Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 22 (2013) 460 – 466

Procedia Computer Science

17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013

A Construction Method of Fuzzy Classifiers using Confidence-Weighted Learning

Tomoharu Nakashima[a,*], Takeshi Sumitani[a], Andrzej Bargiela[b]

[a]*Osaka Prefecture University, Gakuen-cho 1-1, Naka-ku, Sakai, Osaka 599-8531, Japan*
[b]*The University of Nottingham, Wollaton Road, Nottingham, NG8 1BB, United Kingdom*

Abstract

Incremental algorithms for fuzzy classifiers are studied in this paper. It is assumed that not all training patterns are given a priori for training classifiers, but are gradually made available over time. It is also assumed that the previously available training patterns can not be used afterwards. Thus, fuzzy classifiers should be modified by updating already constructed classifiers using the available training patterns. In this paper, a confidence-weighted (CW) learning algorithm is applied to fuzzy classifiers for this task. A series of computational experiments are conducted in order to examine the performance of the proposed method comparing that method with the conventional learning algorithm for fuzzy classifiers.

Keywords: Fuzzy if-then rules; Confidence-weighted; incremental learning; on-line learning

1. Introduction

Fuzzy classifiers have been known for their high classification performance for high-dimensional non-linear problems [1]. 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 value. 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 [2].

Recent development of information and communication technologies results in the increase in information amount that are available to us. Hardware devices such as processing units and fast and large memory devises allow to deal with huge amount of data. It is, however, still difficult to efficiently and effectively process such an intractably large amount of data in a short time with the limited amount of memory size and the limited processing

*Tomoharu Nakashima. Tel.: +81-72-254-9351
E-mail address: tomoharu.nakashima@kis.osakafu-u.ac.jp.

1877-0509 © 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of KES International
doi:10.1016/j.procs.2013.09.124

performance of CPUs. One solution to tackle this issue is online learning where only a small amount of data is processed at a time and then another set of data come over time.

The original work of the online learning focused on linear classification. For example, Confidence-Weighted (CW) learning proposed by Crammer et al. [3] was proposed for making the linear classification capable of incrementally learn through the stream of training patterns. In the CW learning, each of the weights in the linear equation assumes a Gaussian distribution. This means that the set of the weights becomes a set of random variables of probability distribution with a certain mean and variance. When a set of new training patterns come, the correspondent Gaussian distributions are updated. During the course of the algorithm execution, the variances of the Gaussian distributions gradually become small. To the best of our knowledge, online learning using CW learning has not been applied to fuzzy rule-based classifiers.

This paper proposes CW learning for fuzzy classifiers. The effectiveness of the proposed method comparing with the conventional learning algorithm for fuzzy classifiers through a series of computational experiments.

2. Dynamic Classification Problem

In the standard version of pattern classification, the parameters of a classifier are fixed once the learning process completes using a set of training patterns. However, when the number of training patterns are intractably large, it is not practical to train a classifier using the whole set of training patterns as it takes virtually infinite time to use them and also in many cases the memory size necessary for the learning is intractably large. Especially when the characteristic of pattern classification task changes over time, the classifier must deal with more complex issue, dynamical change in classification boundaries. In this paper, this special situation with dynamic classification boundaries is called a dynamic classification problem. The goal of this paper is to propose fuzzy classifiers that can deal with dynamic classification problems.

2.1. Fuzzy If-Then Rules

A fuzzy classifier consists of a set of fuzzy if-then rules. This section describes fuzzy if-then rules for our fuzzy classifiers. Without loss of generality, it is assumed that the pattern classification problem has the dimensionality n and M classes. The following type of fuzzy if-then rules is used for our fuzzy classifier:

$$R_q : \text{ If } x_1 \text{ is } A_{q1} \text{ and } x_2 \text{ is } A_{q2} \text{ and } \cdots \text{ and } x_n \text{ is } A_{qn} \text{ then } y_q, \quad q = 1, 2, \cdots, N \tag{1}$$

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 set of antecedent fuzzy sets, y_q is the consequent value and N is the total number of generated fuzzy if-then rules. we use triangular membership functions as antecedent fuzzy sets.

2.2. 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

$$\overrightarrow{x} \text{ is } \begin{cases} \text{Class1} & \text{if } 0 < \displaystyle\sum_{q=1}^{N} \mu_q(\overrightarrow{x}) \cdot y_q \\ \text{Class2} & \text{if } 0 > \displaystyle\sum_{q=1}^{N} \mu_q(\overrightarrow{x}) \cdot y_q \\ \text{Classification rejected,} & \text{otherwise} \end{cases} \tag{2}$$

where $\mu_q(\overrightarrow{x})$ is the membership value of $\overrightarrow{x}$ in fuzzy if-then rule R_q. It is calculated as follows:

$$\mu_q(\overrightarrow{x}) = \mu_{A_{q1}}(x_1) \cdot \mu_{A_{q2}}(x_2) \cdot \cdots \cdot \mu_{A_{qn}}(x_n) \tag{3}$$

where $\mu_{A_{qi}}(x_i), i = 1, \cdots, n$ is the membership value of x_i in the attribute i of the rule R_q.

2.3. Updating Consequent Real Values by CW Learning

The proposed online fuzzy classifiers are based on CW learning. The consequent real values in (1) are modelled using Gaussian distributions. $\overrightarrow{y} = (y_1, y_2, \cdots, y_q)$ is a set of the consequent values and the value y_q of rule R_q are specified as the mean μ_q and the variance σ_q of the Gaussian distribution functions. The update of consequent values is obtained by solving the following optimization problem::

$$(\overrightarrow{\mu^y_{t+1}}, \overrightarrow{\Sigma^y_{t+1}}) = \underset{(\overrightarrow{\mu}, \overrightarrow{\Sigma})}{\arg\min}\, D_{KL}\left(N(\overrightarrow{\mu}, \overrightarrow{\Sigma}) \;\|\; N(\overrightarrow{\mu^y_t}, \overrightarrow{\Sigma^y_t})\right) \tag{4}$$

$$s.t.\ \ Pr_{\overrightarrow{y} \sim N(\overrightarrow{\mu}, \overrightarrow{\Sigma})}\left[y_t(\overrightarrow{y} \cdot \overrightarrow{x_t}) \geq 0\right] \geq \eta \qquad (0.5 \leq \eta \leq 1.0) \tag{5}$$

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 equation is obtained:

$$\overrightarrow{\mu^y_{t+1}} = \overrightarrow{\mu^y_t} + \alpha_t y_t \overrightarrow{\Sigma^y_t}\overrightarrow{x_t} \tag{6}$$

$$\overrightarrow{\Sigma^y_{t+1}}^{-1} = \overrightarrow{\Sigma^y_t}^{-1} + 2\alpha_t y_t \phi \ \mathrm{diag}(\overrightarrow{x_t}) \tag{7}$$

$$\alpha_t = \max\{\gamma_t, 0\} \tag{8}$$

$$\gamma_t = \frac{-(1 + 2\phi M_t) + \sqrt{(1 + 2\phi M_t)^2 - 8\phi(M_t - \phi V_t)}}{4\phi V_t} \tag{9}$$

$$M_t = y_t(\overrightarrow{x_t} \cdot \overrightarrow{\mu^y_t}) \tag{10}$$

$$V_t = \overrightarrow{x_t}^{\top} \overrightarrow{\Sigma^y_t}\overrightarrow{x_t} \tag{11}$$

$$\phi = \Phi^{-1}(\eta) \tag{12}$$

where $\Phi()$ represents the cumulative distribution function of the standard normal distribution. In this paper, we specify ϕ to 1.0 as commonly used in the literature instead of computing it from the probability η. In this case, η is specified as $\eta \approx 0.84$. The update procedure is shown as follows:

Step 1: Initialize $\overrightarrow{\mu^y_1}$ and $\overrightarrow{\sigma^y_1}$.

Step 2: Update the mean and variance with the expression (6) and (7).

Step 3: 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 2. Otherwise, halt the procedure.

3. Computational Experiments

This section examines the performance of the proposed method and show its effectiveness by comparing with the performance of other online learning methods.

3.1. Experimental Settings

In the computational experiments of this paper, we deal with an artificial two-class two-dimensional dynamic problem in the pattern space $[0.0, 1.0]^2$. The procedure of the experiments are shown in the following:

Step 1: Generate a pattern randomly.

Step 2: Train fuzzy if-then rules.

Step 3: Classify test patterns

Step 4: Rotate the classification boundary by one degree

The above procedure is repeated until the classification boundary makes a 360-degree roll in the counter-clockwise manner. Figure 1 shows how the classification boundary dynamically changes over time. In the phase of evaluation of the current fuzzy classifier, test patterns are generated on the grid that splits each axis into 100 segments as shown Figure 2. By classifying test patterns for each degree of the classification boundary starting from 0 degree (diagonal) to 360 degrees, the generalization performance of the current fuzzy classifier is evaluated.

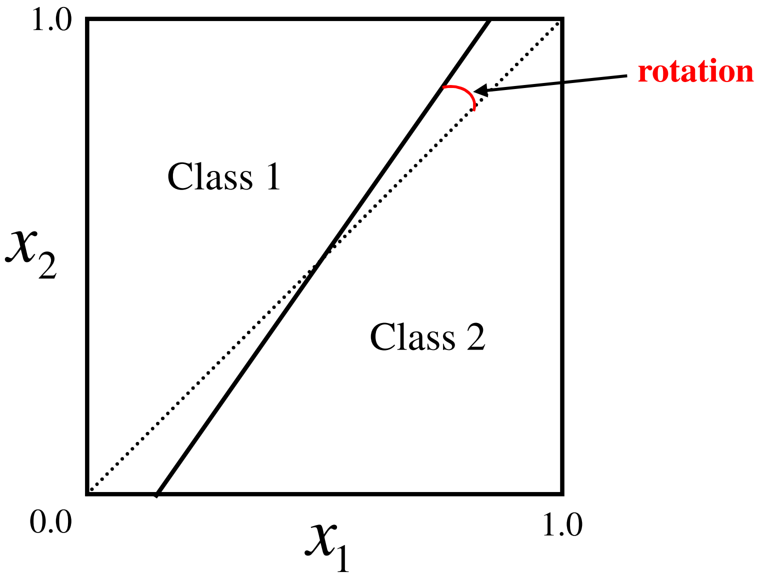

Fig. 1. Dynamic change of classification boundary.

3.2. Performance Comparison

For the purpose of comparing the performance of the proposed method with the other methods, the performance of batch and incremental learning methods for fuzzy classifiers proposed by Nakashima et al. [4], [5], [6] are evaluated. The batch learning and the interpolate learning use the fuzzy if-then rules of following type:

$$R_q: \ \text{If } x_1 \text{ is } A_{q1} \text{ and } x_2 \text{ is } A_{q2} \text{ and } \cdots \text{ and } x_n \text{ is } A_{qn} \ \text{ then Class } C_q \text{ with } CF_q, \quad q = 1, 2, \cdots, N \tag{13}$$

The consequent class and the output value of a fuzzy if-then rule are determined using a set of given training patterns as follows:

$$\beta_h^q = \sum_{\vec{x} \in h} \mu_{A_q}(\vec{x}) \tag{14}$$

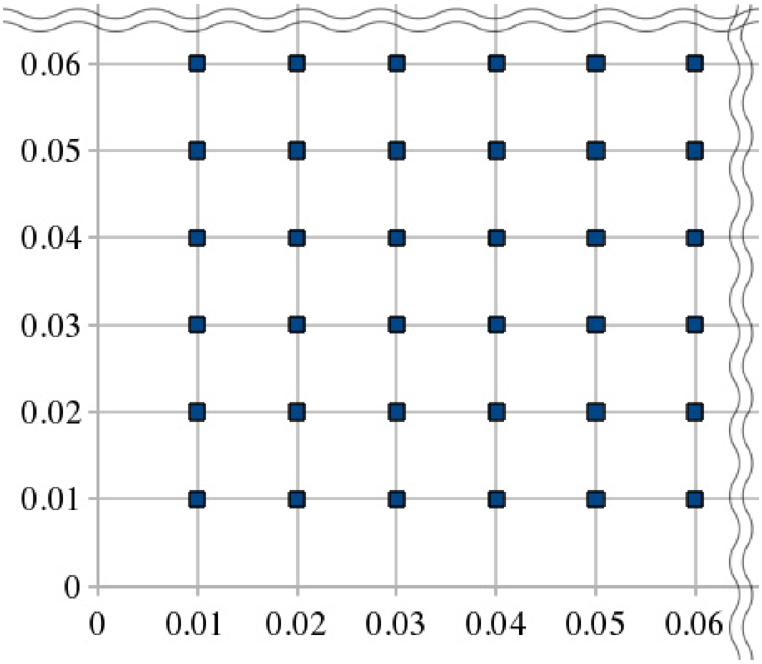

Fig. 2. Grid for generating test patterns.

$$CF_q = \frac{\beta_{C_q} - \bar{\beta}}{\sum_{h=1}^{M} \beta_h^q} \tag{15}$$

$$\overline{\beta} = \frac{1}{M-1} \sum_{h \neq C_q} \beta_h^q \tag{16}$$

3.2.1. Batch Learning

In the batch learning, β_h^q of the q-th fuzzy if-then rule is updated as follows:

$$\beta_h^{q\ new} := \frac{n_h^q \cdot \beta_h^{q\ old} + \mu_q(\overrightarrow{x_p^{t+1}})}{n_h^q + 1}, \ \text{if } \overrightarrow{x_p^{t+1}} \in \text{Class } h \text{ and } \mu_q(\overrightarrow{x_p^{t+1}}) > 0.0 \tag{17}$$

where t is the time and n_h represents the total number of patterns that belong to Class h. As above, the batch learning treats the previous membership values and the new membership values equally.

3.2.2. Interpolate Learning

In the interpolate learning, β_h^q is updated as follows:

$$\beta_h^{q\ new} := (1 - \gamma_\beta) \cdot \beta_h^{q\ old} + \gamma_\beta \cdot \sum_{\overrightarrow{x} \in h} \mu_{A_q}(\overrightarrow{x}) \tag{18}$$

where γ_β is a positive learning rate. In the interpolate learning, the new β_h^q is determined by the interpolation between the old membership values and the new membership values. Thus, the influence of the previous membership values on the update becomes smaller as the time steps increases. In this paper, γ_β is specified as 0.1 using the results of the preliminary experiments.

3.3. *Experimental Results*

Figures 3, 4 and 5 show the transition of the classification accuracy where the horizontal axis represents iterations and the vertical axis represents the classification rate. Figure 3 shows the results of the batch learning.

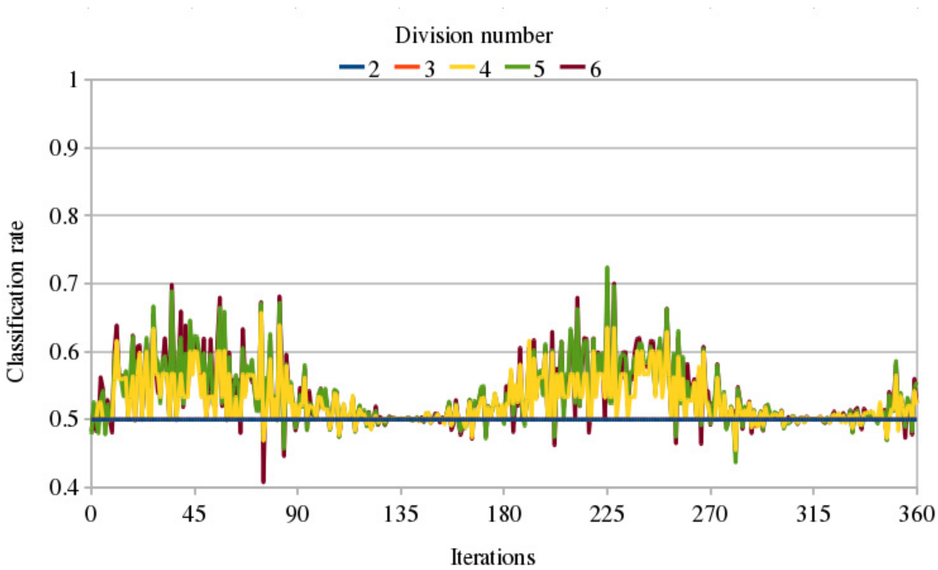

Fig. 3. Batch learning

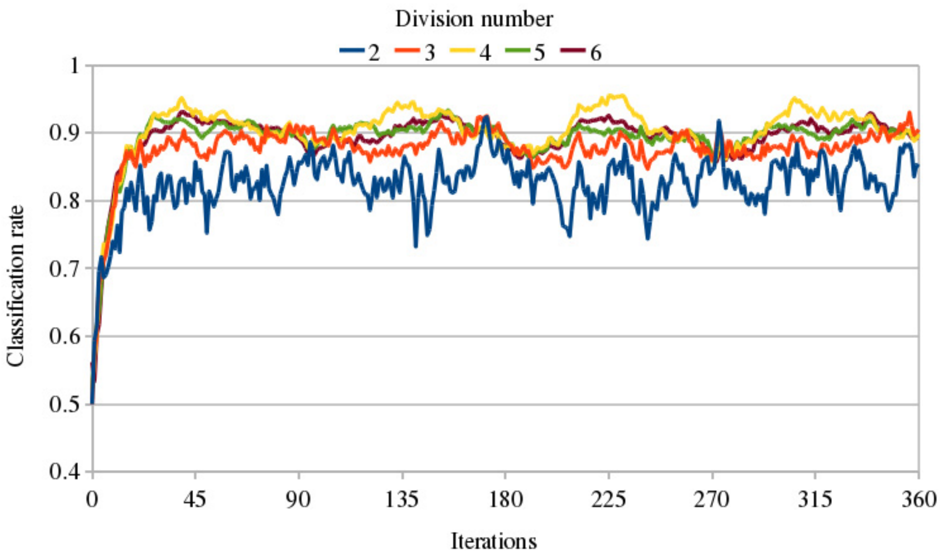

Fig. 4. Interpolate learning

Comparing the results of these methods, the performance of the batch learning is not good. Figure 4 shows the results of Interpolate learning. When the number of fuzzy partitions in this method is large, the classification performance is high. This is because this method appropriately discounts the weight for previously available training patterns and puts more focus on recently available training patterns. Figure 5 shows the results of CW

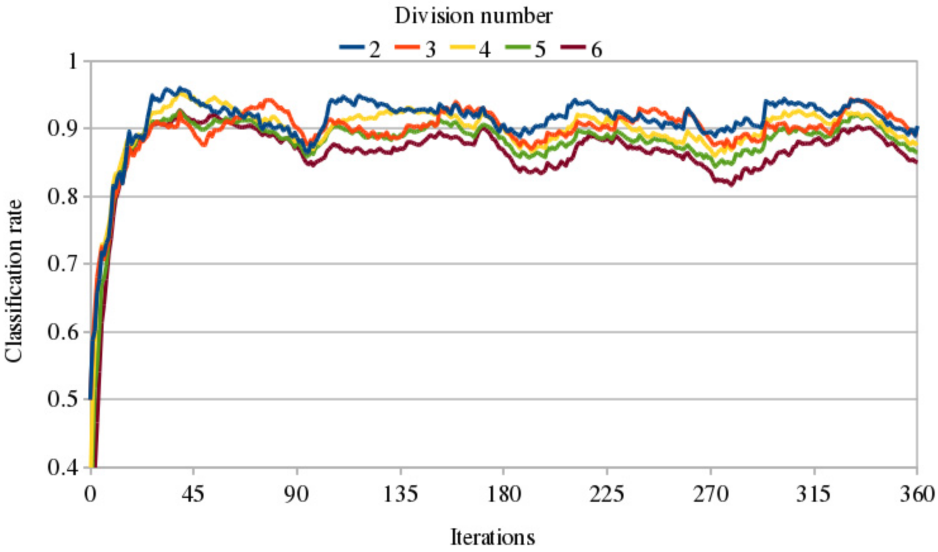

Fig. 5. CW learning

learning. There are no significant differences between the interpolate learning and CW learning in terms of the convergence speed and the classification rate. In this method, however, this results show high performance of this method when the number of fuzzy partitions is small. On the other hand, the classification rate of this method is low when the number of fuzzy partitions is large. This is because the number of involved fuzzy if-then rules becomes larger with the increase of the number of fuzzy partitions becomes large.

4. Conclusions

This paper proposed CW learning that is applied to fuzzy rule-based classifiers. The proposed method was compared with its existing methods. There was no significant difference between CW learning and the other methods in terms of the convergence speed. CW learning, however, showed its high performance which is equivalent to the interpolate learning. The future works include adaptation for multi-class problems and random change of classification boundaries.

References

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

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.