This is a preprint of an article published in *Journal of Chemometrics* 1999; **13**: 435–444 (http://www.interscience.wiley.com/jpages/0886-9383/).

# The flexibility of fuzzy clustering illustrated by examples.

**Tormod Næs and Bjørn-Helge Mevik.**
**MATFORSK, Oslovegen 1, 1430 Ås, Norway**
**e-mail: tormod.naes@matforsk.no**

**ABSTRACT**

This paper presents a discussion of the versatility and flexibility of fuzzy clustering. Three examples of very different applications are presented. The focus is on flexibility with respect to distance measure used and with respect to the possibility of utilising known membership values for some of the samples.

*Key words:* fuzzy clustering, discriminant analysis, non-linear calibration, updating of classifiers, sorting of raw material.

## 1. Introduction.

Classification methods can coarsely be divided in two main groups of techniques, discriminant analysis and cluster analysis[1] . Cluster analysis refers to the unsupervised situation where little or no information is available about group structure prior to the classification. The goal is to *find* groups in the data. Discriminant analysis refers to situations where the membership of a set of training samples is known and the main purpose is to build a classification rule applicable for new and unknown samples.

The present paper is about cluster analysis. In particular, we will discuss three new and quite non-standard examples of the use of so-called fuzzy clustering. The purpose of the paper is to show how classical fuzzy clustering can be extended to handle very different applications. Two of the examples will show the flexibility of fuzzy clustering in handling different types of distance measures. The third example will consider a situation with elements of both supervised and unsupervised classification, thus showing how fuzzy clustering can be extended to combine known membership values with unknown ones.

Fuzzy clustering is a cluster analysis method that has been given quite a lot of attention in the chemometric literature already. Therefore only a brief description of the basics will be provided in Section 2 of the paper. Section 3 will contain the three examples. The relationship to basic theory will be discussed. Section 4 contains some brief conclusions.

## 2. Basic aspects of fuzzy clustering.

Contrary to other methods of clustering, the fuzzy clustering methods provide a number of membership values that indicate the degree of membership of the different samples to the different groups[2,3] . These values can be very important for understanding the data and for assessing how natural the groups are. Other methods like for instance hierarchical clustering methods give crisp groups as result, i.e. the membership values are either 0 or 1.

The membership values are here denoted by $u_{ij}$ and are collected in a matrix denoted by U. Each line in U corresponds to a sample and each column to a group. The $u_{ij}$ values in each line sum to 1. This means that the membership values of each sample (i) to the different groups (j) sum to 1. The number of samples is denoted by N and the number of groups by C. The number C has to be fixed during fuzzy clustering. However, different choices of C can be tested and the one with the best results can be selected. Indices have been developed for studying the quality of the splitting.

As for any other clustering method, fuzzy clustering is based on a distance measure. In classical applications and theory, the distances are either Euclidean or Mahalanobis distances (in the whole space or in a subspace [4,5] . For this introductory section we assume that the distance is Euclidean or Mahalanobis with the same fixed covariance matrix for all groups. Later we will focus on other distances as well. Distances will be denoted by $D_{ij}$ , measuring the distance from sample i to group j.

There exist several fuzzy clustering algorithms[6] . In this paper we focus on the so-called fuzzy k-means algorithm. This is based on minimising the following criterion:

$$J = \sum_{j=1}^{C} \sum_{i=1}^{N} {u_{ij}}^{m} {D_{ij}}^{2} \qquad (1)$$

The user can determine the parameter $m \geq 1$ in the exponent of u. An m equal to 1 gives crisp subsets, while larger values of m give fuzzy subgroups which also may be more robust to outliers. A much used value is m=2. This will be used for the rest of the paper. Note that minimisation of the J-criterion is natural because small values of u combined with large values of D (and vice versa) are favoured. We refer to Reference 7 for an approach which mixes the exponents m=1 and m=2, which yields a combination of crisp and fuzzy memberships.

The solution of the minimisation problem (at least for “traditional distances”) can according to Reference 4 be found by a rather simple numerical optimisation procedure. Initial values of u are either selected at random or determined according to prior information about group structure. Then D values, which minimise J for given u-values, are determined. For m=2, this is done by computing the weighted average

$$\bar{x}_j = \sum_{i=1}^{N} {u_{ij}}^{2} x_i \; / \sum_{i=1}^{N} {u_{ij}}^{2} \qquad (2)$$

and then by computing the Euclidean (or Mahalanobis) distances relative to these centres.

New u values, which minimise J for given D values can be computed by

$$u_{ij} = (\sum_{c=1}^{C} (D_{ij} \; / \; D_{ic})^{2})^{-1} \qquad (3)$$

The procedure is continued until convergence. For other m-values the procedure is similar.

For Mahalanobis distances with different covariance matrix for the different subgroups the procedure is similar, but not identical. This will be discussed below.

Since the procedure is a numerical optimisation, no guarantee for a globally optimal solution can be given. Therefore, repeating the procedure by using different starting points is recommended.

## 3. Three examples.

Fuzzy clustering has a number of useful properties. For instance:

- It provides membership values which are useful for interpretation
- It is flexible with respect to distance used
- If some of the membership values are known this can be incorporated into the numerical optimisation.

The examples to be discussed in this paper will concentrate on the importance of the second and third of these points. The two first examples will show how fuzzy clustering can accommodate very complicated distance measures. These distance measures are defined in such a way that they are updated during the numerical iteration process. They can therefore not be handled easily by any of the classical hierarchical clustering techniques. The first of these two examples is from splitting of non-linear calibration data into linear subgroups. The other one is from sorting of raw material in an industrial production process. The third example will cover the last point above and will show how fuzzy clustering can be used for updating of supervised classification procedures during regular operation.

The first example is already presented in this journal (Reference 8) and will here only be discussed briefly. The other two are new and at present subject to further investigations at MATFORSK. These are so far merely to be considered as ideas illustrated by examples, more than fully investigated strategies. Some ideas for further study will also be indicated.

### 3.1 Non-linear regression solved by local linear regressions

This method is a technique for solving non-linearity problems in regression/calibration where the purpose is to predict y-values from measured x-values. The idea behind the method is to find a splitting of the calibration data into subgroups which are as linear as possible. A linear regression model is then fitted to each group. When an unknown sample is to be predicted, it must first be put in the best group before its unknown y-value can be estimated [9].

In Reference 8 it was suggested to use the residual distance as a criterion for splitting. More specifically, the solution method suggested was minimisation of J with D defined as the residual distance

$$D_{ij}^{\ 2} = (y_i - x_i^{\ T}\hat{b}_j)^2 / \hat{\sigma}^2 \qquad (4)$$

Here the estimate of error variance, $\hat{\sigma}^2$, is incorporated in order to make the distance unitless (as is for instance the Mahalanobis-distance). As can be noted, this is a standard residual distance between the measured and fitted y-value for sample i when fitted to the model in group j.

For given U-values, minimisation of J in this case corresponds to the ordinary weighted least squares (WLS) optimisation. The U-values for a given set of distances can be found as above (equation (3)). Therefore, optimisation was done as for standard fuzzy clustering by switching between two simple steps, each one minimising the criterion J for given values of the other.

It was, however, found that this criterion could lead to groups with samples far from each other in the multivariate X-space. This is an unwanted property since it would be impossible to handle in practice for new unknown prediction samples. To compensate for this defect it was therefore decided to add a Mahalanobis distance (with constant covariance matrix) as a penalty to the residual distance. The combined criterion then became

$$D_{ij} = (vD_{1ij}{}^2 + (1-v)D_{2ij}{}^2)^{1/2} \qquad (5)$$

where $D_{1ij}$ is the residual distance (in (4)) and $D_{2ij}$ is the Mahalanobis distance defined by

$$D_{2ij} = sqrt((x_i - \bar{x}_j)^t M^{-1}(x_i - \bar{x}_j)), \qquad (6)$$

where M is the sample covariance matrix of the whole data set. Optionally, the $x_i$ can be replaced by a vector consisting of both x and y.

As can be seen this is a weighted average of the two distances. The parameter v can be determined by cross-validation (CV) or prediction testing on a grid of v values. The numerical procedure is just as simple as above since for given u-values, optimisation of J leads to separate optimisation of the residual and Mahalanobis distances.

A couple of simple illustrations of the use of the method were given in Reference 8. The method performed well in both cases. The convergence properties using the above suggestion appeared to be good. In other words, even though the method is not longer the standard fuzzy clustering procedure, a similar iterative procedure seemed to work reasonably well. The convergence properties should be studied further.

### 3.2 Sorting of raw material for industrial production

In many cases, in the food industry as well as in other branches, the raw material variation is one of the most important problems when trying to obtain stable product quality. This can in practice be solved in many different ways, by using for instance

- Statistical process control (SPC) techniques (see e.g. Reference 10) or feed-forward/feed-back control schemes
- Making the process robust to raw material variation
- Sorting raw material before production

In this paper we will be interested in the latter of these solutions. More specifically, we will be interested in identifying groups/categories of raw material, which are as homogeneous as possible and which at the same time allow for constant processing within each category.

For the purpose of obtaining these categories, a model relating raw material properties and process settings on one side with end product results on the other side will be needed. Such a model can often be created by using experimental design methodology and polynomial modelling of the data[11]

When the model is established, the problem is to define a criterion that optimises the raw material categories, i.e. a criterion which can be used to establish homogeneous categories of raw material that allow for constant processing and at the same time end results close to the target quality for the products. It will be assumed throughout the

paper that such a target value is available. Building an optimisation criterion can be done directly from the estimated model. This can be done using a continuous approach based on optimisation of integrals. Another strategy is to use a number of "samples" spread out over the actual region of interest. The algorithm will then decide how to split the actual "dataset" into groups.

Here we propose to minimise J with m=2 using the following distance criterion for sample i to group/category j

$$D_{ij}{}^2 = (\hat{y}(P_i, x^0{}_j) - T)^2 \qquad (7)$$

Here $\hat{y}$ represents the predicted value from the model, T is the target, $P_i$ represents raw material properties for sample i and $x^0_j$ represents optimal process conditions for group j. Optimal process conditions are defined in equation (8). Note that this criterion (7) measures the squared difference between the target value and the predicted value from the model for raw material properties $P_i$ and process conditions $x^0_j$. Optimal process conditions for group j are defined by

$$x^0{}_j = \arg\min_x \left(\sum_{i=1}^{N} u_{ij}{}^2 (\hat{y}(P_i, x) - T)^2\right) \qquad (8)$$

Minimisation of J is done as in the original procedure. One starts with a random U matrix and continues to find optimal values of D given U. Note that finding $x^0$ is a part of this optimisation. Then optimal values of U are found given D and so on.

The optimisation gives U value which are used to indicate how to split the data and also optimal process conditions for the different groups (i.e the $x^0$ values). Note that since the optimisation criterion is based on average squared differences between predicted values and a target value, equation (7) is closely related to robustness measures as those proposed by Taguchi.

We refer to Reference 6 for a study of raw materials using fuzzy clustering in another way than proposed here.

Let us now look at an example of this procedure. The example is from production of bread. The raw material property of interest is protein content (p) and the processing conditions are mixing time ($x_1$) and proofing time ($x_2$). Both protein content (in %) and the two process variables (here measured in minutes) are important for bread production. The experiment was originally conducted for a different study with a different purpose. For a more thorough description of the details of the experiment see e.g. Reference 12. Three base flours with different protein content were mixed in 10 different proportions according to a simplex lattice design. The 10 flours were combined with the two process variables at three levels each, leading to 90 different productions. The data were fitted to a polynomial model in the three variables. The response considered is loaf volume (V) and the target value is given as 520 ml. It was found that for the present data, protein content was more important than protein quality for loaf volume[13].

The focus is on how to combine the flour quality (here defined by protein content) and the two process variable in an optimal way to obtain loaf volume close to the target. The model linking p, $x_1$ and $x_2$ to V is found by using polynomial fitting. The model obtained is (with centred regressor variables and with two decimal digits) equal to

$$V = 532.26 + 22.93p + 2.05x_1 + 4.47x_2 + 0.84px_1 + 0.52px_2 - 0.30x_1^2 - 0.08x_2^2 \quad (9)$$

The problem is then to split the flour samples into subgroups with constant processing within each subgroup and loaf volume as close to 520 as possible. We here decided to use the ten samples in the design for this purpose. They cover the region of interest quite evenly. Other selections could have been made. How the selection of "samples" influences the splitting is not investigated in this paper.

The model (9) was then used for the optimisation described above. The procedure was tested for both C=2 and C=3. The convergence was good in both cases (typically 55-60 iterations were needed, different starting points were also used). The u-values for the two situations are given in Figure 1. A different symbol is used for each group. The points are linked with solid lines for the different groups. As can be seen, for the C=2 case, the optimal splitting point seems to be somewhere close to 12% protein. The optimal process conditions within the two groups are ($x_1$=11.5, $x_2$=59.9) and ($x_1$=7.6, $x_2$=50.4) for the low protein and high protein groups respectively. The "expected" loss for this splitting (i.e. average squared difference between $\hat{y}$ and T) is equal to 222.5. Compared to the expected loss when only one group is used, which is equal to 650.5, this represents a substantial improvement. When C=3, the expected loss is 80. 2, which is an even better value.

### 3.3 Updating a classification method during regular operation.

Let us assume that a discriminant rule has been developed for a particular application. Let us also assume that the discriminant rule has been used for a while. In other words, a number of extra measurement vectors are available. The question is then: is it possible to utilise these measurements in a constructive way to update the classifier for better performance, i.e is it possible to update a classifier according to information acquired during regular operation. To some degree, the answer to this is yes (see e.g. Reference 14). In the present example we will demonstrate that fuzzy clustering can be a way of solving this problem.

Let us assume that we have a training set of data X, divided into C subgroups. The number of training samples is $N_1$. The training data have been used to build a classifier. Let us assume that we have got $N_2$ new sample with known X- values only. The membership values of these new samples obtained under regular operation are unknown. They can, however, be estimated by using the classification rule obtained.

Let us assume that the classification rule used is the Mahalanobis distance part of the QDA criterion. Let us further assume that all means and covariance matrices in the distance are determined as weighed averages of data from different samples. The weights reflect the degree of confidence of the data points as members of the different groups. This idea is inspired by robust statistics where this procedure is used to reduce the influence of outlying units. The formulae for the means and covariance matrices to be used are:

$$\bar{x}_j = \sum_{i=1}^{N} u_{ij}^2 x_i \Big/ \sum_{i=1}^{N} u_{ij}^2 \quad (10)$$

$$\hat{\Sigma}_j = \sum_{i=1}^{N} u_{ij}^2 (x_i - \bar{x}_j)(x_i - \bar{x}_j)^T \Big/ \sum_{i=1}^{N} u_{ij}^2 \quad (11)$$

For the training data, all the weights are 1 or 0 for each sample, since membership is known for all samples. The problem is then to update means and covariances by

determining the weights of the new samples. Obviously, most of them should have a different weight than the training samples for which the membership is totally known.

Here we propose to use a criterion based on fuzzy clustering. Before we present the criterion used, we give a brief description of a related approach described in References 2 and 15. It Reference 2 a method for optimising

$$J = \sum_{j=1}^{C} \sum_{i=1}^{N} u_{ij}^{\ 2} (x_i - v_j)^T A_j (x_i - v_j) \qquad (12)$$

is presented. It was stated that in order to make the solution tractable, the determinants of the "inverse covariance matrices" $A_j$, must be fixed (other solutions without this restriction are published in Reference 6). Let us define them here by $\rho_j = \det(A_j)$. It is easy to see that without such a restriction, the J criterion can sometimes become small due to large "covariances" instead of natural groups in the data. The solution to the problem can be found by an iterative procedure similar to the one above where optimal $v_j$ and $A_j$ are found for given U and optimal U is found for given $v_j$ and $A_j$. The $A_j$ can be found according to the formula

$$A_j = (\rho_j \det(S_j))^{1/K} S_j^{\ -1} \qquad (13)$$

where

$$S_j = \sum_{i=1}^{N} u_{ij}^{\ 2} (x_i - \bar{x}_j)(x_i - \bar{x}_j)^T .$$

Here K is the dimension of x. This means that optimisation of J can be formulated as optimisation of

$$J = \sum_{j=1}^{C} \sum_{i=1}^{N} u_{ij}^{\ 2} a_j D_{ij}^{\ 2} \qquad (14)$$

where D's are Mahalanobis distances based on the fuzzy means and covariances (equations 10 and 11) and the a's are normalising "constants" to make the contributions from the different groups comparable.

If we split (14) according to the training set and new data set we obtain

$$J = \sum_{j=1}^{C} \sum_{i=1}^{N_1} u_{ij}^{\ 2} a_j D_{ij}^{\ 2} + \sum_{j=1}^{C} \sum_{i=N_1+1}^{N_1+N_2} u_{ij}^{\ 2} a_j D_{ij}^{\ 2} \qquad (15)$$

$N_1$ is here the number of training samples and $N_2$ is the number of new samples. The u values for the training samples are set to 0 and 1 according to which group they belong to because their membership is known exactly. The other u's (in the last sum) are unknown to us. The problem is then to determine all D's and the u's in the last sum, keeping the u's in the first sum fixed. This can then be considered a kind of semi-supervised classification. Some of the samples have known membership and other have not. The known samples are used to guide the allocation of unknown samples into subgroups. The optimisation procedure discussed above can easily be modified to situations with some of the u's (in the first sum in (15)) kept out of the iteration process.

The procedure has so far only been tested by simulations. In the following, we present some simple results from a two variable situation ($x_1$ and $x_2$ only) with two groups (C=2) each consisting of 20 samples. The new data used for updating were generated the same way (20 for each group). The data are normally distributed, with different means and with different covariance matrices. The two covariance matrices are

$$\Sigma_1 = \begin{pmatrix} 0.98 & 0 \\ 0 & 0.18 \end{pmatrix}$$

$$\Sigma_2 = \begin{pmatrix} 8.00 & 0 \\ 0 & 1.28 \end{pmatrix}$$

The structure of the two groups is indicated by the illustration in Figure 2 for one of the choices of the mean. Different means were used in the simulations (see Table 1). The test data (50 samples from each group) are generated from the same populations. The procedure was repeated 50 times for each set of group means. The classification error before and after updating are reported as a function of the difference between the means, which is probably one of the most important parameters to be varied in this case.

In the procedure it was assumed that the determinant of the two covariance matrices is the same. This is obviously not true for the present data. Therefore, other more sophisticated estimates of the relative “size” of the covariance matrices (from e.g. the training set) could possibly lead to even better results for the updating procedure.

As can be seen, the updating procedure gave a quite substantial improvement over the other classification rule. As can also be seen, the variability of the classification errors for the updated procedure are much smaller than for the original classification rule. This means that not only the average classification error, but also the majority of the errors are better after updating.

This is an area with many open questions and a lot of opportunities for research. One of the areas which has yet not been considered is how this (or other procedures) could be modified in the case of collinearity.

No convergence problems were discovered during the simulations.

**4. Conclusions.**

The present paper has shown the flexibility of fuzzy clustering can be extended to handle distance measures which would have been difficult to handle by standard hierarchical clustering methods. In addition, it has been shown that fuzzy clustering can handle known membership values for some of the samples in the dataset.

Fuzzy clustering is therefore a very interesting concept which can be modified and varied in many different ways.

In the computations, even for the more complex distance measures, the convergence properties of the standard algorithmic structure seem to be reasonably good.

***References.***


1. Mardia, K.V., Kent, J.T. and Bibby. J.M. *Multivariate analysis*. Academic Press, London (1980).

2. Bezdek, J.C. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press. NY (1981).

3. Kaufman, L. and Rousseeuw, P.J.. *Finding groups in data. An introduction to cluster analysis.* J. Wiley and sons. NY (1990).

4. Bezdek, J.C., Coray, C., Gunderson, R. and Wantson, J. . *SIAM J. Appl. Math*., 40, 2, 339 (1981a).

5. Bezdek, J.C., Coray, C., Gunderson, R. and Watson, J. *SIAM J. Appl. Math*, 40, 2, 358 (1981b).

6. Rousseeuw, P.J., Kaufman, L. and Trauwaert, E.. *Computational statistics and data analysis.*, 23, 135 (1996).

7. Rousseeuw, P.J., Trauwaert, E. and Kaufman, L. *Journal of computational and applied mathematics*., 64, 81 (1995).

8. Næs, T. and Isaksson, T. . *J. Chemometrics*, 5, 49 (1991).

9. Næs,T. . *J. Chemometrics*., 5, 487-501 (1991).

10. Montgomery, D.C. *Introduction to statistical quality control.* John Wiley and Sons. NY (1985).

11. Box, G.E.P., Hunter, W.G. and Hunter, J.S. *Statistics for experimenters*. John Wiley and sons, NY (1978).

12. Næs, T. , Færgestad, E.M. and Cornell, J. *Chemometrics and intelligent laboratory systems.* 41, 221 (1998).

13. Færgestad, E.M and Næs, T. *Cereal Chemistry* (1999). (in press).

14. O'Neill. T.J. . *J. Amer. Stat. Assoc*. 73, 364, 821 (1978).

15. Gustafson, D.E. and Kessel, W. *In. Proc. IEEE-CDC, Vol. 2., ed. K.S. Fu, Piscataway, NJ: IEEE Press*, 761 (1979).

**Table 1.** Results from the simulation. The centre of group 1 i always (1,1). The centre of group 2 is indicated in the left margin. The error rates are given together with their standard deviations (in parentheses). There error rates are average of 50 independent simulations.

| Centre of group 2 | Error rate | Error rate after updating |
|---|---|---|
| (2,2) | **32.4** (8.6) | **22.7** (6.2) |
| (2.5,2.5) | **19.7** (7.3) | **11.7** (3.7) |
| (3,3) | **10.9** (5.4) | **6.3** (3.1) |
| (3.5,3.5) | **6.0** (3.6) | **3.3** (1.9) |

**Figure captions.**

Figure 1. Example 2. Plot of the u-values for the different subgroups for C=2 (a) and C=3 (b). The u values are plotted as functions of the protein content of the flours.

Figure 2. Plot of simulated data structure when the average of group 2 is equal to (2.5, 2.5). The average of the other group is (1,1).