Weight learning from cost matrix in weighted least squares model based on genetic algorithm

Hong Zhu, Peng Yao and Xizhao Wang*

College of Computer Science and Software Engineering,
Guangdong Key Lab of Intelligent Information Processing,
Shenzhen University,
Shenzhen, 518060, China
Email: xszhuhong@163.com
Email: 834369657@qq.com
Email: xizhaowang@ieee.org
*Corresponding author

Abstract: In real life, it is a common phenomenon that different misclassification causes different cost. Given a misclassification cost matrix (MCM), cost-sensitive learning is aiming at decreasing the overall misclassification cost rather than simply reducing the misclassification rate. Weighted least squares (WLS) model is acknowledged as an effective way of cost sensitive learning. However, the weights in WLS model are generally unknown and finding these weights is usually difficult. In this paper, we put forward a new approach to learning these weights of WLS model from a given MCM based on a genetic algorithm. A comparative study shows that our proposed approach has an overall cost of misclassification significantly smaller than the existing cost-sensitive learning methods.

Keywords: cost-sensitive learning; misclassification cost matrix; MCM; weighted least squares model; genetic algorithm.

Reference to this paper should be made as follows: Zhu, H., Yao, P. and Wang, X. (2019) 'Weight learning from cost matrix in weighted least squares model based on genetic algorithm', *Int. J. Bio-Inspired Computation*, Vol. 13, No. 4, pp.269–276.

Biographical notes: Hong Zhu received her BSc and MSc degrees from the Hebei University, Baoding, China in 2012 and 2015, respectively and PhD degree from the Macao University of Science and Technology in 2018. She is currently a Post Doctor in the Shenzhen University. Her main research interests include decision tree and neural networks, ensemble learning and cost-sensitive learning.

Peng Yao received his BSc degree from the Dongguan University of Technology in 2015 and MSc degree from the Shenzhen University in 2018. His main research interests include neural networks.

Xizhao Wang served in Hebei University as a Professor and the Dean of School of Mathematics and Computer Sciences before 2014. After 2014, he worked as a Professor in Big Data Institute of ShenZhen University. His major research interests include uncertainty modelling and machine learning for big data. He has edited 10+ special issues and published three monographs, two textbooks, and 200+ peer-reviewed research papers. As a Principle Investigator (PI) or co-PI, he has completed 30+ research projects. He has supervised more than 150 MPhil and PhD students.

1 Introduction

Classification is a basic task in machine learning and data mining. It exists in many fields of our real life, such as internet search (Sun et al., 2014), image processing (Russ and Neal, 2017) and handwriting recognition (Dastidar et al., 2015). Its working process contains two stages: firstly, train a classifier based on training samples whose class labels are marked; secondly, use the trained classifier to classify unseen samples. The commonly used methods to generate classifiers include support vector machine (Liu et al., 2018; Guo et al., 2018) decision tree (Quinlan, 1986), naive Bayes (Rish, 2001), K-nearest neighbor (Peterson, 2009), neural network (Hagan et al., 2002) and so on.

In general classification methods, there is an assumption that for a certain classification problem the costs of all the misclassifications are the same. The goal of these methods is to minimise the misclassification rate. While in real life, it is a common phenomenon that different misclassification leads to different cost. As an example, in medical diagnosis, in comparison with mistaking a healthy person as a patient, mistaking a patient as a healthy person will lead to more serious consequences even life-threatening. Similarly, in the credit card theft detection, missing misappropriation as

normal use will result in greater economic loss than discriminating normal use as misappropriation.

In response to the above problems, cost-sensitive learning assigns different costs to different misclassifications. It aims to minimise the overall misclassification cost, rather than simply minimise the misclassification rate. As an emerging classification strategy, cost-sensitive learning has been deeply studied and explored.

Kukar and Kononenko (1998) put forward a new backward propagation algorithm for neural networks, which can meet the requirements of cost-sensitive learning. Domingos (1999) proposed the MetaCost method which is a way to convert a general classification model into a cost-sensitive model. Drummond and Holte (2000) studied cost-sensitive learning decision trees and proposed a node splitting method. Bradford et al. (1998) studied how to prune decision trees under cost-sensitive conditions and made a conclusion that the pruning approach based on the Laplace method can achieve the best results. Wang et al. (2013) put forward a cost-sensitive Boosting algorithm named Ada-Cost. Geibel et al. (2014) proposed cost-sensitive learning methods based on perceptron and support vector machines. It modifies the class marks of training samples through a 'meta learning' process and relearn a new model by using the modified training set. Based on the idea of cost-proportionate, Zadrozny et al. (2003) adjusted the weights of training data, which is similar to the Boosting algorithm in practical applications. It can be implemented by subsampling or adjusting the weights of classifiers. Abe et al. (2004) explored how to implement cost-sensitive learning in multi-class classification problems and proposed a new iterative learning method. In Zhai et al. (2017) and Mao et al. (2017), cost-sensitive learning methods based on extreme learning machine (ELM) were proposed.

However, these cost-sensitive classifiers cannot guarantee the obtained final overall cost is the minimum. In order to overcome this defect, in this paper we proposed a weighted least squares (WLS) model whose weights are learned from the misclassification cost matrix (MCM) by using the genetic algorithm. WLS is acknowledged as an effective way of cost sensitive learning. It is based on the least squares model which is the essence of ELM. In this paper, we construct a weighted ELM as an specific form of the WLS method.

ELM (Huang et al., 2006) is a single hidden layer feed-forward neural network, which performs nonlinear transformation on the original data firstly and then optimises the parameters through the least squared method. ELM has attracted extensive attention from scholars since it was proposed. To expand the application scope of ELM, many extensions have been proposed, such as the domain space transfer ELM for domain adaptation (Chen et al., 2018), the online kernelised and regularised ELM for wearable-based activity recognition (Hu et al., 2018). In Alshamiri et al. (2018) proposed two swarm intelligence approaches to improve the generalisation performance of ELM. In Zhao et al. (2018) studied the impact of the rank of input data matrix on the performance of ELM. In Luo et al. (2018) put forward the timeliness online regularised ELM. In Ding et al. (2017) proposed an unsupervised ELM with representational features. In Liu et al. (2017) introduced a semi-supervised low rank kernel learning algorithm via ELM.

The main content of our work can be summarised as:

1 We proposed a weight learning method for constructing a cost-sensitive classifier model of ELM. In the model, the loss function is a weighted sum of squared errors and the weights are obtained by minimising the overall misclassification cost through genetic algorithm.

2 We completed the transition from the MCM to the weights in the weighted least square model. Usually, the weights are meaningless. In this paper, the weights are generated based on the MCM, which makes the weights interpretable.

3 We conducted experiments to illustrate the effectiveness of our proposed model. Experimental results showed that in comparison with cost-sensitive ELM and cost-sensitive naive Bayes, our model can achieve the minimum overall cost, which means that our model has much better classification performance when handling cost-sensitive learning problems.

This paper is organised as follows. Section 1 is the introduction. In Section 2 and Section 3, we separately reviewed the basic knowledge of cost-sensitive learning and the WLS model. In Section 4, we described the construction process of the WLS model of ELM based on genetic algorithm in detail. In Section 5, by conducting comparative experiments, we demonstrated the effectiveness of the proposed model. Section 6 is about the conclusion and the future work.

2 Cost-sensitive learning

Many scholars have proposed various cost-sensitive learning methods. According to the learning mechanisms of cost-sensitive learning methods, we can classify them into three categories.

1 Some approaches focus on how to directly construct a cost-sensitive learning model based on traditional classifiers, such as decision trees (Drummond and Holte, 2000; Bradford et al., 1998), neural networks (Kukar and Kononenko, 1998; Zhai et al., 2017; Mao et al., 2017) and the Boosting algorithm (Wang et al., 2013).

2 Some other methods post-process the classification results (Bradford et al., 1998; Zadrozny et al., 2003).

3 The other methods train cost-sensitive models by changing the distribution of the raw training data (Geibel et al., 2004).

In cost-sensitive learning, the misclassification costs are shown by the cost matrix (García-López et al., 2015) which is expressed as follows:

$$C=\begin{bmatrix} 0 & \cdots & c_{1m} \\ \vdots & \vdots & \vdots \\ c_{m1} & \cdots & 0 \end{bmatrix}_{m\times m} \tag{2.1}$$

where m is the number of categories to which the training samples belong; c_{ij} represents the cost of misclassifying a sample of the i^{th} class into the j^{th} class ($0 \le i, j \le m$). Usually, for a certain problem the cost matrix is given by experts with expertise and rich experience.

Following is a detailed description of the cost-sensitive ELM and the cost-sensitive naive Bayes.

Cost-sensitive ELM is the product of the combination of cost-sensitive learning and ELM. It aims to minimise the cumulative error. When dealing with multi-classification problem with m categories, the model of ELM can be summarised as follows:

Minimise:

$$L=\frac{1}{2}\|\beta\|^2+C\frac{1}{2}\sum_{j=1}^{N}\left\|e_j w_j\right\|^2 \tag{2.2}$$

Subject to:

$$h(x_j)\beta=t_j^T-e_j^T,\ j=1,2,\cdots,N \tag{2.3}$$

According to the KKT theorem, the equivalent dual optimisation problem of formulas (2.2) and (2.3) is:

$$\begin{aligned} L=&\frac{1}{2}\|\beta\|^2+C\frac{1}{2}\sum_{j=1}^{N}\left\|e_j w_j\right\|^2 \\ &-\sum_{j=1}^{N}\gamma_j\left(h(x_j)\beta-t_j+e_j\right) \end{aligned} \tag{2.4}$$

where the Lagrange multiplier γ_j is a constant factor of sample x_j. Let the partial derivatives of function (2.4) with respect to each variable β, e_j, γ_j be 0, then we can obtain generated decision function:

$$f(x)=h(x)(H^{+}T)W \tag{2.5}$$

where H is the hidden layer output matrix, which can be expressed as:

$$\begin{aligned} &H(\alpha_1,\alpha_2,\cdots,\alpha_N,b_1,b_2,\cdots,b_N,x_1,x_2,\cdots,x_N) \\ &=\begin{bmatrix} g(\alpha_1\cdot x_1+b_1) & \cdots & g(\alpha_N\cdot x_1+b_N) \\ \vdots & \vdots & \vdots \\ g(\alpha_1\cdot x_N+b_1) & \cdots & g(\alpha_N\cdot x_N+b_N) \end{bmatrix}_{N\times N} \end{aligned} \tag{2.6}$$

H^{+} is the Moore-Penrose generalised inverse of matrix H; $h(x_j)=[g(\alpha_1\cdot x_j+b_1),\cdots,g(\alpha_{\tilde{N}}\cdot x_1+b_{\tilde{N}})]$ is the hidden layer output matrix of the j^{th} sample; e_j is the j^{th} column of the error matrix e.

$x_j=[x_{j1},x_{j2},\cdots,x_{jn}]^T\in R^n$ is the feature vector of the j^{th} sample; $t_j=[t_{j1},t_{j2},\cdots,t_{jm}]^T\in R^m$ is the target value vector of the j^{th} sample; $g(x)$ is the activation function; $\alpha_j=[\alpha_{j1},\alpha_{j2},\cdots,\alpha_{jn}]^T$ is the input weight vector connecting the j^{th} hidden node with the input layer; $\beta_j=[\beta_{j1},\beta_{j2},\cdots,\beta_{jm}]^T$ is the output weight vector connecting the j^{th} hidden node with the output layer; b_j is the bias in the hidden layer; $o_i=[o_{i1},o_{i2},\cdots,o_{im}]^T$ is the output vector; N is the number of training samples; m is the number of categories to which the training samples belong; $\tilde{N}$ is the number of nodes in the hidden layer.

Cost-sensitive naive Bayes is an improved naive Bayes, which can process cost-sensitive problems. Its cost function can be shown as:

$$F(c_i,c_j)=\begin{cases} \left(\dfrac{p_i}{p_j}\right)^u, & p_i>p_j; \\ \left(\dfrac{p_j}{p_i}\right)^v, & p_i<p_j; \\ 1, & p_i=p_j; \\ 0, & i=j. \end{cases} \tag{2.7}$$

where $F(c_i, c_j)$ is the cost caused by misclassifying c_i to c_j; p_i is the proportion of the samples belonging to c_i in all the samples; p_j corresponds to c_j.

The risk function of cost-sensitive naive Bayes is:

$$R(c_i\mid x)=\sum P(c_j\mid x)\times F(c_j,c_i) \tag{2.8}$$

where $P(c_j\mid x)$ can be obtained according to formula (2.9) to formula (2.11):

$$\begin{aligned} &R(c_i\mid x)=\frac{P(x\mid c_i)P(c_i)}{P(x)}=\frac{P(x\mid c_i)P(c_i)}{\sum_{i=1}^{l}P(x\mid c_i)P(c_i)}, \\ &i=1,2,\cdots,l. \end{aligned} \tag{2.9}$$

When the loss function is the 0~1function, we assign the sample x to category c in order to minimise the classification error.

$$c=\arg\max_{1\le i\le l}\left\{P(c_i\mid x)\right\} \tag{2.10}$$

Because for any $i\in\{1,2,\cdots,l\}$, $P(x)=\sum_{i=1}^{l}P(x\mid c_i)P(c_i)$ is a constant. Then based on formula (2.9) and formula (2.10), we can obtain:

$$c=\arg\max_{1\le i\le l}\left\{P(x\mid c_i)P(c_i)\right\} \tag{2.11}$$

Finally, we consider the category with the minimum risk as the class the sample x belonging to.

3 WLS method

WLS method is an extension of the least squares method which is a mathematical optimisation technique. Through minimising the sum of squared errors between output values and target values, the least squares method can approximate the function between the independent variables and the dependent variable.

The least squares method can be mathematically described as follows:

For a given set consisting of N data points $\{(X_i, y_i)\}$ ($i = 1, 2, \cdots, N$), the least squares method aims to find out a function $p(X)$ so that the sum of squared errors $E = \sum_{i=1}^{N} \left(p\left(X_i - y_i\right)\right)^2$ attains its minimum value.

Now we illustrate the process of the least squares method by approximating a linear function. Suppose the approximation function can be expressed as:

$$\sum_{j=1}^{n} x_{ij}\beta = o_j, \left(i = 1, 2, \cdots, N\right) \tag{3.1}$$

where n is the number of independent variables and $N > n$; β_j is the unknown coefficient.

By vectorising equation (3.1) we can obtain equation (3.2).

$$X\beta = o \tag{3.2}$$

where

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nm} \end{bmatrix}, \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, o = \begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_N \end{bmatrix}$$

To find a specific $\hat{\beta}$ such that $\| X\hat{\beta} - y \| = \min_{\beta} \| X\beta - y \|$ is equivalent to minimising the loss function

$$E = \sum_{i=1}^{N} \left(X_i\beta - y_i \right)^2 = \| X\beta - y \|^2 \tag{3.3}$$

where $y = [y_1, y_2, \cdots, y_N]^T$.

when $\beta = \hat{\beta}$, E attains its minimum value, which can be marked as:

$$\hat{\beta} = \arg\min(E) \tag{3.4}$$

By conducting the differentiation operation, we can obtain the following expression:

$$X^T X\hat{\beta} = X^T y \tag{3.5}$$

If X^TX is a nonsingular matrix, then $\hat{\beta}$ is a unique solution and can be expressed as:

$$\hat{\beta} = \left(X^T X \right)^{-1} X^T y \tag{3.6}$$

We call $\hat{\beta}$ is a least squares solution of the linear system $X\beta = y$.

The original least squares model treats each item in the loss function equally. In fact, the impact of each item on the loss function should be different. Therefore, a more reasonable way is to use a weighted method.

The WLS method assigns a weight to each item in the loss function of the original least squares model. It aims to minimise the weighted loss function which can be expressed as follows:

$$E = \sum_{i=1}^{N} w_i \left(p\left(X_i\right) - y_i \right)^2 \tag{3.7}$$

where $w_i (i = 1, 2, \cdots, N)$ is the weight of the i^{th} sample in the approximation model.

Expression (3.7) can be derived into the following form:

$$E = \varepsilon^T W \varepsilon \tag{3.8}$$

where $\varepsilon = X\beta - y$.

W is the weight matrix which is symmetric positive definite and its scale is $N \times N$. In order to ensure that E is a real number, W should be a Hermitian matrix.

Let the partial derivative of function (3.8) with respect to β equal to 0, i.e.

$$\frac{\partial E}{\partial \beta} = 0 \tag{3.9}$$

Then we can obtain the WLS solution

$$\hat{\beta} = \left(X^T W X \right)^{-1} X^T W y \tag{3.10}$$

From expression (3.10) we can see that the solution varies with the change of weights, therefore, in order to find an optimal solution, we should analytically determine the value of weights.

4 Our proposed weight learning from misclassification cost-matrix based on genetic algorithm

In this paper, in order to generate a classifier which can achieve the minimum overall misclassification cost, we proposed a weight learning method for WLS model of ELM. By adopting genetic algorithm which aims to minimise the overall misclassification cost, this method assigns different weights to samples in different categories, that is the samples within the same category have the same weight.

In the WLS model of ELM, the training objective is to minimise the weighted loss function:

$$\begin{aligned} E = {} & w_1 \sum_{i=1}^{N_1} \left(\sum_{j=1}^{\tilde{N}} \beta_j g\left(\alpha_j \cdot x_i + b_j\right) - t_i \right)^2 \\ & + w_2 \sum_{i=N_1+1}^{N_2} \left(\sum_{j=1}^{\tilde{N}} \beta_j g\left(\alpha_j \cdot x_i + b_j\right) - t_i \right)^2 + \cdots \\ & + w_m \sum_{i=N_{m-1}+1}^{N_m} \left(\sum_{j=1}^{\tilde{N}} \beta_j g\left(\alpha_j \cdot x_i + b_j\right) - t_i \right)^2 \end{aligned} \tag{4.1}$$

where $N_k (k = 1, 2, \cdots, m)$ represents the number of samples contained in each category and $\sum_{k=1}^{m} N_k = N$; w_k is the weight of the samples in the k^{th} category, which is generated by genetic algorithm.

After the input weights and the biases are fixed, minimising the objective function (4.1) is equivalent to finding the optimal solution of the following expression:

$$\begin{aligned}
&\min_{\beta}\begin{pmatrix}\left\|w_1H_1\beta-w_1T_1\right\|^2+\left\|w_2H_2\beta-w_2T_2\right\|^2\\+\cdots+\left\|w_mH_m\beta-w_mT_m\right\|^2\end{pmatrix}\\
&=\min_{\beta}\left\|\begin{pmatrix}w_1H_1\\\vdots\\w_mH_m\end{pmatrix}\beta-\begin{pmatrix}w_1T_1\\\vdots\\w_mT_m\end{pmatrix}\right\|^2 \qquad (4.2)\\
&=\min_{\beta}\left\|\begin{pmatrix}w_1H_1\\\vdots\\w_mH_m\end{pmatrix}\beta-\begin{pmatrix}w_1T_1\\\vdots\\w_mT_m\end{pmatrix}\right\|^2
\end{aligned}$$

which can be equivalently expressed as:

$$\begin{pmatrix}w_1H_1\\\vdots\\w_mH_m\end{pmatrix}\hat{\beta}-\begin{pmatrix}w_1T_1\\\vdots\\w_mT_m\end{pmatrix} \qquad (4.3)$$

Then the output weight vector of the WLS model of ELM is analytically determined as:

$$\hat{\beta}=\begin{pmatrix}w_1H_1\\\vdots\\w_mH_m\end{pmatrix}^{+}\begin{pmatrix}w_1T_1\\\vdots\\w_mT_m\end{pmatrix} \qquad (4.4)$$

where '+' is the generalised inverse operator.

The weights w_1, w_2, $\cdots$, w_m can be generated by genetic algorithm which aims to minimise the training overall misclassification cost. Genetic algorithm can guarantee the obtained solution is the optimal one or an approximate optimal one. It consists of a series of genetic operations, such as selection, crossover and mutation.

In comparison with traditional optimisation methods, genetic algorithm has the following advantages.

1 It can find out the optimal solution. It searches from a cluster of potential solutions to another cluster, rather than from a single one to another one, which makes it can reduce the risk of falling into local optimal solutions.

2 It can process potential solutions in parallel.

3 It has the capability of self-organisation, self-adaptation and self-learning. When it uses the information obtained during the evolution process to organise the search by itself, the individuals with higher fitness have higher survival probability and obtain a more adaptive genetic structure.

Following is a detailed description of the construction process of the WLS model based on genetic algorithm.

1 Firstly, randomly assign input weights and biases to ELM and randomly generate a group of weight vectors which are considered as the individuals in an initial population.

2 Secondly, for every weight vector, calculate the output weights $\hat{\beta}$ according to expression (4.4), then we obtain the prediction model and prediction outputs for training samples, then according to the cost matrix, we can calculate the training overall misclassification cost which is considered as the fitness function used to evaluate the quality of individuals in genetic algorithm. Select weight vectors with satisfactory fitness values to perform the following genetic operations.

3 Thirdly, encode the selected weight vectors to strings composed of characters just as chromosomes consisting of genes.

4 Then apply crossover and mutation operators to the encoded individuals and obtain a superior generation.

Crossover refers to the operation of replacing the partial structure of two parent individuals to generate new individuals, which randomly exchanges two individuals in the population according to the crossover rate and is able to generate new combinations of genes, hoping to combine the beneficial genes together.

Mutation is to change the gene values at certain loci of individual strings in the population.

5 Iterate Steps 2–4 until the termination condition is met. Decode the final string and obtain the optimal weight vector which has the minimum training overall misclassification cost.

6 Lastly, calculate the final output weights according to expression (4.4), then we obtain the WLS model of ELM.

The pseudo code of this algorithm is shown as follows:

Algorithm 1 Algorithm for training a WLS model of ELM based on genetic algorithm

Input: input data set $\{(x_i, t_i)\}$, $i = 1, 2, \cdots, N$, where $x_i \in R^n$, $t_i \in R^m$; the number of nodes in the hidden layer: $\tilde{N}$; M: the scale of population; T: the maximum number of generations; Rc: crossover rate; Rm: mutation rate.

Output: the weight of each category $w_1, w_2, \cdots, w_m$; training overall cost; testing overall cost.

```
Begin
    Initialise t = 0;
    Generate the first population P(0) randomly;
    Encode the individuals in P(0) to strings;
    while (t <= T) do
        Calculate the fitness of each individual in P(t);
        Select individuals from P(t) according to their
        fitness values;
        Perform crossover operation on the selected
        individuals according to Rc;
        Perform mutation operation on the selected
        individuals according to Rm;
        Generate a new population P(t + 1);
        t = t + 1;
    end while
    Decode the string with the maximum fitness in P(T) to
    a vector [w1, w2, ···, wm]^T which is the optimal
    solution;
```

14 Generate the input weights and biases randomly;

15 Calculate the output weights of the proposed model according to expression (4.4);

16 end

17 Return the weight of each category w_1, w_2, ⋯⋯, w_m; training overall cost; testing overall cost.

5 Experimental validation

In order to verify the effectiveness of the proposed method, we have done a lot of experiments conducted on various UCI datasets whose detailed information is shown in Table 1.

The processor of the computer we used for experiments is Intel (R) Core (TM) i3-6100 with 16GB memory space. The programming software is MATLAB R2016a 9.0 with 64 bits.

In all the experiments, we use 70% of the samples in the dataset as training data and the rest as test data. The evaluation index is overall cost which is the sum of costs caused by all the misclassifications. Usually the cost matrix in an imbalanced classification problem is given by experts with professional knowledge and rich experience.

In the course of experiments, we found that for a fixed dataset, the finally obtained overall cost is not a fixed value. Next, we will take the result of the Texture-r dataset as an example to illustrate this phenomenon.

Table 1 Information of datasets

Dateset	*Number of attributes*	*Number of categories*	*Number of instances*
autompg	5	3	392
page	10	3	5,473
Page (p10)	10	3	546
segment	19	3	2,310
Texture-r	21	3	5,500
Thyroid (p10)	6	3	720
vehicle	8	3	846
vowel	10	3	528
wineQR	11	3	1,599
wineQW	11	3	4,898
yeast	8	3	1,484
cmc	9	3	1,473
eb	4	3	4,210
krkopt	6	3	2,902
localisation	7	3	2,000
page-blocks	3	3	490
recognition	3	3	658
sat	36	3	4,435

Table 2 Result of the texture-r data set

w_1	w_2	w_3	*overall cost*
0.9667	0.8425	0.0988	1,041
0.4517	0.2935	0.2582	805
0.7203	0.6660	0.0354	882
0.6300	0.4053	0.1724	905
0.7732	0.3178	0.2309	668
0.0352	0.0156	0.0005	508
0.8042	0.4649	0.3048	981
0.6062	0.3223	0.1877	1,034
0.5871	0.1276	0.0310	1,033
0.6153	0.9212	0.8642	802
0.7099	0.6654	0.3369	973
0.8662	0.6599	0.0350	895
0.9569	0.9051	0.6676	1,011
0.9002	0.8150	0.6265	659

In Table 2, w_1, w_2 and w_3 are the weights of the samples in three categories respectively. From Table 2 we can see that the overall cost changes with the weights. Then we can say genetic algorithm has the characteristic of uncertainty. The reason is that in order to find out the global optimal solution in a large range and avoid falling into local optimal solutions, genetic algorithm adopts many random operations where the most representative ones are listed as follows:

1 Use the roulette method which is random to select individuals for reproduction.

2 Randomly select individuals and switching points for crossover.

3 Randomly select individuals and variation points for mutation.

In order to minimise the influence of uncertainty on the proposed model and ensure the fairness of experimental results, we use the average value of overall costs from multiple experimental results to evaluate the performance of a classifier.

By conducting experiments, we compared the proposed model with the cost-sensitive ELM and the cost-sensitive naive Bayes model. The average overall cost on testing data are shown in Table 3.

From the comparative experimental results we can see that the WLS model of ELM has obvious advantage in comparison with the other two methods. It can achieve much less overall misclassification cost. However, on the recognition dataset, the results of these three models are close to each other. The possible reasons may be the following ones.

1 Genetic algorithm has strong global search ability. It can find out the global optimal solution.

2 The dataset itself has an impact on the experimental results. Therefore, the proposed algorithm performs different on various datasets.

Table 3 Comparative average overall cost on testing data

Dataset	*WLS model of ELM*	*cost-sensitive ELM*	*cost-sensitive naive Bayes*
autompg	639.354	3,281	1,108
page	1,048.203	35,698	3,454
Page(p10)	97.318	6,710	519
segment	25.156	2,605	3,402
Texture-r	349.643	6,055	8,688
Thyroid(p10)	501.642	7,071	839
vehicle	1,001.547	3,919	2,521
vowel	38.547	845	894
wineQR	870.317	3,693	1,534
wineQW	2,694.500	12,853	5,581
yeast	3,611.698	6,119	5,947
cmc	5,864.314	8,995	7,663
eb	1,960.417	4,245	2,125
krkopt	246.167	18,241	469
localisation	4,643.258	7,942	8,141
page-blocks	1,905.617	6,922	7,106
recognition	1,894.514	2,052	2,100
sat	1,866.316	2,631	7,111

6 Conclusions and future work

6.1 Conclusions

Usually in real life, for a classification problem, different misclassifications causes different costs. Cost-sensitive learning introduces the concept of misclassification cost into the design of classifiers. However, the existing cost-sensitive classifiers cannot ensure the final overall cost achieves its minimum value. In order to overcome this defect, we did the following work in this paper.

1 We proposed a weight learning method for constructing a WLS model of ELM where the weights of samples in each category are generated by using the genetic algorithm which aims to minimise the training overall misclassification cost. This model can make sure the generated classifier has the minimum overall cost.

2 We completed the transition from the MCM to the weights in the weighted least square model. Usually, the weights are meaningless. In this paper, the weights are generated based on the MCM, which makes the weights interpretable.

3 We conducted comparative experiments on 18 datasets. The results showed that in comparison with the cost-sensitive ELM and the cost-sensitive naive Bayes model, the weighted least square model of ELM based on genetic algorithm has much better performance.

6.2 Future work

Our research work can be further improved from the following aspects:

1 In this paper, only the UCI datasets are used. In the later stage, the size of the dataset will be increased to adapt to the development trend of big data.

2 In this paper, we only discussed the datasets with three categories. We will do further experiments on multi-classification problems to verify the proposed method more comprehensively.

3 At present, we only know that the weight assigned to each category depends on the cost matrix. In the follow-up work, we will explore the specific relationship between them.

4 We have not explained the interpretable feature in detail. We will complete this work through experiment further.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant 61772344 and Grant 61732011), in part by the Natural Science Foundation of SZU (Grant 827-000140, Grant 827-000230 and Grant 2017060).

References

Abe, N., Zadrozny, B. and Langford, J. (2004) 'An iterative method for multi-class cost-sensitive learning', *Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.3–11.

Alshamiri, A.K., Singh. A. and Surampudi, B.R. (2018) 'Two swarm intelligence approaches for tuning extreme learning machine', *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 8, pp.1271–1283.

Bradford, J.P., Kunz, C., Kohavi, R. et al. (1998) 'Pruning decision trees with misclassification costs', *Lecture Notes in Computer Science*, Vol. 1398, No. 1398, pp.131–136.

Chen, Y., Song, S., Li, S. et al. (2018) 'Domain space transfer extreme learning machine for domain adaptation', *IEEE Transactions on Cybernetics*, Vol. PP, No. 99, pp.1–14.

Dastidar, J.G., Sarkar, S., Sinha, R.P. et al. (2015) 'Handwriting recognition', *Invited Session Papers from the Second Asian Conference on Computer Vision: Recent Developments in Computer Vision*, Springer-Verlag, pp.447–456.

Ding, S., Zhang, N., Zhang, J. et al. (2017) 'Unsupervised extreme learning machine with representational features', *International Journal of Machine Learning and Cybernetics*, Vol. 8, No. 2, pp.587–595.

Domingos P. (1999) 'MetaCost:a general method for making classifiers cost-sensitive', *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp.155–164.

Drummond, C. and Holte, R.C. (2000) 'Exploiting the cost (in)sensitivity of decision tree splitting criteria', *Seventeenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., pp.239–246.

García-López, S., Jaramillo-Garzón, J.A., Duque-Muñoz, L. et al. (2015) 'A methodology for optimizing the cost matrix in cost sensitive learning models applied to prediction of molecular functions in embryophyta plants', *Journal of Discrete Algorithms*, Vol. 5, No. 4, pp.696–705.

Geibel, P., Brefeld, U. and Wysotzki, F. (2004)' Perceptron and SVM learning with generalized cost models', *Intelligent Data Analysis*, Vol. 8, No. 5, pp.439–455.

Guo, H., Liu, B., Cai, D. et al. (2018) 'Predicting protein-protein interaction sites using modified support vector machine', *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 3, pp.393–398.

Hagan, M.T., Demuth, H.B. and Beale, M.H. (2002) *Neural Network Design*, China Machine Press,.

Hu, L., Chen, Y., Wang, J. et al. (2018) 'OKRELM: online kernelized and regularized extreme learning machine for wearable-based activity recognition', *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 9, pp.1577–1590.

Huang, G.B., Zhu, Q.Y. and Siew, C.K. (2006) 'Extreme learning machine: theory and applications', *Neurocomputing*, Vol. 70, No. 1, pp.489–501.

Kukar, M.Z. and Kononenko, I. (1998) 'Cost-sensitive learning with neural networks', *13th European Conference on Artificial Intelligence (ECAI 98)*, Brighton England, 23–28 August, pp.445–449.

Liu, M., Liu, B., Zhang, C. et al. (2017) 'Semi-supervised low rank kernel learning algorithm via extreme learning machine', *International Journal of Machine Learning and Cybernetics*, Vol. 8, No. 3, pp.1039–1052.

Liu, S., Tong, J., Meng, J. et al. (2018) 'Study on an effective cross-stimulus emotion recognition model using EEGs based on feature selection and support vector machine', *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 5, pp.721–726.

Luo, X., Yang, X., Jiang, C. et al. (2018) 'Timeliness online regularized extreme learning machine', *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 3, pp.465–476.

Mao, W., Wang, J. and Xue Z. (2017) 'An ELM-based model with sparse-weighting strategy for sequential data imbalance problem', *International Journal of Machine Learning and Cybernetics*, Vol. 8, No. 4, pp.1333–1345.

Peterson, L. (2009) 'K-nearest neighbor', *Scholarpedia*, Vol. 4, No. 2, p.1883.

Quinlan, J.R. (1986) 'Induction on decision tree', *Machine Learning*, Vol. 1, No. 1, pp.81–106.

Rish, I. (2001) 'An empirical study of the naive Bayes classifier', *Journal of Universal Computer Science*, Vol. 1, No. 2, p.127.

Russ, J.C. and Neal, F.B. (2017) 'The image processing handbook', *Computers in Physics*, Vol. 8, No. 2, p.177.

Sun, C.T., Ye, S.H. and Hsieh, H.C. (2014) 'Effects of student characteristics and question design on internet search results usage in a Taiwanese classroom', *Computers and Education*, Vol. 77, No. 77, pp.134–144.

Wang, X., Matwin, S., Japkowicz, N. et al. (2013) 'Cost-sensitive boosting algorithms for imbalanced multi-instance datasets', *Advances in Artificial Intelligence*, Springer Berlin Heidelberg, pp.174–186.

Zadrozny, B., Langford, J. and Abe, N. (2003) 'Cost-sensitive learning by cost-proportionate example weighting', *IEEE International Conference on Data Mining*, IEEE, pp.435–442.

Zhai, J., Zhang, S. and Wang, C. (2017) 'The classification of imbalanced large data sets based on MapReduce and ensemble of ELM classifiers', *International Journal of Machine Learning and Cybernetics*, Vol. 8, No. 3, pp.1009–1017.

Zhao, X., Cao, W., Zhu, H. et al. (2018) 'An initial study on the rank of input matrix for extreme learning machine', *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 5, pp.867–879.