Support Vector Machines, Kernel Logistic Regression, and Boosting

Trevor Hastie and Ji Zhu
Statistics Department
Stanford University

New Trends in Optimization and Computational Algorithms (NTOC2001)
December 9-13, 2001, Kyodai-Kaikan, Kyoto, Japan

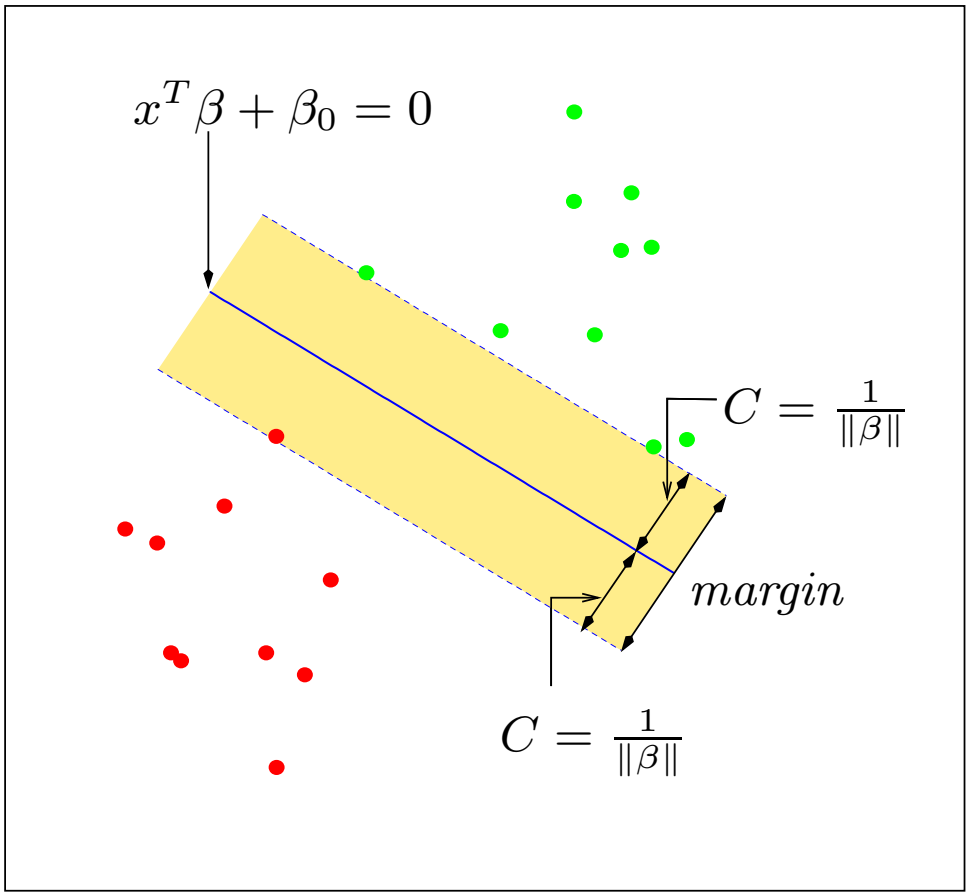
http://www-stat.stanford.edu/~hastie/Papers/ivmtalk.pdf

Outline

- Optimal separating hyperplanes and relaxations
- SVMs: nonlinear generalizations of separating hyperplanes
- SVM as a function estimation problem
- Reproducing kernel Hilbert spaces
- Kernel logistic regression
- Import Vector Machines
- KLR and Boosting.

First part based on work by Vapnik (1996), Wahba (1990), Evgeniou, Pontil, and Poggio (1999), all described in Hastie, Tibshirani and Friedman (2001) *Elements of Statistical Learning*, Springer, NY.

Maximum Margin Classifier

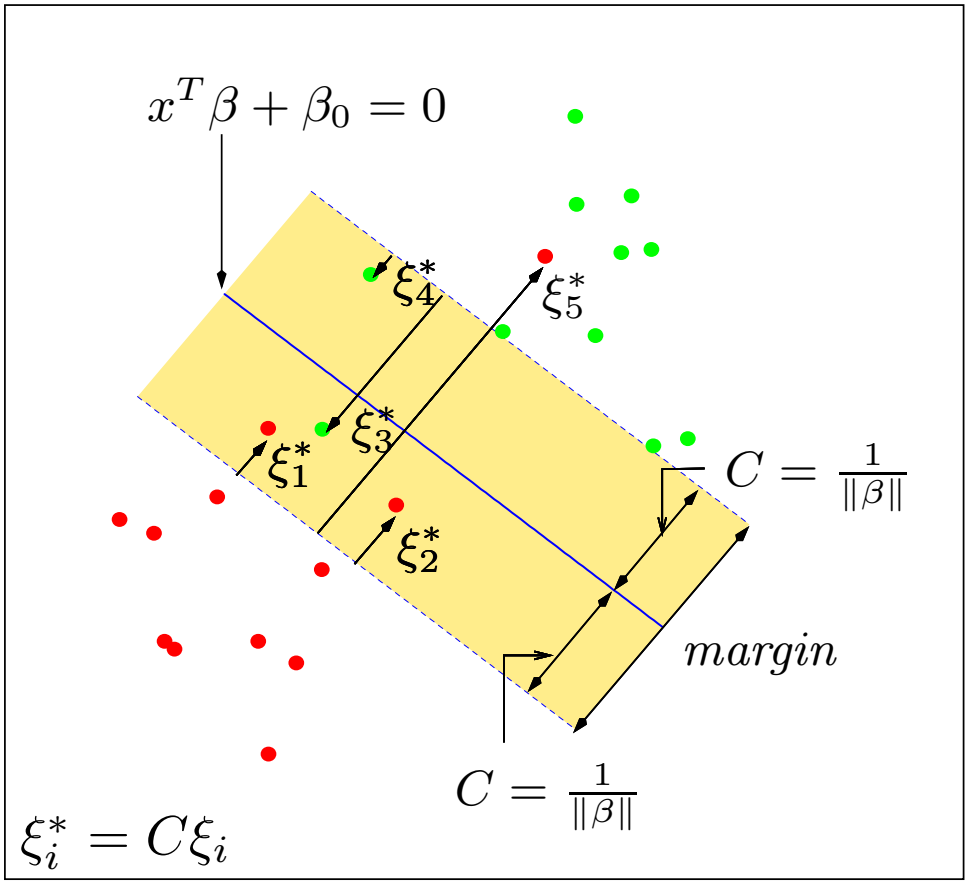


Vapnik(1995) $\quad x_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}$

$$\max_{\beta,\beta_0,\|\beta\|=1} C$$

$$\text{subject to} \quad y_i(x_i^T\beta + \beta_0) \geq C,\ i = 1, \ldots, N.$$

Overlapping Classes

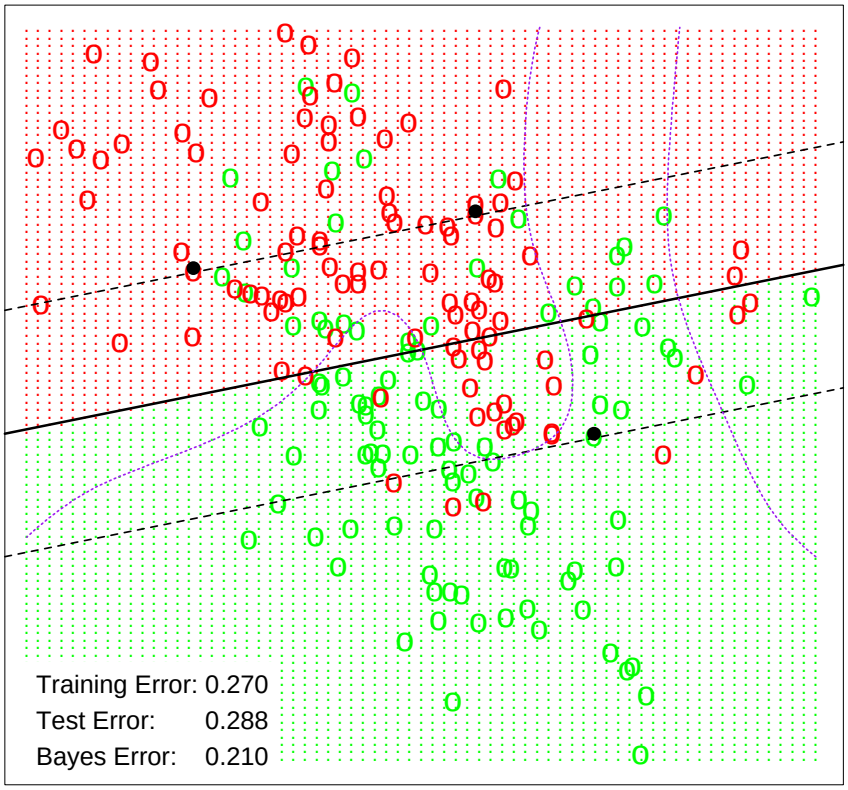


$$\max_{\beta,\beta_0,\|\beta\|=1} C$$

subject to

$y_i(x_i^T\beta + \beta_0) \geq C(1-\xi_i), \;\; \xi_i \geq 0, \;\; \sum_i \xi_i \leq B$

Example



Fitted function is

$$\hat{f}(x) = x^T \hat{\beta} + \hat{\beta}_0$$

Resulting classifier is

$$\hat{G}(x) = \text{sign}[\hat{f}(x)]$$

Quadratic Programming Solution

After a lot of *stuff* we arrive at a Lagrange dual

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'}$$

which we maximize subject to constraints (involving B as well).

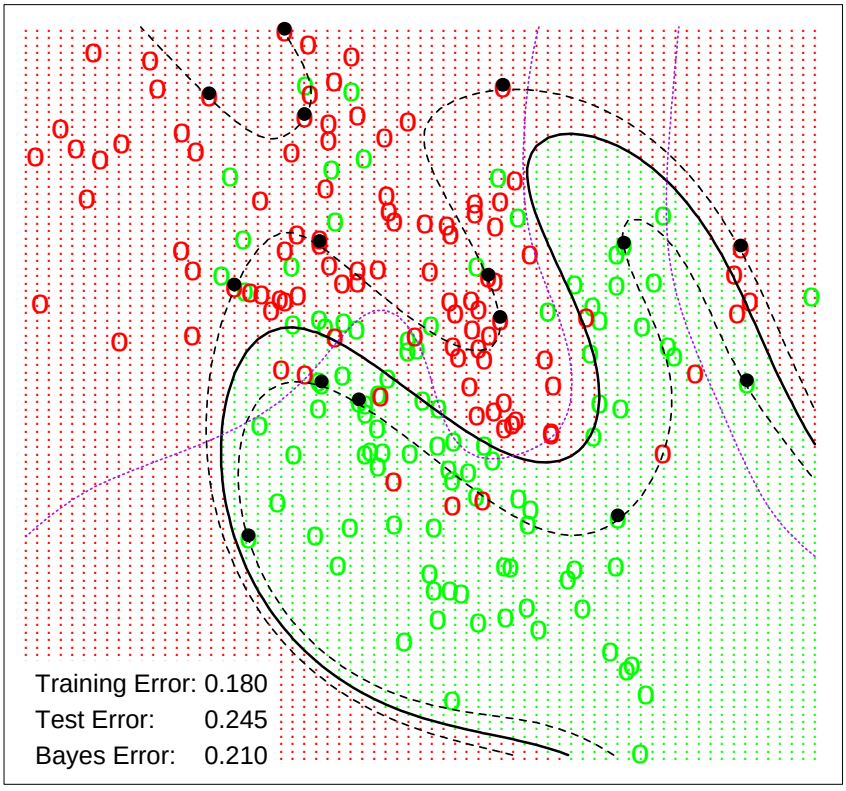
The solution is expressed in terms of the fitted Lagrange multipliers $\hat{\alpha}_i$:

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i$$

Some fraction of α_i are exactly zero (from KKT conditions); the others are called support points

$$\begin{aligned} \hat{f}(x) &= x^T \hat{\beta} + \hat{\beta}^0 \\ &= \sum_{i=1}^{N} \hat{\alpha}_i y_i x^T x_i + \hat{\beta}^0 \end{aligned}$$

Flexible Classifiers



Enlarge the feature space via basis expansions, e.g. polynomials of total degree 4.

$h(x_i) = (h_1(x_i), h_2(x_i), \ldots, h_M(x_i)), \; i = 1, \ldots N$

$$\hat{f}(x) = h(x)^T \hat{\beta} + \hat{\beta}_0$$

SVM

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle$$

$$\begin{aligned} f(x) &= h(x)^T \beta + \beta_0 \\ &= \sum_{i=1}^{N} \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0. \end{aligned}$$

L_D and constraints involve $h(x)$ only through inner-products

$$K(x, x') = \langle h(x), h(x') \rangle$$

Given a suitable positive kernel $K(x, x')$, don't need $h(x)$ at all!

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0$$

Popular Kernels

$K(x, x')$ is a symmetric, positive (semi-)definite function.

dth deg. poly.: $K(x, x') = (1 + \langle x, x' \rangle)^d$

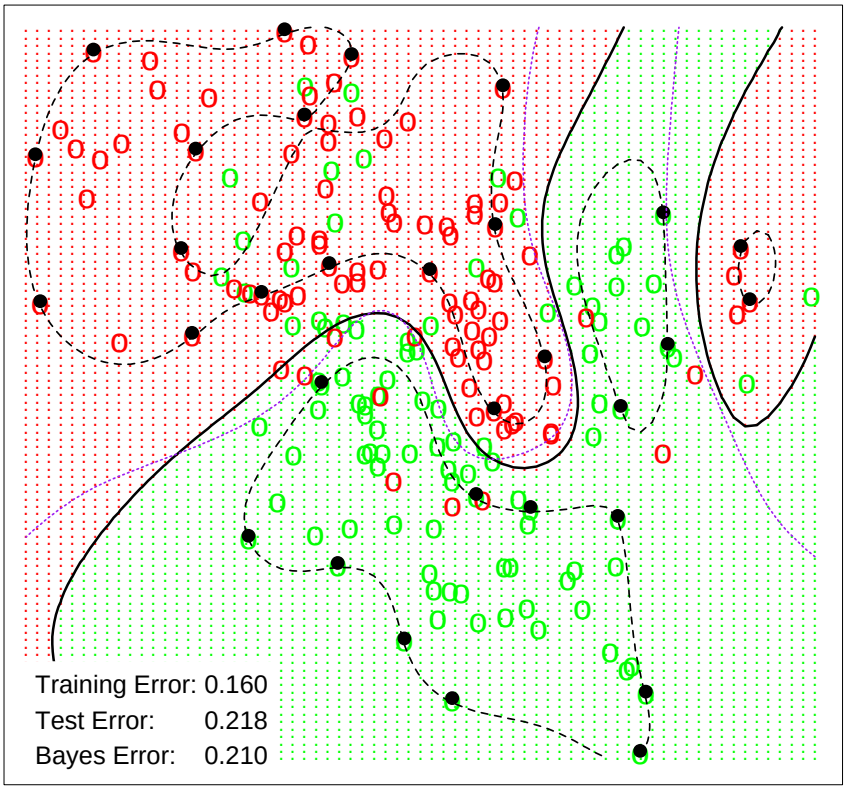
radial basis: $K(x, x') = \exp(-\|x - x'\|^2/c)$

Example: 2nd degree polynomial in $\mathbb{R}^2$.

$$\begin{aligned} K(x, x') &= (1 + \langle x, x' \rangle)^2 \\ &= (1 + x_1 x_1' + x_2 x_2')^2 \\ &= 1 + 2x_1 x_1' + 2x_2 x_2' + (x_1 x_1')^2 \\ &\quad + (x_2 x_2')^2 + 2x_1 x_1' x_2 x_2' \end{aligned}$$

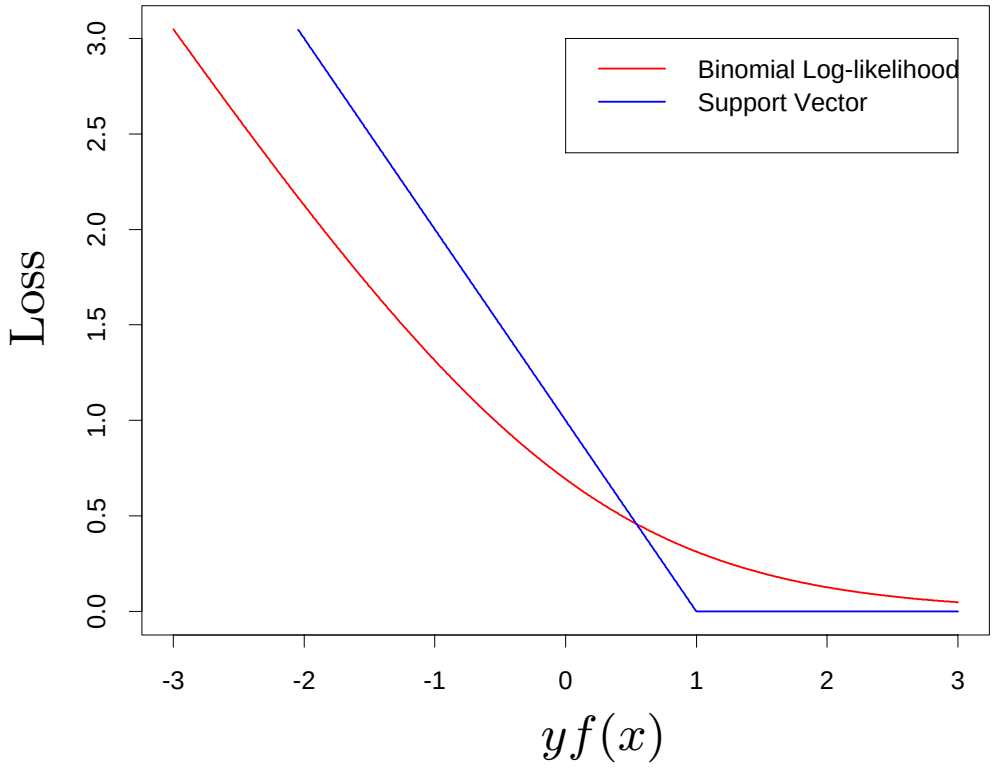
Then $M = 6$, and if we choose
$h_1(x) = 1$, $h_2(x) = \sqrt{2}x_1$, $h_3(x) = \sqrt{2}x_2$,
$h_4(x) = x_1^2$, $h_5(x) = x_2^2$, and $h_6(x) = \sqrt{2}x_1 x_2$,
then $K(x, x') = \langle h(x), h(x') \rangle$.

Dim $h(x)$ infinite



- Fraction of support points depends on overlap; here 45%.
- The less the overlap, the smaller the fraction.
- The smaller B, the smaller the fraction, and more wiggly the function.
- Small fraction $\Rightarrow$ quick lookup.

SVM and function estimation



With $f(x) = h(x)^T\beta + \beta_0$, consider

$$\min_{\beta_0,\,\beta} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \lambda \|\beta\|^2$$

Solution identical to SVM solution, with $\lambda = \lambda(B)$.

$$\text{In general } \min_{\beta_0,\,\beta} \sum_{i=1}^{N} L[y_i, f(x_i)] + \lambda \|\beta\|^2$$

Loss Functions

For $Y \in \{-1, 1\}$

Log-likelihood: $L(Y, f(X)) = \log\left(1 + e^{-Yf(X)}\right)$

- (negative) binomial log-likelihood or deviance.
- estimates

$$f(X) = \log \frac{\Pr(Y = 1|X)}{\Pr(Y = -1|X)}$$

SVM: $L(Y, f(X)) = (1 - Yf(X))_+$.

- Called "hinge loss"
- Estimates

$$C(x) = \text{sign}\left(\Pr(Y = 1|X) - \frac{1}{2}\right)$$

SVM and Function Estimation

SVM with general kernel K solves:

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

with $f = b + h$, $h \in \mathcal{H}_K$, $b \in \mathcal{R}$. $\mathcal{H}_K$ is the reproducing kernel Hilbert space (RKHS) of functions generated by the kernel K. The norm $\|f\|_{\mathcal{H}_K}$ is generally interpreted as a roughness penalty.

More generally we can optimize

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

The solutions have the form

$$\hat{f}(x) = \hat{b} + \sum_{i=1}^{N}\hat{\alpha}_i K(x, x_i),$$

a finite expansion in the representers $K(x, x_i)$.

Aside: RKHS

Function space $\mathcal{H}_K$ generated by a positive (semi-) definite function $K(x, x')$.

$$\text{Eigen expansion: } K(x,y) = \sum_{i=1}^{\infty} \gamma_i \phi_i(x)\phi_i(y)$$

with $\gamma_i \geq 0$, $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$.

$f \in \mathcal{H}_K$ if

$$\begin{aligned} f(x) &= \sum_{i=1}^{\infty} c_i \phi_i(x) \\ c_i &= \int \phi_i(t) f(t) dt \\ ||f||^2_{\mathcal{H}_K} &\stackrel{\text{def}}{=} \sum_{i=1}^{\infty} c_i^2/\gamma_i < \infty \end{aligned}$$

The squared norm $J(f) = ||f||^2_{\mathcal{H}_K}$ is interpreted as a roughness penalty.

Function Estimation in RKHS

$$\min_{f\in\mathcal{H}}\left[\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f)\right]$$

- L is loss function (squared error, log-likelihood, ...)
- J is roughness penalty

With $\mathcal{H} = \mathcal{H}_K$, $J(f) = ||f||^2_{\mathcal{H}_K}$,

$$\min_{f\in\mathcal{H}_K}\left[\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda||f||^2_{\mathcal{H}_K}\right]$$

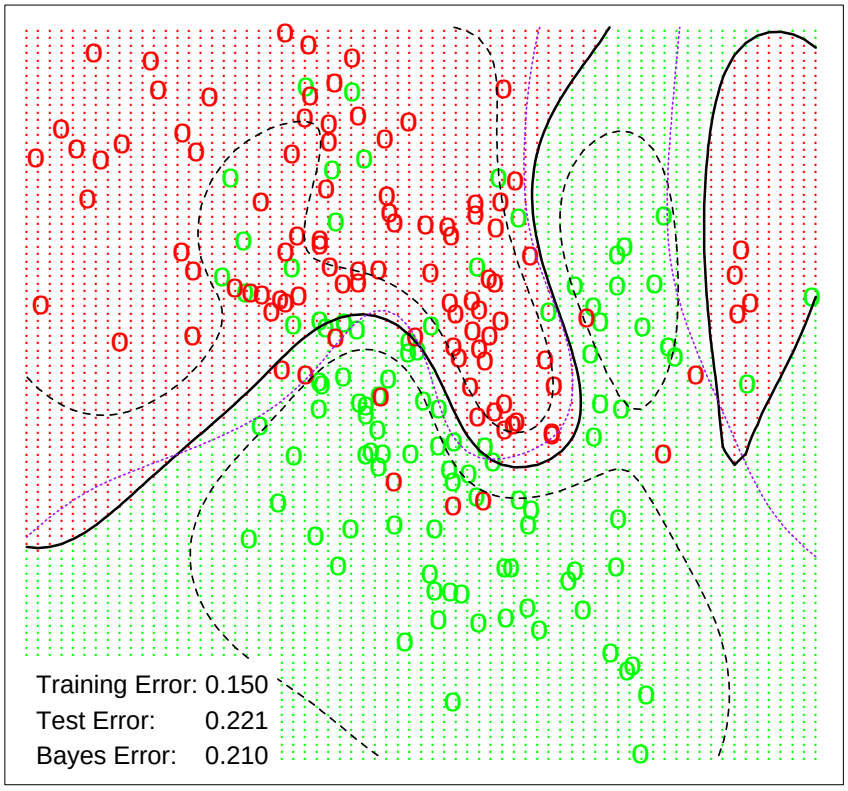
Solution (Wahba, 1990) is finite dimensional, and has form

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

Equivalent finite dimensional criterion (in matrix notation):

$$\min_{\alpha} L(\mathbf{y}, \mathbf{K}\alpha) + \lambda\alpha^T\mathbf{K}\alpha.$$

Kernel Logistic Regression



Replace $(1 - yf)_+$ with $\ln(1 + e^{-yf})$, the binomial deviance.

Note that $\Pr(Y = 1|x) = e^{f(x)}/(1 + e^{f(x)})$, so class probabilities directly available.

We have graphed the 0.5 (solid), 0.25, and 0.75 (broken) contours of $\hat{\Pr}(Y = 1|x)$.

The optimal Bayes decision boundary is purple.

Advantages: KLR vs SVM

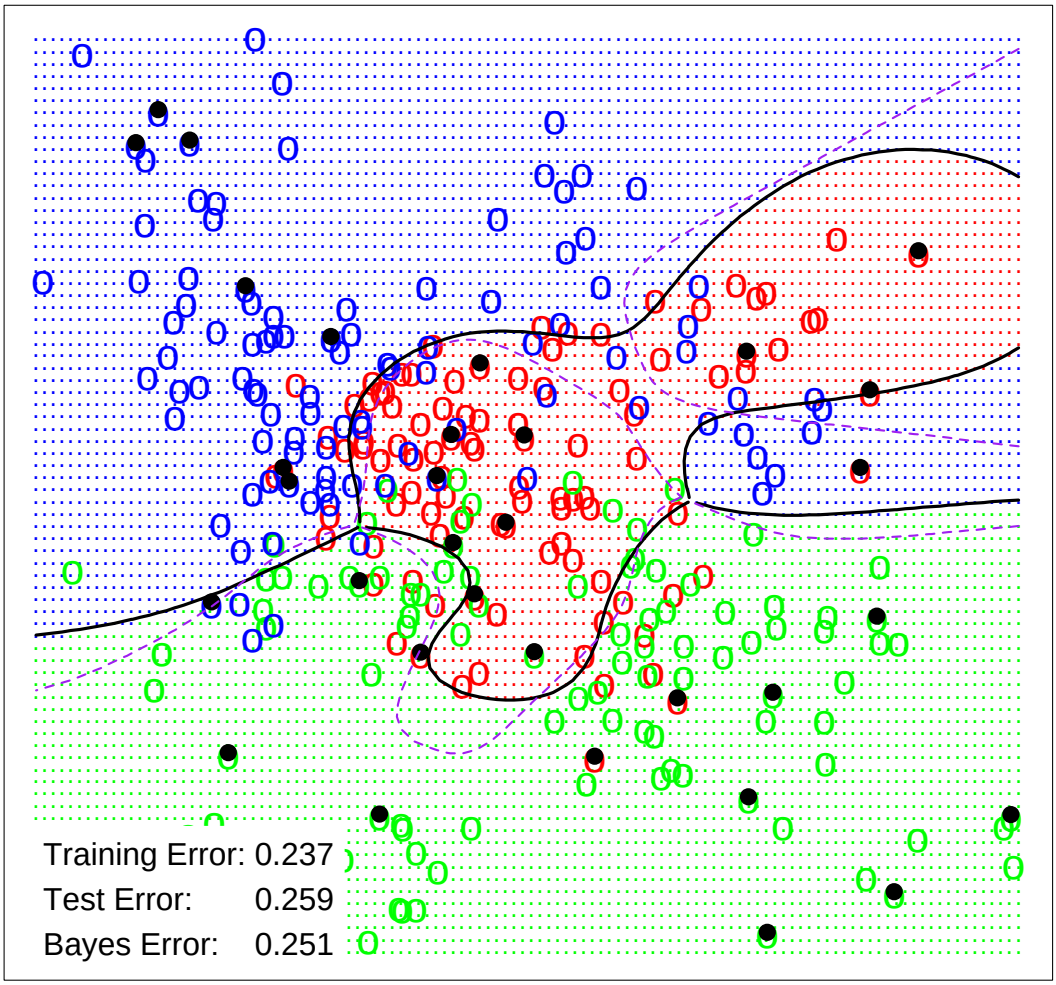
- The classification performance is very similar.
- Offers an estimate of the class probabilities. Often these are more useful than the classifications (e.g. credit risk scoring).
- Generalizes naturally to M-class classification through kernel multi-logit regression:

$$\begin{aligned} f_1(x) &= \log \frac{\Pr(Y=1|x)}{\Pr(Y=M|x)} \\ f_2(x) &= \log \frac{\Pr(Y=2|x)}{\Pr(Y=M|x)} \\ &\vdots \quad \vdots \\ f_{M-1}(x) &= \log \frac{\Pr(Y=M-1|x)}{\Pr(Y=M|x)}. \end{aligned}$$

Fit using multinomial log-likelihood and penalty $\sum_{j=1}^{M-1} \|f_j\|_{\mathcal{H}_K}$.

3 class example

Multi-class IVM - with 32 import points



Disadvantages: KLR vs SVM

- Computationally more expensive $O(N^3)$ versus $O(N^2 m)$, where m is the number of support points. In noisy problems, m can be large, approx $N/2$.
- With KLR fit $\hat{f}(x) = \hat{b} + \sum_{i=1}^{N} \hat{\alpha}_i K(x, x_i)$, all the $\hat{\alpha}_i$ are typically nonzero. For the SVM, only the support points have nonzero $\hat{\alpha}_i$. This allows for a useful data compression and quicker lookup.

Import Vector Machine

A KLR model that uses only a subset of the kernel basis functions $K(x, x_i)$ to approximate the full fit.

Basic idea:

- Suppose $\mathcal{S} \subset \{1, \ldots, N\}$ represents a subset of the training data points. Consider the function

$$f^{\mathcal{S}}(x) = b + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i).$$

- Define

$$R(\mathcal{S}) = \sum_{i=1}^{N} L(y_i, f^{\mathcal{S}}(x_i)) + \lambda \| f^{\mathcal{S}} \|^2_{\mathcal{H}_K},$$

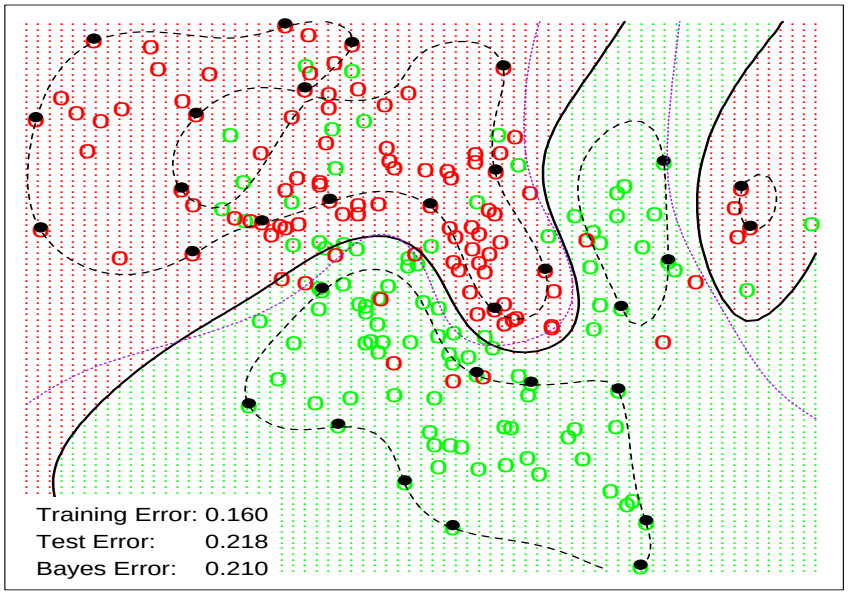
  where L is the binomial deviance.

- For a fixed $\epsilon > 0$ (and fixed λ) solve

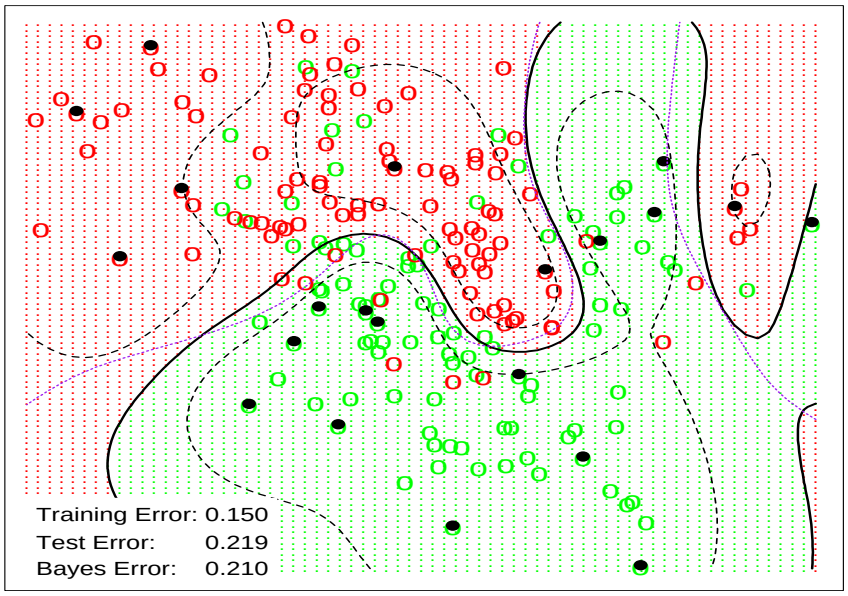
$$\min_{\mathcal{S}} |\mathcal{S}| \text{ subject to } \frac{R(\mathcal{S}) - R(\{1, \ldots, n\})}{R(\{1, \ldots, n\})} < \epsilon$$

Comparison of SVM and IVM

SVM - with 107 support points



IVM - with 21 import points



Operational Details

A useful identity:

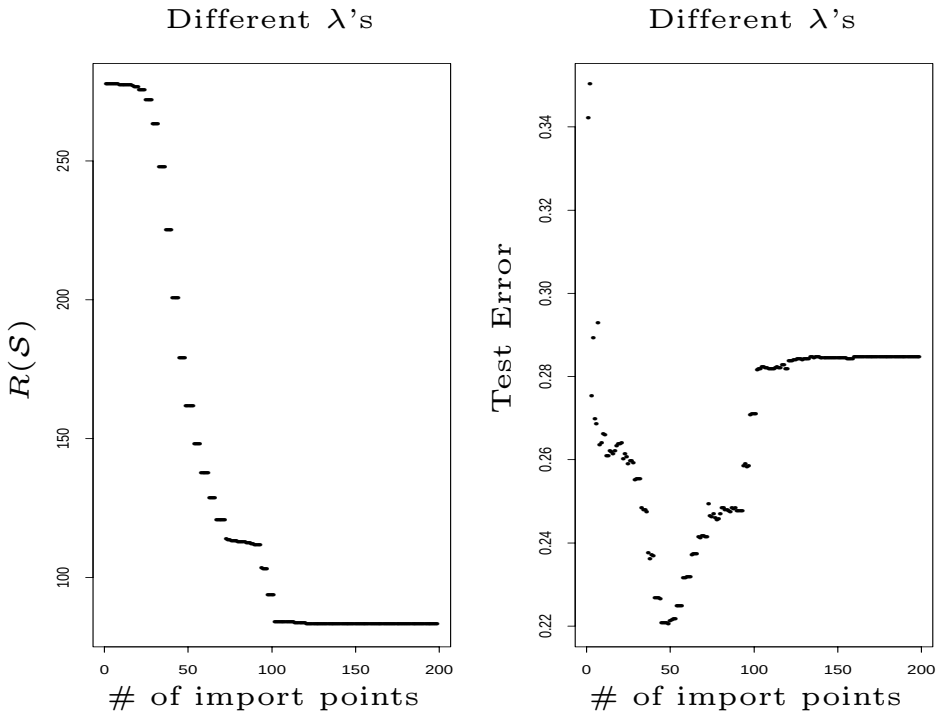
$$\|f^{\mathcal{S}}\|^2_{\mathcal{H}_K} = \alpha^T_{\mathcal{S}}\mathbf{K}\alpha_{\mathcal{S}},$$

where $\alpha_{\mathcal{S}}$ is the coefficient vector with nonzero entries in the $\mathcal{S}$ positions, and $\{\mathbf{K}\}_{ij} = K(x_i, x_j)$.

The IVM problem is combinatorial and computationally prohibitive. We use several approximations to achieve an $O(N^2)$ algorithm:

- Use a greedy forward algorithm to build up $\mathcal{S}$ one element at a time.
- Use a quadratic approximation to $R(\mathcal{S} \cup \{j\})$, $j \notin \mathcal{S}$ at each stage to identify the next basis function.
- Replace the termination criterion with $\frac{R(\mathcal{S}_\ell) - R(\mathcal{S}_{\ell+1})}{R(\mathcal{S}_{\ell+1})} < \epsilon$.

Choosing λ



We can combine the selection of the import vectors and the selection of λ:

- Start with $\lambda_1 = \lambda_{max}$ very large and $\mathcal{S}_1 = \emptyset$.
- Construct a sequence λ_i decreasing from λ_1 to $\lambda_M = 0$ (on exponential scale).
- Obtain (by optimization) a corresponding sequence $\cdots \mathcal{S}_i \subseteq \mathcal{S}_{i+1} \cdots$ in a sequential fashion.
- Monitor the test error on a validation set to pick the optimal λ_i.

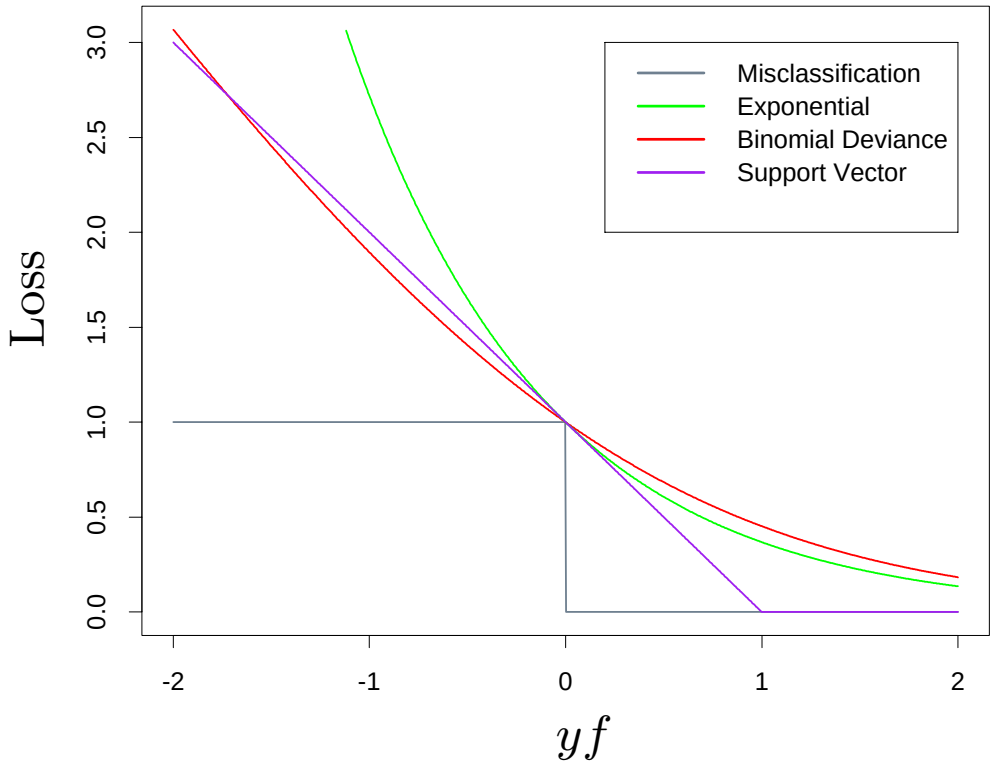
Other Issues

- Many functions spaces to chose from; multi-dimensional smoothing splines popular choice. IVM is computational attractive alternative to the usual $O(N^3)$ algorithms.
- Function fitting is cursed in high dimensions. Both SVM and IVM suffer. Much work in statistics in the last 20 years to develop compromise models, such as low-order ANOVA decompositions for $x \in \mathbb{R}^p$:

$$f(X) = \beta + \sum_{j=1}^{p} g_j(X_j) + \sum_{j \neq k} h_{jk}(X_j, X_k) + \ldots$$

see Hastie, Tibshirani and Friedman (2001) *Elements of Statistical Learning*, Springer, NY, for a survey off such methods.

SVM, KLR and Boosting?



- Boosting builds a sequence of models $f_J(x) = \sum_{j=1}^{J} g_j(x)$, where each $g_j(x)$ is a "weak" classifier fit to weighted training data.
- Even though at stage J, $f_J(x)$ may have zero training errors, boosting increases the "margin".
- Actually, boosting is fitting the model $f(x) = \log \Pr(Y = 1|x)/\Pr(Y = -1|x)$ by stagewise optimization of the loss function $L(Y, f) = \exp(-Yf)$ (Friedman, Hastie and Tibshirani (2000), *Annals of Statistics*.