A Feature Ranking Technique Based on Interclass Separability for Fuzzy Modeling

Domonkos Tikk
Budapest University of Technology and Economics
Dept. of Telecommunications and Media Informatics
H-1117 Budapest, Magyar Tudósok krt 2., Hungary
tikk@tmit.bme.hu

Kok Wai (Kevin) Wong
Murdoch University
School of Information Technology
South Street, Murdoch, 6150 W.A., Australia
K.Wong@murdoch.edu.au

***Abstract*— This paper presents a modified feature ranking method based on interclass separability for fuzzy modeling. Existing feature selection/ranking techniques are mostly suitable for classification problems. These techniques result in a ranking of the input feature or variables. Our modification exploits an arbitrary fuzzy clustering of the model output data. Using these output clusters, similar feature ranking methods can be used as for classification, where the membership in a cluster (or class) will no longer be crisp, but a fuzzy value determined by the clustering. We propose an iterative algorithm to determine the feature ranking by means of different criterion functions. We examined the proposed method and the criterion functions through a comparative analysis.**

I. Introduction

It is well-known that fuzzy systems have exponential time and space complexity in terms of N, the number of variables [1]. The number of rules in a rule base increases exponentially with N. Thus the resulting model of the system is very large. In practice, if N exceeds the experimental limit of about 6 variables, the rule based system becomes intractable. Due to this fact, rule base reduction emerged as an important research field in the past decade including e.g. the topics of fuzzy rule interpolation methods (see e.g. [1], [2], [3]), hierarchical reasoning techniques [4], [5], and other rule base reduction methods [6], [7].

If the design of the modeled system is based on input-output data samples, a possible method of rule base reduction is the omission of those variables which have no relevant effect on the output. In pattern recognition and classification such methods are called feature selection [9]. Henceforth, when we use the terms feature or variable in this paper, we refer to the same notion. In these contexts the output of sample data usually indicates from among a finite number of classes or clusters, which one the actual sample belongs to. Practically it means that the outputs are selected from a finite set of labels or, equivalently, from a closed range of natural numbers.

Feature selection methods are of two main types: feature selection and ranking methods. The methods of the former type determine which input features are relevant in the given model, whilst the ones of the latter type result in a rank of importance, and for feature selection it can be decided how many to select from the head of the rank, e.g. by a trial-and-error procedure. In this paper we modify the interclass separability feature ranking method. (The origin of this method is attributed to [10], who first applied the interclass distance concept to feature selection and extraction problems. Therefore this method is also known as Fischer's interclass separability method.)

By fuzzy modeling we mean the design of a fuzzy system from a set of input-output data samples, where the sample data values (including the output) are real numbers. The methods used in classification problems need to be modified (or the rule base has to be preprocessed) in fuzzy modeling when the range of the output is theoretically continuous. (We remark that in practice the range is discrete due to the accuracy of computers' representation ability for real numbers, however, if one would scale the range to this accuracy using as many clusters as many represented real numbers exist in the given range, the problems again became intractable and computer system dependent.) Therefore we have to somehow group the outputs of the data in order to make use of feature ranking/selection methods.

An obvious way to group is by clustering the output using some fuzzy clustering technique. A fuzzy clustering method divide the clustered space into various regions, called clusters, and determines a vector of membership degrees for each input, which contains the amount to which a particular input data belongs to every cluster. The optimal number of clusters is determined by means of an objective function. In our model we used fuzzy c-means (FCM) clustering [11], but other fuzzy clustering methods are also suitable for this purpose (e.g. subtractive clustering [12]). Because we cluster only the one dimensional output the shape of the clusters (e.g. spherical or ellipsoid) is irrelevant. We exploit only that property of FCM clustering which guarantees that membership degrees are normalized, i.e. $\sum_{i=1}^{C} \mu_{ij} = 1$ holds for all clusters $(j = 1, \ldots, n)$, and n is the number of sample data. For the algorithm of FCM clustering see the Appendix A.

We develop an interclass separability based feature ranking method for fuzzy modeling. The main algorithm is described in section III. In section IV we analyze the proposed method on some sample data sets.

II. Feature selection methods in fuzzy application

Fuzzy modeling based on the clustering of output data was first proposed by Sugeno and Yasukawa [13]. For reducing

1-4244-1146-7/07/$25.00 ©2007 IEEE.

the number of inputs they used the regularity criterion (RC) method [14]. RC creates a tree structure from the variables, where the nodes represent particular subsets of the entire variable set. The nodes are evaluated according to an objective function, and the evaluation process stops if a local minimum is found. We are also working on the automatic design of fuzzy modeling systems but we found the RC method unreliable: it is very sensitive to its parameters [15], therefore we decided to look for an alternative solution. Another deficiency of RC that it uses the fuzzy model itself to evaluate the nodes in the searching tree. Therefore, a preliminary fuzzy model should be built in advance.

Another solution was proposed to solve this problem by Costa Branco *et al* [16]. They used the principal component analysis (PCA) method [17] for identifying the important variables in the fuzzy model of an electro-mechanical system. The PCA method transforms the input data matrix of (possibly highly) correlated variables to an orthogonal system, where the variables become uncorrelated. From the transformed system the variables having eigenvalues under a certain threshold can be omitted. However, by this transformation the meaning of the variables and hence the direct linguistic interpretability of the system is lost, which we consider one of the most important features of a fuzzy system.

In the field of fuzzy classification another group of feature selection algorithms has been applied successfully [18], [9], [19] which are based on the interclass separability criterion. Let us briefly describe this method.

Let us take a given input $\{\underline{x}_1, \ldots, \underline{x}_n\}$ and the corresponding output $\{y_1, \ldots, y_n\}$ data set. $\underline{x}_i$ $(j = 1, \ldots, n)$ are N-dimensional vectors, i.e. N is the number of variables, or features. Let the matrix $\mathbf{X}$ be formed by the vectors $\underline{x}_j$ $(j = 1, \ldots, n)$. The inputs should be categorized into classes $\mathcal{C}_i$ $(i = 1, \ldots, C)$ which possess *a priori* class probability P_i, and the cardinality of the classes is $|\mathcal{C}_i| = n_i$. Let $\mathbf{X}' = \{\underline{x}'_1, \ldots, \underline{x}'_n\}$ be generated by a feature selection technique from $\mathbf{X}$, where $\underline{x}'_i$ are N'-dimensional vectors $N' < N$. $\mathbf{X}'$ is generated by deleting some $N - N'$ rows of $\mathbf{X}$. A criterion function for ranking the features is defined as [20]

$$J(\mathbf{X}') = \frac{1}{2} \sum_{i=1}^{C} P_i \sum_{j=1}^{C} P_j \frac{1}{n_i n_j} \sum_{k=1}^{n_i} \sum_{\ell=1}^{n_j} d(\underline{x}'_{ik}, \underline{x}'_{j\ell}) \quad (1)$$

which is the average distance between the elements of C classes. Here $\underline{x}'_{ik}$, and $\underline{x}'_{j\ell}$ $(k = 1, \ldots, n_i;\ \ell = 1, \ldots, n_j)$ are the elements from the ith and jth class, respectively, and $d(\underline{x}'_{ik}, \underline{x}'_{j\ell})$ denotes a distance metric, usually the square of the Euclidean norm, which is

$$d(\underline{x}'_k, \underline{x}'_\ell) = \sum_{j=1}^{N'} (x'_{kj} - x'_{\ell j})^2 = (\underline{x}'_k - \underline{x}'_\ell)^T (\underline{x}'_k - \underline{x}'_\ell). \quad (2)$$

where T denotes matrix transpose.

When the *a priori* class probabilities, P_i, are not known, they can be estimated from the occurrence of the patterns as

$$\tilde{P}_i = \frac{n_i}{n}. \quad (3)$$

Therefore, by using (3), the expression (1) becomes

$$J_1(\mathbf{X}') = \frac{1}{2n^2} \sum_{i=1}^{C} \sum_{j=1}^{C} \sum_{k=1}^{n_i} \sum_{\ell=1}^{n_j} d(\underline{x}'_{ik}, \underline{x}'_{j\ell}) = \frac{1}{2n^2} \sum_{k=1}^{n} \sum_{\ell=1}^{n} d(\underline{x}'_k, \underline{x}'_\ell). \quad (4)$$

Introducing

$$\underline{v}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} \underline{x}'_{ik}, \quad (5)$$

the mean of the vectors in the ith class, and

$$\underline{v} = \sum_{i=1}^{C} P_i \underline{v}_i, \quad (6)$$

the mixture sample mean (i.e. the mean of the centers), $J(\mathbf{X}')$ can be expressed after some algebraic manipulation as (see Appendix B):

$$J_1(\mathbf{X}') = \mathrm{tr}(\mathbf{Q}_w) + \mathrm{tr}(\mathbf{Q}_b) \quad (7)$$

where "tr" denotes the trace of a matrix, the sum of the diagonal elements, and

$$\mathbf{Q}_w = \sum_{i=1}^{C} P_i \frac{1}{n_i} \sum_{k=1}^{n_i} (\underline{x}'_{ik} - \underline{v}_i)(\underline{x}'_{ik} - \underline{v}_i)^T \quad (8)$$

$$\mathbf{Q}_b = \sum_{i=1}^{C} P_i (\underline{v}_i - \underline{v})(\underline{v}_i - \underline{v})^T. \quad (9)$$

Here $\mathbf{Q}_w$ is the within class, and $\mathbf{Q}_b$ is the between class scatter matrices. Intuitively, for the feature ranking/selection task we prefer to maximize $\mathrm{tr}(\mathbf{Q}_b)$ and at the same time minimize $\mathrm{tr}(\mathbf{Q}_w)$. It can be obtained by maximizing (7), however, in this case the effect of within class distance of samples is unchecked. Therefore, the magnitude of the criterion function (7) is not a good indicator of class separability. A more realistic criterion function to maximize is

$$J_2(\mathbf{X}') = \frac{\mathrm{tr}(\mathbf{Q}_b)}{\mathrm{tr}(\mathbf{Q}_w)} \quad (10)$$

which reflects more to the described intuitive notion. For more details see [9], [20].

The proper modification of this technique for fuzzy modeling is presented in the next section.

III. The feature ranking on fuzzy clustered output (FRFCO) algorithm

Consider again the data set defined in the previous section. For brevity let $\mathcal{F}$ denote the set of all features. Let us cluster the output space by the fuzzy c-means clustering algorithm with parameter m (usually 2). Let the optimal number of clusters be C. The membership degree μ_{ij} $(i = 1, \ldots, C;\ j = 1, \ldots, n)$ denotes the degree by which the data $\underline{x}_j$ belongs to cluster i. Note that $\sum_{i=1}^{C} \mu_{ij} = 1$. With the following algorithm we can rank the input variables in order to find and omit the irrelevant ones.

As shown above, interclass separability criterion is based on the *fuzzy between-class* (11) and the *fuzzy within-class* (12) *scatter matrices* that sum up to the *total fuzzy scatter matrix* (13). Scatter matrices are also known as covariance matrices. These matrices can be defined in our context as

$$\mathbf{Q}_b = \sum_{i=1}^{C}\sum_{j=1}^{n} \mu_{ij}^m (\underline{v}_i - \underline{v})(\underline{v}_i - \underline{v})^T \tag{11}$$

$$\mathbf{Q}_i = \frac{1}{\sum_{j=1}^{n} \mu_{ij}^m} \sum_{j=1}^{n} \mu_{ij}^m (\underline{x}_j - \underline{v}_i)(\underline{x}_j - \underline{v}_i)^T$$

$$\mathbf{Q}_w = \sum_{i=1}^{C} \mathbf{Q}_i = \sum_{i=1}^{C} \frac{1}{\sum_{j=1}^{n} \mu_{ij}^m} \sum_{j=1}^{n} \mu_{ij}^m (\underline{x}_j - \underline{v}_i)(\underline{x}_j - \underline{v}_i)^T \tag{12}$$

$$\mathbf{Q}_t = \mathbf{Q}_b + \mathbf{Q}_w = \sum_{i=1}^{C} \frac{1}{\sum_{j=1}^{n} \mu_{ij}^m} \sum_{j=1}^{n} \mu_{ij}^m (\underline{x}_j - \underline{v})(\underline{x}_j - \underline{v})^T \tag{13}$$

where

$$\underline{v}_i = \frac{1}{\sum_{j=1}^{n} \mu_{ij}^m} \sum_{j=1}^{n} \mu_{ij}^m \underline{x}_j \tag{14}$$

are the fuzzy centers of the ith cluster, and

$$\underline{v} = \frac{1}{n} \sum_{j=1}^{n} \underline{x}_j \tag{15}$$

are the averages of clustered data. Here we assume that matrices $\mathbf{Q}_i$ and $\mathbf{Q}_w$ are nonsingular.

The feature interclass separability criterion is a trade-off between $\mathbf{Q}_b$ and $\mathbf{Q}_w$ as described above. Observe that fuzzy scatter matrices (11) and (12) are generalizations of the crisp scatter matrices (9) and (8), respectively. The class probability P_i is replaced by the normalized membership degrees of the sample vectors. Further, every sample vector participates in the scatter matrices of the clusters to the extent of its membership degree in the given cluster. Needless to say, that the choice of the criterion function is problem dependent. Yet often, certain criterion functions possess better characteristic than others, therefore we present the following algorithm with the best considered one. We will examine the effect of different criterion functions in section IV.

A. Evaluation using determinants of matrices

The proposed feature ranking algorithm proceeds iteratively. In each iteration it determines the most important variable based on the interclass separability criterion function as follows. Let us delete temporarily the variable f ($f \in \mathcal{F}$), i.e. $\mathbf{X}' = \mathbf{X}|_f$, where the latter is obtained by deleting the fth row from $\mathbf{X}$. Calculate the matrices $\mathbf{Q}_b(\mathbf{X}')$ and $\mathbf{Q}_w(\mathbf{X}')$ for the remaining data. One possible criterion function is the ratio of their determinants

$$J_3(\mathbf{X}') = J_3(\mathbf{X}|_f) = \det(\mathbf{Q}_b(\mathbf{X}'))/\det(\mathbf{Q}_w(\mathbf{X}')). \tag{16}$$

Repeat this procedure for all the variables in $\mathcal{F}$. The expression $\min_{f\in\mathcal{F}} J_3(\mathbf{X}|_f)$ attains its minimum when the deviation between $\mathbf{Q}_b$ and $\mathbf{Q}_w$ is the least, e.g. when the most important variable is omitted. Then omitting $f \in \mathcal{F}$ permanently, we can restart the algorithm with the new feature set.

The FRFCO algorithm

1) Let $\mathcal{F} := \{1, \ldots, N\}$.
2) For all $f \in \mathcal{F}$
   a) Let $\mathcal{F} := \mathcal{F} - \{f\}$ and also update matrix $\mathbf{X}$, and vectors $\underline{v}_i$ and $\underline{v}$ by deleting temporarily its fth row or element.
   b) Calculate matrices $\mathbf{Q}_b(\mathbf{X})$, $\mathbf{Q}_w(\mathbf{X})$ and determine $J_3(\mathbf{X}')$.
3) Let $f' = \operatorname{argmin}_{f\in\mathcal{F}} J_3(\mathbf{X}|_f)$, i.e. where J_3 attains its minimal value. Delete permanently the variable(s) f' from $\mathcal{F}$ the corresponding columns from $\mathbf{X}$, $\underline{v}_i$ and $\underline{v}$. Note that f' can contain more than one variable.
4) If $|\mathcal{F}| > 1$ then back to step 2, else stop.

The order of the deleted variables gives their rank of importance.

B. Other criterion functions

Beside (16), other possible criterion functions can be used. One reason to modify J_3 is that it may have negative values which may give inconsistent results. Further, the value of J_3 can vary from very small to very large. This phenomenon can cause problems, as, on one side, a very small value can turn out to be zero if its order attains the accuracy of the underlying computer system, while on the other side, a very large value may result in loss of information, when rounding is involved. To overcome these possible drawbacks other criterion functions can be used.

A natural solution for the negative values can be the absolute value of J_3:

$$J_4(\mathbf{X}') = |\det(\mathbf{Q}_b(\mathbf{X}'))/\det(\mathbf{Q}_w(\mathbf{X}'))|. \tag{17}$$

If the determinants are negative, the FRFCO algorithm need to be modified when criterion function J_4 is used, by deleting the feature corresponding to the *maximal* value in each iteration. However, when (16) has both negative and positive values in an iteration this can result in inconsistent results.

Another solution can be the use of the trace function, which alleviates both aforementioned disadvantages of J_3:

$$J_2(\mathbf{X}') = \operatorname{tr}(\mathbf{Q}_b(\mathbf{X}'))/\operatorname{tr}(\mathbf{Q}_w(\mathbf{X}')). \tag{18}$$

Notice that the trace of the truncated fuzzy scatter matrices is always nonnegative, because the diagonal elements contain the square of distances. As pointed out in [9], the main drawback of the criterion function (18) is, if for a particular feature subset a class $\mathcal{C}_i$ is well scattered and a portion of $\mathcal{C}_i$ is overlapped with another class $\mathcal{C}_j$ but their centers are far away, then J_2 may be greater then for another feature subset, which separates the two classes in such a way that a single hyperplane may pass between them, but their centers are not so far apart. However, we cluster only the output, so this phenomenon may not occur in our case.

The use of these functions does not require other modifications of the FRFCO algorithm. Their effect on the ranking is compared with the function (16) in section IV.

We remark that the normalization of the input values can also solve the problem of computer system accuracy. The effect of this transformation on the ranking is out of the scope of this paper.

It is also worth investigating how the ranking changes if we eliminate the worst feature at each iteration (backward feature selection, BFS). One would expect the inverse ranking order if the features are uncorrelated. This BFS heuristic elimination method is also analyzed in the next section.

IV. The comparative analysis of FRFCO

We applied the proposed method to two data sets. We compare our results to two other methods: the RC method [14] using a fixed setting and input contribution measure (ICM) analyzer (similar to sensibility analysis) [21]. RC divides the data into two groups. For this, we first ordered the data based on their output value, then put them in group A and alternately in group B according to this ordering. Here we remark again that the result obtained by RC is unstable as the RC is very sensitive to its parameters, i.e. how the two groups are selected and how the fuzzy sub-models are built for the data [15]. The ICM makes use of a BPNN for training by using the given data set. It then varies the input variables one at a time to their minimum and maximum limit. The ICM of the input variable is then measured based on the effect of the input variable to the output.

According to [22], some authors determines the fuzzy scatter matrices (expressions (11)–(13)) with fuzzy exponent $m = 1$. We remark below when the change of this parameter modifies the ranking. Otherwise, if it is not explicitly expressed, the fuzzy exponent has the default value $m = 2$.

A. Simple synthetic data set

First, we checked whether FRFCO gives consistent result on the synthetic and real data sets given in [13]. The synthetic data set contained 50 samples with 4 variables. The first two variables were obtained from the function

$$y = f(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5}), \qquad 1 \le x_1, x_2 \le 5$$

and the last two were chosen randomly. The optimal number of clusters is 6. On this synthetic sample our proposed method gives the correct ranking by using function (16) as the criterion function: $\langle 2, 1, 3, 4\rangle$. In this case the determinants are all positive in each iteration. Table I shows the results obtained by FRFCO with other criterion functions and by other methods.

The first and fourth methods, as well as the second and third methods in Table I are identical because all the determinants are positive. The second and the third methods do not give correct rankings. Surprisingly, the first variable gives the worst result at the first iteration, while the order of the remaining variables is correct. This is because the random variables are well correlated with the second variable and not with the first. The criterion function J_2 also finds the correct ranking, and gives the reverse order ranking correctly when the BFS heuristic used. When we used $m = 1$ in the fuzzy scatter matrices, only the rankings using J_2 changed: $\langle 2, 3, 1, 4\rangle$.

B. Real data set of a chemical plant

The second sample data set was the model of a chemical plant with 5 inputs [13]. The inputs were the following: x_1 – monomer concentration, x_2 – change of monomer concentration, x_3 – monomer flow rate, x_4 and x_5 – local temperature inside the plant. The output was the set point for monomer flow rate. 70 sample data were provided.

In [13] the first three variables were found important by means of the RC method. We have to admit that we could not generate this result with our RC implementation regardless of the applied parameter settings [15]. According to the ICM analyzer the third variable is the most important, then the first, while the remaining three were considered irrelevant. These results are compared with the FRFCO method in Table II. The optimal number of clusters is again 6. All the determinants are positive during the elimination process.

The FRFCO gives the same result with all the criterion functions, and this also coincides with other techniques. Because the determinants are positive the rankings obtained with J_4 do not differ from the one obtained with J_3. When BFS heuristic is used, J_3 identifies the three worst and the two best variables, but permutes the order of the worst three. This is not very significant, because their contributions are similar and very low according to the ICM analyzer. With this heuristic J_2

TABLE I

The ranking of 4 input features (synthetic data set, [13])

method	Ranking	Remarks
FRFCO with J_3	$\langle 2, 1, 3, 4\rangle$	
FRFCO with J_4	$\langle 1, 4, 3, 2\rangle$	BFS
FRFCO with J_3	$\langle 1, 4, 3, 2\rangle$	BFS (reverse order expected)
FRFCO with J_4	$\langle 2, 1, 3, 4\rangle$	
FRFCO with J_2	$\langle 2, 1, 3, 4\rangle$	
FRFCO with J_2	$\langle 4, 3, 1, 2\rangle$	BFS (reverse order expected)
ICM with 4–8–1 architecture	$\langle 2, 1, 3, 4\rangle$	the contribution of each variable is: $\langle 67.85, 29.57, 1.79, 0.79\rangle$
RC	$\langle 1, 2\rangle$	Automatically pruned[1]

[1] The RC method automatically prunes the irrelevant variables

TABLE II

The ranking of 5 input features (chemical plant data set, [13])

method	Ranking	Remarks
FRFCO with J_3	$\langle 3, 1, 4, 5, 2\rangle$	
FRFCO with J_4	$\langle 2, 4, 5, 1, 3\rangle$	BFS
FRFCO with J_3	$\langle 2, 4, 5, 1, 3\rangle$	BFS (reverse order expected)
FRFCO with J_4	$\langle 3, 1, 4, 5, 2\rangle$	
FRFCO with J_2	$\langle 3, 1, 5, 2, 4\rangle$	
FRFCO with J_2	$\langle 1, 4, 5, 2, 3\rangle$	BFS (reverse order expected)
ICM with 5–10–1 architecture	$\langle 3, 1, 2, 5, 4\rangle$	the contribution of each variable is: $\langle 75.94, 22.17, 1.10, 0.78, 0.01\rangle$
RC in [13]	$\langle 3, 2, 1\rangle$	Automatically pruned[1]
RC	$\langle 3\rangle$	Automatically pruned[1]

[1] The RC method automatically prunes the irrelevant variables

fails to produce the correct reverse order. In this case, before the first deletion the difference between the first and the last two variables (according to J_2) is insignificant and only due to rounding error of the computer system used. We remark that this value coincides with the one obtained by our RC implementation.

C. The complexity issue

We remark that the required time for running the FRFCO algorithm is quite low. One iteration step needs time proportional to N', where N' is the cardinality of the actual feature set. In total it is less than $N(N-1)/2$ (multiple deletions can occur), so it is proportional with $O(N^2)$. For the most complicated data set the longest evaluation with the full feature set took a few seconds, and the overall time is far under 1 min.

On the other hand, the training time of the 12–24–1 architecture BPNN networks was substantially longer.

V. Summary and hints for use

The best results were obtained in all the three examples with the use of the criterion function J_3. If the determinants are of both signs then the use of absolute value function can improve its performance. When the total number of features is low the J_2 criterion function also provides good results.

As the ranking does not specify how many features to use, the easiest way is to try it experimentally. For this we can start to build up fuzzy rule base models (e.g. with Sugeno and Yasukawa's method [13]) with a small number of top ranked features (one or two). Then they should be evaluated by a proper performance index function. If a local optimum is reached according to this index the fuzzy model structure is accepted. Finally, some parameter identification or tuning algorithms can be applied to improve the performance of the final model (see [19], [13]).

VI. Conclusion

We proposed in this paper a feature ranking algorithm adopted for fuzzy modeling with output from a continuous range. The main idea is to cluster the output data and to use the cluster-membership degrees as weights in the feature ranking method. Several criterion functions were proposed for determining the ranking. We applied our method to real world and synthetic data sets, and it was likely to find the proper or close-to-proper ranking. The complexity behavior of the algorithm was also addressed, and was found very advantageous. Finally, some hints for its use were presented.

Appendix

A. Fuzzy c-means clustering algorithm

Let us take a given $\{\underline{x}_1, \ldots, \underline{x}_n\}$ set of vectors which should be classified into a certain number of clusters. Fuzzy clustering assigns a membership grade μ_{ij} to every vector $\underline{x}_j$ $(j = 1, \ldots, n)$ for every cluster i $(i = 1, \ldots, C)$, where C is the number of clusters. We assume that

$$\sum_{i=1}^{c} \mu_{ij} = 1 \quad \forall j \in [1, n]$$

and we define a matrix $\mathbf{U}$ consisting of μ_{ij}. Our goal is to find an optimal C (according to an objective function) and determine the matrix $\mathbf{U}$. In the algorithm m is an adjustable parameter which may vary in the range of $[1.5, 3]$, and is set to 2 as default. The algorithm proposed by Bezdek is the following [11]:

1. Fix C, the number of clusters, set $\ell = 1$, and initialize $\mathbf{U}$ with $\mathbf{U}^{(1)}$.

2. Calculate the centers $\underline{v}_i$ of the fuzzy clusters as

$$\underline{v}_i = \sum_{j=1}^{n} (\mu_{ij})^m \underline{x}_j \Big/ \sum_{j=1}^{n} (\mu_{ij})^m \quad \forall i \in [1, C].$$

The distance of the jth vector from the ith cluster center is defined by

$$d_{ij} = \|\underline{x}_j - \underline{v}_i\|.$$

3. Calculate the new $\mathbf{U}^{(l)}$ for $\ell := \ell + 1$ as

$$\begin{aligned} I_j &:= \{i | 1 \le i \le C,\, d_{ij} = 0\} \\ \tilde{I}_j &:= \{1, \ldots, C\} - I_j \\ I_k &= \emptyset \Longrightarrow \mu_{ij} = \frac{1}{\sum_{k=1}^{C} (d_{ij}/d_{kj})^{2/(m-1)}} \\ I_k &\neq \emptyset \Longrightarrow \mu_{ij} = 0 \ \forall i \in \tilde{I}_j;\ \mu_{ij} = \frac{1}{|I_j|} \ \forall i \in I_j. \end{aligned}$$

4. If $\|\mathbf{U}^{(\ell-1)} - \mathbf{U}^{(\ell)}\| \le \varepsilon$, where ε is a prescribed error, then stop; otherwise go to step 2.

B. Derivation of (7) from (1) [20]

Substituting (2) into (1) and using (5) and (6) it is easy to show that the average distance between points in the training set is,

$$J_1(\mathbf{X}') = \sum_{i=1}^{C} P_i \left[\left(\frac{1}{n_i} \sum_{k=1}^{n_i} (\underline{x}'_{ik} - \underline{v}_i)^T (\underline{x}'_{ik} - \underline{v}_i) \right) + (\underline{v}_i - \underline{v})^T (\underline{v}_i - \underline{v}) \right]. \tag{19}$$

Here the first term represents the average distance of elements belonging to class c_i from the ith class sample mean vector. The second term of (19) represents the distance of the ith class mean vector from the mixture mean $\underline{v}$. The weighted sum of these terms over the all the classes, in fact, the average distance between the class conditional mean vector:

$$\sum_{i=1}^{C} P_i (\underline{v}_i - \underline{v})^T (\underline{v}_i - \underline{v}) = \frac{1}{2} \sum_{i=1}^{C} P_i \sum_{j=1}^{C} P_j (\underline{v}_i - \underline{v}_j)^T (\underline{v}_i - \underline{v}_j).$$

Then, by the definition of quantities (8) and (9), we obtain immediately (7).

References

[1] L. T. Kóczy and K. Hirota, "Size reduction by interpolation in fuzzy rule bases," *IEEE Trans. on SMC*, vol. 27, pp. 14–25, 1997.

[2] P. Baranyi and L. T. Kóczy, "A general and specialized solid cutting method for fuzzy rule interpolation," *BUSEFAL*, vol. 67, pp. 13–22, 1996.

[3] D. Tikk and P. Baranyi, "Comprehensive analysis of a new fuzzy rule interpolation method," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 3, pp. 281–296, 2000.
[4] L. T. Kóczy and K. Hirota, "Approximate inference in hierarchical structured rule bases," in *Proc. of 5th IFSA World Congress (IFSA'93)*, Seoul, 1993, pp. 1262–1265.
[5] M. Sugeno, M. F. Griffin, and A. Bastian, "Fuzzy hierarchical control of an unmanned helicopter," in *Proc. of the 5th IFSA World Congress (IFSA'93)*, Seoul, 1993, pp. 1262–1265.
[6] P. Baranyi, A. Martinovics, D. Tikk, L. T. Kóczy, and Y. Yam, "A general extension of fuzzy SVD rule base reduction using arbitrary inference algorithm," in *Proc. of IEEE Int. Conf. on System Man and Cybernetics (IEEE-SMC'98)*, San Diego, USA, 1998, pp. 2785–2790.
[7] P. Baranyi and Y. Yam, "Fuzzy rule base reduction," in *Fuzzy IF-THEN Rules in Computational Intelligence: Theory and Applications*, D. Ruan and E. E. Kerre, Eds., pp. 135–160. Kluwer, 2000.
[8] J. Bruinzeel, V. Lacrose, A. Titli, and H. B. Verbruggen, "Real time fuzzy control of complex systems using rule-base reduction methods," in *Proc. of the 2nd World Automation Congress (WAC'96)*, Montpellier, France, 1996.
[9] R. K. De, N. R. Pal, and S. K. Pal, "Feature analysis: Neural network and fuzzy set theoretic approaches," *Pattern Recognition*, vol. 30, no. 10, pp. 1579–1590, 1997.
[10] R. A. Fischer, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.
[11] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
[12] J-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River, NJ, 1997.
[13] M. Sugeno and T. Yasukawa, "A fuzzy logic based approach to qualitative modelling," *IEEE Trans. on Fuzzy Systems*, vol. 1, no. 1, pp. 7–31, 1993.
[14] J. Ihara, "Group method of data handling towards a modeling of complex system IV," *Systems and Control*, vol. 24, pp. 158–168, 1980, (In Japanese).
[15] D. Tikk, T. D. Gedeon, L. T. Kóczy, and Gy. Biró, "Implementation details of problems in Sugeno and Yasukawa's qualitative modelling," Research Working Paper RWP-IT-01-2001, School of Information Technology, Murdoch University, Perth, W.A., 2001, p. 17.
[16] P. J. Costa Branco, N. Lori, and J. A. Dente, "New approaches on structure identification of fuzzy models: Case study in an electromechanical system," in *Fuzzy Logic, Neural Networks, and Evolutionary Computation*, T. Furuhashi and Y. Uchikawa, Eds., pp. 104–143. Springer-Verlag, Berlin, 1996.
[17] D. Kleinbaum, L. L. Kupper, and K. E. Muller, *Applied Regression Analysis and Other Multivariable Methods*, PWS-Kent, Boston, Mass., 2nd edition, 1988.
[18] S. Abe, R. Thawonmas, and Y. Kobayashi, "Feature selection by analyzing class regions approximated by ellipsoids," *IEEE Trans. on SMC, Part C*, vol. 28, no. 2, 1998, http://www.info.kochi-tech.ac.jp/ruck/paper.html.
[19] J.A. Roubos, M. Setnes, and J. Abonyi, "Learning fuzzy classification rules from labeled data," *International Journal of Information Sciences*, submitted, July 2000, http://www.fmt.vein.hu/softcomp.
[20] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall, London, 1982.
[21] C. C. Fung, K. W. Wong, and H. Crocker, "Determining input contributions for a neural network based porosity prediction model," in *Proc. of the Eighth Australian Conference on Neural Network (ACNN97)*, Melbourne, 1997, pp. 35–39.
[22] Robert Babuška, "Fuzzy modelling for matlab," http://lcewww.et.tudelft.nl/˜crweb/software/index.html.