Statistical Machine Learning II

International Undergraduate Summer Enrichment Program (IUSEP)

Linglong Kong

Department of Mathematical and Statistical Sciences
University of Alberta

July 19, 2016

Outline

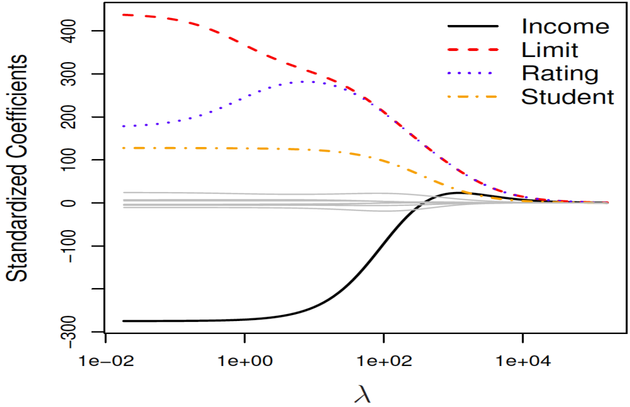
UNIVERSITY OF ALBERTA
EDMONTON·ALBERTA·CANADA

Ridge Regression

- The ridge regression coefficient estimates $\hat{\beta}^R$ are the values that minimize
$$\sum_i \left(y_i - \beta_0 - \sum_j \beta_j x_{ij} \right)^2 + \lambda \sum_j \beta_j^2,$$
where λ is a tuning parameter, to be determined separately.
- The second term $\lambda \sum_j \beta_j^2$ called a shrinkage penalty, is small when $\beta_j,\ j \geq 1$ are close to zero, and so it has the effect t of shrinking the estimates of β_j towards zero.
- The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates.
- Selecting a good value for λ is critical; cross-validation is used for this.

Credit data example

UNIVERSITY OF ALBERTA
EDMONTON·ALBERTA·CANADA



As λ increases, the coefficients are shrunken to zeros.

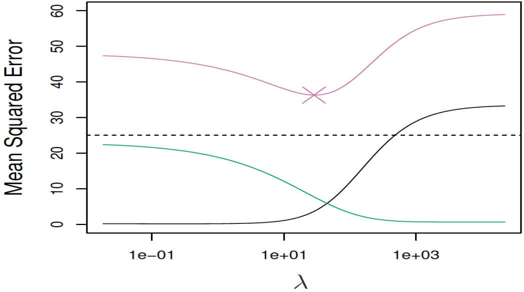
Scaling of predictors

- The standard least squares coefficient estimates are scale equivariant: multiplying X_j by a constant c simply leads to a scaling of the least squares coefficient estimates by a factor of $1/c$. In other words, regardless of how the j-th predictor is scaled $X_j\hat{\beta}_j$ will remain the same.
- In contrast, the ridge regression coefficient estimates can change substantially when multiplying a given predictor by a constant, due to the sum of squared coefficient term in the penalty part of the ridge regression objective function.
- Therefore, it is best to apply ridge regression after standardizing the predictors, using the formula

$$\tilde{x}_{ij} = x_{ij} \Big/ \sqrt{\sum_i (x_{ij} - \bar{x}_j)^2 / n}.$$

Credit data example

UNIVERSITY OF ALBERTA
EDMONTON·ALBERTA·CANADA



Simulated data with $n = 50$ observations, $p = 45$ predictors, all having nonzero coefficient. Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set. The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.

The LASSO

UNIVERSITY OF ALBERTA
EDMONTON·ALBERTA·CANADA

- Ridge regression, unlike subset selection, will generally select models that involve just a subset of the variables, ridge regression will include all p predictors in the final model.
- The LASSO is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficient $\hat{\beta}^L$ minimize the quantity

$$\sum_i \left(y_i - \beta_0 - \sum_j \beta_i x_{ij} \right)^2 + \lambda \sum_j |\beta_j|,$$

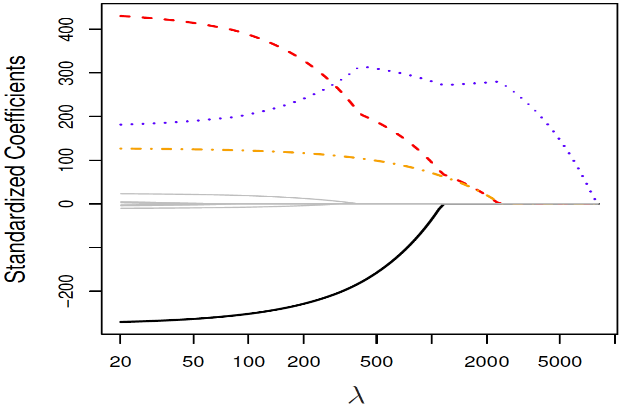
where λ is a tuning parameter.
- The LASSO uses l_1 penalty instead of l_2 (ridge regression).

The LASSO

UNIVERSITY OF ALBERTA
EDMONTON·ALBERTA·CANADA

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero as λ increases.
- However, in the case of the lasso, the l_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficientl large. Thus performs variable selection.
- We say that the lasso yields sparse models — that is, models that involve only a subset of the variables.
- Selecting a good value for λ is critical; cross-validation is again used for this.

Credit data example



As λ increases, the coefficients are shrunken to exact zeros.

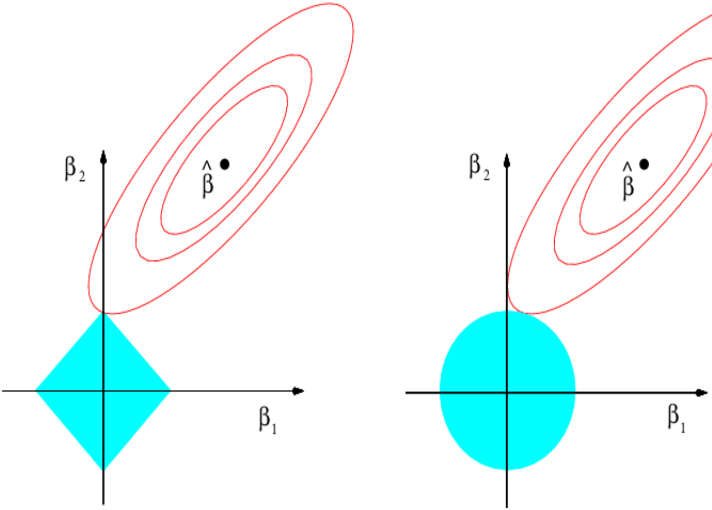
Ridge regression and the LASSO

- Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?
- One can show that the lasso and ridge regression coefficient estimates solve the problems

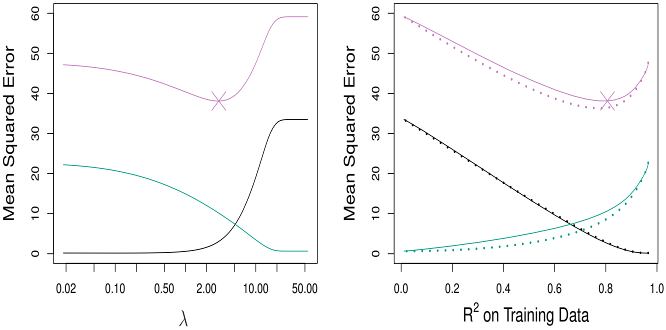
$$\min_{\beta} \sum_i \left(y_i - \beta_0 - \sum_j \beta_i x_{ij} \right)^2, \text{ subject to } \sum_j |\beta_j| \le c;$$

$$\min_{\beta} \sum_i \left(y_i - \beta_0 - \sum_j \beta_i x_{ij} \right)^2, \text{ subject to } \sum_j \beta_j^2 \le c;$$

Ridge regression and the LASSO

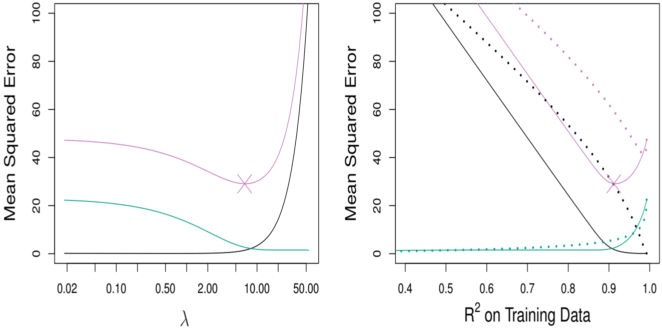


Credit data example



Left: Plots of squared bias (black), variance (green), and test mean squared error (purple) for the LASSO on a simulated data set. Right: Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). The purple crosses indicate the LASSO models for which the MSE is smallest.

Credit data example



Left: Plots of squared bias (black), variance (green), and test mean squared error (purple) for the LASSO on another simulated data set. Right: Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). The purple crosses indicate the LASSO models for which the MSE is smallest.

Conclusions

- These two examples illustrate that neither ridge regression nor the lasso will universally dominate the other.
- In general, one might expect the lasso to perform better when the response is a function of only a relatively small number of predictors.
- However, the number of predictors that is related to the response is never known a priori for real data sets.
- A technique such as cross-validation can be used in order to determine which approach is better on a particular data set.

Separable Hyperplanes

- Imagine a situation where you have a two class classification problem with two predictors X_1 and X_2.
- Suppose that the two classes are linearly separable i.e. one can draw a straight line in which all points on one side belong to the first class and points on the other side to the second class.
- Then a natural approach is to find the straight line that gives the biggest separation between the classes i.e. the points are as far from the line as possible
- This is the basic idea of a support vector classifier.

Hyperplane

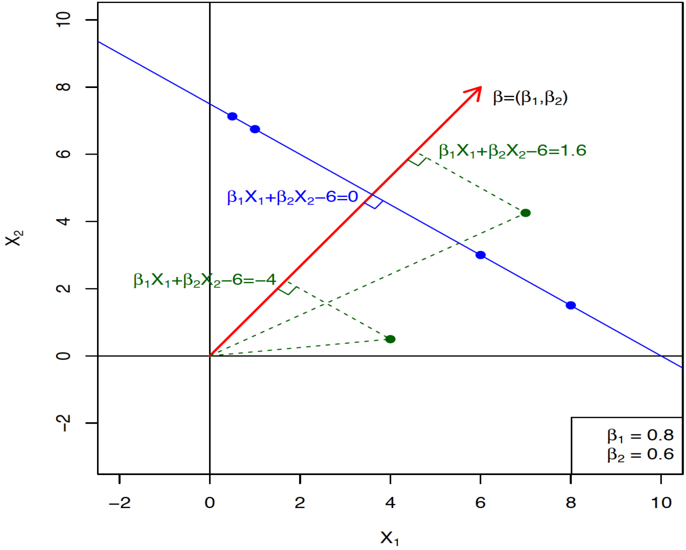
UNIVERSITY OF ALBERTA
EDMONTON·ALBERTA·CANADA

- A hyperplane in p dimensions is a flat affine subspace of dimension $p-1$.
- In general the equation for a hyperplane has the form

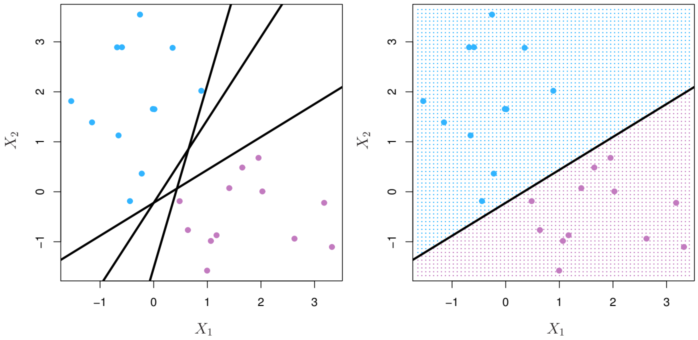
$$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = 0.$$

- In $p = 2$ dimensions a hyperplane is a line.
- If $\beta_0 = 0$, the hyperplane goes through the origin, otherwise not.
- The vector $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)$ is called the normal vector — it points in a direction orthogonal to the surface of a hyperplane.

Hyperplane in 2 Dimensions



Separating Hyperplane



- If $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$, then $f(X) > 0$ for points on one side of the hyperplane, and $f(X) < 0$ for points on the other.
- If we code the colored points as $Y_i = +1$ as blue and $Y_i = -1$ as purple, then if $Y_i \cdot f(X_i) > 0$ for all i, $f(X) = 0$ defines a Separating Hyperplane.

Hard Margin

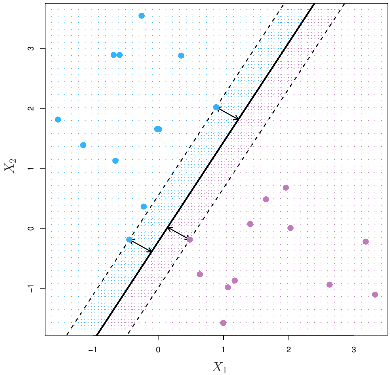
- Among all separating hyperplanes, find the one that makes the biggest gap or margin between the two classes.
- Constrained optimization problem

$$\text{maximize}_{\beta_0,\beta_1,\cdots,\beta_p} M$$

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1$$

$$y_i(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p) \geq M$$

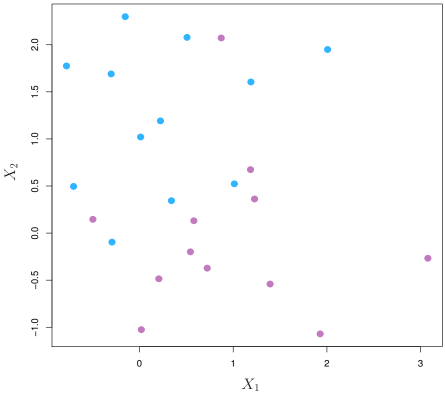
$$\text{for } i = 1, \cdots, n.$$



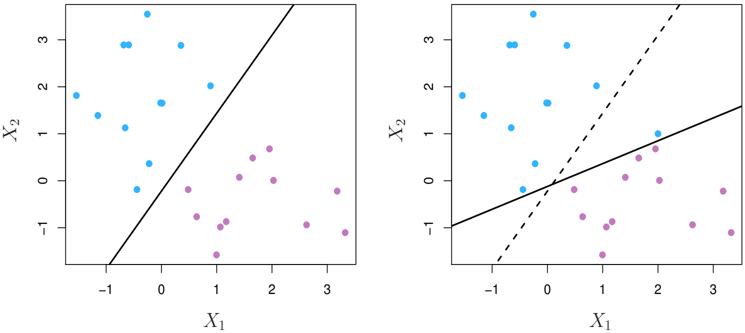
- This can be rephrased as a convex quadratic program, and solved efficiently. The function `svm()` in package `e1071` solves this problem efficiently.

Hard Margin

- The data on the left are not separable by a linear boundary.
- In general it is true for $n < p$.

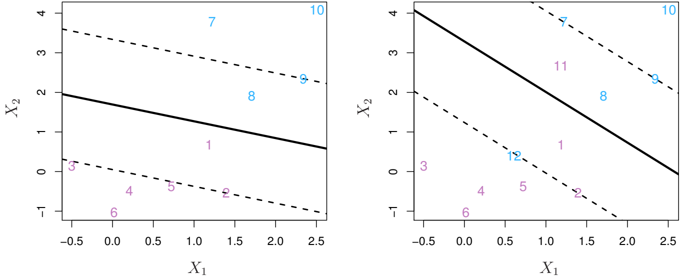


Hard Margin



- Sometimes the data are separable, but noisy. This can lead to a poor solution for the maximal-margin (hard margin) classifier. boundary.

Soft Margin



- The support vector classifier maximizes a soft margin.

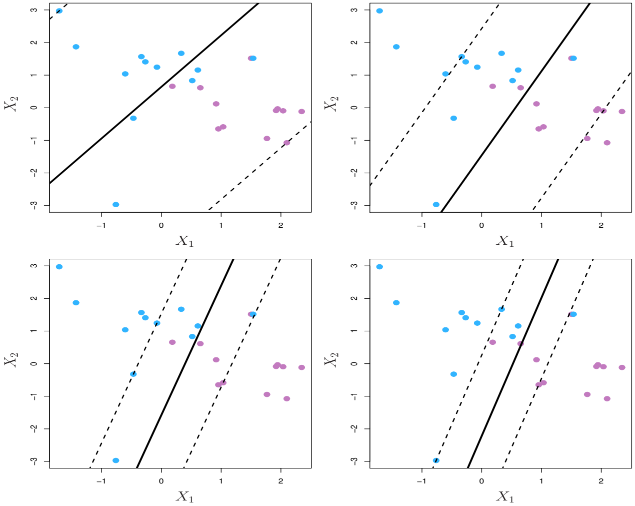
$$\text{maximize}_{\beta_0,\beta_1,\cdots,\beta_p;\epsilon_1,\cdots\epsilon_n} M; \text{ subject to } \sum_{j=1}^{p} \beta_j^2 = 1$$

$$y_i(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p) \geq M(1 - \epsilon_i)$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C.$$

UNIVERSITY OF ALBERTA
EDMONTON·ALBERTA·CANADA

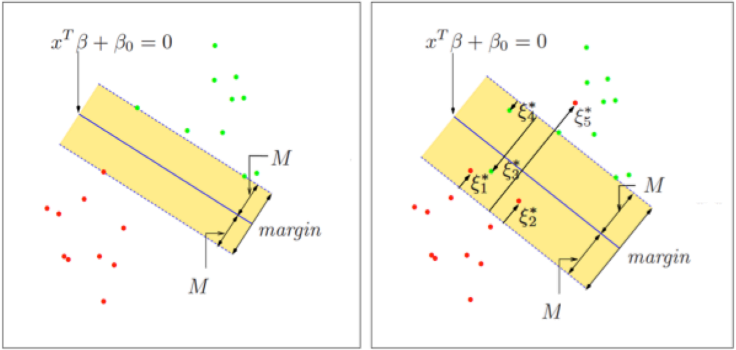
C is a regularization parameter



- C is a regularization parameter and represent the price we need to pay to separate the two classes.

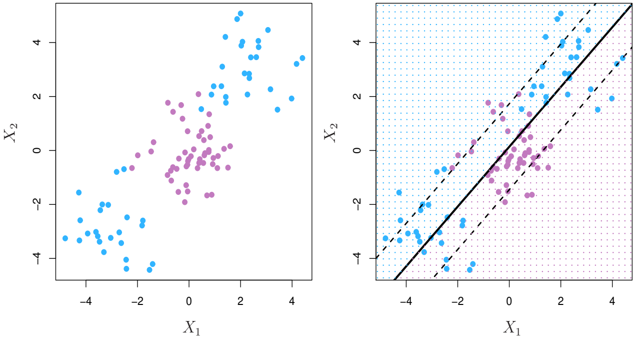
Support Vectors

UNIVERSITY OF ALBERTA
EDMONTON·ALBERTA·CANADA



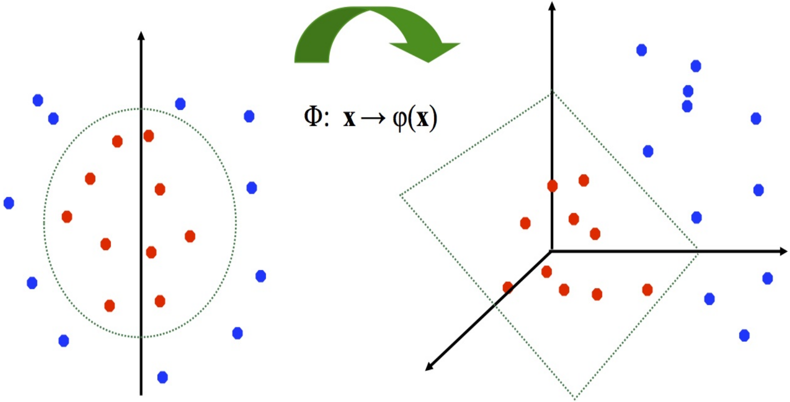
- Only those support vectors determine the optimization solution for both hard margin and soft margin.

Linear boundary can fail



- Sometime a linear boundary simply won't work, no matter what value of C.
- For example, in the situation shown above.
- What do we do? the kernel trick!!!

Feature Expansion



Feature Expansion

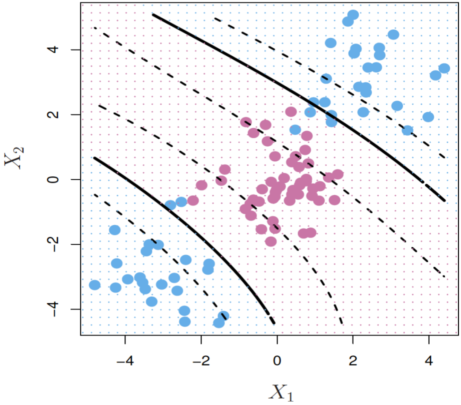
- Enlarge the space of features by including transformations; for example $X_1^2, X_2^3, X_1X_2, X_1X_2^2, \cdots$, Hence go from a p-dimensional space to a $M > p$ dimensional space.
- Fit a support-vector classifier in the enlarged space.
- This results in non-linear decision boundaries in the original space.
- Example: Suppose we use $(X_1, X_2, X_1^2, X_2^2, X_1X_2)$ instead of just (X_1, X_2). Then the decision boundary would be of the form

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

- This leads to nonlinear decision boundaries in the original space (quadratic conic sections).

Cubic Polynomials

- Here we use a basis expansion of cubic polynomials — from 2 variables to 9.
- The support-vectorclassifier in the enlarged space solves the problem in the lower-dimensional space



- The decision boundary is

$$\beta_0+\beta_1X_1+\beta_2X_2+\beta_3X_1^2+\beta_4X_2^2+\beta_5X_1X_2+\beta_6X_1^3+\beta_7X_2^3+\beta_8X_1X_2^2+\beta_9X_1^2X_2=0.$$

Nonlinearities and Kernels

- Polynomials (especially high-dimensional ones) get wild rather fast.
- There is a more elegant and controlled way to introduce nonlinearities in support vector classifier — through the use of kernels.
- Before we discuss these, we must understand the role of inner products in support vector classifier.

Inner products and kernels

- Inner product between vectors

$$\langle x_i, x_{i'} \rangle = \sum_j x_{ij} x_{i'j}.$$

- The linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_i \alpha_i \langle x, x_i \rangle$$

- To estimate parameters $\alpha_1, \cdots, \alpha_n$ and β_0, all we need are $\binom{n}{2}$ inner products $\langle x, x_i \rangle$ between all pairs of training observations.
- It turns out that most of the $\hat{\alpha}_i$ can be zero

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

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

Kernels and Support Vector Machine

- If we can compute inner products between observations, we can fit a support vector classifier — can be very abstract!
- Some special kernel function can do this for us. E.g.

$$K(x_i, x_{i'}) = (1 + \sum_j x_{ij} x_{i"j})^2$$

computes the inner products needed for d dimensional polynomials — $\binom{p+d}{d}$ basis functions!
- The solotion has the form

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

Radial Kernel

- The radial Kernel has the format

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

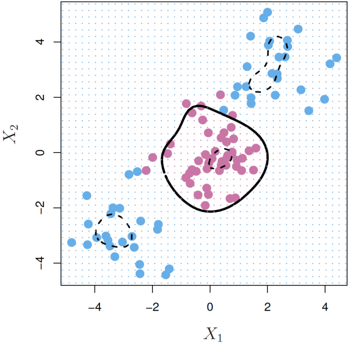
where γ is tuning parameter.

- The decision bounady is,

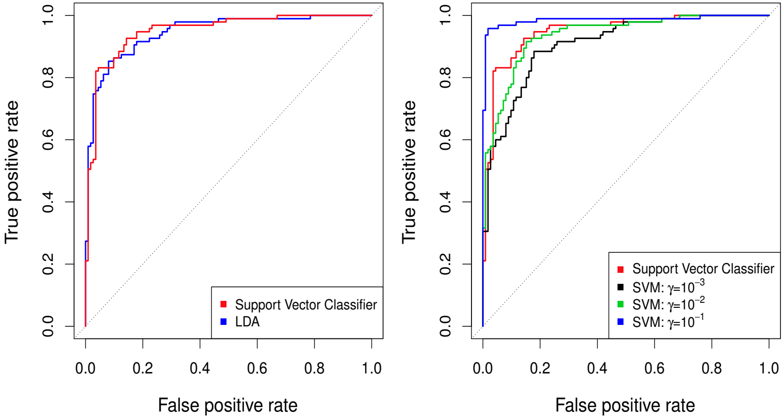
$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \hat{\alpha}_i K(x, x_i),$$

implicit feature space; very high dimensional.

- Controls variance by squaring down most dimenions severely.



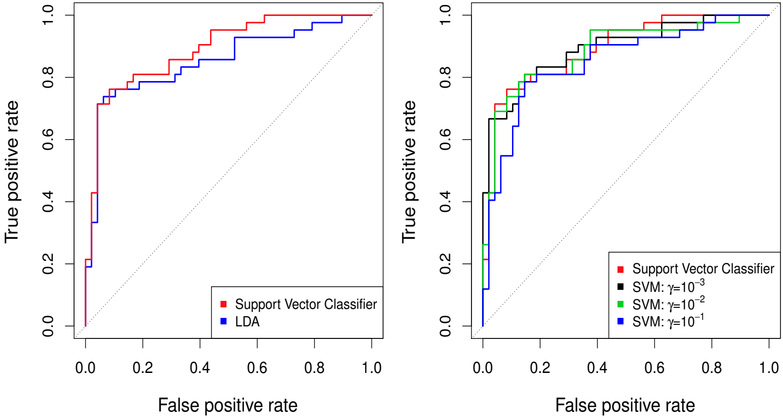
Example - Heart Data



ROC curves on Training 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.

Example - Heart Data



ROC curves on Testing data

SVMs: More than 2 classes

- The SVM as defined works for $K = 2$ classes. What do we do if we have $K > 2$ classes?
- OVA - One versus All. Fit K different 2-class SVM classifiers $\hat{f}_k(x),\ k = 1, \cdots, K$; each class versus the rest. Classify x^* to the class for which $\hat{f}_k(x^*)$ is largest.
- OVO - One versus One. Fit all $\binom{K}{2}$ pairwise classifiers $\hat{f}_{kl}(x)$. Classify x^* to the class that wins the most pairwise competitions.
- Which one to choose? If K is not too large, use OVO.

Summary and Remark

- Install software **R**, if necessary, play demos, browse documentation.
- In my opinion, the best way to learn in this course is to try everything in **R**.
- Once it works, then think why, and how to write it in your own way.