Pattern Recognition Letters 67 (2015) 138–144

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

A new method for constructing granular neural networks based on rule extraction and extreme learning machine☆

Xinzheng Xu[a,b,*], Guanying Wang[a], Shifei Ding[a,b], Xiangying Jiang[a], Zuopeng Zhao[a,b]

[a] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
[b] Jiangsu Key Laboratory of Mine Mechanical and Electrical Equipment, China University of Mining & Technology, Xuzhou 221116, China

ARTICLE INFO

Article history:
Available online 30 May 2015

Keywords:
Granular neural networks
Single-hidden layer feedforward neural networks
Rough set
Rule extraction
Rough rule granular extreme learning machine

ABSTRACT

This paper introduces a framework of granular neural networks named rough rule granular extreme learning machine (RRGELM), and develops its comprehensive design process. The proposed granular neural networks are formed on the basis of rough decision rules extracted from training samples through rough set theory. Firstly, Sample data are reduced by the algorithms of attributes reduction and attributes values reduction in rough set theory, and then they are compressed to an irredundant data set. In this data set, each sample can represent a rough rule, and is expressed as an If-Then rule which indicates the relationship between the input and output pattern. Moreover, the confidence level and the coverage level of each rule are calculated. Secondly, granular-neurons can be constructed through the If-Then rules, and all the granular-neurons constitute rule matching layer which is regarded as the hidden layer of the RRGELM. The linked weights between the input neurons and granular-neurons can be determined by the confidences of rough decision rules, while the linked weights between the output neurons and granular-neurons can be initialized as the contributions of the rough rules to the classification. Finally, the extreme learning machine (ELM) algorithm is introduced to improve the learning speed of the RRGELM, rather than the BP algorithm used by other traditional GNN models. Good performance of the proposed RRGELM is demonstrated on several well-known benchmark data sets.

1. Introduction

As an important class of nonlinear and highly adaptable systems, artificial neural networks (ANNs) display many excellent performances throughout the applications of many areas, such as parallel processing, adaptivity, robustness, ruggedness, generalization, and insensitivity to noise. Generally speaking, ANNs are regarded as black boxes and nontransparent models owing to the distributed style of the underlying information processing. As a consequence, it is very difficult to explain their internal learning processes.

Granular computing (GrC) [26] is an emerging computation theory for effectively using granules, such as classes, clusters, subsets, groups and intervals, to build an efficient computational model for complex applications with huge amounts of data, information and knowledge. Specifically, GrC attempts to imitate the human ability to reason with abstract quantities and groups rather than numeric precision [6]. GrC provides a novel framework for information processing, in which computation and opera here are a number of GrC models, such as sets (interval analysis), rough set, fuzzy set, shadowed sets and probability calculus [8]. There are a number of GrC models, such as sets (interval analysis), rough set, fuzzy set, shadowed sets and probability calculus [20].

As the organic integration of above two technologies, Granular neural networks (GNN), introduced by Pedrycz and Vukovich [20], is a framework to process information granules. By considering sets of objects sharing commonalities and imprecise data items instead of precise singular data items, GNN avoids processing detailed and costly data [14]. GNN does not need to consider all data, which are far more numerous than granules. Furthermore, data can be discarded whenever they match an existing granule. For the past few years, several researchers have described different architectures of GNN. Zhang et al. described two different types of GNN, crisp GNN (CGNN) built on CNN and fuzzy GNN (FGNN) built on FNN in [29], to deal with numerical-linguistic data fusion and granular knowledge discovery in numerical-linguistic databases. Pal et al. presented a rough self-organizing map (RSOM) with a fuzzy discretization of feature space [18]. GNN with the high-speed evolutionary interval learning is designed by Zhang et al., to deal with different membership functions of the same linguistic term [30]. Ganivada et al. introduced a fuzzy rough GNN (FRGNN) model based on the multilayer perceptron using a back-propagation algorithm for the fuzzy classification of patterns [8]. Leite et al. proposed an evolving GNN (eGNN) model, which can build interpretable multi-sized local models using fuzzy neurons for

☆ This paper has been recommended for acceptance by Sankar Kumar Pal and Gabriella Sanniti di Baja.

* Corresponding author. Tel.: +86 159 5215 1616, +86 516 83591726.
E-mail address: xuxinzh@163.com, xxzheng@cumt.edu.cn (X. Xu).

http://dx.doi.org/10.1016/j.patrec.2015.05.006
0167-8655/

information fusion using an online incremental learning algorithm [14]. Song et al. proposed a concept of a granular neural network and develop its comprehensive design process, which is elaborated on with the two main aspects stressed, namely an optimal allocation of information granularity and an objective function guiding the allocation process [17]. Furthermore, in the latest decade, GNN had been used to many fields successfully, such as approximation and prediction of wages [16], land use classification [24], modeling of the charging characteristic [19], stock prediction [28], fast credit card fraud detection [22], satellite image classification [21], imbalanced data classification [4], semi-supervised data stream classification [12], fuzzy time series forecasting [13], early warning of enterprise decline [2] and so on.

In the paper, we introduce a method for constructing GNN within the natural computing paradigm, where the structure of granulation is defined on the base of rough set theory. As a typical model of Grc, rough set (RS) theory has proved to be a powerful tool for uncertainty questions and it has been applied to data reduction, rule extraction, data mining and granularity computation [2]. RS apply the unclear relation and data pattern comparison based on the concept of an information system with indiscernible data, where the data is uncertain or inconsistent [5]. Recently, rough set theory has integrated with neural networks, called rough neural networks (RNN) [3,5,9,15,23,25,27,31]. Generally speaking, there are two integration patterns for RNN, one typical approach is to use RS approach as a pre-processing unit, and another is that RS is used to gain rules from data sets to construct granular-neurons. In the framework of granular-neurons, integration research has been made in the use of rough set for encoding weights of knowledge-based networks. However, the first integration pattern has been considered so far.

This article is an attempt to incorporate the rough set methodology in the framework of granular-neurons to construct RNN using rules obtained by RS. In the proposed method, the attributions of data sets are firstly reduced by RS theory to remove redundant attributes. Then, on the base of reduced data sets, the reduction of attribution values is followed to remove redundant samples. After above two processes, each sample represents a rough decision rules, and all rough decision rules are extracted from reduced samples. Furthermore, the confidence level and the coverage level of each decision rule can be calculated. After this, granular-neurons can be constructed by rough decision rules, and the weights between input neurons and granular-neurons can be determined by the confidences of decision rules. Here, we name the layer which composed of granular-neurons as the rule matching layer. So, the proposed model contains three layers: the input layer, the rule matching layer, and the output layer. In addition, the extreme learning machine (ELM) algorithm, developed by G.-B. Huang et al. [11], is introduced to replace the traditional back propagation (BP) algorithm, which is often used by other GNN models.

The remainder of this paper is structured as follows. A brief description of rough set theory and the ELM algorithm are provided in Section 2. In Section 3, we introduce extraction processes of decision rules from the decision table. We use Zoo data set from UCI data sets [1] as a straightforward example to explain how to gain If-Then rules through attribute reduction and value reduction. Structures and learning process of RRGELM are detailed in Section 4. We present our performance analysis experiments in Section 5. The paper is concluded in Section 6.

2. Theoretical backgrounds

2.1. Rough set theory

2.1.1. Basic conception

The basic concept in rough set theory is an information system which can be expressed by a 4-tuple $S = \langle U, A, U, f\rangle$, where U is a finite set of objects, called the universe; A is a finite set of attributes, $V = \cup_{a\in A} V_a$ is a domain of attribute a, and f: $U \times A \to V$ is an information function such that $f(x, a) \in V_a$, for $\forall a \in A$, $\forall x \in U$. In the classification problems, an information system is also seen as a decision table, assuming that $A = C \cup D$ and $C \cup D = \varnothing$, where C is a set of condition attributes and D is a set of decision attributes.

Let $S = \langle U, A, V, f\rangle$ be an information system, every $P \subseteq A$ generates a binary relation on U called an indiscernibility relation, denoted by $IND(P)$ which is defined as follows:

$$IND(P) = \{(x, y) \in U \times U : f(x, a) = f(y, a), \forall a \in P\} \tag{1}$$

The partition of U generated by $IND(P)$ is denoted by $U/IND(P) = C_1, C_2, \ldots, C_k$, for every C_i is an equivalence class. For $\forall x \in U$ the equivalence class of x in relation $U/IND(P)$ is defined as follows:

$$[x]_{IND(P)} = \{y \in U : f(x, a) = f(y, a), \forall a \in P\} \tag{2}$$

Let $X \subseteq U$ be a target set and $P \subseteq A$ be an attribute subset, The P-lower approximation of X(denoted by P_*X) and P-upper approximation of X (denoted by P^*X), are defined respectively as follows:

$$P_*X = \{y \in U : [y]_{IND(P)} \subseteq X\} \tag{3}$$

$$P^*X = \{y \in U : [y]_{IND(P)} \cap X \neq \varnothing\} \tag{4}$$

The P-boundary (doubtful region) of set X is defined as follows:

$$Bn_p(X) = P^*X - P_*X \tag{5}$$

P_*X is the set of all objects from U which can be certainly classified as elements of X, employing the set of attributes P. P^*X is the set of objects from U which can be possibly classified as elements of X, using the set of attributes P.$Bn_p(X)$ is the set of objects which cannot be certainly classified to X using the set of attributes Ponly.

2.1.2. Attribute reduction of decision table

Let P, $Q \subseteq A$, the dependency of attribute set Q on attribute set P, $\gamma_P(\mathcal{Q})$, is given by

$$\gamma_P(\mathcal{Q}) = \frac{card(\bigcup_{x\in U/IDP(Q)} P_*X)}{card(U)} \tag{6}$$

where $\bigcup_{x\in U/IDP(Q)} P_*X$ can be denoted as $POS_P(Q)$, which means that the objects in it can be classified to one class of the classification$U/IDP(Q)$ by attribute P.

An attribute a is said to be dispensable in $P$with respect to Q, if $\gamma_P(\mathcal{Q}) = \gamma_{P-\{a\}}(\mathcal{Q})$; otherwise a is an indispensable attribute in P with respect to Q.

Let $S = \langle U, A, V, f\rangle$ be a decision table, the set of attributes $P(P \subseteq C)$ is a reduction of attribute C, which satisfies the following conditions:

$$\gamma_P(\mathcal{Q}) = \gamma_C(\mathcal{Q}) \text{ and } \gamma_P(\mathcal{D}) \neq \gamma_{P'}(\mathcal{Q}), \quad \forall P' \subset P \tag{7}$$

A reduction of condition attributes C is a subset that can discern decision classes with the same accuracy as C, and none of the attributes in the reduction can be eliminated without decreasing its distrainable capability.

2.1.3. Value reduction of decision table

After attribute reduction of the decision table, redundant attributes in decision table are removed and those attributes which can influence the classification ability are preserved. Those attributes and attribute sets which have direct impact on decision classification can be picked up immediately based on the reduced decision table. However, the attribute reduction algorithm can only remove the redundant attributes, so the other redundant information cannot be removed. Value reduction furthers the procession of the reduced decision table after attribute reduction. Typical value reduction algorithms include general the value reduction algorithm and heuristic value reduction algorithm. Next, we introduce the latter.

The procedures of the heuristic value reduction algorithm are detailed as follows: