Study of Classification Methods
Based on Three Learning Criteria and Two Basis Functions

Jae Kyu Suhr

Abstract - This paper investigates several classification methods based on the three learning criteria and two basis functions. The three learning criteria are the least squares error, the total error rate, and the area under the receiver operating characteristic curve. The two basis functions are reduced multinomial model and single-hidden layer feedforward neural networks. In the experiment, five classification methods were evaluated by using the UCI database and suitable data normalization procedures for each basis function were discussed.

1. Introduction

Pattern classification has been a key component for decision making in many research fields. In pattern classification, empirical learning constitutes a major paradigm [1]. Under this paradigm, a classifier is designed to minimize a certain cost function (criterion) with a training set.

Least Squares Error (LSE) has been widely used as a cost function for empirical classifier learning. The reasons for the popularity of LSE are its simplicity, clear physical meaning, and tractability for analysis. The embedding of nonlinearities into linear models has widened its application.

Recently, two efficient basis functions were proposed: Reduced multinomial Model (RM) [2] and Extreme Learning Machine (ELM) [3]. RM uses a reduced version of full polynomial and ELM uses a Single-hidden Layer Feedforward Neural networks (SLFNs).

However, the main problem when using the LSE cost function is that it tries to minimize the fitting error rather than the classification error during the learning process.

Therefore, three main approaches have been adopted to overcome this drawback of the LSE cost function. These approaches are the discriminant approach (Fisher Discriminant Analysis and Generalized Discriminant Analysis), the structural approach (Support Vector Machine), and the classification-error approach. In the third approach, two cost functions were recently proposed. One is Total Error Rate (TER) [4,5] and the other is Area under the receiver operating characteristics curve (AUC) [6].

The main breakthrough of these three papers is a smooth approximate formulation for calculating TER and AUC. The step function used for the counting process is approximated by a quadratic function instead of a sigmoid function. This makes it possible to have closed-form solution.

In this paper, the above mentioned five methods (RM, ELM, TER-RM, TER-ELM, and AUC-RM) were investigated. Five two-class problems in the UCI database were used for the experiment since AUC-RM can only be applied to two-class problem so far.

The paper is organized as follows: In Section 2, the investigated classification methods are briefly described. In Section 3, experimental setup, data normalization issue, and evaluation results are presented. Finally, this paper is concluded with a summary in Section 4

2. Method description

2.1 LSE-based method

Least Squares Error (LSE) is a widely used cost function for empirical classifier learning due to the several reasons as mention in the previous section.

Consider an l-dimensional input $\mathbf{x}$ and the following parametric model adopting a basis expansion term:

$$g(\mathbf{a}, \mathbf{x}) = \sum_{i=0}^{K-1} \alpha_i p_i(\mathbf{x}) = \mathbf{p}(\mathbf{x}) \mathbf{a}$$

where $p_i(\mathbf{x})$ corresponds to the $i$th basis term of the row vector $\mathbf{p}(\mathbf{x})=[p_0(\mathbf{x}), p_1(\mathbf{x}), ..., p_m(\mathbf{x})]$, and $\mathbf{a}=[\alpha_0, \alpha_1, ..., \alpha_{K-1}]^T$ is a column parameter vector to be estimated.

When we have $m$ learning data pairs $\mathbf{p}(\mathbf{x})$ vector can be extended to $\mathbf{P} (\in \mathbb{R}^{m \times K})$ and a known label can be denoted by $\mathbf{y} (\in \mathbb{R}^{m})$. In this case, the LSE cost function becomes

$$J(\mathbf{a}) = \frac{1}{2} \| \mathbf{y} - \mathbf{P} \mathbf{a} \|^2 + \frac{b}{2} \| \mathbf{a} \|^2$$

where $b$ controls the weighting of the regularization factor. The estimated training output is given by $\hat{\mathbf{y}} = \mathbf{P} \hat{\mathbf{a}}$, where the solution for $\hat{\mathbf{a}}$, which minimizes $J$ is

$$\hat{\mathbf{a}} = (\mathbf{P}^T \mathbf{P} + b I)^{-1} \mathbf{P}^T \mathbf{y}$$

This LSE cost function has been widen its application by embedding nonlinearities into linear models. Recently, two efficient nonlinear basis functions were introduced. One is the Reduced multivariate Model (RM) [2] and the other is the Extreme Learning Machine (ELM) [3]. RM uses reduced version of full polynomial as a basis function which can be expressed as

$$\hat{f}_{RM}(\mathbf{a}, \mathbf{x}) = \alpha_0 + \sum_{j=1}^{J} \sum_{l=1}^{L} \alpha_{jl} x_j^l + \sum_{i=1}^{l-1} \alpha_{i1} x_i + x_2 + \cdots + x_l$$

$$+ \sum_{j=2}^{J} (\alpha_{jl} \cdot \mathbf{x})(x_1 + x_2 + \cdots + x_l)^{l-1}, \ I, r \geq 2.$$
$q$-dimensional output target vector. The standard SLFNs with $p$ hidden nodes and activation function $\phi$ can be modeled as

$$
\sum_{j=1}^{p} \beta_j \phi(x_i) = \sum_{j=1}^{p} \beta_j (w_j \cdot x_i + h_j) = g_{\eta}, \quad i = 1, \ldots, m
$$

(5)

where $w_j$ is the weight vector connecting the $j$th hidden node to the input nodes, $\beta_j$ is the weight vector connecting the $j$th hidden node to the output nodes, $h_j$ is the threshold of the $j$th hidden node, and $g_{\eta}$ is the $q$-dimensional network output. The $m$ equations above can be written more compactly as

$$
\mathbf{H} \mathbf{\beta} = \mathbf{y}
$$

(6)

where

$$
\mathbf{H} = \begin{bmatrix}
\phi(w_1 \cdot x_1 + h_1) & \cdots & \phi(w_p \cdot x_1 + h_p) \\
\vdots & \ddots & \vdots \\
\phi(w_1 \cdot x_m + h_1) & \cdots & \phi(w_p \cdot x_m + h_p)
\end{bmatrix}
$$

(7)

and $\mathbf{y}$ is a known label. Since $\mathbf{H}$ can be treated as a $\mathbf{P}$ matrix in Eq. (2), a least-squares solution can be obtained for the output weighing parameters as

$$
\mathbf{\hat{\beta}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}
$$

(8)

### 2.2 TER-based method

Although, LSE-based methods (for example RM and ELM) have been widely used, its limitation becomes apparent when high accuracy is required. It is mainly because the LSE cost function minimizes the fitting error rather than the classification error which is mostly desired to be minimized in classification tasks.

To overcome this drawback, the Total Error Rate (TER) cost function was proposed [4,5]. TER can be calculated by summing the false positive rate (FP) and the false negative rate (FN). FP and FN can be expressed as Eq. (9) by denoting the positive and negative examples of variables by the superscripts $+$ and $-$, respectively.

$$
\text{FP} = \frac{1}{m} \sum_{i=1}^{m} \text{L}(g(x_i) \geq \tau), \quad \text{FN} = \frac{1}{m} \sum_{i=1}^{m} \text{L}(g(x_i) \geq \tau)
$$

(9)

where $\text{L}$ is a zero-one step loss function. Let $g(x) = g(\mathbf{a}, \mathbf{x})$ with adjustable parameters $\mathbf{a}$ operating on the feature vector $\mathbf{x}$, then the TER can be written as

$$
\text{TER}(\mathbf{a}, \mathbf{x}^+, \mathbf{x}^-) = \frac{1}{m} \sum_{i=1}^{m} \text{L}(g(\mathbf{a}, \mathbf{x}_i^+) - \tau) + \frac{1}{m} \sum_{i=1}^{m} \text{L}(\tau - g(\mathbf{a}, \mathbf{x}_i^-))
$$

(10)

A natural choice to approximate a step function is a sigmoid function [7]. However, this leads two major problems. First, the formulation becomes nonlinear with respect to the learning parameters. This makes it difficult to find the optimal parameters. Second, the objective function can become ill-conditioned due to the much local plateaus. This causes the optimization procedure to be time-consuming.

Therefore, a quadratic function was proposed to approximate a step function. This is very effective since it results a closed-form classification-error-based solution. By using this approximation and $g(\mathbf{a}, \mathbf{x}) = p(\mathbf{x}) \mathbf{a}$, the TER in Eq. (10) can be rewritten as

$$
\text{TER}(\mathbf{a}, \mathbf{x}^+, \mathbf{x}^-) = \frac{1}{2m} \sum_{i=1}^{m} \left[ p(\mathbf{x}_i^+) \alpha - \eta \right]^2 + \frac{1}{2m} \sum_{i=1}^{m} \left[ \eta - p(\mathbf{x}_i^-) \alpha \right]^2
$$

(11)

where $\eta$ is a positive offset of a quadratic function. The LSE solution of Eq. (11) calculated by letting the derivative of $\text{TER}(\mathbf{a}, \mathbf{x}^+, \mathbf{x}^-)$ with respect to the parameter $\mathbf{a}$ be zero is

$$
\mathbf{a} = \left( b \mathbf{I} + \frac{1}{m} p^T \mathbf{P} + \frac{1}{m} \mathbf{P}^T \mathbf{P} \right)^{-1} \left( \frac{(\tau - \eta)}{m} \mathbf{P}, \frac{\tau + \eta}{m} \mathbf{P}^T \right)
$$

(12)

where $\mathbf{I}$ is an identity matrix of $K \times K$ size. In a more compact matrix form, Eq. (12) can be written as

$$
\mathbf{a} = \left( b \mathbf{I} + \frac{1}{m} \mathbf{P}^T \mathbf{P} + \frac{1}{m} \mathbf{P} \mathbf{P}^T \right)^{-1} \left( \frac{(\tau - \eta)}{m} \mathbf{P} \mathbf{1}, \frac{\tau + \eta}{m} \mathbf{P}^T \mathbf{1} \right)
$$

(13)

where $\mathbf{P}$ and $\mathbf{P}^T$ are the same as $\mathbf{P}$ in Eq. (2) except that they are produced by using the positive and negative samples, respectively, and $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^n$, $\mathbf{1} = [0, \ldots, 0]^T \in \mathbb{R}^n$. This TER cost function based on the quadratic approximation was applied to the RM and ELM basis function in [4,5]. They are called as TER-RM and TER-ELM, respectively.

### 2.3 AUC-based method

The receiver operating characteristics (ROC) curve has been extensively adopted for evaluating the classifier performance. However, the processes of classifier design optimization and the final ROC performance evaluation are usually conducted separately. This is mainly because the ROC does not have a well-posed structure due to the error counting point of view.

To overcome this drawback, a smooth approximate formulation by using a quadratic function was proposed to calculate the Area under ROC curve (AUC) [6]. This enables a direct optimization of the AUC with respect to the classifier parameters. The AUC [8] for the given $m$ training examples can be expressed as

$$
\text{AUC}(\mathbf{x}^+, \mathbf{x}^-) = \frac{1}{m^2 m} \sum_{i=1}^{m} \sum_{j=1}^{m} \text{L}(g(x_i^+) > g(x_j^-))
$$

(14)

where the term $1_{g(x_i^+) > g(x_j^-)}$ corresponds to a ‘1’ whenever the elements $g(x_i^+) > g(x_j^-)$, and ‘0’ otherwise. Let $g(x) = g(\mathbf{a}, \mathbf{x})$ with adjustable parameters $\mathbf{a}$ operating on the feature vector $\mathbf{x}$, then the goal to optimize the classifier’s discrimination performance can be treated as to maximize the AUC:

$$
\arg \max_{\mathbf{a}} \text{AUC}(\mathbf{a}, \mathbf{x}^+, \mathbf{x}^-) = \arg \max_{\mathbf{a}} \frac{1}{m^2 m} \sum_{i=1}^{m} \sum_{j=1}^{m} \left[ g(\mathbf{a}, \mathbf{x}_i^+) - g(\mathbf{a}, \mathbf{x}_j^-) \right]
$$

(15)

where $\eta$ is a unit step function. Maximizing the AUC is equal to minimizing the Area Above ROC curve (AAC) in Eq. (16).
arg min \( AAC(a, x, x') \) 

\[
= \arg \min_a \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{m} \mu \left[ g(a, x_i) - g(a, x'_i) \right]
\]

(16)

Since the approximation by using a sigmoid function can cause some problems as mentioned in the previous section, a quadratic function was used to approximate a step function. The approximated \( AAC \) after letting \( g(a, x) = \psi(x) a \) is

\[
\approx \arg \min_a \left\{ \frac{\beta}{2} |a| + \frac{1}{2m} \sum_{i=1}^{m} \sum_{j=1}^{m} \left[ (p(x_i) - p(x'_i))a + \eta \right] \right\}
\]

(17)

where \( \eta \) is a positive offset of a quadratic function. The optimal parameter \( a \) can be obtained as Eq. (18) by letting the derivative of Eq. (17) with respect to \( a \) be zero.

\[
a = \left( I + \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{m} (p_i - p_j)^T (p_i - p_j) \right)^{-1}
\]

\[
\times \left( \frac{-\eta}{m} \sum_{i=1}^{m} \sum_{j=1}^{m} (p_i - p_j) \right)
\]

(18)

where \( I \) is and identity matrix of \( K \times K \) size. The optimal threshold in the sense of the total error rate criterion [9] can be calculated as

\[
\tau = \frac{1}{2m} \sum_{i=1}^{m} p(x_i) a + \frac{1}{2m} \sum_{i=1}^{m} p(x'_i) a
\]

(19)

### 3. Experiments

#### 3.1 Experimental setup

In the experiment, we applied five different classification methods to five two-class problems in the UCI database [10]. Summaries of classification methods and data sets are shown in Table 1 and Table 2, respectively.

The experimental setup is as follows. Min-max normalization was applied to \( P \) matrix in case of RM-based methods and to the original feature vector in case of ELM-based methods. The reason for this procedure will be discussed later. 10 runs of 10-fold cross validation were performed. 1~10 orders for RM-based methods and 20~200 number of hidden neurons for ELM-based methods. A sigmoid function was used for ELM-based methods as an activation function due to its popularity and effectiveness. We fixed \( \tau = \eta = 0.5 \) for TER-RM and TER-ELM and \( \eta = 1 \) for AUC-RM. The regularization factor \( b \) was set to \( 10^{-4} \) for all methods.

To evaluate and compare each method, two criteria were used. One is the test classification error rate and the other is the LAUC \((-10 \log_{10} \text{AUC})\). For both criteria, a lower percentage means better performance.

### 3.2 Experimental results

The experimental results consist of two parts. First, the feature vector normalization procedure is discussed and then the results of five different classification methods are discussed.

For normalizing the feature vectors, we used the min-max normalization method which is known as a simple and effective technique [11]. Since several normalization methods have been already compared in that paper, we tried to analyze how to use it when using different basis functions (RM and ELM). We applied the min-max normalization technique in three different ways: no normalization, normalization before making \( P \) or \( H \) matrix, and normalization after making \( P \) or \( H \) matrix.

Fig. 1 shows the classification error rates of RM with three different normalization procedures for RM in terms of (a) classification error and (b) singular value ratio.

Fig. 1 shows the classification error rates of RM with three different normalization procedures by using Wdbc data set. In Fig. 1 (a), normalization after making \( P \) matrix produces the best performance and no normalization produces the worst performance. The reason that the results of normalization before and after making \( P \) matrix are better than the result of no normalization is the instability of the parameter estimation caused by a singularity of \( P^TP \) matrix which should be inverted. This singularity comes when generating \( P \) matrix. Since \( P \) matrix is produced by multiplying and adding many feature values, the component values of this matrix are quite unbalanced if feature values are not normalized. This makes \( P^TP \) matrix close to singular in case of no normalization.

One more important thing we can notice from Fig. 1 (a) is that the result of normalization after making \( P \) matrix produces a better performance than the result of normalization before making \( P \) matrix. This is also due to the singularity of

<table>
<thead>
<tr>
<th>Cost func.</th>
<th>Basis func.</th>
<th>LSE</th>
<th>TER</th>
<th>AUC</th>
</tr>
</thead>
</table>
PTP matrix. Even though feature values are normalized before making P matrix, the normalized feature values are multiplied and added each other while making P matrix. This also produces unbalanced component values in P matrix which causes a singularity of P^TP matrix.

Fig. 1 (b) shows how much singular P^TP matrix is in each normalization process and each order. The degree of singularity was measured by using a ratio between the largest and smallest singular values of P^TP matrix. Since a singular value indicates an importance of corresponding singular vector, it can be said that a larger singular value ratio means more singularity. Singular value ratios are presented in log scale. In cases of no normalization and normalization before making P matrix, singular value ratios increase with RM order. This is because a higher order leads more unbalanced components. However, this ratio is almost constant in case of normalization after P matrix. This is the reason that normalization after making P matrix produces a good and stable performance in all orders.

![Fig. 2. Comparison of three normalization procedures for ELM in terms of (a) classification error and (b) kurtosis value.](image)

Fig. 2 shows classification error rates of ELM with three different normalization procedures by using Wdbc data set. In Fig. 2 (a), the results of no normalization and normalization before making H matrix are almost the same and the result of normalization before making H matrix is the best. The reason that normalization after making H matrix and no normalization produce similar results are due to the nature of a sigmoid function used as an activation function. Fig. 3 shows histograms of feature values at each step of ELM in case of normalization after making H matrix. Fig. 3 (a) shows a histogram of the original feature values. These feature values are in the range of about 0–3500. Fig. 3 (b) shows a histogram of feature values after the random weight and bias. Most of the values are in the range of about 2000–2000. Fig. 3 (c) shows a histogram of feature values after applying a sigmoid function. Since a sigmoid function stretches out the values around zero and makes negative small values zeros and positive large values ones, most of the feature values are concentrated on zero and one. Therefore, it is almost meaningless to normalize feature values after making H matrix.

In Fig. 2 (a), the result of normalization before making H matrix is better than the results of others. This is because the pre-normalization makes the feature values bounded near zero before applying a sigmoid function. Fig. 4 (a), (b), and (c) show a histogram of the original feature values, a histogram of normalized feature values before making H matrix and a histogram of feature values after the random weight and bias, respectively. It can be seen that the feature values after the random weight and bias are in the range of about -4–5. Therefore, after applying a sigmoid function, the feature values become much richer (Fig. 4 (d)) than those of the normalization after making H matrix (Fig. 3 (d)). This is the reason that normalization before making H matrix outperforms than the others. Fig. 2 (b) shows kurtosis of the feature values of H matrix in each normalization procedure and each order. This value is near 3 for Gaussian distribution and 1 for sub-Gaussian distribution. Since the feature value distribution in cases of no normalization and normalization after making H matrix are shaped like sub-Gaussian, the kurtosis values are close to 1. In case of normalization before making H matrix, the feature value distribution is shaped like Gaussian, so the kurtosis values are close to 3.
case of normalization before making H matrix.

Fig. 4. Histograms of feature values at each step of ELM in case of normalization before making H matrix.

Fig. 5. TER and LAUC of RM, ELM, TER-RM, TER-ELM, and AUC-RM in five data sets. (a) and (b) columns show TER and LAUC in each data set, respectively.

The experimental results of five methods by using five two-class problems are shown in Fig. 5. (a) and (b) columns in this figure show the test classification error rates and LAUC values in each data set, respectively. The performances of all methods are very similar. Especially, TER-RM and AUC-RM show almost the same performance in terms of test error rate and LAUC. This is because TER-RM finds the optimal parameters with a fixed threshold to minimize the total error rate and AUC-RM find the optimal threshold with the fixed parameters to minimize the total error rate. Test error rate and LAUC value show very similar trend. It might be said that there is a strong correlation between test error rate and LAUC.

4. Conclusions

In this paper, five different classification methods based on three learning criteria and two basis functions were investigated. Appropriate normalization procedures for RM and ELM-based methods were also discussed. For two-class problems, five methods showed similar performances, especially, TER-RM and AUC-RM were quite similar. The results also showed that the classification error rate and LAUC are highly correlated. For data normalization, it was shown that the normalization should be applied after making P matrix in case of RM and before making H matrix in case of ELM. This can be generalized as follows: the data normalization procedure should be chosen with a consideration of a basis function properties.

References