Pattern Recognition Letters 80 (2016) 76–83

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Design of self-adaptive and equilibrium differential evolution optimized radial basis function neural network classifier for imputed database☆

Ch. Sanjeev Kumar Dash[a], Amitav Saran[a], Pulak Sahoo[a], Satchidananda Dehuri[b,*], Sung-Bae Cho[c]

[a] Silicon Institute of Technology, Silicon Hills, Patia, Bhubaneswar 751024, Odisha, India
[b] Department of Systems Engineering, Ajou University, San 5, Woncheon-dong, Yeongtong-gu, Suwon 443-749, South Korea
[c] Soft Computing Laboratory, Department of Computer Science, Yonsei University, 50 Yonsei-ro, Sudaemoon-gu, Seoul 120-749, South Korea

ARTICLE INFO

Article history:
Received 1 January 2015
Available online 13 May 2016

Keywords:
Data mining
Imputation
Classification
Radial basis function neural networks
Differential evolution
k-nearest neighbor

ABSTRACT

The occurrence of missing values is not uncommon in real life databases like industrial, medical, and life science. The imputation of these values has been realized through the mean/mode of known values (for a quantitative/qualitative attribute) or nearest neighbors. Mean based imputation considerably underestimates the population variance and tends to weaken the attribute relationships. Similarly, the nearest neighbor approach uses only information of the nearest neighbors and leaving other observations aside. Hence to overcome the shortcomings of these methods, we have introduced a method known as medoid based imputation to impute missing values. Further, to achieve better performance, we have devised a novel classifier for imputed datasets, by using the self-adaptive control parameters of differential evolution (DE) with equilibrium of exploitation and exploration optimized radial basis function neural networks (RBFNs). By newly associating a weight parameter with target vector during mutation, we maintain equilibrium on the exploration and exploitation mechanism of DE. The self-adaptive equilibrium DE (SAEDE) is used to explore and exploit the suitable kernel parameters of RBFNs along with bias and then used for classifying unknown samples. The performance of the proposed classifier named as SAEDE-RBFN has been extensively evaluated on seven datasets retrieved from University of California, Irvine (UCI) and KEEL machine learning repositories after imputation by mean, nearest neighbor, and proposed method. The average performance of classifiers has been listed based on the imputation by K-nearest neighbor (Knn = 1, Knn = 3, Knn = 5, and Knn = 7), mean, and medoid, respectively. Outcome of the experimental study shows that the performance of SAEDE-RBFN on medoid based imputed dataset is relatively better than DE-RBFN.

1. Introduction

In Data Mining [1] and Big Data analysis [2], classification, prediction, association rule mining, and clustering are identified as fundamental activities in dredging knowledge from the real life databases. Over the years, for each of these tasks, many models have been proposed with their associated pros and cons [3]. However, the common consensus is that the accuracy of the model depends highly on the quality of the data being mined. In many application areas, quality datasets are difficult to obtain and are small in size. It is not uncommon to encounter industrial and life science databases with half of the entries of a particular instance missing [4,5]. It has been observed that some of the datasets contain only a few samples with missing values in useful attributes. For example, in patient diagnosis data, we often have missing values [6–8] corresponding to a particular feature that would have helped in predicting the disease and subsequent treatment. There are various reasons for these missing values, such as human errors, machine errors, and incorrect measurements.

In the last decades, many approaches have been developed to impute missing values [4]. The simplest technique used in many literatures is to remove the samples with missing attributes [9] and use the remaining samples for modeling. But, there can be substantial loss of information due to simple deletion of samples with missing attributes, especially in the case of small and imbalanced datasets. Moreover, this method is practical only when the data

☆ This paper has been recommended for acceptance by L. Heutte.

* Corresponding author. Tel.: +91 966 832 1964; fax: +82 2365 2579.
E-mail addresses: sanjeev_dash@yahoo.com (Ch.S.K. Dash), amitavsaran@yahoo.com (A. Saran), sahoo_pulak@yahoo.com (P. Sahoo), satchi@ajou.ac.kr (S. Dehuri), sbcho@yonsei.ac.kr (S.-B. Cho).

http://dx.doi.org/10.1016/j.patrec.2016.05.002
0167-8655/

contain relatively small number of samples with missing values and the analysis of the complete data will not lead to serious bias during the inference.

Many researchers [10–14] are working on how to handle these missing values (Farhangfar et al., and [15]) in both training and testing datasets. Imputation is a very popular approach to replace missing attribute values. There are two main types of imputation approaches: (1) Value imputation and (2) Distribution-based imputation. Value imputation estimates a value to be used by the model in place of the missing feature [16, 17] and is common in statistical community. In contrast distribution-based imputation estimates the conditional distribution of the missing value ([18]; Grzymala-Busse and Hu, 2001; [19, 20]) and the prediction will be based on this estimated distribution.

In this paper, we propose a new way of imputing the missing values called medoid based imputation. In addition, we have tried multiple imputation approaches to handle missing values and avoid simple removal of samples which will result in useful information being lost. We have applied imputation by k-nearest neighbor (Knn = 1, Knn = 3, Knn = 5, and Knn = 7) [21] and mean techniques to study their effect on classification performances [22–26]. After imputing the datasets, our self-adaptive equilibrium DE optimized RBFN is used to study the influence of the imputed techniques such as k-nearest neighbor, mean, and medoid. We have conducted the simulation on seven benchmark datasets obtained from University of California, Irvine (UCI) machine learning repository [27] and KEEL repository.

2. Background

The background of this research work (i.e., missing data imputation and classification, the salient features of RBFN, and DE) are discussed distinctly in Sections 2.1–2.3, respectively.

2.1. Missing data imputation and classification

Missing data [28, 29] are source of common problems in all type of research work. Imputation techniques are based on the idea that any subject in a study can be replaced by a new randomly chosen subject from the same source. Imputation of missing data on a variable is replaced by a value that is drawn from an estimate of the distribution of this variable. In methods like complete or available case analysis, the missing indicator, and overall mean imputation lead to inefficient analysis and more seriously, produce severely biased estimates of the association(s) investigated. Hence, to reduce the biased-ness or increase the quality of analysis, we have proposed the medoid based imputation. Additionally, the Knn based analysis is also been examined. The problem of classification [30] is basically the core of partitioning the feature space into regions, one region for each category of inputs. Classifiers are generally, but not always designed with labeled data, in which case these problems are sometimes referred to as supervised classification (where the parameters of a classifier are learnt). In general, supervised classification with missing data [31] focuses on two distinct tasks: handling missing values (i.e., imputing values) and pattern classification [12]. Table 1 illustrates a few imputation methods and usages of some of the machine learning classifiers on imputed database. However, it is evident that not a single literature has been suggested the medoid based imputation.

Table 1
Imputation method and usage of classifiers on imputed dataset.

Authors	Imputation method	Classifiers
[46]	Mode, C4.5, LERS ML	LERS
[31]	10-NNI, Mean, Mode	C4.5, CN2
[30]	Case Deletion, Mean, Median, Knn	LDA, Knn
[24]	GA, Mean, HD	MLP + RBF
[4]	KSOP	KSOP
[47]	Mean, HD, FNB NB, FHD	RIPPER, SVM, C4.5, Knn, NB
[48]	MI-Knn	Knn, MI-Knn
[21]	Knn, SKnn, IKnn, K-Means, EAC	MLP, NB, J4.8, Knn

2.2. Radial basis function networks

The RBFN is a two layered network [32], where each hidden unit of hidden layer implements a radial activation function and each output unit of output layer implements a weighted sum of hidden units' output. This network is a special class of neural network in which the activation of a hidden neuron is determined by the distance between the input and a prototype vector. Prototype vectors refer to centers of clusters formed by the patterns in the input space. The centers are determined during RBFN training. Two parameters are associated with each RBFN node, the center and the width. In the input layer, the number of input neurons is determined based on the input features that connect the network to the environment. As stated, the second layer (hidden layer) consists of a set of kernel units that carry out a nonlinear transformation from the input to the hidden space. Usually, a nonlinear transformation is made based on Gaussian kernel as described in Eq. (1).

$$\varphi_i(x) = \exp\left(-\frac{\|x-\mu_i\|^2}{2\sigma_i^2}\right) \tag{1}$$

where ||...|| represents Euclidean norm, μ_i, σ_i, and φ_i are center, spread, and the output of ith hidden unit, respectively.

Radial basis function networks use in literature is extensive and its application varies from pattern classification to time series prediction. Training of RBF network involves two steps: (1) the basis function parameters corresponding to hidden neurons are determined by clustering and or heuristic method; (2) the final-layer weights are determined by least square which reduces to solve a simple linear system. This stage requires the solution of a linear problem, which is very fast.

The problem of selecting suitable number of basis functions is an important issue for RBFN [33]. The number of basis functions controls the complexity and the generalization ability of RBFN. Too few basis functions give poor predictions on new data due to limited flexibility [34]. It results in a high bias and low variance estimator. Too many basis functions also result in poor generalization as it is too flexible and fits the noise in the training data. It causes a low bias but high variance estimator. The best generalization performance is obtained by a compromise between conflicting requirements of reducing bias as well as variance.

The center of gravity and width is of particular importance for the improvement of the performance of RBFN. There are many approaches along the line with their own pros and cons. This paper proposes the use of differential evolution to find hidden centers and spreads. The motivation using differential evolution over other EAs such as GAs ([35].) is that, in DE string encoding are typically represented as real valued vectors, and the perturbation of solution vectors is based on the scaled difference of two randomly selected individuals of the current population. Unlike GA, the resulting step size and orientation during the perturbation process automatically adopt to the fitness function landscape.

Regarding missing values treatment in RBFNs there are some contributions using the RBFNs to predict the missing values [36] or obtaining the Radial Basis Function from a Vector Quantization of the dataset with missing values [37]. Also, the impact of missing values in the RBFNs has been considered in [38], but only in a case study.