Neurocomputing 000 (2017) 1–11

Contents lists available at ScienceDirect

Neurocomputing

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

Robust subspace neuro-fuzzy system with data ordering

Krzysztof Siminski

Institute of Informatics, Silesian University of Technology, ul. Akademicka 10, Gliwice 44–100, Poland

ARTICLE INFO

Article history:
Received 22 July 2016
Revised 4 December 2016
Accepted 13 January 2017
Available online xxx

Communicated by Prof. Hamid Reza Karimi

Keywords:
Subspace paradigm
Data ordering
Neuro-fuzzy system
Outliers
Noise

ABSTRACT

Neuro-fuzzy systems are known for their ability to both approximate and generalize presented data. In real life data sets not always all attributes (dimensions) of data are relevant or have the same importance. Some of them may be noninformative or unnecessary. This is why subspace technique is applied. Unfortunately this technique is vulnerable to noise and outliers that are often present in real life data. The paper describes a subspace neuro-fuzzy system with data ordering technique. Data items are ordered and assigned with typicalities. Data items with low typicalities have lower influence on the elaborated fuzzy model. This technique makes fuzzy models more robust to noise and outliers. The paper is accompanied by numerical experiments on real life data sets.

1. Introduction

Fuzzy systems use fuzzy logic to handle imprecise data. They can elaborate answers for the presented data. A crucial part of a fuzzy systems is a set of fuzzy rules (fuzzy rule base). Answers of fuzzy rules are aggregated into a final answer of a fuzzy system. In fuzzy systems rule base has to be provided by an expert. Neuro-fuzzy systems are an extension of fuzzy systems. They both use fuzzy logic. But neuro-fuzzy systems can elaborate automatically rules. They are able to create rules for the presented data. They can modify their parameters to minimise the error of the neuro-system. In this aspect neuro-fuzzy systems are similar to artificial neural networks. Many architectures differing in applied fuzzy sets, interpretation of fuzzy implications in rules, tuning techniques etc have been proposed and practically used.

Creation of fuzzy rules in neuro-fuzzy systems with presented data is quite a complicated task. There are three essential methods of automatic creation of rules: grid partition [1], scatter partition (clustering), and hierarchical partition [2–4]. Scatter partition is the most popular method of identification of fuzzy rules. Clustering avoids the curse of dimensionality, which is the main problem of a grid partition. Many good clustering algorithms cannot determine the best number of clusters. This number has to be passed as a parameter of these algorithms. This is their main disadvantage.

In real life data sets not always all dimensions (attributes) are relevant or have the same importance. Some of them may be less important, noninformative, or even unnecessary. Global reduction of dimensionality (e.g., PCA or SVD) may cause problem with interpretability of elaborated model. Local reduction of dimensionality handles attributes in each data granule (cluster) individually. This is the idea of subspace clustering [5–7]. Subspace clustering algorithms can be divided into two classes: algorithms that elaborate crisp subspaces or fuzzy subspaces. The algorithms of the former class select some attributes for each data granule [8–14]. The attributes are assigned weights either 0 or 1 (hard weights). The latter group of algorithms elaborate fuzzy (non binary) subspace granules [6,15–17]. The attributes are assigned weights from interval [0, 1]. In some approaches the attributes are gathered in groups. The weights of attributes in groups are crisp (0 or 1), but the memberships of attributes to groups are fuzzy [18]. One of the problems of the soft weight clustering is its vulnerability to noise and outliers [18]. Main techniques for handling noisy data sets are: modification of clustering algorithm (e.g., applications of *medians* instead of mean (fuzzy C-medians clustering algorithm) as median is an estimator of average value more robust to outliers and noise [19]), modification of objective function in clustering (e.g., possibilistic clustering [20], possibilistic clustering with repulsion of clusters [21], alternative c-means clustering [22], possibilistic clustering with weaker dependency on parameters [23]), application of special metrics (e.g., L_p norms and quasi-norms [24], point-to-centroid distance functions [25]); assignment of noise and outliers to a special noise cluster [26]; evidential clustering with belief function [27] based on the Dempster–Shafer theory; data ordering technique [28].

The problem of robustness to outliers and noise in neuro-fuzzy systems gave rise to new neuro-fuzzy systems. A robust neuro-

E-mail address: krzysztof.siminski@polsl.pl

http://dx.doi.org/10.1016/j.neucom.2017.01.034
0925-2312/

fuzzy system can be exemplified by a system with robust fuzzy regression agglomeration (RFRA) clustering and robust tuning procedure [29]. This system incorporates a cost function of “the errors between the desired output and the output of the corresponding rule instead of the distance between input data to some prototype of the considered cluster” [29]. In the clustering procedure the Tukey's biweight function is used, in the tuning procedure – tanh-estimator loss function. The paper [30] describes the two-step identification of neuro-fuzzy model. In the first step the support vector regression (SVR) technique is used to detect a regression hyperplane robust to outliers. In the second step support vectors are used to initialize a neuro-fuzzy model. The paper [31] presents a learning method tolerant to imprecision and robust to outliers what improves the generalization ability of a neuro-fuzzy system. The method applies an ε-insensitive learning. Unfortunately these neuro-fuzzy systems do not apply the subspace paradigm.

The algorithms based on ordering paradigm are robust to outliers and even their high ratio does not distort the clustering results severely [32,33]. The data items are ordered and their typicality is updated in each iteration of the clustering procedure. The distant items from all prototype centres have lower weights. Outliers and noise data items are assigned low typicality and do not distort the clustering process. The proposed system is based on subspace neuro-fuzzy system with logical implication of fuzzy rules [34]. In Section 2.1, we present an architecture of a subspace neuro-fuzzy system with logical interpretation of fuzzy rules and data ordering technique. Section 2.2 describes the creation of a fuzzy rule base with a fuzzy weighted ordered clustering algorithm and tuning of system's parameters. Finally, Section 3 presents numerical experiments.

2. Subspace neuro-fuzzy system

Fuzzy inference system with parametrized consequences and weights attributes is an extension of the neuro-fuzzy system with parametrized consequences ANNBFIS [35] whose important feature is the logical interpretation of fuzzy implications (cf. Eq. (10)).

2.1. Architecture of system

The system with parametrized consequences is a MISO system. Its rule base $\mathbb{L}$ contains fuzzy rules l in a form of fuzzy implications

$$l : \mathbf{x} \text{ is } \mathfrak{a} \rightsquigarrow y \text{ is } \mathfrak{b}, \tag{1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_D]^{\mathrm{T}}$ is a vector of descriptors of a data item and y is a decision attribute for this data item. Both the descriptors and decision are real numbers.

The text below describes the architecture of the system only for one rule in order to keep the notation simpler.

The linguistic variable $\mathfrak{a}$ in the rule premise is represented as a fuzzy set $\mathbb{A}$ in a D-dimensional space. In each dimension d the set $\mathbb{A}_d$ is described with a Gaussian fuzzy set:

$$u_{\mathbb{A}_d}(x_d) = \exp\left(-\frac{(x_d - \upsilon_d)^2}{2s_d^2}\right), \tag{2}$$

where υ_d is the core location for the dth attribute and s_d is the fuzziness of this attribute. The Gaussian membership function is differentiable in its whole domain what enables application of the gradient descent optimization procedure. The memberships of all attributes (descriptors) are aggregated in order to elaborate the membership $u_{\mathbb{A}}$ of a data item to the premise of the rule. A T-norm is used as an aggregation operator. The attributes have their weights so the weighted T-norm is applied [36]:

$$\begin{aligned} u_{\mathbb{A}} &= T\left(u_{\mathbb{A}_1}, \ldots, u_{\mathbb{A}_D}; z_1^{\phi}, \ldots, z_D^{\phi}\right) \\ &= T\left(1 - z_1^{\phi}(1 - u_{\mathbb{A}_1}), \ldots, 1 - z_D^{\phi}(1 - u_{\mathbb{A}_D})\right), \end{aligned} \tag{3}$$

where $\phi \in (0, 1) \cup (1, +\infty)$ is a weighting exponent for attribute weights. If $\phi = 0$ the mechanism of attribute weights is switched off. High values of ϕ result in maximal weight of one attribute in a rule, whereas all other attributes in the have minimal (zero) weight.

The T-norm is implemented as a product:

$$u_{\mathbb{A}} = T\left(u_{\mathbb{A}_1}, \ldots, u_{\mathbb{A}_D}; z_1^{\phi}, \ldots, z_D^{\phi}\right) = \prod_{d=1}^{D}\left(1 - z_d^{\phi}\left(1 - u_{\mathbb{A}_d}\right)\right). \tag{4}$$

If the formula (4) is applied, the membership of the data items to the rule tends to one independently whether the descriptors have high or low membership. The more attributes each data item has, the more intensely this phenomenon is expressed. Fortunately is can be avoided by an augmentation of weights [34]. The attribute weights for one data item are divided by the maximal values of them. The maximal value is always greater than zero. The augmented values are calculated with formula

$$\hat{z}_{cd} \leftarrow \frac{z_{cd}}{\max_{i \in [1,\ldots,D]} z_{ci}}, \tag{5}$$

where $\hat{z}_{cd}$ is an augmented weight of the dth attribute in the cth cluster. Thus instead of formula (4) we use the following formula:

$$u_{\mathbb{A}} = \prod_{d=1}^{D}\left(1 - \hat{z}_d^{\phi}\left(1 - u_{\mathbb{A}_d}\right)\right). \tag{6}$$

From now on the rule index l will be used again. Thus $u_{\mathbb{A}}$ becomes $u_{l\mathbb{A}}$.

To avoid misunderstandings please keep in mind the meanings of the symbols:

- $u_{\mathbb{A}_d}$ stands for the membership of the dth descriptor to the fuzzy set $\mathbb{A}_d$ in the premise for dth attribute of a certain rule (the index of which we omit here) as in formulae (2)–(4),
- $u_{l\mathbb{A}}$ stands for the membership of the whole data item to the premise of the lth rule.

Combining (2) and (6) we get the value $u_{l\mathbb{A}}$ of the premise of lth rule for data item $\mathbf{x}$:

$$u_{l\mathbb{A}}(\mathbf{x}) = \prod_{d=1}^{D}\left(1 - \hat{z}_{ld}^{\phi}\left\{1 - \exp\left[-\frac{(x_d - \upsilon_{ld})^2}{2s_{ld}^2}\right]\right\}\right), \tag{7}$$

which is a real number: $u_{l\mathbb{A}} \in (0, 1]$.

The term $\mathfrak{b}$, in formula (1), describing the lth rule's consequence is represented by a normal isosceles triangle fuzzy set $\mathbb{B}_l$ with the base width w_l. The localization y_l of the core of the triangle fuzzy set is determined by linear combination of input attribute values with attribute weights taken into account:

$$\begin{aligned} y_l &= \mathbf{p}_l^{\mathrm{T}} \cdot diag\left(\left[1, \mathbf{z}_l^{\mathrm{T}}\right]\right) \cdot \left[1, \mathbf{x}^{\mathrm{T}}\right]^{\mathrm{T}} = \\ &= [p_{l0}, p_{l1}, \ldots, p_{lD}] \cdot \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & z_{l1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & z_{lD} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_D \end{bmatrix}. \end{aligned} \tag{8}$$

The above formula (8) can also be written as

$$y_l = \sum_{d=1}^{D} p_{ld} z_{ld} x_d + p_{l0} = \sum_{d=0}^{D} p_{ld} z_{ld} x_d, \tag{9}$$

where $z_{l0} = 1$ and $x_0 = 1$.

The output $u_{l\mathbb{B}'}$ of the lth rule is the fuzzy value of the fuzzy implication:

$$u_{l\mathbb{B}'}(\mathbf{x}) = u_{l\mathbb{A}}(\mathbf{x}) \rightsquigarrow u_{l\mathbb{B}}(\mathbf{x}), \tag{10}$$

Please cite this article as: K. Siminski, Robust subspace neuro-fuzzy system with data ordering, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.034