Class Probability Output Network: a New Paradigm of Learning Based on Beta Distribution

Rhee Man Kil

College of Information and Communication Engineering

Sungkyunkwan University

Contents

Motivation

- There are various ways of implementing pattern classification: the most popular way is to use the discriminant function whose value is supposed to indicate **the degree of confidence for the classification.**
- The natural way of representing the degree of confidence is to use **the posterior probability for the decision of classification.**
- There are some methods that estimates the posterior probabilities: Parzen window, kernel logistic regression (KLR), relevance vector machine (RVM), etc.

Motivation

- These methods of estimating posterior probabilities require enough number of samples and kernel functions for the accurate estimation of data distribution.
- Furthermore, these methods require high computational complexity for the training even though the classification performances are usually less than those of classifiers using discriminant function such as the support vector machine (SVM).
- In this context, we consider a new method of estimating the conditional class probabilities using the beta distribution referred to as the **class probability output network (CPON).**

Motivation

- The suggested CPON provides **accurate estimation of conditional class probabilities and better performances** over the SVM/SVM-related methods and other probabilistic scaling methods.

- Furthermore, the CPON also provides the **confidence intervals for the conditional class probabilities** which are important to resolve the ambiguity of the decision of classification.

Discriminant Functions

- Pattern classifiers provide an output $\hat{y}$ as a discriminant value for the given input pattern x.

- Suppose we have K classes. We construct the K discriminant functions for each class:

$$\hat{y}_k = h_k(x), \ \ for\ k \ = \ 1, \ .., \ K.$$

- The decision is made by the maximum discriminant such as

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

- Linear discriminant functions are thoese of the form

$$h(x) = \sum_{j=1}^{d} w_j x_j$$

where d is the dimension of the intput pattern x.

Discriminant Functions

- In general, the discriminant function h_k can be constructed as a linear combination of kernel functions:

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

where m_k, w_{kj}, and ϕ_{kj} represent the number of kernel functions, jth weight value, and jth kernel function for the kth discriminant function respectively.

- The goal of learning is to construct an estimation function h_k that minimizes the expected risk

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

where y_k represents the target value for the kth class and $L(y_k,\ h_k(\mathbf{x}))$ represents a loss functional.

Discriminant Functions

- For the classification problem, the loss functional L is given by

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

- To minimize the expected risk of (3), it is necessary to identify the distribution $P(\mathbf{x},\ y_k)$.

- However, we can't know the distribution $P(\mathbf{x},\ y_k)$.

- We find h_k by minimizing the empirical risk $R_{emp}(h_k)$ evaluated by the mean of loss function values for the given samples, that is,

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

where y_{ki} represents the ith output sample for the kth class.

Discriminant Functions

- If the number of parameters of a classifier is exceedingly small compared with the number of data, then the classification performance may not be optimal; that is, **under-fitting** occurs.
- If the number of parameters of a classifier is excessively large compared with the number of data, then the classification performance may not be optimal; that is, **over-fitting** occurs.
- One of good ideas is to use the **structural risk minimization** (**SRM**) principle.

Support Vector Machines

- Let us consider the discriminant function for binary classification with the additional bias term w_0.

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

where m is the num. of kernel functions and K is the Mercer's 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 $y_i \cdot h(x_i) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1,..,n$

where C is the positive regularization constant and ξ_i is the slack variable for the ith pattern.

Support Vector Machines

- The learning problem is equivalent to solving the dual problem:

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

subject to $0 \le \alpha_i \le C,\ i = 1,\ldots,n,$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

where α_i represent the Lagrangian multiplier for the ith sample.

Support Vector Machines

- After learning, the support vectors provide the **sparse representation of data.**
- In many classification problems, the SVM provides good performances due to the SRM principle.
- However, the output of SVM does not necessarily mean the degree of confidence for the decision of classification.
- Furthermore, in the case of **unbalanced data distributions**, the SVM does not provide the optimal hyperplane for the classification.

Kernel Logistic Regression

- In the logistic regression, the posterior probability of the class memebership via the linear discriminant function is obtain.

- The kernel logistic regression (KLR) is the kernelized version of logistic regression technique to solve the nonlinear classification problems.

- $h(\mathbf{x}) = \text{logit}(P(y=1 \mid \mathbf{x})) = \log \dfrac{P(y=1 \mid \mathbf{x})}{1 - P(y=1 \mid \mathbf{x})}$. (11)

- To choose the optimal parameters of $w_i s$ in (5), the following cost function representing the regularized negative log likelihood of the data is minimized:

$$L(w) = -\sum_{i=1}^{n} (y_i \log h(x_i) + (1-y_i)\log(1-h(x_i))) + \lambda \|w\|^2 \qquad (12)$$

where λ is a regularization parameter controlling the bias variacne trade-off.

Kernel Logistic Regression

- The KLR provides the class probabilities using the logistic function.
- The classification performance is similar with that of SVM.
- However, the logistic function itself does not provide the accurate estimation of condition class probabilities.
- Furthermore, the training of KLR requires **high computational complexity** compared with the SVM method.

Scaling Functions

- The output of 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 $P(y \mid x)$ as a func -tion of the output of the SVM, i.e., $P(y=1 \mid h(x)) = \sigma(h(x))$ with an appro -priate scaling function σ.

1. Soft-max scaler

$$P(y=1 \mid h) = \frac{1}{1+\exp(-h)} \quad (13)$$

where h represents the output of SVM.

Scaling Functions

2. Hastie and Tibshirani

- $\Pr(y=1 \mid h) = \dfrac{1}{1+\exp(ah^2+bh+c)}$ (14)

where a, b, and c represent variables trained from the SVM's outputs.

- The bias of sigmoid is adjusted so that the point $\Pr(y=1 \mid h) = 0.5$ occurs at $h=0$.

3. Platt

- $\Pr(y=1 \mid h) = \dfrac{1}{1+\exp(ah+b)}$ (15)

where a and b represent variable strained from the SVM's outputs.

- These parameters are fit using MLE from the training set(h_i, y_i), $i=1,\ldots,n$ where h_i and y_i arepresent the SVM's output and the target value for the ith samples repectively.

Scaling Functions

- The scaling function itself does not provide the accurate estimation of conditional probabilities since the distribution of classifier's output usually does not fit with the scaling function.
- On the average, the classification performance of the scaling method is almost same as that of the original classifier.

Beta Distribution

- For the modeling of classifier's output distribution, a beta distribution is used since it is a conjugate prior of binomial distribution; that is, it represents the **distribution of probabilities for the binary classification problem**.
- Furthermore, the beta distribution is a good model for the data within a finite range assuming that they have an **unimodal distribution**.

Beta Distribution

- From a view point of classification performances, it is favorable that the classifier's output has an unimodal distribution rather than a multimodal distribution.

- In this context, the **beta distribution parameters as well as the classifier's parameters** are adjusted in such a way that the classifier's output representing the conditional class probability has a beta distribution.

Beta Distribution

- A random variable X has a binomial distribution with parameters (n, p) if its probability mass function is given by

$$p_X(i) = \Pr\{X = i\} = \binom{n}{i} p^i (1-p)^{n-i}, \quad i = 0, 1, \cdots, n$$

where p represents the probability of success.

- $X \sim B(n, p)$ represents the number of successes in n independent trials.

- Question: What is the distribution of p?

Beta Distribution

Let X be a probability of binomial distribution.

Then,

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} = \frac{\Gamma(n+1)}{\Gamma(i+1)\Gamma(n-i+1)}$$

where Gamma function is defined by

$$\Gamma(\alpha) = \int_0^\infty e^{-y} y^{\alpha-1} dy.$$

If α is an integer, $\Gamma(\alpha) = (\alpha - 1)!$

Beta Distribution

Thus, the PDF of X is given by

$$f_X(x) = \frac{\Gamma(n+1)}{\Gamma(i+1)\Gamma(n-i+1)} x^i (1-x)^{n-i}, \quad 0 \leqq x \leqq 1.$$

Let $\alpha = i+1$ and $\beta = n-i+1$.

Then, the PDF of X becomes

$$f_X(x) = \frac{\Gamma(\alpha+\beta-1)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}, \quad 0 \leqq x \leqq 1.$$

Beta Distribution

However, the Beta function is given by

$$B(\alpha,\beta)=\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}.$$

Therefore, after rescaling the PDF of X, we get the Beta PDF described by

$$f_X(x)=\frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1},\quad 0\leqq x\leqq 1.$$

Beta Distribution

- The beta(α,β) PDF is

$$f(y\mid\alpha,\beta)=\frac{1}{B(\alpha,\beta)}y^{\alpha-1}(1-y)^{\beta-1},\quad 0<y<1,\quad \alpha>0,\quad \beta>0,$$

where $B(\alpha,\beta)$ denotes the beta function, $B(\alpha,\beta)=\int_0^1 y^{\alpha-1}(1-y)^{\beta-1}dy.$

- The beta function is related to the gamma function:

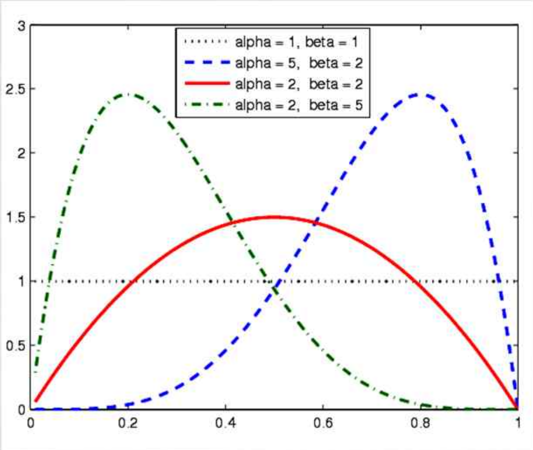
$$B(\alpha,\beta)=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)},\quad \Gamma(\alpha)=\int_0^\infty \lambda e^{-\lambda y}(\lambda y)^{\alpha-1}dy,\quad \lambda>0.$$

- The beta(α,β) CDF is

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

Beta Distribution

- As the parameters α and β vary, the beta distribution takes on many shapes.



Beta Distribution

- We calculate the mean and variance of the beta(α,β) distribution as

$$E(Y)=\frac{\alpha}{\alpha+\beta} \quad \text{and} \quad Var(Y)=\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

- From these results, we can obtain the parameters of the beta distribution:

$$\alpha=E(Y)\left(\frac{E(Y)(1-E(Y))}{Var(Y)}-1\right),\ \beta=(1-E(Y))\left(\frac{E(Y)(1-E(Y))}{Var(Y)}-1\right).$$

Estimation of Beta Parameters

- Aforementioned moment matching method is simple to estimate parameters. However, it requires enough number of samples for the accurate estimation of parameters.
- For smaller number of samples, maximum likelihood estimation (MLE) is an alternative choice.
- In our approach, we consider the method of adjusting parameters in such a way that **the CDF values have an uniform distribution**.

Uniform Distribution of CDF data

- If the estimation of PDF for the given data is accurate, then
the CDF for the given data $F_Y(Y_i),\ i=1,\ldots,n$ becomes uniformly distributed:
gLet us consider a random variable $u_i = F_Y(Y_i),\ i=1,\ldots,n$; then
gthe probability density function for a random variable u,

$$f_U(u) = \frac{f_Y(y)}{|dF_Y/dy|} = \frac{f_Y(y)}{|f_Y(y)|} = 1 \quad \text{where } u = F_Y(y).$$

Kolmogorov-Smirnov test

- We need to check the uniformity of $F_Y(Y_i),\ i=1,\dots,n$ using the Kolmogorov-Smirnov test (K-S test).

- The procedure of the K-S test

gLet $S_Y = \{Y_i \mid i=1,\dots,n\}$ be a sample from the cumulative distribution function F_Y, and let F_n^* be the corresponding empirical cumulative distribution function.

gThe K-S statistic D_n is defined by $D_n = \sup_{y\in S_Y} | F_n^*(y) - F_Y(y) |$.

gIn the case of uniform distribution, $F_Y(y) = y$.

gThe K-S statistic D_n can be used to test the following hypotheses:

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

Kolmogorov-Smirnov test

- To test hypotheses, we can calculate the p-value from the K-S distribution:

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

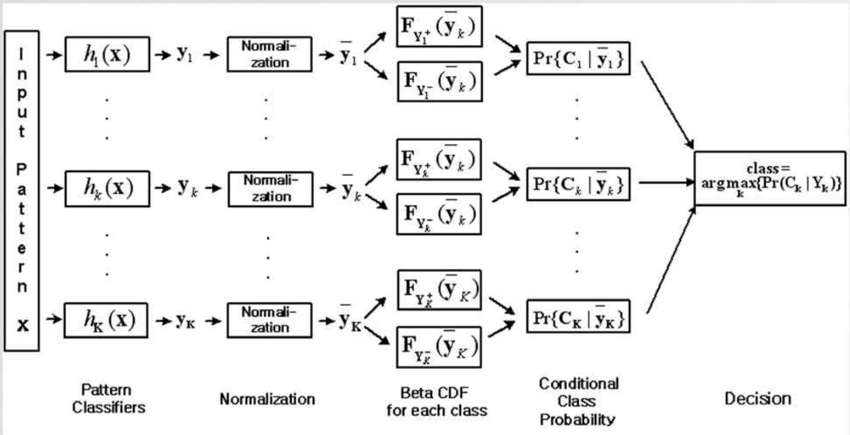
where t represents a variable defined by $t = \sqrt{n}\,y$ and $H(t)$ represents the CDF of the K-S statistic determined by

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

- Finally, the testing of the hypotheses with a significance level δ is to accept H_0, if p-value $\geq \delta$; reject H_0, otherwise.

- If the hypothesis test of uniform distribution is failed, we consider the fine tuning of α and β to improve the uniformity of $F_Y(Y_i),\ i=1,\dots,n$.

Class Probability Output Network



Pattern Classification with CPON

Class Probability Output Network

- We consider the following form of conditional class probability:

$$\Pr\left\{C_k^+ \mid Y_k^+ \le \bar{y}_k \text{ and } Y_k^- \ge \bar{y}_k\right\} = \frac{\Pr\left\{Y_k^+ \le \bar{y}_k \mid C_k^+\right\}\Pr\{C_k^+\}}{\Pr\left\{Y_k^+ \le \bar{y}_k \mid C_k^+\right\}\Pr\{C_k^+\} + \Pr\left\{Y_k^- \ge \bar{y}_k \mid C_k^-\right\}\Pr\{C_k^-\}}$$

$$= \frac{F_{Y_k^+}\left(\bar{y}_k\right)\Pr\{C_k^+\}}{F_{Y_k^+}\left(\bar{y}_k\right)\Pr\{C_k^+\} + \left(1 - F_{Y_k^-}\left(\bar{y}_k\right)\right)\Pr\{C_k^-\}}.$$

$$= \frac{F_{Y_k^+}\left(\bar{y}_k\right)}{F_{Y_k^+}\left(\bar{y}_k\right) - F_{Y_k^-}\left(\bar{y}_k\right) + 1}$$

- class = $\arg\max_k \Pr\left\{C_k^+ \mid Y_k^+ \le \bar{y}_k \text{ and } Y_k^- \ge \bar{y}_k\right\}$.

Class Probability Output Network

- Let us consider the p-values of testing hypotheses H_k^+ and H_k^- :

p-value of testing $H_k^+ = \Pr\{Y_k^+ \le \bar{y}_k\} = F_{Y_k^+}(\bar{y}_k)$ and

p-value of testing $H_k^- = \Pr\{Y_k^- \ge \bar{y}_k\} = 1 - F_{Y_k^-}(\bar{y}_k)$.

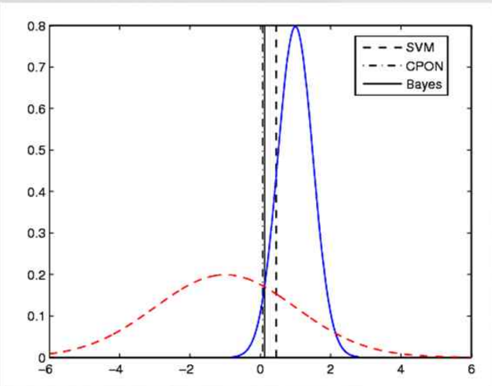
where H_k^+ and H_k^- represent the hypotheses that the given instance belongs to C_k^+ and C_k^-, respectively.

- $\Pr\left\{C_k^+ \mid Y_k^+ \le \bar{y}_k \text{ or } Y_k^- \ge \bar{y}_k\right\} = \dfrac{\Pr\left\{\text{error at } \bar{y}_k,\ C_k^+\right\}}{\Pr\left\{\text{error at } \bar{y}_k\right\}}$

$$= \frac{p\text{-values of testing hypotheses } H_k^+}{p\text{-values of testing hypotheses } H_k^+ + p\text{-values of testing hypotheses } H_k^-}$$

- class $= \underset{k}{\operatorname{argmax}} \Pr\{C_k^+ \mid Y_k^+ \le \bar{y}_k \text{ or } Y_k^- \ge \bar{y}_k\}$.

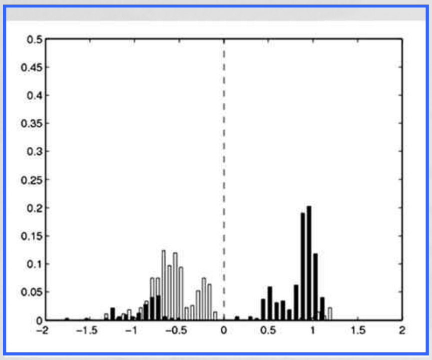
Class Probability Output Network



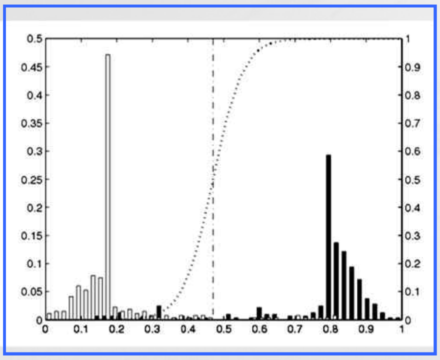
- Train sample: 200 samples :
  100 samples ~ $N(-1,\ 4)$.
  100 samples ~ $N(1,\ 1/4)$.

- Error rate: SVM: 0.1851
  CPON: 0.1639
  Bayes: 0.1635.

Class Probability Output Network



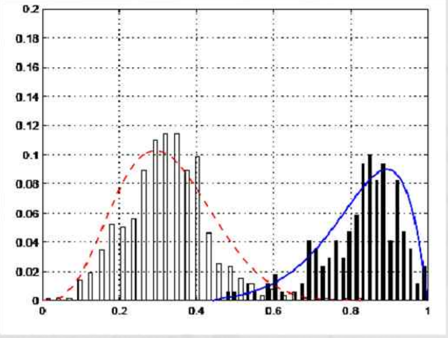
SVM

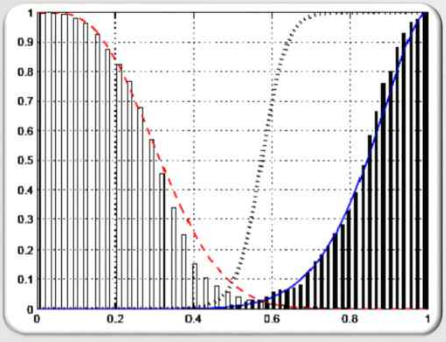


CPON

Comparison of output distributions of the SVM and the CPON for the credit approval data set.

Class Probability Output Network





Estimation of the probability density function and cumulative distribution function

Characteristics of CPON

- The suggested CPON provides conditional class probabilities for the soft decision of classification.
- In the training CPON, the parameters of a classifier are adjusted in such a way that **the distribution of classifier's output fits with the ideal distribution** (in this case, the beta distribution), not just a risk function.

Simulation

- For the simulation for classification problems, we selected 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

Description of the data sets from UCI database

Simulation

- The suggested method (CPON) was applied to the SVM.
- To see the effect of fine tuning of beta parameters, we compared the performance of classification results SVM-Beta and SVM- Beta with FT.
- We made the performaces of classification results using SVM, KLR with Platt's scaling method (SVM-Platt).
- For all of classifiers, we used Gaussian kernel functions

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

where x_j represents the center of the jth kernel function and σ_k represents the kernel width for the kth classifier.

Simulation

- For the optimal setting of kernel width σ_k, we selected the value of σ_k from the range of values, $2^{i-4}, i = 1,...,7$.
- The Receiver Operating Characteristic (ROC) curve is the graphed by plotting the true-positive fraction in the vertical axis and the false-positive fraction on the horizontal axis.
- The top-left region of the ROC plane represent good performance, with few false positive and many true positives.
- The area under the ROC curve (AUC), provides a single-number summary for the performance of the learning algorithms.
- The performance measure: Error rate and Area Under the Curve(AUC).

Simulation

- To find the performance improvement ratio, we calculate the following measure:

$$\text{Improvement Ratio} = \frac{\text{Classification Error of SVM - Classification Error of SVM-Beta with FT.}}{\text{Classification Error of SVM.}}$$

- After the training of classifiers, each classifier was evaluated using the 10-fold cross-validation method.

Simulation Results

Data Name	SVM	KLR	SVM-Platt	SVM-Beta	SVM-Beta with 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

Comparison of average test errors for UCI data sets using the classification methods of SVM, KLR, SVM-Platt, SVM-Beta, SVM-Beta with FT

Simulation Results

Data Name	SVM	KLR	SVM-Platt	SVM-Beta	SVM-Beta with 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**

Comparison of AUC values for the binary classification problems of UCI data sets

Accuracy of the CPON Output

In the K-S test, a critical value of a test statistic is given by D_α such that

$$\Pr\{D_n > D_\alpha\} = \alpha \quad \text{or} \quad \Pr\{\sqrt{n}\, D_n > D_\alpha\},$$

where α represents the level of significance

We treat $K_n = \sqrt{n}\, D_n$ and $K_\alpha = \sqrt{n}\, D_\alpha$.

Then, from the K-S statistic,

$$x = K_\alpha,$$

$$\Pr\{K_n > x\} = 1 - \frac{\sqrt{2\pi}}{x} \sum_{i=1}^{\infty} e^{-(2i-1)^2\pi^2/(8x^2)} = \alpha.$$

Accuracy of the CPON Output

The confidence intervals for the output of positive and negative classes are determined as follows:

with a probability of $1-\alpha$,

$$\widehat{F_k^+}(\overline{y_k}) - D_{\alpha,k}^+ \leq F_k^+(\overline{y_k}) \leq \widehat{F_k^+}(\overline{y_k}) + D_{\alpha,k}^+ \text{ and}$$

$$(1-\widehat{F_k^-}(\overline{y_k})) - D_{\alpha,k}^- \leq F_k^-(\overline{y_k}) \leq (1-\widehat{F_k^-}(\overline{y_k})) + D_{\alpha,k}^-,$$

where $\widehat{F_k^+}(\overline{y_k})$ and $\widehat{F_k^-}(\overline{y_k})$ represent the empirical CDF obtained from the output of positive and negative classes, respectively.

Since the CPON output of the kth class is determined by

$$\Pr\{C_k^+|\overline{y_k}\} = \frac{F_k^+(\overline{y_k})}{F_k^+(\overline{y_k}) + 1 - F_k^-(\overline{y_k})},$$

Accuracy of the CPON Output

With a probability of $(1-\alpha)^2 \approx 1-2\alpha$, the true conditional class probability of the kth class $F_k(\overline{y_k})$ lies within the following range:

$$\frac{\widehat{F_k^+}(\overline{y_k}) - D_{\alpha,k}^+}{\widehat{F_k^+}(\overline{y_k}) - D_{\alpha,k}^+ + 1 - \widehat{F_k^+}(\overline{y_k}) + D_{\alpha,k}^-} \leq F_k(\overline{y_k})$$

$$\leq \frac{\widehat{F_k^+}(\overline{y_k}) + D_{\alpha,k}^+}{\widehat{F_k^+}(\overline{y_k}) + D_{\alpha,k}^+ + 1 - \widehat{F_k^+}(\overline{y_k}) - D_{\alpha,k}^-}.$$

- The suggested confidence interval of CPON output can be used to **identify the possible misclassification for the given pattern.**

- Further improvement is possible by investigating the possible class candidates which have overlapped confidence intervals.

Conclusion

- The suggested **beta-distribution-based estimation of conditional class probabilities referred to as the class probability output network (CPON)** was very effective to improve the classification performances of discriminant-function-based classifiers.

- Ref: IEEE Tr. NN, 20(10):1659-1673, 2009.
  IEEE Tr. CE, 56(4):2296-2302, 2010.
  Neural Networks, 64:19-28, 2015.
  Neurocomputing, 248:67-75, 2017.

- Further improvement can be achieved by considering the **deep structure of CPON models.**

- This method can be applied to wide range of pattern classification problems.