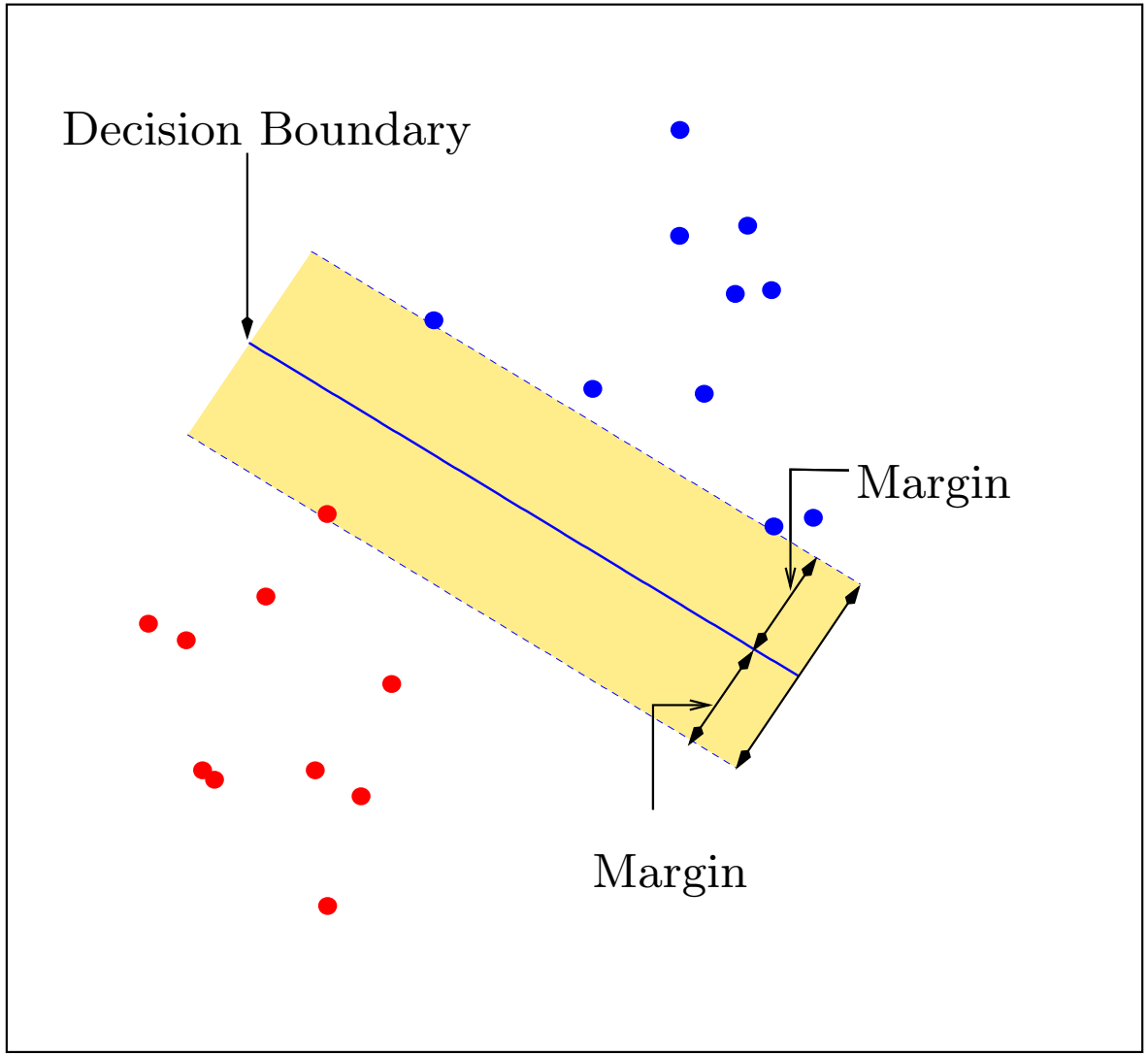
Support Vector Machines + Kernel Technology

- Maximal Margin Classifier
- Support Vector Machine
- Kernel Machines
- Controlling Model Complexity
- Examples

Two-class Classification

- Observations are divided into two classes, coded by a response variable Y taking values $+1$ or -1.
- We have a *feature vector* $X = (X_1, X_2, \ldots, X_p)$, and we hope to build a classification rule $C(X)$ to assign a class label to an individual with feature X.
- We have a sample of pairs (y_i, x_i), $i = 1, \ldots, N$. Note that each of the x_i are vectors $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$.
- Example: Y indicates whether an email is spam or not. X represents the relative frequency of a subset of specially chosen words in the email message.
- The technology described here estimates $C(X)$ directly.

Maximum Margin Classifier



Maximum Margin Classifier

- Find the linear *decision boundary* that creates the biggest gap between the blue points (+1) and red points (-1).
- The idea is that if the gap is big on the training data, it will also be big on any future test data.
- This is a classical EE problem in *convex optimization*, and can be solved with standard quadratic programming methodology.
- Caveat: the training data must be separable!

Maximum Margin Classifier — Details

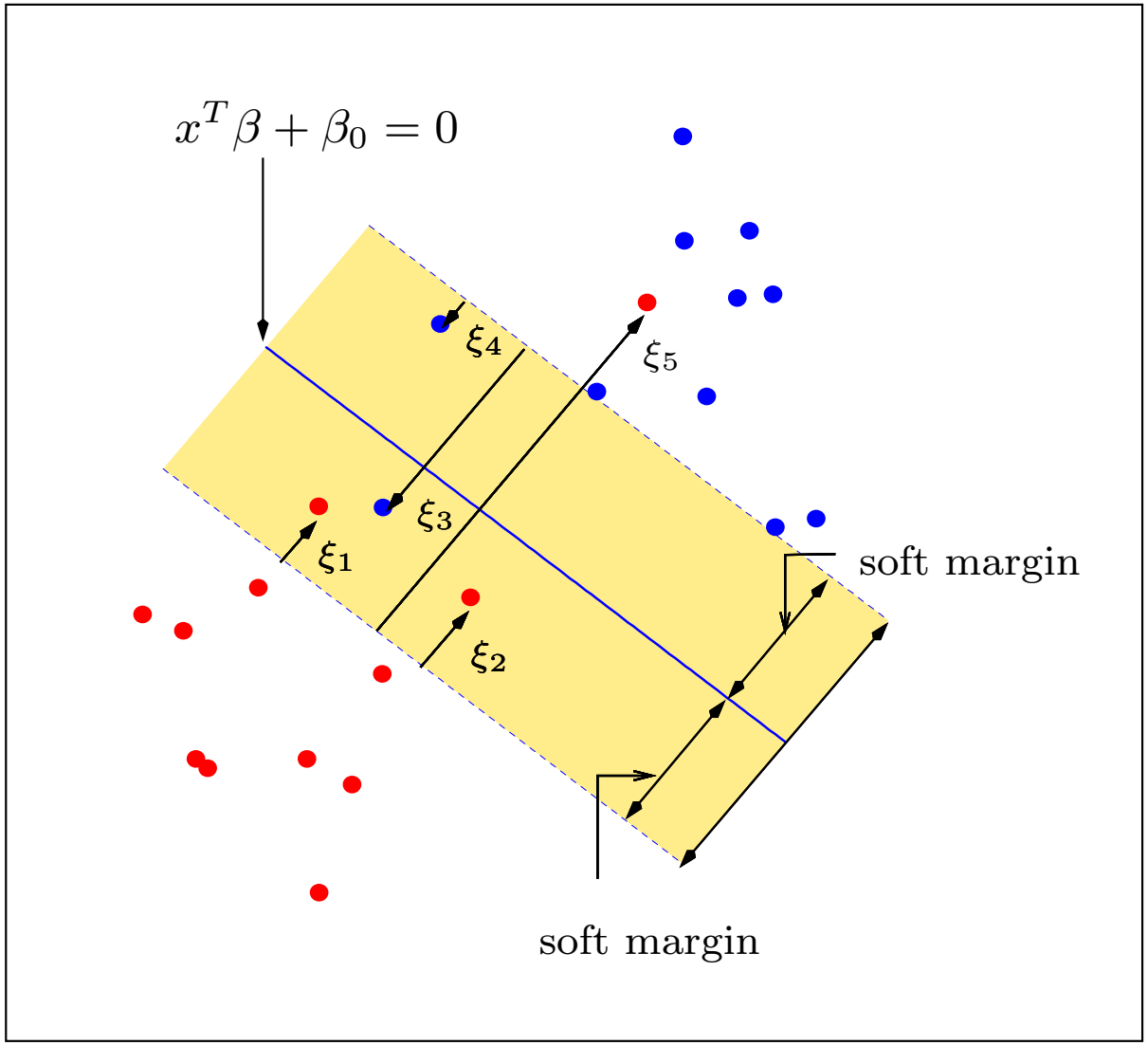
- The procedure fits a linear function $f(X) = \alpha_0 + X^T\beta$.
- The classifier $C(X)$ is defined to be

$$C(X) = \begin{cases} +1 & \text{if } f(X) > 0 \\ -1 & \text{if } f(X) < 0 \\ \text{spin a coin} & \text{if } f(X) = 0 \end{cases}$$

- The decision boundary is the set of values of X for which $f(X) = 0$.
- The solution coefficient $\hat{\beta}$ is defined in terms of a (often small) set of *support points* (points on the boundary; see previous figure):

$$\hat{\beta} = \sum_{j \in \mathcal{S}} \alpha_j x_j$$

Overlapping Classes and Soft Margins



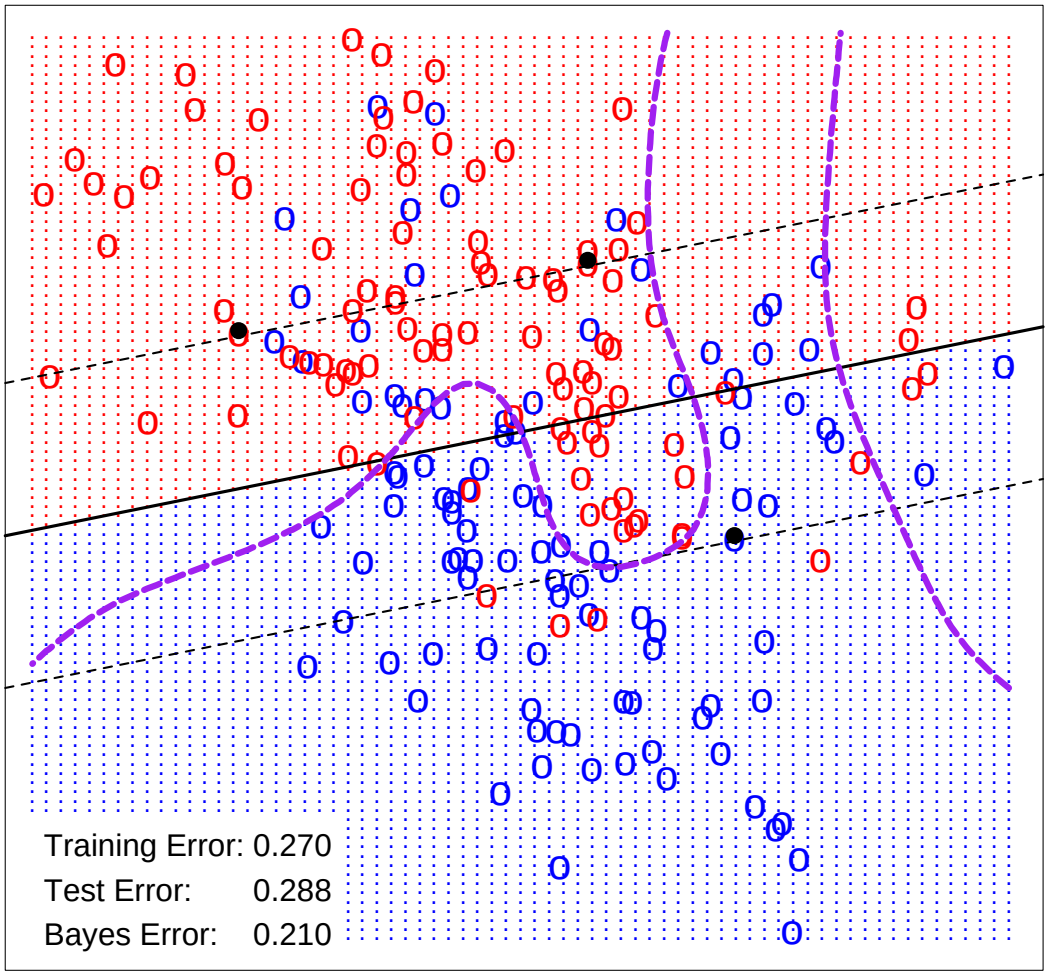
Soft Margin Classifier — Details

- Allow points to overlap the margin, making it a *soft margin*. Let ξ_i be the distance by which point i is on the *wrong side* of its margin.
- Find the linear *decision boundary* that creates the biggest gap between the blue points (+1) and red points (-1), but allowing a total budget B of cumulative overlap. i.e.

$$\sum_{i=1}^{N} \xi_i \leq B.$$

- The solution takes the same form as before; $f(X) = \alpha_0 + X^T\beta$ with $\hat{\beta} = \sum_{j\in\mathcal{S}} \alpha_j x_j$.
- Here the set of support points $\mathcal{S}$ are points on the boundary, or on the wrong side of their boundary.

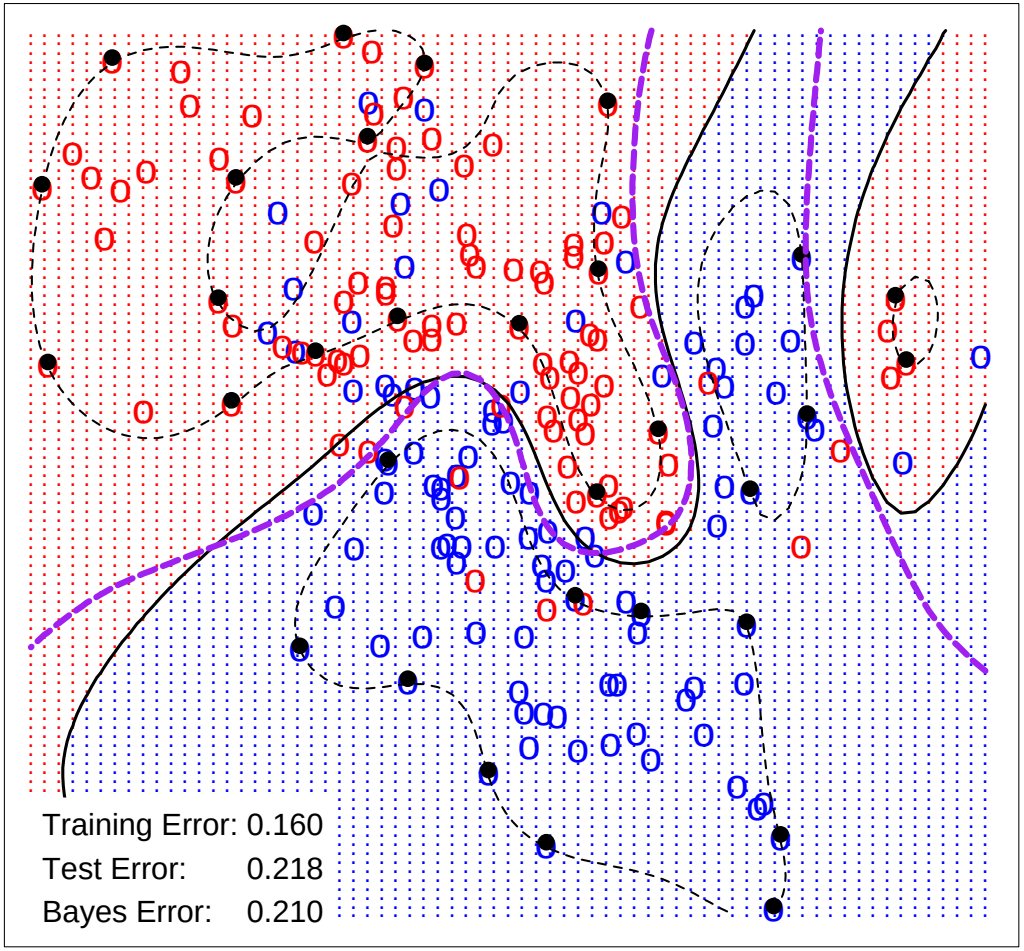
Linear Decision Boundary is often too Restrictive



Feature Space Enlargement

- Expand the original variables $X = (X_1, X_2, \ldots, X_p)$ into a vector of *basis functions* $h(X) = h_1(X), h_2(X), \ldots, h_M(X))$. Note: typically $M \gg p$.
- Example: degree-2 polynomials.
  $h_1(X) = X_1$, ..., $h_p(X) = X_p$, $h_{p+1}(X) = X_1^2$, ...,
  $h_{p+2}(X) = X_1 \cdot X_2$, ..., $h_M = X_p^2$.
- In the enlarged space, fit the linear model
  $f(X) = \alpha_0 + h(X)^T\beta$ using the techniques described above.
- In the enlarged space with coordinates $h = h(X)$, the decision boundary $\alpha_0 + h^T\beta = 0$ is linear.
- In the original space $f(X) = 0$ is nonlinear and can adapt to the data better.

Radial Kernel Support Vector Machine



Kernels

- If $f(x) = \alpha_0 + x^T\beta$ and $\beta = \sum_{j\in\mathcal{S}} \alpha_j x_j$, then

$$f(x) = \alpha_0 + \sum_{j\in S} \alpha_j \langle x, x_j\rangle,$$

  where $\langle x, x_j\rangle = x^T x_j$, the *inner product*.

- For certain basis expansions $h(x)$, there exists a simple-to-compute *kernel function* K such that

$$K(x, x_j) = \langle h(x), h(x_j)\rangle.$$

  For example, for the polynomials above,
  $K(x, x_j) = (1 + \langle x, x_j\rangle)^2 = \langle h(x), h(x_j)\rangle$.

- In general $K(x, x_j) = (1 + \langle x, x_j\rangle)^d = \langle h(x), h(x_j)\rangle$ where h is a polynomial expansion of x with total degree d.

Kernels continued

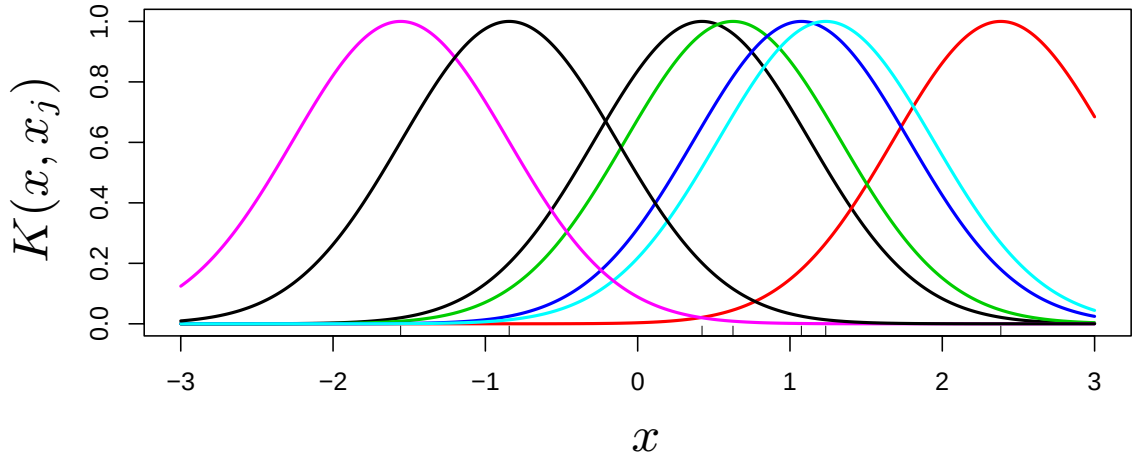
- With such a kernel, $f(x) = \alpha_0 + \sum_{j \in S} \alpha_j K(x, x_j)$.
- The *radial kernel* is defined

$$K(x, x_j) = e^{-\gamma ||x - x_j||^2}.$$

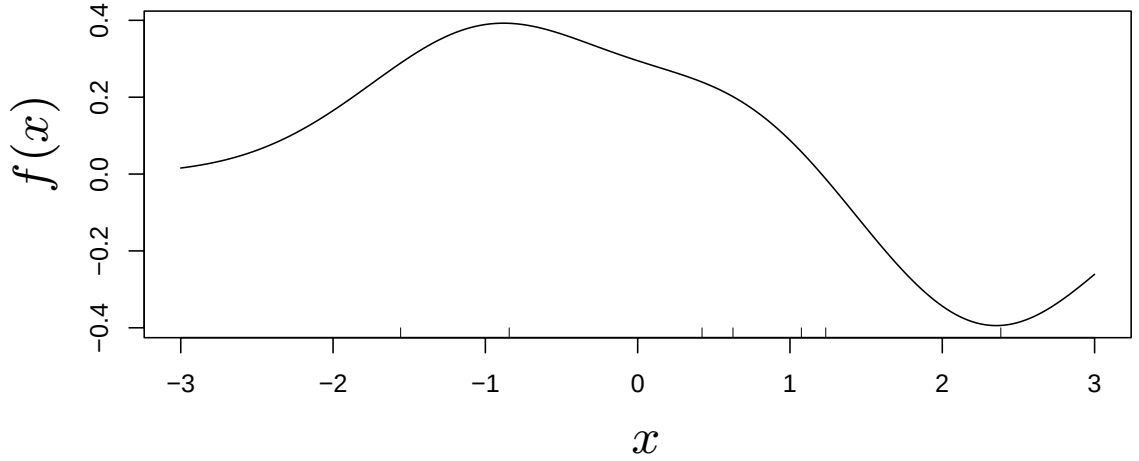
  Also known as a radial basis function (RBF).

- The *implicit basis* $h(x)$ for a radial kernel is in theory infinite dimensional, and in practice very high dimensional.
- The kernel parameter γ controls the width of the kernel.
- The bound $\sum_{i=1}^{N} \xi_i = B$ controls how wiggly the function can get. If B is small, it will try and make no errors, and be wiggly.

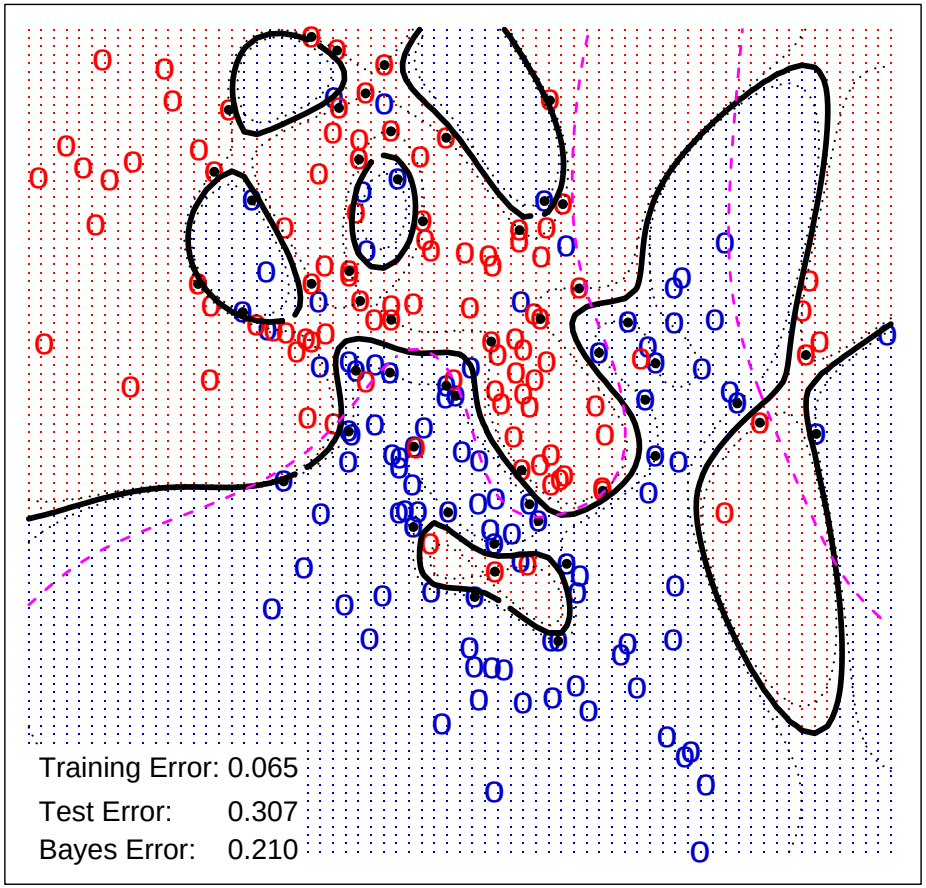
Radial Basis Functions



$$f(x) = \alpha_0 + \sum_j \alpha_j K(x, x_j)$$

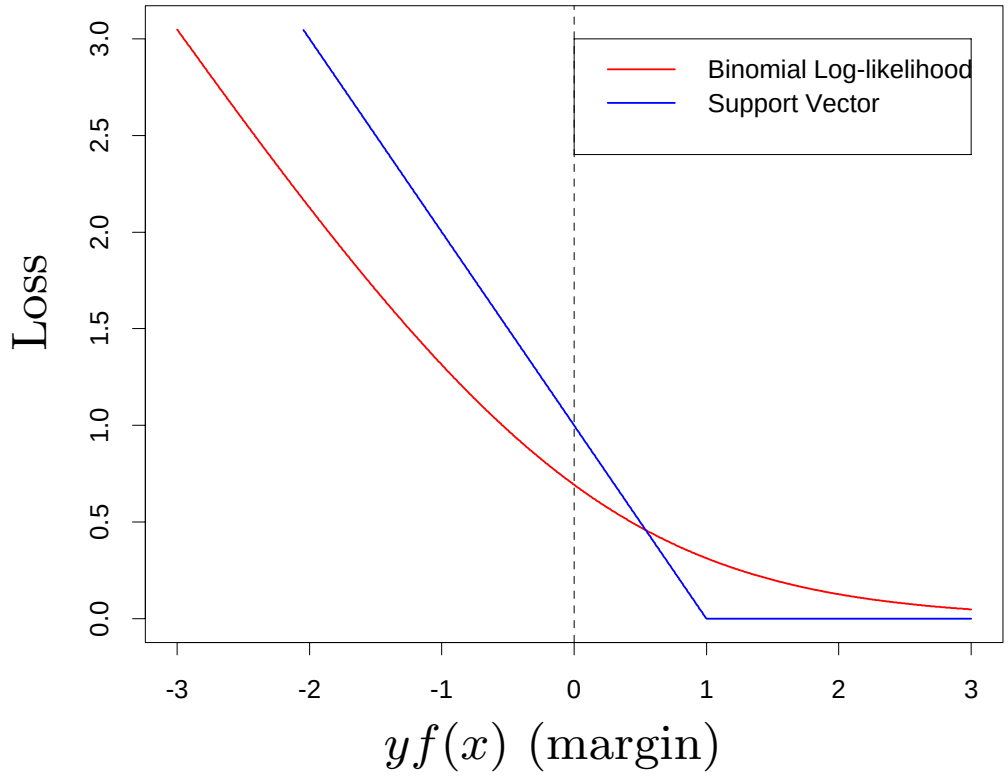


Radial Kernel SVM - small B



MOVIES!

SVM via Loss + Penalty



With $f(x) = x^T\beta + \beta_0$ and $y_i \in \{-1, 1\}$, consider

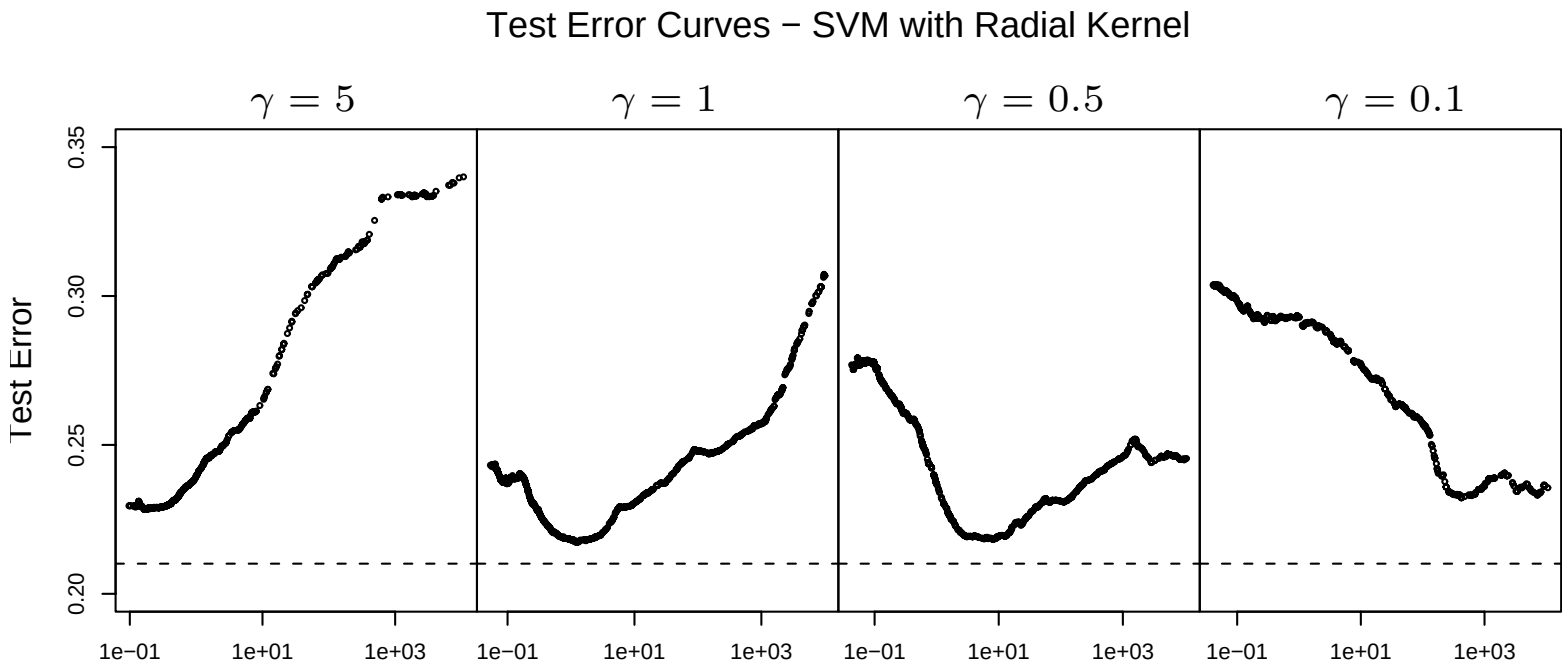
$$\min_{\beta_0,\,\beta} \sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \frac{\lambda}{2}\|\beta\|^2$$

This *hinge loss* criterion is equivalent to the SVM, with λ monotone in B. Compare with

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

This is *binomial deviance loss*, and the solution is "ridged" linear logistic regression.

The Need for Regularization



- γ is a kernel parameter: $K(x, z) = \exp(-\gamma||x - z||^2)$.
- λ (or C) are regularization parameters, which have to be determined using some means like cross-validation.

Kernel Machines

The kernel representation $f(x) = \alpha_0 + \sum_{j=1}^{N} \alpha_j K(x, x_j)$ has many applications:

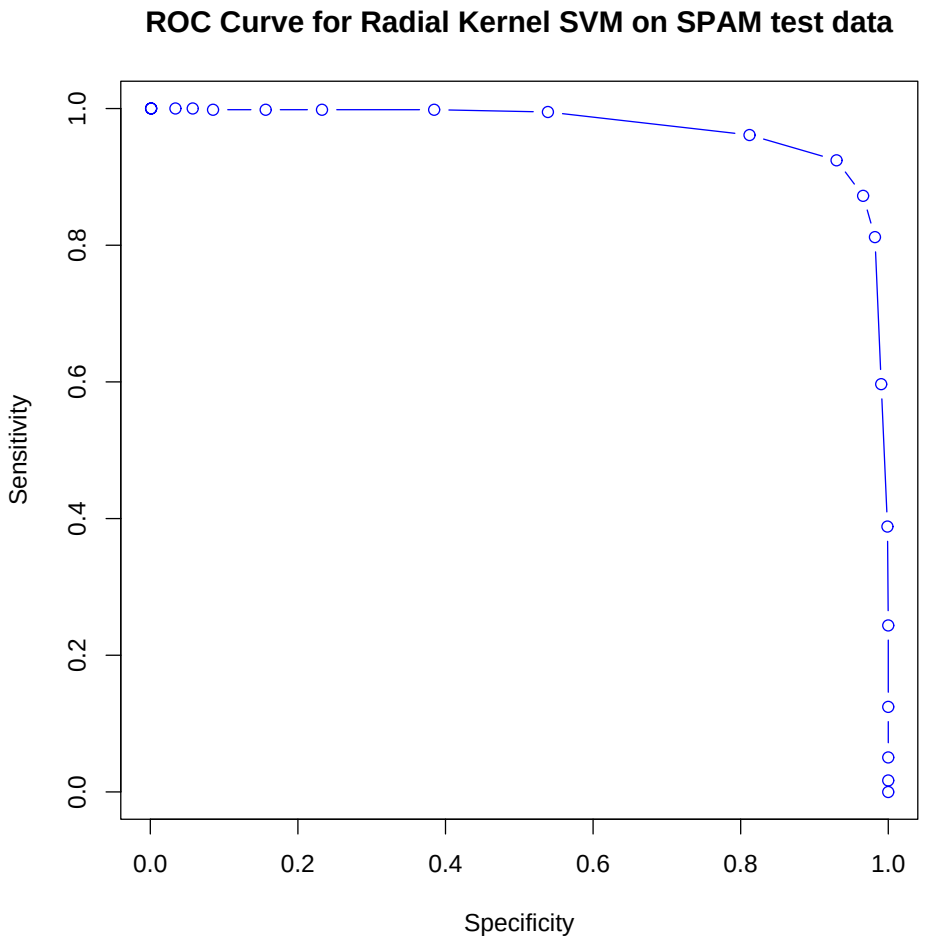
- Regression and generalized regressions; for example the logistic regression model

$$\log\left(\frac{\Pr(Y=+1|x)}{\Pr(Y=-1|x)}\right) = \alpha_0 + \sum_{j=1}^{N} \alpha_j K(x, x_j)$$

- Kernel principal components and canonical correlation analysis
- Kernel discriminant analysis
- Any linear model can be *kernelized*.

Examples

- Email spam-filtering. Classify email as *spam* or *email*, based on the relative frequency of 57 commonly used words and tokens (adapted to user).
- Cancer classification. Based on RNA expression for thousands of genes, classify a cancer sample into one of 14 cancer classes.



SPAM Data

Overall error rate on test data: 6.7%.

ROC curve obtained by varying the *threshold* c_0 of the classifier:

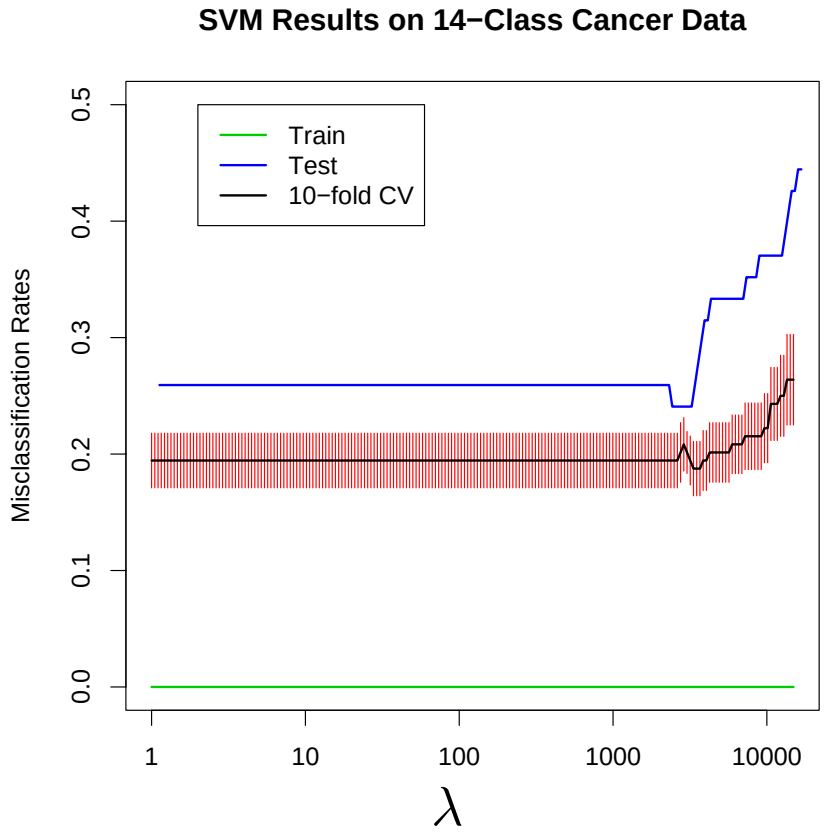
$C(X) = +1$ if $f(X) > c_0$.

Sensitivity: proportion of true spam identified

Specificity: proportion of true email identified.

We may want specificity to be high, and suffer some spam:

Specificity : 98% $\Longrightarrow$ Sensitivity : 82%



- Expression data for 14 cancer classes (Ramaswamy et al, 2001). 144 training observations, 54 test observations, 16000 genes.
- Modeled by 14 different SVMs; each class vs the rest.

Caveats

- Kernel methods do not scale well; computational costs run at $O(N^2 \cdot M)$ where $M \leq N$. Feasible for problems around 5–10K max observations.
- Kernel methods do not do variable selection in any reasonable or automatic way.
- Potentially they suffer if the number of features is large, and a large fraction of them are garbage.

Software

- SVM^{light}: Thorsten Joachims - free software in C, known for quality and speed.
- *LIB SVM*: free software based on Platt's SMO algorithm and Joachims code, written by Chih-Chung Chang and Chih-Jen Lin.
- *Equbits*: Commercial software package which automates the tuning and model selection with SVMs