The Korean Communications
in Statistics Vol. 10 No. 1, 2003
pp. 225-232

# Polychotomous Machines

Ja-Yong Koo1) and Heon Jin Park2) and Daewoo Choi3)

## Abstract

The support vector machine (SVM) is becoming increasingly popular in classification. The import vector machine (IVM) has been introduced for its advantages over SMV. This paper tries to improve the IVM. The proposed method, which is referred to as the polychotomous machine (PM), uses the Newton-Raphson method to find estimates of coefficients, and the Rao and Wald tests, respectively, for addition and deletion of import points. Because the PM basically follows the same addition step and adopts the deletion step, it uses, typically, less import vectors than the IVM without loosing accuracy. Simulated and real data sets are used to illustrate the performance of the proposed method.

*Keywords* : classification, import vector, maximum likelihood, Newton-Raphson, reproducing kernel, stepwise algorithm

## 1. Introduction

One of the fundamental problems of learning theory is the multiple classification problem. Suppose we are given multi-classes of objects, each of which consists of a predictor $x$ and a qualitative label $y$. Let the labels $y \in \mathbf{M}=\{1,\ldots,M.\}$ When $M=2$, the classification is referred to as binary classification; when $M>2$ , it becomes a polychotomous classification problem. A polychotomous learning algorithm uses the training data $L=\{(x_n,y_n):n=1,\ldots,N\}$ to construct a function $C=C(x,L)$such that we have to assign a new input $x$ to one of the $M$ classes.

The support vector machine (SVM) is becoming increasingly more popular in classification problems with many successful applications, see Schölkopt and Smola(2002) and Vapnik (1998).

1) Department of Statistics, Inha University, Incheon 402-751, Korea.
E-mail : jykoo@stat.inha.ac.kr
2) Department of Statistics, Inha University, Incheon 402-751, Korea.
E-mail : hjpark@anova.inha.ac.kr
3) Department of Statistics, Hankuk University of Foreign Studies, Yongin 449-791, Korea.
E-mail : dachoi@freechal.com

The SVM can be described by a positive definite reproducing kernel $K$. Zhu and Hastie (2001) has proposed the import vector machine (IVM) based on the following observation that fitting an SVM is equivalent to minimizing over $f \in H_K$

$$\frac{1}{N}\sum_{n=1}^{N}(1-y_n f(x_n))_+ + \lambda \| f \|_{H_K}$$

and the loss $(1-yf)_+$ can be approximated by the binomial likelihood. It has been shown that the IVM has advantages over the SVM in the following aspects: (i) the IVM can naturally handle the polychotomous classification problem; (ii) the IVM can provide estimates of the posterior probabilities $P(Y=k \mid X=x)$ (iii) the computational cost of the IVM is typically cheaper than that of the SVM of Zhu and Hastie (2001).

This paper proposes a polychotomous machine (PM) which is an improved version of the IVM. The PM tries to improve the performance of IVM in three respects. We use the Newton-Raphson method to find estimates of coefficients, and the Rao and Wald tests, respectively, for addition and deletion of import points. One may have to use one-step Newton-Raphson method when the computation of estimates is computationally prohibitive for $N$ large. However, the stepwise deletion step can be carried out with small amount of extra computation, because the deletion algorithm is much less computer intensive than the addition algorithm. Because the PM basically follows the same addition step as the IVM and adopts the deletion step, it uses, typically, less import vectors than the IVM without loosing accuracy. A limited number of numerical simulations have shown that the number of the import points from PM is usually less than the number of import points from IVM.

This paper is organized as follows. Section 2 describes the polychotomous machines. Section 3 explains the stepwise algorithm for the selection of the import points. Section 4 illustrate the performance of the proposed algorithm via simulations using both real and simulated data sets.

## 2. Polychotomous machine

Consider a training data $L=\{(x_n, y_n): n=1,\dots,N\}$ where the instances $x_n$ belong to some domain $X \subset R^d$ and the labels $y_n \in \boldsymbol{M}=\{1,\dots,M\}$ The set $\boldsymbol{X}$ is a set from which the patterns $x_n$ (cases, inputs, instances) are taken and the $y_n$ are called labels, targets or outputs. A polychotomous learning algorithm uses $L$ to construct a function $C=C(\cdot, L)$: $\boldsymbol{X} \rightarrow \boldsymbol{M}$ such that we have to assign a new input $x$ to $C(x)$, one of the $M$ classes.

Let $K$ be a positive definite kernel. Several kernel functions satisfy Mercer's conditions so that they are positive definite. Popular kernels are polynomial kernels of the form

$K(x,x')=(1+\langle x,x'\rangle)^q$and Gaussian radial basis functions of the form $K(x,x')=\exp(-\|x-x'\|^2/(2\sigma^2))$Denote by $f(\cdot\mid \boldsymbol{a})$ a linear combination of the form

$$f(\cdot\mid \boldsymbol{a})=a_0+\sum_{s_l\in S}a_lK(\cdot,s_l),$$

where $S=\{s_1,\dots,s_J\}\subset A=\{x_1,\dots,x_N\}$Definite $\varphi_l=K(\cdot,s_l)$for $l=1,\dots,J$and set $\varphi_0(x)\equiv 1$. The set

$$\{\varphi_l;\ l=0,1,\dots,J\}$$

forms a (possibly over-complete) basis.

Define

$$\eta(m|x)=\sum_{l=0}^{J}a_{ml}\varphi_l(x),\qquad 1\le m\le M-1$$

and $\eta(m|x)\equiv 0$. Let $\boldsymbol{a}_m$ denote the $J+1$ dimensional (column) vector of entries $a_{ml}$ for $0\le l\le J$ and $1\le m\le M$, and set $\boldsymbol{a}_M=0$ Let a the $(M-1)(J+1)$ dimensional vector of entries of $\boldsymbol{a}_1,\dots,\boldsymbol{a}_{M-1}$A polychotomous model has the form

$$p(m|x)=p(m|x;\boldsymbol{a})=\frac{\exp(\eta(m|x))}{\sum_{l=1}^{M}\exp(\eta(l|x))},\qquad 1\le m\le M$$

It can be seen that $p(m|x)\in(0,1)$ for $1\le m\le M$ and $\sum_m p(m|x)=1$ for any $x\in \boldsymbol{X}$ .

Let $\boldsymbol{K}$ denote the $J\times J$ matrix with entries $K(x_l,x_{l'})$ for $x_l,x_{l'}\in S$ and define $\boldsymbol{K}^0$ to be

$$\boldsymbol{K}^0=\begin{bmatrix}0 & \boldsymbol{0}^T\\ \boldsymbol{0} & \boldsymbol{K}\end{bmatrix}.$$

The multinomial log-likelihood based on $L$ is defined by

$$\ell_\lambda(\boldsymbol{a})=\sum_{n=1}^{N}\left[\eta(y_n|x_n)-\log\left(\sum_{l=1}^{M}\eta(l|x_n)\right)\right]-\frac{\lambda}{2}\sum_{m=1}^{M-1}\boldsymbol{a}^T{}_m\,\boldsymbol{K}^0\,\boldsymbol{a}_m \tag{1}$$

the (regularized) maximum likelihood estimator $\hat{\boldsymbol{a}}$ is defined to be the minimizer of $\ell_\lambda(\boldsymbol{a})$. It can be noted from (1) that the constant basis $\varphi_0$ is not regularized. The polychotomous machine classifies a new input $x$ into the class $\widehat{m}$ which is defined by

$$\widehat{m}=\arg\max\hat{P}(m|x;\hat{\boldsymbol{a}}).$$

To find $\hat{\boldsymbol{a}}$, we use the the Newton-Rapshon method. Let $S_\lambda(\boldsymbol{a})$ denote the score at $\boldsymbol{a}$ with entries $\partial\ell_\lambda(\boldsymbol{a})/\partial a_{ml}$ , and let $I_\lambda(\boldsymbol{a})$ denote the information matrix with entries $-\partial^2\ell_\lambda(\boldsymbol{a})/\partial a_{ml}\partial a_{m'l'}$. The maximum likelihood estimate $\widehat{\boldsymbol{a}}$ satisfies the likelihood equation $S(\hat{a})=0$ The Newton-Raphson method for computing $\hat{a}$ is to iteratively determine $a^{m+1}$ from $a^{m}$ according to the formula

$$a^{m+1}= a^{m}+I_{\lambda}(a^{m})^{-1}S_{\lambda}(a^{m}).$$

## 3. Sparse Stepwise Algorithm

Finding an optimal set $\hat{S}\subset A$ is not an easy problem since an iterative algorithm such as the Newton-Raphson algorithm is necessary to compute $\hat{a}$ and the size of $\hat{S}$ is not known beforehand. Hence, we use the following SSA algorithm which is a sparse greedy algorithm to find near optimal solutions among the subsets of $A$.

**Sparse Stepwise Algorithm(SSA)**

```
Reguire: Set of functions $\phi_l = K(\cdot, s_l)$, $s_l \in A$
  Divide $L$ into the training and the test data sets. Set $\lambda=\lambda_{\max}$, $S=\emptyset$ and fit the constant model.
  reduce $\lambda$
    repeat addition with given $\lambda$
      Pick $\hat{s}=\arg\max_{x_l\in A-S} R(x_l)$
      $S=S\cup\{\hat{s}\}$
      Compute $\hat{a}_{\lambda}$ and test error rate
    until $\ell(\hat{a}_{\lambda})$ does not increase much
  until $\lambda\le\lambda_{\min}$
  repeat deletion with optimal $\hat{\lambda}$
    Pick $\hat{s}=\arg\min_{s_l\in S} W(s_l)$
    $S=S\setminus\{\hat{s}\}$
    Compute $\hat{a}_{\hat{\lambda}}$ and test error rate
  until $S=\emptyset$
Output: $S=\emptyset$ and $\hat{a}_{\hat{\lambda}}$ corresponding to the model having minimum test error rate
```

In the SSA algorithm, those points in $\hat{S}$ are referred to as import points, $R(x_l)$ and

$W(s_l)$ denote the Rao and Wald statistics, see Rao (1973). The addition step of SSA and the choice of the optimal $\lambda$ is essentially the same as that of Zhu and Hastie (2001), except that we use the Newton-Raphson method to find $\hat{a}_\lambda$ .

Using Theorem A.80 of Herbrich (2001), one can compute the Rao statistic $R(x_l)$ and the Wald statistic $W(s_l)$ by one inversion of a $(M-1)\times(M-1)$ matrix rather than a full inversion of a big $(M-1)\times(J+1)\times(M-1)\times(J+1)$ matrix.

## 4. Numerical results

In this section, the performance of polychotomous machine is illustrated by simulation method using simulated and real data sets.

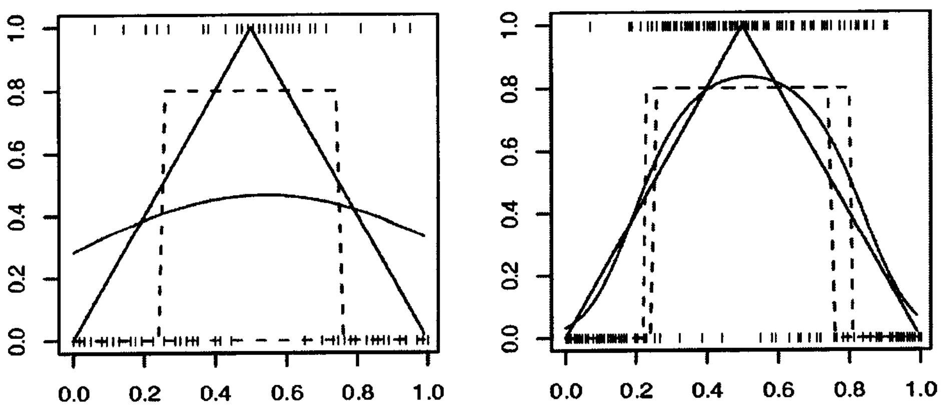

Figure 1. Posterior probability estimates using PM. Left panel shows the result for $N=64$; right panel for $N=128$. Red solid lines denote the estimates of $p(x)$ and red dotted lines show the estimates of $\text{sign}\,[p(x)-1/2]$. Blue solid and dotted lines show the true $p(x)$ and $\text{sign}\,[p(x)-1/2]$. Small tick marks show the location of the predictor values, where the upper ones correspond to label 1 and the bottom ones show the predictor values with label 0.

### 4.1 Posterior estimation

Standard SVM approach does not automatically produce a posterior probability estimate, which is often useful and required in many practical classification applications. Several methods have been proposed to modify the standard SVM to produce posterior probabilities. Wahba (1999) and Platt (1999). Platt (1999) has shown that a sigmoid method for SVM posterior estimation as a postprocessing gives an accurate estimates. However, the PM can give those estimate automatically without further postprocessing.

To show the performance for posterior estimation, data points are generated as in Lin (1999), where $N$ equidistant points on the interval [0, 1] were taken as the predictor values. Let the posterior probability function

$$p(x) = P(Y=1|X=x) = 1-|1-2x|$$

and Bernoulli random variables were generated with success probabilities $p(x_n)$. Though the shape of $p(x_n)$ is simple, $f(x) = \log[p(x)/(1-p(x))]$ approaches to $\infty$ as $x \rightarrow 1/2$, and to $-\infty$ as $x$ converges to 0 and 1 so that the precise estimation of $p(x)$ appears to be difficult for the PM.

Figure 1 displays the PM estimates using $N=64, 128$data points and the Gaussian kernel with $\sigma=1$ according to the SSA algorithm in Section 3. The import points were 0.02 and 0.09 for $N=64$, and 0.03, and 0.11 for $N=128$, which implies that similar locations were necessary. The optimal $\lambda$ were 0.05 and 0.0025 for $N=64$ and $N=128$, respectively. As the sample size increases, the shape of the estimated function looks more similar to that of the posterior probability function $p(x)$. When $N=64$, the estimate of sign $[p(x)-1/2]$ missed the shape of the true one. Note that the function sign $[p(x)-1/2]$ and its estimates were scaled so that they did not interfere the other plots

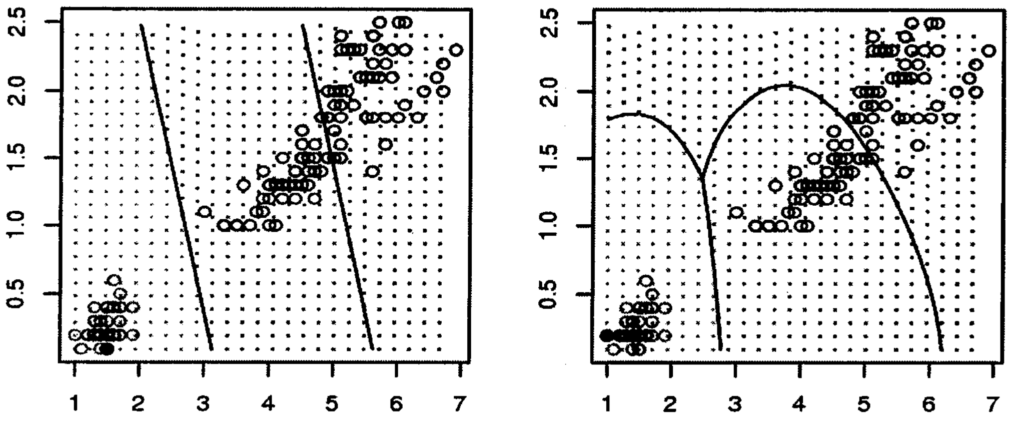

Figure 2. PM fit to the **iris** data. The left panel shows the result using Gaussian kernel and the right used linear kernel. Green, red and blue points, respectively, denote the data points from the class **setosa, versicolor** and **versinica**. Black lines display the decision boundaries for three classes and black solid points are the import points. The shaded regions denote classification regions.

## 4.2 Iris data

The **iris** data set is an well-known one in classification literature. It has three classes,

**setosa, versicolor** and **versinica**; each class has 50 data points; the number of predictors is four. For the convenience of display, the result is shown using the variables the last two variables, **petal.length** and **petal.width**.

Figure 2 contains the result for the **iris** data. The left panel shows the result using the linear kernel, for which the optimal $\lambda = 1$; the right panel displays the PM fit using the Gaussian kernel with scale $\sigma = 1$ for which the optimal $\lambda = 0.0018$ The numbers of import vectors were three and one for those two fits. It may be concluded from the Figure 2 that that linear decision boundary is enough for the **iris** data and only one import vector suffices to classify the classes of **iris**.

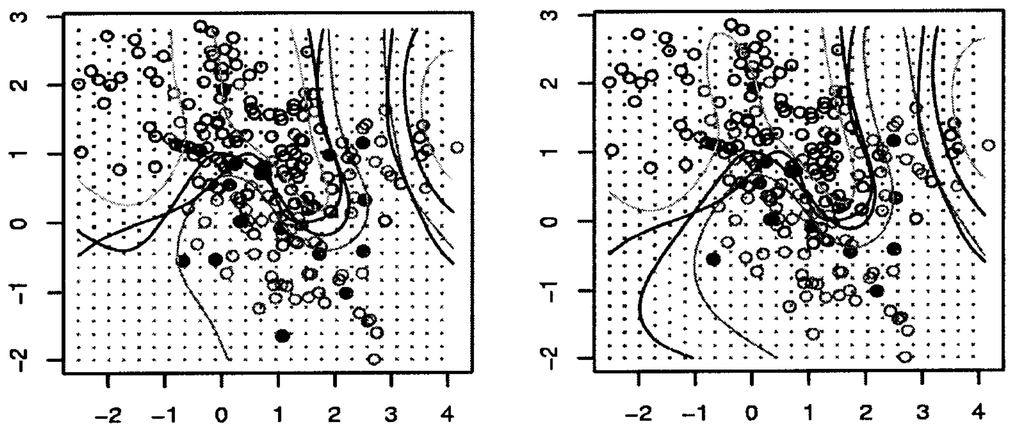

Figure 3. PM fit to the mixture data. The black lines are the classification lines; the blue lines are the Bayes rule boundaries; the yellow lines are the $\hat{p}(x)$ = 0.25 and 0.75. The black points are the import points. The train error rates were 0.155 for both panels; the test error rates were 0.22 for both cases.

## 4.3 Mixture of Gaussians

To show the performance of PM when the Bayes decision boundary is nonlinear, we generated data as in Figure 2.3 of Hastie, Tibshirani and Friedman (2001).

Figure 3 displays the result for training sample size $N=200$ with 100 for each class. The left panel shows the result without the stepwise deletion, for which case the number of import points was 18; the result with stepwise deletion is presented in the right panel, for which case 4 import points were deleted during the stepwise deletion. The optimal regularizing parameter was $\lambda$ = 0.0183. Given the estimated decision boundaries, the test error rates based on 1000 test data points were approximately 0.22 depending on the random seed. The minor difference in the test error rate seems to come from the difference of the decision boundaries in left part of figures. The appearance of the boundary around this area does not seem to affect the error rate much. Though we use a different algorithm, the overall shape of from the SSA appears to be quite similar to that of Figure 3 of Zhu and Hastie (2001), except that the import points of the PM is less.

## Acknowledgement

This work was supported by INHA UNIVERSITY Research Gran (INHA-22433). The authors wish to thank Trevor Hastie for his helpful comments and for providing the means in the mixture experiment.

## References

[1] Schölkopt, B. and Smola, J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press, Massachusetts.

[2] Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer, New York.

[3] Herbrich, R. (2001). *Learning Kernel Classifiers: Theroy and Algorithms.* The MIT Press, Massachusetts.

[4] Lin, Y. (1999). Support Vector Machines and the Bayes rule in classification. *Technical Report 1014.* Department of Statistics, University of Wisconsin.

[5] Platt, J. (1999). *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.* In Smola, Bartlett, Schölkopt and Schuurmans. (eds.) *Advances in Large Margin Classifiers.* The MIT press, Massachusetts

[6] Rao, C. R. (1973). *Linear Statistical Inference and Its Applications. 2nd edn.* Wiley, New York.

[7] Vapnik, V. (1998). *Statistical Learning Theory.* Wiley, New York.

[8] Wahba, G. (1999). The Bias-Variance Tradeoff and the Randomized GACV. In Cohn, Kearns and Solla. (eds): *Advances in Neural Information Processing Systems, Vol. 11.* The MIT Press, Massachusetts.

[9] Zhu, J. and Hastie, T. (2001). Kernel Logistic Regression and the Import Vector Machines. *Advances in Neural Information Processing Systems, 14.*

[ Received February 2003, Accepted March 2003 ]