Resampling for Fuzzy Clustering

Christian Borgelt

European Center for Soft Computing
c/ Gonzalo Gutiérrez Quirós s/n, 33600 Mieres, Spain
christian.borgelt@softcomputing.es

Abstract. Resampling methods are among the best approaches to determine the number of clusters in prototype-based clustering. The core idea is that with the right choice for the number of clusters basically the same cluster structures should be obtained from subsamples of the given data set, while a wrong choice should produce considerably varying cluster structures. In this paper I give a brief overview how such resampling approaches can be transferred to fuzzy and probabilistic clustering.

1 Introduction

A core problem of prototype-based clustering algorithms—like classical c-means [12, 17], its fuzzy counterpart (fuzzy c-means) [2, 13], or expectation maximization for mixtures of Gaussians [5, 7]—is that they require the number of clusters to be known in advance. A common approach to tackle this problem is to cluster the data set several times, each time with a different number of clusters from a user-specified range, and then to choose the number of clusters yielding the best evaluation (see, for example, [2, 13, 4] for overviews of evaluation measures).

In this paper I study an alternative approach that has recently attracted a lot of attention in crisp and probabilistic clustering. The core idea is that if we cluster subsamples of the given data set with the "right" number of clusters, we should end up with basically the same cluster structure in each run. With a "wrong" number of clusters, however, the clustering result should be unstable, showing considerable variation between different subsamples. Thus, by measuring the stability of the clustering result w.r.t. subsampling (similarity of results from different runs), one may be able to determine the "best" number of clusters: it is the one for which the clustering results are most stable.

Intuitively, one may think of this as follows: if the "true" number of clusters is c and we try to find $c+1$ clusters, one cluster has to be split. If we try to find $c-1$ clusters, some pair of clusters has to be merged. As it depends on particular properties of the subsample which cluster is split or which clusters are merged, we should get somewhat differing structures in each run. By measuring how well the clustering results coincide, we can thus discover such situations and choose the number of clusters based on this information.

In addition to a general discussion of this highly promising approach, I study experimentally how the choice of t-norms in the needed relative cluster evaluation measures (to combine membership degrees) affects the quality and clarity of the results, that is, how well the "best" number of clusters can be determined.

	$u_{kj}^{(2)} = 1$	$u_{kj}^{(2)} = 0$	Σ
$u_{ij}^{(1)} = 1$	$n_{11}^{(i,k)}$	$n_{10}^{(i,k)}$	$n_{1.}^{(i,k)}$
$u_{ij}^{(1)} = 0$	$n_{01}^{(i,k)}$	$n_{00}^{(i,k)}$	$n_{0.}^{(i,k)}$
Σ	$n_{.1}^{(i,k)}$	$n_{.0}^{(i,k)}$	n

Table 1. Contingency table comparing rows of two (crisp) partition matrices (i and k are the cluster indices).

2 Relative Cluster Evaluation Measures

Relative cluster evaluation measures compare two partitions of given data, each of which can be described by a $c \times n$ partition matrix $\mathbf{U} = (u_{ij})_{1 \le i \le c, 1 \le j \le n}$, where c is the number of clusters and n the number of data points. An element u_{ij} of such a matrix states, in the crisp case, whether the j-th data point belongs to the i-th cluster ($u_{ij} = 1$) or not ($u_{ij} = 0$). In the fuzzy case, u_{ij} is the degree of membership to which the j-th data point belongs to the i-th cluster (usually satisfying the constraint $\forall j; 1 \le j \le n : \sum_{i=1}^{c} u_{ij} = 1$).

The main problem of the comparison is how to relate the clusters of one partition to the clusters of the other. There are basically three solutions: (1) for each cluster in the one partition we determine the *best fitting* cluster in the other, (2) we find the *best row permutation*, that is, the best one-to-one mapping of the clusters, or (3) we compare indirectly by first setting up a *coincidence matrix* for each partition matrix, which records for each pair of data points whether they are assigned to the same cluster or not, and then compare these matrices. Here I confine myself to the second and the third alternative.

2.1 Comparing Partition Matrices

To compare two $c \times n$ partition matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ directly, we need a measure that compares two rows, one from each matrix. Such measures can be derived from measures comparing binary classifications, like, for example, the *accuracy* or the *F_1-measure* [19]. Formally, we set up a 2×2 contingency table for each pair of rows, one from each matrix (cf. Table 1). That is, for each pair $(i, k) \in \{1, \ldots, c\}^2$ and each row-column pair $(a, b) \in \{0, 1\}^2$ we compute

$$n_{ab}^{(i,k)}\big(\mathbf{U}^{(1)}, \mathbf{U}^{(2)}\big) = \sum_{j=1}^{n} \Big((1-a) + (2a-1)\, u_{ij}^{(1)}\Big) \cdot \Big((1-b) + (2b-1)\, u_{kj}^{(2)}\Big).$$

(In the following I generally drop the arguments $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ to make the formulae easier to read.) These numbers may also be computed from fuzzy membership degrees, where they have a fairly natural interpretation: in the crisp case, n_{11} is the number of data points that are assigned to the i-th cluster of the first partition *and* to the k-th cluster of the second partition, where the *and* is formally expressed by a product. Allowing membership degrees from $[0, 1]$ and drawing on the theory of fuzzy logic, we see that this is only a special case of a t-norm that combines the two statements. Hence, in the general case, we may

replace the product by an arbitrary t-norm. Analogously, the expressions $1 - u_{ij}$ (for $a = 0$ or $b = 0$) can be seen as resulting from an application of the standard fuzzy negation, and indeed: they refer to negated statements "The j-th data point does *not* belong to the i-th cluster." In this way we achieve a straightforward generalization of all following measures to fuzzy clustering results.

From the numbers $n_{ab}^{(i,k)}$ computed above we may now compute any measure for evaluating a binary classification, maximizing the result over all row permutations.[1] An example is the (averaged) F_1 *measure* [19]

$$F_1\big(\mathbf{U}^{(1)}, \mathbf{U}^{(2)}\big) = \max_{\varsigma \in \Pi(c)} \frac{1}{c} \sum_{i=1}^{c} \frac{2\,\pi_{i,\varsigma(i)} \rho_{i,\varsigma(i)}}{\pi_{i,\varsigma(i)} + \rho_{i,\varsigma(i)}},$$

where $\Pi(c)$ is the set of all permutations of the c numbers $1, \ldots, c$ and cluster-specific precision and recall are

$$\pi_{i,k} = \frac{n_{11}^{(i,k)}}{n_{01}^{(i,k)} + n_{11}^{(i,k)}} \qquad \text{and} \qquad \rho_{i,k} = \frac{n_{11}^{(i,k)}}{n_{10}^{(i,k)} + n_{11}^{(i,k)}}.$$

Another example is *(cross-classification) accuracy*, averaged over all columns:

$$Q_{\text{acc}}\big(\mathbf{U}^{(1)}, \mathbf{U}^{(2)}\big) = \max_{\varsigma \in \Pi(c)} \frac{1}{cn} \sum_{i=1}^{c} \left(n_{00}^{(i,\varsigma(i))} + n_{11}^{(i,\varsigma(i))} \right).$$

Two partition matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are the more similar, the higher the values of the (averaged) F_1 measure or the (cross-classification) accuracy. An alternative is a simple mean squared difference comparison of the partition matrices (which, at least to my knowledge, has not been used before). That is, we compute

$$Q_{\text{diff}}\big(\mathbf{U}^{(1)}, \mathbf{U}^{(2)}\big) = \min_{\varsigma \in \Pi(c)} \frac{1}{cn} \sum_{i=1}^{c} \sum_{j=1}^{n} \left(u_{ij}^{(1)} - u_{\varsigma(i)j}^{(2)} \right)^2.$$

The smaller this measure, the more similar are the partitions.

2.2 Comparing Coincidence Matrices

As an alternative to comparing partition matrices directly, one may first compute from each of them an $n \times n$ *coincidence matrix*, also called a *cluster connectivity matrix* [16], which states for each pair of data points whether they are assigned to the same cluster or not. Formally, a coincidence matrix $\mathbf{\Psi} = (\psi_{jl})_{1 \le j,l \le n}$ can be computed from a partition matrix $\mathbf{U} = (u_{ij})_{1 \le i \le c, 1 \le j \le n}$ by

$$\psi_{jl} = \sum_{i=1}^{c} u_{ij} u_{il}.$$

These values may also be computed from fuzzy membership degrees, possibly replacing the product (which represents a conjunction) by some other t-norm.

[1] Note that with the so-called *Hungarian method* for solving optimum weighted bipartite matching problems [18] the time complexity of finding the maximum over all permutations for given pairwise column comparison values is $O(c^3)$ and not $O(c!)$.

	$\psi_{jl}^{(2)} = 1$	$\psi_{jl}^{(2)} = 0$	Σ
$\psi_{jl}^{(1)} = 1$	N_{11}	N_{10}	$N_{1\cdot}$
$\psi_{jl}^{(1)} = 0$	N_{01}	N_{00}	$N_{0\cdot}$
Σ	$N_{\cdot 1}$	$N_{\cdot 0}$	$N_{\cdot\cdot}$

Table 2. Contingency table for comparing (crisp) coincidence matrices (the indices 1 and 0 mean same and different cluster, respectively).

Such matrices are compared by computing statistics of the number of data point pairs that are in the same group in both partitions, in the same group in one, but in different groups in the other, or in different groups in both. Formally, we compute a 2×2 contingency table (cf. Table 2) containing the numbers (which are basically counts of the different pairs $\left(\psi_{jl}^{(1)}, \psi_{jl}^{(2)}\right)$)

$$N_{ab}\big(\mathbf{\Psi}^{(1)}, \mathbf{\Psi}^{(2)}\big) = \sum_{j=2}^{n}\sum_{l=1}^{j-1}\left((1-a)+(2a-1)\,\psi_{jl}^{(1)}\right)\left((1-b)+(2b-1)\,\psi_{jl}^{(2)}\right),$$

where an index $a, b = 1$ stands for "same group" and an index $a, b = 0$ stands for "different groups". (The arguments $\mathbf{\Psi}^{(1)}$ and $\mathbf{\Psi}^{(2)}$ are dropped in the following.) Again the product may be replaced by any t-norm (note that $\psi_{jl} \in [0, 1]$, since fuzzy clustering satisfies $\forall j; 1 \leq j \leq n : \sum_{i=1}^{c} u_{ij} = 1$). From these numbers a large variety of measures may be computed, including the *Rand statistic*

$$Q_{\text{Rand}}\big(\mathbf{\Psi}^{(1)}, \mathbf{\Psi}^{(2)}\big) = \frac{N_{11}+N_{00}}{N_{\cdot\cdot}},$$

which is a simple ratio of the number of data point pairs treated the same in both partitions to all data point pairs, and the *Jaccard coefficient*

$$Q_{\text{Jaccard}}\big(\mathbf{\Psi}^{(1)}, \mathbf{\Psi}^{(2)}\big) = \frac{N_{11}}{N_{11}+N_{10}+N_{01}},$$

which ignores negative information, that is, pairs that are assigned to different groups in both partitions. Both measures are to be maximized. Another frequently encountered measure is the *Folkes–Mallows index*

$$Q_{\text{Folkes-Mallows}}\big(\mathbf{\Psi}^{(1)}, \mathbf{\Psi}^{(2)}\big) = \frac{N_{11}}{\sqrt{(N_{11}+N_{10})(N_{11}+N_{01})}},$$

which can be interpreted as a cosine similarity measure and thus is also to be maximized. A final example is the *Hubert index*

$$Q_{\text{Hubert}}\big(\mathbf{\Psi}^{(1)}, \mathbf{\Psi}^{(2)}\big) = \frac{N_{\cdot\cdot}N_{11}-N_{1\cdot}N_{\cdot 1}}{\sqrt{N_{1\cdot}N_{\cdot 1}N_{0\cdot}N_{\cdot 0}}},$$

which may either be interpreted as a product-moment correlation or as the square root of the (normalized) χ^2 measure. It should be clear that this list does not exhaust all possibilities. Basically all of the abundance of measures, by which (binary) vectors and matrices can be compared, are applicable.

3 Resampling

Resampling methods can be found with basically two sampling strategies. In the first place, one may use *subsampling* [8], that is, the samples are drawn without replacement from the given data set, so that each data point appears in at most one data subset. This strategy is usually applied in a cross validation style, that is, the given data set is split into a certain number of disjoint subsets (with two subsets being the most common choice). The alternative is *bootstrapping* [6], in which samples are drawn with replacement, so that a data point may appear multiple times in the same data subset. There are good arguments in favor and against both approaches, but the results often do not differ much.

Resampling is used for cluster validation and model selection as follows: a cluster model can usually be applied as a classifier, thus enabling us to assign data points, which have not been used to build the cluster model, to the clusters. In this way we obtain, with the same algorithm, two different groupings of the same set of data points. For example, one may be obtained by clustering the data set, the other by applying a cluster model that was built on another data set. These two groupings can be compared using, for example, one of the measures discussed in the preceding section. By repeating such comparisons with several samples drawn from the original data set, one can obtain an assessment of the variability of the cluster structure (or, more precisely, an assessment of the variability of the evaluation measure for the similarity of partitions). Such an approach may be applied to select the most appropriate cluster model—and in particular, the "best" number of clusters—by executing the above algorithm for different parameterizations of the clustering algorithm and then to select the one showing the lowest variability. Specific algorithms following this general scheme have been proposed in [16, 20, 15], which differ in the exact resampling strategies and the evaluation measures used. All indicate that this approach is very robust and a fairly reliable way of choosing the number of crisp clusters.

4 Experiments

I carried out several experiments by applying a resampling approach for fuzzy clustering based on the above explanations to five data sets. The first three are artifical two-dimensional data sets of 400 data points each with three, four, and six clusters, respectively. They are shown in Figure 1. The fourth data set is an artificial three-dimensional data set of 400 data points with five equally populated, but ellipsoidal clusters. It is shown on the left in Figure 2. The last data set is the well-known wine data set from the UCI machine learning repository [3], a view of which is shown on the right in Figure 2. It comprises three classes of Italian wines and thus one can expect to find three clusters. I used attributes 7, 10, and 13, which are the most informative w.r.t. the class.

Before clustering all datasets were normalized in all dimensions to mean 0 and standard deviation 1 to rule out scaling effects. The experiments were carried out with the following resampling scheme: first the whole data set was clustered.

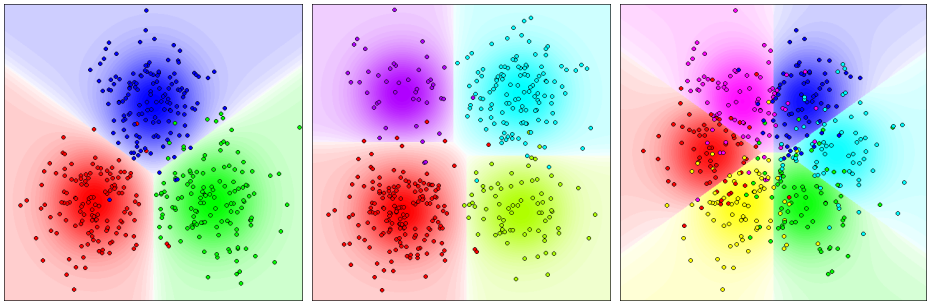

Fig. 1. Artificial data sets with 3 (equally populated), 4 (differently populated), 6 (equally populated) spherical clusters.

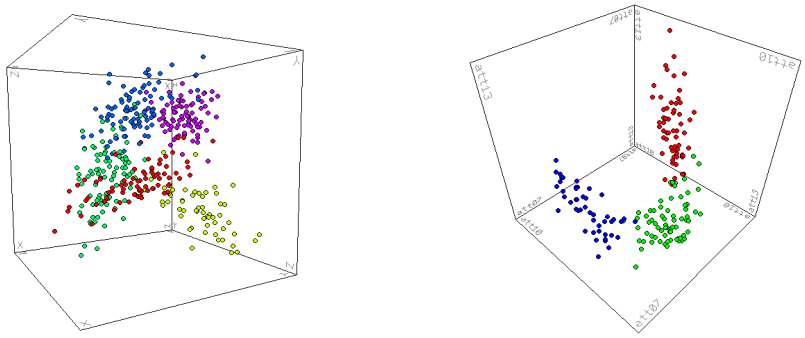

Fig. 2. An artificial data set with 5 (equally populated) ellipsoidal clusters and a view of the wine data set (attributes 7, 10, and 13).

Then 100 random samples (without replacement) were drawn from the data set, each of which comprised about half of the data points. (The data set was split into two equal parts, one of which was used). Each sample was clustered with the same number of clusters as the full data set and then the two cluster structures (one obtained from the full data set and one from the sample) were compared on the full data set using the measures described in Section 2. The evaluation results were averaged over the 100 samples, thus yielding a stability measure.

In the measures I used four different t-norms to combine membership degrees and the coincidence matrix entries (see Figure 3 for illustrations):

$$\top_{\min}(a,b) = \min\{a,b\}, \quad \top_{\text{minnp}}(a,b) = \min\{a,b\} \text{ if } a+b \geq 1, \text{ 0 otherwise},$$
$$\top_{\text{prod}}(a,b) = a \cdot b, \quad \top_{\text{Luka}}(a,b) = \max\{0, a+b-1\},$$

where $\top_{\text{minnp}}$ is the so-called *nil-potent minimum*. Since there are two places where a t-norm is needed in the measures based on comparing coincidence matrices, I tried all pairs of t-norms to explore their interactions. As it turns out, they cannot be combined freely: some combinations do not work well.

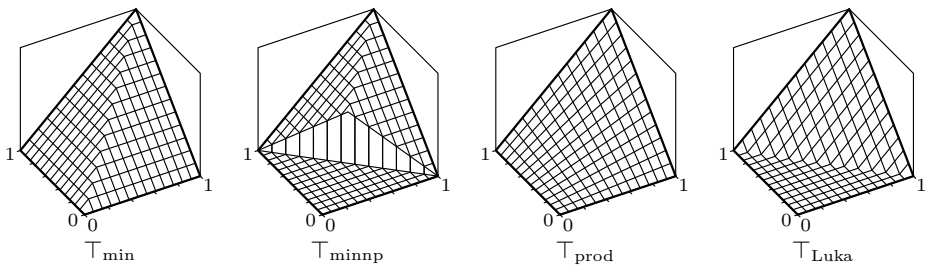

Fig. 3. The different t-norms used in the experiments.

data	diff	accuracy				F_1			
		min	mnp	prd	Luk	min	mnp	prd	Luk
art. 3	4	4	1	1	1	4	5	4	4
art. 4	3	2	4	0	0	3	3	1	3
art. 6	6	6	6	6	6	6	6	6	6
wine	4	4	0	1	1	4	4	0	4
art. 5	4	4	0	1	1	1	1	0	6
wine	5	4	4	0	0	0	0	0	1

Table 3. Overview of the results of comparing partition matrices with different measures and t-norms on the different data sets. Fuzzy c-means clustering was used for the first four rows, Gustafson–Kessel clustering for the last two.

Since it is not possible to show all individual results in this paper (there are simply too many different experiments), I try to give an impression of the performance of the different measures (in combination with different selections of t-norms) by providing a rough overview and reporting some individual results. The overview is shown in Tables 3 and 4 and uses grades to assess the performance of the different measures, with the following meanings:

- 6: clear global optimum at the correct cluster number, no local optimum at any other cluster number
- 5: clear global optimum at the correct cluster number, but there is a (weak) local optimum at another cluster number
- 4: only weak global optimum at the correct cluster number, or a competing local optimum at another cluster number
- 3: clear local optimum at the correct cluster number, but global optimum is at another cluster number
- 2: only weak local optimum at the correct cluster number, or global optimum is significantly higher than local optimum
- 1: only a discernable step at the correct cluster number, but not even a weak local optimum
- 0: no discernable characteristics at the correct cluster number

With grades 6 and 5, maybe also 4, the measure is usable for fully automatic selection, with grades 4, 3 and 2 for semi-automatic processing (with user interaction). With grades 1 and 0 a measure fails to find the correct cluster number.

Rand	min				minnp				prod				Luka			
data	min	mnp	prd	Luk	min	mnp	prd	Luk	min	mnp	prd	Luk	min	mnp	prd	Luk
art. 3	4	1	1	0	4	1	1	1	4	1	1	0	4	4	4	4
art. 4	4	1	1	0	4	0	1	1	2	1	1	0	3	3	2	2
art. 6	3	6	6	2	3	6	6	6	3	6	1	3	6	6	6	6
wine	4	0	1	0	4	1	1	1	4	0	0	0	4	0	0	0
art. 5	4	0	0	0	4	1	0	0	4	0	0	0	4	4	4	4
wine	4	1	0	0	4	0	0	1	0	1	1	0	0	4	4	0

Jaccard	min				minnp				prod				Luka			
data	min	mnp	prd	Luk	min	mnp	prd	Luk	min	mnp	prd	Luk	min	mnp	prd	Luk
art. 3	0	5	4	5	0	5	6	5	0	5	1	5	5	5	6	5
art. 4	1	3	1	3	1	3	3	3	0	3	0	3	3	3	3	3
art. 6	3	6	6	6	3	6	6	6	3	6	6	6	3	6	6	6
wine	0	4	1	4	0	0	0	5	0	0	0	4	0	0	0	5
art. 5	0	6	0	6	0	6	0	1	0	6	0	1	6	6	1	6
wine	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Folkes	min				minnp				prod				Luka			
data	min	mnp	prd	Luk	min	mnp	prd	Luk	min	mnp	prd	Luk	min	mnp	prd	Luk
art. 3	1	4	4	4	1	5	6	5	0	5	1	5	4	5	6	5
art. 4	1	3	1	3	1	3	3	3	0	3	0	3	3	3	3	3
art. 6	3	6	6	6	3	6	6	6	3	6	6	6	3	6	6	6
wine	0	4	0	4	0	0	0	4	0	0	0	4	0	0	0	4
art. 5	0	6	0	6	0	6	0	1	0	6	0	1	6	6	1	6
wine	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Hubert	min				minnp				prod				Luka			
data	min	mnp	prd	Luk	min	mnp	prd	Luk	min	mnp	prd	Luk	min	mnp	prd	Luk
art. 3	4	4	4	4	5	5	6	5	5	4	6	5	5	5	5	5
art. 4	6	4	4	4	3	6	6	6	3	6	6	5	3	3	3	3
art. 6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
wine	4	0	0	4	4	4	4	4	6	4	4	4	4	4	4	4
art. 5	1	6	1	4	0	6	1	6	0	6	1	6	6	6	6	6
wine	6	4	4	6	6	6	1	6	1	6	1	6	1	1	1	1

Table 4. Overview of the results of comparing coincidence matrices with different measures and t-norms on the different data sets. In each table the upper header row shows the t-norm for combining coincidence matrix entries, the lower header row the t-norm for combining membership degres. Fuzzy c-means clustering was used for the first four rows, Gustafson–Kessel clustering for the last two rows of each table.

	partition matrix			coincidence matrix			
#	diff	acc	F1	Rand	Jacc.	Folkes	Hubert
2	**.0001**	**.9918**	**.7606**	.7028	**.5769**	**.7317**	.3987
3	.0157	.9521	.6799	.6859	.4988	.6650	.3696
4	**.0009**	**.9850**	**.6851**	**.7123**	.4903	.6579	**.4097**
5	.0203	.9365	.5885	.6741	.4156	.5870	.3178
6	.0150	.9461	.5691	.6741	.3926	.5636	.3036
7	.0132	.9520	.5542	.6756	.3769	.5473	.2946
8	.0159	.9470	.5213	.6767	.3633	.5329	.2858

Table 5. Fuzzy clustering results on second artifical data set (4 clusters). All measures were computed with the minimum for the t-norm(s).

	partition matrix			coincidence matrix			
#	diff	acc	F1	Rand	Jacc.	Folkes	Hubert
2	.1135	.4729	.6971	.5968	.2925	.4470	.1299
3	.0337	.6125	.7659	.7745	.2580	.4057	.2667
4	.0066	.7224	.8618	.8709	**.3140**	**.4768**	.4033
5	**.0022**	.7636	**.8781**	.9076	.3081	.4709	**.4203**
6	.0109	.7663	.7036	.9299	.2365	.3820	.3449
7	.0122	.7838	.6049	.9477	.2008	.3340	.3068
8	.0103	**.8030**	.5652	**.9602**	.1786	.3024	.2820

Table 6. Fuzzy clustering results on fourth artifical data set (5 clusters). All measures were computed with the Łukasiewicz t-norm to combine the membership degrees and the product to combine the coincidence matrix entries.

These result tables show that one has to be very careful when choosing the measure and the t-norm(s), since a lot of combinations fail miserably. However, there are also a lot of combinations that work very nicely. Especially the Hubert index, which appears to be fairly robust w.r.t. the choice of the t-norms yields excellent results if either the Łukasiewicz t-norm or the nil-potent minimum are chosen to combine the membership degrees. (The t-norm used to combine the membership degrees is stated in the second header row.) This behavior is almost independent of the t-norm that is used to combine the coincidence matrix entries. All other coincidence matrix based measures seem to have problems with the wine data set (see below for a possible explanation).

Among the partition matrix based measures the newly introduced simple mean squared difference comparison performs fairly reliably, followed by the accuracy computed with the minimum as the t-norm. However, none of these measures quite reaches the performance of the properly parameterized Hubert index. Therefore the Hubert index seems to be the best choice.

To give an impression of individual results, Tables 5 to 8 show detailed tables for two artificial data sets and the wine data set. The results in Tables 6 and 8 are based on Gustafson–Kessel clustering [9], the rest on fuzzy c-means clustering. The used t-norms are indicated in the table captions. For each column the global and, if it exists, a relevant local optimum are highlighted.

The results on the wine data set (Table 7) indicate that maybe five clusters are an alternative to the number of classes (three). However, this may also be explained by ellipsoidal cluster shapes. The results shown in Table 8 make this likely, as here no local optima can be observed for five clusters.

	partition matrix			coincidence matrix			
#	diff	acc	F1	Rand	Jacc.	Folkes	Hubert
2	.0102	.7747	**.7668**	.7007	**.5566**	**.7139**	.4009
3	**.0013**	.8539	.7689	.8176	.5489	.7091	.5770
4	.0244	.8180	.6032	.8232	.4200	.5878	.4761
5	**.0056**	**.8669**	**.6409**	.8753	**.4345**	**.6049**	**.5313**
6	.0125	.8655	.5556	.8921	.3463	.5129	.4525
7	.0115	.8760	.5039	.9124	.3174	.4813	.4337
8	.0133	**.8837**	.4510	**.9244**	.2874	.4463	.4060

Table 7. Fuzzy clustering results on the wine data set (3 classes), processed with fuzzy c-means clustering. All measures were computed with the nil-potent minimum for the t-norm(s).

	partition matrix			coincidence matrix			
#	diff	acc	F1	Rand	Jacc.	Folkes	Hubert
2	.0054	.7395	**.9581**	.7321	**.5419**	**.7023**	.4589
3	**.0037**	**.7695**	.9305	.8109	.4561	.6262	**.4997**
4	.0260	.7153	.7099	.8430	.2819	.4388	.3477
5	.0231	.7471	.6421	.8794	.2344	.3789	.3123
6	.0254	.7730	.5537	.9046	.2078	.3433	.2921
7	.0279	.7891	.4587	.9225	.1683	.2883	.2477
8	.0285	.8092	.4075	**.9328**	.1442	.2534	.2187

Table 8. Fuzzy clustering results on the wine data set (3 classes), processed with Gustafson–Kessel clustering. All measures were computed with the nil-potent minimum for the t-norm(s).

5 Conclusions

In this paper I transferred resampling ideas that have been used in classical crisp clustering to fuzzy clustering and introduced the mean squared difference as a simple, but effective measure for comparing fuzzy and probabilistic partition matrices. In addition, I explored the influence of different t-norms, which can be used to combine membership degrees and coincidence matrix entries. As the experiments show, the resampling approach is applicable to fuzzy clustering, but one has to be careful which relative cluster evaluation measure to choose and how to parameterize it: not all measures that work with crisp clustering also work with fuzzy clustering. The best results are obtained with the Hubert index, parameterized with either the nil-potent minimum or the Łukasiewicz t-norm to combine the membership degrees. A close competitor, which has the advantage of being simple and straightforward, is a direct comparison of the partition matrices based on the mean squared difference.

References

1. J.C. Bezdek and N. Pal. *Fuzzy Models for Pattern Recognition.* IEEE Press, New York, NY, USA 1992
2. J.C. Bezdek, J. Keller, R. Krishnapuram, and N. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing.* Kluwer, Dordrecht, Netherlands 1999

3. C.L. Blake and C.J. Merz. *UCI Repository of Machine Learning Databases*. University of California, Irvine, CA, USA 1998
http://www.ics.uci.edu/~mlearn/MLRepository.html
4. C. Borgelt. *Prototype-based Classification and Clustering*. Habilitation thesis, University of Magdeburg, Germany 2005
5. A.P. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society (Series B)* 39:1–38. Blackwell, Oxford, United Kingdom 1977
6. B. Efron and R.J. Tibshirani. An Introduction to the Boostrap. Chapman & Hall, London, United Kingdom 2003
7. B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman & Hall, London, United Kingdom 1981
8. P. Good. *Resampling Methods*. Springer-Verlag, New York, NY, USA 1999
9. E.E. Gustafson and W.C. Kessel. Fuzzy Clustering with a Fuzzy Covariance Matrix. *Proc. of the IEEE Conf. on Decision and Control (CDC 1979, San Diego, CA)*, 761–766. IEEE Press, Piscataway, NJ, USA 1979. Reprinted in [1], 117–122
10. M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering Validity Checking Methods: Part I. *ACM SIGMOD Record* 31(2):40–45. ACM Press, New York, NY, USA 2002
11. M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering Validity Checking Methods: Part II. *ACM SIGMOD Record* 31(3):19–27. ACM Press, New York, NY, USA 2002
12. J.A. Hartigan and M.A. Wong. A k-means Clustering Algorithm. *Applied Statistics* 28:100–108. Blackwell, Oxford, United Kingdom 1979
13. F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis*. J. Wiley & Sons, Chichester, England 1999
14. A.K. Jain and J. Moreau. Bootstrap Technique in Cluster Analysis. *Pattern Recognition* 20:547–569. Pergamon Press, Oxford, United Kingdom 1986
15. M.H.C. Law and A.K. Jain. *Cluster Validity by Bootstrapping Partitions*. Technical Report MSU-CSE-03-5, Dept. of Computer Science and Engineering, Michigan State University, Michigan, , USA 2003
16. E. Levine and E. Domany. Resampling Method for Unsupervised Estimation of Cluster Validity. *Neural Computation* 13:2573–2593. MIT Press, Cambridge, MA, USA 2001
17. S. Lloyd. Least Squares Quantization in PCM. *IEEE Trans. on Information Theory* 28:129–137. IEEE Press, Piscataway, NJ, USA 1982
18. C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ, USA 1982
19. C.J. van Rijsbergen. *Information Retrieval*. Butterworth, London, United Kingdom 1979
20. V. Roth, T. Lange, M. Braun, and J.M. Buhmann. A Resampling Approach to Cluster Validation. *Proc. Computational Statistics (CompStat'02, Berlin, Germany)*, 123–128. Springer-Verlag, Heidelberg, Germany 2002