New York University: Machine Learning in Economics

Lecture 5: Kernel Methods and Support Vector Machines

Dr. George Lentzas

Kernel Smoothing

What are Kernel Smoothing Methods?

- Kernel smoothing methods estimate the function $f(X)$ by fitting a different model separately at each point x_0.
- This works by using observations close to this point x_0 but in a way that the estimated function is smooth.
- This localization feature is achieved by a weighting function

$$K_\lambda(x_0, x_i)$$

  which is known as a *Kernel*.
- This requires little or no training; the work happens at evaluation time and the only parameter that needs to be estimated is λ the width of the neighborhood.
- These Kernels are different from the more recent use of the term Kernel, in SVMs.

Kernel Smoothing

One Dimensional Kernel Smoothers

- To avoid discontinuities and bumps when estimating $\hat{f}(X)$ we can assign weights that decrease smoothly with distance from the target point.
- This gives the Nadaraya-Watson estimator

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i)\, y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

- Obviously the ND estimator depends on the choice of Kenrel function to use; a common choice is the Epanechnikov Kernel

$$K_\lambda(x_0, x_i) = D\left(\frac{|x - x_0|}{\lambda}\right)$$

if lambda small: distance large, so you need points VERY near x0 so that they;re selected

with

$$D(t) = \frac{3}{4}\left(1 - t^2\right) \text{ if } |t| \leq 1 \text{ and 0 otherwise.}$$

here we're selecting those points near x0

Kernel Smoothing

One Dimensional Kernel Smoothers

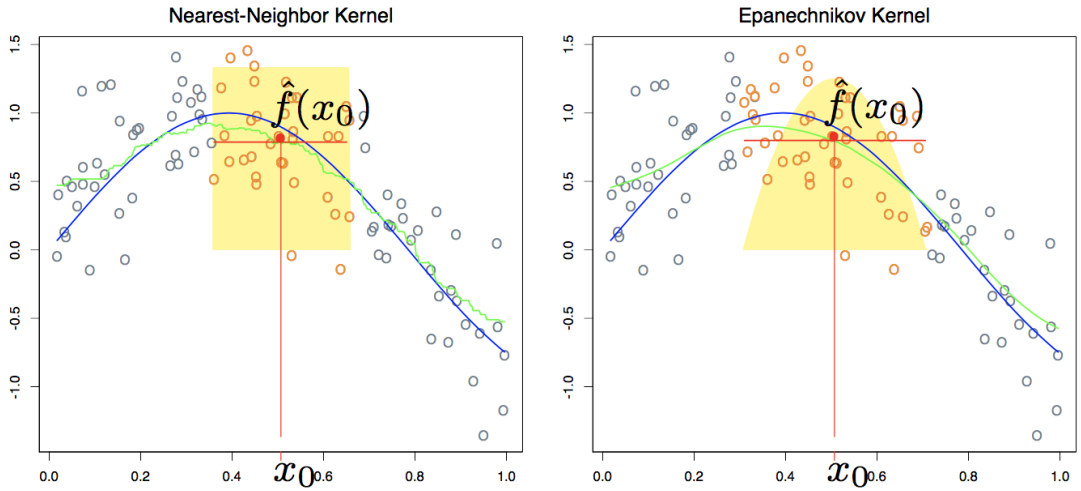

Figure: Nearest Neighbor Kernel vs Epanechnikov Kernel

Kernel Smoothing

One Dimensional Kernel Smoothers

- The smoothing parameter λ needs to be estimated. This doesn't need to be a constant, it can adapt with the neighborhood.
- With constant λ boundary issues arise due to the scarcity of observations.
- Other popular Kernels are the Tricube and Gaussian.
- Cross validation can be use to select the λ.
- The main issue with the NW estimator is that it can be badly biased on the boundaries of the domain. Local linear regression makes a first order correction for this which makes for a big difference.
- Local regression can be extended to local polynomial regression which corrects the bas to the same order of the degree of the polynomial. For example, local quadratic fits tend to help reduce the bias due to the curvature in the interior of the domain of the function.

Kernel Smoothing

One Dimensional Kernel Smoothers

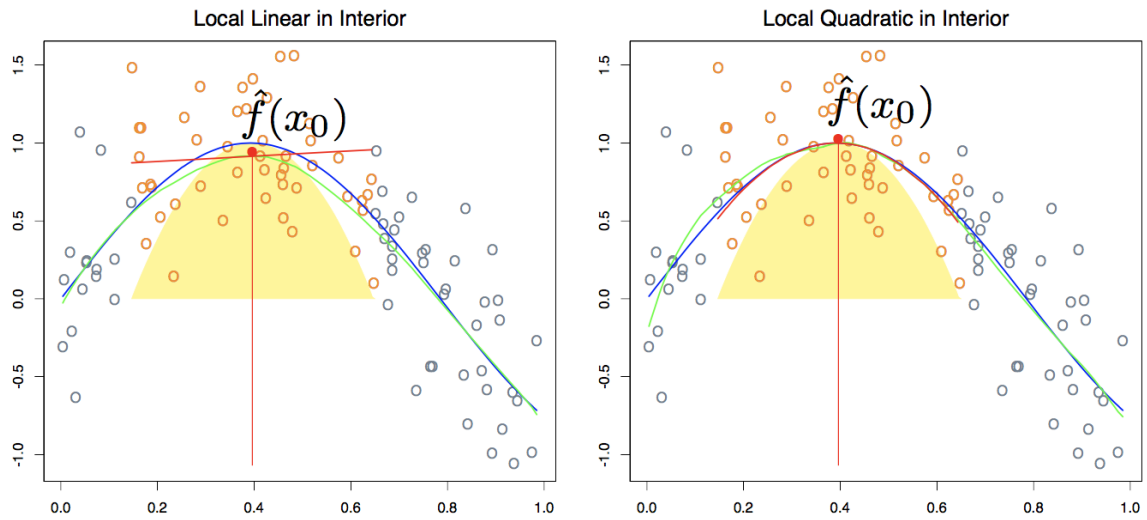

Figure: ~~Nearest Neighbor Kernel vs Epanechnikov Kernel~~
local linear in interior vs local quadratic in interior

Kernel Smoothing

One Dimensional Kernel Smoothers

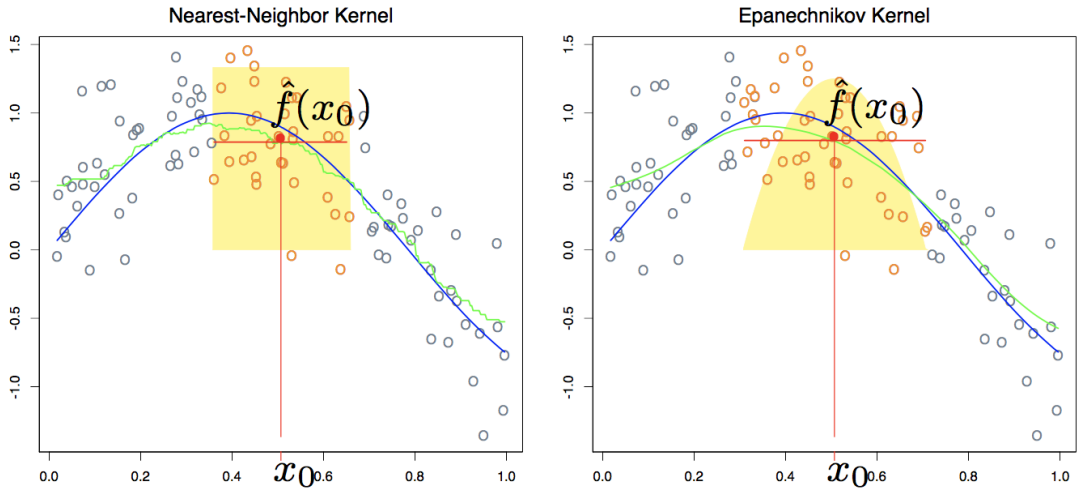

Figure: Nearest Neighbor Kernel vs Epanechnikov Kernel

Support Vector Machines

What are Support Vector Machines?

- Support Vector Machines are powerful, "black box" classifiers that generalize the concept of geometric separation of classes.
- Here we will distinguish between *maximal margin classifiers* (MMCs), *support vector classifiers* (SVCs) and *support vector machines* (SVMs).

Separating Hyperplanes

- SVMs are built based on the concept of hyperplanes. In a p-dimensional space, a hyperplane is a flat (affine) subspace of dimensions $p-1$. For example, in two dimensions, a separating hyperplane is a straight line.
- Formally it is defined as

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

Maximal Margin Classifier

Classification & Separating Hyperplanes

- Hyperplanes are a very natural way to classify observations. In a simplified ideal scenario we can imagine that it could be possible to find a hyperplane that separates the training observation perfectly into two classes. This is called a *separating hyperplane*.
- If a separating hyperplane exists we can use it to assign test observations to a class depending on which side of the hyperplane they fall on. That is we classify x^* depending on the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + \cdots + \beta_p x_p^*$.
- We can also use the distance from the separating hyperplane as a measure of confidence about the class assignment; the farther from the hyperplane, the higher the confidence in the test observation class assignment.

Maximal Margin Classifier

Classification & Separating Hyperplanes

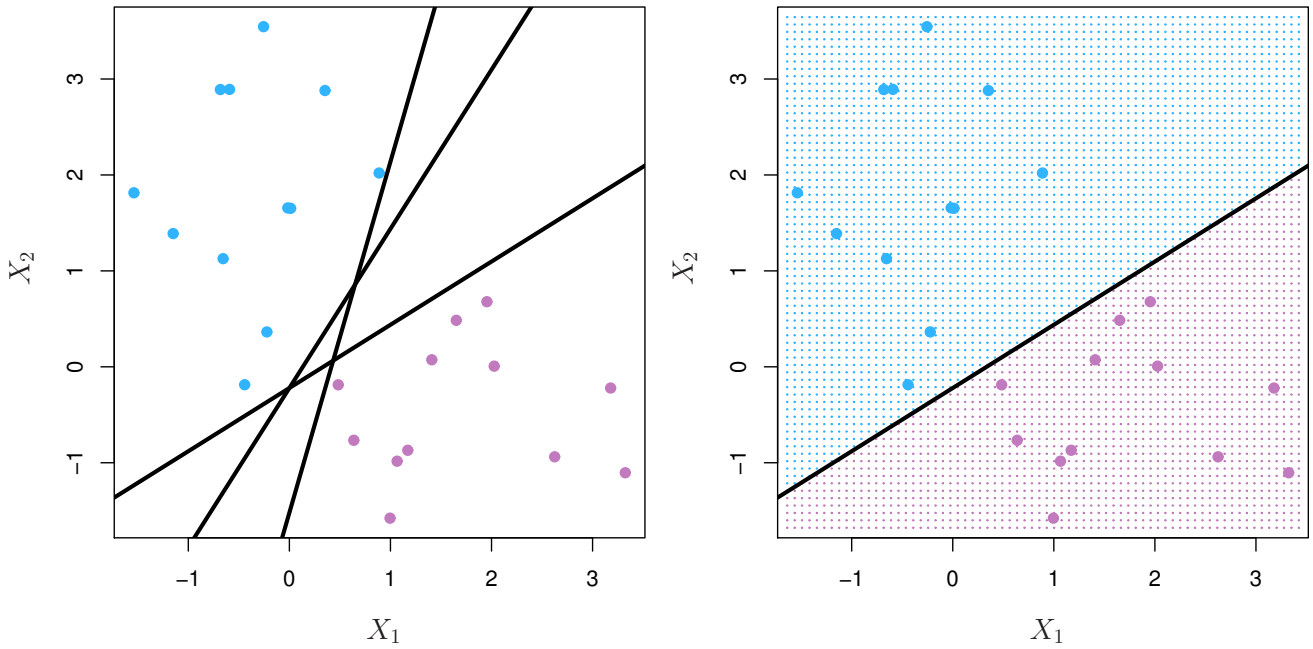

Figure: Separating hyperplanes and associated decision rules

Maximal Margin Classifier

The Maximal Margin Classifier

- If the data can be perfectly separated by a hyperplane then there will actually exist infinite such hyperplanes. How can we choose (the best) one?
- A natural choice is the one that is the farthest (in perpendicular distance) from the training observation and is called the Maximal Margin Hyperplane (MMH). The corresponding classifier is the MMC.
- The *margin* is the smallest distance from the hyperplane to the training observations.
- The observations whose distance from the maximal margin hyperplane is equal to the margin are known as the *support vectors*.
- Somewhat counterintuitively, the maximal margin hyperplane depends directly on the support vectors but not on the other observations.

Maximal Margin Classifier

The Maximal Margin Classifier

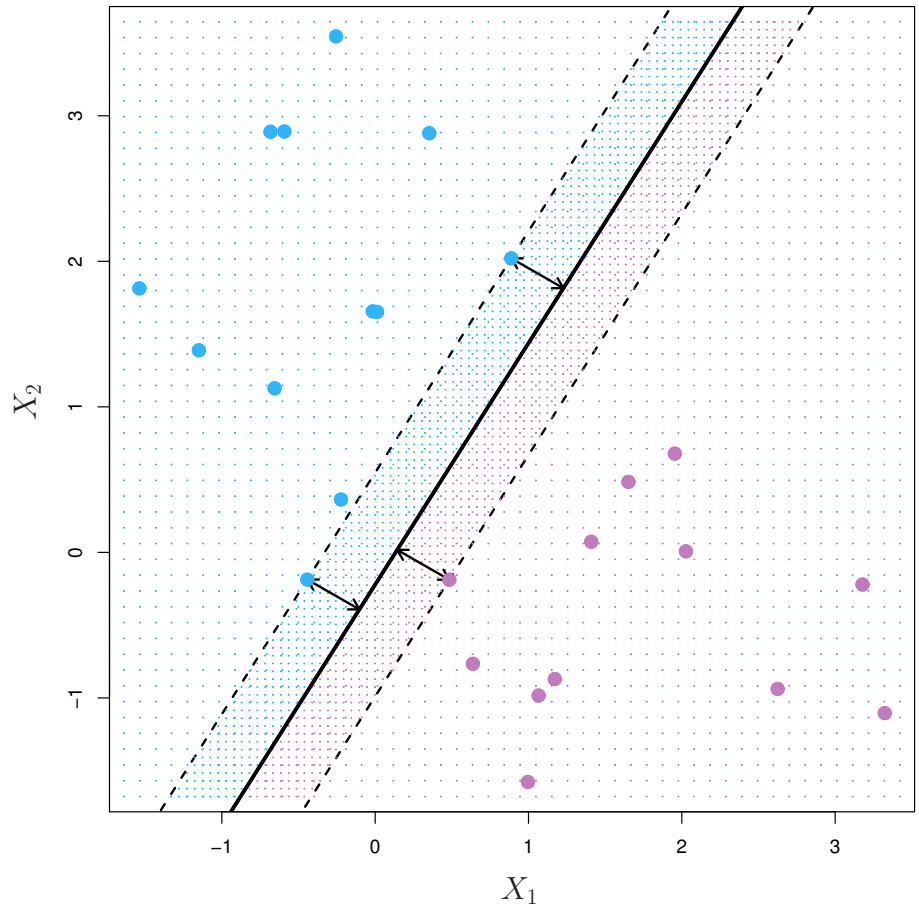

Figure: Maximum Margin Hyperplane, MMC, Margin and Support Vectors

Maximal Margin Classifier

Mathematical Construction

- In order to derive the MMH we formulate the problem as follows:

$$\max_{\beta_0,\ldots,\beta_p} M$$

$$s.t. \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i \left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right) \geq M \; \forall \; i = 1, \ldots, n.$$

- Here y_i is the class label (either 1 or -1) for the i^{th} observation in the training data.
- These constraints ensure that all observations are on the correct side of the hyperplane and at least a distance M from it. In other words, M is the margin of the hyperplane.

Support Vector Classifiers

MMC Issues and Soft Margins

- The MMC is a very intuitive way to perform classification, assuming that a separating hyperplane exists. But what if there is no separating hyperplane? In this case the optimization problem above has no solution and no MMC exists.
- The solution is to create a hyperplane that *almost* separates the classes, using a so-called *soft margin*. This gives us the SVC.
- This is also useful for another reason; the MMC can be very sensitive to individual observations. This does not only mean that it can easily overfit the data but also that distance from the MMH as a measure of classification confidence is also problematic due to instability.
- The answer of course is to consider a hyperplane that does not perfectly fit the data.

Support Vector Classifiers

MMC Issues and Soft Margins

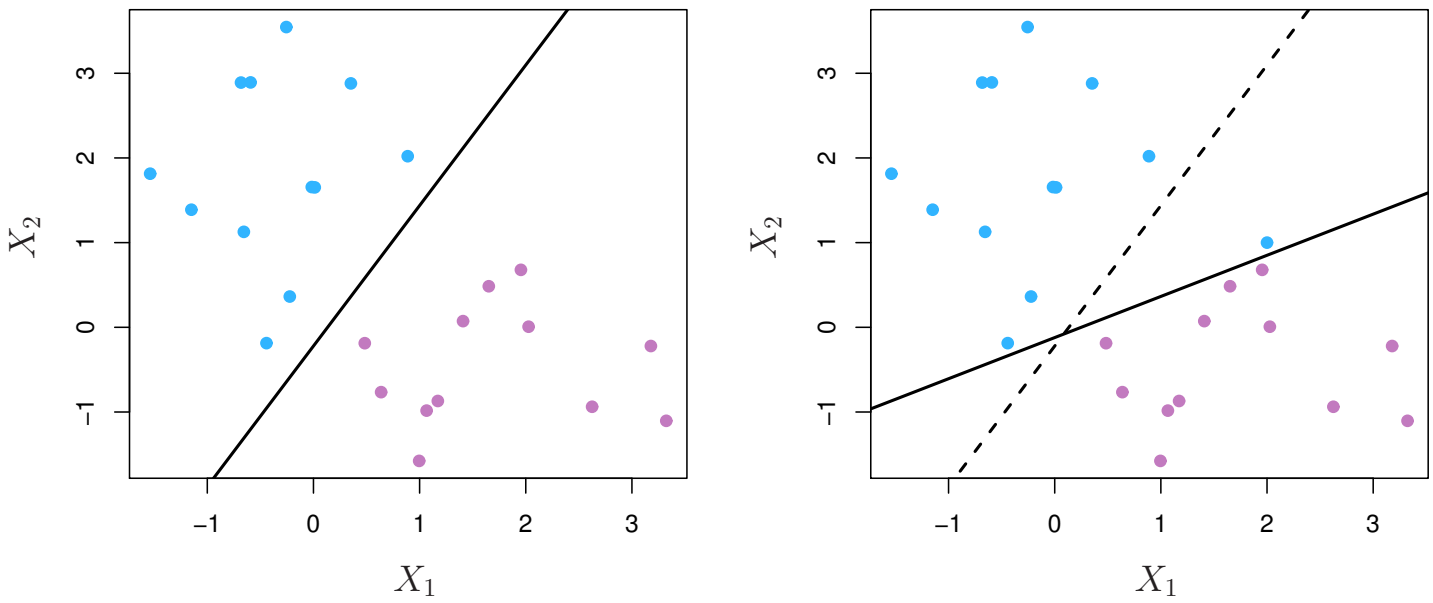

Figure: The change in MMC when adding a single new observation.

Support Vector Classifiers

The Details

- The SVC (instead of seeking the largest possible margin subject to all observations being on the correct side of the margin as well as the correct side of the hyperplane instead) allows some training observations to fall on the wrong side of the margin and even on the wrong side of the hyperplane.
- Formally this is expressed as

$$\max_{\beta_0,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n} M$$

$$s.t. \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i \left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}\right) \geq M(1 - \epsilon_i)$$

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

where C is a tuning parameter.

Support Vector Classifiers

The Details

- The ϵ_is are slack variables that allow individual observations to be on the wrong side of the margin or the hyperplane. They tell us where the ith observation is located with respect to the margin M.
- Specifically, if $\epsilon_i > 0$ then observation i is on the wrong side of the margin while if $\epsilon_i > 1$ it is on the wrong side of the hyperplane.
- The tuning parameter C determines the number and severity of the sum of total violations that will be tolerated. For positive C no more than C observations can be on the wrong side of the hyperplane.
- As you might expect, C is set via cross-validation. As C increase, the classifier has lower variance but potentially higher bias (and vice versa).
- Interestingly, in the optimization problem above, only observations that are either on or that violate the margin affect the hyperplane. Such observations are known as *support vectors*. This means that the SVC is typically robust to the behavior of observations that are far away from the hyperplane.

Support Vector Classifiers

The Details

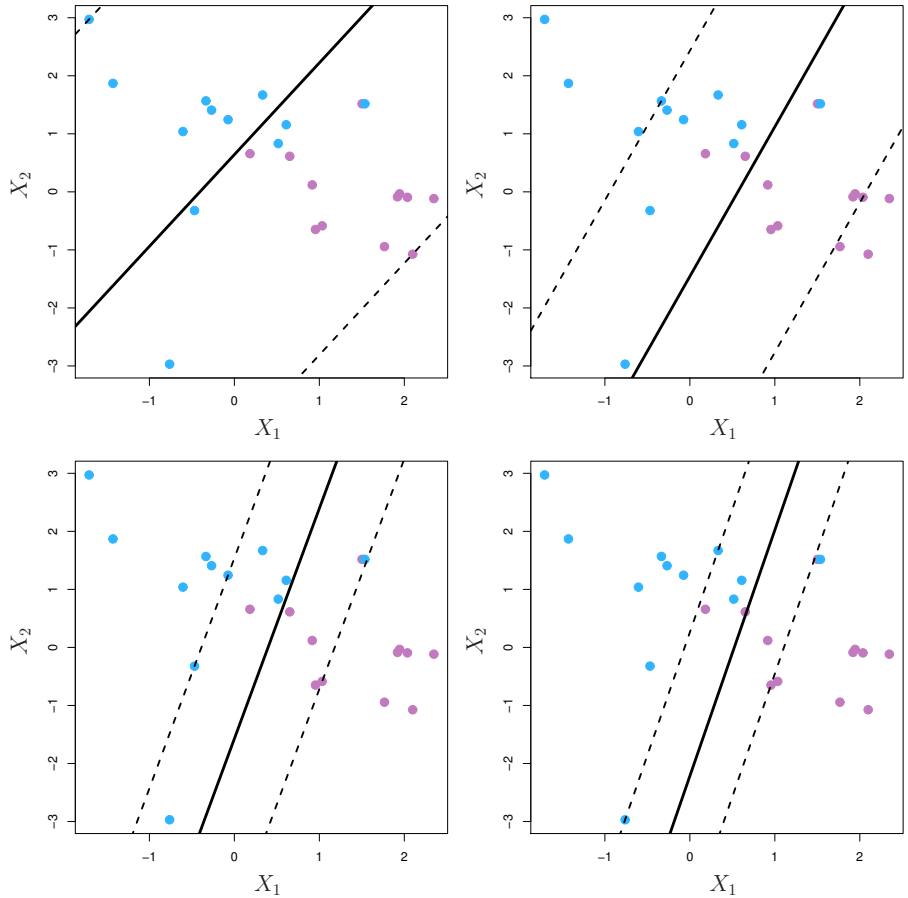

Figure: An SVC with four different values for the tuning parameter C. WIth higher C, the margin is larger, the support vectors are more and the variance is lower.

Support Vector Classifiers

Inner Products

- Both MMC and SVC are natural approaches to classification in the case that the decision boundary is linear. In practice, however, we will commonly deal with non-linear class boundaries.
- Adding higher order polynomial features is an approach that quickly becomes computationally unmanageable. Instead we use the so-called Kernel approach.
- It can be shown that the linear support classifier can be written as

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

what is alpha_i?

where the alphas are parameters that have non-zero values only for the support vectors.

- The intuition behind the Kernel approach is to replace this inner product with a more general, non-linear form. When a SVC is paired up with a non-linear Kernel the result is the so-called SVM.

Support Vector Classifiers

Kernels

- A common choice is the *polynomial kernal of order d*:

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

- Another popular choice is the *radial kernel*:

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

- In these cases the (non-linear) classifier has the form

$$f\left(x\right) = \beta_0 + \sum_{i=1}^{n} \alpha_i K\left(x, x_i\right)$$

- The parameters d and γ (as well as the cost parameter C) are calculated using cross validation. Lets look at an example.

Support Vector Classifiers

Kernels

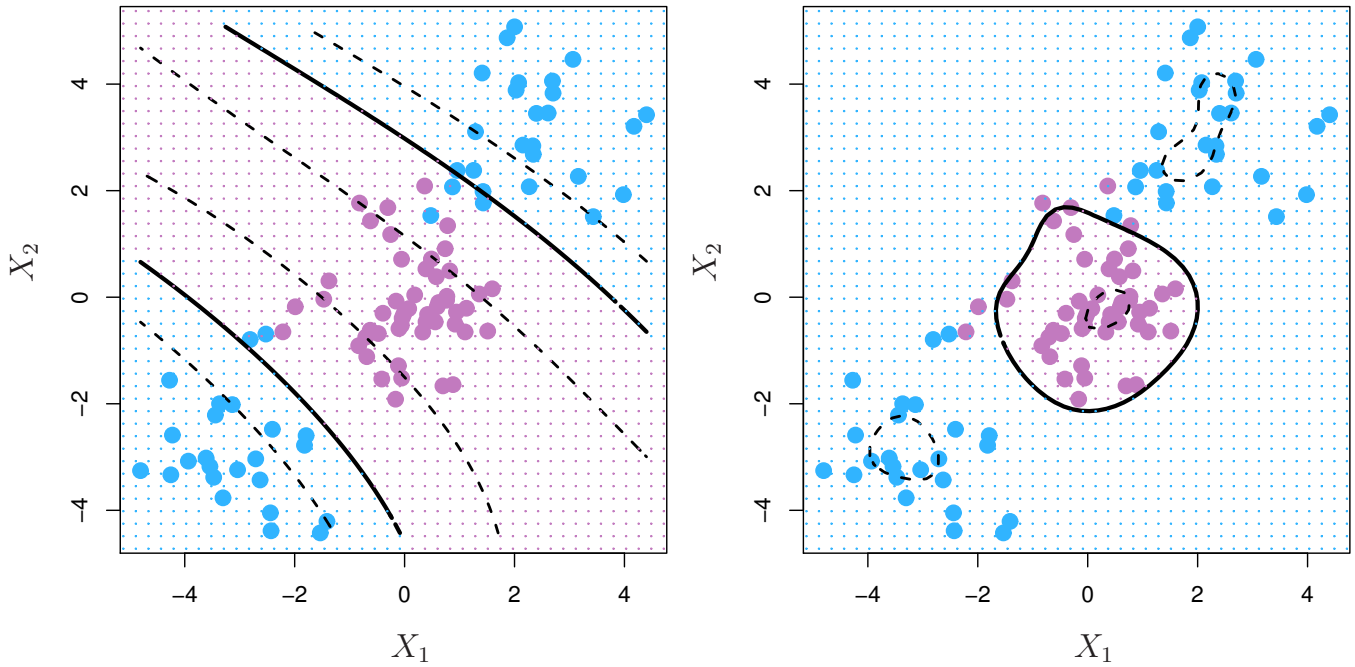

Figure: A polynomial of order 3 (left) and a radial kernel are applied to non-linear data.

Support Vector Classifiers

Discussion

- SVMs naturally lend themselves to two-class classification problems. If we wish to extend the SVM to a multi-class problem we can follow either of two approaches:
- $1-vs.-1$ Classification: we construct $\binom{K}{2}$ SVMs, each of which compares a pair of classes. Then we add the number of times the test observation is assigned to each class and assign it to the one that is the most frequent.
- $1-vs.-all$ Classification: We fit K SVMs, each time comparing one of the classes to the total of the remaining classes and we assign the observation to the class for which the test observation is farthest from the decision boundary.
- An interesting feature of SVMs is that they are mathematically very close to Logistic Regression (LR). Compared to LR however they perform better when the classes are well separated.

Support Vector Classifiers

Discussion

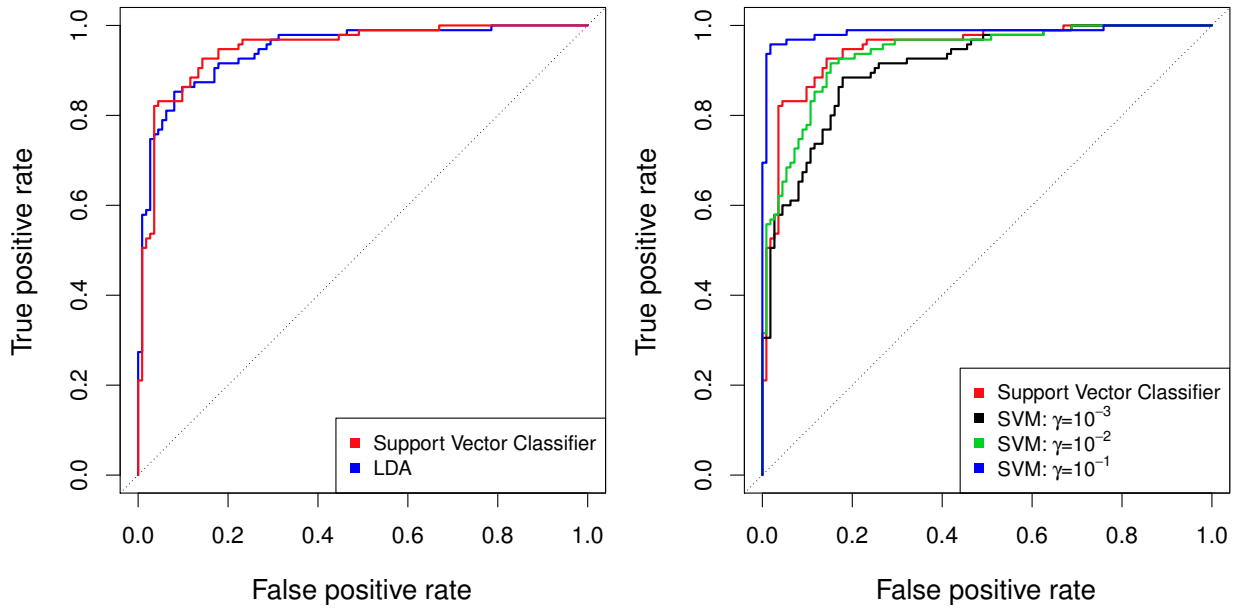

Figure: ROC Curves for the Heart training data.

Support Vector Classifiers

Discussion

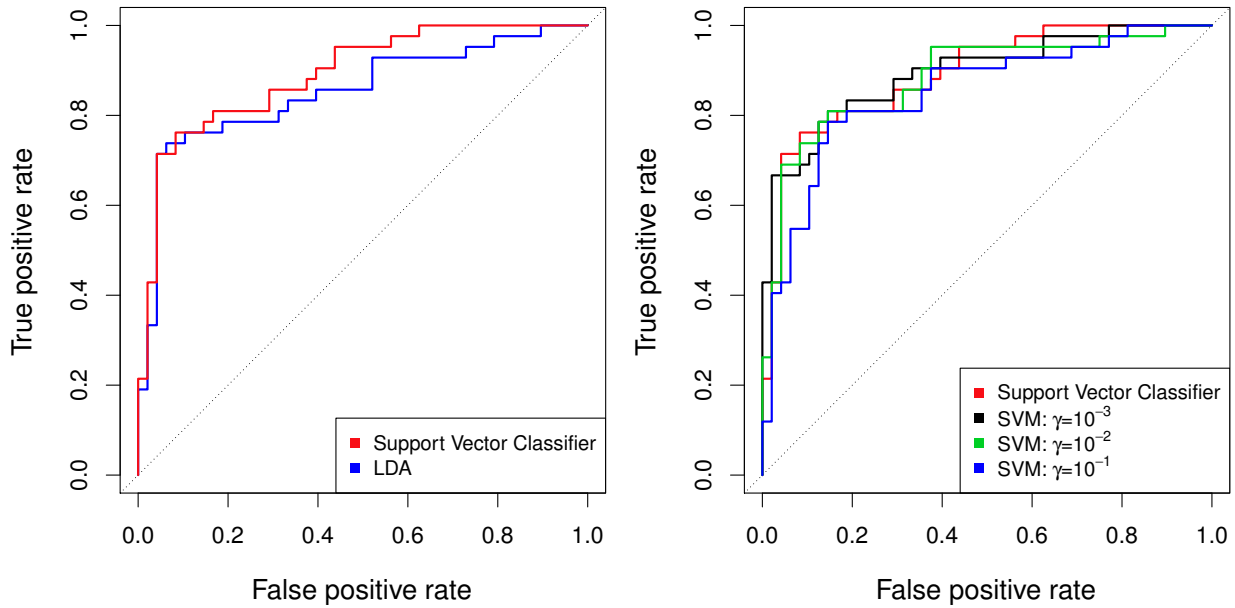

Figure: ROC Curves for the Heart test data.

Acknowledgements

- This presentation is based on and follow closely the excellent books (i) "Introduction to Statistical Learning with applications in R" by G. James, D. Witten, T. Hastie and R. Tibshirani and (ii) "The Elements of Statistical Learning" by T. Hastie, R. Tibshirani and J. Friedman.
- Also, some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani as well as from from "The Elements of Statistical Learning" (Springer, 2016) with permission from the authors: T. Hastie, R. Tibshirani and J. Friedman.