A Comparative Study on Confidence-Weighted Learning between Fuzzy and Non-Fuzzy Classifiers

Takeshi Sumitani[†] **Tomoharu Nakashima**[†] **Andrzej Bargiela** [‡]
[†]Graduate School of Engineering
Osaka Prefecture University, Japan
[‡]School of Computer Science
The University of Nottingham, U.K.
takeshi.sumitani@ci.cs.osakafu-u.ac.jp
tomoharu.nakashima@kis.osakafu-u.ac.jp
andrzej.bargiela@nottingham.ac.uk

Abstract

The performance of fuzzy and non-fuzzy on-line classifiers are examined in this paper. Confidence-weighted learning is employed as an on-line algorithm. In the case of non-fuzzy classifiers, the linear combination of input attributes is used to determine the class of the input pattern. The weights in the linear combination are modelled using Gaussian distributions. The mean and variance values are modified so that the new training patterns are correctly classified while remembering the information of the previous training patterns. On the other hand, a fuzzy classifier comprises a set of fuzzy if-then rules, in each of which the membership value for the correspondent attribute value is calculated. The membership values for an input vector with the fuzzy if-then rules are then used as an input vector in the linear combination. The weight in the linear combination is specified in the consequent part of each fuzzy if-then rule. A series of computational experiments are conducted to examine the classification performance of the fuzzy and the non-fuzzy on-line learning methods for large scale real-world classification problems that are available in the UCI repository.

Keywords: Fuzzy if-then rules, On-line learning, Pattern classification, Linearly weighted function

1 Introduction

On-line learning plays an important role specifically in a dynamic problem where the decision making for a situation changes over time. It is also useful when a problem size is too big to handle at one time. In this case, it is necessary for a decision making model to learn incrementally using a stream of training patterns where only a portion of the training data set is available at each time step.

There are several approaches to the on-line learning. For example, Crammer et al.[1] proposed passive-aggressive algorithm where parameters of a classification model is updated after a new set of training patterns are available so that the new patterns are correctly classified with a specified maximum error. For fuzzy rule-based classifiers, Nakashima et al.[5; 6; 7] proposed incremental learning algorithms where the internal parameters for determining the weight of the fuzzy if-then rules are updated according to the new set of training patterns.

Dredze et al.[4] proposed a confidence-weighted learning algorithm for linear classifiers. In the confidence-weighted learning algorithm, the weights in the linear discriminant function are modelled not by a real value but by a Gaussian function. The parameters of the Gaussian function such as a mean and a variance is updated so that the amount of the update in Gaussian functions is minimum in terms of KL-divergence. The variance of a Gaussian function represents the confidence of the weight. The larger the variance is, the less confident the weight is. Those weights whose variance of the Gaussian function is large are largely updated while those with a small variance are updated only a little as the weights are already confident.

Fuzzy rule-based systems involves a set of fuzzy if-then rules that represents the domain knowledge. Fuzzy if-then rules can be generated manually through the interview with the domain experts although it is sometimes difficult for the domain experts to quantitatively and qualitatively explain their expertise. If there are available numerical data that represent the domain knowledge, there is a room for learning to automatically generate fuzzy if-then rules from them. The automatic generation of fuzzy if-then rules has been investigated in soft computing community for the last 20 to 30 years. In this paper, an automatic generation method of fuzzy if-then rules is considered to construct fuzzy classifiers.

Fuzzy classifiers have been known for their high classification performance for high-dimensional non-linear problems [2]. A fuzzy classifier is composed of a set of fuzzy if-then rules, which has its antecedent part and consequent part. The antecedent part of fuzzy if-then rules represents a fuzzy covered area in the input space. Each fuzzy set in the antecedent part has a linguistic label (such as "big" and "small") that shows a fuzzy supported interval in the correspondent attribute axis. Therefore, it is possible to linguistically understand the mapping that is achieved by the fuzzy classifier. This is one of the characteristic feature of fuzzy classifiers. This characteristic feature allows to acquire linguistic knowledge from numerical data. The linguistic knowledge acquisition using fuzzy classifiers is one of active research areas in data mining and soft computing [3]. However, there are not many articles that deal with the combination of fuzzy classifiers and on-line learning, especially the confidence-weighted learning.

In this paper, the classification performance of fuzzy and non-fuzzy on-line classifiers are examined. For the on-line learning algorithm, the confidence-weighted learning method is used for both fuzzy and non-fuzzy classifiers. A series of computational experiments are conducted using several large-scale real-world classification problems that are available from the UCI repository [10].

2 Confidence-Weighted Learning for Linear Discriminant Functions

This section describes a confidence-weighted learning algorithm for linear discrimination. First the standard procedure of classification using a linear discrimination function is explained, which is followed by the explanation of the confidence-weighted learning algorithm for training the linear discriminant function. In this paper only binary classification is considered where the number of class labels is just two. An extension to the classifiers for allowing multi-class classification is not our focus of this paper, but will be addressed in our future work.

2.1 Linear Discriminant Function

Let us define without losing generality that an input pattern is represented as an n-dimensional real vector in a hyper cube $[0.0, 1.0]^n$. It is also assumed that the n-dimensional weight vector $\vec{w} = (w_0, w_1, \ldots, w_n)$ is already specified. An unseen pattern $\vec{x} = (x_1, x_2, \ldots, x_n)$ is classified as follows:

$$\begin{cases} \vec{x} \text{ is Class 1,} & \text{if } y > 0.0, \\ \vec{x} \text{ is Class 2,} & \text{if } y < 0.0, \\ \text{Classification rejected,} & \text{otherwise,} \end{cases} \tag{1}$$

where

$$y = w_0 + w_1 \cdot x_1 + \ldots + w_n \cdot x_n. \tag{2}$$

Thus, the classification of unseen patterns is determined according to the sign of the inner product of the weight vector and the augmented input vector $\vec{x'} = (1, x_1, x_2, \ldots, x_n)$.

2.2 Confidence-Weighted Learning

In the confidence-weighted learning, each element (i.e., each weight) in the weight vector $\vec{w} = (w_0, w_1, \ldots, w_n)$ is modelled by a Gaussian function with a mean and a variance. Let us denote that the mean value of the i-th Gaussian function for modelling w_i is μ_i and its variance is σ_i. Then the weights become random variables of the probability distribution with specified mean and variance. When a set of new training patterns become available, the Gaussian distributions are updated so that the classification of the new training patterns are more reliable with a minimum amount of update. During the course of the algorithm execution, the variances of the Gaussian distributions gradually become small.

The update formulation of the Gaussian functions for the weight vector is obtained by solving the following optimization problem:

$$(\overrightarrow{\mu^w_{\text{new}}}, \overrightarrow{\Sigma^w_{\text{new}}}) = \\ \mathop{\arg\min}_{(\overrightarrow{\mu_t}, \overrightarrow{\Sigma_t})} D_{KL}\left(N(\overrightarrow{\mu_t}, \overrightarrow{\Sigma_t}) \parallel N(\overrightarrow{\mu^w_{\text{old}}}, \overrightarrow{\Sigma^w_{\text{old}}})\right), \tag{3}$$

subject to:

$$Pr_{\overrightarrow{w} \sim N(\overrightarrow{\mu}, \overrightarrow{\Sigma})} \left[y_t(\overrightarrow{w} \cdot \overrightarrow{x}) \geq 0 \right] \geq \eta, \quad (0.5 \leq \eta \leq 1.0), \tag{4}$$

where y_t represents the target output value (1 or -1). $\overrightarrow{\mu^w}$ and $\overrightarrow{\Sigma^w}$ are the mean and the variance vectors for modelling $\vec{w}$, respectively. From this optimization problem, the following equation is obtained:

$$\begin{aligned}
\overrightarrow{\mu^w_{\text{new}}} &= \overrightarrow{\mu^w} + \alpha \cdot y_t \cdot \overrightarrow{\Sigma^w} \cdot \overrightarrow{x}, & (5)\\
\overrightarrow{\Sigma^w_{\text{new}}}^{-1} &= \overrightarrow{\Sigma^w}^{-1} & \\
&\quad +2 \cdot \alpha \cdot y_t \cdot \phi \cdot \text{diag}(\overrightarrow{x}), & (6)\\
\alpha &= \max\{\gamma, 0\}, & (7)
\end{aligned}$$

$$\gamma = \frac{-(1+2\phi M)}{4\phi V} + \frac{\sqrt{(1+2\phi M)^2 - 8\phi(M - \phi V)}}{4\phi V}, \tag{8}$$

$$\begin{aligned}
M &= y_t \cdot (\overrightarrow{x} \cdot \overrightarrow{\mu^w}), & (9)\\
V &= \overrightarrow{x}^{\top} \cdot \overrightarrow{\Sigma^w} \cdot \overrightarrow{x}, & (10)\\
\phi &= \Phi^{-1}(\eta), & (11)
\end{aligned}$$

where $\Phi()$ represents the cumulative distribution function of the standard normal distribution. The common specification of the value ϕ is 1.0 in the literature instead of computing it from the probability η. In this case, η is calculated as $\eta \approx 0.84$.

The update procedure is shown in the following:

Step 1: Initialize $\overrightarrow{\mu^w}$ and $\overrightarrow{\sigma^w}$.

Step 2: Obtain a new training pattern $\vec{x}$.

Step 3: Update the mean and variance using Equations (18) and (19).

Step 4: Classify unknown patterns using $\overrightarrow{y}$ to evaluate the performance of the current fuzzy classifier. If any termination criterion are not satisfied, return to Step 3. Otherwise, halt the procedure.

3 Confidence-Weighed Learning for Fuzzy Classifiers

The original work of the on-line learning was applied to linear classification where the attribute values of an input value are linearly combined using Gaussian modelled weights. When a set of new training patterns are obtained, the correspondent Gaussian distributions to the weights are updated so that the linear classifier correctly classify not only the new patterns but also the previous training patterns with least amount of modification. During the course of the algorithm execution, the variances of the Gaussian distributions gradually become small as the confidence of the weights gets higher.

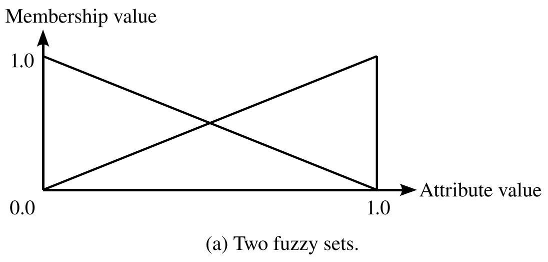
This section describes a fuzzy classifier and its on-line adaptive procedure using confidence-weighted learning. A fuzzy classifier consists of a set of fuzzy if-then rules. As in the last section, it is assumed that the pattern classification problem has the dimensionality n and two classes.

3.1 Fuzzy If-Then Rules

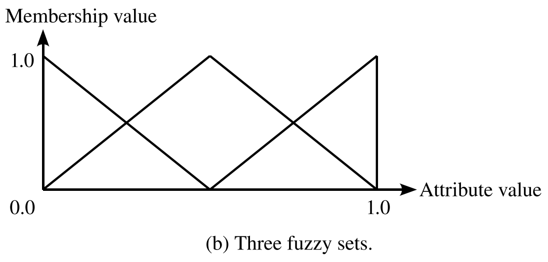
The following type of fuzzy if-then rules is used for our fuzzy classifier:

$$\begin{aligned}
R_q :\ &\text{If } x_1 \text{ is } A_{q1} \text{ and } \cdots \text{ and } x_n \text{ is } A_{qn}\\
&\text{then } y_q, \quad q = 1, 2, \cdots, N
\end{aligned} \tag{12}$$

where R_q is the label of the q-th fuzzy if-then rule, $\overrightarrow{A_q} = (A_{q1}, A_{q2}, \cdots, A_{qn})$ represents a vector of antecedent fuzzy sets, y_q is the consequent real value and N is the total number of generated fuzzy if-then rules. Triangular membership functions are used for antecedent fuzzy sets in this paper. The number of triangular fuzzy membership functions for each attribute must be specified before a fuzzy classifier is constructed. Examples of the triangular fuzzy partition are shown in Figure 1. In Figure 1, two examples of fuzzy partitions are shown where the attribute is divided into two and three triangular fuzzy sets. It should be noted that the fuzzy partition has the effect on the complexity of constructed fuzzy classifiers. The number of generated fuzzy if-then rules in constructing a fuzzy classifier depends on the fuzzy partition for attributes. For example, if the number of triangular fuzzy membership functions is two for a ten-dimensional problem, the number of fuzzy if-then rules is $2^{10} = 1024$ while it is $3^{10} = 59049$ when the number of triangular membership functions is three for each attribute. Although a fuzzy classifier become intractably complicated with a huge number of fuzzy if-then rules, it is able to realize highly non-linear mapping with high accuracy. On the other hand, it is difficult to linguistically understand the input-output mapping that the constructed fuzzy classifier achieves. Thus, the number of fuzzy membership functions should be specified carefully considering the balance between the accuracy and the complexity.



(a) Two fuzzy sets.



(b) Three fuzzy sets.

Figure 1. Triangular fuzzy membership functions.

3.2 Classification of Unknown Patterns

The classification of an unknown pattern is done using the output value that are calculated through fuzzy inference process. In this paper, we classify unknown patterns $\overrightarrow{x} = (x_1, x_2, \cdots, x_n)$ as follows:

$$\begin{cases} \vec{x} \text{ is Class 1}, & \text{if } y > 0.0, \\ \vec{x} \text{ is Class 2}, & \text{if } y < 0.0, \\ \text{Classification rejected}, & \text{otherwise}, \end{cases} \quad (13)$$

where

$$y = y_1 \cdot \mu_1 + y_2 \cdot \mu_2 + \ldots + y_N \cdot \mu_N, \quad (14)$$

where y_q is the consequent real value for the q-th fuzzy if-then rule and $\mu_q(\vec{x})$, $q = 1, 2, \ldots, N$ is the membership value of $\vec{x} = (x_1, x_2, \ldots, x_n)$ with the q-th fuzzy if-then rule R_q. It is calculated as follows:

$$\mu_q(\vec{x}) = \mu_{A_{q1}}(x_1) \cdot \mu_{A_{q2}}(x_2) \cdot \ldots \cdot \mu_{A_{qn}}(x_n), \quad (15)$$

where $\mu_{A_{qi}}(x_i)$, $i = 1, 2, \ldots, n$, $q = 1, 2, \ldots, N$ is the membership value of x_i for the i-th attribute with the rule R_q.

3.3 Updating Consequent Real Values by CW Learning

The confidence-weighted learning is realized for fuzzy classifiers by taking the consequent real values in (14) as the weight in the linear classification (i.e., $w_0, w_1, \ldots, w_n$ in (2)). Thus, in the case of the fuzzy classifiers, the consequent real values in(12) are modelled using Gaussian distributions so that the value y_q of rule R_q is treated as the mean μ_q of a Gaussian distribution with the variance σ_q. Following the case for the linear classification, the update rule of consequent real values is obtained by solving the following optimization problem::

$$(\overrightarrow{\mu^y_{\text{new}}}, \overrightarrow{\Sigma^y_{\text{new}}}) = \underset{(\overrightarrow{\mu_t}, \overrightarrow{\Sigma_t})}{\arg\min} D_{KL}\Big(N(\overrightarrow{\mu_t}, \overrightarrow{\Sigma_t}) \parallel N(\overrightarrow{\mu^y_{\text{old}}}, \overrightarrow{\Sigma^y_{\text{old}}})\Big) \quad (16)$$

subject to

$$Pr_{\overrightarrow{y} \sim N(\overrightarrow{\mu}, \overrightarrow{\Sigma})}\Big[y_t(\overrightarrow{y} \cdot \overrightarrow{x}) \geq 0\Big] \geq \eta, \quad (17)$$
$$(0.5 \leq \eta \leq 1.0)$$

where y_t represents the target output value (1 or -1). $\overrightarrow{\mu^y}$ and $\overrightarrow{\Sigma^y}$ are the mean and the variance of $\overrightarrow{y}$, respectively. From this optimization problem, the following equations are obtained:

$$\begin{aligned} \overrightarrow{\mu^y_{\text{new}}} &= \overrightarrow{\mu^y_{\text{old}}} + \alpha y_t \overrightarrow{\Sigma^y} \overrightarrow{x}, & (18) \\ \overrightarrow{\Sigma^y_{\text{new}}}^{-1} &= \overrightarrow{\Sigma^y_{\text{old}}}^{-1} \\ &\quad +2\alpha \cdot y_t \cdot \phi \cdot \text{Diag}(\overrightarrow{x}), & (19) \\ \alpha &= \max\{\gamma, 0\}, & (20) \end{aligned}$$

$$\gamma = \frac{-(1 + 2\phi M)}{4\phi V} + \frac{\sqrt{(1 + 2\phi M)^2 - 8\phi(M - \phi V)}}{4\phi V}, \quad (21)$$

(22)

Table 1. Details of the data sets that are used in the computational experiments.

Data set	# of patterns	Dimensionality	Classes
Abalone	4177	8	2
Breastcancer	609	10	2
Pima-indians-diabetes	768	8	2
Transfusion	748	4	2

$$M = y_t(\overrightarrow{x} \cdot \overrightarrow{\mu^y}), \tag{23}$$
$$V = \overrightarrow{x}^\top \overrightarrow{\Sigma^y} \overrightarrow{x}, \tag{24}$$
$$\phi = \Phi^{-1}(\eta), \tag{25}$$

where $\Phi()$ represents the cumulative distribution function of the standard normal distribution as in the case of the linear classification. The update procedure is also the same as in the update procedure of the linear classifiers.

4 Computational Experiments

This section examines the performance of the fuzzy and non-fuzzy classifiers with the confidence-weighted learning algorithm. In the experiments of this paper, classifiers are applied to a set of large-scale real-world pattern classification problems that are available from UCI repository [10].

This subsection deals with static but large-scale classification problems. High-dimensional data sets with a large number of training patterns are picked up from the UCI repository [10]. The details of the data sets that are used in the computational experiments in this subsection are shown in Table 1. Since the fuzzy and linear classifiers considered in this paper are designed for two-class pattern classification problem, the selected date sets have only two classes.

4.1 Experimental Settings

Since the purpose of the experiments in this subsection is to examine the classification performance of classifiers with on-line learning, not all training patterns are given for constructing a classifier but only one training pattern becomes available at each time step. Thus, a classifier must be constructed incrementally for each new training pattern. The procedure of this experimental setting is shown below:

Step 1: Randomly specify the parameters of the classifiers (i.e., the weights in the case of the linear classification and the consequent real values in the case of the fuzzy classifiers).

Step 2: Randomly select a training pattern from the training set without replacement.

Step 3: Update classifiers using the selected training pattern.

Step 4: If all the training patterns are selected, terminate the procedure. Otherwise go to Step 2.

The classification performance is evaluated by averaged accuracy. Ten-fold cross-validation was iterated ten times. In the ten-fold cross validation, the entire data set is divided into ten groups and each group is used as test patterns for examining the generalisation ability of the classifiers. This procedure is iterated until all the ten groups are used as test patterns just once. The averaged classification rate for ten trials of the ten-fold cross validation is calculated to use as a performance measure.

4.2 Experimental Results

The results of the experiments are shown in Table 2. Table 2 shows the average classification rates of the fuzzy and linear classifiers on the data sets in Table 1. From the experimental results, it cam be seen that the fuzzy classifier achieved higher classification rate than the linear classifier for three data sets out of the four.

Table 2. Experimental results (Average classification rates over ten trials of 10-CV).

Data set	Fuzzy	Linear
Abalone	0.8772	0.8671
Breastcancer	0.9591	0.9213
Pima-indians-diabetes	0.6766	0.5706
Transfusion	0.7061	0.7319

5 Conclusions

This paper investigates the classification performance of the fuzzy and the non-fuzzy classifiers. Both types of classifiers adopt Confidence-Weighted (CW) learning where the weight parameters in the classifiers are modified so that new training patterns are correctly classified while as many previous patterns as possible are also correctly classified.

The common characteristic feature of the fuzzy and the non-fuzzy classifiers is that both classifiers use a linear combination to obtain the final classification results. In the non-fuzzy classifier, the input values are summed up with their associated weights while in the case of fuzzy classifiers the membership values of fuzzy if-then rules that constitute the classifier are linearly combined. Thus, the number of weight parameters for the fuzzy classifiers are higher than that of non-fuzzy classifiers in general. The advantage of fuzzy classifiers over non-fuzzy ones is that the input-output mapping can be linguistically understood at the cost of a large number of weight parameters.

The computational experiments in this paper used a large-scale pattern classification problems that are available in the UCI repository. It was shown that the generalization ability of the fuzzy classifiers is higher than that of the non-fuzzy classifiers. Thus, it can be said that the fuzzy classifiers have high adaptation ability for incremental classification problems where all training patterns can not be used for constructing a classifier at once.

Future works include other formulation of fuzzy classifiers such as one-class classification and regression problems.

References

[1] K. Crammer and O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online Passive-Aggressive Algorithms," *Journal of Machine Learning Research*, Vol. 7, pp. 551–585, 2006.

[2] H. Ishibuchi, T. Nakashima and M. Nii,*Classification and Modeling with Linguistic Information Granules*, Springer, 2003.

[3] L. I. Kuncheva,"How Good Are Fuzzy If-Then Classifiers," IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics, vol. 30, no. 4, pp. 501–509, August, 2000.

[4] M. Dredze, K. Crammer, and F. Pereira,"Confidence-Weighted Linear Classification,"*Proc. of the 25th International Conference On Machine Learning*, pp. 264–271, 2008.

[5] T. Nakashima, T. Sumitani, and A. Bargiela, "Incremental Update of Fuzzy Rule-Based Classifiers for Dynamic Problems," *Computer and Information Science 2012*, pp.209–219, 2012.

[6] T. Nakashima, T. Sumitani, and A. Bargiela, "Incremental Learning of Fuzzy Rule-Based Classifiers for Large Data Sets," *Proc. of 2012 World Automation Congress*, CD-ROM, 2012.

[7] T. Nakashima, T. Sumitani, and A. Bargiela, "Performance Evaluation of Incremental Fuzzy Rule-Based Classifiers," *Computers, Networks, Systems and Industrial Application,* pp.234–241, 2012.

[8] T. Nakashima, Y. Kuroda, Y. Higuchi, Y. Maekawa, "Occupancy Simulation in an Elderly Monitoring Environment," *Proc. of SIG-ECOmp, SOFT*, pp.30–34, 2013.

[9] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern classification, 2nd edition*, Wiley, New York, Wiley, 2001.

[10] A. Asuncion and D. J. Newman, UCI Machine Learning Repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 2007