Ji Zhu and Trevor Hastie

Department of Statistics
Stanford University
Stanford, CA 94305
{jzhu,hastie}@stat.stanford.edu

Abstract. The support vector machine is known for its excellent
formance in binary classification, i.e., the response $y \in \{-1, 1\}$
its appropriate extension to the multi-class case is still an on-goi
search issue. Another weakness of the SVM is that it only esti
$sign[p(x) - 1/2]$, while the probability $p(x)$ is often of interest
where $p(x) = P(Y = 1|X = x)$ is the conditional probability of a
being in class 1 given $X = x$. We propose a new approach for class
tion, called the import vector machine, which is built on kernel lo
regression (KLR). We show on some examples that the IVM per
as well as the SVM in binary classification. The IVM can natura
generalized to the multi-class case. Furthermore, the IVM provid
estimate of the underlying class probabilities. Similar to the "su
points" of the SVM, the IVM model uses only a fraction of the tra
data to index kernel basis functions, typically a much smaller fra
than the SVM. This can give the IVM a computational advantage
the SVM, especially when the size of the training data set is larg
illustrate these techniques on some examples, and make connections
boosting, another popular machine-learning method for classificati

Keywords: classification, kernel methods, logistic regression, multi-cla
ing, radial basis, reproducing kernel Hilbert space (RKHS), suppor
machines.

1 Introduction

In standard classification problems, we are given a set of training da
$(x_2, y_2), \ldots (x_N, y_N)$, where the output y_i is qualitative and assumes
finite set $\mathcal{C}$. We wish to find a classification rule from the training d
when given a new input x, we can assign a class c from $\mathcal{C}$ to it. U
assumed that the training data are an independently and identically
sample from an unknown probability distribution $P(X, Y)$.

The support vector machine (SVM) works well in binary classif
$y \in \{0, 1\}$, but its appropriate extension to the multi-class case is
going research issue. Another weakness of the SVM is that it onl

F. Roli and J. Kittler (Eds.): MCS 2002, LNCS 2364, pp. 16–26, 2002.

naturally be generalized to the multi-class case. Furthermore, the IV
an estimate of the probability $p(x)$. Similar to the "support points" o
the IVM model uses only a fraction of the training data to index
basis functions. We call these training data *import points*. The con
cost of the SVM is $O(N^3)$, while the computational cost of the IVM i
where q is the number of import points. Since q does not tend to
N increases, the IVM can be faster than the SVM, especially for lar
data sets. Empirical results show that the number of import point
much less than the number of support points.

In section (2), we briefly review some results of the SVM for bin
cation and compare it with kernel logistic regression (KLR). In sect
propose our IVM algorithm. In section (4), we show some simulation
section (5), we generalize the IVM to the multi-class case.

2 Support Vector Machines and Kernel Logistic Regression

The standard SVM produces a non-linear classification boundary in
input space by constructing a linear boundary in a transformed ver
original input space. The dimension of the transformed space can be
even infinite in some cases. This seemingly prohibitive computation
through a positive definite reproducing kernel K, which gives the inn
in the transformed space.

Many people have noted the relationship between the SVM
ularized function estimation in the reproducing kernel Hilb
(RKHS). An overview can be found in Evgeniou, Pontil, and Pog
Hastie, Tibshirani, and Friedman 2001) and Wahba, Lin, and Zh
Fitting an SVM is equivalent to minimizing:

$$\frac{1}{N}\sum_{i=1}^{N}(1-y_i f(x_i))_+ + \lambda\|f\|^2_{\mathcal{H}_K}.$$

with $f = b + h,\ h \in \mathcal{H}_K,\ b \in \mathcal{R}$. $\mathcal{H}_K$ is the RKHS generated by th
The classification rule is given by $sign[f]$.

By the representer theorem (Kimeldorf and Wahba 1971), the op
has the form:

$$f(x) = b + \sum_{i=1}^{N} a_i K(x, x_i).$$

non-zero a_i's. The corresponding x_i's are called support points.

Notice that (1) has the form $loss + penalty$. The loss function (plotted in Figure 1, along with several traditional loss functions. As the negative log-likelihood (NLL) of the binomial distribution has a si to that of the SVM. If we replace $(1 - yf)_+$ in (1) with $\ln(1 + e^{-yf}$ of the binomial distribution, the problem becomes a KLR problem. that the fitted function performs similarly to the SVM for binary cla

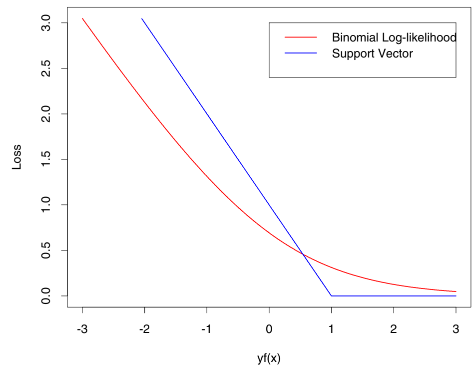

Fig. 1. *The binomial log-likelihood and* hinge *loss function,* $y \in \{-$

There are two immediate advantages of making such a replaceme sides giving a classification rule, the KLR also offers a natural esti probability $p(x) = e^f/(1 + e^f)$, while the SVM only estimates $sign[$ (b) The KLR can naturally be generalized to the multi-class case thr multi-logit regression, whereas this is not the case for the SVM. H cause the KLR compromises the hinge loss function of the SVM, it n the "support points" property; in other words, all the a_i's in (2) are

KLR has been studied by many research Wahba, Gu, Wang, and Chappell (1995) and references there; Green and Silverman (1994) and Hastie and Tibshirani (1990).

The computational cost of the KLR is $O(N^3)$; to save the co cost, the IVM algorithm will find a sub-model to approximate the fu given by the KLR. The sub-model has the form:

$$f(x) = b + \sum_{x_i \in \mathcal{S}} a_i K(x, x_i)$$

$\mathcal{S}$. Lin, Wahba, Xiang, Gao, Klein, and B. (2001) divide the trainin several clusters, then randomly select a representative from each clus up $\mathcal{S}$. Smola and Schölkopf (2000) develop a greedy technique to s select q columns of the kernel matrix $[K(x_i, x_j)]_{N\times N}$, such that the sp q columns approximates the span of $[K(x_i, x_j)]_{N\times N}$ well in the Frob Williams and Seeger (2001) propose randomly selecting q points of t data, then using the Nystrom method to approximate the eigen-dec of the kernel matrix $[K(x_i, x_j)]_{N\times N}$, and expanding the results ba dimensions. None of these methods uses the output y_i in selecting t (i.e., the procedure only involves x_i). The IVM algorithm uses both y_i and the input x_i to select the subset $\mathcal{S}$, in such a way that the approximates the full model well.

3 Import Vector Machine

Following the tradition of logistic regression, we let $y_i \in \{0, 1\}$ for this paper. For notational simplicity, the constant term in the fitted ignored.

In the KLR, we want to minimize:

$$H = -\sum_{i=1}^{N}[y_i f(x_i) - \ln(1 + \exp(f(x_i)))] \; + \; \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K}$$

From (2), it can be shown that this is equivalent to the finite dimens

$$H = -\boldsymbol{y}^T(K_a\boldsymbol{a}) + \mathbf{1}^T\ln(1 + \exp(K_a\boldsymbol{a})) \; + \; \frac{\lambda}{2}\boldsymbol{a}^T K_q \boldsymbol{a}$$

where $\boldsymbol{a} = (a_1, \dots a_N)^T$; the regression matrix $K_a = [K(x_i, x_j)]_{N\times}$ regularization matrix $K_q = K_a$.

To find $\boldsymbol{a}$, we set the derivative of H with respect to $\boldsymbol{a}$ equal to the Newton-Raphson method to iteratively solve the score equation shown that the Newton-Raphson step is a weighted least squares ste

$$\boldsymbol{a}^{(k)} = (K_a^T W K_a + \lambda K_q)^{-1} K_a^T W \boldsymbol{z}$$

where $\boldsymbol{a}^{(k)}$ is the value of $\boldsymbol{a}$ in the kth step, $\boldsymbol{z} = (K_a\boldsymbol{a}^{(k-1)} + W^{-1}(\boldsymbol{y}$ weight matrix is $W = diag[p(x_i)(1 - p(x_i))]_{N\times N}$.

As mentioned in section 2, we want to find a subset $\mathcal{S}$ of $\{x_1,$ such that the sub-model (3) is a good approximation of the full mode it is impossible to search for every subset $\mathcal{S}$, we use the following gre strategy:

$$f_l(x) = \sum_{x_j \in \mathcal{S} \cup \{x_l\}} a_j K(x, x_j)$$

Find $\boldsymbol{a}$ to minimize

$$H(x_l) = -\sum_{i=1}^{N} [y_i f_l(x_i) - \ln(1 + \exp(f_l(x_i)))] + \frac{\lambda}{2} \| f_l(x)$$

$$= -\boldsymbol{y}^T (K_a^l \boldsymbol{a^l}) + \mathbf{1}^T \ln(1 + \exp(K_a^l \boldsymbol{a^l})) \ + \ \frac{\lambda}{2} \boldsymbol{a^l}^T$$

where the regression matrix $K_a^l = [K(x_i, x_j)]_{N \times (q+1)}$, $x_i \in \{x_1,$ $x_j \in \mathcal{S} \cup \{x_l\}$; the regularization matrix $K_q^l = [K(x_j, x_l)]$ $x_j, x_l \in \mathcal{S} \cup \{x_l\}$; $q = |\mathcal{S}|$.

$(B3)$ Let

$$x_{l^*} = \text{argmin}_{x_l \in \mathcal{R}} H(x_l).$$

Let $\mathcal{S} = \mathcal{S} \cup \{x_{l^*}\}$, $\mathcal{R} = \mathcal{R} \setminus \{x_{l^*}\}$, $H_k = H(x_{l^*})$, $k = k + 1$.

$(B4)$ Repeat steps $(B2)$ and $(B3)$ until H_k converges.

We call the points in $\mathcal{S}$ import points.

3.2 Revised Algorithm

The above algorithm is computationally feasible, but in step $(B2)$ we the Newton-Raphson method to find $\boldsymbol{a}$ iteratively. When the numbe points q becomes large, the Newton-Raphson computation can be ex reduce this computation, we use a further approximation.

Instead of iteratively computing $\boldsymbol{a}^{(k)}$ until it converges, we ca one-step iteration, and use it as an approximation to the converged a good approximation, we take advantage of the fitted result from "optimal" $\mathcal{S}$, i.e., the sub-model when $|\mathcal{S}| = q$, and use it as the initial one-step update is similar to the score test in generalized linear mod but the latter does not have a penalty term. The updating formula weighted regression (5) to be computed in $O(Nq)$ time.

Hence, we have the revised step $(B2)$ for the basic algorithm:

$(B2^*)$ For each $x_l \in \mathcal{R}$, correspondingly augment K_a with a colun with a column and a row. Use the updating formula to fin Compute (6).

compare H_k with H_{k-r}, where r is a pre-chosen small integer, for exa
If the ratio $\frac{|H_k - H_{k-r}|}{|H_k|}$ is less than some pre-chosen small number α, f
$\alpha = 0.001$, we stop adding new import points to $\mathcal{S}$.

3.4 Choosing the Regularization Parameter λ

So far, we have assumed that the regularization parameter λ is fixed.
we also need to choose an "optimal" λ. We can randomly split all the
training set and a tuning set, and use the misclassification error on th
as a criterion for choosing λ. To reduce the computation, we take a
the fact that the regularized NLL converges faster for a larger λ. Th
of running the entire revised algorithm for each λ, we propose th
procedure, which combines both adding import points to $\mathcal{S}$ and c
optimal λ:

($C1$) Start with a large regularization parameter λ.
($C2$) Let $\mathcal{S} = \emptyset$, $\mathcal{R} = \{x_1, x_2, \ldots, x_N\}$, $k = 1$.
($C3$) Run steps ($B2^*$), ($B3$) and ($B4$) of the revised algorithm, until t
criterion is satisfied at $\mathcal{S} = \{x_{i1}, \ldots, x_{iq_k}\}$. Along the way, al
the misclassification error on the tuning set.
($C4$) Decrease λ to a smaller value.
($C5$) Repeat steps ($C3$) and ($C4$), starting with $\mathcal{S} = \{x_{i1}, \ldots, x_{iq_k}\}$

We choose the optimal λ as the one that corresponds to the minimu
sification error on the tuning set.

4 Simulation

In this section, we use a simulation to illustrate the IVM
The data in each class are generated from a mixture of
(Hastie, Tibshirani, and Friedman 2001). The simulation results ar
Figure 2.

4.1 Remarks

The support points of the SVM are those which are close to the cl
boundary or misclassified and usually have large weights $[p(x)(1 -$
import points of the IVM are those that decrease the regularize
most, and can be either close to or far from the classification bou
difference is natural, because the SVM is only concerned with the cl
$sign[p(x) - 1/2]$, while the IVM also focuses on the unknown proba

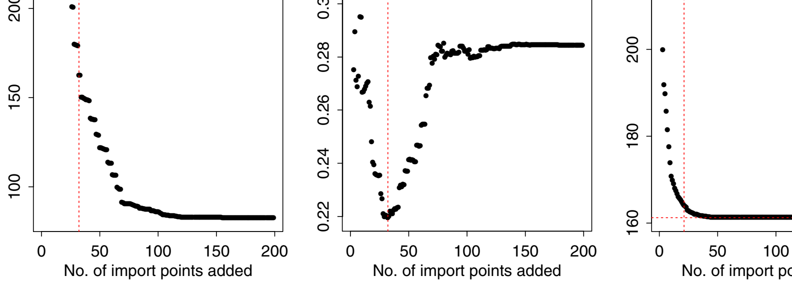

Fig. 2. *Radial kernel is used.* $N = 200$*. The left and middle panels illus* *choose the optimal* λ*.* $r = 1$*,* $\alpha = 0.001$*,* λ *decreases from* e^{10} *to* e^{-10}*. T* *misclassification rate* 0.219 *is found to correspond to* $\lambda = 0.135$*. The right* *the optimal* $\lambda = 0.135$*. The stopping criterion is satisfied when* $|\mathcal{S}| = 21$*.*

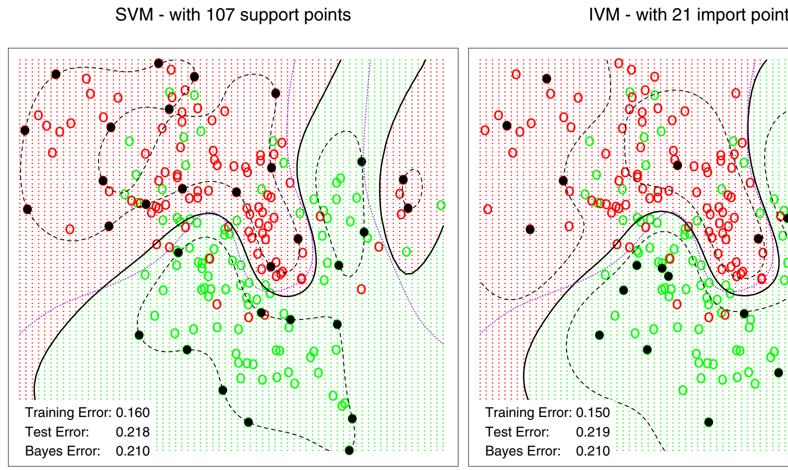

Fig. 3. *The solid black lines are the classification boundaries; the dashed* *are the Bayes rule boundaries. For the SVM, the dashed black lines are the* *margin. For the IVM, the dashed black lines are the* $p(x) = 0.25$ *and* 0.7 *the black points are the import points.*

Though points away from the classification boundary do not contri termining the position of the classification boundary, they may co estimating the unknown probability $p(x)$. Figure 3 shows a compar SVM and the IVM. The total computational cost of the SVM is O the computational cost of the IVM method is $O(N^2q^2)$, where q is

5 Multi-class Case

In this section, we briefly describe a generalization of the IVM to
classification. Suppose there are $M+1$ classes. We can write the
an M-vector $\boldsymbol{y}$, with each component being either 0 or 1, indicating
the observation is in. Therefore $y_k = 1,\ y_j = 0,\ j \neq k,\ j \leq M$ in
response is in the kth class, and $y_j = 0, j \leq M$ indicates the res
the $M+1$th class. Using the $M+1$th class as the basis, the multi-l
written as $f_1 = \ln(p_1/p_{M+1}),\ \ldots,\ f_M = \ln(p_M/p_{M+1}),\ f_{M+1} = 0.$
Bayes classification rule is given by:

$$c = \text{argmax}_{k \in \{1,2,\ldots,M+1\}} f_k$$

We use i to index the observations, j to index the classes, i.e. i
$j = 1, \ldots M$. Then the regularized negative log-likelihood is

$$H = -\sum_{i=1}^{N} [\boldsymbol{y}_i^T \boldsymbol{f}(x_i) - \ln(1 + e^{f_1(x_i)} + \cdots + e^{f_M(x_i)})] + \frac{\lambda}{2} \| f \|$$

where $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iM})^T,\ \boldsymbol{f}(x_i) = (f_1(x_i), f_2(x_i), \ldots, f_M(x_i))$

$$\| f \|^2_{\mathcal{H}_K} = \sum_{j=1}^{M} \| f_j \|^2_{\mathcal{H}_K}$$

Using the representer theorem (Kimeldorf and Wahba 1971), the
of $\boldsymbol{f}(x)$, $f_j(x)$, which minimizes H has the form

$$f_j(x) = \sum_{i=1}^{N} a_{ij} K(x, x_i).$$

Hence, (7) becomes

$$H = -\sum_{i=1}^{N} [\boldsymbol{y}_i^T (K_a(i,)A)^T - \ln(1 + \mathbf{1}^T e^{(K_a(i,)A)^T})] + \frac{\lambda}{2} \sum_{j=1}^{M} \boldsymbol{a}_j^T$$

where $A = (\boldsymbol{a}_1 \ldots \boldsymbol{a}_M) = (a_{ij})$, K_a and K_q are defined in the same
the binary case; and $K_a(i,)$ is the ith row of K_a.

The multi-class IVM procedure is similar to the binary case, an
putational cost is $O(MN^2q^2)$. Figure 4 is a simulation of the multi-
The data in each class are generated from a mixture of Gaussians.

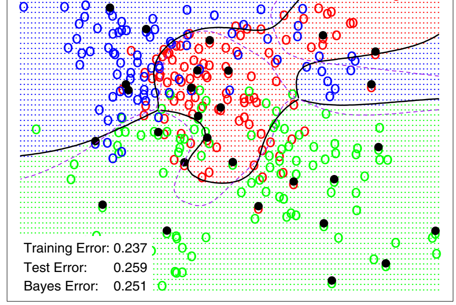

Fig. 4. *Radial kernel is used.* $M + 1 = 3$*,* $N = 300$*,* $\lambda = 0.368$*,* $|\mathcal{S}|$

6 Discussion

Although the intuitive motivation of the SVM is via separating hyper
intuition gets a murky when the classes overlap. In this case it is pe
intuitive to pose the problem as that of regularized function estima
loss function particularly suited to classification. Furthermore, the
the kernel finds its proper home when this function estimation takes
producing kernel Hilbert spaces. We have argued in this paper that th
loss function offers several advantages over the hinge loss, in partic
estimates class probabilities, and generalizes naturally to problems
than two classes. Although KLR lacks the "support vector" propert
we propose the IVM, a simple and attractive compromise with perfor
ilar to that of the SVM. The computational cost of the IVM is $O(N$
binary case and $O(MN^2q^2)$ for the multi-class case, where q is the
import points.

The loss function representation of the SVM and
courages a comparison with *boosting* (Freund and Sch
Hastie, Tibshirani, and Friedman 2001). Figure 5 is similar to
and includes the exponential loss function that drives the boosting
Boosting has been shown to fit a logistic regression model by a fo
parametric gradient descent (Friedman, Hastie, and Tibshirani 2000
exponential loss function. It is also motivated as a means for generat
fier that creates a wide margin between the classes (Schapire and Fr
The comparisons in Figure 5 make it clear that all three metho
ilar in this regard, although the exponential left tail in boosting
non-robustness to clumps of observations far from their parent class

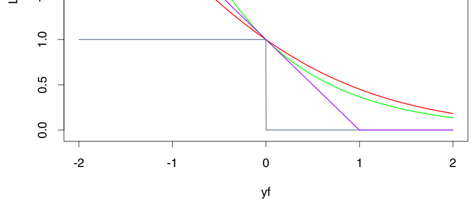

Fig. 5. *The two loss functions from Figure 1, along with the exponential*
implicit in boosting.

Acknowledgments We thank Dylan Small, John Storey, Rob Tibs
Jingming Yan for their helpful comments. Ji Zhu is partially suppo
Stanford Graduate Fellowship. Trevor Hastie is partially supporte
DMS-9803645 from the National Science Foundation, and grant ROI
01 from the National Institutes of Health.

References

Evgeniou, T., M. Pontil, and T. Poggio (1999). Regularization ne
support vector machines. *Advances in Computational Mathema*
pear).

Freund, Y. and R. E. Schapire (1996). Experiments with a new bo
rithm. In *Machine Learning: Proceedings of the Thirteenth In*
Conference, pp. 148–156. Morgan Kauffman, San Francisco.

Friedman, J., T. Hastie, and R. Tibshirani (2000). Additive logistic
a statistical view of boosting (with discussion). *Annals of St*
337–307.

Green, P. and B. Silverman (1994). *Nonparametric Regression and*
Linear Models. Chapman and Hall.

Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models*
and Hall.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements*
cal Learning; Data mining, Inference and Prediction. New Yor
Verlag.

Kimeldorf, G. and G. Wahba (1971). Some results on tchebyche
functions. *J. Math. Anal. Applic. 33*, 82–95.

Lin, X., G. Wahba, D. Xiang, F. Gao, R. Klein, and K. B. (2001
ing spline anova models for large data sets with bernoulli o

Smola, A. and B. Schölkopf (2000). Sparse greedy matrix approx
machine learning. In *Proceedings of the Seventeenth Internatio
ence on Machine Learning.* Morgan Kaufmann.

Wahba, G., C. Gu, Y. Wang, and R. Chappell (1995). *The Mat
Generalization.*, Chapter Soft Classification, a.k.a. Risk Esti
Penalized Log Likelihood and Smoothing Spline Analysis o
Santa Fe Institute Studies in the Sciences of Complexity. Addi

Wahba, G., Y. Lin, and H. Zhang (2000). Gacv for support vector m
A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans (Eds.
in Large Margin Classifiers, Cambridge, MA, pp. 297–311. MI

Williams, C. and M. Seeger (2001). Using the nystrom method t
kernel machines. In T. K. Leen, T. G. Diettrich, and V. Tresp
vances in Neural Information Processing Systems, Volume 13.