# Pattern Classification With Class Probability Output Network

Woon Jeung Park and Rhee Man Kil, *Senior Member, IEEE*

***Abstract*—The output of a classifier is usually determined by the value of a discriminant function and a decision is made based on this output which does not necessarily represent the posterior probability for the soft decision of classification. In this context, it is desirable that the output of a classifier be calibrated in such a way to give the meaning of the posterior probability of class membership. This paper presents a new method of postprocessing for the probabilistic scaling of classifier's output. For this purpose, the output of a classifier is analyzed and the distribution of the output is described by the beta distribution parameters. For more accurate approximation of class output distribution, the beta distribution parameters as well as the kernel parameters describing the discriminant function are adjusted in such a way to improve the uniformity of beta cumulative distribution function (CDF) values for the given class output samples. As a result, the classifier with the proposed scaling method referred to as the class probability output network (CPON) can provide accurate posterior probabilities for the soft decision of classification. To show the effectiveness of the proposed method, the simulation for pattern classification using the support vector machine (SVM) classifiers is performed for the University of California at Irvine (UCI) data sets. The simulation results using the SVM classifiers with the proposed CPON demonstrated a statistically meaningful performance improvement over the SVM and SVM-related classifiers, and also other probabilistic scaling methods.**

***Index Terms*—Bayes decision, beta distribution, classification, probabilistic scaling, support vector machine (SVM).**

## I. Introduction

THERE are various ways of implementing pattern classifiers. The most popular way is using a discriminant function whose values indicate the degree of confidence for the classification; that is, the decision of classification is made by selecting the class that has the greatest discriminant value. However, more natural way of representing the degree of confidence for the classification is using the posterior probability for the decision. For the posterior probability, we can use the Bayes theorem to implement the pattern classifiers (Bayes classifiers) [1]–[4]. For these classifiers, we need to acquire information of the probability distribution of samples; however, in many real applications, we do not have prior knowledge of the probability distribution of samples and this makes it difficult to implement the Bayes classifiers. One way to solve this problem is using the nonparametric estimation; that is, the unknown form of the probability distribution is determined from data. For example, the Parzen window approach [5] can be applied to construct the probability density function of the sample population by locating the window function at each sample. However, one of the difficulties in this estimation is that we need enough number of samples for the accurate estimation of density functions and in many cases, this estimation requires many kernel functions. On the other hand, the classifiers using discriminant functions can have sparse kernel functions compared to the density estimation method while the output of the classifier does not necessarily represent the posterior probability. Furthermore, in many cases, the classifiers using discriminant functions are better than the classifiers using class probability estimates from the viewpoint of classification performance. In this context, we consider the postprocessing of the pattern classifiers using discriminant functions in such a way that the output of classifier represents the posterior probability of soft decision while the classifier preserves the sparsity of kernel functions. As a result, we may expect better classification performances compared to the canonical classifiers using discriminant functions alone.

For the training of pattern classifiers, it is important to avoid the over-fit or under-fit of the training data. Bearing this in mind, the training of classifiers from the viewpoint of the structural risk minimization (SRM) principle such as the training of support vector machines (SVMs) [6] is one of effective methods to deal with the problem of the over-fit or under-fit in pattern classification problems. In this context, we assume that the pattern classifier with sparse kernels such as the SVM is obtained from the training of a classifier for the given data. Here, we can consider this classifier as a many-to-one dimension reduction process so that the input pattern (or vector) is mapped into a scalar value representing the degree of confidence for the class. However, in general, this output value is not related to the probability of the class. In this context, we try to obtain the posterior probability of class membership from the distribution of classifier's output samples.

For the probabilistic scaling of classifier's output, one popular method is to use the logistic regression in which the posterior probability of class membership is estimated using the sigmoid type of "logit" transfer function. The kernel logistic regression (KLR) [7]–[10] is the kernelized version of logistic regression technique to solve the nonlinear classification problems. Another way is scaling the classifier's output using the scaler functions. Wahba [7], [11] suggested the soft max scaler method in which the "logit" transfer function representing the class conditional probability is used. On the other hand, the sigmoid-type scaling functions were also suggested using the first-order [12] and second-order [13] polynomials of the classifier's

Manuscript received July 01, 2008; revised June 05, 2009; accepted July 21, 2009. First published September 18, 2009; current version published October 07, 2009.

The authors are with the Department of Mathematical Sciences, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: tomato0720@kaist.ac.kr; rmkil@kaist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2009.2029103

output. In these methods, the parameters of the sigmoid-type scaling function are estimated from the classifier output samples. These probabilistic scaling methods showed the improved performances for some cases of classification problems, but they do not provide consistent improvement of the classification performances because the scaling method does not always estimate the posterior probability of class membership accurately if the distribution of the classifier's output does not fit the suggested scaling function. In this context, we consider a new method of class probability output network (CPON) in which the posterior probability of class membership is estimated using the beta distribution parameters under the assumption that the output of classifier lies within the finite range and the distribution of classifier's output is usually unimodal; that is, the distribution has one value that occurs with the greatest frequency. This assumption is quite reasonable for many cases of classification problems with the proper selection of kernels for the given data. The good features of beta distribution are that any symmetric or nonsymmetric unimodal distribution is well fitted to the beta distribution using two parameters $\alpha$ and $\beta$, and that these parameters are easily estimated from the mean and variance of data. Here, for each class, we can estimate the beta cumulative distribution function (CDF) using the parameters $\alpha$ and $\beta$. On the other hand, other density functions of exponential families such as Gaussian, Laplacian, exponential, gamma, etc., have long tails in positive side or both positive and negative sides of data values, and they are not so flexible as compared to the beta distribution. Furthermore, the beta distribution represents the conjugate prior of the binomial distribution; that is, in our case, the class probability in binary classification problems.

One of characteristics in this statistical estimation is that the CDF values for the classifier's output samples become uniformly distributed if the parameters of beta distribution are well fitted to the real distribution of the classifier's output. From this point of view, we checked the uniformity of the distribution of the CDF values for the classifier's output samples using the Kolmogorov–Smirnov (K–S) test [14] in order to select the parameters of beta distribution as well as the shape parameters of kernel functions describing the discriminant function. After the CDF for the classifier's output is estimated using beta distribution, we can determine the posterior probability of class membership using the CDF vales for each class. Then, the final decision for the class can be determined by identifying the class which has the maximum posterior probability of class membership. As a result, the suggested CPON method is able to provide consistent improvement of classification performances for the classifiers using discriminant functions alone. To show the effectiveness of the proposed method, the simulation for pattern classification using the SVM, KLR, SVM with Platt's scaling [12], and the suggested CPON method is performed for the University of California at Irvine (UCI) benchmark data [15]. We also applied the suggested method to improved versions of the SVM method such as the multicategory SVM (MSVM) [16]–[18] and $\psi$-learning methods [19], [20]. The simulation results using the suggested CPON method demonstrated a statistically meaningful performance improvement over the SVM, SVM-related, and also other probabilistic scaling methods.

This paper is organized as follows. In Section II, the classification using discriminant functions and probabilistic scaling methods are explained. In Section III, a detailed description of the suggested CPON method is presented. In Section IV, the simulation for pattern classification problems using the probabilistic scaling methods including the suggested CPON method is performed. Finally, in Section V, the conclusion to our work is presented.

## II. Pattern Classifiers With Probabilistic Scaling of Class Outputs

Pattern classifiers provide an output $\hat{y}$ as a discriminant value for the given input pattern $\mathbf{x}$. First, let us consider a binary classification problem. Then, the discriminant function $h$ can be formulated as

$$\hat{y} = h(\mathbf{x}) \tag{1}$$

where $\mathbf{x} \in X$ (an input space, an arbitrary subset of $\mathbf{R}^d$) represents the $d$ dimensional input pattern and $\hat{y} \in \mathbf{R}$ represents the output of discriminant function $h$ as an estimate for the output pattern $y \in Y$ (an output space, a set of $\{-1, +1\}$).

Suppose we have $K$ classes. Then, one of methods of implementing multiclass discriminant functions is to construct $K$ discriminant functions for each class; that is

$$\hat{y}_k = h_k(\mathbf{x}), \qquad \text{for} \quad k = 1, \ldots, K. \tag{2}$$

From these class outputs, the decision is made by the maximum discriminant value such as

$$\text{class} = \arg\max_k \hat{y}_k. \tag{3}$$

In general, the discriminant function $h_k$ can be constructed by a linear combination of kernel functions; that is

$$h_k(\mathbf{x}) = \sum_{j=1}^{m_k} w_{kj}\phi_{kj}(\mathbf{x}) \tag{4}$$

where $m_k$, $w_{kj}$, and $\phi_{kj}$ represent the number of kernel functions, the $j$th weight value, and the kernel function for the $k$th discriminant function, respectively.

These discriminant functions are trained for a set of $n$ training samples

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) | i = 1, \ldots, n\}$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ represent the $i$th input and output samples which are drawn randomly and independently from a population of samples with the probability distribution of $P(\mathbf{x}, \mathbf{y})$. Here, we assume that the $d$ dimensional input pattern $\mathbf{x}_i \in \mathbf{R}^d$ and the output pattern $\mathbf{y}_i \in [-1, +1]^K$ are given as a pair of samples.

For these samples, the goal of learning for each class is to construct an estimation function $h_k$ that minimizes the expected risk

$$R(h_k) = \int_{X \times Y} L\left(y_k, h_k(\mathbf{x})\right) dP(\mathbf{x}, y_k) \tag{5}$$

where $y_k$ and $L$ represent the target value and loss functional for the $k$th class, respectively.

In the case of classification problems, the loss functional $L$ is given by

$$L\left(y_k, h_k(\mathbf{x})\right) = \begin{cases} 1, & \text{if } y_k \cdot h_k(\mathbf{x}) \leqslant 0 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

To minimize the expected risk of (5), it is necessary to identify the probability distribution $P(\mathbf{x}, y_k)$; however, this is usually unknown. Rather, we usually find $h_k$ by minimizing the empirical risk $R_{\text{emp}}(h_k)$ evaluated by the mean of loss function values for the given samples; that is

$$R_{\text{emp}}(h_k) = \frac{1}{n}\sum_{i=1}^{n} L\left(y_{ki}, h_k(\mathbf{x}_i)\right) \tag{7}$$

where $y_{ki}$ represents the $i$th output sample for the $k$th class.

In this learning procedure, if the number of parameters of a classification model is exceedingly small and the performance is thus not optimal due to too much simplified description of the discriminant function, a situation known as underfitting of learning arises. On the other hand, if the number of parameters of a classification model is especially large and the performance is thus not optimal due to too much detailed description of discriminant function, overfitting of learning arises. Hence, a tradeoff exists between the underfitting and the overfitting of classification models. One way of solving this problem is using the structural risk minimization (SRM) principle such as the support vector machine (SVM) [6]. For the description of the SVM, let us consider the discriminant function for binary classification such as the form of (4) with the additional bias term $w_0$

$$h(\mathbf{x}) = \sum_{j=1}^{m} w_j K(\mathbf{x}, \mathbf{x}_j) + w_0 \tag{8}$$

where $m$ represents the number of kernel functions and $K$ represents Mercer's kernel, that is, a continuous, symmetric, and positive–definite kernel.

In the SVM, the problem of learning for binary classification is given by

$$\min_{w_j} \frac{1}{2}\sum_{j=0}^{m} w_j^2 + C\sum_{i=1}^{n}\xi_i$$

subject to

$$y_i \cdot h(\mathbf{x}_i) \geqslant 1 - \xi_i, \qquad \xi_i \geqslant 0, \qquad i = 1, \ldots, n \tag{9}$$

where $C$ represents the positive regularization constant indicating the tradeoff between the empirical error and the complexity term, and $\xi_i$ represents the slack variable for the $i$th pattern indicating the tolerance of the margin for the separating hyperplane.

This problem can be converted to the following optimization problem by the duality principle [6]:

$$\max_{\alpha_i} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to

$$0 \leqslant \alpha_i \leqslant C, \qquad i = 1, \ldots, n \quad \text{and} \quad \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{10}$$

where $\alpha_i$ represents the Lagrangian multiplier for the $i$th sample.

Then, this problem can be solved by the quadratic optimization problem [21]. As a result, only the support vectors, that is, the samples corresponding to nonzero values of $\alpha_i$, remain to constitute the solution of hyperplane as described in the Karush–Kuhn–Tucker (KKT) theorem [22], [23]. Let $\hat{\alpha}_i$ represent the solution of (10). Then, the final form of the hyperplane is determined by

$$h(\mathbf{x}) = \sum_{i\in I} y_i\hat{\alpha}_i K(\mathbf{x}, \mathbf{x}_i) + \hat{w}_0 \tag{11}$$

where $I$ represents the index set of support vectors and $\hat{w}_0$ represents the solution of bias term. Here, the number of support vectors is much smaller than the number of samples and this provides the sparsity of kernel functions.

In the logistic regression [24], the posterior probability of the class membership via the linear discriminant function is obtained. For the nonlinear form of logistic regression, a linear combination of kernel functions for the input space induced by Mercer kernels, referred to as the KLR [7]–[10], can be considered, that is, the form of (8). In this model, the output $h(\mathbf{x})$ is interpreted as an estimate of a probability $P(y = 1|\mathbf{x})$ using the following "logit" transfer function:

$$h(\mathbf{x}) = \text{logit}\left(P(y = 1|\mathbf{x})\right) = \log\frac{P(y = 1|\mathbf{x})}{1 - P(y = 1|\mathbf{x})}. \tag{12}$$

To choose the optimal parameters of $w_i$'s in (8), the following cost function representing the regularized negative-log likelihood of the data [25] is minimized:

$$L(\mathbf{w}) = -\sum_{i=1}^{n}\left(y_i \log h(\mathbf{x}_i) + (1 - y_i)\log\left(1 - h(\mathbf{x}_i)\right)\right) + \lambda\|\mathbf{w}\|^2 \tag{13}$$

where $\lambda$ represents a regularization parameter controlling the bias variance tradeoff.

The representing theorem [26] states that the solution to the above optimization problem has the form of a linear combination of the training patterns, that is, the form of (8). Furthermore, once the kernel functions are determined, $L$ becomes a convex function, so there is only one global minimum corresponding to the model parameters $w_i$'s. The optimization of $L$ can be implemented by the Newton method or other iterative least square methods [27].

In many cases, the classification performance of KLR is similar to that of SVM. One of the characteristics of the KLR is that it can provide class probabilities and sometimes these are more useful than the classification itself, for example, the problem of credit risk scoring. However, the KLR requires high computational complexity compared to the SVM, for example, $O(n^3)$ in the case of KLR versus $O(n^2 l)$ in the case of SVM where $l$ represents the number of support vectors and this number is quite small compared to the number of samples $n$.

Since the output of the SVM is not a class probability but a distance measure between a pattern and the decision boundary, it

seems suitable to model the conditional class probability $P(y|\mathbf{x})$ as a function of the output of the SVM, that is, $P(y=1|h) = \sigma(h)$ with an appropriate scaling function $\sigma$. One of methods in producing probable output from the discriminant function is using the soft-max scaler [7], [11] in which the following logistic link function is used:

$$\Pr\{y=1|h\} = \frac{1}{1+\exp(-h)} \tag{14}$$

where $h$ represents the output of SVM.

This is a monotonously increasing function of $h$ whose range is between 0 and 1. This scaler assumes that the decision function is the signum function "sign" of $h$, that is

$$\text{sign}(h) = \begin{cases} +1, & \text{for } h > 0 \\ 0, & \text{for } h = 0 \\ -1, & \text{for } h < 0. \end{cases}$$

Hence, the value of $h = 0$ represents the conditional class probability as 0.5. Hastie and Tibshirani [13] also proposed a method that fits Gaussians to the prior probabilities $\Pr\{h|y=1\}$ and $\Pr\{h|y=-1\}$ for the probabilistic scaling of SVM's output. In this method, a single tied variance is estimated for both Gaussians. Then, the posterior probability rule $\Pr\{y=1|h\}$ has the sigmoid form whose slope is determined by this tied variance. Here, the bias of the sigmoid is adjusted so that the point $\Pr\{y=1|h\} = 0.5$ occurs at $h=0$. One can also use a more flexible version of the Gaussian fit to $\Pr\{h|y=1\}$. Then, the mean and the variance for each Gaussian is determined from a data set. In this method, the conditional class probability has the form of

$$\Pr\{y=1|h\} = \frac{1}{1+\exp(ah^2+bh+c)} \tag{15}$$

where $a$, $b$, and $c$ represent variables trained from the output samples of SVM.

The class-conditional densities between the margins are apparently exponential. From this point of view, Platt [12] also suggested the following form of scaling the SVM's output:

$$\Pr\{y=1|h\} = \frac{1}{1+\exp(ah+b)} \tag{16}$$

where $a$ and $b$ represent variables trained from the output of SVM.

These parameters are adjusted by the maximum-likelihood estimation from the training set $(h_i, y_i)$, $i=1,\ldots,n$, where $h_i$ and $y_i$ represent the output of SVM and the target value for the $i$th sample, respectively. In this training set, the SVM's output $h_i$ is transformed to the target probability $t_i$ as

$$t_i = \frac{h_i+1}{2}. \tag{17}$$

Then, the parameters $a$ and $b$ are determined by minimizing the negative-log likelihood of the training data, which is a cross-entropy error function $E$

$$L(a,b) = -\sum_i \left(t_i \log(p_i) + (1-t_i)\log(1-p_i)\right) \tag{18}$$

where $p_i (= \Pr\{y_i = 1|h_i\})$ is given by (16).

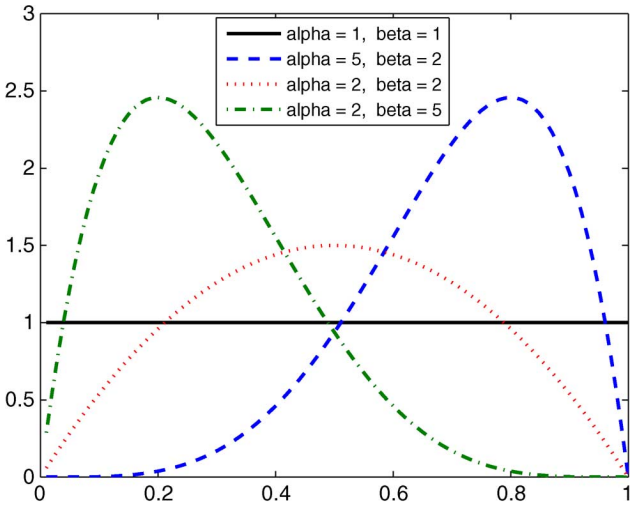


Fig. 1. Beta probability density functions for various parameter values of $\alpha$ and $\beta$.

This sigmoid model is different from the one proposed by Hastie and Tibshirani because it has two parameters trained discriminatively, rather than three parameters estimated from a tied variance. Through the simulation for classification problems using the UCI data sets, Platt showed that this type of scaling method was able to improve the classification performance of SVM. However, in the aforementioned methods, the network output does not always estimate probabilities accurately if the class probability does not fit the suggested function. This makes wrong decision of the target value corresponding to the network output. In this context, we estimate the posterior probability of class membership using the beta distribution from the network output and make the decision function according to the estimated posterior probability. In this estimation, there is no restriction about the shape of the true model as long as the distribution of the classifier's output is unimodal. A detailed description of the suggested method follows in the next section.

## III. Construction of Class Probability Output Network

The distribution of the classifier's output can be approximated by the beta distribution under the assumption that the classifier's output has unimodal distribution; that is, one output value corresponding to the greatest frequency exists, and the range of output values is finite. Under these assumptions, we consider the following probability density function $f_Y$ of beta random variable $Y$ whose values lie between 0 and 1:

$$f_Y(y|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} y^{\alpha-1}(1-y)^{\beta-1} \tag{19}$$

where $\alpha$ and $\beta$ represent positive constants for the beta density function, and $B(\alpha,\beta)$ represents the beta function defined by

$$B(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx. \tag{20}$$

Here, we assume that the values of the classifier's output are normalized between 0 and 1. As the parameters $\alpha$ and $\beta$ vary, the

beta distribution takes on various shapes as illustrated in Fig. 1. This beta function is related to the gamma function $\Gamma$ through the following identity:

$$B(\alpha,\beta)=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{21}$$

where the gamma function is defined by

$$\Gamma(\alpha)=\int_0^\infty \lambda e^{-\lambda x}(\lambda x)^{\alpha-1}dx, \qquad \lambda>0.$$

Then, the CDF of beta random variable $Y$ is described by

$$F_Y(y|\alpha,\beta)=\frac{1}{B(\alpha,\beta)}\int_0^y x^{\alpha-1}(1-x)^{\beta-1}dx. \tag{22}$$

In this beta distribution, the calculation of moments is quite convenient due to the particular form of beta function (21), that is, for $n>-\alpha$

$$\begin{aligned}E[Y^n]&=\frac{1}{B(\alpha,\beta)}\int_0^1 y^n y^{\alpha-1}(1-y)^{\beta-1}dy\\&=\frac{B(\alpha+n,\beta)}{B(\alpha,\beta)}\\&=\frac{\Gamma(\alpha+n)\Gamma(\alpha+\beta)}{\Gamma(\alpha+\beta+n)\Gamma(\alpha)}.\end{aligned} \tag{23}$$

From this result, we get the mean and variance of beta distribution

$$E[Y]=\frac{\alpha}{\alpha+\beta} \tag{24}$$

and

$$\mathrm{Var}(Y)=E[Y^2]-E^2[Y]=\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{25}$$

using the recurrence relationship of the gamma function

$$\Gamma(\alpha+1)=\alpha\Gamma(\alpha), \qquad \alpha>0.$$

From these results, if we know the mean and variance of data, we can obtain the parameters of the beta distribution, that is, $\alpha$ and $\beta$ directly

$$\alpha=E[Y]\left(\frac{E[Y](1-E[Y])}{\mathrm{Var}(Y)}-1\right) \tag{26}$$

and

$$\beta=(1-E[Y])\left(\frac{E[Y](1-E[Y])}{\mathrm{Var}(Y)}-1\right). \tag{27}$$

In our case, the parameters $\alpha$ and $\beta$ should be estimated from the data. Here, let the output of classifiers be given by $Y_i$, $i=1,\ldots,n$. Then, the sample mean $\bar{Y}$ and sample variance $S_Y^2$ are given by

$$\bar{Y}=\frac{1}{n}\sum_{i=1}^n Y_i \tag{28}$$

and

$$S_Y^2=\frac{1}{n-1}\sum_{i=1}^n (Y_i-\bar{Y})^2. \tag{29}$$

Then, the estimates of the parameters $\alpha$ and $\beta$ can be obtained by replacing $E[Y]$ and $Var(Y)$ with the sample mean $\bar{Y}$ and sample variance $S_Y^2$ in (26) and (27). However, in general, these estimates do not always provide the optimal values of beta distribution parameters due to not enough samples. If these estimates are accurate, the distribution of the CDF of data, that is, $F_Y(Y_i)$, $i=1,\ldots,n$, becomes uniformly distributed by the following reason:

- let us consider random variables $U_i = F_Y(Y_i)$, $i=1,\ldots,n$;
- then each $U_i$ has uniform distribution because its probability density function is given by

$$f_U(u)=\frac{f_Y(y)}{|dF_Y/dy|}=\frac{f_Y(y)}{|f_Y(y)|}=1 \tag{30}$$

  where $u=F_Y(y)$.

From this point of view, we need to check the uniformity of $F_Y(Y_i)$, $i=1,\ldots,n$, using the hypothesis test such as the K–S test [14], which is performed for the empirical cumulative distribution function (ECDF). The procedure of the K–S test is given as follows.

- Let $\mathcal{S}_Y=\{Y_i|i=1,\ldots,n\}$ be a sample from the CDF $F_Y$, and let $F_n^*$ be the corresponding ECDF. Then, the K–S statistic $D_n$ is defined by

$$D_n=\sup_{y\in\mathcal{S}_Y}\left|F_n^*(y)-F_Y(y)\right|. \tag{31}$$

  In the case of uniform distribution, $F_Y(y)=y$.
- The K–S statistic $D_n$ can be used to test the following hypotheses:

$$H_0: F_n^*(y)=F_Y(y) \quad \text{versus} \quad H_1: F_n^*(y)\neq F_Y(y).$$

  Let the value of $D_n$ be $d_n$. Then, to test these hypotheses, we can calculate the $p$-value from the K–S distribution

$$p\text{-value}=\Pr\left\{D_n\geqslant\frac{t}{\sqrt{n}}\right\}=1-H(t) \tag{32}$$

  where $t$ represents a variable defined by $t=\sqrt{n}d_n$ and $H(t)$ represents the CDF of the K–S statistic determined by

$$H(t)=1-2\sum_{i=1}^\infty(-1)^{i-1}e^{-2i^2t^2}. \tag{33}$$

- Finally, the testing of the hypotheses at a significance level $\delta$ is to

$$\text{accept } H_0, \text{ if } p\text{-value}\geqslant\delta; \text{ reject } H_0, \text{ otherwise.}$$

If the data $F_n^*(Y_i)$, $i=1,\ldots,n$, pass the hypothesis test of uniform distribution, it implies that the estimates of $\alpha$ and $\beta$ are good enough to represent the distribution of the classifier's output. However, in some cases, this hypothesis of uniform distribution may be rejected due to the inaccurate estimation of

beta distribution parameters. In this context, we also consider the fine tuning of $\alpha$ and $\beta$ to improve the uniformity of $F_n^*(Y_i)$, $i = 1, \ldots, n$. First, let us consider the approximation of CDF $F_n^*$ of beta distribution using the following trapezoidal rule.

- Let the normalized classifier's outputs $Y_i$, $i = 0, 1, \ldots, n$, be sorted in ascending order where $Y_0 = 0$ and $Y_n = 1$;
- then the area under the curve of $f_Y$ between $Y_0 = 0$ and $Y_n = 1$ can be divided into $n$ strips each with a width of $\Delta Y_i = Y_i - Y_{i-1}$ and the shape of each strip is approximated as a trapezium. Here, the area $a_i$ of the $i$th strip is approximated as

$$a_i = \frac{\Delta Y_i}{2}\left(f_Y(Y_i) + f_Y(Y_{i-1})\right). \tag{34}$$

Adding up these areas gives us an approximated value for the CDF

$$F_n^*(y = Y_j|\alpha, \beta) \approx \sum_{i=1}^{j} \frac{\Delta Y_i}{2}\left(f_Y(Y_i) + f_Y(Y_{i-1})\right). \tag{35}$$

Using this trapezoidal rule, we can adjust the beta distribution's parameters $\alpha$ and $\beta$ to improve the uniformity of $F_n^*(Y_i)$. Here, let us define the difference of $F_n^*$ as

$$\begin{aligned}\Delta_i &= F_n^*(Y_i) - F_n^*(Y_{i-1}) \\ &\approx \frac{C_i}{2}\left(Y_i^{\alpha-1}(1-Y_i)^{\beta-1} + Y_{i-1}^{\alpha-1}(1-Y_{i-1})^{\beta-1}\right)\end{aligned} \tag{36}$$

where $C_i$ represents a constant defined by

$$C_i = \Delta Y_i / B(\alpha, \beta).$$

Then, the sample mean and variance of $\Delta_i$, $i = 1, \ldots, n$, are determined by

$$\bar{\Delta} = \frac{1}{n}\sum_{i=1}^{n} \Delta_i \tag{37}$$

and

$$S_\Delta^2 = \frac{1}{n-1}\sum_{i=1}^{n} (\Delta_i - \bar{\Delta})^2. \tag{38}$$

Here, to improve the uniformity of $F_n^*(Y_i)$, we need to adjust the beta distribution's parameters $\alpha$ and $\beta$ in such a way to minimize the sample variance $S_\Delta^2$. From this point of view, we can consider the following update rule of $\alpha$ and $\beta$:

$$\alpha^{\text{new}} = \alpha^{\text{old}} + \eta \Delta\alpha \tag{39}$$

and

$$\beta^{\text{new}} = \beta^{\text{old}} + \eta \Delta\beta \tag{40}$$

where $\eta$ represents a positive constant referred to as the learning rate. Here, $\Delta\alpha$ and $\Delta\beta$ can be determined in such a way to minimize the sample variance $S_\Delta^2$, that is

$$\begin{aligned}\Delta\alpha &= -\frac{\partial S_\Delta^2}{\partial \alpha} \\ &= -\frac{2}{n-1}\sum_{i=1}^{n}(\Delta_i - \bar{\Delta})\left(\frac{\partial \Delta_i}{\partial \alpha} - \frac{1}{n}\sum_{j=1}^{n}\frac{\partial \Delta_i}{\partial \alpha}\right) \\ &= -\frac{1}{n-1}\sum_{i=1}^{n} C_i(\Delta_i - \bar{\Delta})(\alpha - 1)\left(A_i - \frac{1}{n}\sum_{j=1}^{n} A_j\right)\end{aligned} \tag{41}$$

where

$$A_i = Y_i^{\alpha-2}(1 - Y_i)^{\beta-1} + Y_{i-1}^{\alpha-2}(1 - Y_{i-1})^{\beta-1}$$

and

$$\begin{aligned}\Delta\beta &= -\frac{\partial S_\Delta^2}{\partial \beta} \\ &= -\frac{2}{n-1}\sum_{i=1}^{n}(\Delta_i - \bar{\Delta})\left(\frac{\partial \Delta_i}{\partial \beta} - \frac{1}{n}\sum_{j=1}^{n}\frac{\partial \Delta_i}{\partial \beta}\right) \\ &= -\frac{1}{n-1}\sum_{i=1}^{n} C_i(\Delta_i - \bar{\Delta})(\beta - 1)\left(B_i - \frac{1}{n}\sum_{j=1}^{n} B_j\right)\end{aligned} \tag{42}$$

where

$$B_i = Y_i^{\alpha-1}(1 - Y_i)^{\beta-2} + Y_{i-1}^{\alpha-1}(1 - Y_{i-1})^{\beta-2}.$$

This update rule of beta distribution parameters provides more accurate estimation of the distribution of classifier's output samples by assuring the higher confidence of the uniformity of $F_n^*(Y_i)$. To explain this effect, let us decompose $F_n^*(Y_i)$ by

$$F_n^*(Y_i) = F_Y(Y_i) + \epsilon_i \tag{43}$$

where $F_Y(Y_i)$ represents the $i$th value of theoretical distribution and $\epsilon_i$ represents the $i$th independent noise term with mean $\mu$ and variance $\sigma^2$. In the case of uniform distribution with $n$ segments between 0 and 1, $F_Y(Y_i) = i/n$.

Then, $\Delta_i$, that is, the difference between $F_n^*(Y_i)$ and $F_n^*(Y_{i-1})$ becomes

$$\Delta_i = F_n^*(Y_i) - F_n^*(Y_{i-1}) = \frac{1}{n} + \epsilon_i - \epsilon_{i-1}. \tag{44}$$

The expectation and the variance of the above term are determined by

$$E[\Delta_i] = \frac{1}{n} \tag{45}$$

and

$$\begin{aligned}\mathrm{Var}(\Delta_i) &= E\left[(\epsilon_i - \epsilon_{i-1})^2\right] \\ &= 2\sigma^2. \end{aligned} \tag{46}$$

The sample mean and variance of $\Delta_i$ are determined by

$$\bar{\Delta} = \frac{1}{n} + \frac{1}{n}\sum_{i=1}^{n}(\epsilon_i - \epsilon_{i-1}) \tag{47}$$

and

$$\begin{aligned} S_\Delta^2 &= \frac{1}{n-1}\left\{\sum_{i=1}^{n}(\Delta_i - \bar{\Delta})^2\right\} \\ &= \frac{1}{n-1}\left\{\sum_{i=1}^{n}\Delta_i^2 - n\bar{\Delta}^2\right\}. \end{aligned} \tag{48}$$

This sample variance is an unbiased estimator of $Var(\Delta_i)$ since

$$\begin{aligned} E\left[S_\Delta^2\right] &= \frac{1}{n-1}\left\{\sum_{i=1}^{n}E\left[\Delta_i^2\right] - nE[\bar{\Delta}^2]\right\} \\ &= \frac{n}{n-1}\left\{\left(\mathrm{Var}(\Delta_i) + E^2[\Delta_i]\right) - \left(\mathrm{Var}(\bar{\Delta}) + E^2[\bar{\Delta}]\right)\right\} \\ &= \mathrm{Var}(\Delta_i). \end{aligned} \tag{49}$$

Here, let us consider the following probability of $|\epsilon_i - \mu|$:

$$\Pr\left\{|\epsilon_i - \mu| < \theta\right\} \geqslant 1 - \left(\frac{\sigma}{\theta}\right)^2 \tag{50}$$

where $\theta$ represents a positive constant.

The above inequality always holds regardless of the distribution of $\epsilon_i$ according to the Chebyshev inequality. Then, for $\epsilon_i$, $i = 1, \ldots, n$

$$\Pr\left\{\sup_{1\leqslant i\leqslant n}|\epsilon_i - \mu| < \theta\right\} \geqslant \left(1 - \left(\frac{\sigma}{\theta}\right)^2\right)^n. \tag{51}$$

Here, let us set

$$1 - \delta = \left(1 - \left(\frac{\sigma}{\theta}\right)^2\right)^n \tag{52}$$

where $\delta$ represents a positive constant between 0 and 1.

Then, with a probability of at least $1 - \delta$

$$\theta - k\sigma < \epsilon_i, \qquad i = 1, \ldots, n < \theta + k\sigma \tag{53}$$

where

$$k = \left(1 - (1-\delta)^{1/n}\right)^{-1/2}. \tag{54}$$

That is, with a probability of at least $1 - \delta$

$$D_n = \sup_{1\leqslant i\leqslant n}|\epsilon_i| < \max\left(|\theta - k\sigma|, |\theta + k\sigma|\right). \tag{55}$$

This implies that the distance measure of K–S statistic $D_n$ is limited by $\sigma$, that is, the standard deviation of $\epsilon_i$. In this context, if the sample variance $S_\Delta^2$ becomes smaller by adjusting beta parameters, $\sigma$ is likely to be smaller and $D_n$ decreases according to (55) with a probability of at least $1 - \delta$. This causes $p$-value of (32) likely to be higher. In other words, we have higher confidence of the uniformity of $F_n^*(Y_i)$. To see the effectiveness of this fine-tuning method of beta distribution parameters, 100 sets of 100, 500, and 1000 samples were generated using the beta distribution parameters $\alpha = 1$ and $\beta = 3$. Then, beta parameters $\alpha$ and $\beta$ were estimated using the moment-matching (MM) method of (26) and (27), the maximum-likelihood estimation (MLE) method [28], and the suggested minimum variance (MV) method of (39) and (40). As a measure of these estimates, we used the $p$-value of (32), that is, the $p$-value of testing the uniformity of empirical CDF since it represents the confidence of how well the estimated distribution is matched with the ideal distribution. The simulation results of average $p$-values for 100 sets of 100, 500, and 1000 samples are described in Table I. These simulation results demonstrate that: 1) $p$-values become higher as the number of samples increases; 2) the MLE method provides better $p$-values than the MM method; and 3) the suggested MV method provides the best $p$-values, and this is evident especially for smaller number of samples. These results show that the suggested MV method is quite effective in estimating beta distribution parameters.

TABLE I
AVERAGE $p$-VALUES FOR 100 SETS OF 100, 500, AND 1000 SAMPLES IN THE ESTIMATION OF BETA PARAMETERS USING THE MM, MLE, AND MV METHODS: EACH SET OF SAMPLES IS GENERATED FROM THE BETA DISTRIBUTION WITH PARAMETERS $\alpha = 1$ AND $\beta = 3$

| Number of Samples | MM | MLE | MV |
|---|---|---|---|
| 100 | 0.8577 | 0.8800 | **0.8958** |
| 500 | 0.8603 | 0.9050 | **0.9103** |
| 1000 | 0.9470 | **0.9541** | **0.9541** |

On the other hand, the uniformity of empirical CDF of $Y_i$ is also dependent upon the kernel parameters describing the discriminant function. From this point of view, we also consider the selection of kernel parameters maximizing the uniformity of empirical CDF of $Y_i$ in the suggested method. This point is also distinctive compared to the conventional kernel selection method in which the kernel parameters are usually selected in such a way as to minimize the mean square error (MSE) of classification. Summarizing the aforementioned methods, we suggest the following learning algorithm of the pattern classifier with class probability output network (CPON).

## Learning Algorithm of CPON

For the given samples of a $K$ class pattern classification problem, we construct the $K$ binary (one-against-the-rest) pattern classifiers. The pattern classifier for each class is constructed by the following procedure.

Step 1. For the given sample set $\mathcal{S}_k$ for the $k$th class, divide into the training sample set $\mathcal{S}_k^{\mathrm{train}}$ and the validation set $\mathcal{S}_k^{\mathrm{valid}}$.

Step 2. Train the classifier of the $k$th class using the training sample set $\mathcal{S}_k^{\mathrm{train}}$ for the candidates of kernel parameters $\theta_j$, $j = 1, \ldots, m$.

Step 3. For each set of kernel parameters $\theta_j$, do the following procedure.

3-1) Obtain the outputs of classifiers for the training sample set $\mathcal{S}_k^{\text{train}}$, that is, $y_{kji} = f_k(\mathbf{x}_i|\theta_j)$, $i = 1, \ldots, n$.

3-2) Normalize the outputs of classifiers such that the range of values is between 0 and 1, that is, for $i = 1, \ldots, n$

$$Y_{kji} = \frac{y_{kji} - \min_i y_{kji}}{\max_i y_{kji} - \min_i y_{kji}}. \tag{56}$$

3-3) Divide samples $Y_{kji}$, $i = 1, \ldots, n$, into the samples of "+" and "−" groups according to the class labels, that is, $Y_{kji}^+$, $i = 1, \ldots, n_k^+$, $Y_{kji}^-$, $i = 1, \ldots, n_k^-$, and $n_k^+ + n_k^- = n$.

3-4) Compute the sample mean $\bar{Y}_{kj}^{\pm}$ and sample variance $S^2_{Y_{kj}^{\pm}}$ for $Y_{kji}^{\pm}$ using (28) and (29).

3-5) Compute the parameters of beta distribution, that is, $\alpha_{kj}^{\pm}$ and $\beta_{kj}^{\pm}$ from the sample mean $\bar{Y}_{kj}^{\pm}$ and sample variance $S^2_{Y_{kj}^{\pm}}$ using the MM or MLE methods [28].

3-6) Compute the CDF values for $Y_{kji}^{\pm}$, that is, $F_{Y_{kj}^{\pm}}$ using (22).

3-7) Determine the $p$-value to check the uniformity of $F_{Y_{kj}^{\pm}}$ using the validation set $\mathcal{S}_k^{\text{valid}}$.

- Divide the validation set $\mathcal{S}_k^{\text{valid}}$ into two groups $\mathcal{S}_k^{\text{valid}+}$ and $\mathcal{S}_k^{\text{valid}-}$ according to the class labels. Here, we assume that the sample sizes of $\mathcal{S}_k^{\text{valid}+}$ and $\mathcal{S}_k^{\text{valid}-}$ are given by $l_k^+$ and $l_k^-$, respectively.
- Compute the K–S statistic using (31)

$$y_{kj}^{\pm} = \sup_{y \in \mathcal{S}_k^{\text{valid}\pm}} \left| F_{Y_{kj}^{\pm}}(y) - y \right|$$

- Calculate the $p$-values for "+" and "−" groups

$$p_{kj}^{\pm} = \Pr\left\{ D_{l_k^{\pm}} \geqslant y_{kj}^{\pm} \right\} = 1 - H\left( \sqrt{l_k^{\pm}}\, y_{kj}^{\pm} \right)$$

where $H$ represents the CDF of K–S statistic determined by (33).

Step 4. Select the best kernel parameter maximizing the uniformity of $F_{Y_{kj}^{\pm}}$, that is, select the index of kernel parameter maximizing the $p$-value

$$j^* = \arg\max_j p_{kj}^+ \cdot p_{kj}^-.$$

Step 5. Perform the fine tuning of the beta distribution parameters $\alpha_{kj^*}^{\pm}$ and $\beta_{kj^*}^{\pm}$ for the samples set $\mathcal{S}_k$ using the update rule of (39) and (40).

After training the CPON, the classification for an unknown sample can be determined by the beta distribution for each class. First, for an unknown sample, the normalized output $\bar{y}_k$ for the classifier of the $k$th class is computed using (56). Here, if the value of $\bar{y}_k$ is greater than 1, we set that value as 1; on the other hand, if the value of $\bar{y}_k$ is less than 0, we set that value as 0. Then, the conditional class probability for the given normalized output $\bar{y}_k$ can be obtained by the following Bayesian formula:

$$\Pr\{C_k|Y_1 = \bar{y}_1, \ldots, Y_K = \bar{y}_K\} = \frac{\Pr\{Y_1 = \bar{y}_1, \ldots, Y_K = \bar{y}_K|C_k\} \Pr\{C_k\}}{\sum_{j=1}^K \Pr\{Y_1 = \bar{y}_1, \ldots, Y_K = \bar{y}_K|C_j\} \Pr\{C_j\}}. \tag{57}$$

Assuming the conditional independence between $Y_i$'s given the class $C_k$, the above equation is redescribed by

$$\Pr\{C_k|Y_1 = \bar{y}_1, \ldots, Y_K = \bar{y}_K\} = \frac{\Pr\{C_k\} \prod_{i=1}^K \Pr\{Y_i = \bar{y}_i|C_k\}}{\sum_{j=1}^K \Pr\{C_j\} \prod_{i=1}^K \Pr\{Y_i = \bar{y}_i|C_j\}}. \tag{58}$$

This implies that we need to estimate $K$ distributions for each output, that is, $\Pr\{Y_i = \bar{y}_i|C_k\}$, $k = 1, \ldots, K$. Thus, in total, we need to estimate $K^2$ output distributions. If the estimation of output distributions for a certain class is not good enough (for example, due to not enough samples), this will result in inaccurate calculations of conditional class probabilities because of the product form in (58). In this context, we consider two groups of samples in each class, that is, the "+" group of samples that belong to the class, and the "−" group of samples that belong to the rest of classes. Then, we need to estimate $2K$ output distributions. In this case, each conditional probability represents a class probability that the given instance belongs to the "+" group among two groups, that is, the "+" and "−" groups. Here, each conditional probability for the given normalized output $\bar{y}_k$ can be obtained by the following Bayesian formula:

$$\Pr\left\{C_k^+|Y_k = \bar{y}_k\right\} = \frac{\Pr\left\{Y_k = \bar{y}_k|C_k^+\right\} \Pr\left\{C_k^+\right\}}{\Pr\left\{Y_k = \bar{y}_k|C_k^+\right\} \Pr\left\{C_k^+\right\} + \Pr\left\{Y_k = \bar{y}_k|C_k^-\right\} \Pr\left\{C_k^-\right\}} \tag{59}$$

where $C_k^+$ and $C_k^-$ represent the "+" and "−" groups in the $k$th class.

For the computation of $\Pr\{Y_k = \bar{y}_k|C_k^+\}$ and $\Pr\{Y_k = \bar{y}_k|C_k^-\}$, we can use the probability density functions $f_{Y_k^+}(\bar{y}_k)$ and $f_{Y_k^-}(\bar{y}_k)$, respectively. In this case, if both density functions have the same symmetric shape around mean such as Gaussian density function, this decision guarantees the minimum Bayes error rate. However, in general, the density functions can have different shapes and the classification performance is very much dependent upon the shapes of both density functions. In this context, more meaningful computation can be obtained if we use the CDF values $F_{Y_k^+}(\bar{y}_k)$ and $F_{Y_k^-}(\bar{y}_k)$ since $\bar{y}_k$ itself represents the degree of confidence for the $k$th class and this may be somehow related to the conditional class probability. This method of using CDF values is also related to the $p$-value of the hypothesis test, whether the sample belongs to the class, when the normalized output $\bar{y}_k$ is given. Here, let us consider the threshold value $\theta$ for binary classification, that is, if $\bar{y}_k > \theta$, the decision is "+"; otherwise, the decision is "−." Then, the error for binary classification is

$$\begin{aligned}\Pr\{\text{error at } \theta\} &= \Pr\left\{\text{error at } \theta|C_k^+\right\} \Pr\left\{C_k^+\right\} + \Pr\left\{\text{error at } \theta|C_k^-\right\} \Pr\left\{C_k^-\right\} \\ &= F_{Y_k^+}(\theta) \Pr\left\{C_k^+\right\} + \left(1 - F_{Y_k^-}(\theta)\right) \Pr\{C_k^-\}.\end{aligned} \tag{60}$$

Note that in the case of $C_k^-$, the probability of error for $C_k^-$ is $1 - F_{Y_k^-}(\theta)$ since the direction of output for high confidence is reversed from the case of $C_k^+$. For the minimum Bayes error rate, $\theta$ that minimizes $\Pr\{\text{error at } \theta\}$ should be selected.

Here, let us consider the following posterior probability of class membership:

$$\begin{aligned}&\Pr\{C_k^+|Y_k^+ \leqslant \bar{y}_k \text{ or } Y_k^- \geqslant \bar{y}_k\}\\&=\frac{\Pr\{Y_k^+ \leqslant \bar{y}_k|C_k^+\}\Pr\{C_k^+\}}{\Pr\{Y_k^+ \leqslant \bar{y}_k|C_k^+\}\Pr\{C_k^+\}+\Pr\{Y_k^- \geqslant \bar{y}_k|C_k^-\}\Pr\{C_k^-\}}\\&=\frac{F_{Y_k^+}(\bar{y}_k)\Pr\{C_k^+\}}{F_{Y_k^+}(\bar{y}_k)\Pr\{C_k^+\}+\left(1-F_{Y_k^-}(\bar{y}_k)\right)\Pr\{C_k^-\}}\end{aligned} \tag{61}$$

where $Y_k^+$ and $Y_k^-$ represent the random variable for the normalized output when the class label is given by $C_k^+$ and $C_k^-$, respectively.

If the probabilities of the "+" and "−" groups are the same, that is, $\Pr\{C_k^+\} = \Pr\{C_k^-\}$, the posterior probability of class membership for the given $\bar{y}_k$, becomes

$$\Pr\left\{C_k^+|Y_k^+ \leqslant \bar{y}_k \text{ or } Y_k^- \geqslant \bar{y}_k\right\} = \frac{F_{Y_k^+}(\bar{y}_k)}{F_{Y_k^+}(\bar{y}_k) - F_{Y_k^-}(\bar{y}_k) + 1}. \tag{62}$$

Here, let us consider the following $p$-values of testing hypotheses $H_k^+$ and $H_k^-$:

$$\begin{aligned}p\text{-value of testing } H_k^+ &= \Pr\left\{Y_k^+ \leqslant \bar{y}_k\right\}\\&= F_{Y_k^+}(\bar{y}_k)\end{aligned} \tag{63}$$

and

$$\begin{aligned}p\text{-value of testing } H_k^- &= \Pr\left\{Y_k^- \geqslant \bar{y}_k\right\}\\&= 1 - F_{Y_k^-}(\bar{y}_k)\end{aligned} \tag{64}$$

where $H_k^+$ and $H_k^-$ represent the hypotheses that the given instance belongs to $C_k^+$ and $C_k^-$, respectively.

Then, from the definitions of the error for binary classification and also of $p$-values of testing hypotheses of classification, the posterior probability of (62) can be redescribed by

$$\begin{aligned}&\Pr\left\{C_k^+|Y_k^+ \leqslant \bar{y}_k \text{ or } Y_k^- \geqslant \bar{y}_k\right\}\\&= \frac{\Pr\left\{\text{error at } \bar{y}_k, C_k^+\right\}}{\Pr\{\text{error at } \bar{y}_k\}}\\&= \frac{p\text{-value of testing } H_k^+}{p\text{-value of testing } H_k^+ + p\text{-value of testing } H_k^-}.\end{aligned} \tag{65}$$

If this value is close to 1, this implies that the probability of error for $C_k^+$ is almost the same as the probability of total error when the threshold value for binary classification is set as $\bar{y}_k$. In other words, the $p$-value of testing $H_k^+$, that is, the confidence of $H_k^+$, is much higher than that of $H_k^-$. Thus, the given sample whose normalized output is $\bar{y}_k$ should be classified as $C_k^+$. On the other hand, if the value of (65) is near 0, the given sample has low confidence of $C_k^+$ or high confidence of $C_k^-$. If the value of (65) is 1/2, this implies that the probability of error for $C_k^+$ is the same as the probability of error for $C_k^-$, that is, both confidence of $C_k^+$ and confidence of $C_k^-$ are the same. In this case, if both density functions of $C_k^+$ and $C_k^-$ have the same symmetric shape, $\bar{y}_k$ represents the optimal threshold value for binary classification that guarantees the minimum Bayes error rate. Even though both density functions have different shapes, the optimal threshold value is usually near $\bar{y}_k$. As an example, we consider binary classification samples generated from two normal distributions with different variances as illustrated in Fig. 2. The SVM is trained for 200 samples composed of two sets of 100 samples for binary class data generated from two normal distributions $\mathcal{N}(-1.0, 4.0)$ and $\mathcal{N}(1.0, 0.25)$. Then, we determined the classification boundaries for the SVM, CPON, and Bayes classifiers. As shown in Fig. 2, the suggested CPON provides better classification boundary than the SVM and this boundary is close to the Bayes decision boundary: the classification errors for the SVM, CPON, and Bayes classifiers are determined by 0.1851, 0.1639, and 0.1635, respectively. Even in the case of nonunimodal distributions of classifier's output, the CPON is quite well fitted since the kernel parameters are adjusted (using Steps 3 and 4 of the CPON algorithm) in such a way of maximizing the $p$-value for the fitness of the ideal distribution, that is, the beta distribution in our case. As an example, the output distributions of the SVM for the credit approval data set from the UCI database are not unimodal as illustrated in Fig. 3(a). In this case, the Gaussian kernels are used and the kernel width is adjusted in such a way as to minimize the MSE of the validation set. After applying the CPON algorithm, the output distributions of the CPON are more likely unimodal as illustrated in Fig. 3(b) due to the different setting of kernel parameters. Here, the error rates of the SVM and the CPON are determined by 0.2000 and 0.1824, respectively. These examples demonstrate the effectiveness of the CPON for determining the classification boundaries using the suggested training algorithm and the CDF-based decision rule. The suggested training algorithm provides accurate beta density function by including the estimation of beta parameters and also the selection of kernel parameters from the viewpoint of the uniformity criteria of the CDF of normalized SVM's output samples, and the suggested CDF-based decision rule provides accurate decision boundaries by investigating $p$-values of testing the classification hypotheses. In this

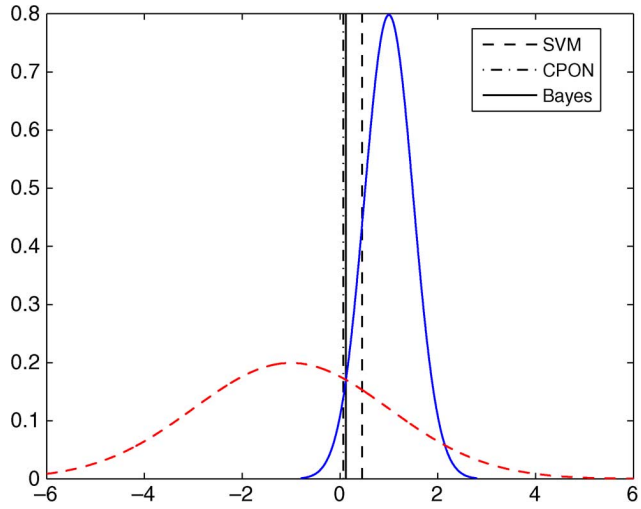


Fig. 2. Comparison of decision boundaries for classifying binary class data generated from two normal distributions $\mathcal{N}(-1.0, 4.0)$ and $\mathcal{N}(1.0, 0.25)$ using the SVM, CPON, and Bayes classifiers: the classification errors for the SVM, CPON, and Bayes classifiers are determined by 0.1851, 0.1639, and 0.1635, respectively.

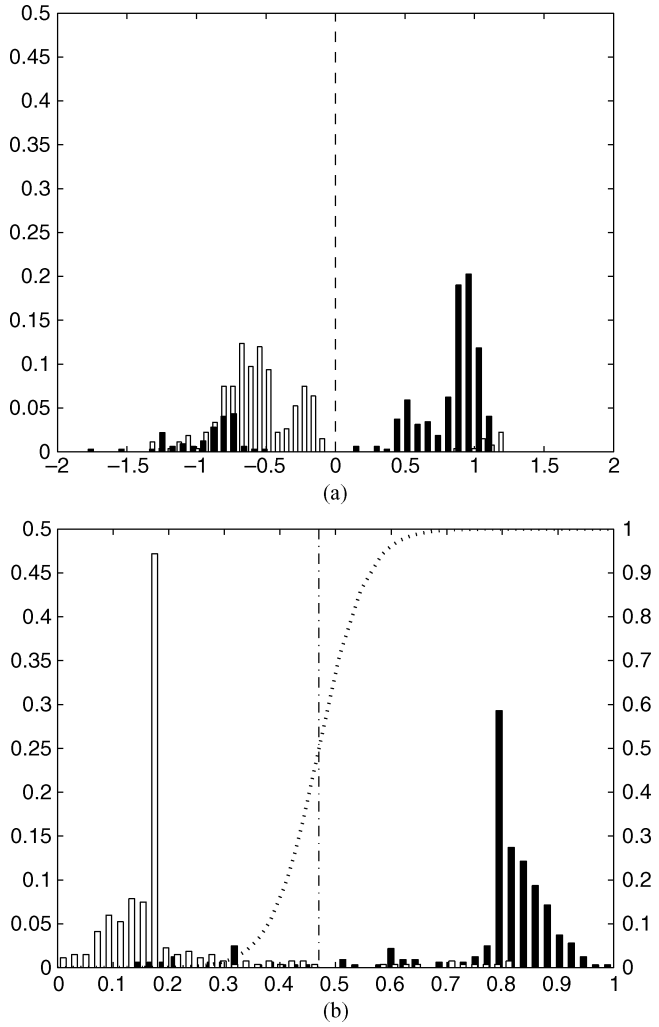


Fig. 3. Comparison of output distributions of the SVM and the CPON for the credit approval data set: (a) and (b) represent the relative frequencies of the output of the SVM and the CPON, respectively. In (a) and (b), the black and white bars represent the relative frequencies of the classifier's output for the samples of "+" and "−" groups, respectively, while the dashed lines represent the decision boundaries of classification. In (b), the dotted line represents the posterior probability of class membership and the scale is shown in the right axis of the plot.

context, the final decision of the class is determined by investigating the class which has the maximum posterior probability of class membership, that is

$$\text{class} = \arg\max_k \Pr\left\{C_k^+ | Y_k^+ \leqslant \bar{y}_k \text{ or } Y_k^- \geqslant \bar{y}_k\right\}. \quad (66)$$

This determination of the class has the meaning that the class $C_k$ with the greatest $p$-value ratio, that is, the relative $p$-value of testing $H_k^+$ compared to the sum of $p$-values of testing the hypotheses $H_k^+$ and $H_k^-$, is selected. In other words, the class $C_k$ with the highest confidence of $H_k^+$ compared to that of $H_k^-$ is selected.

The schematic diagram of the proposed method is illustrated in Fig. 4. In this method, for the given input pattern $\mathbf{x}$ of a $K$ class pattern classification problem, the $K$ binary pattern classifiers $h_k(\mathbf{x})$, $k = 1, \ldots, K$, are constructed using the training method of discriminant functions such as the SVM algorithm. Then, each output of the classifier is normalized between 0 and 1 using (56). Here, the beta CDFs for the "+" and "−" groups are determined according to (22), and posterior probability of each class is determined from these beta CDFs according to (61) or (62). As an example, the estimated beta density functions for the samples of the "+" and "−" groups are illustrated in Fig. 5(a). The estimated beta CDFs for the samples of the "+" and "−" groups are also illustrated in Fig. 5(b). The final form of posterior probability of class membership is calculated from the estimated beta CDFs as illustrated in Fig. 5(b). This plot shows that the shape of posterior probability of class membership resembles the logistic function but it does not exactly match with the logistic form. Finally, the decision of the class is determined by the estimated posterior probability of class membership, that is, the class that has the maximum posterior probability of class membership is selected using (66).

Some attractive features of the proposed method are as follows. 1) There is no need to make an assumption about class output distributions while other probabilistic scaling methods require an assumption on the shape of class probability distribution such as the sigmoid function. 2) The proposed method can be used for an unbalanced data set because the ratio of the number of samples for each class is not important, rather the number of samples itself is important for the accurate estimation of beta distribution parameters. 3) The beta distribution parameters as well as the kernel parameters are adjusted for the accurate estimation of classifier's output distributions and this estimation is evaluated by the uniformity test of the output samples which are mapped from the normalized SVM's output using the estimated beta CDF. As a result, we can obtain the better estimation of posterior probabilities of class membership and this contributes the improvement of the classification performances. This is demonstrated through the simulation for pattern classification problems as described in the next section.

## IV. Simulation for Classification Problems

For the simulation for classification problems, we selected the data sets from the UCI database [15]: they are breast cancer, BUPA liver disorders, credit approval, hepatitis domain, ionosphere, iris, vehicle, and wine recognition. A brief description of the data sets is given in Table II. These data sets were selected since they were quite frequently used to evaluate the performance of classifiers. Overall, for these data sets, the distributions of canonical SVM's output are quite overlapped between the "+" and "−" groups of samples, non-Gaussian, and usually nonunimodal. For example, the distributions of SVM's output for the credit approval data set are quite overlapped and not unimodal as illustrated in Fig. 3(a). Even for these data sets, the CPON is quite well fitted as demonstrated in this simulation for classification problems.

For the classification of these data sets, the suggested method of scaling of classifier's output using beta distribution was applied to the SVM. To see the effect of fine tuning of beta distribution parameters, we have the results of classification performances using the beta scaling method alone (CPON-Beta) and also using the beta scaling and the fine-tuning method (CPON-Beta-FT). For the purpose of comparison, we have the results of classification performances using the SVM, KLR, and SVM with Platt's scaling method (SVM-Platt).

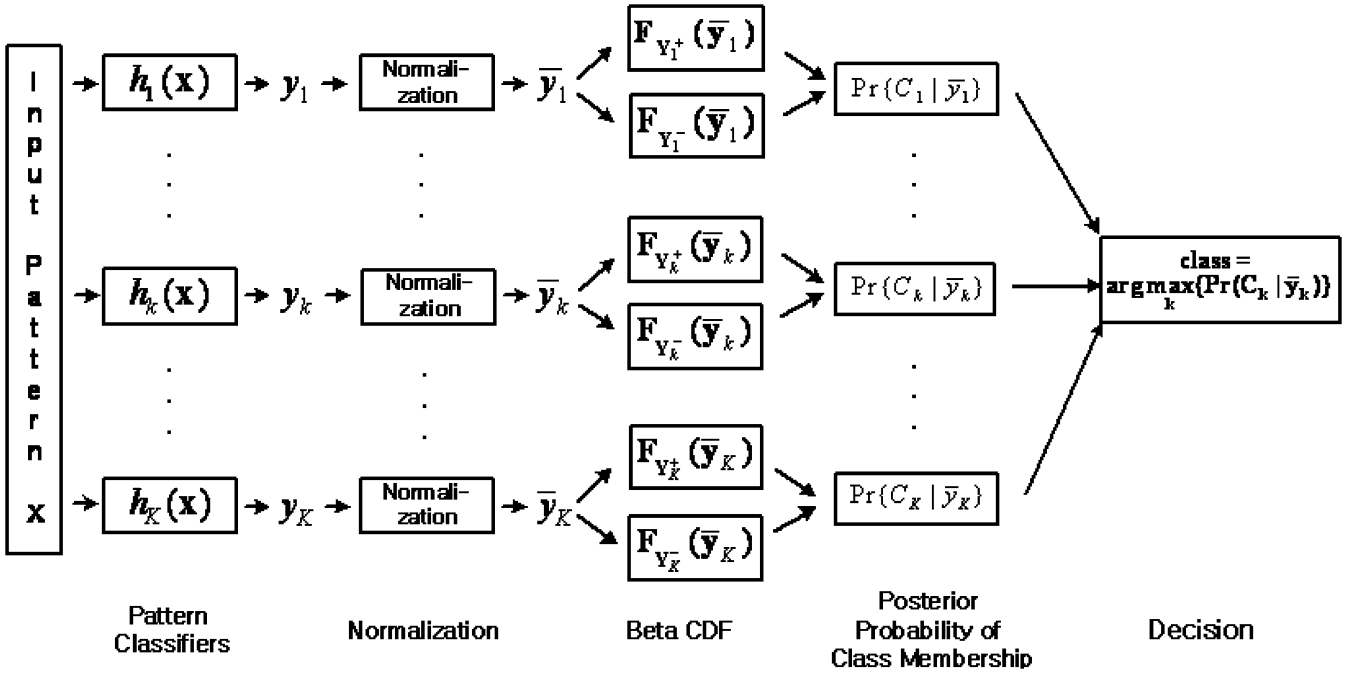


Fig. 4. Pattern classifier with CPON: for the given input pattern $\mathbf{x}$ of a $K$ class pattern classification problem, $K$ binary pattern classifiers $h_k(\mathbf{x}), k = 1, \ldots, K$, are constructed, each output of classifier is normalized between 0 and 1, the beta CDFs for the "+" and "−" groups are determined, the posterior probability of each class is determined from the beta CDFs, and finally, the class that has the maximum posterior probability of class membership is selected.

TABLE II
DESCRIPTION OF THE DATA SETS FROM THE UCI DATABASE

| Data Name | Size of Data | Input Dimension | Number of Classes |
|---|---|---|---|
| Breast Cancer | 699 | 10 | 2 |
| BUPA Liver Disorders | 345 | 6 | 2 |
| Credit Approval | 653 | 15 | 2 |
| Hepatitis Domain | 80 | 19 | 2 |
| Ionosphere | 351 | 34 | 2 |
| Iris | 150 | 4 | 3 |
| Vehicle | 846 | 18 | 4 |
| Wine Recognition | 178 | 13 | 3 |

As improved versions of the SVM method, we also performed the multicategory SVM (MSVM) method [16]–[18] in which the constraint of zero sum of class label values is applied and the $\psi$-learning method [19], [20] in which the $\psi$-function, that is, monotonously increasing function, is applied to the output of SVM. For the learning of MSVM, the class labels of each output is defined in a symmetric way: for $K$ category classification problems, the class label of an instance that belongs to a specific class is defined by the value of 1 and the rest of class labels are defined by the value of $-1/(K-1)$, that is, the total sum of class labels is zero. With this definition of class labels, the quadratic optimization of the sum of loss function and the regularization term of the complexity of discriminant function was performed for the training of MSVM [18]. The $\psi$-learning requires handling a nonconvex minimization problem because of the $\psi$-function applied to the output of SVM. To solve this problem, we can use the difference convex algorithm (DCA) [29], which solves nonconvex minimization via sequential quadratic programming. We can also use the outer approximation method for difference convex optimization suggested by Blanquero and Carrizosa [30]. For our simulation, the DCA was used for the $\psi$-learning.

For all of classifiers, we used Gaussian kernel functions in (4)

$$\phi_{kj}(\mathbf{x}) = \exp\left(-\frac{(\mathbf{x}-\mathbf{x}_j)^2}{2\sigma_k^2}\right)$$

where $\mathbf{x}_j$ represents the center of the $j$th kernel function and $\sigma_k$ represents the kernel width for the $k$th classifier.

For the optimal setting of kernel width $\sigma_k$, we selected the value of $\sigma_k$ from the range of values $2^{i-4}$, $i = 1, \ldots, 7$. The selection was performed by a tenfold cross-validation method in which the training set was split into ten parts, each one of ten classifiers was trained on permutations of nine out of ten parts, and the classifier was evaluated on the remaining tenth part. In the cases of SVM, KLR, and SVM-Platt classifiers, the best value of $\sigma_k$ for minimizing the average error for the validation sets was selected. In the cases of the suggested CPON-Beta and CPON-Beta-FT classifiers, the best value of $\sigma_k$ for maximizing the uniformity of the beta CDF values for the validation sets was selected as described in the learning algorithm of CPON.

For the training of SVM classifiers, we used the statistical pattern recognition toolbox for MATLAB in which the sequential minimal optimization (SMO) function [31] is used to solve the quadratic programming (QP) problem of (10). We also used MATLAB Arsenal [32] for the training of KLR. In the case of SVM-Platt, the parameters of $a$ and $b$ in (16) were estimated using the output of trained SVM. First, as suggested in [12], the output of SVM was transformed to new target probabilities using (17). Then, to avoid the overfitting problem in the estimation of $a$ and $b$, the threefold cross-validation method was used: the training set was split into three parts, each one of three SVMs was trained on permutations of two out of three parts, and the

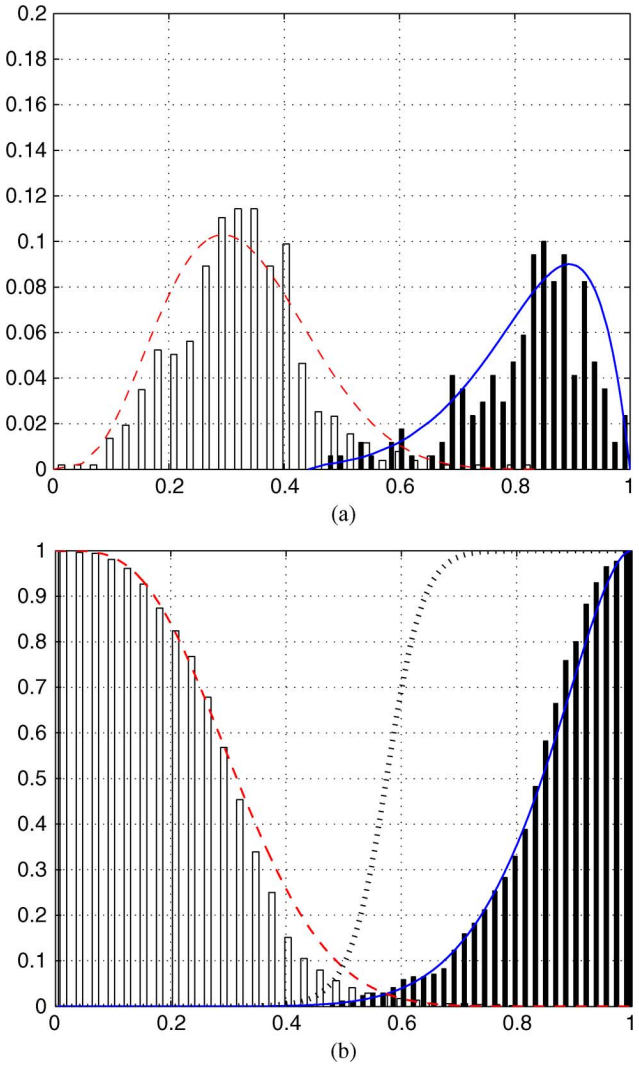


Fig. 5. Estimation of the posterior probability of class membership: (a) and (b) represent the probability density function and posterior probability of class membership using the beta distribution parameters for the vehicle data set, respectively. In (a), the black and white bars represent the relative frequencies of CPON's output for the samples of the "+" and "−" groups, respectively, while solid and broken lines represent the estimated beta density functions for the samples of the "+" and "−" groups, respectively. In (b), the black and white bars represent the cumulative distribution of CPON's output for the samples of the "+" and "−" groups, respectively, while solid and broken lines represent the estimated beta CDFs functions for the samples of the "+" and "−" groups, respectively. The dotted line represents the posterior probability of class membership calculated from the estimated beta CDFs.

SVM was evaluated on the remaining third part. The union of all three sets of evaluated outputs was used as the training samples for the estimation of $a$ and $b$. For the test of this classifier, the scaled output of SVM using (16) was calculated. In the case of CPON-Beta, the best values of kernel width and beta distribution parameters were obtained by maximizing the uniformity of the beta CDF values as described in the learning algorithm of CPON, that is, up to Step 10 of the learning algorithm of CPON.

In the case of CPON-Beta-FT, the procedure of fine tuning of beta distribution parameters, that is, Step 11 of the learning algorithm of CPON in addition to the procedure of CPON-Beta, was performed to improve the uniformity of the beta CDF values. In this tuning of beta distribution parameters, the learning rate $\eta$ was set as $10^{-3}$ and the update rule was repeated until no further change in the beta distribution parameters was found. Then, for the test of this classifier, the posterior probability of class membership was calculated using (62), that is, the assumption of equal "+" and "−" class probabilities was made.

After the training of classifiers, each classifier was evaluated using the tenfold cross-validation method in which the sample set was split into ten parts, each one of ten classifiers was trained on permutations of nine out of ten parts, and the classifier was evaluated on the remaining tenth part as the test samples. For each data set, the average value of test errors for ten classifiers was obtained. The simulation results of average test errors for the UCI data sets are summarized in Table III. The simulation results showed that: 1) the improvement of classification performances of the KLR and SVM-Platt methods compared to the SVM method was dependent upon data sets, that is, in some cases, the classification performances of the KLR and SVM-Platt methods were better than the SVM method and worse in other cases; 2) the suggested CPON-Beta and CPON-Beta-FT method always provided better classification performances than the SVM method; and 3) the CPON-Beta-FT method achieved better or equal classification performances compared to the CPON-Beta method. Consequently, the CPON-Beta-FT method exhibited the top level classification performances compared to other methods. This implies that fine tuning of beta distribution parameters in such a way as to maximize the uniformity of the beta CDF values for the SVM output samples is quite effective to improve the classification performances. To find the performance improvement ratio, we also calculated the measure shown at the bottom of the page. As shown in Table III, the improvement ratio of the suggested method lies between 8.96% and 75.52% for these UCI data sets. These results showed that the suggested classification method using CPON was quite effective to achieve improved classification performances.

The predictive accuracy or error rate is a typical measure of the classification problem; however, most classifiers can also produce probability estimation or confidence of the class prediction. In this context, the receiver operating characteristic (ROC) curve was introduced to evaluate machine learning algorithms [33], [34]. The ROC curves are traditionally graphed by plotting the true-positive fraction on the vertical axis against the false-positive fraction on the horizontal axis. The top-left region of the ROC plane represents good performance, with few false positives and many true positives. The extreme lower left corner represents an observer that classifies all samples as negative, whereas the extreme top-right corner represents an observer that

$$\text{Improvement Ratio} = \frac{\text{Classification Error of SVM} - \text{Classification Error of CPON}}{\text{Classification Error of SVM}}.$$

TABLE III
COMPARISON OF AVERAGE TEST ERRORS FOR THE UCI DATA SETS USING THE SVM, KLR, SVM-PLATT, CPON (WITH BETA SCALING ONLY), AND CPON (WITH BETA SCALING AND FINE TUNING) METHODS

| Data Name | SVM | KLR | SVM-Platt | CPON -Beta | CPON -Beta-FT | Improvement Ratio (%) |
|---|---|---|---|---|---|---|
| Breast Cancer | 0.0346 | 0.0470 | 0.0333 | **0.0315** | **0.0315** | 8.96 |
| BUPA Liver Disorders | 0.3040 | 0.2892 | 0.4587 | 0.2584 | **0.2474** | 18.62 |
| Credit Approval | 0.1403 | 0.2009 | 0.1488 | 0.1250 | **0.1211** | 13.68 |
| Hepatitis Domain | 0.3250 | 0.3000 | 0.5250 | 0.2750 | **0.2250** | 30.77 |
| Ionosphere | 0.0598 | 0.1083 | 0.1594 | **0.0483** | **0.0483** | 19.23 |
| Iris | 0.0533 | 0.0733 | **0.0333** | **0.0333** | **0.0333** | 37.52 |
| Vehicle | 0.2203 | 0.2687 | 0.2560 | 0.2054 | **0.1835** | 16.70 |
| Wine Recognition | 0.0727 | 0.0452 | 0.1591 | 0.0235 | **0.0178** | 75.52 |

TABLE IV
COMPARISON OF AUC VALUES FOR THE BINARY CLASSIFICATION PROBLEMS OF THE UCI DATA SETS

| Data Name | SVM | KLR | SVM-Platt | CPON -Beta | CPON -Beta-FT |
|---|---|---|---|---|---|
| Breast Cancer | 0.9739 | 0.9553 | 0.9692 | **0.9755** | **0.9755** |
| BUPA liver Disorders | 0.6694 | 0.6616 | 0.5769 | 0.7178 | **0.7188** |
| Credit Approval | 0.8652 | 0.7919 | 0.8483 | 0.8732 | **0.8749** |
| Hepatitis Domain | 0.7679 | 0.7921 | 0.5842 | 0.7921 | **0.7988** |
| Ionosphere | 0.9354 | 0.8727 | 0.8666 | **0.9517** | **0.9517** |

classifies all samples as positive. However, sometimes there is no clear dominating relation between two ROC curves in the entire range. In this sense, the area under the ROC curve, or simply the area under the curve (AUC), provides a single-number summary for the performance of the learning algorithms. The AUC can be considered as a measure for the discriminative ability of a test. Tests with good performance have an ROC curve that is close to the top-left corner, with an area that is close to 1. In general, the area under an ROC curve can range from 0 to 1 and the area under the diagonal is 0.5. This implies that any ROC curve with an area less than 0.5 is usually classified as particularly bad. From this point of view, we also made the simulation results of AUC values for binary classification data sets. These simulation results are summarized in Table IV. As shown in these simulation results, the suggested CPON-Beta and CPON-Beta-FT methods provided better AUC values than other classification methods. This implies that the distribution of the classifier's output using the CPON method is well formulated for the decision of classification problems so that the classification with few false positives and many true positives is possible compared to other methods.

As alternative methods of SVM, we also performed the simulation of classification for UCI data sets using the MSVM and $\psi$-learning methods and compared these simulation results with the canonical SVM and the suggested CPON (-Beta-FT) methods. To test the effectiveness of the suggested beta-scaling and fine-tuning method, that is, Steps 3 and 5 of the learning algorithm of CPON, we also performed the application of the suggested method to the SVM, MSVM, and $\psi$-learning methods after training these classifiers: they are the SVM-Beta-FT, MSVM-Beta-FT, and $\psi$-learning-Beta-FT methods, respectively. The simulation results of average test errors for the UCI data sets are summarized in Table V. The simulation results showed that: 1) in comparison to the canonical SVM, the MSVM method provided improved classification performances in the multicategory classification problems such as the vehicle and wine recognition data sets while the $\psi$-learning method provided improved classification performances in both binary and multicategory classification problems except the cases of credit approval and ionosphere data sets; 2) in all cases, the classification performances were improved after applying the suggested beta-scaling and fine-tuning method; 3) the MSVM-Beta-FT method provided better classification performances in multicategory classification problems compared to the SVM-Beta-FT and $\psi$-learning-Beta-FT methods and the opposite phenomenon was observed in the binary classification problems; and 4) the suggested CPON method provided the top level classification performances compared to other methods. These simulation results imply that the suggested beta-scaling and fine-tuning methods are also quite effective for other learning methods such the MSVM and $\psi$-learning methods and the CPON method in which the kernel selection method maximizing the $p$-value of testing the uniformity of the CDF of beta-scaled output samples in addition to the beta-scaling and fine-tuning methods is quite effective for classification problems.

We have shown that the suggested CPON method provides improved performances compared to the SVM, MSVM, and $\psi$-learning classifiers, and also more effective than other probabilistic scaling methods. This is possible because: 1) the distribution of the classifier's output can be effectively approximated in a flexible manner by adjusting the beta distribution parameters while other probabilistic scaling methods have the fixed form of distribution such as the sigmoid function; 2) the estimation of classifier's output distributions is performed accurately by controlling the beta distribution parameters in such a way as to have a uniform distribution of beta CDF values for the given class output samples; 3) the kernel parameters are also selected in such a way as to contribute the estimation of classifier's output distributions accurately; and 4) the final decision is made using the estimated posterior probabilities of class membership. Furthermore, the suggested CPON method can be easily applied

TABLE V
COMPARISON OF AVERAGE TEST ERRORS FOR THE UCI DATA SETS USING THE SVM, SVM-BETA-FT, MSVM, MSVM-BETA-FT, $\psi$-LEARNING, $\psi$-LEARNING-BETA-FT, AND CPON METHODS

| Data Name | SVM | SVM -Beta-FT | MSVM | MSVM -Beta-FT | $\psi$-Learning | $\psi$-Learning -Beta-FT | CPON |
|---|---|---|---|---|---|---|---|
| Breast Cancer | 0.0346 | 0.0330 | 0.0646 | 0.0646 | 0.0330 | 0.0330 | **0.0315** |
| BUPA Liver Disorders | 0.3040 | 0.2547 | 0.3513 | 0.3312 | 0.2854 | 0.2513 | **0.2474** |
| Credit Approval | 0.1403 | **0.1211** | 0.1502 | 0.1436 | 0.1456 | 0.1325 | **0.1211** |
| Hepatitis Domain | 0.3250 | 0.3000 | 0.3125 | 0.3000 | 0.3089 | 0.2876 | **0.2250** |
| Ionosphere | 0.0598 | 0.0511 | 0.1161 | 0.0975 | 0.0764 | 0.0654 | **0.0483** |
| Iris | 0.0533 | 0.0533 | 0.0533 | 0.0400 | 0.0476 | 0.0450 | **0.0333** |
| Vehicle | 0.2203 | 0.1984 | 0.2011 | 0.1873 | 0.2136 | 0.1977 | **0.1835** |
| Wine Recognition | 0.0727 | 0.0727 | 0.0355 | 0.0235 | 0.0426 | 0.0257 | **0.0178** |

to any type of classifiers using discriminant functions by collecting the data from the classifier's output and estimating the beta distribution parameters.

## V. CONCLUSION

In this paper, a new method of scaling the classifier's output was presented for the purpose of estimating the posterior probability of class membership. For this purpose, the classifier's output was analyzed and the distribution of the output was described by the beta distribution parameters. For more accurate approximation of class output distribution, the beta distribution parameters as well as the kernel parameters describing the discriminant function were adjusted in such a way as to improve the uniformity of beta CDF values for the given class output samples. As a result, the classifier with the proposed scaling method referred to as the class probability output network (CPON) can provide accurate posterior probabilities for the soft decision of classification. To show the effectiveness of the proposed method, the simulation for pattern classification using the SVM classifier was performed for the UCI data sets. The simulation results showed that the proposed CPON method provided the improved classification performances compared to the SVM and SVM-related classifiers, and it was more effective than other probabilistic scaling methods. The main reasons for the effectiveness of the CPON are as follows: 1) the distribution of the classifier's output can be approximated effectively in a flexible manner by adjusting the beta distribution parameters while other probabilistic scaling methods have the fixed form of distribution; 2) the kernel parameters are also adjusted in such a way as to improve the uniformity of beta CDF values for the given class output samples; and 3) the final decision of classification is made according to the estimated posterior probabilities of class membership. Furthermore, the proposed learning algorithm of CPON can be easily applied to various types of classifiers using discriminant functions for the substantial improvement of classification performances.

## REFERENCES


[1] K. Fukunaga and D. Kessell, "Nonparametric bayes error estimation using unclassified samples," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 434–439, Jul. 1973.

[2] N. Glick, "Additive estimators for probabilities of correct classification," *Pattern Recognit.*, vol. 10, pp. 211–222, 1978.

[3] J. Kittler and P. Devijver, "An efficient estimator of pattern recognition system error probability," *Pattern Recognit.*, vol. 13, pp. 245–249, 1981.

[4] A. G. Curieses, J. C. Sueiro, R. A. Rodriguez, and A. R. Vidal, "Local estiamtion of posterior class probabilities to minimize classification errors," *IEEE Trans. Neural Netw.*, vol. 15, no. 2, pp. 309–317, Mar. 2004.

[5] E. Parzen, "On the estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.

[6] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[7] G. Wahba, "Advances in kernel methods," in *Support Vector Learning, Reproducing Kernel Hilbert Spaces and the Randomized GACV*. Cambridge, MA: MIT Press, 1999, pp. 69–88.

[8] T. S. Jaakkola and D. Haussler, "Probabilistic kernel regression models," in *Proc. Conf. Artif. Intell. Statist.*, 1999.

[9] V. Roth, "Probabilistic discriminative kernel classifiers for multi-class problems," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2001, vol. 2191, pp. 246–253.

[10] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine," *J. Comput. Graph. Statist.*, vol. 14, no. 1, pp. 185–205, 2005.

[11] G. Wahba, "Multivariate function and operator estimation, based on smoothing splines and reproducing kernels," in *Proceedings of the Sciences of Complexity*, M. Casdagli and S. Eubandk, Eds. Reading, MA: Addison-Wesley, 1992, vol. 12, pp. 95–112.

[12] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large-Margin Classifers*, P. J. Bartlett, B. Schölkopf, D. Schuurmans, and A. Smola, Eds. Cambridge, MA: MIT Press, 2000, pp. 61–74.

[13] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *Anal. Statist.*, vol. 26, no. 2, pp. 451–471, Apr. 1998.

[14] V. K. Rohatgi and A. K. M. E. Saleh, "Nonparametric statistical Inference," in *An Introduction to Probability and Statistics*, 2nd ed. New York: Wiley-Interscience, 2001, pp. 608–633.

[15] P. M. Murphy and D. W. Aha, UCI Repository of Machine Learning Databases, Univ. California at Irvine, Irvine, CA, 1994 [Online]. Available: http://www.archive.ics.uci.edu/ml/

[16] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data," *J. Amer. Statist. Assoc.*, vol. 99, pp. 67–81, 2004.

[17] Y. Lee, G. Wahba, and S. Ackerman, "Classification of satellite radiance data by multicategory support vector machines," Dept. Statist., Univ. Wisconsin-Madison, Madison, WI, Tech. Rep. 1075, Feb. 2003.

[18] Y. Lee, Multicategory Support Vector Machines (MSVM), Sep. 2003 [Online]. Available: http://www.stat.wisc.edu/~yklee/sharecode.html

[19] X. Shen, G. Tseng, X. Zhang, and W. Wong, "On $\psi$-learning," *J. Amer. Statist. Assoc.*, vol. 98, no. 462, pp. 724–734, 2003.

[20] Y. Liu, X. Shen, and H. Doss, "Multicategory $\psi$-learning and support vector machine: Computational tools," *J. Amer. Statist. Assoc.*, vol. 14, pp. 219–236, 2005.

[21] D. Luenberger, *Optimization by Vector Space Method*. New York: Wiley, 1969.

[22] M. Guignard, "Generalized kuhn-tucker conditions for mathematical programming problems in banach space," *SIAM J. Control*, vol. 7, pp. 232–241, 1969.

[23] A. B. Tal and J. Zowe, "A unified theory of first and second order conditions for extremum problems in topological vector space," *Math. Programm. Study*, vol. 19, pp. 39–76, 1982.

[24] T. Hastie, R. Tibshirani, and J. Friedman, "Linear methods for classification," in *The Elements of Statistical Learning*. New York: Springer-Verlag, 2002, pp. 95–105.

[25] P. M. Wiliams, "Bayesian regularization and pruning using a Laplace prior," *Neural Comput.*, vol. 7, pp. 117–143, 1995.

[26] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2001, vol. 2111, pp. 416–426.
[27] D. Luenberger, *Introduction to Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1973.
[28] R. Gnanadeskan, R. S. Pinkham, and L. P. Hughes, "Maximum likelihood estimation of the parameters of the beta distribution from smallest order statistics," *Technometrics*, vol. 9, pp. 607–620, 1967.
[29] L. T. An and P. D. Tao, "Solving a class of linearly constrained indefinite quadratic problems by d. c. Algorithms," *J. Global Optim.*, vol. 11, pp. 253–285, 1997.
[30] R. Blanquero and E. Carrizosa, "On covering methods for d. c. optimization," *J. Global Optim.*, vol. 18, pp. 256–274, 2000.
[31] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods-Support Vector Learning*, B. Schölkopf, C. J. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 185–208.
[32] Y. Rong, A MATLAB Package for Classification Algorithms Mar. 2006 [Online]. Available: http://www.informedia.cs.cmu.edu/yanrong/MATLABArsenal/MATLABARsenal.htm
[33] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance comparison under imprecise class and cost distribution," in *Proc. 3rd Int. Conf. Knowl. Disc. Data Mining*, 1997, pp. 43–48.
[34] F. Provost, T. Fawcett, and R. Kohavi, "The case accuracy estimation for comparing induction algorithms," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 445–453.

**Woon Jeung Park** received the B.S. degree from the Department of Mathematics, Soongsil University, Seoul, Korea, in 2002 and the M.S. degree from the Department of Applied Mathematics, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2005, where she is currently working towards the Ph.D. degree at the Department of Mathematical Sciences.

Her research interests include machine learning, statistical learning theory, and data mining.

**Rhee Man Kil** (M'94–SM'09) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1979 and the M.S. and Ph.D. degrees in computer engineering from the University of Southern California, Los Angeles, in 1985 and 1991, respectively.

Currently, he is an Associate Professor at the Department of Mathematical Sciences, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. His general research interests lie in theories and applications of machine learning. His current interests focus on pattern classification, model selection in regression problems, text mining, financial data mining, noise-robust feature extraction, and binaural speech segregation and recognition.