Advanced Topics in Statistics

Dorka Fekete, Gavin Shaddick

D.Fekete@exeter.ac.uk, G.Shaddick@exeter.ac.uk

UNIVERSITY OF EXETER

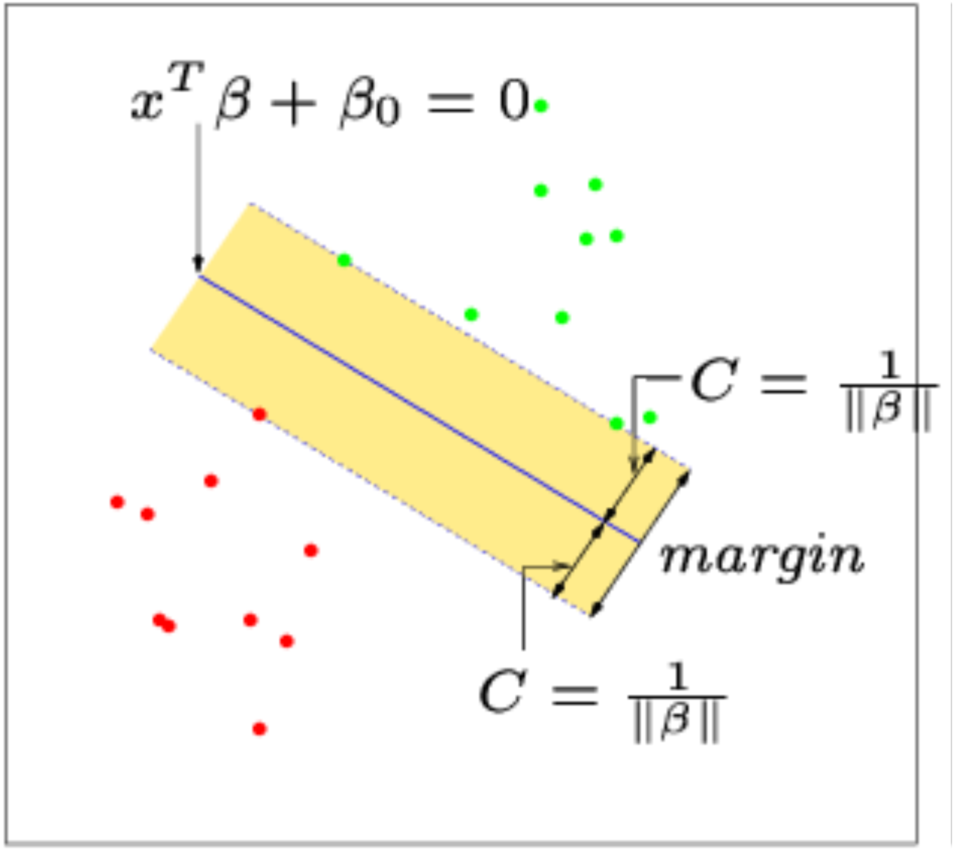
Support Vector Machines

SUPPORT VECTOR CLASSIFIER: SEPARABLE HYPERPLANES

- Imagine a situation where you have a two classes 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.

ITS EASIEST TO SEE WITH A PICTURE

- C is the minimum perpendicular distance between each point and the separating line.
- We find the line which maximises C.
- This line is called the "optimal separating hyperplane"
- The classification of a point depends on which side of the line it falls on.



MORE THAN TWO PREDICTORS

- This idea works just as well with more than two predictors.
- For example, with three predictors you want to find the plane that produces the largest separation between the classes.
- With more than three dimensions it becomes hard to visualise a plane but it still exists. In general they are called hyper-planes.

NON-SEPARATING CLASSES

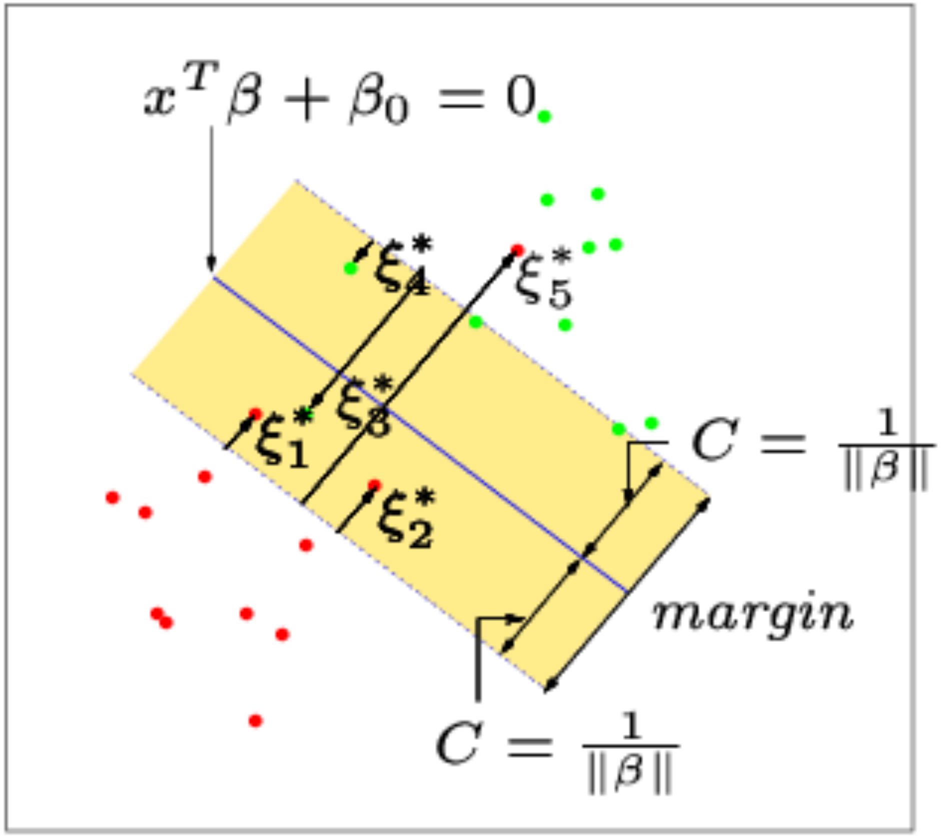
- Of course in practice it is not usually possible to find a hyper-plane that perfectly separates two classes.
- In other words, for any straight line or plane that I draw there will always be at least some points on the wrong side of the line.
- In this situation we try to find the plane that gives the best separation between the points that are correctly classified subject to the points on the wrong side of the line not being off by too much.
- It is easier to see with a picture!

NON-SEPARATING EXAMPLE

- Let ξ^*_i represent the amount that the ith point is on the wrong side of the margin (the dashed line).
- Then we want to maximise C subject to

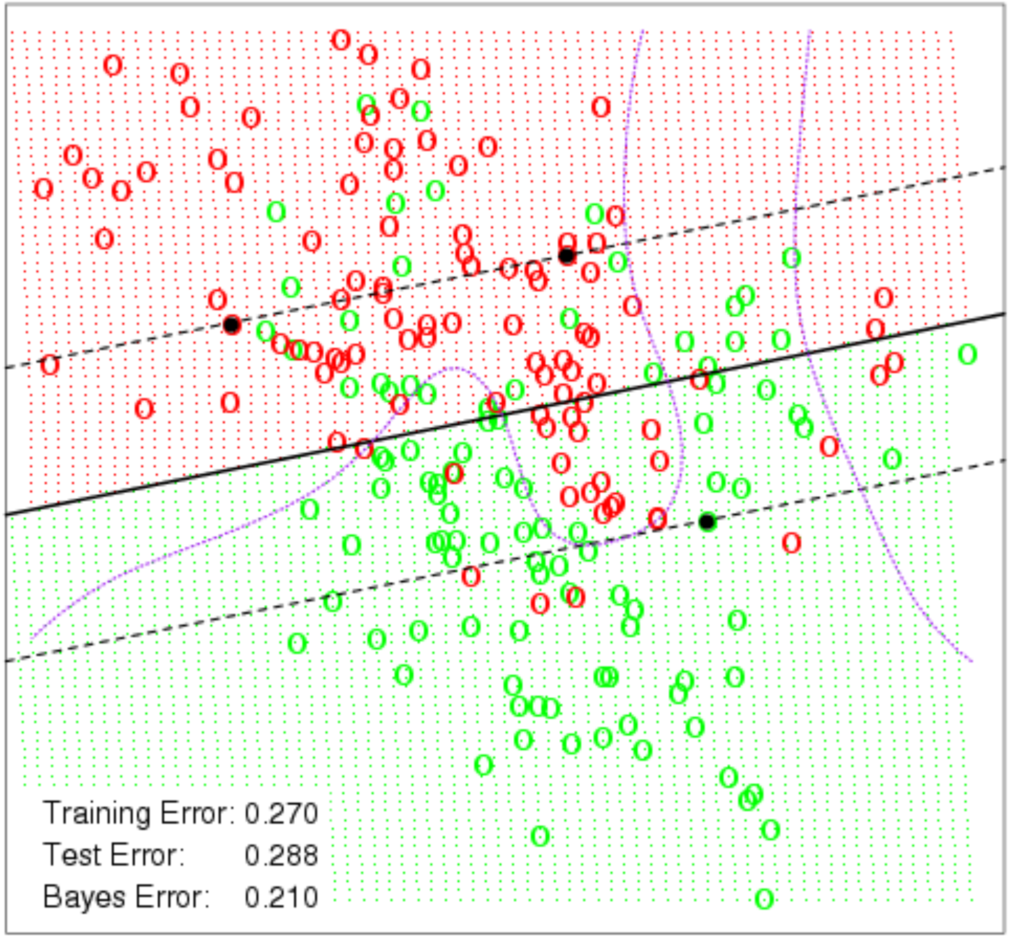
$$\frac{1}{C}\sum_{i=1}^{n}\xi_i^* \leq \text{Constant}$$

- The constant is a tuning parameter that we choose.



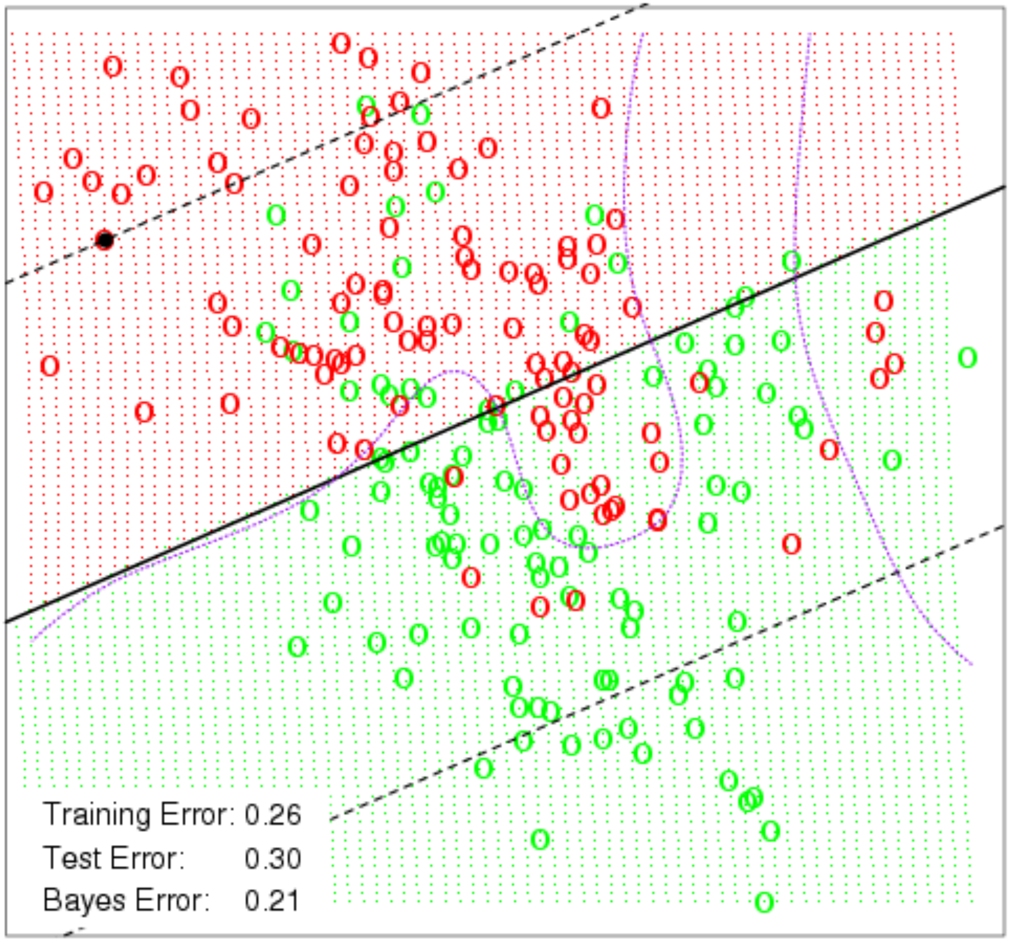
A SIMULATION EXAMPLE WITH A SMALL CONSTANT

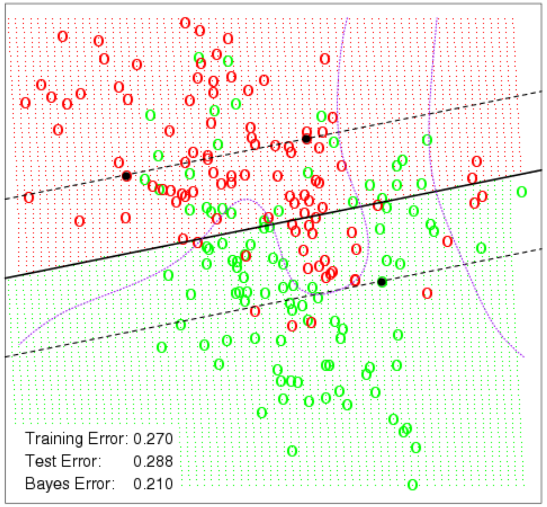
- This is the previously seen simulation example.
- The distance between the dashed lines represents the margin or 2C.
- The purple lines represent the Bayes decision boundaries



THE SAME EXAMPLE WITH A LARGER CONSTANT

- Using a larger constant allows for a greater margin and creates a slightly different classifier.
- Notice, however, that the decision boundary must always be linear.





SUPPORT VECTOR MACHINE CLASSIFIER: NON-LINEAR CLASSIFIER

- The support vector classifier is fairly easy to think about. However, because it only allows for a linear decision boundary it may not be all that powerful.
- Recall that we can extend linear regression to non-linear regression using a basis function (as in GAMs) i.e.

$$Y_i = \beta_0 + \beta_1 b_1(X_i) + \beta_2 b_2(X_i) + \cdots + \beta_p b_p(X_i) + \varepsilon_i$$

A BASIS APPROACH

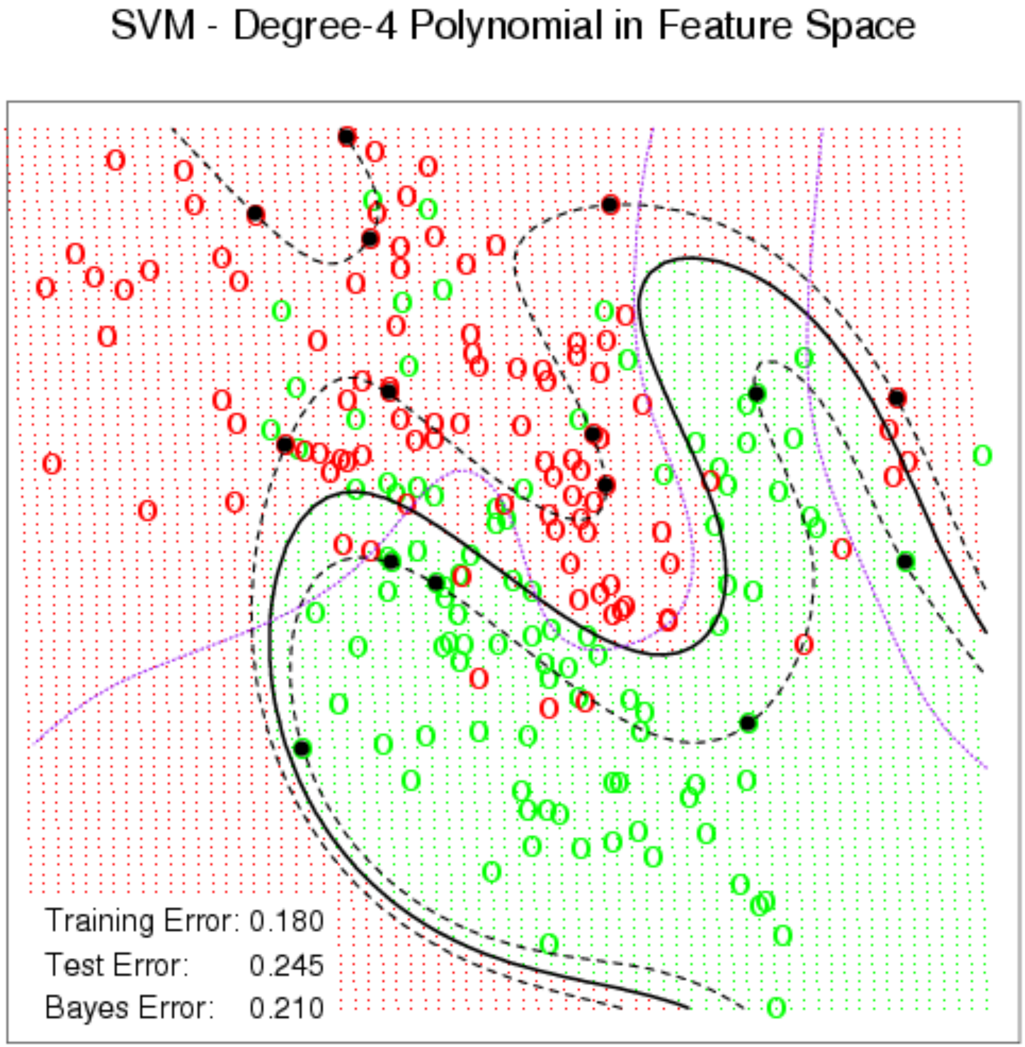
- Conceptually, we can take a similar approach with the support vector classifier.
- The support vector classifier finds the optimal hyper-plane in the space spanned by X_1, X_2,…, X_p.
- Instead we can create transformations (or a basis) $b_1(x)$, $b_2(x)$, …, $b_M(x)$ and find the optimal hyper-plane in the space spanned by $b_1(\mathbf{X})$, $b_2(\mathbf{X})$, …, $b_M(\mathbf{X})$.
- This approach produces a linear plane in the transformed space but a non-linear decision boundary in the original space.
- This is called the Support Vector Machine Classifier.

IN REALITY

- While conceptually the basis approach is how the support vector machine works, there is some complicated math (which I will spare you) which means that we don't actually choose $b_1(x)$, $b_2(x)$, …, $b_M(x)$.
- Instead we choose something called a Kernel function which takes the place of the basis.
- Common kernel functions include
  - Linear
  - Polynomial
  - Radial Basis
  - Sigmoid

POLYNOMIAL KERNEL ON SIM DATA

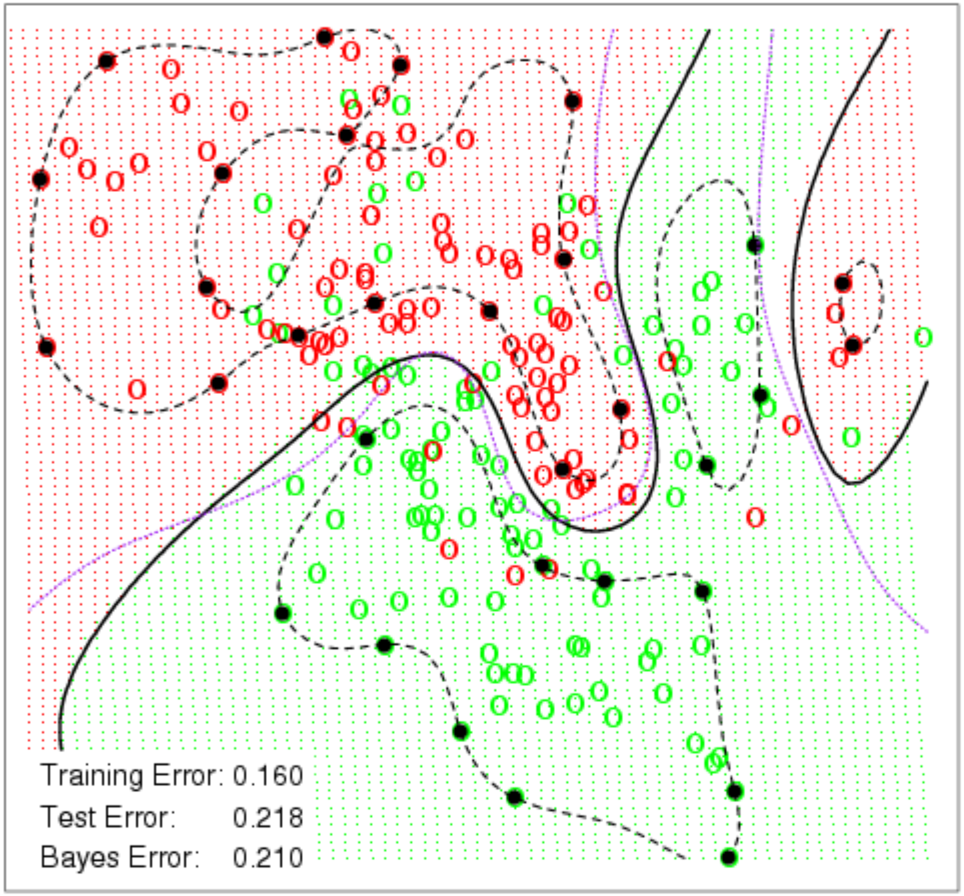
- Using a polynomial kernel we now allow SVM to produce a non-linear decision boundary.
- Notice that the test error rate is a lot lower.



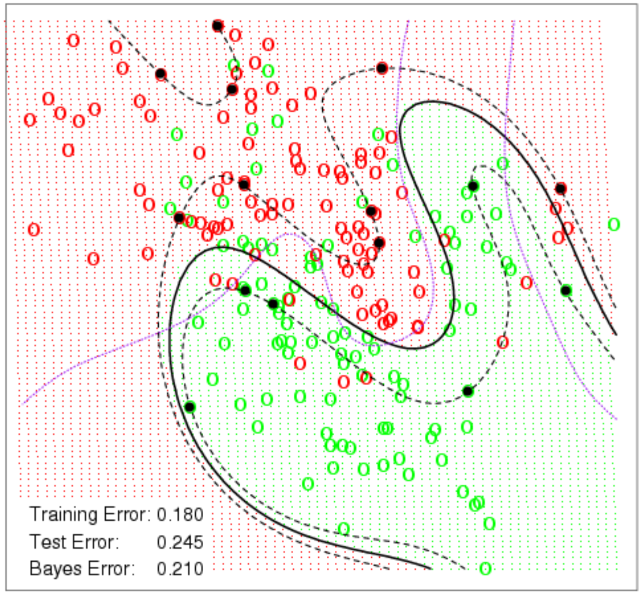
RADIAL BASIS KERNEL

- Using a Radial Basis Kernel you get an even lower error rate.

SVM - Radial Kernel in Feature Space

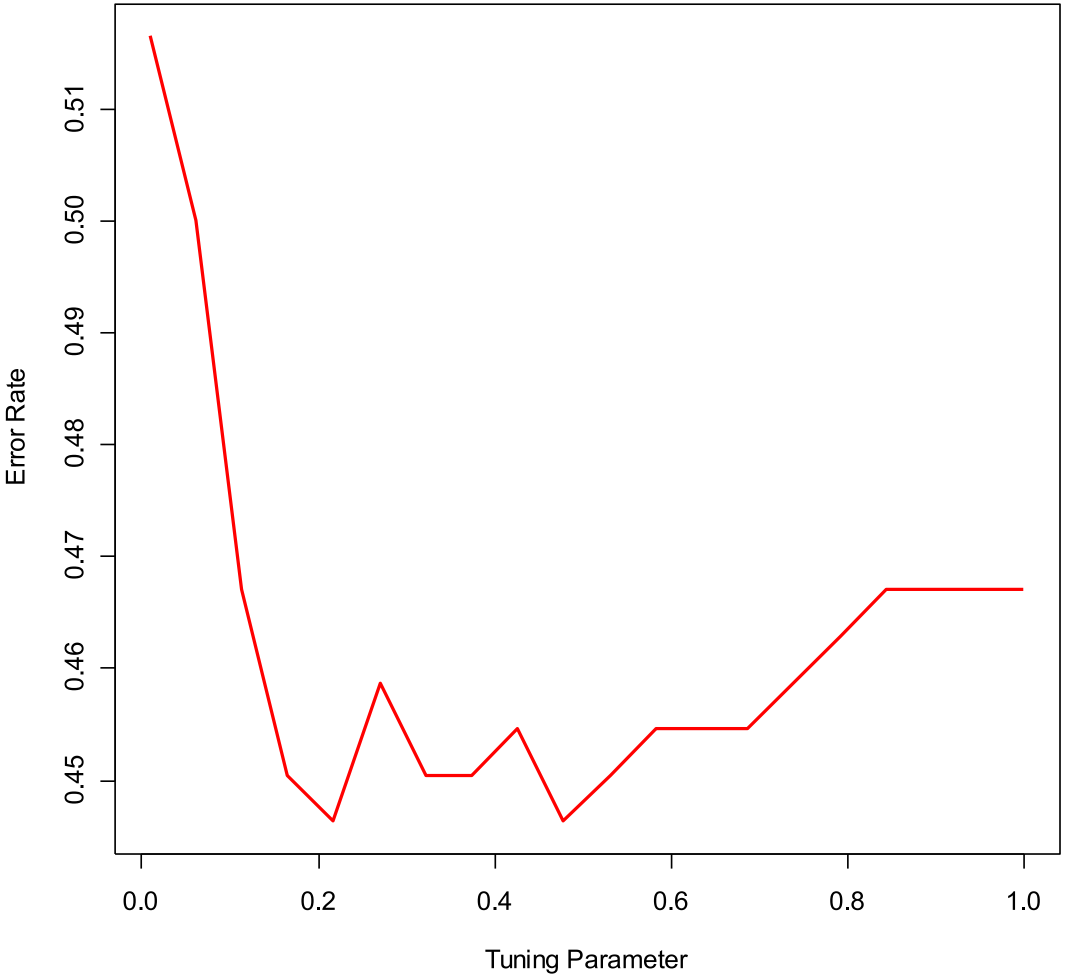


SVM - Degree-4 Polynomial in Feature Space



ERROR RATES ON S&P DATA

- Here I used a Radial Basis Kernel and calculated the error rate for different values of the tuning parameter.
- The results on this data were similar to GAM but not as good as Boosting.


Error Rate
Tuning Parameter