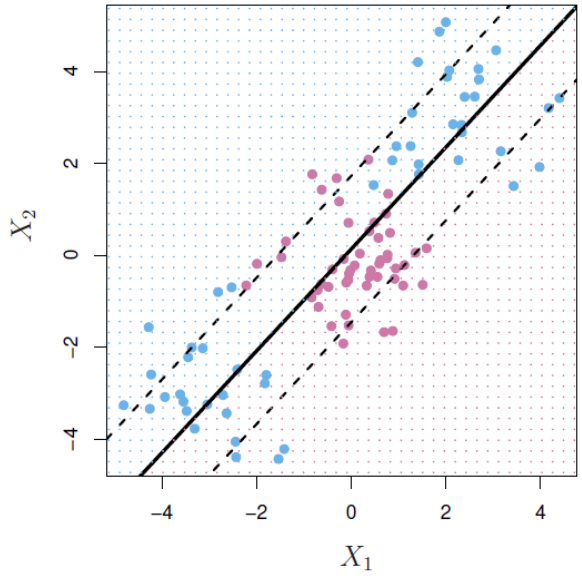
9.3-9.5 Support Vector Machines

Non-Linear Decision Boundaries



Sometime a linear boundary simply won't work, no matter what value of C.

The example on the left is such a case.

What to do?

Non-Linear Decision Boundaries

- Consider the linear regression case.
  - If a linear relationship fails, we enlarge the feature space using some $f(X)$.
- In the support vector classifier, we can do something similar.
  - Include transformations on X to go from a p-dimensional space to a $> p$ dimensional space.
  - This results in non-linear decision boundaries in the original space.

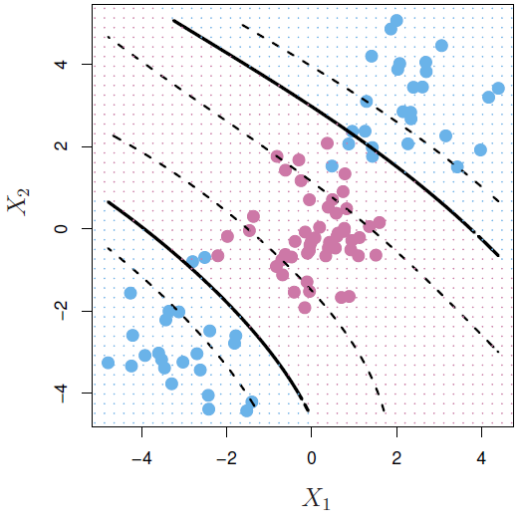
Example

Suppose we have two features (X_1, X_2) and use $(X_1, X_2, X_1^2, X_2^2, X_1X_2)$.

Then the decision boundary would be of the form

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 = 0$$

Example



$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 + \beta_6 X_1^3 + \beta_7 X_1^2 + \beta_9 X_1 X_2^2 + \beta_9 X_1^2 X_2 = 0$$

Nonlinearities and Kernels

- Polynomials get out of control quickly.
- *Kernels* are a more controlled way to introduce nonlinearities in support vector classifiers.
- A kernel is a function that quantifies the similarity of two observations.
  - For example, we might take $K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$

Inner Products and Support Vectors

- The inner product between two vectors is

$$< x_i, x_{i'} >= \sum_{j=1}^{p} x_{ij} x_{i'j}$$

- The linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i < x, x_i >$$

- To estimate the parameters, we just need the $\binom{n}{2}$ inner products between all pairs of training observations.
- Since most observations are not involved in the support vectors, most $\hat{\alpha}_i$ will be zero.

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{i \in S} \hat{\alpha}_i < x, x_i >$$

where S is the support set of indices i such that $\hat{\alpha} > 0$

Kernels and Support Vector Machines

- If we can compute inner-products between observations, we can fit a SV classifier.
- Some special *kernel functions* can do this for us.
  - For example,

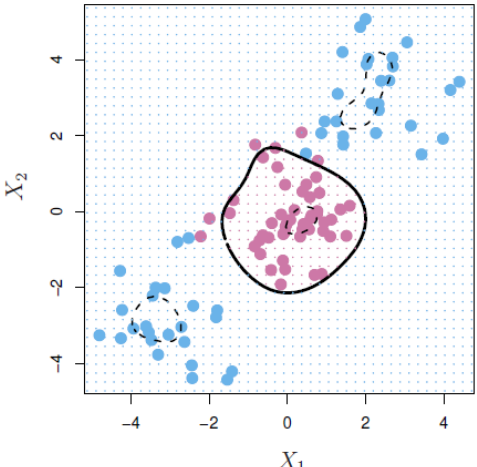
$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d$$

    computes the inner-products needed for d dimensional polynomials, or $\binom{p+d}{d}$ basis functions.
  - The solution has the form

$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i K(x, x_i)$$

Radial Kernel

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2\right)$$

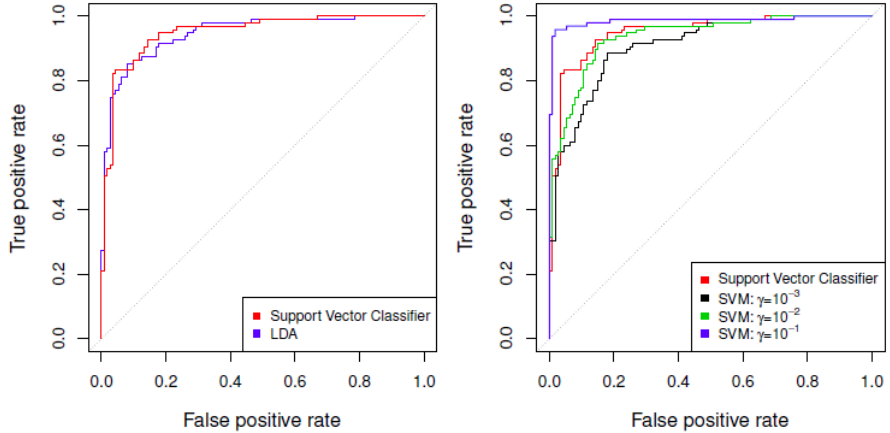


$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \hat{\alpha}_i K(x, x_i)$$

Implicit feature space; very high dimensional.

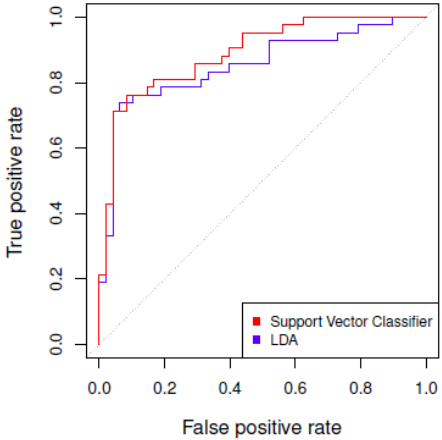
Controls variance by squashing down most dimensions severely

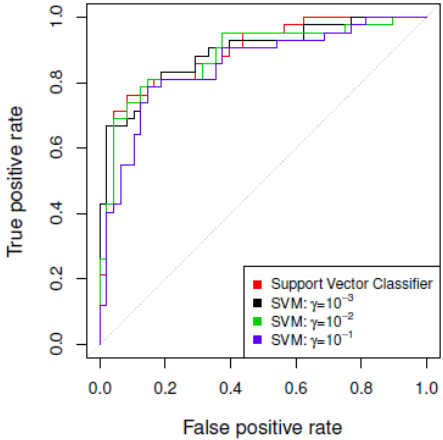
Example: Heart Data



ROC curve is obtained by changing the threshold 0 to threshold t in $\hat{f}(X) > t$, and recording *false positive* and *true positive* rates as t varies. Here we see ROC curves on training data.

Example: Heart Data (Test Data)





What about $K > 2$ classes?

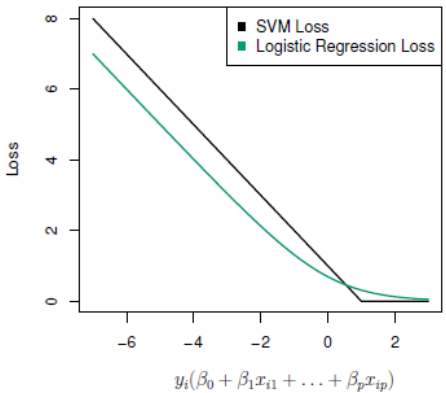
- One versus all (OVA)
  - Fit K different 2-class SVM classifiers (each class vs the rest). Classify x^* to the class for which $\hat{f}_k(x^*)$ is largest.
- One versus one (OVO)
  - Fit all $\binom{K}{2}$ pairwise classifiers. Classify x^* to the class that wins the most pairwise comparisons.

Which to choose? In general, we want to use OVO as long as K isn't too large.

Support Vector vs Logistic Regression

With $f(X) = X\beta$, we can rephrase support vector classifier optimization as

$$\text{minimize}_\beta \left\{ \sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$



This has the form *loss plus penalty*.
The loss is known as the *hinge loss*.
Very similar to "loss" in logistic regression (negative log-likelihood).

Which to Use?

- When classes are (nearly) separable, SVM (and LDA) outperform Log Reg.
- When they are not, Log Reg (with ridge penalty) is very similar to SVM
- If estimating probabilities is important, use Log Reg
- For nonlinear boundaries, kernel SVMs are popular.
  - We can use kernels with Log Reg and LDA as well, but it's more computationally expensive.