BITS Pilani presentation

BITS Pilani
Pilani Campus

Tanmay Tulsidas Verlekar
CSIS

BITS Pilani
Pilani Campus

Applied Machine Learning SE ZG568 / SS ZG568

Lecture No.7

Support Vector Machine (SVM)

SVM operates by identifying widest possible street (represented by the parallel dashed lines) between the classes. This is called large margin classification.

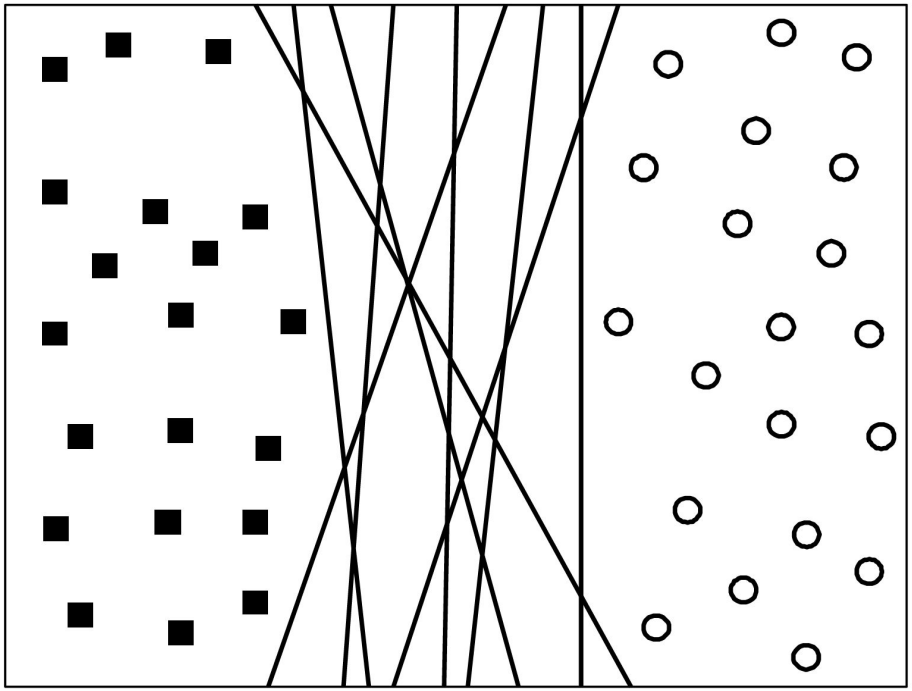


Figure 5.21. Possible decision boundaries for a linearly separable data set.

Rationale for Maximum Margin

Decision boundaries with large margins tend to have better generalization errors than those with small margins.

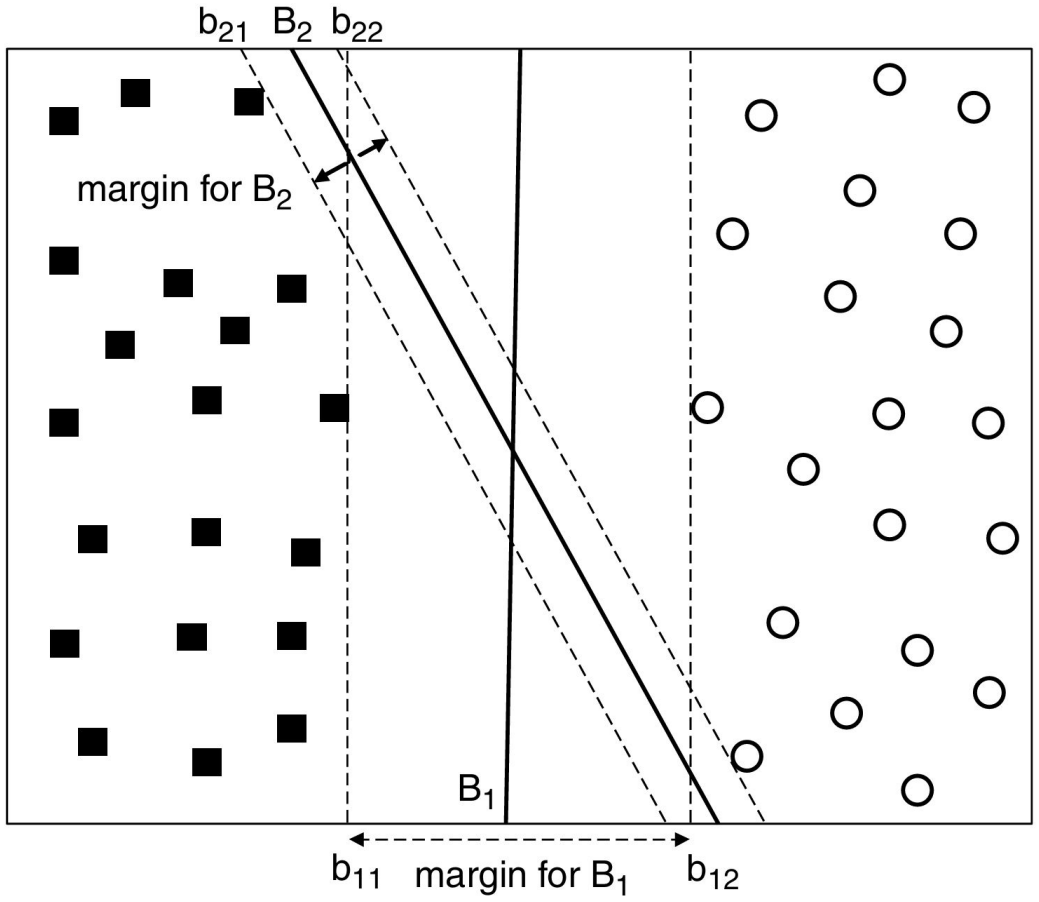


Figure 5.22. Margin of a decision boundary.

Structural risk minimization (SRM)

the generalization error of a classifier (R) in terms of its training error (R_e), the number of training examples (N), and the model complexity, otherwise known as its **capacity** (h). More specifically, with a probability of $1 - \eta$, the generalization error of the classifier can be at worst

$$R \leq R_e + \varphi\left(\frac{h}{N}, \frac{\log(\eta)}{N}\right), \tag{5.27}$$

Linear SVM: Separable Case

A decision boundary that bisects the training examples into their respective classes

$$\mathbf{w} \cdot \mathbf{x} + b = 0,$$

If xa and xb are two points located on the decision boundary, then

$$\mathbf{w} \cdot \mathbf{x}_a + b = 0,$$
$$\mathbf{w} \cdot \mathbf{x}_b + b = 0.$$

where xb − xa is a vector parallel to the decision boundary and is directed from xa to xb. Since the dot product is zero, the direction for w must be perpendicular to the decision boundary,

$$\mathbf{w} \cdot (\mathbf{x}_b - \mathbf{x}_a) = 0,$$

Illustration

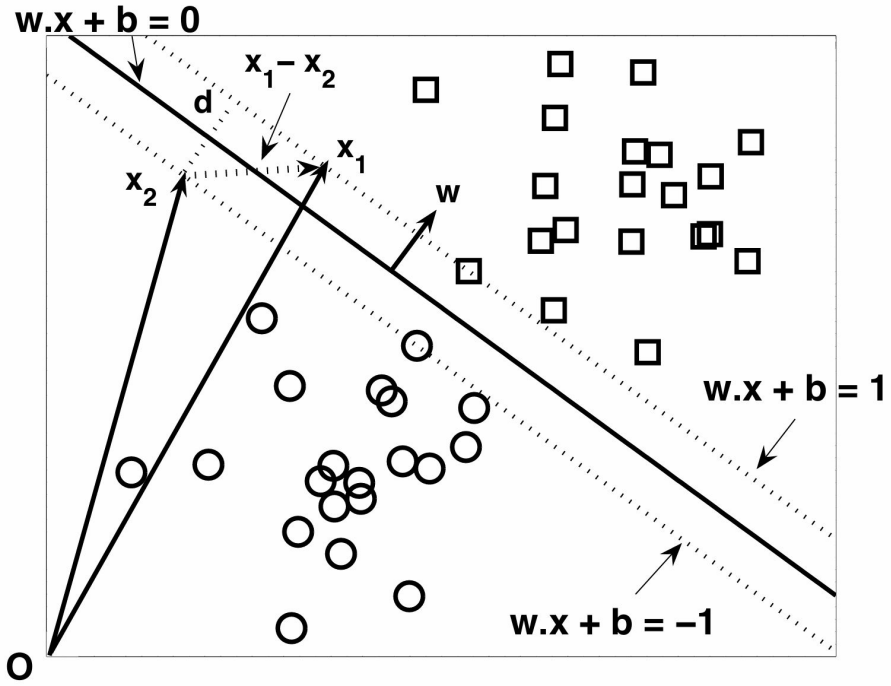


Figure 5.23. Decision boundary and margin of SVM.

For any square $\mathbf{x}_s$ located above the decision boundary, we can show that

$$\mathbf{w} \cdot \mathbf{x}_s + b = k, \tag{5.29}$$

where $k > 0$. Similarly, for any circle $\mathbf{x}_c$ located below the decision boundary, we can show that

$$\mathbf{w} \cdot \mathbf{x}_c + b = k', \tag{5.30}$$

where $k' < 0$.

Margin of a Linear Classifier

Rescaleing the parameters w and b of the decision boundary so that the two parallel hyperplanes bi1 and bi2 can be expressed as follows:

$$b_{i1} : \ \mathbf{w} \cdot \mathbf{x} + b = 1,$$

$$b_{i2} : \ \mathbf{w} \cdot \mathbf{x} + b = -1.$$

Margin d

The margin of the decision boundary is given by the distance between these two hyperplanes. To compute the margin, let $\mathbf{x}_1$ be a data point located on b_{i1} and $\mathbf{x}_2$ be a data point on b_{i2}, as shown in Figure 5.23. Upon substituting these points into Equations 5.32 and 5.33, the margin d can be computed by subtracting the second equation from the first equation:

$$\begin{aligned} \mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) &= 2 \\ \|\mathbf{w}\| \times d &= 2 \\ \therefore d &= \frac{2}{\|\mathbf{w}\|}. \end{aligned} \tag{5.34}$$

Learning a Linear SVM Model

The parameters must be chosen in such a way that the following two conditions are met

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 \text{ if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 \text{ if } y_i = -1.$$

SVM imposes an additional requirement that the margin of its decision boundary must be maximal.

$$f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}.$$

Linear SVM: Separable Case

The learning task in SVM can be formalized as the following constrained optimization problem:

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1, \quad i = 1, 2, \ldots, N.$$

Standard Lagrange multiplier method

Lagrangian for the optimization problem:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \lambda_i \left(y_i(\mathbf{w} \cdot \mathbf{x_i} + b) - 1 \right)$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Longrightarrow \mathbf{w} = \sum_{i=1}^{N} \lambda_i y_i \mathbf{x}_i,$$

$$\frac{\partial L_p}{\partial b} = 0 \Longrightarrow \sum_{i=1}^{N} \lambda_i y_i = 0.$$

$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}.$$

The decision boundary can be expressed as follows

$$\left(\sum_{i=1}^{N} \lambda_i y_i \mathbf{x_i} \cdot \mathbf{x} \right) + b = 0.$$

Linear SVM: Nonseparable Case

Although the decision boundary B1 misclassifies the new examples, while B2 classifies them correctly, this does not mean that B2 is a better decision boundary than B1 because the new examples may correspond to noise in the training data.

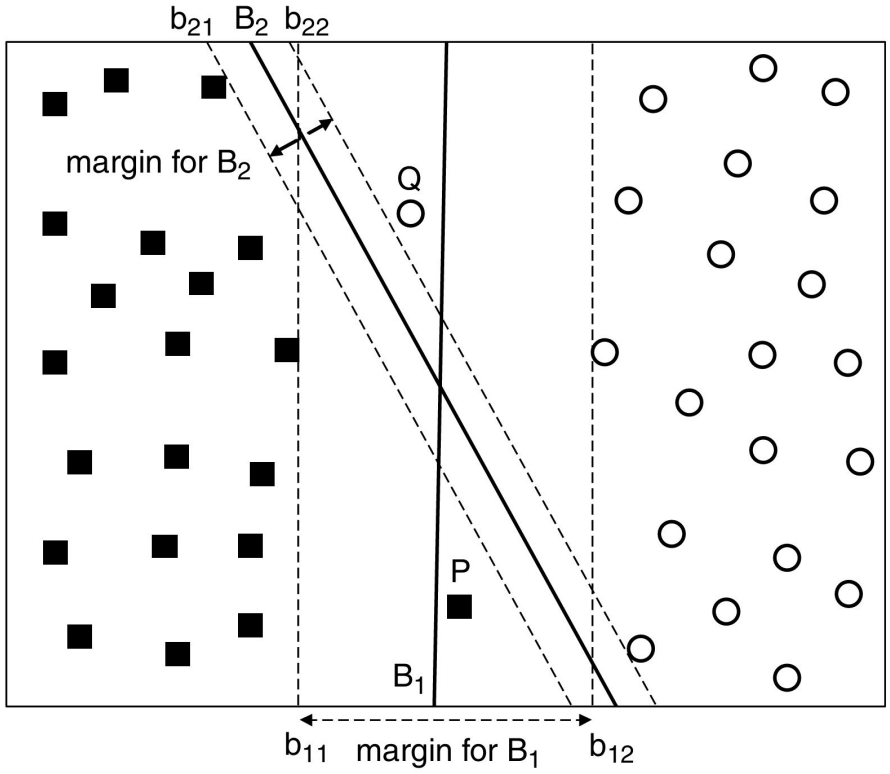


Figure 5.25. Decision boundary of SVM for the nonseparable case.

the soft margin approach

The inequality constraints must be relaxed to accommodate the non linearly separable data.

$$\mathbf{w} \cdot \mathbf{x_i} + b \geq 1 - \xi_i \quad \text{if } y_i = 1,$$
$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1 + \xi_i \quad \text{if } y_i = -1,$$

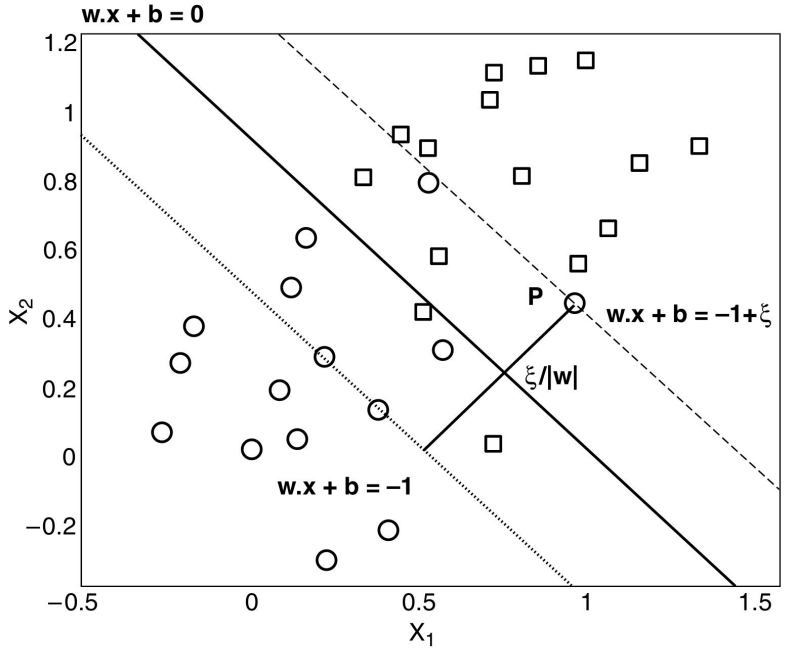


Figure 5.26. Slack variables for nonseparable data.

Constraining the slack

Since there are no constraints on the number of mistakes the decision boundary can make, the learning algorithm may find a decision boundary with a very wide margin but misclassifies many of the training examples

$$f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} + C(\sum_{i=1}^{N} \xi_i)^k,$$

It follows that the Lagrangian for this constrained optimization problem can be written as follows:

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \lambda_i\{y_i(\mathbf{w}\cdot\mathbf{x_i} + b) - 1 + \xi_i\} - \sum_{i=1}^{N} \mu_i\xi_i, \quad (5.46)$$

Lagrangian for this constrained optimization

Setting the first-order derivative of L with respect to $\mathbf{w}$, b, and ξ_i to zero would result in the following equations:

$$\frac{\partial L}{\partial w_j} = w_j - \sum_{i=1}^{N} \lambda_i y_i x_{ij} = 0 \implies w_j = \sum_{i=1}^{N} \lambda_i y_i x_{ij}. \tag{5.50}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \lambda_i y_i = 0 \implies \sum_{i=1}^{N} \lambda_i y_i = 0. \tag{5.51}$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \mu_i = 0 \implies \lambda_i + \mu_i = C. \tag{5.52}$$

dual Lagrangian

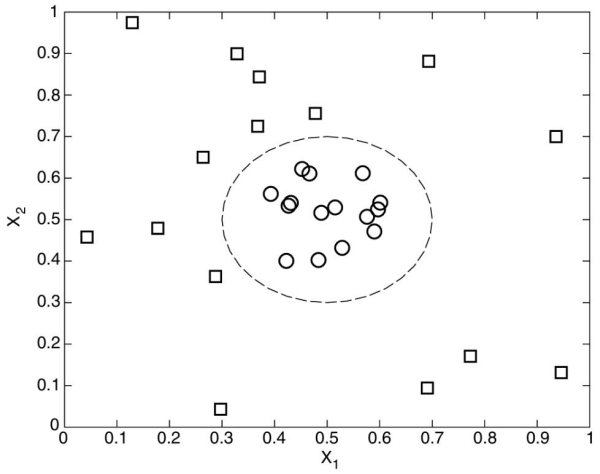
$$
\begin{aligned}
L_D &= \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j + C\sum_i \xi_i \\
&\quad - \sum_i \lambda_i \{y_i(\sum_j \lambda_j y_j \mathbf{x}_i \cdot \mathbf{x}_j + b) - 1 + \xi_i\} \\
&\quad - \sum_i (C - \lambda_i)\xi_i \\
&= \sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\lambda_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j},
\end{aligned}
$$

Nonlinear SVM

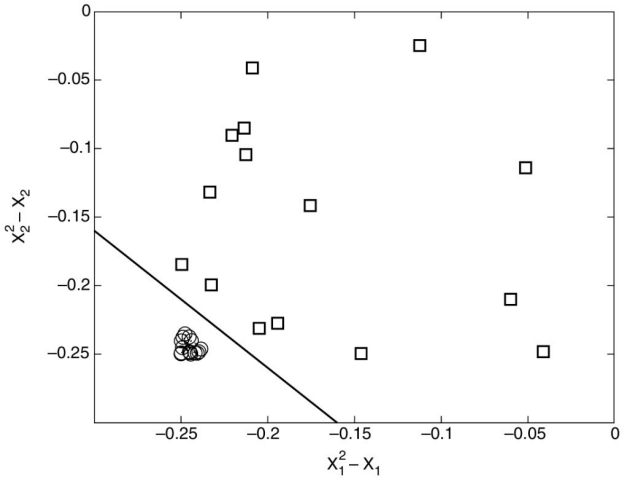
The trick here is to transform the data from its original coordinate space in x into a new space Φ(x) so that a linear decision boundary can be used to separate the instances in the transformed space.

After doing the transformation, we can apply the methodology presented in the previous sections to find a linear decision boundary in the transformed space.

Attribute Transformation



(a) Decision boundary in the original two-dimensional space.



(b) Decision boundary in the transformed space.

$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2, \\ -1 & \text{otherwise.} \end{cases}$$

$$w_4 x_1^2 + w_3 x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.$$

This approach is that it may suffer from the curse of dimensionality problem

Kernel Trick

The dot product is often regarded as a measure of similarity between two input vectors.

$$\begin{aligned}\Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) &= (u_1^2, u_2^2, \sqrt{2}u_1, \sqrt{2}u_2, 1) \cdot (v_1^2, v_2^2, \sqrt{2}v_1, \sqrt{2}v_2, 1) \\ &= u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 + 2u_2 v_2 + 1 \\ &= (\mathbf{u} \cdot \mathbf{v} + 1)^2.\end{aligned}$$

This analysis shows that the dot product in the transformed space can be expressed in terms of a similarity function in the original space:

$$K(\mathbf{u}, \mathbf{v}) = \Phi(\mathbf{u}) \cdot \Phi(\mathbf{v}) = (\mathbf{u} \cdot \mathbf{v} + 1)^2. \tag{5.61}$$

Decision boundary

$$
\begin{aligned}
f(\mathbf{z}) &= sign\left(\sum_{i=1}^{n} \lambda_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{z}) + b\right) \\
&= sign\left(\sum_{i=1}^{n} \lambda_i y_i K(\mathbf{x}_i, \mathbf{z}) + b\right) \\
&= sign\left(\sum_{i=1}^{n} \lambda_i y_i (\mathbf{x}_i \cdot \mathbf{z} + 1)^2 + b\right),
\end{aligned}
$$

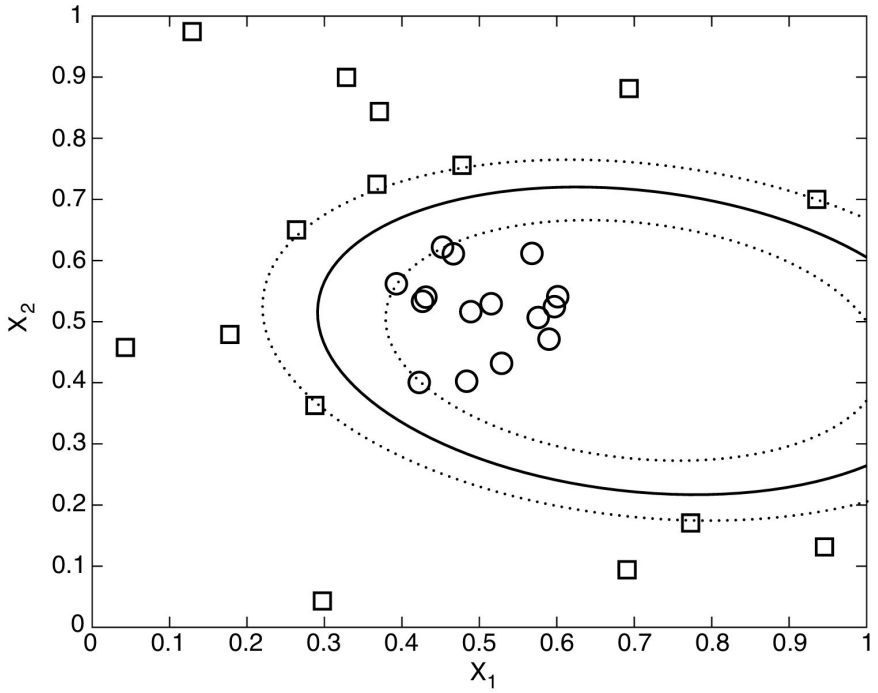


Figure 5.29. Decision boundary produced by a nonlinear SVM with polynomial kernel.

positive definite kernel functions

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/(2\sigma^2)}$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh(k\mathbf{x} \cdot \mathbf{y} - \delta)$$