Feature Selection and Clustering Based Fuzzy Modeling

CHONG, A.[1], GEDEON, T.D.[1], KOCZY, L.T.[1, 2]

[1]School of Information Technology
Murdoch University
South Street, Murdoch, WA, 6150
AUSTRALIA

[2]Department of Telecommunication & Telematics
Budapest University of Technology and Economics
and
Institute of Information Technology and Electrical Engineering, Széchenyi István University, Győr
HUNGARY

cchong@murdoch.edu.au

Abstract: In this paper, we propose a fast feature selection technique for clustering-based fuzzy modeling. The technique involves the creation of 'rough' fuzzy systems quickly from a set of data and chooses the one with the lowest error. The set of features used by the chosen fuzzy system is accepted as the optimal set of features. The effectiveness and efficiency of the proposed technique is validated using artificially generated data.

1 Introduction

Feature analysis plays an important role in fuzzy modeling. By fuzzy modeling, we refer to the process of automatic fuzzy rules extraction from a set of training data. It is well known that fuzzy rulebases suffer from rules explosion. That is, the number of fuzzy rules needed to cover a problem domain densely grows exponentially with the increase of system inputs (i.e. features). It is therefore essential to rely on feature analysis techniques to eliminate less important features prior to the modeling process. Apart from this, unimportant features also have negative impact on the reliability of clustering algorithms. Thus, fuzzy modeling techniques that make use of fuzzy clustering might suffer from low accuracy when unimportant features exist in the set of training data.

Feature analysis techniques can be broadly categorized into three types: feature ranking, feature selection and feature extraction. Given a set of features, $F = \{x_i \mid i = 1\ldots n\}$, feature ranking results in the rank of importance for each feature x_i. Feature selection on the other hand, aims at selecting $n_0 < n$ features to produce the set of important features $F_0 \subset F$. Feature extraction concerns with finding a set of transformed features F' where $|F'| < n$. It has been suggested in [1, 2] that given the rank of importance produced in feature ranking, feature selection can be performed via a trial and error approach. The idea is as follows. Initially, a fuzzy rule base is constructed using any rules extraction technique with a small number of top ranked features (e.g. one or two). The completed fuzzy rulebase is evaluated by a performance index. The process is repeated with an increased number of features. If a local optimum is reached according to the performance index, the set of features is accepted as the optimal set of features. That is, the entire process is repeated until the fuzzy rulebase using n features has a lower performance than the previous fuzzy rulebase that uses n-1 features.

The straightforward feature selection technique described above effectively bridges the gap between feature ranking and feature selection in the context of fuzzy modeling. One significant drawback of the method is the high demand in time and computing power. The main purpose of this paper is to propose a fast feature selection technique that is more computationally efficient.

This paper is organized as follows. Section 2 presents and overview to fuzzy clustering and feature ranking. Section 3 investigates the use of clustering in fuzzy modeling. The proposed feature selection technique is explained in section 4. The experiments are reported in section 5. This is followed by the conclusion in section 6.

2 Fuzzy Clustering and Feature Ranking

Given a set of data, Fuzzy c-Means clustering (FCMC) [3] performs clustering by iteratively searching for a set of fuzzy partitions and the associated cluster centers that represent the structure of the data as best as possible. Given the number of clusters c, FCMC partitions the data $X = \{x_1, x_2, \ldots, x_n\}$ into c fuzzy partitions by minimizing the

Δ Research supported by the the Australian Research Council, National Scientific Research Fund OTKA T034233 and T034212, a Main Research Direction Grant 2002 by Széchenyi István University, and the National Research and Development Project Grant NKFP-2/0015/2002.

within group sum of squared error objective function as follows (Eqn 1).

$$J_m(U,V)=\sum_{k=1}^{n}\sum_{i=1}^{c}(U_{ik})^m \| x_k - v_i \|^2,\ 1\le m\le\infty$$

Eqn 1

where J_m(U,V) is the sum of squared error for the set of fuzzy clusters represented by the membership matrix U, and the associated set of cluster centers V. Here, $\|x_k - v_i\|^2$ represents the distance between the data x_k and the cluster center v_i. At each iteration, the cluster centers are calculated using (Eqn 2) and (Eqn 3). The optimal number of clusters in the data is determined by means of the FS index [4]. The number of clusters, c, is determined so that $S(c)$ in (Eqn 4) reaches a local minimum as c increases.

$$U_{ik}=\left(\sum_{j=1}^{c}\left(\frac{\| x_k - v_i \|}{\| x_k - v_j \|}\right)^{2/(m-1)}\right)^{-1}\ \forall i, \forall k\ ,\qquad \text{Eqn 2}$$

$$v_i=\frac{\sum_{k=1}^{n}(U_{ik})^m x_k}{\sum_{k=1}^{n}(U_{ik})^m},\qquad \text{Eqn 3}$$

$$S(c)=\sum_{k=1}^{n}\sum_{i=1}^{c}(U_{ik})^m(\|x_k-v_i\|^2-\|v_i-\bar{x}\|^2)\ \ 2<c<n,$$

Eqn 4

Fuzzy clustering plays an important role in feature ranking. The idea is that given a set of clusters, the importance of each feature can be determined by considering its capability to separate the clusters [2]. One of the well-known methods used for this purpose is by using the interclass separability criterion. Consider a set of N input-output pairs $F = \{X; y\}$, $X = \{x_i \mid i \in I\}$ where I is the index set, x_i and y are column vectors. By deleting some features (input variables), we obtain a subspace $X' = \{x_i \mid i \subset I\}$. Suppose that the input X is clustered into clusters C_i (i = 1, …, N_c) then the criterion function for feature ranking based on the interclass separability is formulated by means of the following fuzzy between-class (Eqn 5) and within-class (Eqn 7) scatter (covariance) matrices.

$$Q_b=\sum_{i=1}^{N_c}\sum_{j=1}^{N}\mu_{ij}^m\left(v_i-\bar{v}\right)\left(v_i-\bar{v}\right)^T\qquad \textbf{Eqn 5}$$

$$Q_i=\frac{1}{\sum_{j=1}^{N}\mu_{ij}^m}\sum_{j=1}^{N}\mu_{ij}^m\left(x_j-v_i\right)\left(x_j-v_i\right)^T\qquad \textbf{Eqn 6}$$

$$Q_w=\sum_{i=1}^{N_c}Q_i\qquad \textbf{Eqn 7}$$

where

$$\bar{v}=\frac{1}{N_c}\sum_{i=1}^{N_c}v_i\qquad \textbf{Eqn 8}$$

Here, v_i is given by (Eqn 3). The criterion is a trade off between Q_b (Eqn 5) and Q_w (Eqn 7), often expressed as:

$$J(X')=\frac{tr(Q_b)}{tr(Q_w)}\qquad \textbf{Eqn 9}$$

where '*tr*' denotes the trace of a matrix. In [2], the set of classes C is determined by clustering the output space using fuzzy clustering algorithms such as fuzzy c-means [3]. The resulting partition matrix $U = \{\mu_{ik} \mid i = 1\ldots N_c, k = 1\ldots N\}$ are then used as weights (Eqn 5 - Eqn 8). Each feature can be ranked using the sequential backwards algorithm. Firstly, different subsets of data are obtained by temporary deleting each feature. This is followed by deleting permanently the feature whose removal resulted in the largest value. This process is repeated until all features are deleted and the order of the deleted variables gives their rank of importance.

3 Clustering Based Fuzzy Modeling

The main aim of this research is to design a feature selection technique for clustering based fuzzy modeling. To do so, it is essential to understand the use of clustering in rules extraction. A significant amount of work has been carried out in this area [5]. The general idea is to perform clustering on the training data and convert the fuzzy clusters obtained into fuzzy rules. Figure 1a shows a simplified example where 5 fuzzy clusters are converted into 5 fuzzy rules. In rules extraction, determining the 'right' number of clusters is not as crucial as it often is in the pattern recognition point of view. Since more clusters lead to more fuzzy rules, the increase in the number of clusters almost always improves the accuracy of the fuzzy systems. Figure 1b shows the situation where 8 instead of 5 fuzzy clusters are obtained. The finer patches (clusters) in the figure leads to more exact rules. In this case, the 'right' number of clusters can be determined by considering the trade-off between system accuracy and complexity.

Fuzzy clustering can be performed on the different domains, namely X, XY, and Y for rules extraction. The proper choice of the clustering domain often depends on the data at hand. In this paper, the issue of clustering domain selection is investigated further. Let us consider the approximation of a nonlinear function as illustrated in Figure 2. It can be observed from the figure that when the data points are sampled differently, different clustering domains are needed for effective rules extraction.

In Figure 2a, the data points are sampled at a fixed interval along X. This results in data points whose projections to X and Y are evenly distributed along the domains. Consequently, clustering on X and Y independently is likely to fail since no cluster structure exists in along X and Y independently. By using a fuzzy clustering method

(e.g. the GK clustering [6]) along *XY* on the other hand, one might obtain a set of clusters shown as the shaded area in Figure 2a. The resulting clusters in this case are intuitively suitable for rules extraction. Figure 2b shows a different sampling of data. In this example, clustering in *XY* or *X* might lead to clusters that span across the entire *Y*. The only intuitively effective domain to use, in this case, is *Y*. In the last example Figure 2c, the data is sampled in a way such that their projections in *Y* are evenly distributed along the entire domain. In this case, clustering on both *X* and *XY* yields reasonable results.

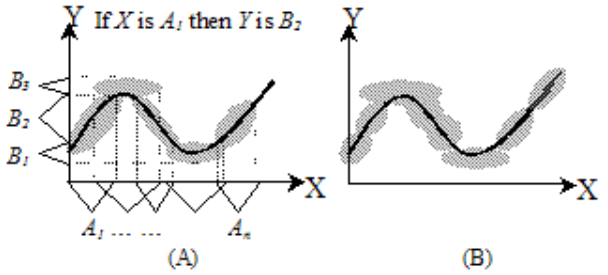

Figure 1: Fuzzy clusters to be converted into fuzzy rules that patch the function being approximated.

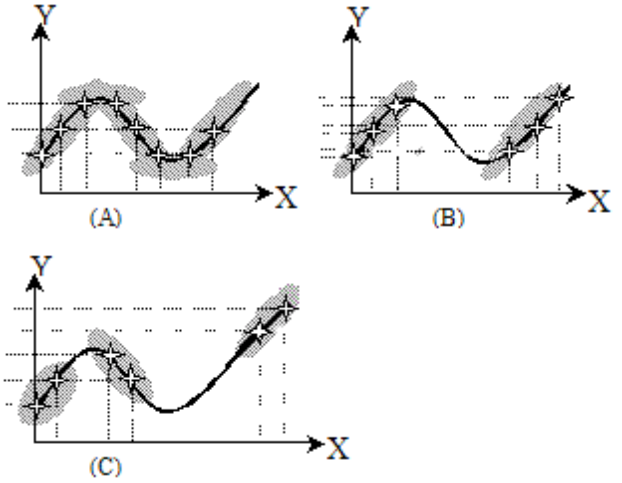

Figure 2: A nonlinear function sampled differently creating situations that requires different clustering domain for fuzzy rules extraction.

From the examples, it can be concluded that it is unlikely to find a single 'best' choice of clustering domain for rules extraction. It all depends on the data at hand. In [7], we proposed a projection-based approach for rules extraction. The idea is that clustering is first performed on the output space. The data points from each output cluster are then projected back to each input dimension. Clustering is performed on the data projections to produce 1D clusters at each input dimension. The 1D clusters are then combined to form the fuzzy rules using a computational effective algorithm. The rulebase is then optimized by merging pairs of fuzzy rules. Using this approach, we first obtain a large number of clusters that is likely to cover sufficiently the problem domain. The merging process is then carried out to reduce the number of fuzzy rules whilst maintaining the accuracy of the system.

Fuzzy rules formation from clusters by projecting the cluster to individual dimension (Figure 1) results in the loss of information. Therefore, the fuzzy modeling process is often followed by a parameter tuning process [1]. The parameters of each fuzzy set in the rules are adjusted to improve the system performance.

4 Fast Feature Selection

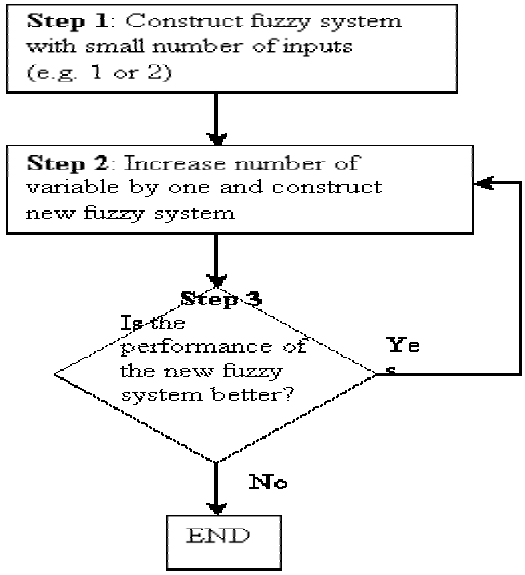

Figure 3: Flow chart for the feature selection process used in [1].

The main idea of feature selection in [1] is summarized in Figure 3. The feature selection process involves the construction of several fuzzy systems. This can be very time and resource consuming. Moreover, when a clustering-based rule extraction technique is used, the process can lead to comparing performance of fuzzy systems with different number of rules. Theoretically, such comparison can produce misleading results since fuzzy systems with more rules are likely to have better accuracy (see section 3).

In this paper, we propose a more computationally efficient algorithm for feature selection by simplifying step 1 and 2 in Figure 3. Instead of performing fuzzy rules extraction to generate a complete fuzzy system, we aim at generating 'rough' fuzzy systems quickly from the training data. The idea is that the actual performance of each fuzzy system is not important since we are only concerned with the performance comparison.

The proposed algorithm is explained as follows. Given a set of N input-output pairs $F = \{X; y\}$, $X = \{x_i \mid i \in I\}$ where I is the index set, x_i and y are column vectors, the first step is to perform clustering on the output space y into C clusters. With the partition matrix $U = \{\mu_{ik} \mid i = 1\ldots C, 1\ldots N\}$ obtained, the n most important features is selected such that (Eqn 10) reaches a local optimum.

$$\sum_i \left\{ y_i - defuzz \left(\frac{\sum_c \mu_{ci}}{\left(\sum_{j=1}^{c} \left(\frac{\| x_k - v_i \|}{\| x_k - v_j \|} \right)^{2/(m-1)} \right)} \right) \right\}^2 \quad \textbf{Eqn 10}$$

where

$$v_i = \frac{1}{\sum_k \mu_{ik}^m} \sum_k \mu_{ik}^m x_k \quad \textbf{Eqn 11}$$

Here, *defuzz*(.) denotes a defuzzification method, such as center of area (COA), that is often used in fuzzy rulebases.

It can be observed that the right most term in (Eqn 10) resembles a simplified fuzzy system that uses the clusters as multi-dimensional fuzzy rules. In this case, the number of rules used is fixed as the number of clusters obtained from output clustering. This allows us to compare the performance of fuzzy systems with the same amount of fuzzy rules generated using different sets of features.

The use of multi-dimensional fuzzy sets as the rule antecedent relieves the algorithm from the computationally complex task of parameter tuning (see section 3). The algorithm scales linearly with the increase of input features due to the use of Euclidean distances in the calculation.

The selection of Y as the clustering domain is motivated by the fact that fuzzy systems output is often one-dimensional. In general, multi-dimensional output systems can be split into multiple single output systems. Performing clustering on one-dimensional data is relatively more computationally efficient.

5 Experiments

Experiments have been carried out to validate the effectiveness and efficiency of the proposed fast feature selection technique. In the first experiment, three sets of 4-dimensional data, each consisting of 200 rows, are generated using the three functions (Eqn 12 - Eqn 14). It can be observed that each function only uses a subset of input features (e.g. Eqn 13 uses only X_1 and X_2). Both the proposed fast feature selection and the original feature selection technique in [1] are applied to datasets. With the original technique, the fuzzy modeling process is performed using [7]. The completed fuzzy system then goes through a parameter tuning process. The tuning is performed until either the performance improvement is lower than a threshold or a maximum number of iteration is reached. We remark that the parameter tuning process is necessary for the original feature selection technique to produce accurate results. Our experiments show that without tuning, the feature selection algorithm produces counter intuitive results. Feature ranking was performed using (Eqn 9).

Both algorithms were able to correctly identify the true inputs. Table 1 shows the amount of time taken for the algorithms to process each set of data on a Pentium 4 processor with 256 MB RAM. It can be seen from Table 1 that the proposed technique takes much less time to complete the task. The time taken to process each of the 3 datasets is about the same. With the original approach, the time taken by the algorithm increases significantly with the increase of input features. This is mainly due to the parameter tuning process that has to tune the increased number of membership functions in the system.

$$y = 2x_1, \qquad 1 \le x_1 \le 5 \quad \textbf{Eqn 12}$$

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \le x_1, x_2 \le 5 \quad \textbf{Eqn 13}$$

$$y = (1 + x_1^{-2} + x_2^{-1.5} + x_4^{-3})^2, \quad 1 \le x_1, x_2, x_3 \le 5 \quad \textbf{Eqn 14}$$

Dataset	Feature Select	Proposed
1 (Eqn 12)	5.4770	1.9630
2 (Eqn 13)	28.5510	2.0430
3 (Eqn 14)	82.8890	1.9320

Table 1: Time (in milliseconds) taken for the feature selection algorithms to process data with different number of true inputs.

Data	Feature Select	Proposed
1 (100 row)	17.8860	0.6910
2 (200 row)	28.4610	1.9630
3 (300 row)	282.7760	4.5160

Table 2: Time (in milliseconds) taken for the feature selection algorithms to process data with different number of rows.

In the next experiment, datasets with different number of rows (100,200, and 300) were generated using (Eqn 13). Table 2 shows the results in terms of time taken for each algorithm to process the datasets. The time taken for both algorithms increases as the number of data increases because of the need to perform clustering on the larger dataset. The original feature selection technique took significantly less time in processing the first 2 datasets (compared to the 3rd) because the parameter tuning

process stopped early due to the low accuracy improvement in the new iteration. Overall, both algorithms were able to correctly identify the true inputs but the proposed approach is significantly faster.

6 Conclusion

A fast feature selection technique has been proposed. Given the feature ranking in a set of data, the optimal number of top-ranked features can be selected by creating several 'rough' fuzzy systems and selecting the one with the best performance. The set of features used by the chosen fuzzy system is then accepted as the optimal set of features. The proposed technique can be used by most of the clustering-based fuzzy rules extraction techniques to eliminate less important variables in the training data. The effectiveness and efficiency of the feature selection algorithm has been validated using artificially generated data. In our subsequent work, real world data will be used to validate the algorithm.

Reference:

[1] Sugeno, M. and T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1993. **1**(1): p. 7-31.

[2] Tikk, D. and T.D. Gedeon. Feature ranking based on interclass separability for fuzzy control aplication. in *Proceedings of the International Conference on Artificial Intelligence in Science and Technology (AISAT'2000)*. 2000. Horbat: p. 29-32.

[3] Bezdek, J.C., *Pattern Reconition with Fuzzy Objective Function Algorithms*. 1981, New York: Plenum Press.

[4] Fukuyama, Y. and M. Sugeno. A new method of choosing the number of clusters for fuzzy c-means method. in *Proceedings of the 5th Fuzzy Systems Symposium*. 1989: p. 247-250.

[5] Bezdek, J.C., J. Keller, R. Krishnapuram, and N.R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. 1 ed. 1999, US: Kluwer.

[6] Gustafson, E.E. and W. Kessel. Fuzzy clustering with a fuzzy covariance matrix. in *Proceedings of the IEEE conference on Decision and Control*. 1979. San Diego: p. 761-766.

[7] Chong, A., T.D. Gedeon, and L.T. Koczy. Projection Based Method for Sparse Fuzzy System Generation. in *2nd WSEAS Int. Conf. on Scientific Computation and Soft Computing*. 2002. Crete: p. 321-325.