidence of the cluster analysis of low-dimensional data needs to be carried out or other methods of data comparison, such as direct comparison of the absolute data values, must complement the data analysis.

Funding: Danish Council for Independent Research, Natural Sciences (FNU) to V.S.

Conflict of Interest: none declared.

REFERENCES

Babuska,R. (1998) *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Dordrecht.

Bezdek,J.C. (1974) Cluster validity with fuzzy sets. *J. Cybernetics*, **3**, 58–72.

Bezdek,J.C. (1975) Mathematical models for systematics and taxonomy. In Estabrook,G.F. (ed.) *Proceedings of the 8th International Conference on Numerical Taxonomy*, San Francisco, Freeman.

Bezdek,J.C. (1981) *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, New York.

Cho,R.J. *et al.* (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65–73.

Dave,R.N. (1996) Validating fuzzy partition obtained through c-shells clustering. *Pattern Recogn. Lett.*, **17**, 613–623.

Dembélé,D. and Kastner,P. (2003) Fuzzy C-means method for clustering microarray data. *Bioinformatics*, **19**, 973–980.

Döring,C. *et al.* (2006) Data analysis with fuzzy clustering methods. *Comput. Stat. Data An.*, **51**, 192–214.

Dunn,J.C. (1973) A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *J. Cybernet.*, **3**, 32–57.

Eisen,M.B. *et al.* (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.

Fukuyama,Y. and Sugeno,M. (1989) A new method of choosing the number of clusters for the fuzzy c-means method. *Proc. 5th Fuzzy Syst. Symp.*, p. 247.

Futschik,M.E. and Carlisle,B. (2005) Noise-robust soft clustering of gene expression time-course data. *J. Bioinform. Comput. Biol.*, **3**, 965–988.

Hanai,T. *et al.* (2006) Application of bioinformatics for DNA microarray data to bioscience, bioengineering and medical fields. *J. Biosci. Bioeng.*, **101**, 377–384.

Höppner,F. *et al.* (1999) *Fuzzy Cluster Analysis*. John Wiley & Sons, Inc., New York.

Horton,P. and Nakai,K. (1996) A probabilistic classification system for predicting the cellular localization sites of proteins. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **4**, 109–115.

Iyer,V.R. *et al.* (1999) The transcriptional program in the response of human fibroblasts to serum. *Science*, **283**, 83–87.

Krishnapuram,R. and Freg,C.-P. (1992) Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recogn.*, **25**, 385–400.

Nash,W.J. *et al.* (1994) The population biology of Abalone (Haliotis species) in Tasmania. I. Blacklip Abalone (H. rubra) from the north coast and islands of Bass Strait. *Sea Fish. Div. Tech. Rep.*, **48**.

Olsen,J.V. *et al.* (2006) Global, in vivo, and site-specific phosphorylation dynamics in signaling networks. *Cell*, **127**, 635–648.

Pal,N.R. and Bezdek,J.C. (1995) On cluster validity for the fuzzy c–means model. *Fuzzy Syst.*, **3**, 370–379.

Pierce,A. *et al.* (2008) Eight-channel iTRAQ enables comparison of the activity of six leukemogenic tyrosine kinases. *Mol. Cell Proteomics*, **7**, 853–863.

Sigillito,V.G. *et al.* (1989) Classification of radar returns from the ionosphere using neural networks. *John Hopkins APL Tech. Digest*, **10**, 262–266.

Tamayo,P. *et al.* (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.

Tavazoie,S. *et al.* (1999) Systematic determination of genetic network architecture. *Nat. Genet.*, **22**, 281–285.

Wolf-Yadlin,A. *et al.* (2007) Multiple reaction monitoring for robust quantitative proteomic analysis of cellular signaling networks. *Proc. Natl Acad. Sci. USA*, **104**, 5860–5865.

Wu,K.-L. *et al.* (2005) A novel fuzzy clustering algorithm based on a fuzzy scatter matrix with optimality tests. *Pattern Recogn. Lett.*, **26**, 639–652.

Xie,X.L. and Beni,G. (1991) A validity measure for fuzzy clustering. *IEEE Trans. Pattern. Anal. Mach. Intell.*, **13**, 841–847.

Zadeh,L.A. (1965) Fuzzy sets. *Inf. Control*, **8**, 338–353.