Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 279 (2015) 64–86

FUZZY sets and systems

www.elsevier.com/locate/fss

Design of a new adaptive neuro-fuzzy inference system based on a solution for clustering in a data potential field

Sy Dzung Nguyen [a,b,c,1], Seung-Bok Choi [c,*]

[a] *Inha University, Republic of Korea*

[b] *Department of Mechanical Engineering, Industrial University of Ho Chi Minh City, Viet Nam*

[c] *Smart Structures and Systems Laboratory, Department of Mechanical Engineering, Inha University, Incheon 402-751, Republic of Korea*

Received 27 January 2014; received in revised form 21 February 2015; accepted 24 February 2015

Available online 27 February 2015

Abstract

In this study, we propose a new method for building adaptive neuro-fuzzy inference systems (ANFIS) via datasets. In order to improve the performance of conventional ANFIS to handle noisy data, we focus on ameliorating the cluster-data space established from a given dataset. To achieve this, we propose a weighted clustering process in the joint input–output data space. Thus, during the clustering process, the cluster with the smallest potential distance, which is a combination of the Euclidean distance and the size of the clusters, has priority when obtaining the surveyed sample. Based on this principle, we formulate a new algorithm for synthesizing an ANFIS via the proposed data potential field, called ANFIS-PF, which has the following features: it establishes a data potential field that covers the entire initial data space, a cluster-data space is built based on the generated data potential field, and the ANFIS is synthesized using this cluster-data space. Finally, we performed experiments using datasets with and without noise to demonstrate the effectiveness of the proposed method in several applications, including dynamic-response noisy datasets obtained from a magnetorheological damper.

Keywords: Adaptive neuro-fuzzy inference system; Clustering data; Fuzzy model identification; Hyper-box cluster; Hyper-plane cluster; Potential field

1. Introduction

Fuzzy logic has been applied successfully in many different areas, especially complex systems, which are very difficult or impossible to describe mathematically with the usual approaches, such as identifying nonlinear systems, processing images, or controlling nonlinear engineering systems [1–5]. In order to build a fuzzy system via datasets, i.e., a data-driven model, the main steps are as follows. First, we select an appropriate fuzzy model to establish a fuzzy structure and estimate its parameters, such as the fuzzy model proposed by Takagi and Sugeno [6] (the T–S

* Corresponding author.
E-mail addresses: nsidung@yahoo.com (S.D. Nguyen), seungbok@inha.ac.kr (S.-B. Choi).

[1] Postdoctoral fellow at Inha University, Republic of Korea.

http://dx.doi.org/10.1016/j.fss.2015.02.012

model) and the switching regression model (the C-regression model) presented by Hathaway and Bezdek [7]. Next, a cluster-data space is usually built based on a reasonable clustering solution, where a fuzzy structure is created to express the relationship between the input and output spaces. In this process, the dataset is separated into data clusters of similar individuals [8–20]. This can be implemented in the input data space, output data space, input–output data space, or in the joint input–output data space based on different strategies, such as the hard C-mean [14,19,21] or fuzzy C-mean (FCM) [10,13,22,23].

During clustering in the input data space, the input space of the dataset is first separated to build input data clusters, and the data clusters are then established based on the distribution status created in the input space [8–10]. Thus, Kasabov et al. [10] presented a dynamic evolving neural-fuzzy inference system (DENFIS) using the FCM. The DENFIS model was developed using the evolving clustering method (ECM). The idea of the ECM is that depending on the position of the input vector in the input space and the fuzzy rules created during the past learning process, the fuzzy inference system used for calculating the output is produced dynamically. Obviously, this strategy does not consider the opposing relationship between the output samples and the corresponding input-data points that are available in the data space. Indeed, the synthesis of the fuzzy system depends mainly on the regularity properties of the input–output mapping. Hence, using this method, the fuzzy system created will be perfect if the close relationships between data points in a cluster in the input space are mapped exactly onto close points in the corresponding cluster in the output space, and vice versa. Unfortunately, this is not always the case because of several reasons such as the noisy status of the dataset. Thus, in this case, the distributive features of the data space in both the input and output space are not reflected satisfactorily by the clusters created. As a result, contradictory rules may exist in the fuzzy structure built using this cluster-data space, which reduces the effectiveness of the fuzzy system [19].

A solution to the problem mentioned above is clustering in the input–output data space, which considers both the input and output spaces during separation [12,13]. Kasabov et al. [13] proposed the use of evolving fuzzy neural networks (EFuNNs) as adaptive intelligent systems. Based on the FCM, a neuro-fuzzy structure with five layers is obtained, where the third layer contains rule nodes. Each rule node is represented graphically as an association among hyperspheres from the fuzzy input and fuzzy output spaces. Rule node $\bar{r}_j$ is defined by two vectors of connection weights, where $W_1(\bar{r}_j)$ are the coordinates of the center of the sphere in the fuzzy input space, and $W_2(\bar{r}_j)$ are the coordinates in the fuzzy output space. The radius of the input hypersphere of a rule node is defined as R_j. The rule nodes are considered to be data clusters that represent prototypes of input–output data associations. The rule nodes are then updated and modified during the learning process via a knowledge-based technique called *rule node aggregation*. Let us consider an input–output data point $(\bar{x}_i, y_i)$, the fuzzy input–output data vector of which is signed $(\bar{x}_i^f, y_i^f)$. The vector $(\bar{x}_i^f, y_i^f)$ will be fixed to the rule node $\bar{r}_j$ if the local normalized fuzzy difference between $\bar{x}_i^f$ and $W_1(\bar{r}_j)$ is smaller than R_j, and the normalized output error is smaller than an error threshold, $E_i = \|\hat{y}_i - y_i\|/N_{out} \le [E]$, where $\hat{y}_i$ is the output value of the EFuNNs and N_{out} is the number of outputs. It can be seen that clustering in the input–output data space, as mentioned above, is also based on the data relationship from the input to the output in only one dimension. This is appropriate if the mapping f from the input data space X onto the output data space Y, and the converse mapping f^{-1}, is a univalent 1–1 mapping, which means that:

$$\begin{aligned} & f: X \to Y \\ & \quad \bar{x}_i \mapsto y_i = f(\bar{x}_i) \\ & \forall \bar{x}_k, \bar{x}_h \in X: \bar{x}_k \neq \bar{x}_h \Rightarrow f(\bar{x}_k) \neq f(\bar{x}_k) \quad \text{or} \quad \forall \bar{x}_k,\ \bar{x}_h \in X: f(\bar{x}_k) = f(\bar{x}_k) \Rightarrow \bar{x}_k = \bar{x}_h, \end{aligned} \tag{1}$$

where this can also be expressed as, $f^{-1}: Y \to X$. However, in the case where the mapping f (or f^{-1}) is not a univalent 1–1 mapping, the following cases may occur:

$$\begin{aligned} & \exists \bar{x}_k, \bar{x}_h \in X: \quad \bar{x}_k \neq \bar{x}_h,\ \text{but } f(\bar{x}_k) = f(\bar{x}_k) \\ & \exists y_k, y_h \in Y: \quad y_k \neq y_h,\ \text{but } \bar{x}_k = \bar{x}_h \text{ in which } \bar{x}_k = f^{-1}(y_k);\ \bar{x}_h = f^{-1}(y_h). \end{aligned} \tag{2}$$

This usually affects nonlinear systems where the database is based on datasets that always contain noise. Thus, in this case, the clustering process mentioned above is certainly not appropriate. This reduces the accuracy and effectiveness of the created fuzzy system. In order to overcome the problems related to (2), clustering in the joint input–output space is considered.