Applied Soft Computing 15 (2014) 231–242

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An accuracy-oriented self-splitting fuzzy classifier with support vector learning in high-order expanded consequent space

Guo-Cyuan Chen, Chia-Feng Juang*

Department of Electrical Engineering, National Chung-Hsing University, Taichung 402, Taiwan, ROC

ARTICLE INFO

Article history:
Received 27 August 2012
Received in revised form 11 January 2013
Accepted 6 November 2013
Available online 22 November 2013

Keywords:
Fuzzy classifiers
Fuzzy neural networks
Support vector machines
Clustering algorithms

ABSTRACT

This paper proposes a self-splitting fuzzy classifier with support vector learning in expanded high-order consequent space (SFC-SVHC) for classification accuracy improvement. The SFC-SVHC expands the rule-mapped consequent space of a first-order Takagi-Sugeno (TS)-type fuzzy system by including high-order terms to enhance the rule discrimination capability. A novel structure and parameter learning approach is proposed to construct the SFC-SVHC. For structure learning, a variance-based self-splitting clustering (VSSC) algorithm is used to determine distributions of the fuzzy sets in the input space. There are no rules in the SFC-SVHC initially. The VSSC algorithm generates a new cluster by splitting an existing cluster into two according to a predefined cluster-variance criterion. The SFC-SVHC uses trigonometric functions to expand the rule-mapped first-order consequent space to a higher-dimensional space. For parameter optimization in the expanded rule-mapped consequent space, a support vector machine is employed to endow the SFC-SVHC with high generalization ability. Experimental results on several classification benchmark problems show that the SFC-SVHC achieves good classification results with a small number of rules. Comparisons with different classifiers demonstrate the superiority of the SFC-SVHC in classification accuracy.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Many classification models have been proposed for pattern classification using numerical data. Examples of the classification models are neural networks (NNs) [1,2], fuzzy classifiers (FCs) [3], and statistical models [4,5], such as a mixture of Gaussian classifier [4] and support vector machines (SVMs) [5]. FCs are based on fuzzy if-then classification rules. Neural networks and evolutionary computation approaches are characterized with optimization ability and have been applied to solve different optimization problems [6–10]. These approaches have also been applied to automate the design of classification rules using numerical data [11–20]. One popular approach is to bring the learning ability of neural networks into a fuzzy system, and the model designed is usually called a fuzzy neural network (FNN) or a neural fuzzy system [11–15]. Another popular approach is to use the optimization ability of genetic algorithms (GAs) for fuzzy rule generation [16–20]. The NN- and GA-based approaches generate fuzzy rules based on empirical risk minimization, which does not account for small structural risk. The generalization performance may be poor when the FC is over-trained.

In contrast to the NN- and GA-based design approaches, a relatively new learning method, the support vector machine (SVM), has been proposed based on the principle of structural risk minimization [5]. Several studies on introducing SVMs into fuzzy-classification-rule generation have been proposed to improve the generalization performance of an FC [21–25]. This paper proposes a self-splitting fuzzy rule-based classifier with support vector learning in expanded high-order consequent space (SFC-SVHC). Based on the self-splitting clustering algorithm in [25], the antecedent parameters in the SFC-SVHC are determined using a variance-based self-splitting clustering (VSSC) algorithm. The SFC-SVHC differs from the NN, GA, and SVM-based FCs above in rule form and consequent parameter learning. That is, contributions of the SFC-SVHC are twofold. First, FCs typically use zero- or first-order TS-type fuzzy rules [11–25], where the consequent of a fuzzy rule is a linear decision function and may restrict the rule discrimination capability. For regression problems, the use of different nonlinear functions in the consequent of a fuzzy rule for regression performance improvement has been recently proposed in [26,27]. This motivates the new idea of expanding the entire rule-mapped consequent space of a first-order TS-type fuzzy classifier, which is used in the SFC-SVHC. Different from the rule forms in previous FCs [11–25], the SFC-SVHC expands the entire rule-mapped consequent space of a first-order TS-type fuzzy system via trigonometric function transformations. The expanded rule-mapped consequent (ERMC) space can be regarded as the inclusion of high-order function terms for

* Corresponding author. Tel.: +886 4 22840688x806; fax: +886 4 22851410.
E-mail address: cfjuang@dragon.nchu.edu.tw (C.-F. Juang).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.11.004

discrimination capability improvement. Second, the SFC-SVHC uses a linear SVM for consequent parameter optimization in the ERMC space. The cost function used in the optimization considers not only training error but also separation margin. The objective of using a linear SVM is to endow the SFC-SVHC with high generalization ability.

The rest of this paper is organized as follows. Section 2 presents surveys on design of FCs. Section 3 introduces the SFC-SVHC structure. Section 4 describes the SFC-SVHC structure learning using the VSSC algorithm. Section 5 introduces SFC-SVHC parameter learning using a linear SVM. Section 6 demonstrates the SFC-SVHC classification performance by applying it to several benchmark classification problems. This section also compares the performance of the SFC-SVHC with those of different classifiers. Section 7 presents discussion. Finally, Section 8 presents conclusions.

2. Literature survey

This section presents surveys of different data-driven FCs using NNs, GAs, and statistical learning. NN-based FCs are typically designed based on structure and parameter learning [11–15]. The neural fuzzy classifier in [11] uses *k*-means [28] for antecedent parameter initialization. The neural fuzzy classifier in [12] starts with a large rule base and a learning algorithm is used to prune the rules. The use of fuzzy *C*-means (FCM) [29] to determine the initial antecedent parameters is suggested in [12]. For the two clustering-based approaches in [11,20], the FC performance depends on the random distributions of the initial cluster centers. The neural fuzzy classifiers in [13,15] use rule-firing strength as a rule-generation criterion for automatic generation of the rules from training data. The maximizing-discriminability-based self-organizing fuzzy network (MDSOFN) in [14] also uses the same rule generation approach as in [13,15]. The characteristic of the MDSOFN is that the consequent parameters are mapped to a linear-discriminant-analysis space to improve the classification discriminability. Parameters in these neural fuzzy classifiers are all tuned using the gradient descent algorithm with the objective of training error minimization.

GA-based FCs use GAs to optimize the structure and parameters in an FC. Several GA-based FCs have been proposed, such as a structural learning algorithm in a vague environment (SLAVE) [16], a fuzzy hybrid genetics-based machine learning (FH-GBML) [17], and a steady-state genetic algorithm for extracting fuzzy classification rules from data (SGERD) [18]. In the SLAVE, an iterative learning algorithm using GA is applied to find the number of rules and the parameters in rules. The FH-GBML uses the hybridization of Michigan and Pittsburgh approaches to optimize an FC. The SGERD uses a new method to extract a compact set of readable fuzzy rules from numerical data in much lower computational efforts than the SLAVE and FH-GBML. The performance of these GA-based FCs depends on the assignment of the initial rule base for selection. Many learning coefficients (more than five) in these GA-based FCs have to be properly assigned in advance for good classification performance. In addition, different runs generate different results due to the stochastic learning property of GAs.

For statistical learning of FCs, the application of a statistical logitboost algorithm to the design of an FC (called LogitBoost) was proposed in [30]. For statistical SVM-based FCs, a positive definite fuzzy classifier (PDFC) was proposed in [21], where a support vector (SV) generates a fuzzy rule. Because the number of SVs in an SVM is usually very large, especially for complex classification problems, the number of rules in a PDFC is equivalently large. The support-vector-based fuzzy neural network (SVFNN) proposed in [22] also builds initial rules from SVs. Then, a learning algorithm is used to remove irrelevant fuzzy rules. However, this rule reduction approach does not maintain the generalization ability and thereby degrades the performance of the original classification model. A fuzzy system learned through fuzzy clustering and SVM (FS-FCSVM) was proposed in [23], where zero-order Takagi-Sugeno (TS)-type fuzzy rules were used. A self-organizing TS-type fuzzy network with support vector learning (SOTFN-SV) was proposed in [24], where first-order TS-type fuzzy rules were used. As in [13,15], these two FCs generate rules based on the rule firing strength of an input sample instead of SVs to achieve a small model size. This kind of structure learning approach generates a new rule and assigns its antecedent part parameters (i.e., center and width of a fuzzy set) according to the location of a single training sample, which does not consider input data distributions around a rule. The VSSC algorithm used in SFC-SVHC determines the antecedent part parameters according to the input data distributions around a rule.

3. SFC-SVHC structure

3.1. SFC-SVHC structure and functions

The SFC-SVHC is based on functional expansion of the ERMC space in a first-order TS-type fuzzy system. Each rule in a first-order TS-type fuzzy system is of the following form:

Rule *i*: IF x_1 is A_{i1} and x_2 is $A_{i2}\cdots$and x_n is A_{in}, then

$$\hat{y} = h_{i0} + \sum_{j=1}^{n} h_{ij}x_j \tag{1}$$

where $x_1, \ldots, x_n$ are inputs, A_{ij} is a fuzzy set, and h_{ij} is a real number. Fig. 1 shows the SFC-SVHC structure, which has a total of six layers. Detailed mathematical functions of each layer are introduced layer by layer as follows.

Layer 1 (*input layer*): Each node in this layer corresponds to one input variable. The node first scales a real input variable to the range [−1,1] and then transmits the scaled value to the next layer. The training data is represented by a labeled set *S* with

$$S = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_N, y_N)\}, \tag{2}$$

where $\vec{x}_k \in \mathbb{R}^n$ and $y_k \in \{+1, -1\}$.

Layer 2 (*fuzzification layer*): Each node in this layer corresponds to a fuzzy set A_{ij} and computes the degree to which an input value belongs to it. Fuzzy set A_{ij} is employed with the following Gaussian membership function:

$$M_{ij}(x_j) = \exp\left\{-\frac{(x_j - m_{ij})^2}{{d_i}^2}\right\} \tag{3}$$

where m_{ij} and d_i denote the center and width of the fuzzy set, respectively. Eq. (3) shows all fuzzy sets in rule *i* share the same width d_i. The SFC-SVHC uses a VSSC algorithm to automatically determine the values of m_{ij} and d_i, details of which are described in Section 4. The number of fuzzy sets in each input variable x_i is equal to the number of fuzzy rules *r*. Therefore, this layer has a total of *nr* nodes.

Layer 3 (*rule layer*): A node in this layer represents one fuzzy rule and performs antecedent matching of a rule. The number of nodes in this layer is equal to the number of rules *r*. Each node performs a t-norm operation on inputs from layer 2 using the algebraic product operation to obtain a firing strength $\mu_i(\vec{x})$. Thus, given an input data set $\vec{x} = [x_1, x_2, \ldots, x_n]$, the firing strength $\mu_i(\vec{x})$of rule *i*

Download English Version:

https://daneshyari.com/en/article/495513

Download Persian Version:

https://daneshyari.com/article/495513

Daneshyari.com