Neurocomputing 128 (2014) 500–506

Contents lists available at ScienceDirect

Neurocomputing

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

Extreme learning machine for classification over uncertain data

Yongjiao Sun*, Ye Yuan, Guoren Wang

Northeastern University, Shenyang 110004, China

ARTICLE INFO

Article history:
Received 17 June 2013
Received in revised form
23 July 2013
Accepted 26 August 2013
Communicated by G.-B. Huang
Available online 18 October 2013

Keywords:
Extreme learning machine
Uncertain data
OS-ELM
SVM
Single hidden layer feedforward neural networks

ABSTRACT

Conventional classification algorithms assume that the input data is exact or precise. Due to various reasons, including imprecise measurement, network delay, outdated sources and sampling errors, data uncertainty is common and widespread in real-world applications, such as sensor database, location database, biometric information systems. Though there exist a lot of approaches for classification, few of them address the problem of classification over uncertain data in database. Therefore, in this paper, we propose classification algorithms based on conventional and optimized ELM to conduct classification over uncertain data. Firstly we view the instances of each uncertain data as the training data for learning. Then, the probabilities of uncertain data in any class are computed according to learning results of each instance. Finally, using a bound-based approach, we implement the final classification. We also extend the proposed algorithms to classification over uncertain data in a distributed environment based on OS-ELM and Monte Carlo theory. The experiments verify the performance of our proposed algorithms.

1. Introduction

Recently, classification over uncertain data has gained much attention, due to the inherent uncertainties of data in many real-world applications, such as sensor network monitoring [1], object identification [2], moving object search [3–5], and the like [6,7,36]. A number of factors induce the uncertainty, including data collection error, measurement, data sampling error, obsolete source, network latency and transmission error. For example, in the moving objects databases, due to the limited resources, it is impossible for the database server to know the exact positions of all objects all the time. In this condition, there are two kinds of uncertainty, measurement error and sampling error. The measurement errors are derived from the imprecision of GPS devices, while in the sampling errors, the uncertainty derives from the update frequency of moving objects. Therefore, it is very important to manage and analyze uncertain data effectively and efficiently.

However, many traditional data classification problems become particularly challenging in the uncertain case, since traditional classification algorithms cannot work for the uncertain data. An uncertain data object may have many instances, and the traditional classification algorithms view each instance as a data object. Thus an uncertain data object can be categorized into many classes, but an uncertain data object only belongs to one class actually. Moreover, an uncertain data object may be attached a probability density function (pdf) that describes the probability of each instance appearing in this uncertain object. The uncertain classification algorithm should consider this uncertain semantics and efficiently process the computation associated with pdf. Obviously, traditional classification algorithms cannot deal with such challenges. Therefore in this paper, based on *extreme learning machine* (ELM) [9–17], we propose a new classification algorithm to process uncertain data objects. Specifically, we use the conventional ELM [10] for uncertain data to obtain binary classifications and the optimized ELM [9] is used for binary and multiclass classifications over uncertain data. We also extend these algorithms to distributed environments based on OS-ELM [8]. Conventional ELM is a good learning method to class data due to good generalization performance as well as improving the learning speed of neural network, maximizing the separating margin, and minimizing the training errors. However, optimized ELM tends to have better scalability and achieve similar (for regression and binary class cases) or much better (for multiclass cases) generalization performance at much faster learning speed than conventional SVM and LS-SVM [18,19]. OS-ELM on the basis of ELM is an algorithm that can handle data arriving or chunk-by-chunk with varying chunk size.

To implement uncertain classifications, we model uncertain data as an object consisting of instances with arbitrary probability distribution. Based on ELM, firstly, we train each instance associated with the uncertain data object. Then, the class probabilities of each instance are computed according to the learning results. Finally, we can obtain the final classification results by using a probability bound-based approach. To obtain more accurate classification results,

* Corresponding author. Tel.: +86 1390 9838 790.
E-mail address: sunyongjiao@ise.neu.edu.cn (Y. Sun).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.08.011

we train the huge and non-huge samples using the optimized ELM. In both cases, we can obtain multiclass results at a time, while SVM needs a lot of iterations. In a distributed environment, based on OS-ELM, we train data one-by-one or chunk-by-chunk with varying or fixed chunk length, so that we can transfer data objects with minimal network overheads in the training and testing phases.

Our contributions in this paper are as follows:

- We develop efficient classification algorithms on uncertain data.
- We use the conventional ELM for uncertain data objects to obtain binary classifications.
- We use the optimized ELM for uncertain data objects to obtain binary and multiclass classifications.
- We also adapt these algorithms to distributed environments based on OS-ELM and a sampling method of Monte Carlo.

The remainder of the paper is organized as follows. The related works are presented in Section 2. We formally define uncertain classification and give a review of ELM in Section 3. We propose uncertain classification algorithms based on conventional and optimized ELMs in Section 5. We adapt the uncertain classification algorithms to distributed scenarios in Section 6. We discuss the results of performance tests on real datasets in Section 7 and the conclusions of our work in Section 8.

2. Related works

There are some works for centralized classification. For centralized massive data, base-level classifiers are generated by applying different learning algorithms with heterogeneous models [23,24], or a single learning algorithm to different versions of the given data. Lin et al. [25] theoretically analyzed the rationale behind plurality voting, and Demrekler et al. [26] investigated how to select an optimal set of classifiers. While [27,28] study the classification of uncertain data using the support vector model, [29] performs classification using decision trees. Artificial neural network has been used in model-based clustering with a probability gained from expectation maximization algorithm for classification-likelihood learning [30].

There are some classification approaches for distributed scenarios. Collaborative [31,32] are the mainly P2P classification approaches. Collaborative generates a single model for the classification, while ensemble combines multiple models (classifiers) for predictions. This is a much more efficient approach which propagates only the statistics of the peers local data, with the decision tree of each peer converging to the global solution over time. Luo et al. [33] proposed building local classifiers using Ivotes [34] and performed prediction using a communication-optimal distributed voting protocol that requires the propagation of unseen data to most.

3. Problem definition and preliminaries

3.1. Problem definition

Uncertainty data model: Consider a set of *uncertain data objects* $U=\{U_1,\ldots,U_n\}$. Each uncertain object U_i consists of a set of *instances* $u_1,\ldots,u_m$. Each instance u_j is associated with a probability p_{u_j} called *appearance probability* with the constraint that $\sum_{j=1}^{m} p_{u_j}=1$. Without loss of generality, we assume that each object is independent of other objects.

Note that each instance in an uncertain object is a multidimensional vector. Thus each instance can be viewed a multidimensional training data for ELM.

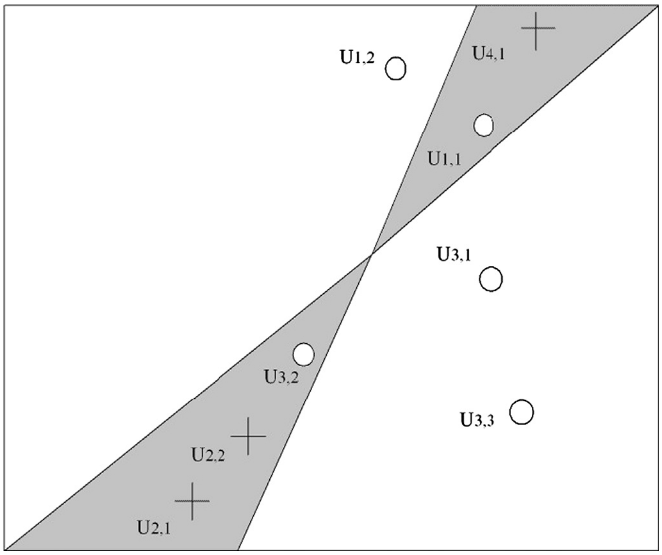

Fig. 1. An example of uncertain data model.

Table 1
Example of uncertain data model.

Uncertain object	# Instances	# Probability
U_1	$u_{1,1}$, $u_{1,2}$	$p_{1,1}=0.2$, $p_{1,2}=0.8$
U_2	$u_{2,1}$, $u_{2,2}$	$p_{2,1}=0.3$, $p_{2,2}=0.7$
U_3	$u_{3,1}$, $u_{3,2}$, $u_{3,3}$	$p_{3,1}=0.1$, $p_{3,2}=0.5$, $p_{3,3}=0.4$
U_4	$u_{4,1}$	$p_{4,1}=1$

For example, Fig. 1 shows uncertain objects U_1, U_2, U_3 and U_4 whose instances and corresponding appearance probabilities are given in Table 1.

In this example, we assume that the classification of uncertain data objects is a simple linear classification. In Fig. 1, the shadow area and the white area represent two classes, and each instance is classified into a corresponding category. However, many instances in the same uncertain data object are classified into different categories. Thus the problem of uncertain classification should consider the probability distributions of its all instances. The uncertain data classification is defined as follows.

Definition 1. Let Ω be the set of instances in all uncertain objects. Given the number of classes m, ELM can classify all instances in Ω into m categories. Then an uncertain object U belongs to a class C_i $(1 \le i \le m)$ if the summarized probability of instances in C_i is the largest.

4. Preliminaries

This subsection briefly gives an overview of ELM.

ELM was originally proposed for the single-hidden-layer feedforward neural networks and then extended to the generalized single-hidden layer feedforward networks where the hidden layer need not be neuron [5,6,37]. In ELM, all the hidden node parameters are randomly generated without tuning. The output function of ELM for generalized SLFNs is

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}) = \boldsymbol{\beta} \cdot \mathbf{h}(\mathbf{x}) \tag{1}$$

where $\boldsymbol{\beta}=[\beta_1,\ldots,\beta_L]^T$ is the vector of the output weights between the hidden layer of L nodes and the output node. $\mathbf{h}(\mathbf{x})=[G(\mathbf{a}_1,b_1,\mathbf{x}),\ldots,G(\mathbf{a}_L,b_L,\mathbf{x})]^T$ is the output (row) vector of