International Journal of Database Theory and Application
Vol.9, No.6 (2016), pp.285-298
http://dx.doi.org/10.14257/ijdta.2016.9.6.29

Dynamic Cost-Sensitive Extreme Learning Machine for Classification of Incomplete Data Based on the Deep Imputation Network

Fuxian Huang[1,3], Chunying Liu [2,3], Yuwen Huang[2,3] and Jijiang Yu[3,4,*]

[1] *Dean's Office, Heze University, Shandong, China*
[2]*Department of Computer and Information Engineering, Heze University, Heze 274015, Shandong, China*
[3] *Key Laboratory of computer Information Processing, Heze University, Heze 274015, Shandong, China*
[4]*State-owned assets management, Heze University, Shandong, China*
[*] *Corresponding Author E-mail: hzxy_yjj@163.com*

Abstract

Due to its importance in many applications, the incomplete data mining has received increasing attention in recent years, but there has been little study of the cost-sensitive classification on incomplete data. Therefore this paper proposes the dynamic cost-sensitive extreme learning machine for classification of incomplete data based on the deep imputation network (DCELMIDC). Firstly, we propose an approach for incomplete data imputation based on the deep imputation network model, and offer the cost-sensitive extreme learning machine. Secondly, this paper introduces dynamic misclassification and test cost, and gives the chromosome coding and an evaluation method of the optimal cost. At last, on the basis of the genetic algorithm, the dynamic cost-sensitive extreme learning machine classification algorithm for mining incomplete data is given, which can search the optimal misclassification and test cost in cost spaces. The experiment results show that DCELMIDC is effective and feasible for classification of incomplete data, and can reduce the total cost.

Keywords*: Extreme learning machine, cost-sensitive, deep imputation network, incomplete data*

1. Introduction

In the medical diagnosis and industrial field, there are more and more incomplete data for the measurement error, limitations on getting data, and random noise. The traditional data mining techniques are designed for complete data, and do not mine effectively the incomplete data. Therefore, it is significant for the researcher of classification method of incomplete data. The paper [1] proposed a Robust Bayes Classifier with the feature selection and classification tasks for incomplete data. The paper [2] gave a new approach for mining incomplete data with singleton, subset and concept probabilistic approximations. The paper [3] proposed a method for mining incomplete quantitative data sets by rough sets. The paper [4] proposed a new approach of mining missing values for optimizing semiconductor-manufacturing processes. The paper [5] proposed the elastic extreme learning machine based on MapReduce framework, which can classify big data sets by using the update matrix multiplications.

In 2004, professor Huang proposed firstly the extreme learning machine algorithm for the single hidden layer feed forward network. The extreme learning machine randomly assigns the weights and hidden layer nodes bias of the neural network, and can calculate the output weights by only one step, so the learning speeds of the neural network are greatly improved with abnormal generalization performance. The paper [6] considered the problem of extending extreme learning machine to non-redundant synergy pattern based graph classification. The paper [7] gave a new image classification method based on extreme k-means and effective extreme learning machine. The paper [8] proposed a method of the extreme learning machines for soybean classification, which can obtain accurate thematic maps of soybean crops. The paper [9] gave a new approach of extreme learning machine for multi-category sparse data classification by selecting the optimal number of hidden neurons.

The cost is an important factor in the practical application, so the cost-sensitive learning has received increased attention in recent years. The main goal of cost-sensitive learning is to generate the classifier with minimum cost, and the cost-sensitive learning has been widely used in data mining. The paper [10] proposed a cost-sensitive classification approach for imbalanced big data under MapReduce framework. The paper [11] offered a novel cost-sensitive ensemble based on decision trees for imbalanced classification. The paper incorporated various Bayesian networks to form cost-sensitive Bayesian network classifiers [12]. The paper [13] used cost-sensitive Support Vector Machines for electrocardiogram beat classification. But so far, few research on cost-sensitive learning with data uncertain. The cost is difference in the real world, for example, the same blood examination is different cost in different hospital, so the cost is dynamic. There has been little study of the dynamic cost-sensitive classification of incomplete data, so this paper proposes the dynamic cost-sensitive extreme learning machine for classification of incomplete data based on the deep imputation network. Our goals are 1) to devise an algorithm for building dynamic cost-sensitive extreme classifier for incomplete data based on the deep imputation network, 2) to investigate whether the method for incomplete data classification with deep imputation could lead to a higher classification accuracy compared with the averaging approach, and 3) to establish a theoretical foundation on dynamic cost-sensitive extreme learning machine for incomplete data.

2. Data Imputation for Incomplete Data Based on the Deep Imputation Network Model

2.1. Fill-in Automatic Code Machine

Data set $X=\{X_1,X_2,....,X_n\}$, $X_i=\{x_{i1},x_{i2},...,x_{im}\}$, $X=CX\cup UX$, $CX=\{CX_1,CX_2,...CX_t\}$ is certain data set, $UX=\{UX_1,UX_2,...,UX_p\}$ is incomplete data set , $t+p=n$, $CX\cap UX=\phi$. This paper builds the depth fill-in network on the basis of fill-in automatic code machine, which randomly selects the portion of the objects from the complete data as the instances for parameters of the fill-in automatic code machine, and the incomplete data are filled in. At first, Data set X is preprocessed, and all attributes are transferred as the converted into numerical values. In order to train the network parameter, in the process of structuring the fill-in automatic code machine, the selected objects simulate the missing object, and their part attributes are randomly set as zero for simulating the input of the fill-in automatic code machine by the incomplete objects. The structure of the fill-in automatic code machine is as follows.

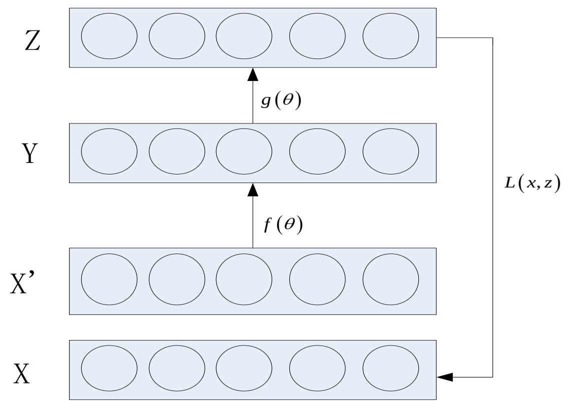

Figure 1. Structure of the Fill-in Automatic Code Machine

The incomplete samples $X=\{X_1,X_2,....,X_n\}$, $X_i=\{x_{i1},x_{i2},...,x_{im}\}$, the part attributes of X are set as zero to turn into the incomplete samples X'. In input layer, the fill-in automatic code machine receives the input X' , and the mapping function from the hide layer is as follows.

$$y=f_\theta(x)=\sigma(Wx+b)$$

$\theta=\{W,b\}$ is the network parameter, and W is the weight of each layer nerve cell. The matrix size is $m'\times m$, and b is the offset. $\sigma(x)=1/(1+\exp(-x))$ is Sigmoid function. In decode procedure, and Z is reconstructed by the backward mapping.

$$Z=g_{\theta'}(y)=\sigma(W'y+b')$$

$$\theta'=\{W',b'\},W'=W^T \tag{1}$$

By the minimum average reconstruction error training the model parameter, the reconstruction data approaches the real instance objects.

$$\theta^*,\theta^{*'}=\arg\min_{\theta^*,\theta^*}\frac{1}{n}\sum_{i=1}^{n}L(x_i,z_i) \tag{2}$$

Use the cross entropy function $-\sum_i p_i\log p_i$ to calculate the distance of x and z, and the loss function is :

$$L(x,z)=-\sum_{k=1}^{m}\left(x_k\log z_k+(1-x_k)\log(1-z_k)\right) \tag{3}$$

According to the gradient descent algorithm, when a instance is chosen from complete data set X , the fill-in automatic code machine chooses randomly the part attributes that are set as zero, and obtains the data X and x' . At the same time, the parameters w and b are updated as follows.

$$W'=W-\eta\left(\frac{\partial(L(X,Z))}{\partial W}+\lambda W\right)$$

$$b'=b-\eta\left(\frac{\partial(L(X,Z))}{\partial b}+\lambda b\right) \tag{4}$$

η is learning rate, λ is the attenuation factor of the weight for avoiding the overfitting.

2.2. Imputation Algorithm for Uncertain Data Based on the Deep Imputation Network

In order to acquire the supervision objects of each network layer, the superposed automatic code machine is constructed by the instance data, and the superposed automatic code machine is as follows.

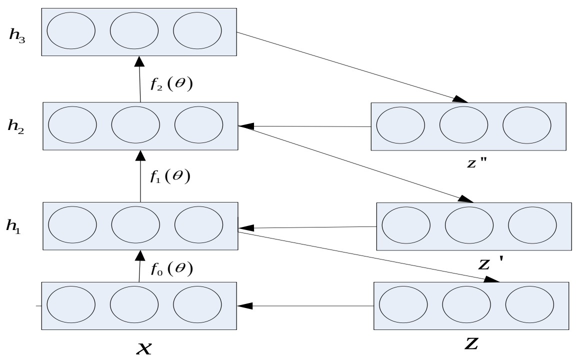

Figure 2. Three Layers Superposed Automatic Code Machine

The bottom network gets the character $h_1 = f_{\theta^0}(x)$ by the input data x, and h_1 is trained as the input of its top network. $h_2 = f_{\theta^1}(h_1)$, the second layer network renews the network weight, and it does not influence the lower network. $h_0 = x$, $h_k = f_{\theta^{k-1}}(h_{k-1})$. The network weights are initialized by the hierarchical iteration, and the total network weights are renewed by the reversed dissemination method.

Imputation algorithm for incomplete data based on the deep imputation network is as follows.

Input: Data set $X = \{X_1, X_2, ..., X_n\}$, $X_i = \{x_{i1}, x_{i2}, ..., x_{im}\}$, $X = CX \cup UX$, Certain data set $CX = \{CX_1, CX_2, ...CX_t\}$, Incomplete data set $UX = \{UX_1, UX_2, ..., UX_p\}$.

Output: Incomplete data set $X = \{X_1, X_2, ..., X_n\}$

Step 1: Input the certain data set x, $x \in CX$, and use the superposed automatic code machine to get the character $t1, t2$ of instance data .

Step 2: Set the part attributes of x as zero to get the incomplete sample x', and use x, $t1, t2$ as the supervised data.

Step 3: Input x' to get the first layer character $t_1^{'} = f_{\theta^0}(x')$ and $z = g_{\theta^0}(t_1^{'})$. Use the superposed automatic code machine to get the character t_1 of instance data x, and get the network parameter θ^0.

Step 4: Input $t_1^{'}$ to get the second layer character $t_2^{'} = f_{\theta^1}(t_1^{'})$ and $z_1 = g_{\theta^1}(t_2^{'})$. Use the superposed automatic code machine to get the character t_2 of instance data x, and get the network parameter θ^1.

Step 5: Input $t_2^{'}$ to get the third layer character $t_3^{'} = f_{\theta^2}(t_2^{'})$ and $z_2 = g_{\theta^1}(t_3^{'})$ x, and get the network parameter θ^2.

Step 6: Take out from data set one by one $CX=\{CX_1,CX_2,...CX_t\}$, and perform repeatedly from steps 1 to step 6, until the network tends towards stability, and get the network parameter $\theta^0,\theta^1,\theta^2$.

Step 6: Set the uncertain attributes of incomplete sample UX_i as zero to get $UX_i^{'}$, and extract its depth features t, $t=f_{\theta^2}\left(f_{\theta^1}\left(f_{\theta^0}\left(UX_i^{'}\right)\right)\right)$.

Step 7: use $\tilde{ux}_i=g_{\theta^0}\left(g_{\theta^1}\left(g_{\theta^2}\left(t\right)\right)\right)$ to restore the incomplete sample, and get the missing attributes for uncertain data set UX_i.

3. Cost-Sensitive Extreme Learning Machine

3.1. Extreme Learning Machine

Extreme learning machine is an algorithm of the feed forward Neural Network, and it does not need to adjust the weights of the input layer and hidden layer during its execution. The extreme learning machine can generate the unique optimization solution by setting the node numbers of network hidden layer, and has the fast learning speed and good generalization performance.

The training sample (x_i,t_i) , $x_i\in R^m,t_i\in R^n,i=1,2,...N$. $x_i=(x_{i1},x_{i2},...,x_{in})^T$, $t_i=(t_{i1},t_{i2},...,t_{im})$. The mathematical model of standard single hidden layer feedforward neural network can be expressed as follows.

$$\sum_{i=1}^{L}\beta_i f\left(w_i\bullet x_j+b_i\right)=t_j \quad j=1,2,...,N$$

$$w_i=\left(w_{i1},w_{i2},...,w_{in}\right)^T$$

$$\beta_i=\left(\beta_{i1},\beta_{i2},...,\beta_{im}\right)^T \tag{5}$$

L is input neuron hidden layer node number. $f(x)$ is the excitation function. w_i is the input weight between the input neuron and the ith layer node. b_i is the offset vector of the ith layer node. β_i is the output weight between the input neuron and the ith layer node. $w_i\bullet x_j$ is the inner product. The above formula is abbreviated as $H\beta=T$.

$$H\left(w_1,w_2,...,w_N,b_1,b_2,...,b_N,x_1,x_2,...,x_N\right)=$$

$$\begin{bmatrix} f(w_1\bullet x_1+b_1) & f(w_2\bullet x_1+b_2) & ...f\left(w_N\bullet x_1+b_N\right) \\ f(w_1\bullet x_2+b_1) & f(w_2\bullet x_2+b_2) & ...f\left(w_N\bullet x_2+b_N\right) \\ ... & ... \quad ... & ... \\ f(w_1\bullet x_N+b_1) & f(w_2\bullet x_N+b_2) & ...f\left(w_N\bullet x_N+b_N\right) \end{bmatrix}$$

$$\beta=\begin{bmatrix}\beta_1^T\\ \beta_2^T\\ ...\\ \beta_N^T\end{bmatrix}_{N\times m} \qquad T=\begin{bmatrix}t_1^T\\ t_2^T\\ ...\\ t_N^T\end{bmatrix}_{N\times m}$$

H is output matrix of neural network hidden layer, and $x_1, x_2, ..., x_N$ is the ith hidden node. When w and β are a fixed value, the output matrix H in hidden layer is also the fixed matrix, and the train of the feed forward Neural Network is how to solve the least square solution β for the linear system.

$$H\beta = T, \beta = H^{+}T \tag{6}$$

H^{+} is the Moore-penrose generalized inverse matrix of the hidden output matrix H.

3.2. Cost-Sensitive Extreme Learning Machine

The cost-sensitive machine learning considers that the cost is asymmetric in the course of classification, and the minimum cost based on Bayesian risk is:

$$R(\alpha_i | x) = \min \arg_{\alpha_i \in C} \sum_{j=1}^{M} \lambda(\alpha_i | w_j) P(w_j | x) \tag{7}$$

$R(\alpha_i | x)$ is the conditional risk that x is classified as α_i. $\lambda(\alpha_i | w_j)$ is the loss cost. $P(w_j | x)$ is the prior probability from which the classification w_j is selected.

The steps of the cost-sensitive extreme learning machine are:

Input: Training data set $X = \{x_1, x_2, ..., x_n\}$, the classification number n, the misclassification cost matrix $mis\_\mathrm{cost}_{N\times N}$, the test cost matrix $test\_\mathrm{cost}_N$, the hidden layer nodes number N.

Output: Classification results of $T = \{t_1, t_2, ..., t_n\}$.

```
Step 1. for (i=1; i<=K; i++)
{
    1. Set $(w_j^i, b_j^i)$  $j = (1, 2, ..., N)$.
    2. The output matrix $H_{L\times M}^i$ is as follows.
       $H_{L\times M}^i = \begin{bmatrix} f(w_1x_1+b_1) & ... & f(w_1x_M+b_1) \\ ... & ... & ... \\ f(w_Lx_1+b_L) & ... & f(w_Lx_M+b_L) \end{bmatrix}$.
    3. $\beta^i = (H^i)^{+} T$, $T$ is the output matrix .
    4. Save the current extreme learning machine.
}
Step 2. Use each extreme learning machine to predict $x_i$. If the classification $j Î \{c_1, c_2, ..., c_n\}$, $W_{k,x_i}(j) = W_{k,x_i}(j) + 1$. Get K classifications for each training data $x_i$.
Step 3. Calculate $P(\mathrm{j} | x_i) = \frac{W_{k,x_i}(j)}{K}$, x = $1,2,...,n$, $j Î \{c_1, c_2, ..., c_n\}$.
Step 4. $R(C_i | x) = \min \arg_{iÎM} \overset{M}{\underset{j=1}{å}} Cost(i,j) P(j | x)$
    , $Cost(i,j) = mis\_\mathrm{cost}_{ij} + tes\_\mathrm{cost}_j$, and output the classifications of $\{t_1, t_2, ..., t_n\}$.
```

4. The Proposed Scheme

4.1. Dynamic Misclassification and Test Cost

There are many costs in the classification process, and the misclassification and test costs are common. The misclassification cost and test cost is the consumption for obtaining the test attribute value. $mis\_\mathrm{cost}_{M\times M}$ is a misclassification cost matrix, and each element $C_{i,j}$ is the cost when a real class C_i is misclassified as the wrong class C_j. $C_{i,j}=0$ if i and j is equal.

$$mis\_\mathrm{cost}_{M\times M}=\begin{bmatrix} mis\_\cos t_{11}, mis\_\cos t_{12},\ldots, mis\_\cos t_{1\mathrm{M}} \\ mis\_\cos t_{21}, mis\_\cos t_{22},\ldots, mis\_\cos t_{2\mathrm{M}} \\ \ldots\ldots\ \ldots\ \ldots\ \ldots\ldots\ \ldots\ \ldots\ \ldots\ldots\ \ldots\ \ldots\ \ldots \\ mis\_\cos t_{\mathrm{M}1}, mis\_\cos t_{\mathrm{M}2},\ldots, mis\_\cos t_{\mathrm{MM}} \end{bmatrix}$$

$Test\_\cos t$ is test-cost vector, and each entry $test\_\cos t_i$ is the cost of testing on attribute A_i . The current research concentrates on the static cost, and the cost is the same in different applications. However, in real life, costs are dynamic and always changing and the different applications have different costs. To overcome the limitations of the static cost, this paper integrates the experts' experience and knowledge, and searches the suitable misclassification and test costs for finding the optimal cost in every sub dataset. All costs in every sub dataset make up the cost space, and the structure process for cost space refers to the paper [14].

4.2. Evaluation Method for the Optimal Cost

It is important to find the optimal evaluation method for a classifier performance, and G-mean is usually used. The different applications have different costs, and we find the optimal misclassification and test cost by searching the reasonable misclassification and test cost space. The genetic algorithm is an excellent algorithm which is often used to search the optimal cost by the fitness function. This paper uses the extended $G-mean$ as the fitness function, and Precision and Recall are two parameters of G-mean. The fitness function is as follows.

$$\begin{aligned} fitness(\cos t) &= G(Precision, Recall) \\ &= \sqrt{Precision \times Recall} \end{aligned} \quad Precision=\frac{TP}{TP+FP}\ .$$

$$Recall=\frac{TP}{TP+FN} \tag{8}$$

TP is the number of true positive, FP is the number of false positive, FN is the number of false negative. $\cos t$ is a reasonable value of misclassification and test cost space that is given by application specialists, and the fitness function can determine the optimal cost by balancing Precision and Recall of a classifier.

4.3. Chromosome Coding

The cost $mis\_\cos t$ and $tes\_\cos t$ is the interval number, $mis\_\mathrm{cost}_{ij}\in\left[mis\_\cos t_{ijl}, \mathrm{mis\_cost}_{ijh}\right]$, $tes\_\cos t_i \in\left[test\_\cos t_{il}, \mathrm{test\_cost}_{\mathrm{ih}}\right]$. In order to insure the upper is lower than the upper in the coding, the center and the radius of the interval numbers are coded. Each chromosome code is as follows.

$$\underbrace{test\_c_1, test\_r_1, test\_c_2, test\_r_2, ..., test\_c_n, test\_r_n}_{test\ cost},$$

$$\underbrace{mis\_c_{11}, mis\_r_{11}, mis\_c_{12}, mis\_r_{12}, ..., mis\_c_{1m}, mis\_r_{1m}}_{mistake\ test},$$

$$\underbrace{mis\_c_{21}, mis\_r_{21}, mis\_c_{22}, mis\_r_{22}, ..., mis\_c_{2m}, mis\_r_{2m}}_{mistake\ test},$$

....

$$\underbrace{mis\_c_{m1}, mis\_r_{m1}, mis\_c_{m2}, mis\_r_{m2}, ..., mis\_c_{mm}, mis\_r_{mm}}_{mistake\ test}$$

$test\_c_i$ is the center for test cost, and mis_r_i is its radius. mis_c_{ij} is the center for misclassification cost, and mis_r_{ij} is its radius. When the chromosome is decoded, $mis\_cost_{ijl} = mis\_c_{ij} - mis\_r_{ij}$, $mis\_cost_{ijh} = mis\_c_{ij} + mis\_r_{ij}$, $test\_cost_{il} = test\_c_i - test\_r_i$, $test\_cost_{ih} = test\_c_i + test\_r_i$.

4.4. Dynamic Cost-Sensitive Extreme Learning Machine for Mining Incomplete Data Base on the Genetic Algorithm

There are many fast algorithms for data mining [15,16], and this paper fills firstly the uncertain value by the deep filling algorithm, and uses the cost-sensitive extreme learning machine to classify the data set. In the classification process, use the genetic algorithm to search the optimal cost. The cost parameters is set in within certain realms, and initial cost can been assigned randomly. The steps of classification algorithm for cost-sensitive extreme learning machine are as follows.

Import:

D : Uncertain data set $X = \{X_1, X_2, ..., X_n\}$, $X_i = \{x_{i1}, x_{i2}, ..., x_{im}\}$, $X = CX \cup UX$, CX is certain data set, UX is uncertain data set.

$Test\_cost$: Test cost vector, and an element is $[tc_{il}, tc_{ih}]$.

$mis\_cost_{M\times M}$: Matrix for misclassification cost, and an element is $[c_{ijl}, c_{ijh}]$.

P_c , P_m : Probability of crossover and mutation.

Export: Classification result for uncertain data set $X = \{X_1, X_2, ..., X_n\}$.

Step 1. Initialize: Randomly generate an initial population with chromosome number N , T , P_c , P_m , TC , $mis\_cost_{M\times M}$, $Test\_cost$.

Step 2. Calculate the network parameters by the drill training ideas and back-propagation algorithm, and build the deep imputation network model by analyzing the deep features of incomplete data. Use the deep imputation network to impute the missing values.

Step 3. Use Cost-sensitive Extreme learning machine to build classifier M, and get Recall and Precision by 10-fold cross validation. Calculate the fitness $fitness(x_i)$ of each chromosome by the fitness function.

Step 4. Selecting operation.

$$p_i = \frac{fitness(x_i)}{\sum_{j=1}^{m} fitness(x_j)} \quad i = 1, 2, ..., m \ .$$

Calculate the probability p_i for each chromosome, and form a new population New_Pop by roulette selection.

Step 5. Crossover operation.

Select chromosomes for crossover operation by the probability Pc in New_Pop, and reform a new population $cross\_pop$.

Step 6. Mutation operation.

Alter chromosomes by the probability Pm in population $cross\_pop$, and structure a new population mut_pop.

Step 7. If the iteration is greater than 1000, save the chromosome with the maximum fitness value $G_{\max}$ in population mut_pop, and turn to step 8. Otherwise, turn to step 3.

Step 8. Get the misclassification cost mis_cost and test cost $test\_cost$ from the chromosome with the maximum fitness value. Apply mis_cost and $test\_cost$ to data set D, and structure the classifier M by the Cost-sensitive Extreme Learning Machine algorithm.

Step 9. Use the classifier M to classify the data set.

Step 10. The algorithm ends.

5. Simulation Experiment

5.1. Data Processing and Background Parameters

There are no open standard incomplete data sets, so some data are removed artificially from the standard UCI data sets for simulating the incomplete data. In the experiments, in order to simplify the model design, we ignore the influence of various parameters in genetic algorithm, and focus only on the structure and performance of the algorithms, while at the same time, assuming that units of all kinds of cost are the same. Misclassification cost matrix and test cost are artificially set by the previous empirical data. When classification is correct, the misclassification is zero, and the cost of TP and TN are zero because they are classified correctly. In order to observe the influence of different misclassification cost, the ratio of FP and FN is 1/10 when classification is error. Test cost is a random number of [0,100], and we assume that all attributes are unknown in the experiment, so test cost is paid if the attribute value is determined. Choose Windows 7 with Intel E5800 (3.2GHz) and Weka3.7.1 as development platform, and RAM is 4.0GB. Use the discretization class of Weka to discretize the quantitative attribute in UCI.4.2.

5.2. Result of Experiments

The cost-sensitive extreme learning machine based on the deep imputation network and genetic algorithm for classification of incomplete data is abbreviated to DCELMIDC. In order to test the infectiveness of DCELMIDC for classifying incomplete data, choose Cylinder, Mushroom and Vote in UCI dataset as test data, and the ratio of incomplete data is increased from 10% to 50% at the rate of 10%. RBC [17], GBSD [18] are other classifiers for incomplete data. Average accuracy of the three classifiers is as follows after 100 repeated experiments in each data set.

Table 1. Average Accuracy of the Three Classifiers

Algorithm	Cylinder	Mushroom	Vote	Bridges
RBC	71.72%	95.97%	90.34%	62.17%
GBSD	78.13%	99.69%	96.42%	70.25%
DCELMIDC	80.25%	96.73%	97.53%	78.49%

Form Table 1, DCELMIDC has higher classification accuracy than RBC in all data sets, and has higher than GBSD except Mushroom, so DCELMIDC is effective and feasible for classification of incomplete data.

In order to verify the performance differences of dynamic and static cost-sensitive classifiers for classification of incomplete data, the dynamic cost in DCELMIDC is replaced by static cost, and gets static cost-sensitive classifier for incomplete data (SCELMIDC). The missing data ratio is set as 10%, and use Area Under the ROC Curve (AUC) as the performance criteria. The experiment results are as follows in Figure 3 after 100 times in Cylinder, Mushroom, Vote, Bridges and Zoo.

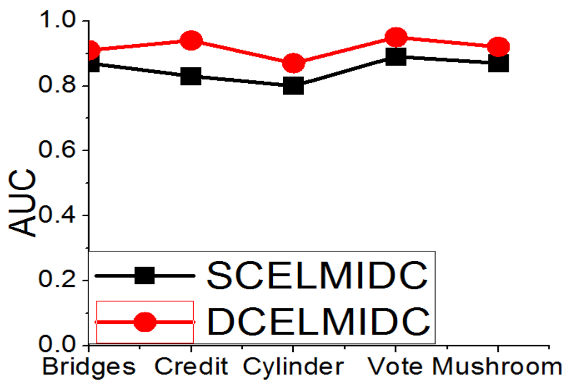

Figure 3. AUC of SCELMIDC and DCELMIDC

As can be seen from Figure 3, AUC of DCELMIDC is better than SCELMIDC, so dynamic cost has superior performance than static cost.

In order to verify the average total cost of classifiers, set the fixed cost in RBC and GBSD, and DCELMIDC using the genetic to search the optimal cost near the fixed number in experiments. Test cost is paid if the incomplete data are filled. The misclassification cost is paid when the classification is in error. The ratio of incomplete data is set as 10%, and average total costs of DCELMIDC, RBC and GBSD are as follows.

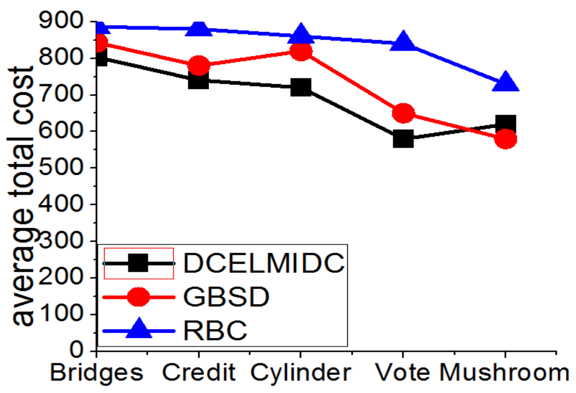

Figure 4. Average Cost of DCELMIDC, RBC and GBSD

From Figure 4, it can be seen that average total cost of DCELMIDC is smaller than RBC and GBSD, so it has better performance than the others.

In order to verify the influence of the incomplete ration, the incomplete ratio is set from 10% to 50%. The Classification accuracies with incomplete ratio in Cylinder and Vote are as follows.

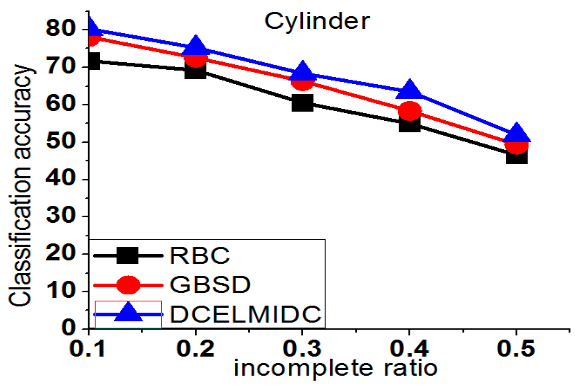

Figure 5. Classification Accuracy with Various Incomplete Ratios in Cylinder

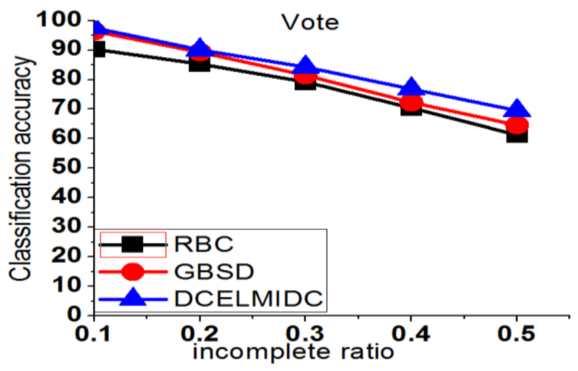

Figure 6. Classification Accuracy with Various Incomplete Ratios in Zoo

From Figure 5 and Figure 6, it can be seen that the Classification accuracy with incomplete ratio of DCELMIDC in Cylinder and Vote is higher than RBC and GBSD, so DCELMIDC has a better performance when it classifies incomplete data with different ratio.

6. Conclusion

There are more and more incomplete data in the world applications, and this paper proposes the Dynamic cost-sensitive extreme learning machine for Classification of incomplete Data, which overcomes the limitations of the stationary cost. We dispose incomplete data by the deep imputation network model, and offer the cost-sensitive extreme learning machine. This paper introduces dynamic misclassification and test cost, and gives chromosome coding and an evaluation method of the optimal cost. The Dynamic cost-sensitive extreme learning machine for incomplete data can determine the optimal cost by searching the cost space. The results of the experiments demonstrate that DCELMIDC comprises effective classification algorithms for incomplete data, which can dispose the incomplete data with various missing ratios.

Acknowledgement

This work was supported by the science and technology projects of the Shandong province universities (No. J13LN53, No. J14LN65, No. J15LN80). This work has been partially supported by grants from the research fund of heze university (No. XY14KJ08, XY15kj09).

References

[1] H. C. Lin and C. T. Su, "A selective Bayes classifier with meta-heuristics for incomplete data", Neurocomputing, Elsevier, vol. 106, **(2013)**, pp. 95-102.
[2] G. C. Patrick, W. G. B. Jerzy and W. Rzasa, "Mining incomplete data with singleton, subset and concept probabilistic approximations", Information Sciences, Springer International Publishing, vol. 280, **(2014)**, pp. 68-384.
[3] T. P. Hong, L. H. Tseng and B. C. Chien, "Mining from incomplete quantitative data by fuzzy rough sets", Expert Systems with Applications, Elsevier, vol. 37, no 3, **(2010)**, pp. 2644-2653.
[4] D. S. Kwak and K.J. Kim, "A data mining approach considering missing values for the optimization of semiconductor-manufacturing processes", Expert Systems with Applications, Elsevier, vol. 39, **(2012)**, no 3, **(2010)**, pp. 2590-2596.
[5] J. C. Xin, Z. Q. Wang, L. X. Qu and G. R. Wang, "Elastic extreme learning machine for big data classification", Neurocomputing, Elsevier, vol. 149, no. 1, **(2015)**, pp. 464-471.
[6] Z. H. Wang, Y. H. Zhao, G. R. Wang, Y. Li and X. Wang, "On extending extreme learning machine to non-redundant synergy pattern based graph classification", Neurocomputing, Elsevier, vol. 149, no. 1, **(2015)**, pp. 330-339.
[7] F. L. Cao, B. Liu and D. S. Park, "Image classification based on effective extreme learning machine", Neurocomputing, Elsevier, vol. 102, **(2013)**, pp. 90-97
[8] R. Moreno, F. Corona, A. Lendasse, M. Graña and L. S. Galvão, "Extreme learning machines for soybean classification in remote sensing hyperspectral images", Neurocomputing, Elsevier, vol. 128, **(2014)**, pp. 207-216.
[9] S. Suresh, S. Saraswathi and N. Sundararajan, "Performance enhancement of extreme learning machine for multi-category sparse data classification problems", Engineering Applications of Artificial Intelligence, Neurocomputing, Elsevier, vol. 23, no. 7, **(2010)**, pp. 1149-1157.
[10] V. López, S. del Río, J. M. Benítez and F. Herrera, "Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data", Fuzzy Sets and Systems, Elsevier, vol. 258, **(2015)**, pp. 5-38.
[11] B. Krawczyk, M. Woźniak and G. Schaefer, "Cost-sensitive decision tree ensembles for effective imbalanced classification", Applied Soft Computing, vol. 14, **(2014)**, pp. 554-562.
[12] L. X. Jiang, C. Q. Li and S. S. Wang, "Cost-sensitive Bayesian network classifiers", Pattern Recognition Letters, Elsevier, vol. 45, **(2014)**, pp. 211-216
[13] Z. Zidelmal, A. Amirou, D. O. Abdeslam and J. Merckle, "ECG beat classification using a cost sensitive classifier", Computer Methods and Programs in Biomedicine, Elsevier, vol. 111, no. 3, **(2013)**, pp. 570-577
[14] Y. W. Huang, "Research on dynamic cost-sensitive Decision Tree for Mining Uncertain Data Based on the Genetic Algorithm", International Journal of Database Theory and Application. SERSC, vol. 7, no. 5, **(2014)**, pp. 201-210.
[15] K. P. Malay, "A Fast K-Means Algorithm Using Cluster Shifting to Produce Compact and Separate Clusters", International Journal of Engineering Transactions A: Basics, vol. 28, no. 1, **(2015)**, pp. 35-43.
[16] Z. Shaeiri and R. Ghaderi, "Modification of the fast global k-means using a fuzzy relation with application in microarray data analysis", International Journal of Engineering-Transactions C: Aspects, vol. 25, no. 4, **(2012)**, pp. 283-292.
[17] M. Ramoni amd P. Sebastiani, "Robust Bayes classifiers", Artificial Intelligence, Elsevier, **(2001)**, pp. 209-226.
[18] J. N. Chen, H. K. Huang, L. Xu and Y. Chuanhua, "Constructing Hybrid Selective Classifiers for Incomplete Data with Gain-Ratio", Journal of Beijing Jiao tong University, **(2009)**, pp. 117-121.

Authors

Fuxian Huang, was born in 1972 at Chengwu, and received the Master of Engineering in Computer Science from the “Shandong University of Science and Technology” in 2005. He is now a professor at the Department of Computer and Information Engineering, Heze University. His research interests include the data-mining, intelligence Calculation.

z

ChunYing Liu, received the Master of Engineering in Computer Science from the “Shandong University of Science and Technology” in 2008. She is now a lecturer at the Department of Computer and Information Engineering, Heze University. His research interests include the computer network, data-mining, Intelligence Calculation.

Yuwen Huang, was born in 1978 at Shanxian, and received the Master of Engineering in Computer Science from the “Guangxi Normal University” in 2009. She is now a lecturer at the Department of Computer and Information Engineering, Heze University. His research interests include the data-mining, intelligence Calculation.

Jijiang Yu, received the Master of Engineering in Computer Science from the “Shan Dong University of Science and Technology” in 2009. He is now an associate professor at the Department of Computer and Information Engineering, Heze University. His research interests include the computer network, Intelligence calculation, data-mining.