Generating Fuzzy Rules for classifier fusion

Ming Liu[1,2] Baozong Yuan[1] Xiaofang Tang[1] Ming Li[1]

[1]Institute of Information Science, Beijing Jiaotong Univ., Beijing 100044, China

[2] College of Electronic and Information Engineering, Hebei Univ., Baoding, Hebei 071002, China

Abstract

This paper addresses the issues in developing a fuzzy system for multiple classifier fusion. The proposed new approach has advantage in automatically generating fuzzy rules and determining the structure of the fuzzy system. Partitioning the attribute space into small regions can increase the description accuracy of the fuzzy system, but the number of training samples fall in each region will be decreased, and the reliability of the derived rules will be decreased also. To resolve this problem, we use membership functions with broad support set in the learning step, and propose an evaluation method to determine the structure of the fuzzy system. Experimental results are compared with other classifier fusion methods such as multi-response linear regression and neural network.

Key words: Classification, multiple classifier fusion, fuzzy system.

1. Introduction

This paper focuses on integrating multiple classifiers for the case of each classifier outputs a probability mass function. The system integrates all these functions and outputs a new probability mass function which is used to classify incoming samples. Function approximation methods, such as Multi-response linear regression (MLR) is recommended for this task [1,2]. However, MLR integrates classifiers in a linear manner which makes it ineffective for non-linear applications.

Recently, there has been a great deal of interest in applying fuzzy techniques for nonlinear function approximation [3]. Fuzzy techniques have many attractive characters: (1) Comprehensible; (2) Model-free; (3) Providing a linguistic framework for the fusion of expert or background knowledge and knowledge derived from data [4].

Fuzzy system has been used for data fusion by many users. See reference [5,6] for example. One problem has not been resoled in this field is how to determine the structure of the fuzzy system automatically. In reference [5], the number of membership functions used for fuzzification is determined in advance. In reference [6], a quality evaluation measure is proposed to help the user to adjust the parameters of the fuzzy system. But the proposed measure is not a complete result evaluation measure, and the parameters can not be determined automatically.

Based on Label semantics [7], N.J.Randon proposed a Fuzzy Semi-Naïve-Bayes classifier [8]. It is reported that the best result of the Semi-Naïve-Bayes classifier outperform a set of well known benchmark classification algorithms. However, the method to determine the number of membership functions used for each attribute is not resolved. I.Rojas proposed a self-organized fuzzy rule generating procedure for function approximation in [3]. In his method a large number of membership functions are assigned in the regions where the fuzzy system's error value is high. For classifier fusion problem, such regions usually contain small number of samples, further partition them will result in over learning.

In this paper we discusses the issues in developing a fuzzy system to integrate multiple classifiers, Partition the attribute space into small regions can increase the description accuracy of the fuzzy system, but the number of training samples fall in each region will be decreased, and the reliability of derived rules will be decrease also. To resolve this problem, we use membership functions with broad support set in the learning step, and propose an evaluation method to determine the appropriate structure of the fuzzy system.

2. Classifier Fusion by Fuzzy System

The proposed approach for classifier fusion has four steps. Firstly, the output values of the classifiers are fuzzified initially. Secondly, a learning procedure is used to generate the fuzzy rules. Thirdly, a classifier is derived based on the fuzzy rules. Finally, the parameters of the fuzzy system are determined using an evaluation method.

Let $\{C_1, \cdots C_j, \cdots C_N\}$ be a set of N classifiers generated on a single training data set $\boldsymbol{S}$, and $\boldsymbol{\Omega} = \{\omega_1, \omega_2 \cdots \omega_K\}$ be the set of possible class values. For each sample *x*, the output of a classifier C_j is a probability distribution vector:

$$\boldsymbol{P}_j(x) = \{P_j(\omega_1 \mid x), \cdots, P_j(\omega_K \mid x)\}$$

where $P_j(\omega_i \mid x) \in [0,1]$. For simplicity, denote $P_j(\omega_i \mid x)$ by attribute v_j^i. Based on a cross validation procedure each training sample x_i can be transformed into a meta-level training sample $\{\vec{v}_1, \vec{v}_2 \cdots \vec{v}_N\}$, with $\vec{v}_j = \{v_j^1, v_j^2 \cdots v_j^K\}$, and the training set $\boldsymbol{S}$ can be transformed into a meta-level training set [2]：

$$\boldsymbol{DB} = \{(y_n, \vec{v}_1, \vec{v}_2 \cdots \vec{v}_N), n = 1, \cdots N\}$$

where y_n is the class label.

2.1. Fuzzification

For an attribute v_C^i, we first partition the range of it into crisp intervals and then specify membership functions for every interval. Represent each interval by a linguistic label, Then the attribute can be described by a set of linguist labels: $LA^i = \{T_1^i, \cdots, T_M^i\}$.

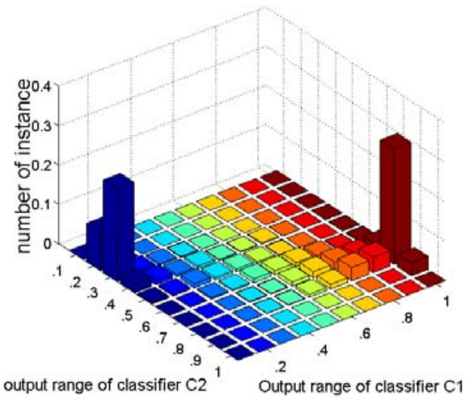

Fig.1 Distribution of the classifiers output

If the classification accuracy of the classifiers is high, then the outputs of the classifiers will concentrate on some regions in the output space, and there will be many regions that contain rather few samples, as can be seen in fig.1.

After fuzzification, the output space is partitioned to many overlapped fuzzy cells. Each fuzzy cell corresponds to a set of linguistic labels. If the fuzzy cells are small, then there will be many fuzzy cells that contain rather few even none of samples. Thus in the learning step, the knowledge acquired from training samples in these fuzzy cells will be very little, and the generated fuzzy rules will be not reliable.

The size of the fuzzy cells is determined by the support set of the membership functions. Small support set membership functions will lead to small fuzzy cells.

Many authors use triangular membership functions or trapezoidal membership functions when designing fuzzy system [3,4,5,6,7,8]. The support set of these membership functions is rather small, so they can not be used for the case of classifier fusion. In our work, we use membership functions in the form:

$$\mu_{x_v^j} = \begin{cases} 1 & if\ x = c_v^j \\ 0 & if\ x = c_v^i \ \&\ i \neq j \\ \dfrac{1}{|x - c_v^j|^r \left(\sum_j \dfrac{1}{|x - c_v^j|^r}\right)} & therwise \end{cases} \tag{1}$$

where, c_v^j represents the center of the jth membership function of attribute v, $r \in \{1,2\}$. This kind of membership function has been used in fuzzy c-mean clustering successfully [9]. Its support set contain almost the whole range of the attribute.

In the classification step, in order to get proper rules to calculate the output value, we use triangular membership functions which have small support set.

Having decided the form of the membership functions, a question remains of how to partition the attribute range. In [8] two partition method are recommended: Uniform Partitioning and Non-Uniform Partitioning. The former evenly partitions the range into a pre-specified number of regions, whereas the later uses additional information extracted from database by sorting the values of attributes into ascending order and then defines a crisp partition for each linguistic value so that each partition contains equal number of values.

In order to generate fuzzy rules for classifier fusion, we use Uniform Partitioning in our system. This is based on two reasons. First, the distributions of the output probabilities for different class are different. If we use Non-Uniform Partitioning method, then the partitions for different class may be different, and the output probabilities for different class will be not compatible. Second, the samples that result in classifier conflict usually fall in the regions which contain few samples. Properly classify these samples is rather important in classifier fusion system. So by Uniform Partitioning we treat all the regions evenly.

2.2. Generating fuzzy rules

For a class ω_q, let the outputs of the classifiers be attributes: $v_1^q, v_2^q, \cdots v_N^q$. Further suppose that we select a set of labels LA_j for each attribute v_j^q. From the meta-level training set $\boldsymbol{DB}$ we select the samples that belong to class ω_q as a subset $\boldsymbol{DB}_q$.Then a joint probability distribution over the linguistic labels conditional on class ω_q can be learned from $\boldsymbol{DB}_q$ [8]:

$$p(T^1,\cdots,T^N \mid \omega_q) = \frac{1}{\left|\boldsymbol{DB}_q\right|}\sum_{x_i\in DB_q}\prod_{j=1}^{N}\mu_{T^j}(v_j^q(x_i)) \quad (2)$$

where $T^j \in LA_j$ and $\left|\boldsymbol{DB}_q\right|$ is the number of the samples in subset $\boldsymbol{DB}_q$, $v_j^q(x_i)$ is the probability of $x_i \in \omega_q$, output by classifier C_j.

Based on Bayes' theorem, we can estimate the conditional probability of class ω_q given a set of linguistic labels $T^1, T^2 \cdots T^N$:

$$p(\omega_q \mid T^1,\cdots,T^N) = \frac{p(\omega_q)p(T^1,\cdots,T^N \mid \omega_q)}{\sum_{j=1}^{K} p(\omega_j)p(T^1,\cdots,T^N \mid \omega_j)} \quad (3)$$

Suppose the prior probabilities for each class are the same, we have:

$$p(\omega_q \mid T^1 \cdots T^N) = \frac{p(T^1,\cdots,T^N \mid \omega_q)}{\sum_{j=1}^{K} p(T^1,\cdots,T^N \mid \omega_j)} \quad (4)$$

Thus for each set of linguistic labels $T^1,\cdots,T^N$ we get a rule:

IF classifier C_1's output IS T^1,

$\cdots$,

AND classifier C_N's output IS T^N

THEN the probability that the sample belongs to class ω_q IS $p(\omega_q \mid T^1,\cdots,T^N)$.

2.3. Classification

To classify an incoming sample x, we transform it into a meta-level sample and estimate the post probability of class ω_q as:

$$p(x\in\omega_q) = \sum_{T^1 LA_1,\cdots,T^N\in LA_N} p(\omega_q \mid T^1,\cdots,T^N)\prod_{j=1}^{N}\mu_{T_j}(v_j^q(x)) \cdot \quad (5)$$

Then we derive a Bayes classifier which classifies an incoming sample into the class which has the maximum post probability.

2.4. Determining the system structure

Because we use Uniform Partitioning method to partition the attribute range, only the number of membership functions used for each attribute needs to be determined

The proposed fuzzy system is designed for classification problem, so the classification error rate on the training data set is a good measure to evaluate the system performance. Cross-validation can be used to do such work, but it is time-consuming.

In the cross-validation procedure, each turn to classify one or more samples, the system needs to be redesigned. To overcome this problem, we develop a compensation method to reduce the influence of the testing sample on the fuzzy system.

Having learned a joint probability distribution over the linguistic labels according to formula (3), we now modify it to reduce the influence of a meta-level sample $v_1^q(x_k), v_2^q(x_k), \cdots v_N^q(x_k)$:

$$p_n(T^1,\cdots,T^N \mid \omega_q)$$
$$= \frac{1}{\left|\boldsymbol{DB}_q\right|-1}\Big(\sum_{x_i\in DB_q}\prod_{j=1}^{N}\mu_{T_j}(v_j^q(x_i)) - \prod_{j=1}^{N}\mu_{T_j}(v_j^q(x_k))\Big)$$

$$= \frac{1}{\left|\boldsymbol{DB}_q\right|-1}\Big(\left\|\boldsymbol{DB}_q\right\|\big(p(T^1,\cdots,T^N \mid \omega_q) - \prod_{j=1}^{N}\mu_{T_j}(v_j^q(x_k))\big)\Big) \quad (6)$$

After that, we modify the conditional probability of class ω_q given a set of linguistic labels $T^1,\cdots,T^N$ and classify the training sample. The classification error rate based on this procedure is compatible with that based on cross-validation. So it can be used to evaluate the performance of the fuzzy system.

In experiments, we assume that the number of membership functions used for each attribute can be limited in a reasonable range, and use the proposed evaluation method to determine the proper number of labels for each attribute.

3. Experiment design

To evaluate the performance of the new approach, we select four dataset from the ELENA project as well as the UCI data set repository: Clouds, Phoneme, Satimage and waveform. These data sets are carefully selected so that they have distinct characteristics and are deemed useful by previous researchers.

Two classifiers are used in the experiments: support vector machine and k-nearest neighbor classifier.

For the k-nearest neighbor classifier, the output class probability is determined by the ratio of the number of the samples that belong to each class in the *k* nearest neighbors of the incoming sample to *k*.

As for the support vector machine, there are two cases: two-class problem and multiple-class problem. In two-class problem, the value of the decision function is used to determine the output probability mass function.

The multiple-class problem is transformed into multiple two-class problems. Suppose j classifiers from the all the k classifiers assign a sample x to class ω_*, then the probability that x belongs to this class can be estimated by j/k .

In experiment, the four data set used are partitioned into training set and testing set with equal size. Then the initial data sets are transformed into meta-level data sets based on the methods above. After that we use ten run cross-validation to obtain the classification error rates compares our fusion method with other fusion methods.

Table Ⅰ

Classification error rates of the individual classifiers and different classifier fusion method on testing sets of four data sets.

	Clouds (%)	Phoneme (%)	Satimage (%)	Waveform (%)
SVM	11.32 ± 0.38	16.52 ± 0.86	11.95 ± 0.42	13.84 ± 0.46
k-NN	11.49 ± 0.41	14.85 ± 0.43	11.06 ± 0.56	16.33 ± 0.72
MLR+	11.17 ± 0.33	14.89 ± 0.49	10.97 ± 0.50	16.00 ± 1.08
MLR	11.17 ± 0.24	15.34 ± 0.48	10.91 ± 0.56	14.99 ± 0.57
MLP	11.04 ± 0.44	15.61 ± 0.48	11.71 ± 0.51	13.78 ± 0.40
FR	**10.92 ± 0.38**	**14.80 ± 0.68**	**10.49 ± 054**	**13.70 ± 0.38**

4. Experiment results

Our method (FR) was compared with multi-response linear regression and neural network. Two kinds of multi-response linear regression method are used: multi-response linear regression with restriction that regression coefficient to be non-negative (MLR+) and multi-response linear regression without restriction that regression coefficient to be non-negative (MLR) [1]. Multilayer perceptron (MLP) is used for a representative of neural network. Based on ten-run cross-validation we obtain the classification error rates on the four data set used. Table Ⅰ presents the results.

Table Ⅰ shows that the classification error rate of our method is lower than that of the other three fusion methods. Moreover, our method has advantage in comprehensibility and the capability to integrate background knowledge and data.

5. Conclusions

Methods for developing fuzzy system for multiple classifier fusion were proposed. Two special problems have been solved: learning in the regions which contain few samples and the problem to determine the structure of the fuzzy system.

Experiments results show that the new approach has increased the classification accuracy. Further more, fuzzy system has many attractive characters, so the new approach is expected for many real applications.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 60441002) and the University Key Research Project (No. 2003SZ002).

7. References

[1] K. M. Ting, I. H. Witten, “Issues in stacked generalization”. *Journal of Artificial Intelligence Research*, Vol.10, pp. 271–289,1999.

[2] S.Dzeroski, B.Zenko, “Is combining classifiers better than selecting the best one?”, *Machine Learning*, Vol. 54, No.3,PP. 195–209.2004.

[3] I.Rojas, H.Pomares, J.Ortega, A.Prieto, “Self-organized fuzzy system generation from training examples”, *IEEE Trans. on Fuzzy Systems*, vol. 8, No.1, pp. 23-33,2000.

[4] J.Lawry, J. W. Hall and R. Bovey, “Fusion of expert and learnt knowledge in a framework of fuzzy labels”, *International Journal of Approximate Reasoning*, Vol. 36, No.2, pp .151-198, 2004.

[5] P.Wide, D.Driankov, “A fuzzy approach to multi-sensor data fusion for quality profile classification”, *MFI’96*, Washington, USA, dec.1996, pp 215-221.

[6] L. Valet, G. Mauris, P. Bolon, N. Keskes, “A fuzzy rule-based interactive fusion system for seismic data analysis”. *Information Fusion* 4(2), pp: 123-133,2003.

[7] J. Lawry, “A framework for linguistic modelling”, *Artificial Intelligence*, Vol.155 pp.1-39, 2004.

[8] N. J. Randon, “Fuzzy and Random Set Based Induction Algorithms”， Ph.D Thesis, University of Bristol, 2004..

[9] R. J. Hathaway and J. C. Bezdek, “Optimization of clustering criteria by reformulation,” *IEEE Trans. Fuzzy Syst*., vol. 3, No.2, pp. 241–245, 1995.