Computing (2019) 101:571–585
https://doi.org/10.1007/s00607-018-00698-w

Improved randomized learning algorithms for imbalanced and noisy educational data classification

Ming Li[1] · Changqin Huang[1,2] · Dianhui Wang[3,4] · Qintai Hu[1,2] · Jia Zhu[2] · Yong Tang[2]

Received: 16 October 2018 / Accepted: 27 December 2018 / Published online: 3 January 2019

Abstract
Despite that neural networks have demonstrated their good potential to be used in constructing learners which exhibit strong predictive performance, there are still some uncertainty issues that can greatly affect the effectiveness of the employed supervised learning algorithms, such as class imbalance and labeling errors (or class noise). Technically, imbalanced data resource can cause more difficulties or limitations for learning algorithms to distinguish different classes, while data with labeling errors can lead to an unreasonable problem formulation due to incorrect hypotheses. Indeed, noise and class imbalance are pervasive problems in the domain of educational data analytics. This study aims at developing improved randomized learning algorithms by investigating a novel type of cost function that focuses on the combined effects of class imbalance and class noise. Instead of concerning these uncertainty issues isolation, we present a convex combination of robust and imbalanced modelling objectives, contributing to a generalized formulation of weighted least squares problems by which the improved randomized learner models can be built. Our experimental study on several educational data classification tasks have verified the advantages of our proposed algorithms, in comparison with some existing methods that either takes no account of class imbalance and labeling errors, or merely consider one specific aspect in problem-solving.

Keywords Imbalanced data classification · Randomized algorithms · Noisy data classification · Educational data analytics

Mathematics Subject Classification 68W20 · 68T01

This work was supported by the National Natural Science Foundation of China (No. 61802132 and 61877020), the China Postdoctoral Science Foundation Grant (No. 2018M630959), and the S&T Projects of Guangdong Province (No. 2016B010109008).

✉ Changqin Huang
cqhuang@scnu.edu.cn

Extended author information available on the last page of the article

1 Introduction

One of the most common tasks in educational data analytics is student learning performance classification/recognition, that is, how to build an effective learner model to predict the students' learning outcomes in terms of certain specific categories/classes. However, some challenges can be encountered in problem-solving (especially the learning algorithm design) due to imbalanced and noisy training data. For example, the task is formulated as a binary classification problem, i.e., 'pass' ('1') or 'fail' ('0'). Students who performed well in regular lectures, activities, practices should be labelled as '1', however, in reality, they failed in the final exam due to some physical/mental troubles, meaning that their practical class labels are assigned as '0'. Apparently, that can be viewed as the problem of labeling errors (or class noise). On the other hand, class imbalance issue sounds much more obvious as there are only few students failed and the most majority of students passed in the exam. Indeed, the minority class, i.e., students who are predicted as 'fail', is typically the class of interest in real practice.

While many studies ave investigated class imbalance and class noise in isolation, very few have addressed their combined effecters, not to mention in the range of educational data analytics. Technically, methods for imbalanced data modelling can be summarized into two general categories: re-sampling techniques in the data level [9–11] and new type of cost function/learning mechanism (e.g. cost sensitive learning) in the algorithm level [23]. Readers can refer to the survey [6] for more details. As for labeling errors (or class noise), two kinds of methodologies are commonly used in problem-solving: (i) the heuristic manner that involves some preprocessing of the training set for data removal or relabelling [2]; (ii) developing models that are robust to the presence of label noise via using robust losses [17,22] to develop label noise-tolerant learning algorithms, preventing instances to take too large weights in neural networks [8], support vector machines [16] and leaner ensembles obtained with bagging [1] and boosting [18]. Interested readers can refer to the review paper [4] for a comprehensive view on this direction.

This paper studies simultaneously the issue of class imbalance and class noise for educational data analytics. Randomized learner models are employed to reduce the computational cost in problem-solving. For the purpose of developing an advanced algorithm that can not only exhibit label noise-tolerant property but also alleviate negative effects of imbalanced data distribution, we present a new type of cost function that consists of two contributing components. In particular, one component aims at enforcing the learner model to become robust to the presence of label noise by considering a weighted least squares (WLS) problem; the role of the other component lies in imbalanced data classification modelling. We apply similar frameworks that has been successfully used in [13] and [26] for robust data analytics (corresponding to the first component), at the same time, present a novel weighting scheme in the second part for relieving the influences caused by imbalanced training samples. These two parts are integrated together by following a convex combination, in which the corresponding coefficients act as the user-specified weighting parameters. Three educational data sets are used in our experimental study and the simulation results demonstrate the advantages of our proposed methods.

The remainder of this paper is organized as follows: Sect. 2 reviews some basics of neural networks with random weights. Section 3 details our proposed methods for building improved RVFL learner models by referring to a new type of objective function. The effectiveness and advantages of our proposed approaches are demonstrated on several real-world cases for educational data analytics, in comparison with some existing randomized learning techniques. Section 5 concludes this work with further remarks.

2 Randomized neural networks

In this section, we review some technical issues around neural networks with random weights. Two types of randomized learner models, that is, random vector functional-link (RVFL) networks and stochastic configuration networks (SCNs), are recalled with highlights on their characteristics and differences.

2.1 Neural networks with data-independent random weights

Since the late 90s, Random vector functional-link (RVFL) networks have received careful attention due to its functionality in fast modelling, that is, the hidden parameters (input weights and biases) are randomly generated and remain fixed while the output weights are optimized according to a linear least square problem [7,19,20]. This trivial idea of performing random weights in neural networks has been used and frequently renamed in the following decades. For a detail history, we recommend the readers a comprehensive survey paper [21]. Although it sounds easier for algorithm implementation once only the output weights need to be tuned in the training process, the universal approximation capability of RVFL networks are in probability sense, which means that a preferable approximation performance is not guaranteed for every random assignment of the hidden parameters.

Basically, the approximation theorem established in [7] only states the existence of reasonable random parameters that can lead to a universal approximator, rather than clarify the particular learning algorithm or the implementation issues of RVFL networks. Indeed, there are some practical issues and common pitfalls in applying this kind of randomized learner model, as theoretically and empirically studied in [14]. Technically, the widely-used strategy for the hidden parameters assignment is to fix the scope $[-\lambda, \lambda]^d$ and $[-\lambda, \lambda]$ $(\lambda > 0)$ for randomly (uniformly) assigning the input weights and biases, respectively. However, it can be very difficult to assign a reasonable λ in practical implementations. Also, from an implementation perspective, the scope setting with a fixed value of λ may lead to a collection of random basis functions with a single scale, which inherently limits the capability of the learner model, compared with a random learner with multi-scale basis functions obtained via using randomization at each scale [5].

To make it concise for algorithm presentation, we only consider a special RVFL architecture without a direct link from the input to the output, as shown in Fig. 1 and mathematically described as

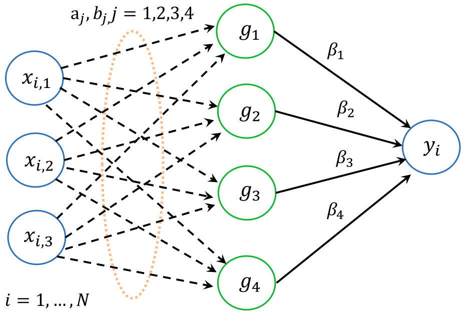


Fig. 1 Schematic depiction of RVFL networks with three input nodes, four hidden nodes, and one output node. Fixed connections are shown as dashed lines, whilst trainable connections are shown as fixed lines

$$G_L(x; a, b) = \sum_{j=1}^{L} \beta_j g(a_j^{\mathrm{T}} x + b_j), \tag{1}$$

where L is the number of hidden nodes, $x \in \mathbb{R}^d$ is the input vector, g is the activation function, $b_j \in \mathbb{R}$ is the bias, $a_j = [a_{j1}, a_{j2}, \ldots, a_{jd}]^{\mathrm{T}} \in \mathbb{R}^d$ is the input weight, $\beta_j \in \mathbb{R}$ is the output weight connecting the j-th hidden node and the output node.

Now, we briefly describe the learning process for RVFL networks. Given a training set $\{x_i, y_i\}$ with N samples of the target function ($i = 1, 2, \ldots, N$), $x_i = [x_{i,1} \ldots, x_{i,2}]^{\mathrm{T}} \in \mathbb{R}^d$, $y_i \in \mathbb{R}$. Remember that a_j and b_j are randomly selected and fixed, therefore, the learning objective is to solve the followed linear least square problem

$$\min_{\beta_1,\ldots,\beta_L} \sum_{i=1}^{N} \left(\sum_{j=1}^{L} \beta_j g(a_j^{\mathrm{T}} x_i + b_j) - y_i \right)^2,$$

which can be converted into a matrix expression, i.e.,

$$\beta^* = \arg \min_{\beta \in \mathbb{R}^L} \|H\beta - Y\|_2^2 \tag{2}$$

where

$$H = \begin{pmatrix} g(a_1^{\mathrm{T}} x_1 + b_1) & \cdots & g(a_L^{\mathrm{T}} x_1 + b_L) \\ \vdots & \ldots & \vdots \\ g(a_1^{\mathrm{T}} x_N + b_1) & \cdots & g(a_L^{\mathrm{T}} x_N + b_L) \end{pmatrix} \tag{3}$$

is the hidden layer output matrix, $Y = [y_1, y_2, \ldots, y_N]^{\mathrm{T}}$, $\beta = [\beta_1, \beta_2, \ldots, \beta_L]^{\mathrm{T}}$.

A closed form solution of the output weights can be obtained by using the pseudo-inverse method [12], i.e., $\beta^* = H^{\dagger} Y$.

2.2 Stochastic configuration networks

Wang and his team developed an advanced randomized learner model, named stochastic configuration networks (SCNs) [25], with both theoretical foundations and algorithm implementations. Distinguished from RVFL networks, SCNs can successfully configure the hidden input weights and biases by meeting a data-dependent supervisory mechanism. Importantly, the universal approximation capability of SCNs can be guaranteed, while its good learning power and sound generalization ability has been empirically verified in a wide range of benchmark datasets and real applications. To the best of our knowledge, SCNs can be viewed as the first work that considers accurately the problem of how to assign reasonable random parameters for the purpose of building a universal approximator, and also a complete framework within the research area (of randomized learner models and techniques) with both theoretical and practical values. So far, Wang's group has successfully extended SCN framework in terms of various viewpoints, such as deep SCNs [27], robust SCNs [13,14], SCN ensembles [24], 2D SCNs for image data analytics [15]. Interested readers can follow their research homepage[1] for more interesting and useful materials. We hereby would like to mention that we only use RVFL networks in this paper and leave the SCN-based proposal for educational data analytics in the future work.

3 Proposed methods

In this section, we develop improved RVFL networks by using a hybrid cost function that considering both the robust and imbalanced data modelling objectives. In particular, we refer to the ideas performed in [13,26] for formulating the component of robust data modelling, aiming at reducing the impacts caused by the noisy data or outliers throughout the training session. As for the part of imbalanced data modelling, we investigate a weighted least squares problem with a novel way to determine the weighted factors that are used to balance/distinguish the associated contribution of training samples. The hybrid cost function is expressed by a convex combination of these two parts (i.e., 'robust' and 'imbalanced'), and some technical tricks such as alternating optimization, half-quadratic technique [13,26] are used again for problem-solving.

For a target function $f : \mathbb{R}^d \to \mathbb{R}^m$, given a training dataset with inputs $X = \{x_1, x_2, \ldots, x_N\}$, $x_i = [x_{i,1}, \ldots, x_{i,d}]^{\mathrm{T}} \in \mathbb{R}^d$ and outputs $Y = \{y_1, y_2, \ldots, y_N\}$, where $y_i = [y_{i,1}, \ldots, y_{i,m}]^{\mathrm{T}} \in \mathbb{R}^m$, $i = 1, \ldots, N$, an improved RVFL approximator of f can be obtained by solving the following hybrid weighted least squares (HWLS) problem, that is,

$$\min_{\beta,\theta} \left(\alpha_1 \sum_{i=1}^{N} \theta_i \| \sum_{j=1}^{L} \beta_j g(a_j, b_j, x_i) - y_i \|^2 + \alpha_2 \sum_{i=1}^{N} w_i \| \sum_{j=1}^{L} \beta_j g(a_j, b_j, x_i) - y_i \|^2 \right), \tag{4}$$

alternatively,

$$\max_{\beta} \left(\alpha_1 \sum_{i=1}^{N} \Psi\big(y_i - \sum_{j=1}^{L} \beta_j g(a_j, b_j, x_i)\big) - \alpha_2 \sum_{i=1}^{N} w_i \| \sum_{j=1}^{L} \beta_j g(a_j, b_j, x_i) - y_i \|^2 \right), \tag{5}$$

[1] http://www.deepscn.com/.

where $\alpha_1 + \alpha_2 = 1$, $0 \leq \alpha_1, \alpha_2 \leq 1$, $G_L(x) = \sum_{j=1}^{L} \beta_j g(a_j, b_j, x)$ is an RVFL network, in which g is the activation function and L is the number of hidden nodes, a_j, b_j are the input weights and biases that are randomly assigned from $[-\lambda, \lambda]^d$ and $[-\lambda, \lambda]$, respectively, and β_j represents the output weights. w_i stands for the weighting coefficients that is used to balance the role of minority/majority class (according to the class imbalance ratio), and we will present its exact expression later. Specifically, in (4), $\theta_i \geq 0$ $(i = 1, 2, \ldots, N)$ is the i-th penalty weight that can represent the contribution of the corresponding sample to the first part of objective function (4). In (5), $\Psi(u) = \exp\left(-\frac{\|u\|^2}{2\sigma^2}\right)$ is a Gaussian kernel function with a fixed scale parameter σ.

Similarly, we can apply half-quadratic (HQ) technique as used in [13] to refine the optimization problem (5) as follows

$$\max_{\beta, \vartheta} \left(\alpha_1 \sum_{i=1}^{N} \Big(\vartheta_i \frac{\|G_L(x_i) - y_i\|^2}{2\sigma^2} - \varphi(\vartheta_i) \Big) - \alpha_2 \sum_{i=1}^{N} w_i \| \sum_{j=1}^{L} \beta_j g(a_j, b_j, x_i) - y_i \|^2 \right), \tag{6}$$

where $G_L(x) = \sum_{j=1}^{L} \beta_j g(a_j, b_j, x)$ is an RVFL network, $\vartheta = (\vartheta_1, \vartheta_2, \ldots, \vartheta_N)$ are the 'auxiliary' variables appearing in the HQ optimization procedure, φ is the convex conjugate function of Ψ. Then, for a fixed β, it can be verified that (5) and (6) are equivalent.

Now we present the way to solve the optimization problems (4) and (6), respectively, i.e., how to adjust θ_i in (4), ϑ_i in (6), evaluate β_j and set w_i for both (4) and (6), as we present the details as follows.

First, we focus on the penalty weights θ_i $(i = 1, 2, \ldots, N)$ in (4). Let us denote the current error matrix as $e := e(X) = [e_1(X), e_2(X), \ldots, e_m(X)]^{\mathrm{T}} \in \mathbb{R}^{N \times m}$, where $e_q(X) = [e_q(x_1), \ldots, e_q(x_N)] \in \mathbb{R}^N$, $q = 1, 2, \ldots, m$, $H \in \mathbb{R}^{N \times L}$ as the hidden layer output matrix. Based on the method used in [26], we can apply kernel density estimation method to obtain the value of θ_i. Specifically, the probability density function of the residual e can be estimated as

$$\Phi(e) = \frac{1}{\tau N} \sum_{k=1}^{N} \mathcal{K}\left(\frac{\|e - e(x_k)\|}{\tau} \right), \tag{7}$$

where $e(x_k) = [e_1(x_k), \ldots, e_m(x_k)]^{\mathrm{T}} \in \mathbb{R}^m$, $\tau = 1.06\hat{\sigma}N^{-1/5}$ is the estimation window width that exhibits a strong influence on the resulting estimate, $\hat{\sigma}$ is the standard deviation of the residual, $\mathcal{K}$ is a Gaussian function defined as $\mathcal{K}(t) = (1/\sqrt{2\pi}) \exp(-t^2/2)$. Therefore, the probability of each residual error $e(x_i)$ $(i = 1, 2, \ldots, N)$ can be obtained by calculating $\Phi(e(x_i))$. Concretely, the penalty weight θ_i can be assigned as:

$$\theta_i = \Phi(e(x_i)) = \frac{1}{\tau N} \sum_{k=1}^{N} \mathcal{K}\left(\frac{\|e(x_i) - e(x_k)\|}{\tau} \right), \quad i = 1, 2, \ldots, N. \tag{8}$$

Suppose that each penalty weight θ_i and weighting factor w_i is assigned and fixed, then the output weights $\beta^* = [\beta_1^*, \beta_2^*, \ldots, \beta_L^*]$ can be evaluated by solving the following linear optimization problem:

$$\beta^* = \arg\min_{\beta} \alpha_1 (H\beta - Y)^{\mathrm{T}} \Theta (H\beta - Y) + \alpha_2 (H\beta - Y)^{\mathrm{T}} W (H\beta - Y) \tag{9}$$

$$= (\alpha_1 H^{\mathrm{T}} \Theta H + \alpha_2 H^{\mathrm{T}} W H)^{\dagger} (\alpha_1 H^{\mathrm{T}} \Theta Y + \alpha_2 H^{\mathrm{T}} W Y), \tag{10}$$

where $\beta = [\beta_1, \beta_2, \ldots, \beta_L], \Theta = \mathrm{diag}\{\theta_1, \theta_2, \ldots, \theta_N\}, W = \mathrm{diag}\{w_1, w_2, \ldots, w_N\}$.

Based on the algorithmic procedures detailed in [26], we can evaluate the penalty weights θ_i and the output weights β iteratively by means of the alternating optimization (AO) strategy, that is,

$$\theta_i^{(\nu+1)} = \frac{1}{\tau N} \sum_{k=1}^{N} \mathcal{K} \left(\frac{e^{(\nu)}(x_i) - e^{(\nu)}(x_k)}{\tau} \right), \tag{11}$$

and

$$\beta^{(\nu+1)} = (\alpha_1 H^{\mathrm{T}} \Theta^{(\nu+1)} H + \alpha_2 H^{\mathrm{T}} W H)^{\dagger} (\alpha_1 H^{\mathrm{T}} \Theta^{(\nu+1)} Y + \alpha_2 H^{\mathrm{T}} W Y), \tag{12}$$

where ν denotes the ν-th iteration of alternating optimization, and $\Theta^{(\nu+1)} = \mathrm{diag}\{\theta_1^{(\nu+1)}, \theta_2^{(\nu+1)}, \ldots, \theta_N^{(\nu+1)}\}$. Here we use $e^{(\nu)}(x_i)$ to represent the residual error value for x_i with $\theta_i^{(\nu)}$ used as the present penalty weights.

As for the optimization of ϑ and β in (6), we can apply the half-quadratic (HQ) technique for problem-solving. Interested readers can refer to [13] (or earlier works listed in their reference) for more technical details. To put it simply, for a fixed $\beta = [\beta_1, \beta_2, \ldots, \beta_L]$, the 'auxiliary' parameters $\vartheta_1, \vartheta_2, \ldots, \vartheta_N$ can be evaluated as

$$\vartheta_i = -\Psi \left(G_L(x_i) - y_i \right). \tag{13}$$

Assume that $\vartheta = (\vartheta_1, \vartheta_2, \ldots, \vartheta_N)$ and the weighting factor w_i are obtained and fixed, the output weights $\beta^* = [\beta_1^*, \beta_2^*, \ldots, \beta_L^*]$ can be evaluated by solving the following linear optimization problem:

$$\beta^* = \arg\min_{\beta} \alpha_1 (H\beta - Y)^{\mathrm{T}} \bar{\Theta} (H\beta - Y) + \alpha_2 (H\beta - Y)^{\mathrm{T}} W (H\beta - Y) \tag{14}$$

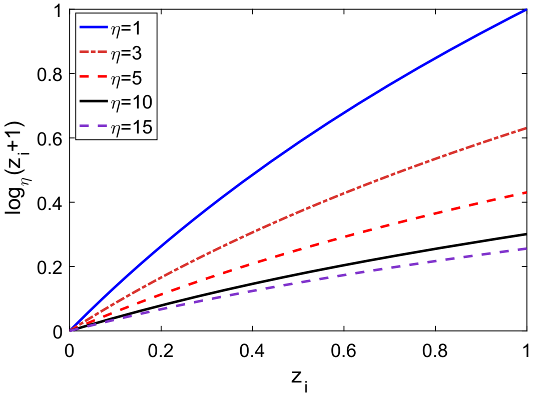
$$= (\alpha_1 H^{\mathrm{T}} \bar{\Theta} H + \alpha_2 H^{\mathrm{T}} W H)^{\dagger} (\alpha_1 H^{\mathrm{T}} \bar{\Theta} Y + \alpha_2 H^{\mathrm{T}} W Y), \tag{15}$$

where $\beta = [\beta_1, \beta_2, \ldots, \beta_L]$, $\bar{\Theta} = \mathrm{diag}\{p_1, p_2, \ldots, p_N\}$, $p_i = -\vartheta_i/(2\sigma^2)$, $i = 1, 2, \ldots, N$. $W = \mathrm{diag}\{w_1, w_2, \ldots, w_N\}$

Now we can formulate the iterative solution for ϑ and β by following the AO strategy, i.e.,

$$\vartheta_i^{(\nu+1)} = -\exp\left(-\frac{\| y_i - \sum_{j=1}^{L} \beta_j^{(\nu)} g(a_j, b_j, x_i) \|^2}{2\sigma^2} \right), \tag{16}$$

Fig. 2 Demonstration of the changing trend of S_i value



and

$$\beta^{(\nu+1)} = (\alpha_1 H^{\mathrm{T}} \bar{\Theta}^{(\nu+1)} H + \alpha_2 H^{\mathrm{T}} W H)^{\dagger} (\alpha_1 H^{\mathrm{T}} \bar{\Theta}^{(\nu+1)} Y + \alpha_2 H^{\mathrm{T}} W Y), \quad (17)$$

where ν denotes the ν-th iteration of alternating optimization, and $\bar{\Theta}^{(\nu+1)} = \mathrm{diag}\{p_1^{(\nu+1)}, p_2^{(\nu+1)}, \ldots, p_N^{(\nu+1)}\}$, $p_i^{(\nu+1)} = -\vartheta_i^{(\nu+1)}/(2\sigma^2)$, $i = 1, 2, \ldots, N$.

Now we can determine the weighting factors w_i performed in the second part. It is logical to think that the value of w_i should closely relate to the degree of re-balance expected in the training process. Technically, the way to assign these weights should to some extent follow the imbalance ratio of the given training samples. In this work, two options are presented as follows:

$$\text{Option 1}: w_i = 1/\#\{y_i\}, i = 1, 2, \ldots, N \quad (18)$$

$$\begin{aligned} \text{Option 2}: w_i &= (\log_\eta(z_i + 1))/\#\{y_i\}, \\ z_i &= \#\{y_i\}/\max\{\#\{y_i\}\}, \eta > 1, i = 1, 2, \ldots, N, \end{aligned} \quad (19)$$

where $\#\{y_i\}$ denotes the number of samples that belong to class label y_i.

Basically, Option 1 is a trivial way in which the weight sums of each class, i.e., $S_i = w_i\#\{y_i\}$, are equal to 1. However, in Option 2, the weight sum value is evaluated by $S_i = (\log_\eta(z_i + 1))$, which means that its value is positively correlated with the number of samples in each class. Specifically, the value of S_i decays with the decrease in the sample number $\#\{y_i\}$, such as shown in Fig. 2.

So far, we can summarize the above algorithmic procedures as the following pseudo-codes.

4 Educational data analytics

In this section, we compare our proposed methods with some existing randomized learning techniques on three real-world case studies for student learning performance recognition. Our experimental results demonstrate the feasibility and effectiveness of

Algorithm 1: Improved RVFL for Imbalanced and Noisy Data Modelling

```
Input  : Training inputs $X = \{x_1, x_2, \ldots, x_N\}$, $x_i \in R^d$, outputs $Y = \{y_1, y_2, \ldots, y_N\}$, $y_i \in R^m$;
         The maximum number of alternating optimization $AO_{max}$; The number of hidden nodes $L$.
         The scope parameter $\lambda > 0$. Contributing factors $\alpha_1, \alpha_2 \in [0, 1]$. Sigmoid activation
         function $g = 1/(1 + \exp(-x))$.
Output: An RVFL Model
Randomly assign the input weights $a_j$ and biases $b_j$ from $[-\lambda, \lambda]^d$ and $[-\lambda, \lambda]$;
Calculate the hidden layer output matrix $H$ by Eq. (3);
Calculate the initialization error matrix by $e = HH^{\dagger}Y - Y$;
Set the weighting factor $w_i$ according to Eq. (18)                              // Option 1
:(Alternatively set the weighting factor $w_i$ according to Eq. (19))             // Option 2
while $\nu \leq AO_{max}$ do
    Update $\theta_i^{(\nu)}$ by Eq. (11), $\beta^{(\nu)}$ by Eq. (12)                        // Method 1
    :(Alternatively update $\vartheta_i^{(\nu)}$ by Eq. (16), $\beta^{(\nu)}$ by Eq. (17) )   // Method 2
    Renew $\nu := \nu + 1$;
end
Return: The output weights $\beta^*$
```

our proposed methods on educational data analytics, in which considerable uncertainty issues always exist. First, we detail these three datasets, followed by elaborating mathematically the used evaluation metrics. Later, some existing methods to be compared in the experiments are listed with highlights of their key features. Then, some simulation results are presented with discussion and explanation. Finally, further robustness analysis is provided to investigate the impacts of the scope setting (in RVFL-based leaner models) on the performance.

4.1 Datasets

We use three different educational datasets in the experiments.

Case 1 This database contains two subjects: Mathematics (mat) and Portuguese language (por) that are conducted in two secondary schools in Portuguese. The original database contains 33 attributes, for examples, student grades, demographic, social and school related items, etc., obtained by using school mark reports and well-designed questionnaires [3]. In our experiments, we take the attribute G3 ('final grade') as the output variable while the left features apart from 'Medu' (stands for mother's education) and 'Fedu' (stands for father's education) as input variables. Also, for the purpose of classification problem formulation, we convert the feature G3 to binary type, that is, label 0 for G3 $\geq$ 10, while label 1 for G3 $<$ 10. As for the page limit, we cannot present the complete summary of this dataset, however, interested readers can refer to our recent work [13] for more details. In the experiments, we use the 649 samples of course Portuguese language (por) for training while the instances of Mathematics (mat) for testing. Then, the input values for training and testing samples are normalized into [0,1].

It is apparent that this dataset exhibits the issue of imbalanced data distribution as there are considerable number of students who have passed the exam or received a pass score in the course (viewed as the majority class) while only few students who could not pass the exam or get a low score (viewed as the minority). To fit our problem formulation and distinguish our proposed methods in problem-solving, we artificially add some labelling errors (or class noise) in the training set by randomly select 10% samples and change their labels accordingly. We should note that the test samples are remain outlier-free.

Case 2 This data set contains a total 5820 evaluation scores provided by students from Gazi University in Ankara (Turkey).[2] There are 32 input features (28 course specific questions and additional 4 attributes) and one output feature (the number of times the student is taking this course). We randomly selected 75% of the samples for training and the left 25% for testing. Similar as Case 1, the input values for training and testing samples are normalized into [0,1]. Specifically, our focus is binary classification problem, therefore for the output feature, we view the group of samples with repeat number equal to 1 as the majority class, while the others together as the minority class. Indeed, there are far more students who only repeat this course one time than the students who repeat two times or more, which can be considered as imbalanced data classification issue. On the other hand, a similar way as performed in Case 1 is used to add some class noise in the training set.

Case 3 This database has 7543 records collected from the World University City Platform[3] that is widely used for educational data analytics. In our experiments, we consider nine input features, including student (online learning user) gender, engagement in learning space, the liveness of online learning, the relevance of learning resources and contents, the emotion in learning activities, the duration of learning, the score of after-class assignments and tests, the score of progress test average, the learning target matching rate, and one output feature, that is, student final performance (pass with final score ≥ 50 or fail with final score < 50). Some basic information for this database is presented in Table 1. In our experiments, we use the similar manner to normalize the inputs and also put class noise for the problem-formulation.

4.2 Algorithm setting

We compare our proposed two methods with five randomized learning techniques, including original RVFL, improved RVFL with KDE (termed Imp-RVFL-KDE), improved RVFL with MCC (termed Imp-RVFL-MCC), imbalanced RVFL with Option 1 (named Imb-RVFL-1), imbalanced RVFL with Option 2 (named Imb-RVFL-2). All the reported results are averaged based on 50 independent trials. Indeed, as aforementioned in the previous section, our proposal already covers these methods by simply specifying the setting of α_1 and α_2. In particular, when $\alpha_1 = 1, \alpha_2 = 0$, Imp-RVFL-KDE and Imp-RVFL-MCC can be implemented by setting the AO number as 1 in our propose method 1 and 2, respectively; when $\alpha_1 = 0, \alpha_2 = 1$, Imb-RVFL-KDE and Imb-RVFL-MCC can be implemented by using Option 1 and Option 2 respectively. To show the advantages of our proposed methods in addressing noisy and imbalanced

[2] http://archive.ics.uci.edu/ml/datasets/turkiye+student+evaluation.

[3] http://www.worlduc.com/.

Table 1 Basic information for the database used in Case 3

Attributes	Description	Value type/range
I_1	Gender: 0-male, 1-female	Discrete/{0, 1}
I_2	Engagement in learning space	Discrete/{0, 1, . . . , 14}
I_3	Learning liveness	Continuous/[5,68]
I_4	Learning content relevance	Continuous/[8,86]
I_5	Learning emotion	Continuous/[0,12]
I_6	Learning duration	Continuous/[273,779]
I_7	After-class assignment score	Continuous/[35,39]
I_8	Progress test average score	Continuous/[60,99]
I_9	Learning target matching rate	Continuous/[1,22]
O_1	Student performance level	Binary/0 for pass and 1 for fail

Table 2 Basic information for algorithms performed in performance comparison

Abbreviations	Description
Method1-Opt1	Our Proposed Method 1 by Option 1, $\alpha_1 = \alpha_2 = 0.5$, see Algorithm 1
Method1-Opt2	Our Proposed Method 1 by Option 2, $\alpha_1 = \alpha_2 = 0.5$, see Algorithm 1
Method2-Opt1	Our Proposed Method 2 by Option 1, $\alpha_1 = \alpha_2 = 0.5$, see Algorithm 1
Method2-Opt2	Our Proposed Method 2 by Option 2, $\alpha_1 = \alpha_2 = 0.5$, see Algorithm 1
Original-RVFL	Original RVFL performed in Sect. 2
Imb-RVFL-KDE	Robust RVFL with KDE, same as set $\alpha_1 = 1, \alpha_2 = 0$, AO = 0
Imb-RVFL-MCC	Robust RVFL with MCC, same as set $\alpha_1 = 1, \alpha_2 = 0$, AO = 0
Imb-RVFL-Opt1	Imbalanced RVFL by Option 1, same as set $\alpha_1 = 0, \alpha_2 = 1$, AO = 0
Imb-RVFL-Opt2	Imbalanced RVFL by Option 2, same as set $\alpha_1 = 0, \alpha_2 = 1$, AO = 0

data classification problem, we select $\alpha_1 = \alpha_2 = 0.5$ in both Method 1 and Method 2. Overall, the complete algorithm list compared in the simulations is summarized in Table 2, in which some abbreviations are used as the index of algorithm to help in the following performance comparison.

4.3 Results and discussion

We use G-mean as the evaluation metric for these binary classification problems, as expressed by

$$\text{G-mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}},$$

where TP, FN, FP, TN represents the number of 'True Positive', 'False Negative', 'False Positive', 'True Negative' records in the confusion matrix, respectively.

Table 3 demonstrates the simulation results for performance comparison, in which the records for all these methods are the best ones after implementing different setting of λ (e.g., λ ∈ {0.1, 0.3, 0.5, 1, 3, 5, 10, 50}). Our aim lies in that the normally used

Table 3 Performance comparisons on three case studies

Algorithms	Test performance (G-mean)		
	Case 1	Case 2	Case 3
Method1-Opt1	**0.7495**	0.7256	0.8445
Method1-Opt2	0.7486	**0.7276**	**0.8478**
Method2-Opt1	0.7488	0.7267	0.8469
Method2-Opt2	0.7478	0.7259	0.8453
Original-RVFL	0.7113	0.6887	0.8015
Imp-RVFL-KDE	0.7242	0.7068	0.8269
Imp-RVFL-MCC	0.7257	0.7072	0.8196
Imb-RVFL-Opt1	0.7198	0.7158	0.8102
Imb-RVFL-Opt2	0.7217	0.7123	0.8123

Bold values indicate the best results obtained for each case study

Table 4 Robustness analysis for scope parameter setting on Case 1

Algorithms	Test performance (G-mean)					
	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$
Method1-Opt1	0.7456	0.7482	**0.7495**	0.7486	0.7213	0.7011
Method1-Opt2	0.7413	**0.7486**	0.7475	0.7310	0.7011	0.6812
Method2-Opt1	0.7479	**0.7488**	0.7481	0.7426	0.7253	0.7191
Method2-Opt2	**0.7478**	0.7472	0.7395	0.7068	0.6879	0.6852

Bold values indicate the best results obtained for each case study

scope setting $\lambda = 1$ is not always the reasonable choice. Further investigation can be found in later robustness analysis. It is clear from Table 3 that our proposed methods (corresponding to the first four rows) outperform the other randomized approaches, which has verified the effectiveness of our proposed hybrid cost function and the advantages of the associated solutions.

To demonstrate the impacts of λ on the randomized learner models' performance, we present the test performance of our methods in terms of different setting of λ. We here only list the records for Case 1 and similar results can be observed in both Case 2 and 3 based on our experience. As can be found in Table 4, $\lambda = 0.1, 0.3, 0.5$ can lead to slightly better results than $\lambda = 1$, however, highly better results than the other settings, implying that λ in RVFL-based learners should be data-dependent and of great importance in problem-solving. More empirical study, especially the superiority of SCNs [25] on addressing this issue, is expected (Fig. 3).

5 Conclusion

This paper presents a framework for building randomized learner models with training samples exhibiting class imbalance and labeling errors. A hybrid type of cost function, consisting two parts focusing on robust and imbalanced data modelling respectively, is

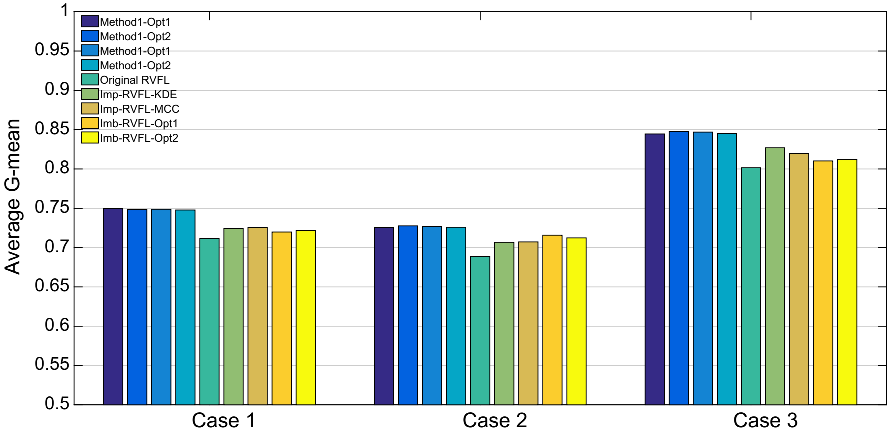


Fig. 3 Performance comparison for three case studies

proposed as the training objective. RVFL networks are used as the randomized leaner models in problem formulation. Basically, our method can be viewed as an extension of robust RVFL that merely considers noisy training samples in training process, a generalization of imbalanced RVFL that only concerns imbalanced data modelling. Our experimental study on three educational data classification tasks have verified the advantages of our proposed algorithms in addressing simultaneously imbalanced and noisy data modelling.

This work opens future efforts on using more and different classifiers to check the feasibility and effectiveness of the proposal on hybrid cost function. In particular, following work can be performed in the direction of employing SCN framework [25] to further improve the randomized learner model's performance, with their advantages over RVFL networks in both learning and generalization. Besides, a further investigation of the way to determine the weighting factors in imbalanced data modelling is interesting and useful. Finally, additional and more in-depth research into the robustness analysis of some key parameters in our framework can be undertaken, with improvements recommended to enhance the performance of our proposed approaches.

References

1. Abellán J, Masegosa AR (2010) Bagging decision trees on data sets with classification noise. In: International symposium on foundations of information and knowledge systems, Springer, pp 248–265
2. Brodley CE, Friedl MA (1999) Identifying mislabeled training data. J Artif Intell Res 11:131–167
3. Cortez P, Silva AMG (2008) Using data mining to predict secondary school student performance. In: Proceedings of the 5th future business technology conference, pp 5–12
4. Frénay B, Verleysen M (2014) Classification in the presence of label noise: a survey. IEEE Trans Neural Netw Learn Syst 25(5):845–869
5. Gorban AN, Tyukin IY, Prokhorov DV, Sofeikov KI (2016) Approximation with random bases: Pro et contra. Inf Sci 364:129–145
6. He H, Garcia EA (2008) Learning from imbalanced data. IEEE Trans Knowl Data Eng 9:1263–1284

7. Igelnik B, Pao YH (1995) Stochastic choice of basis functions in adaptive function approximation and the functional-link net. IEEE Trans Neural Netw 6(6):1320–1329
8. Khardon R, Wachman G (2007) Noise tolerant variants of the perceptron algorithm. J Mach Learn Res 8(Feb):227–248
9. Khoshgoftaar TM, Van Hulse J, Napolitano A (2010) Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors. IEEE Trans Neural Netw 21(5):813–830
10. Khoshgoftaar TM, Van Hulse J, Napolitano A (2011) Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Trans Syst Man Cybern A Syst Hum 41(3):552–568
11. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. Prog Artif Intell 5(4):221–232
12. Lancaster P, Tismenetsky M (1985) The theory of matrices: with applications, 2nd edn. Academic Press, San Diego
13. Li M, Huang C, Wang D (2019) Robust stochastic configuration networks with maximum correntropy criterion for uncertain data regression. Inf Sci 473:73–86
14. Li M, Wang D (2016) Insights into randomized algorithms for neural networks: Practical issues and common pitfalls. Inf Sci 382:170–178
15. Li M, Wang D (2018) Two dimensional stochastic configuration networks for image data analytics. arXiv:1809.02066
16. Lin CF, Wang SD (2004) Training algorithms for fuzzy support vector machines with noisy data. Pattern Recognit Lett 25(14):1647–1656
17. Masnadi-Shirazi H, Vasconcelos N (2009) On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In: Advances in neural information processing systems, pp 1049–1056
18. Oza NC (2004) Aveboost2: boosting for noisy data. In: International workshop on multiple classifier systems, Springer, pp 31–40
19. Pao YH, Park GH, Sobajic DJ (1994) Learning and generalization characteristics of the random vector functional-link net. Neurocomputing 6(2):163–180
20. Pao YH, Takefuji Y (1992) Functional-link net computing: theory, system architecture, and functionalities. Computer 25(5):76–79
21. Scardapane S, Wang D (2017) Randomness in neural networks: an overview. WIREs Data Min Knowl Discov 7(2):e1200. https://doi.org/10.1002/widm.1200
22. Stempfel G, Ralaivola L (2009) Learning SVMs from sloppily labeled data. In: International conference on artificial neural networks, Springer, pp 884–893
23. Sun Y, Kamel MS, Wong AK, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. Pattern Recognit 40(12):3358–3378
24. Wang D, Cui C (2017) Stochastic configuration networks ensemble with heterogeneous features for large-scale data analytics. Inf Sci 417:55–71
25. Wang D, Li M (2017) Stochastic configuration networks: fundamentals and algorithms. IEEE Trans Cybern q 47(10):3466–3479
26. Wang D, Li M (2017) Robust stochastic configuration networks with kernel density estimation for uncertain data regression. Inf Sci 412:210–222
27. Wang D, Li M (2018) Deep stochastic configuration networks with universal approximation property. In: Proceedings of international joint conference on neural networks, IEEE, pp 1–8

Affiliations

Ming Li[1] · Changqin Huang[1,2] · Dianhui Wang[3,4] · Qintai Hu[1,2] · Jia Zhu[2] · Yong Tang[2]

[1] School of Information Technology in Education, South China Normal University, Guangzhou, China

[2] Guangdong Engineering Research Center for Smart Learning, South China Normal University, Guangzhou, China

[3] Department of Computer Science and Information Technology, La Trobe University, Melbourne, Australia

[4] The State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China