A Feature Ranking Algorithm for Fuzzy Modelling Problems*

Domonkos Tikk[1,3], Tamás D. Gedeon[2], and Kok Wai Wong[2]

[1] Dept. of Telecommunications & Telematics,
Budapest University of Technology and Economics,
1117 Budapest, Magyar Tudósok Körútja 2., Hungary,
e-mail: tikk@ttt.bme.hu
[2] School of Information Technology, Murdoch University
South Street, Murdoch, 6150 W.A., Australia
e-mail: tgedeon@murdoch.edu.au
[3] Intelligent Integrated Systems Japanese–Hungarian Laboratory
1111 Budapest, Műegyetem rakpart 3., Hungary

Abstract. This paper presents a feature ranking method adapted to fuzzy modelling with output from a continuous range. Existing feature selection/ranking techniques are mostly suitable for classification problems, where the range of the output is discrete. These techniques result in a ranking of the input feature (variables). Our approach 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 the application of the Sequential Backward Selection (SBS) search method to determine the feature ranking by means of different criterion functions. We examined the proposed method and the criterion functions through a comparative analysis.

1 Introduction

It is well-known that traditional fuzzy rule-based systems (FRBSs) have exponential time and space complexity in terms of N, the number of variables [24]. 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 5–8 variables [24], 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. [24,2,16,28]), hierarchical reasoning techniques [23,26], and other rule base reduction methods [3–5,7].

If the design of the modelled 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

* This research was supported by the Australian Research Council, and by the Hungarian Scientific Research Fund (OTKA) Grants No. D034614, and T34212.

use the terms feature or variable in this paper, we refer to the same notion. In these contexts, having a finite number of classes (or clusters), the output of a sample data indicates which class the sample belongs to. Practically it means that 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 a given model, whilst the ones of the latter type result in a rank of importance. Feature ranking methods can be considered as preprocession of feature selection, because relevant features can be selected by taking the first k elements of the head of the feature ranking, and then, by optimizing the number of k, e.g. by a trial-and-error procedure. We aim at providing a reliable feature ranking method for fuzzy modelling problems.

On fuzzy modelling we mean the automatic design of a fuzzy system from a set of input-output data samples, where the sample data values (including the output) are real numbers. It means that opposing classification problems, the range of the output is continuous, so feature selection/ranking algorithms developed for pattern recognition or classification task can not be applied directly. Our goal is, therefore, to bridge the gap between existing feature selection/ranking methods and fuzzy modelling by the adaptation of the referred methods.

Classical feature selection/ranking methods need to be modified or the rule base has to be preprocessed before being applied to fuzzy modelling. A straightforward solution of this problem is the grouping of output data values. An obvious way of grouping is the clustering of output using some fuzzy clustering technique. A fuzzy clustering method divides the clustered space into various regions, called clusters, and determines a vector of membership degrees for each data, which indicates the grade to which the particular data belongs to the clusters. Because we cluster only the one dimensional output, the shape of the clusters (e.g. spherical or ellipsoid) is irrelevant, due to the fact that in our case clusters are interval. In our model we used fuzzy c-means (FCM) clustering ([6]; see Subsection 3.2), but other fuzzy clustering methods are also suitable for this purpose (e.g. subtractive clustering [21] or Gustafson–Kessel algorithm [17]).

Let us now briefly summarize our proposed method: first, we cluster the output data by FCM, then we apply an appropriately modified feature selection/ranking method. We adapt the interclass separability based feature selection/ranking method for this task. (The origin of this method is attributed to [12], 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.)

This paper is organized as follows. In Section 2 we give a short overview of feature selection methods in fuzzy applications. In Section 3 we outline the interclass distance based feature selection problem in probabilistic case,

and recall the FCM method. The main algorithm is described in section 4. In section 5 we analyze the proposed method on some sample data sets.

2 Feature selection methods in fuzzy application

Fuzzy modelling based on the clustering of output data was first proposed by Sugeno and Yasukawa [27]. For reducing the number of inputs they used the regularity criterion (RC) method [20]. 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 modelling systems but we found the RC method unreliable: it is very sensitive to its parameters [29], 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 feature selection problem by Costa Branco *et al* [8]. They used the principal component analysis (PCA) method [22] for identifying the important variables in the fuzzy model of an electromechanical 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.

Hong and Chen proposed a general learning algorithm which automatically derives fuzzy if–then rules and membership functions from a set a given training examples using a decision table [18]. They improved Hong's and Lee's algorithm [19] by first selecting relevant features and building an appropriate initial membership function in order to avoid very large rule bases. This method, although shows impressing performance benchmarks, is only suitable for finding relevant attributes for classification problems.

In the field of fuzzy classification another group of feature selection algorithms has been applied successfully [1,9,25] which are based on the interclass separability criterion. Let us briefly describe this method in the probabilistic case [10].

3 Preliminaries

3.1 The idea of interclass separability based classification

Let us take a data set consisting of N-dimensional vectors $\{\underline{x}_1, \ldots, \underline{x}_n\}$. Here N is the number of variables, features, or attributes. The vectors $\underline{x}_j$ $(j =$

$1, \ldots, n$) 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 us denote the features by f_k, $k = 1, \ldots, N$, and the original feature set of all the measured N features by $\mathcal{F}_N = \{f_k | k = 1, \ldots, N\}$. Analogously, denote by $\mathcal{F}_{N'}$, $N' < N$, the feature set where $N - N'$ features are removed from $\mathcal{F}_N$. Obviously, by deleting different features, we obtain different feature sets $\mathcal{F}_{N'}$.

Let us assume that the classes occupy different regions in the multi-dimensional feature space. Intuitively, more distant the classes are from each other, the better the chances of successful classification of input vectors. It is reasonable, therefore, to select as feature space that subspace of the original N-dimensional feature space in which classes are maximally separated in terms of certain distance function. Thus, we seek a feature set $\mathcal{F}$, which maximizes the average distances of C classes.

Let the matrix $\mathbf{X}_{\mathcal{F}_N}$ be formed by the vectors $\underline{x}_j$ ($j = 1, \ldots, n$) associated with the original feature set and vectors. Similarly, we denote by $\mathbf{X}_{\mathcal{F}_{N'}} = \left[\underline{x}'_1 \ldots \underline{x}'_n\right]$ the matrix associated with feature set $\mathcal{F}_{N'}$, and consisting of vectors obtained by deleting $N - N'$ features from vectors $\underline{x}_j$ ($j = 1, \ldots, n$). A general criterion function that maximizes the interclass distance, and thus can be exploited to rank features, is defined as [10]

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

which is the average distance between the elements of C classes. Here vectors $\underline{x}'_{ik}$, and $\underline{x}'_{j\ell}$ ($k = 1, \ldots, n_i$; $\ell = 1, \ldots, n_j$) are the elements of 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

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

where T denotes matrix transpose. The optimal feature set $\mathcal{F}$ satisfies

$$J(\mathbf{X}_{\mathcal{F}}) = \max_{\mathcal{F}_{N'}} J(\mathbf{X}_{N'}). \tag{3}$$

There are various choices for the criterion function. This issue is addressed in Subsection 4.3.

In order to rank the elements of the feature set, one may apply e.g. one of the following well-known search methods: Sequential Backward Selection (SBS), or Sequential Forward Selection (SFS) search method [10]. SBS is a simple top-down search procedure where one feature at a time is deleted from the current feature set. At each stage, the attribute to be removed from the feature set is selected from among the elements of the feature set so that the

new shrunk set of features yields a minimum value of the criterion function used. SFS is the bottom-up counterpart of SBS search method, where the one feature at a time, having the largest effect on the criterion function, is added to the current feature set.

3.2 Fuzzy c-means (FCM) clustering algorithm

Let us now describe FCM algorithm proposed by Bezdek [6]. As we mentioned, we shall apply FCM method to group output values of a data set, where these values are from a continuous range.

Fuzzy clustering (in general) 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 require that

$$\sum_{i=1}^{C} \mu_{ij} = 1 \quad \forall j \in [1, n] \tag{4}$$

i.e. cluster membership degrees are normalized. We define a matrix $\mathbf{U}$ consisting of μ_{ij}. Our goal is to find an optimal C (according to an objective function) and to determine the matrix $\mathbf{U}$. Clusters are represented by their center, $\underline{v}_i$. In the presented algorithm $m > 1$ is an adjustable real valued parameter, the so-called fuzzifier, that may vary usually in the range of $[1.5, 3]$, and is set to 2 as default. The original algorithm is the following [6]:

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

$$\underline{v}_i = \sum_{j=1}^{n} (\mu_{ij})^m \underline{x}_j \Big/ \sum_{j=1}^{n} (\mu_{ij})^m \quad \forall i \in [1, C]. \tag{5}$$

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.

The resulting FCM algorithm is able to recognize spherical clusters (clouds of points) of approximately the same size. FCM clustering reaches its limits for clusters of different shapes, sizes and densities. If these attributes of the clusters are very inhomogeneous, one can substitute FCM by an alternative clustering, e.g. Gustafson–Kessel method [17]. In our application we cluster one dimensional values, therefore the shape of clusters is homogeneous (interval).

There are many studies about FCM containing various objective functions to determine the optimal number of clusters [6,11,13]. We used the one proposed in [13], and also applied in [27], in which the goal is to minimize

$$S(C) = \sum_{j=1}^{n} \sum_{i=1}^{C} (\mu_{ij})^m \left(\|\underline{x}_j - \underline{v}_i\|^2 - \|\underline{v}_i - \underline{x}\|^2 \right). \tag{6}$$

Here

$$\underline{x} = \frac{1}{n} \sum_{j=1}^{n} \underline{x}_j \tag{7}$$

denotes the averages of all input vectors.

4 The feature ranking on fuzzy clustered output (FRFCO) algorithm

4.1 Problem definition

Let us now turn to formalize our problem: feature ranking of fuzzy modelling systems. Now, the data set consists of pairs $(\underline{x}_j, y_j)$ $(j = 1, \ldots, n)$ determining the behavior of the modelled system. We intend to filter the features (variables), and keep only those ones, that have significant effect on the output. To achieve our purpose, we rank the features of $\mathcal{F}$ depending on their relevance to the determination of the output.

Let us assume that we already performed FCM clustering on the output values y_j $(j = 1, \ldots, n)$, and we obtained C as the optimal number of clusters by means of (6), and values μ_{ij} as membership grades in class $\mathcal{C}_i$ $(i = 1, \ldots, C)$, where condition (4) is satisfied. Thus, having classified the model output, instead of real values, we are now able to assign cluster membership grades to each input vector, where the membership grade μ_{ij} represents the distance (more precisely: closeness) of the original output value, y_j, from the representative cluster center v_i (see (5) with substitution $\underline{x}_j \to y_j$). Considering that y_j is the projection of the input vector $\underline{x}_j$ onto the output space, we can assign membership degree μ_{ij} to $\underline{x}_j$ as associated grade in the class $\mathcal{C}_i$.

Now, on the analogy of the class probability in the probabilistic model, we can define the *fuzzy class frequency of occurrence* of a class $\mathcal{C}_i$ as the

normalized sum of total membership grades assigned to the given class

$$P_i \;\sim\; F_i := \frac{1}{n}\sum_{j=1}^{n}\mu_{ij}$$

Observe that due to the condition (4), we have $\sum_{i=1}^{C} F_i = 1$. Similarly we can define *fuzzy class cardinality* on the analogy of class cardinality, n_i, by the following expression:

$$n_i \;\sim\; \sum_{j=1}^{n}\mu_{ij}.$$

We introduce matrices measuring between-class (interclass) and within-class (intraclass) averaged distances of input vectors $\underline{x}_j$, $(j = 1, \ldots, n)$:

$$\mathbf{Q}_b(\mathbf{X}_{\mathcal{F}}) = \sum_{i=1}^{C} F_i(\underline{v}_i - \underline{x})(\underline{v}_i - \underline{x})^T \tag{8}$$

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

$$\mathbf{Q}_w(\mathbf{X}_{\mathcal{F}}) = \sum_{i=1}^{C}\mathbf{Q}_i = \sum_{i=1}^{C}\frac{1}{\sum_{j=1}^{n}\mu_{ij}}\sum_{j=1}^{n}\mu_{ij}(\underline{x}_j - \underline{v}_i)(\underline{x}_j - \underline{v}_i)^T \tag{9}$$

$$\mathbf{Q}_t(\mathbf{X}_{\mathcal{F}}) = \mathbf{Q}_b + \mathbf{Q}_w = \sum_{i=1}^{C}\frac{1}{\sum_{j=1}^{n}\mu_{ij}}\sum_{j=1}^{n}\mu_{ij}(\underline{x}_j - \underline{x})(\underline{x}_j - \underline{x})^T \tag{10}$$

where

$$\underline{v}_i = \frac{1}{\sum_{j=1}^{n}\mu_{ij}}\sum_{j=1}^{n}\mu_{ij}\underline{x}_j \tag{11}$$

are the fuzzy centers of the ith cluster, and $\underline{x}$ is defined in (7).

Here (8) and (9) are called *fuzzy between-class (interclass)*, and *fuzzy within-class (intraclass) scatter matrices*, respectively, that sum up to the *total fuzzy scatter matrix* (10), fuzzy generalization of their crisp counterparts, see also [9]. (Scatter matrices are also known as covariance matrices.) The defined quantities are function of the matrix $\mathbf{X}_{\mathcal{F}}$ assigned to the feature set $\mathcal{F}$ in the sense that all vectors $\underline{x}_j$ contain exactly those features, that appear in the given feature set. When the context makes clear that we mean the quantities (8) and (9) associated to the feature set $\mathcal{F}$, we use simpler notation $\mathbf{Q}_b$ and $\mathbf{Q}_w$. In (9) and (10), membership grades μ_{ij} can be interpreted as a class weight of input vectors. We assume that $\mathbf{Q}_b$ and $\mathbf{Q}_w$ are nonsingular, positive semi-definite matrices.

The feature selection based on interclass separability concept is a trade-off between $\mathbf{Q}_b$ and $\mathbf{Q}_w$. Intuitively, the classes are well-separated, if the average interclass distance, $\mathbf{Q}_w$, is large, while the average intraclass distance, $\mathbf{Q}_b$, is

low. Therefore, an appropriate solution to this problem is to find the feature set $\mathcal{F}$ which simultaneously maximizes (9) and minimizes (8).

Based on these quantities we can define appropriate criterion functions in Subsection 4.3. 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 will examine the effect of different criterion functions in Section 5, and introduce the FRFCO algorithm with a general criterion function.

4.2 The proposed algorithm

The proposed feature ranking algorithm is an instance of SBS search method. In our application, SBS method applies the interclass separability criterion function. In the ith step we delete temporarily a variable f_{temp} ($f_{\text{temp}} \in \mathcal{F}_{k-1}$), so we have feature set $\mathcal{F}_k = \{f | f \in \mathcal{F}_{k-1}, f \neq f_{\text{temp}}\}$ and input matrix $\mathbf{X}_{\mathcal{F}_k}$, where the starting feature set is $\mathcal{F}_0 = \mathcal{F}_N$, and we calculate the matrices $\mathbf{Q}_b(\mathbf{X}_{\mathcal{F}_k})$ and $\mathbf{Q}_w(\mathbf{X}_{\mathcal{F}_k})$ to be used in the criterion functions. This procedure is repeated for all the variables in $\mathcal{F}_{k-1}$. By means of an appropriate criterion function, the expression $\min\limits_{f \in \mathcal{F}_{k-1}} J(\mathbf{X}_{\mathcal{F}_k})$ attains its minimum when the deviation between $\mathbf{Q}_b$ and $\mathbf{Q}_w$ is the least, i.e. when the most important variable is omitted. Then we remove the selected feature $f \in \mathcal{F}_{k-1}$ permanently, we can restart the algorithm with the updated feature set $\mathcal{F}_k$. The algorithm ends when the cardinality of the feature set is 1.

The FRFCO algorithm

1. Let $\mathcal{F}_0 := \{f_1, \ldots, f_N\}$, $k = 1$.
2. For all $f_{\text{temp}} \in \mathcal{F}_{k-1}$
   (a) Let $\mathcal{F}_k := \mathcal{F}_{k-1} - \{f_{\text{temp}}\}$, and update matrix $\mathbf{X}$ by deleting temporarily its f_{temp}th row, and vectors $\underline{v}_i$ (see (11)) and $\underline{x}$ (see (7)) by deleting temporarily their f_{temp}th element.
   (b) Calculate matrices $\mathbf{Q}_b(\mathbf{X}_{\mathcal{F}_k})$, $\mathbf{Q}_w(\mathbf{X}_{\mathcal{F}_k})$ and determine $J(\mathbf{X}_{\mathcal{F}_k})$.
3. Let $f_{\text{perm}} = \operatorname{argmin}\limits_{f \in \mathcal{F}_{k-1}} J(\mathbf{X}_{\mathcal{F}_k})$, i.e. where J attains its minimal value. We obtain the final $\mathcal{F}_k$ by deleting permanently the variable f_{perm} from $\mathcal{F}_{k-1}$, and we update expressions $\mathbf{X}$, $\underline{v}_i$ and $\underline{x}$ appropriately.
4. If $k < N - 1$ then back to step 2, else stop.

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

Remark 1. Note that f_{perm} can contain more than one variable. In such a case we delete all of them at a time.

Remark 2. Primarily, we apply the SBS search method, but we also run experiments with SFS search method for comparison, see Section 5.

4.3 Selection of criterion function

In [10] it is shown that (1) can be transformed with algebraic manipulation to

$$J_1(\mathbf{X}_{\mathcal{F}}) = \mathrm{tr}(\mathbf{Q}_w) + \mathrm{tr}(\mathbf{Q}_b) \tag{12}$$

where "tr" denotes the trace of a matrix, the sum of the diagonal elements. Recall that we intend to maximize $\mathbf{Q}_b$ and at the same time minimize $\mathbf{Q}_w$. It can be obtained by maximizing (12), however, in this case the effect of intraclass distance of input vectors is unchecked. Therefore, the magnitude of the criterion function (12) is not a good indicator of class separability.

A more realistic criterion function to maximize is

$$J_2(\mathbf{X}_{\mathcal{F}}) = \frac{\mathrm{tr}(\mathbf{Q}_b)}{\mathrm{tr}(\mathbf{Q}_w)} \tag{13}$$

which reflects more to the described intuitive notion. For more details see [9,10]. However, (13) is only suitable if the domain of variables are homogeneous or at least comparable, because if the dimensionality are different, e.g. one variable is in the linear domain and another is in the logarithmic, then the comparison is meaningless. Another drawback of J_2 was pointed out in [9], that is 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.

Moreover, expression (13) ignores the effect on the separability of the correlated variables. This shortcoming can be overcome by preprocessing the scatter matrix $\mathbf{Q}_w$ by a suitable transformation $\mathbf{V}$ average covariance of the input vectors is the identity matrix: $\mathbf{V}^T\mathbf{Q}_w\mathbf{V} = \mathbf{I}$. It is easy to see that matrix $\mathbf{Q}_w^{1/2}$ is an appropriate choice for $\mathbf{V}$. In the transformed space criterion J_2 becomes

$$J_3(\mathbf{X}_{\mathcal{F}}) = \mathrm{tr}(\mathbf{Q}_w^{-1/2}\mathbf{Q}_b\mathbf{Q}_w^{1/2}) = \mathrm{tr}(\mathbf{Q}_w^{-1}\mathbf{Q}_b) = \sum_{k=1}^{N'} \lambda'_k \tag{14}$$

where λ'_k, $k = 1, \ldots, N'$ are the eigenvalues of matrix, $\mathbf{\Lambda}'$, i.e. the product matrix $\mathbf{Q}_w^{-1}\mathbf{Q}_b$, and N' is the size of the feature set $\mathcal{F}$. It can be shown [10], that J_3 corresponds to a separability measure which uses the quadratic metric with the average covariance of input vectors as scaling matrix. We can criticize J_3 from the point we mentioned for PCA, namely that it transforms the original features, hence the linguistic interpretability of the elements of a transformed feature set is questionable.

Another possible criterion function can be formulated by means of the determinants of scatter matrices. Determinant of scatter matrices has a geometrical interpretation [10], meaning the volume occupied by the elements

of clusters in the multidimensional space. Intuitively, the greater the ratio of the determinants of between- and within-class matrices, or more conveniently, of the determinants of the total- and within-class matrices, the greater the spatial separation of classes. Hence, the function

$$J_4(\mathbf{X}_{\mathcal{F}}) = \det(\mathbf{Q}_t)/\det(\mathbf{Q}_w) \tag{15}$$

is a suitable indicator of class separability.

Let us now investigate J_4. Since matrices $\mathbf{Q}_t$ and $\mathbf{Q}_w$ are symmetric, there exists a matrix $\mathbf{V}$ that diagonalizes both of them:

$$\begin{aligned} \mathbf{V}^T\mathbf{Q}_t\mathbf{V} &= \mathbf{\Lambda}, \\ \mathbf{V}^T\mathbf{Q}_w\mathbf{V} &= \mathbf{I}. \end{aligned} \tag{16}$$

Because of (16), J_4 can be expressed in terms of diagonal elements of matrix $\mathbf{\Lambda}$ as:

$$J_4(\mathbf{X}_{\mathcal{F}}) = \frac{\det(\mathbf{V}^T\mathbf{Q}_t\mathbf{V})}{\det(\mathbf{V}^T\mathbf{Q}_w\mathbf{V})} = \sum_{k=1}^{N'} \lambda_k$$

Furthermore, from (16) follows

$$\mathbf{Q}_w^{-1}\mathbf{Q}_t\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \tag{17}$$

so $\mathbf{\Lambda}$ is the matrix of eigenvalues of $\mathbf{Q}_w^{-1}\mathbf{Q}_t$. Due to (10), we have

$$\mathbf{Q}_w^{-1}\mathbf{Q}_t = \mathbf{I} + \mathbf{Q}_w^{-1}\mathbf{Q}_b$$

Now, from (14) and (17) we obtain

$$\mathbf{\Lambda}' = \mathbf{\Lambda} - \mathbf{I}$$

so $\lambda'_k = \lambda_k - 1$ and consequently J_4 can be expressed as

$$J_4(\mathbf{X}_{\mathcal{F}}) = \sum_{k=1}^{N'} (1 + \lambda'_k)$$

Comparing J_3 and J_4 we can state that the latter is likely to be more reliable in cases where the contributions of total interclass distance are distributed evenly over all the axes of the coordinate system. Thus, in contrast to J_3, it is unlikely, that J_4 would select a feature set which separates two classes well, but only at the expense of the ability of discriminate between all other classes. This situation can be characterized, in terms of eigenvalues λ_k, by the domination of a single eigenvalues in the criterion J_3.

We note that the value of J_4 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.

Another problem can arise if any of the matrices appearing in (15) are singular, and hence the determinant is negative. This problem can be solved by taking the absolute value of the determinant.

The use of these functions does not require any modifications of the FRFCO algorithm. We examine their effect on the ranking in Section 5.

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.

5 The comparative analysis of FRFCO

We applied the proposed method to several data sets. We compare our results to two other methods: the RC method [20] using a fixed setting and input contribution measure (ICM) analyzer (similar to sensibility analysis) [14]. 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 submodels are built for the data [29]. 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.

For FCM we fixed the value of fuzzifier to the default ($m = 2$), and we use SBS search method unless stated otherwise.

5.1 Simple synthetic data set

First, we checked whether FRFCO gives consistent result on the synthetic and real data sets given in [27]. The synthetic data set contained 50 samples with 4 variables. The first two variables were obtained from the function

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

and the last two were chosen randomly. The optimal number of clusters determined by (6) equals 6. In this case the determinants are all positive in each iteration. Table 1 shows the results obtained by FRFCO with various criterion functions with different search method and by other techiques.

On this synthetic sample our proposed method gives the correct ranking by using function J_3 or J_4 as the criterion function. Because all the determinants are positive we do not need to take their absolute value to solve singularity problem. The criterion function J_2 also finds the most important

Table 1. Ranking of the 4 input features of the synthetic data set from [27]

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

[a] The RC method automatically prunes the irrelevant variables.

variable, but fails to find the second one. When SFS search method is used the ranking is very similar, just the order of random variables is changed.

5.2 Real data set of a chemical plant

The second sample data set was the model of a chemical plant with 5 inputs [27]. 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 [27] 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 [29]. 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 2. 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 is correct, and absolute value function is need not to be applied to J_4. Criterion functions permute the order of the last three variables, but this is not very significant, because their contributions are similar and very low according to the ICM analyzer. Figure 1 shows the values of J_2 before the first deletion. Clearly, the difference between the first and the last two variables 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.

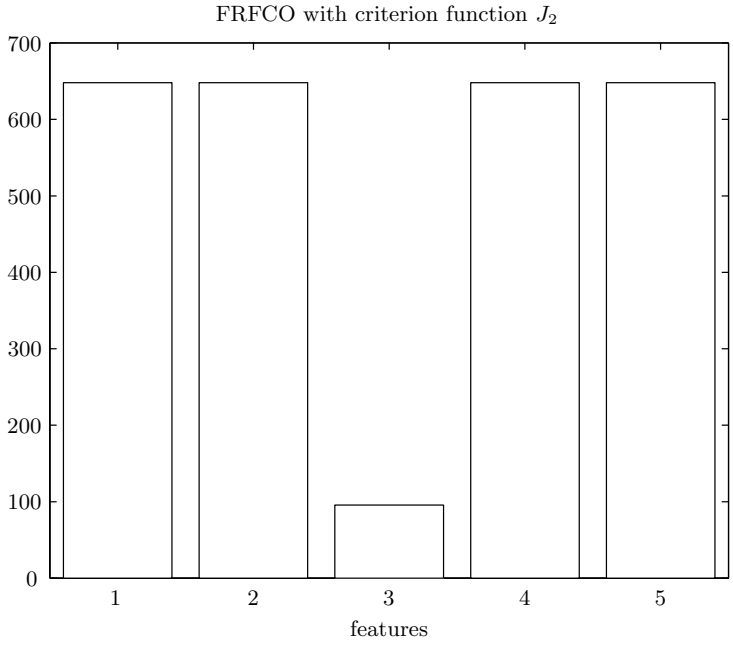

Fig. 1. The values of J_2 before the first elimination (on chemical plant data)

5.3 Eye-gaze data set

We also applied our technique to eye-gaze data set which was analyzed thoroughly by Gedeon in [15]. There, the author studied trained back propagation neural networks for ranking the 12 input features. The number of sample data

Table 2. Ranking of the 5 input features of the chemical plant data set from [27]

method	Ranking	Remarks
FRFCO with J_2	$\langle 3, 1, 5, 2, 4\rangle$	
FRFCO with J_3	$\langle 3, 1, 2, 5, 4\rangle$	
FRFCO with J_4	$\langle 3, 1, 4, 5, 2\rangle$	
ICM with 5–10–1	$\langle 3, 1, 2, 5, 4\rangle$	contribution of variables:
architecture		$\langle 75.94, 22.17, 1.10, 0.78, 0.01\rangle$
RC in [27]	$\langle 3, 2, 1\rangle$	Automatically pruned [a]
RC	$\langle 3\rangle$	Automatically pruned [a]

[a] The RC method automatically prunes the irrelevant variables.

was 909. The data set is described in details in [15]. We used the eye-gaze data set with its original output, but also with the network outputs of the best trained network. The optimal number of clusters was 2 in both cases. There are both negative and positive determinants during the iteration steps. The results (for both data sets) of the FRFCO with various criterion functions are compared with the result of the best trained network and the ICM analyzer and are depicted in Table 3. The first part of the table contains the results with the original data, while the second part with network output data.

Table 3. Ranking of the 12 input features of eye-gaze data set

method	Ranking	Remarks
best of [15]	$\langle 6,4,3,2,7,5,1,10,9,8,12,11\rangle$	
FRFCO with J_2	$\langle 7,1,8,12,3,2,11,10,4,5,9,6\rangle$	
FRFCO with J_4	$\langle 6,5,3,2,9,7,4,1,12,10,8,11\rangle$	
FRFCO with J_4'	$\langle 7,4,3,8,11,5,1,2,9,10,12,6\rangle$	
ICM with 12–24–1	$\langle 12,7,9,6,1,3,8,4,2,10,11,5\rangle$	contribution of variables:
architecture	$\langle 25.22, 19.66, 17.66, 9.57, 9.09, 5.34, 4.17, 3.29, 2.18, 1.83, 1.18, 0.81\rangle$	
FRFCO with J_2	$\langle 7,12,1,3,2,8,10,11,9,5,6,4\rangle$	
FRFCO with J_4	$\langle 5,4,6,9,8,1,11,2,3,10,12,7\rangle$	
FRFCO with J_4'	$\langle 7,1,3,10,12,4,5,6,9,11,2,8\rangle$	
ICM with 12–24–1	$\langle 7,11,9,10,12,6,8,3,1,2,5,4\rangle$	contribution of variables:
architecture	$\langle 28.1, 14.28, 10.31, 10.3, 9.82, 6.65, 5.95, 5.95, 3.26, 2.92, 1.71, 0.73\rangle$	

The results in Table 3 give many possibilities for comparison, from which we would like to emphasize the following. The results by J_2 are not in accordance with Gedeon's best considered one. The reason of this result partly is that the domains of variables vary. Due to the various signs of the determinants the use of the absolute value gives different results. Surprisingly, the result of J_4 without absolute value is more similar to the reference result of Gedeon than J_4 with absolute value.

The ICM analyzer also does not give correct results. It is in accordance with [15], where the similar sensitivity analysis was studied and it failed to find the correct or close-to-correct ranking. It may be argued that in this case we have only two clusters, and a small change in a variable can change the dominant cluster if the point is close to the boundary and the change applied ia towards the other cluster, but on the other hand, if a huge change applies

in the opposite direction the dominant cluster remains the same. Note, that in this case the membership in the clusters is not fuzzy, but crisp!

Naturally, the results with the first data set is closer to Gedeon's results, and the evaluation with the network output the ranking gives a different result. Nevertheless, the six most significant variables coincide with Gedeon's result in 3 places, and the three most significant variables in 2 places.

5.4 Summary and hints for use

The best results were obtained in all the three examples with the use of the criterion function J_4. If the determinants are of both signs, then the use of absolute value function can improve its performance. When the domain of the features is homogeneous or at least comparable 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 [27]) 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 [25,27]).

6 Conclusion

We proposed in this paper a feature ranking algorithm adapted to fuzzy modelling 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. Finally, some hints for the use of the algorithm were presented.

Acknowledgement

The authors would like to thank the anonymous referees for their very valuable and constructive comments and help.

References

1. S. Abe, R. Thawonmas, and Y. Kobayashi. Feature selection by analyzing class regions approximated by ellipsoids. *IEEE Trans. on SMC, Part C*, 28(2), 1998. http://www.info.kochi-tech.ac.jp/ruck/paper.html.

2. P. Baranyi and L. T. Kóczy. A general and specialized solid cutting method for fuzzy rule interpolation. *BUSEFAL*, 67:13–22, 1996.
3. 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)*, pages 2785–2790, San Diego, USA, 1998.
4. P. Baranyi and Y. Yam. Fuzzy rule base reduction. In D. Ruan and E. E. Kerre, editors, *Fuzzy IF-THEN Rules in Computational Intelligence: Theory and Applications*, pages 135–160. Kluwer, 2000.
5. P. Baranyi, Y. Yam, C. T. Yang, P. Várlaki, and P. Michelberger. Inference algorithm independent SVD fuzzy rule base complexity reduction. *International Journal of Advanced Computational Intelligence*, 5(1):22–30, 2001.
6. J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
7. 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.
8. P. J. Costa Branco, N. Lori, and J. A. Dente. New approaches on structure identification of fuzzy models: Case study in an electro-mechanical system. In T. Furuhashi and Y. Uchikawa, editors, *Fuzzy Logic, Neural Networks, and Evolutionary Computation*, pages 104–143. Springer-Verlag, Berlin, 1996.
9. R. K. De, N. R. Pal, and S. K. Pal. Feature analysis: Neural network and fuzzy set theoretic approaches. *Pattern Recognition*, 30(10):1579–1590, 1997.
10. P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, London, 1982.
11. J. C. Dunn. Well-separated cluster and optimal fuzzy partition. *J. Cybern.*, 4:95–104, 1974.
12. R. A. Fischer. The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7:179–188, 1936.
13. Y. Fukuyama and M. Sugeno. A new method of choosing the number of clusters for fuzzy c-means method. In *Proc. of the 5th Fuzzy System Symposium*, pages 247–250, 1989. (in Japanese).
14. 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)*, pages 35–39, Melbourne, 1997.
15. T. D. Gedeon. Data mining of inputs: Analysing magnitude and functional measures. *International Journal of Neural Systems*, 8(2):209–218, 1997.
16. T. D. Gedeon and L. T. Kóczy. Conservation of fuzziness in rule interpolation. In *Proc. of the Symp. on New Trends in Control of Large Scale Systems*, volume 1, pages 13–19, Herľany, Slovakia, 1996.
17. D. E. Gustafson and W. C. Kessel. Fuzzy clustering with a covariance matrix. In *Proc. of the IEEE International Conf. of Fuzzy Systems*, pages 761–766, San Diego, 1979.
18. T.-P. Hong and J.-B. Chen. Finding relevant attributes and membership functions. *Fuzzy Sets and Systems*, 103(3):389–404, 1999.
19. T.-P. Hong and C.-Y. Lee. Induction of fuzzy rules and membership functions from training examples. *Fuzzy Sets and Systems*, 84:33–47, 1996.
20. J. Ihara. Group method of data handling towards a modeling of complex system IV. *Systems and Control*, 24:158–168, 1980. (In Japanese).

21. 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.
22. D. Kleinbaum, L. L. Kupper, and K. E. Muller. *Applied Regression Analysis and Other Multivariable Methods*. PWS-Kent, Boston, Mass., 2nd edition, 1988.
23. L. T. Kóczy and K. Hirota. Approximate inference in hierarchical structured rule bases. In *Proc. of 5th IFSA World Congress (IFSA'93)*, pages 1262–1265, Seoul, 1993.
24. L. T. Kóczy and K. Hirota. Size reduction by interpolation in fuzzy rule bases. *IEEE Trans. on SMC*, 27:14–25, 1997.
25. 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.
26. 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)*, pages 1262–1265, Seoul, 1993.
27. M. Sugeno and T. Yasukawa. A fuzzy logic based approach to qualitative modelling. *IEEE Trans. on Fuzzy Systems*, 1(1):7–31, 1993.
28. D. Tikk and P. Baranyi. Comprehensive analysis of a new fuzzy rule interpolation method. *IEEE Trans. on Fuzzy Systems*, 8(3):281–296, 2000.
29. 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-02-2001, School of Information Technology, Murdoch University, Perth, W.A., 2001. p. 17.