Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 8056253, 9 pages
http://dx.doi.org/10.1155/2016/8056253

Research Article

Applying Cost-Sensitive Extreme Learning Machine and Dissimilarity Integration to Gene Expression Data Classification

Yanqiu Liu,[1] Huijuan Lu,[1] Ke Yan,[1] Haixia Xia,[2] and Chunlin An[1]

[1] *College of Information Engineering, China Jiliang University, Hangzhou 310018, China*
[2] *College of Informatics, Zhejiang Sci-Tech University, Hangzhou 310014, China*

Correspondence should be addressed to Huijuan Lu; hjlu@cjlu.edu.cn and Ke Yan; yanke@cjlu.edu.cn

Received 1 April 2016; Revised 15 July 2016; Accepted 26 July 2016

Academic Editor: J. Alfredo Hernández-Pérez

Embedding cost-sensitive factors into the classifiers increases the classification stability and reduces the classification costs for classifying high-scale, redundant, and imbalanced datasets, such as the gene expression data. In this study, we extend our previous work, that is, Dissimilar ELM (D-ELM), by introducing misclassification costs into the classifier. We name the proposed algorithm as the cost-sensitive D-ELM (CS-D-ELM). Furthermore, we embed rejection cost into the CS-D-ELM to increase the classification stability of the proposed algorithm. Experimental results show that the rejection cost embedded CS-D-ELM algorithm effectively reduces the average and overall cost of the classification process, while the classification accuracy still remains competitive. The proposed method can be extended to classification problems of other redundant and imbalanced data.

1. Introduction

With the appearance of gene chips, the classification methodology for gene expression data is developed into molecule phase [1]. The classification of gene expression data represents a crucial component in next generation cancer diagnosis technology [2]. For a particular tumor tissue with a series of known features, scientists believe that the classification of the gene array tells important information for identifying the tumor type and consequently influences the treatment plan [3–5]. However, the gene expression data on the other hand is known as large-scale, highly redundant, and imbalanced data, usually with relatively small sample size. Specifically, the number of features can be a hundred times larger than the number of samples [6]. This particular property of the gene expression data makes most of the traditional classifiers, such as extreme learning machine (ELM) [7], support vector machine (SVM), and multilayer neural networks, face difficulty in producing accurate and stable classification result. In 2012, we presented the integrated algorithm of Dissimilar ELM (D-ELM) by selective elimination of ELM based on V-ELM, which provided stable classification results compared to individual ELMs [8, 9].

Besides the accuracy, classification cost is another important aspect in performance evaluation for classification problems. In the cancer diagnosis progress, the cost of classifying a patient with cancer into negative class (false-negative) is much higher than that of classifying a patient without cancer into positive class (false-positive) [10]. Both false-negative and false-positive cases are recognized as misclassification cases. However, the costs of false-negative can be human lives due to the wrong medical treatments. Besides the misclassification cost, in recent years, the rejection cost also catches people's attention for cost-sensitive classifier development [11]. By considering misclassification and rejection cost, the classifiers become more stable and reliable.

In this study, aiming at extending the D-ELM to increase its classification stability, we embedded misclassification costs into D-ELM and named the proposed extension as CS-D-ELM. Furthermore, we embed the rejection costs into the CS-D-ELM to increase the classification stability of the proposed algorithm. The rejection cost embedded CS-D-ELM algorithm achieves the minimum classification cost with competitive classification accuracy. We validated CS-D-ELM by several commonly used gene expression datasets and compared the experimental results of using D-ELM, CS-ELM, and CS-SVM. The results show that the CS-D-ELM and rejection cost embedded CS-D-ELM both effectively reduce the overall misclassification costs and consequently enhance the classification reliability.

The rest of the paper is organized as follows. Related works, such as ELM, extensions of ELM, and cost-sensitive classifiers, are introduced in Section 2. In Section 3, the proposed algorithm is described in detail. The original D-ELM algorithm is extended by embedding misclassification costs and rejection costs. The experimental results are shown in Section 4. Conclusion, limitation, and future works are stated in Section 5.

2. Related Work

2.1. Extreme Learning Machine (ELM). In 2004, Huang et al. first proposed the extreme learning machine as a single-hidden layer feedforward neural network (SLFN) [12–14]. The most famous advantage of ELM is the one-step training process, which results in much faster learning speed compared with traditional machine learning techniques, such as multilayer neural networks or support vector machine (SVM). The SLFN can also be applied to other research fields [15]. However, problems arise while the classification accuracy performance of a single ELM is not stable. Integrated ELM algorithms are developed to solve the above problem. Wang et al. [16] proposed an upper integral network with extreme learning mechanism. The upper integral extracts the maximum potential of efficiency for a group of features with interaction. Lan et al. [17] presented an enhanced integration algorithm with more stable performance and higher classification accuracy for Ensemble of Online Sequential ELM (EOS-ELM). Tian et al. [18, 19] used the Bagging Integrated Model and the modified AdaBoost RT to modify the conventional ELM, respectively. Lu et al. [20] proposed several algorithms to reduce the computational cost of the Moore-Penrose inverse matrices for ELM. Zhang et al. [21] introduced an incremental ELM which combines the deep feature extracting ability of Deep Learning Networks with the feature mapping ability of the ELM. Cao et al. [22] presented the majority Voting ELM (V-ELM), and this algorithm is widely used in various fields. Lu et al. [8, 9] presented the integrated algorithms of Dissimilar ELM (D-ELM) which is more adaptive for different individual ELMs compared with [22].

2.2. Cost-Sensitive Classifiers. In most integrated algorithms, the possibilities of samples belonging to given classes are calculated before judging the class labels of samples. However, if there are two or more probabilities which are equal or close to each other, misclassification is likely to happen. Therefore, the misclassification cost issue is studied to improve the classification performance of integrated algorithms. Foggia et al. [23] proposed a method to calculate the misclassification value and false rejection value of multiexpert systems based on Bayesian decision rules. Experimental results showed that their method was optimal. In 2003, Zadrozny et al. [24] introduced cost-sensitive factors into machine learning techniques, which further reduced the classification costs. In 2010, Masnadi-Shirazi and Vasconcelos [25] introduced both the misclassification costs and rejection costs into SVM, which improved the performance of cost-sensitive SVM algorithms. In 2011, Fu [26] proposed a cost-sensitive classification algorithm named Cost-MCP Boost for multiclassification problems. The Cost-MCP Boost algorithm solved the cost merge problem while the multiclass cost-sensitive classifications were converted into two-class cost-sensitive classifications; and the classification results were determined by the classes with smaller misclassification costs. In 2013, Cao et al. [27] proposed an optimized cost-sensitive SVM to deal with the imbalanced data learning problem.

Embedding classification costs into the ELM improves both the classification accuracy and the stability of the classifier [28]. Lu et al. [29–31] proposed cost-sensitive ELM for gene expression data classification. Experimental results showed that the misclassification cost dropped drastically and the classification accuracy was raised. Zong et al. [32] and Mirza et al. [33] utilized a weighted ELM to deal with imbalanced data. By assigning different weights to samples following user instructions, the weighted ELM can be generalized to cost-sensitive ELM. Riccardi et al. [34] worked on a cost-sensitive AdaBoost algorithm which is based on ELM. The cost-sensitive ELM is used for ordinal regression, which turns out to produce competitive results. Most recently, Fu et al. [35] showed some experimental results on the stability and generalization of ELM. The study provides some useful guidelines to ensemble ELM with cost-sensitive factors to produce more stable classification results. Wang et al. [36] indicated that samples with higher fuzziness outputted by the classifier mean a bigger risk of misclassification. They proposed a fuzziness category based divide-and-conquer strategy to promote the classifier performance.

3. Cost-Sensitive Dissimilar Extreme Machine (CS-D-ELM)

The ultimate goal of this study is to minimize the conditional risk:

$$\arg\min R(i \mid x) = \arg\min \sum_{j} P(j \mid x) \cdot C(i, j), \tag{1}$$

where $R(i \mid x)$ is the conditional probability risk that the sample x is classified into the class I. $P(j \mid x)$ is the probability that sample x belongs to the class j. $C(i, j)$ is the risk that a sample belonging to class j is misclassified to class i, where i and j belong to the set $\{C_1, C_2, \dots, C_m\}$ and m is the class number.

3.1. The General Form of D-ELM. The D-ELM is an improved algorithm for majority Voting ELM [37]. It selects the most suitable ELM individuals after a training process in order to improve the consistency in the voting procedure and therefore increase the classification accuracy. For example, suppose there are N ELMs and M training samples available for initialization. We construct a dissimilarity matrix to eliminate inappropriate ELM individuals. The remaining

ELMs are considered in consistent form and are able to provide more stable classification results. The dissimilarity matrix is defined by inconsistency degree of outputs. If the ith ELM and the jth ELM have equal judgement results of f_{ik} and f_{jk} for the sample k, respectively, we mark $\mathrm{Dif}(f_{ik}, f_{jk}) = 0$, ($i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, N$; $K = 1, 2, \ldots, M$). Otherwise, we mark $\mathrm{Dif}(f_{ik}, f_{jk}) = 1$. Suppose that $\mathrm{Div}_{i,j} = \sum_{k=1}^{N} \mathrm{Dif}(f_{ik}, f_{jk})$ denotes the difference between the ith ELM and the jth ELM. The dissimilarity matrix is expressed as

$$\mathrm{Div} = \begin{bmatrix} \mathrm{Div}_{1,1} & \cdots & \mathrm{Div}_{1,j} & \cdots & \mathrm{Div}_{1,N} \\ \vdots & & \vdots & & \vdots \\ \mathrm{Div}_{i,1} & \cdots & \mathrm{Div}_{i,j} & \cdots & \mathrm{Div}_{i,N} \\ \vdots & & \vdots & & \vdots \\ \mathrm{Div}_{N,1} & \cdots & \mathrm{Div}_{N,j} & \cdots & \mathrm{Div}_{N,N} \end{bmatrix}. \tag{2}$$

Obviously, Div is a matrix with zeros for all diagonal elements.

η_i denotes the dissimilarity between the ith ELM and the rest of the ELMs. It is defined as

$$\eta_i = \sum_{j=1}^{N} \mathrm{Div}_{i,j}. \tag{3}$$

The average classification accuracy is denoted as $\overline{p}$. The ELM with smaller η value is eliminated under the condition of $0 < \overline{p} \leq 0.5$. And the ELM with bigger η value is eliminated under the condition of $0.5 < \overline{p} < 1$. The remaining ELMs are selected to proceed to the voting procedure.

The overall D-ELM algorithm can be divided into three parts. First, N independent ELMs are trained using given data; and a number of ELMs are eliminated according to dissimilarity theory. Second, the remaining K ELMs are trained again using the same hidden layer node number and activation function. For each independent ELM input layer, hidden layer weights and bias are randomly generated and unrelated. Last, for each testing sample tx, the K independent ELMs can maximally predict K individual classification results. An initial empty vector $(W_{k,tx}(c_1), W_{k,tx}(c_2), \ldots, W_{k,tx}(c_m))$ (m is the number of classes) is used to store the classification results of tx for the K ELMs. For example, for the lth ($l \in [1, \ldots, K]$) ELM classifier, if the classification result of tx is i ($i \in \{C_1, C_2, \ldots, C_m\}$), then the following operations are carried out:

$$W_{K,tx}(i) = W_{K,tx}(i) + 1. \tag{4}$$

The final vector $W_{k,tx}$ is obtained after all K ELMs are processed. We get the probability for each class in the classification result:

$$P(i \mid tx) = \frac{W_{K,tx}(i)}{K}, \quad i \in \{c_1, c_2, \ldots, c_m\}. \tag{5}$$

After calculating the conditional probability of the test sample tx by D-ELM, if tx is classified into sth class correctly, the probability that tx belongs to the sth class is bigger than the probability that it belongs to other classes; that is, there is an inequality:

$$P(s \mid tx) \geq \max \{P(i \mid tx)\}_{i \in [c_1, c_2, \ldots, c_m]}. \tag{6}$$

For example, in a two-class classification, the probabilities that a testing sample tx belongs to the positive class and negative class are $P(p \mid tx) = W_{k,tx}(p)/K$ and $P(n \mid tx) = W_{k,tx}(n)/K$, respectively.

3.2. Embedding Misclassification Costs into D-ELM.

For each test sample tx, it is not enough to only know the probability $P(j \mid tx)$ ($j \in \{C_1, C_2, \ldots, C_m\}$) of tx. When the cost is unequal, even if the inequality (6) is satisfied, we cannot determine the class label s of tx. Therefore, in this section, the asymmetric misclassification costs are embedded in order to improve D-ELM to CS-D-ELM.

Suppose the probability $P(j \mid tx)$ is calculated by the D-ELM method in Section 3.1; the class label of tx is determined by taking the least cost that tx belongs to a class i. The following equation is derived from (1):

$$\begin{aligned} \overline{ty} &= \arg\min_i \{R(i \mid tx)\} \\ &= \arg\min_i \left\{ \sum_j P(j \mid x) \cdot C(i, j) \right\}. \end{aligned} \tag{7}$$

All class labels of test samples are recalculated according to the principle of minimizing the average misclassification costs. Let $\overline{ty}$ be the real class label of sample tx after integrating the misclassification cost information of tx. After embedding the misclassification cost into the D-ELM, the classification results are as follows:

$$\begin{aligned} \overline{ty} &= \begin{bmatrix} \overline{ty}_1 \\ \vdots \\ \overline{ty}_{\widetilde{N}} \end{bmatrix} = \begin{bmatrix} \arg\min_i \{R(i \mid tx_1)\} \\ \vdots \\ \arg\min_i \{R(i \mid tx_{\widetilde{N}})\} \end{bmatrix} \\ &= \begin{bmatrix} \arg\min_i \sum_j \{P(j \mid tx_1)\} \cdot C\{i, j\} \\ \vdots \\ \arg\min_i \sum_j \{P(j \mid tx_{\widetilde{N}})\} \cdot C\{i, j\} \end{bmatrix}, \end{aligned} \tag{8}$$

where $P(j \mid tx) = R_{K,tx}(j)/K$ ($j \in \{C_1, C_2, \ldots, C_m\}$) is the probability calculated by D-ELM in Section 3.1.

3.3. The CS-D-ELM Algorithm.

A general form of CS-D-ELM algorithm can be described as follows:

(1) Set initial values for all N ELMs.

(2) Randomly generate the input layer parameters (a_j^i, b_j^i), $j = 1, \ldots, L$ (L is the number of nodes in hidden layer), of the ith ELM.

(3) Calculate the hidden layer output matrix of the ith ELM.

Table 1: Datasets.

Datasets	Sample number	Feature number	Class distribution	
			Class name	Sample number
Diabetes	97	8	Relapse	46
			Nonrelapse	51
Heart	270	13	Negative	150
			Positive	120
Colon	62	52	Negative	19
			Positive	43
Mushroom	263	43	Negative	111
			Positive	152
Protein	334	73	Negative	215
			Positive	119
Leukemia	72	7129	ALL	24
			MLL	20
			AML	28

(4) Calculate the ith output weights, that is, a target output matrix.

(5) Process the N ELMs using dissimilarity elimination, assuming that k ELMs are left after elimination.

(6) For the test samples, predict each class of tx by using the remaining k ELMs. Assuming tx belongs to class j, where $j \in \{C_1, C_2, \ldots, C_m\}$, we have $W_{K,tx}(j) = W_{K,tx}(j) + 1$.

(7) Calculate the probability that the test set belongs to each class $P(j \mid tx) = R_{K,tx}(j)/K$.

(8) Use (8) to calculate the true class labels.

(9) End.

3.4. Embedding Rejection Costs into CS-D-ELM. The samples with low classification reliability are more likely to be misclassified. In order to reduce the high cost of misclassification, rejection options are embedded into CS-D-ELM to prevent automatic classification of samples with low classification reliabilities. There are three kinds of rejection costs:

(1) The costs that require further analysis for unclassified samples.

(2) Loss of samples because of the rejection decision.

(3) Both circumstances above.

The rejection cost is defined as follows. For a given small positive number δ (rejection threshold) and any test samples tx with the following equations being true:

$$\begin{gathered} R(s \mid tx) < \max\{R(i \mid tx)\}_{i \in [c_1, \ldots, c_m], i \neq s} \\ f(tx) = \min\{R(i \mid tx)\}_{i \in [c_1, \ldots, c_m], i \neq s} - R(s \mid tx), \end{gathered} \tag{9}$$

if $f(tx) \geq \delta$, the test sample is classified into the sth class. If $f(tx) < \delta$, the test sample is processed by rejection treatment.

For the two-class classification problems, embedded misclassification costs and rejection costs, given a test sample

$$tx = \{(tx_1, ty_1), \ldots, (tx_i, ty_i), \ldots, (tx_{\widetilde{N}}, ty_{\widetilde{N}})\}, \tag{10}$$

where $tx_i \in R^n$ and $ty_i \in \{n, p\}$, $i = 1, \ldots, \widetilde{N}$; and the cost matrix is:

$$C = \{C(p, n), C(n, p), C(0, n), C(0, p)\}, \tag{11}$$

where $C(p, n)$ and $C(n, p)$ are misclassification costs and $C(0, n)$ and $C(0, p)$ are rejection costs. $P(0 \mid x)$ (rejection rate), $P(n \mid x)$, and $P(p \mid x)$ can be calculated on the basis of the rejection threshold δ. Then, the test sample is evaluated by calculating minimum average misclassified cost; that is,

$$\begin{aligned} \overline{ty} &= \arg\min_i \{R(i \mid tx)\} \\ &= \arg\min_i \sum_j P(i \mid x)(i \mid x) C(j, i), \end{aligned} \tag{12}$$

where $i, j \in \{0, n, p\}$. The rejection threshold δ is determined on a case-by-case basis which depends on the specific dataset. An example of a particular δ value calculation can be found in Section 4.3.

4. Experiments and Analysis

4.1. Experimental Datasets. For the performance evaluation of the CS-D-ELM algorithm, we perform experiments on six reduced gene expression datasets, that is, Diabetes, Heart, Colon, Mushrooms, Protein, and Leukemia. All datasets are preprocessed by feature selection methods to reduce the data size. The Diabetes dataset includes 97 gene samples and each sample includes 8 gene expression features. The Heart dataset consists of 270 samples with 13 features. The Colon dataset consists of 62 samples with 52 features. The Protein dataset contains 334 samples with 73 gene features. The Mushroom dataset has 263 samples with 43 gene features. The Leukemia dataset includes 72 acute leukemia samples, and the expression data of each sample contains 7129 genes. The first five datasets belong to the two-class classification problem, while the Leukemia dataset belongs to the multiclass classification problem. Table 1 concludes the information of all the six datasets.

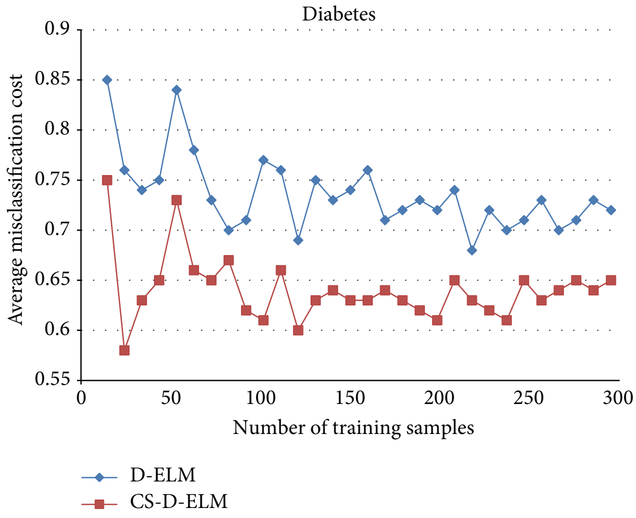

Figure 1: Average misclassification costs for Diabetes dataset.

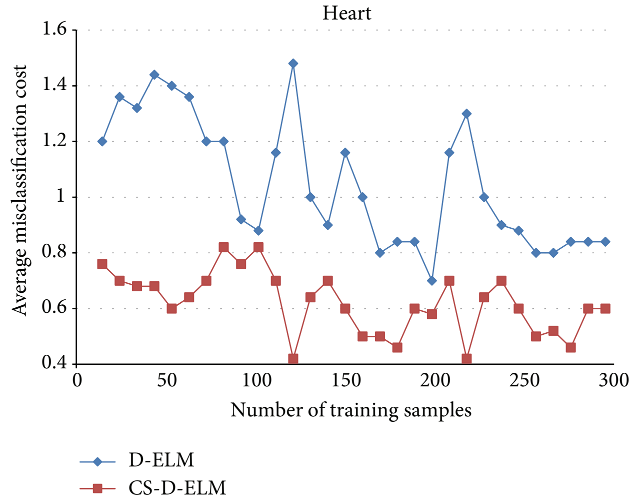

Figure 2: Average misclassification costs for Heart dataset.

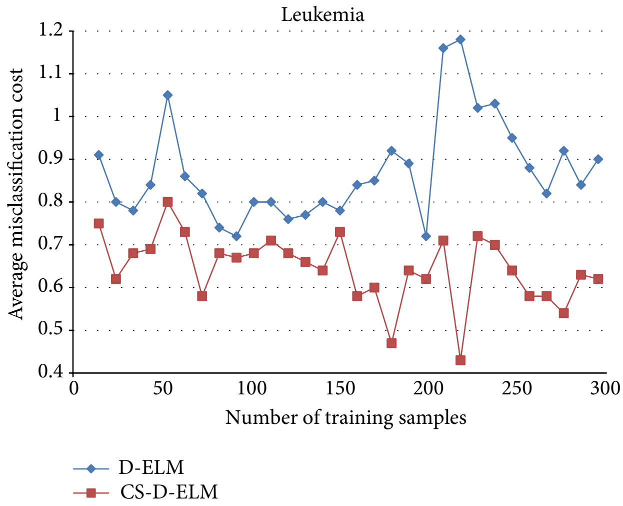

Figure 3: Average misclassification costs for Leukemia dataset.

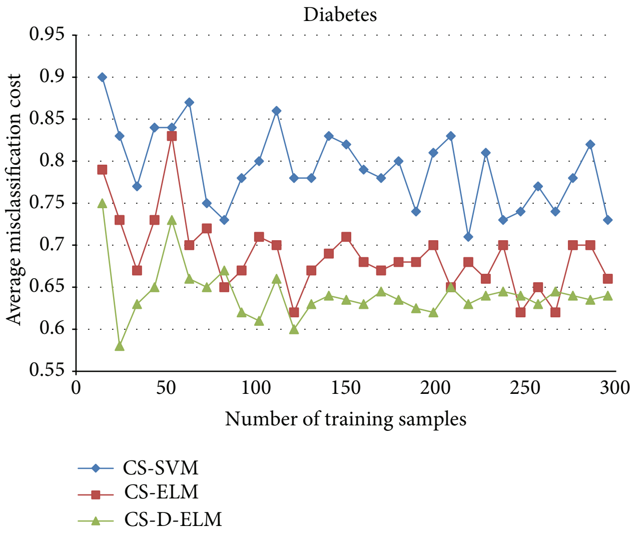

Figure 4: Comparison of average classification costs for Diabetes dataset.

4.2. Experimental Results of CS-D-ELM. The misclassification costs of false-positive and false-negative are set as $C(1,-1) = 1$ and $C(-1,1) = 5$. The experiments are repeated 30 times in each dataset; and average result of the 30 times is recorded as the final experimental results. In every experiment, 300 samples are randomly selected as training set; and the remaining samples are used for testing purpose.

Figures 1, 2, and 3 represent the average misclassification costs of D-ELM and CS-D-ELM on Diabetes, Heart, and Leukemia datasets, respectively. The average misclassification costs based on CS-D-ELM are lower than those based on D-ELM. While the number of training samples increases, the average misclassification cost decreases.

To further verify the effectiveness of CS-D-ELM, we compared CS-D-ELM with cost-sensitive ELM (CS-ELM) and mature cost-sensitive support vector machine (CS-SVM) [27]. Figures 4, 5, and 6 represent the average misclassification costs of CS-ELM, CS-SVM, and CS-D-ELM on Diabetes, Heart, and Leukemia datasets, respectively. The average misclassification costs produced by CS-D-ELM are lower than those based on CS-ELM and CS-SVM.

4.3. Experimental Results of CS-D-ELM with Embedded Rejection Costs. Before embedding the rejection costs into the CS-D-ELM, an important preprocessing step must be performed, which is the rejection threshold determination. Taking the Heart dataset as an example, on the basis of the invariant cost matrix, we set the rejection costs to $C(0,1) = C(1,0) = 0.2$. Randomly selected 300 samples are treated as the training sample set; and the remaining samples are treated as testing sample set. Again, 30 independent experiments are performed with average misclassification costs recorded.

Figure 7 shows the relationship between the rejection threshold and the average misclassification costs. The average misclassification cost is minimized while rejection threshold

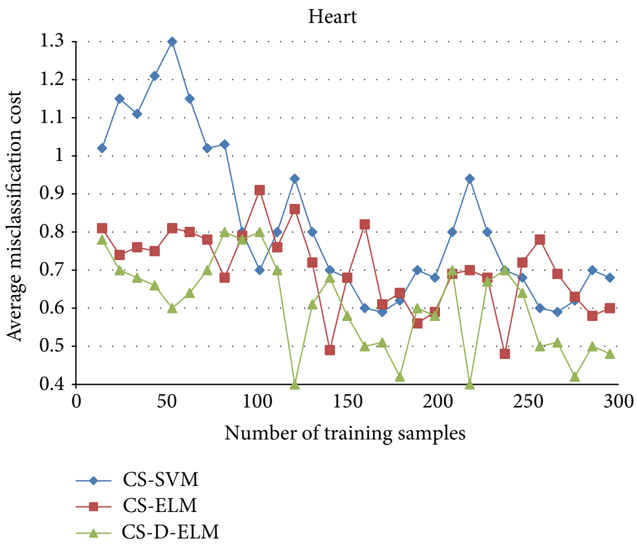

Figure 5: Comparison of average classification costs for Heart dataset.

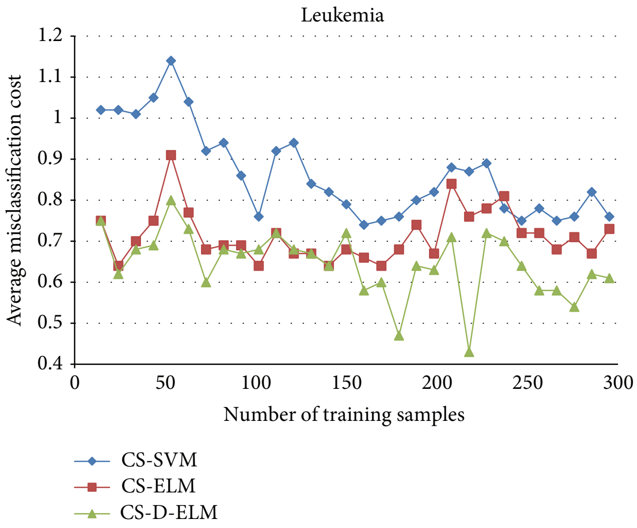

Figure 6: Comparison of average classification costs for Leukemia dataset.

δ equals 0.04. Using $\delta = 0.04$ in the CS-D-ELM method, we get the results in Figure 8. In Figure 8, D1, D2, and D3 denote the results under the condition of not considering misclassification costs and embedding misclassification costs and rejection costs, respectively. The embedding of rejection costs reduces the average misclassification cost greatly. Considering that the D-ELM algorithm includes a voting process of ELMs to filter dissimilarity, the time performance of CS-D-ELM is 3-4 times slower than each individual ELM. A speed comparison between ELM, D-ELM, CS-D-ELM, and rejection cost embedded CS-D-ELM is shown in Table 2. We use a 32 Core@ 2.26 Hz Dell server machine with 128 G RAM. The voting process is conducted with 100 ELM individuals. Under the assumption that time is not the top priority in this study, the following conclusion can be drawn from this experiment: while the sample misclassification costs are not equal, the average misclassification can be effectively reduced by embedding rejection costs into CS-D-ELM.

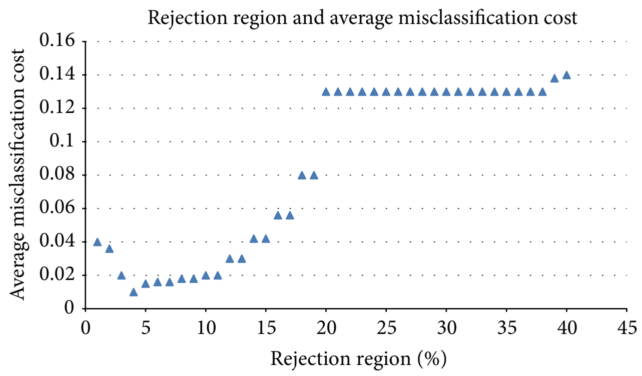

Figure 7: Relationship between the rejection threshold and the average misclassification costs.

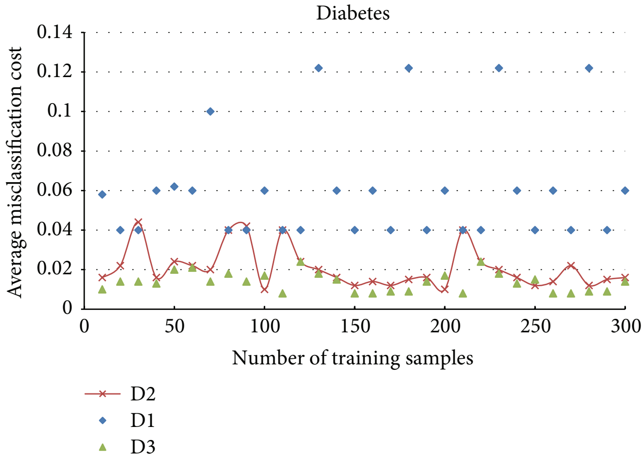

Figure 8: Comparison of embedding different cost-sensitive factors.

In order to further verify the universal applicability of CS-D-ELM on gene expression data, we compare D-ELM, CS-D-ELM, and CS-D-ELM with embedded rejection costs on 4 datasets, namely, Leukemia, Colon, Mushroom, and Protein. The classification accuracies on the positive class and the negative class are denoted as NC and PC. In each dataset, the average value of 30 experiments is used as the experimental results. The NC values, PC values, G-means values, and the average misclassification costs when using the three algorithms are shown in Table 3.

As results shown in Tables 3 and 4, the G-means values are improved in CS-D-ELM and CS-D-ELM with embedded reject costs, and the effect of CS-D-ELM with embedded rejection costs is better. The average misclassification costs of CS-D-ELM are smaller than those of D-ELM in all datasets, indicating that CS-D-ELM can effectively reduce the misclassification costs in the classification process. The average misclassification costs of CS-D-ELM with embedded rejection costs are lower than those of the standard CS-D-ELM in all datasets. Again, the same conclusion can be drawn:

Table 2: Running time comparison between ELM, D-ELM, CS-D-ELM, and rejection cost embedded CS-D-ELM.

Dataset	Average running time for different algorithms (recorded in sec.)			
	ELM	D-ELM	CS-D-ELM	Rejection cost embedded CS-D-ELM
Diabetes	0.4312	1.4536	1.5330	1.5470
Heart	0.6670	1.8579	1.9353	1.9455
Colon	0.5183	1.6743	1.7458	1.7561
Mushroom	0.7214	1.8654	1.9583	1.9836
Protein	0.7551	2.0836	2.1349	2.2655
Leukemia	1.2319	3.9593	4.0346	4.1692

Table 3: NC value and PC value of D-ELM, CS-D-ELM, and embedded rejection costs into S-D-ELM.

Dataset	NC value			PC value		
	D-ELM	CS-D-LM	Rejection CS-D-LM	D-ELM	CS-D-LM	Rejection CS-D-LM
Leukemia	0.3815	0.4714	0.5874	0.9561	0.8626	0.8215
Colon	0.4132	0.5127	0.6322	0.9722	0.9152	0.8464
Mushroom	0.4325	0.5423	0.6929	1.0000	0.9605	0.9313
Protein	0.3237	0.4621	0.5433	0.9433	0.8956	0.7751

Table 4: Experiment results of G-means value and average misclassification costs.

Dataset	G-means value			Average misclassification costs		
	D-ELM	CS-D-LM	Rejection CS-D-LM	D-ELM	CS-D-LM	Rejection CS-D-LM
Leukemia	0.6416	0.6508	0.6714	0.4271	0.3522	0.2543
Colon	0.6852	0.7203	0.7313	0.3814	0.2232	0.2123
Mushroom	0.7125	0.7877	0.7922	0.1102	0.0755	0.0411
Protein	0.5333	0.7122	0.7328	0.4843	0.3812	0.2043

when the misclassification costs of samples are not equal, rejection cost embedded CS-D-ELM can effectively reduce the average misclassification costs.

5. Conclusion and Discussion

The traditional classification algorithms are all based on the classification accuracy. When the misclassification costs are not equal, they cannot achieve the minimum average misclassification cost requirements in cost-sensitive classification process. This paper first reconstructs the classification results by introducing the probability estimation and misclassification costs into the classification process and proposes the cost-sensitive D-ELM algorithm which is called CS-D-ELM. Furthermore, by embedding the rejection costs, the average misclassification costs are further reduced.

For computational complexity evaluation, the time taken by the voting procedure is negligible as the training process of each ELM is the most costly part. The speed of the proposed algorithm depends on the number of parallelizable cores in the machine. In our case, a 32-core server machine is utilized to run the CS-D-ELM with 100 ELM individuals. The overall running time of the proposed algorithm is three to four times slower than each ELM individual. However, as the average running time for each ELM individual is generally less than one second, the overall running speed is still fast.

The embedding of misclassification costs and rejection costs is proved to be useful for classification cost reduction and accuracy improvement [8, 11, 30]. The way of embedding misclassification costs and rejection costs into the D-ELM can be employed for other cost-sensitive algorithms. As a future work, we are implementing a cost-sensitive rotational forest algorithm (CS-RoF) for gene expression data classification. Similar algorithms can also be extended for classification problems of other imbalanced datasets.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (no. 61272315, no. 61303146, no. 61402417, and no. 61572164), Zhejiang Provincial Natural Science Foundation (no. Y1110342, no. LY15F020037), and the National Security Bureau Project (Zhejiang-00062014AQ). It is also supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

References

[1] M. J. Heller, "DNA microarray technology: devices, systems, and applications," *Annual Review of Biomedical Engineering*, vol. 4, no. 1, pp. 129–153, 2002.

[2] S. Li and D. Li, "DNA microarray technology," in *DNA Microarray Technology and Data Analysis in Cancer Research*, pp. 1–9, World Scientific, Singapore, 2008.

[3] Z. Yu, H. Chen, J. You et al., "Double selection based semi-supervised clustering ensemble for tumor clustering from gene expression profiles," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 4, pp. 727–740, 2014.

[4] Z. Yu, H. Chen, J. You, G. Han, and L. Li, "Hybrid fuzzy cluster ensemble framework for tumor clustering from biomolecular data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 3, pp. 657–670, 2013.

[5] Z. Yu, L. Li, J. You, H.-S. Wong, and G. Han, "SC^3: triple spectral clustering-based consensus clustering framework for class discovery from cancer gene expression profiles," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 6, pp. 1751–1765, 2012.

[6] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of Bioinformatics and Computational Biology*, vol. 3, no. 2, pp. 185–205, 2005.

[7] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[8] H.-J. Lu, E.-H. Zheng, Y. Lu, X.-P. Ma, and J.-Y. Liu, "ELM-based gene expression classification with misclassification cost," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 525–531, 2014.

[9] H. Lu, *A Study of Tumor Classification Algorithms Using Gene Expression Data*, China University of Mining and Technology, Xuzhou, China, 2012 (Chinese).

[10] C. X. Ling and V. S. Sheng, "Cost-sensitive learning," in *Encyclopedia of Machine Learning*, pp. 231–235, Springer, New York, NY, USA, 2011.

[11] H. Lu, S. Wei, Z. Zhou, Y. Miao, and Y. Lu, "Regularised extreme learning machine with misclassification cost and rejection cost for gene expression data classification," *International Journal of Data Mining and Bioinformatics*, vol. 12, no. 3, pp. 294–312, 2015.

[12] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, IEEE, Budapest, Hungary, July 2004.

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[14] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[15] W. Xi-Zhao, S. Qing-Yan, M. Qing, and Z. Jun-Hai, "Architecture selection for networks trained with extreme learning machine using localized generalization error model," *Neurocomputing*, vol. 102, pp. 3–9, 2013.

[16] X. Wang, A. Chen, and H. Feng, "Upper integral network with extreme learning mechanism," *Neurocomputing*, vol. 74, no. 16, pp. 2520–2525, 2011.

[17] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, no. 13–15, pp. 3391–3395, 2009.

[18] H. Tian and B. Meng, "A new modeling method based on bagging ELM for day-ahead electricity price prediction," in *Proceedings of the IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA '10)*, pp. 1076–1079, Changsha, China, September 2010.

[19] H.-X. Tian and Z.-Z. Mao, "An ensemble ELM based on modified AdaBoost.RT algorithm for predicting the temperature of molten steel in ladle furnace," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 73–80, 2010.

[20] S. Lu, X. Wang, G. Zhang, and X. Zhou, "Effective algorithms of the Moore-Penrose inverse matrices for extreme learning machine," *Intelligent Data Analysis*, vol. 19, no. 4, pp. 743–760, 2015.

[21] J. Zhang, S. Ding, N. Zhang, and Z. Shi, "Incremental extreme learning machine based on deep feature embedded," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 1, pp. 111–120, 2016.

[22] J. Cao, Z. Lin, G.-B. Huang, and N. Liu, "Voting based extreme learning machine," *Information Sciences*, vol. 185, pp. 66–77, 2012.

[23] P. Foggia, C. Sansone, F. Tortorella, and M. Vento, "Multiclassification: reject criteria for the Bayesian combiner," *Pattern Recognition*, vol. 32, no. 8, pp. 1435–1447, 1999.

[24] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM '03)*, pp. 435–442, IEEE, Melbourne, Fla, USA, November 2003.

[25] H. Masnadi-Shirazi and N. Vasconcelos, "Risk minimization, probability elicitation, and cost-sensitive SVMs," in *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 759–766, Haifa, Israel, June 2010.

[26] Z. L. Fu, "Cost-sensitive AdaBoost algorithm for multi-class classification problems," *Acta Automatica Sinica*, vol. 37, no. 8, pp. 973–983, 2011.

[27] P. Cao, D. Zhao, and O. Zaiane, "An optimized cost-sensitive SVM for imbalanced data learning," in *Advances in Knowledge Discovery and Data Mining*, pp. 280–292, Springer, Berlin, Germany, 2013.

[28] Z. H. A. I. Yun, Y. A. N. G. Bingru, and Q. Wu, "Survey of mining imbalanced datasets," *Computer Science*, vol. 37, no. 10, pp. 27–32, 2010.

[29] H. J. Lu, C. L. An, X. P. Ma et al., "Disagreement measure based ensemble of extreme learning machine for gene expression data classification," *Chinese Journal of Computers*, vol. 36, no. 2, pp. 341–348, 2013.

[30] H.-J. Lu, C.-L. An, E.-H. Zheng, and Y. Lu, "Dissimilarity based ensemble of extreme learning machine for gene expression data classification," *Neurocomputing*, vol. 128, pp. 22–30, 2014.

[31] E. Zheng, C. Zhang, X. Liu, H. Lu, and J. Sun, "Cost-sensitive extreme learning machine," in *Advanced Data Mining and Applications: 9th International Conference, ADMA 2013, Hangzhou, China, December 14–16, 2013, Proceedings, Part II*, pp. 478–488, Springer, Berlin, Germany, 2013.

[32] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.

[33] B. Mirza, Z. Lin, and K.-A. Toh, "Weighted online sequential extreme learning machine for class imbalance learning," *Neural Processing Letters*, vol. 38, no. 3, pp. 465–486, 2013.

[34] A. Riccardi, F. Fernández-Navarro, and S. Carloni, "Cost-sensitive AdaBoost algorithm for ordinal regression based on extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1898–1909, 2014.

[35] A. Fu, C. Dong, and L. Wang, "An experimental study on stability and generalization of extreme learning machines," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 1, pp. 129–135, 2014.

[36] X.-Z. Wang, R. A. R. Ashfaq, and A.-M. Fu, "Fuzziness based sample categorization for classifier performance improvement," *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 3, pp. 1185–1196, 2015.

[37] H. Lu, W. Chen, X. Ma, and L. Yi, "A dataset splitting based neural network ensemble method for tumor classification," *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 5, pp. 167–173, 2012.