Lecture 22: Support vector machines

Reading: Sections 9.3, 9.4

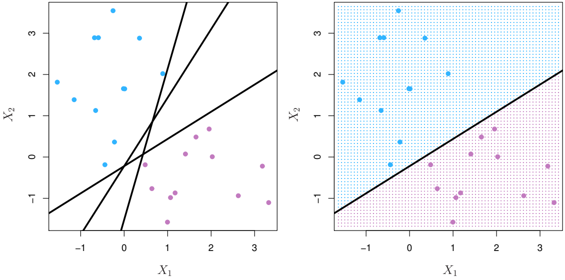
STATS 202: Data mining and analysis

Sergio Bacallado

November 13, 2013

Maximal margin classifier

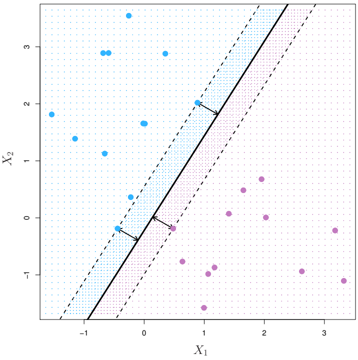
- Suppose we have a classification problem with response $Y = -1$ or $Y = 1$.
- If the classes can be separated, most likely, there will be an infinite number of hyperplanes separating the classes.



Maximal margin classifier

Idea:

- Draw the largest possible empty margin around the hyperplane.
- Out of all possible hyperplanes that separates the 2 classes, choose the one with the widest margin.

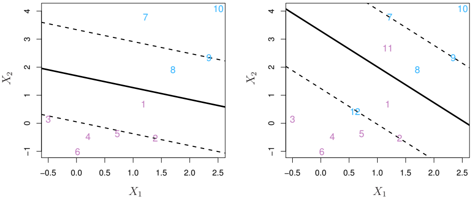


Support vector classifier

Problem: It is not always possible to separate the points using a hyperplane.

Support vector classifier:

- Relaxation of the maximal margin classifier.
- Allows a number of points points to be on the wrong side of the margin or the margin or even the hyperplane.



Support vector classifier

This can be written as an optimization problem:

$$\max_{\beta_0,\beta,\epsilon} \quad M$$

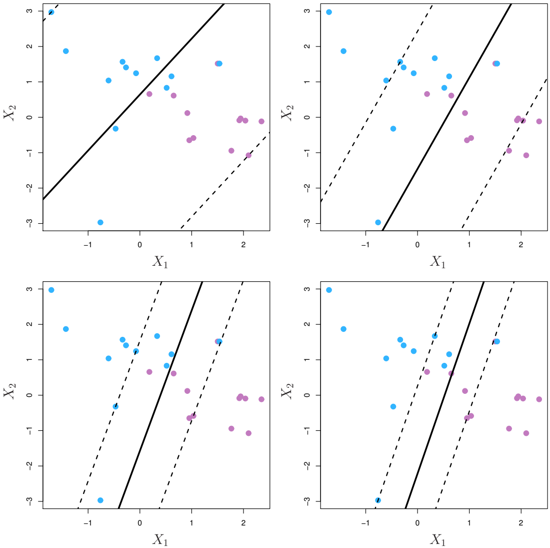
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$\underbrace{y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})}_{\text{How far is } x_i \text{ from the hyperplane}} \geq M(1 - \epsilon_i) \quad \text{for all } i = 1, \ldots, n$$

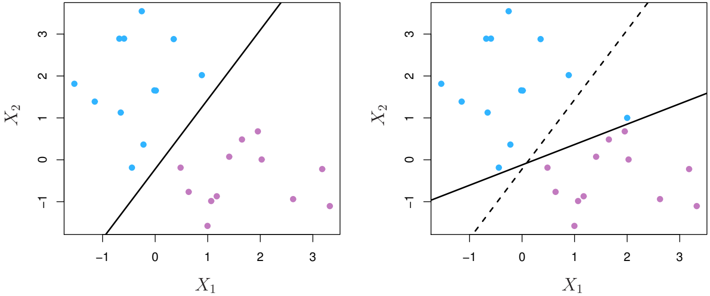
$$\epsilon_i \geq 0 \text{ for all } i = 1, \ldots, n, \quad \sum_{i=1}^{n} \epsilon_i \leq C.$$

M is the width of the margin in either direction.
$\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ are called *slack* variables.
C is called the *budget*.

Tuning the budget, C (high to low)



If the budget is too low, we tend to overfit



Maximal margin classifier, $C = 0$. Adding one observation dramatically changes the classifier.

Finding the support vector classifier

We can reformulate the problem by defining a vector $w = (w_1, \ldots, w_p) = \beta/M$:

$$\min_{\beta_0, w, \epsilon} \quad \frac{1}{2}\|w\|^2 + D\sum_{i=1}^{n} \epsilon_i$$

subject to

$$y_i(\beta_0 + w \cdot x_i) \geq (1 - \epsilon_i) \quad \text{for all } i = 1, \ldots, n,$$

$$\epsilon_i \geq 0 \quad \text{for all } i = 1, \ldots, n.$$

The penalty $D \geq 0$ serves a function similar to the budget C, but is inversely related to it.

Finding the support vector classifier

$$\min_{\beta_0, w, \epsilon} \quad \frac{1}{2}\|w\|^2 + D \sum_{i=1}^{n} \epsilon_i$$

subject to

$y_i(\beta_0 + w \cdot x_i) \geq (1 - \epsilon_i) \quad$ for all $i = 1, \ldots, n.$

$\epsilon_i \geq 0 \quad$ for all $i = 1, \ldots, n.$

Introducing Lagrange multipliers, α_i and μ_i, this is equivalent to:

$$\max_{\alpha, \mu} \; \min_{\beta_0, w, \epsilon} \quad \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\beta_0 + w \cdot x_i) - 1 + \epsilon_i] + \sum_{i=1}^{n} (D - \mu_i)\epsilon_i$$

subject to $\quad \alpha_i \geq 0, \mu_i \geq 0, \quad$ for all $i = 1, \ldots, n.$

Finding the support vector classifier

$$\max_{\alpha,\mu} \min_{\beta_0,w,\epsilon} \quad \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i[y_i(\beta_0 + w \cdot x_i) - 1 + \epsilon_i] + \sum_{i=1}^{n}(D - \mu_i)\epsilon_i$$

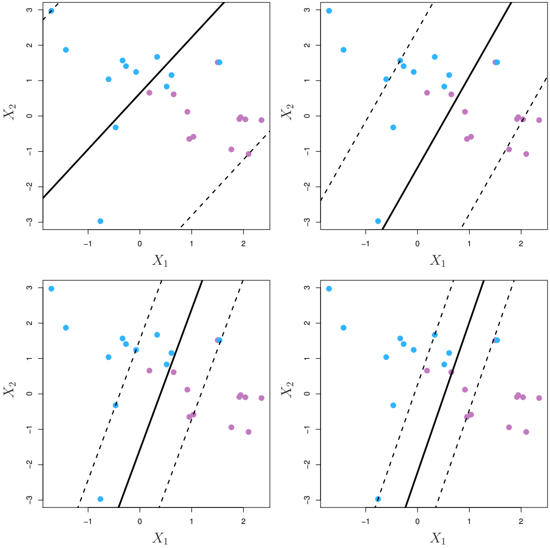
subject to $\alpha_i \geq 0, \mu_i \geq 0, \quad$ for all $i = 1, \ldots, n.$

Setting the derivatives with respect to w to 0, we obtain that the solution is of the form:

$$\hat{w} = \sum_{i=1}^{n} \alpha_i y_i x_i$$

Furthermore, $\alpha_i > 0$ if and only if $y_i(\beta_0 + w \cdot x_i) \leq 1$, that is, if x_i falls on the wrong side of the margin.

Support vectors



The problem only depends on $x_i \cdot x_{i'}$

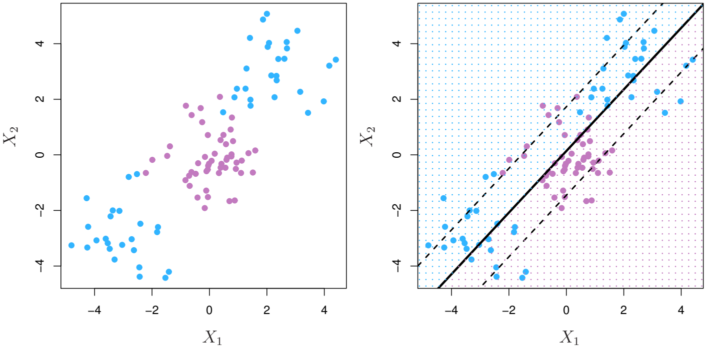
As with the Maximal Margin Classifier, the problem can be reduced to finding $\alpha_1, \dots, \alpha_n$:

$$\begin{aligned}
&\max_{\alpha} \ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} (x_i \cdot x_{i'}) \\
&\text{subject to} \quad 0 \leq \alpha_i \leq D \ \text{ for all } i = 1, \dots, n, \\
&\quad \sum_i \alpha_i y_i = 0.
\end{aligned}$$

As before, this only depends on the training sample inputs through the inner products $x_i \cdot x_j$ for every pair i, j.

How to deal with non-linear boundaries?

The support vector classifier can only produce a linear boundary.



How to deal with non-linear boundaries?

- In **logistic regression**, we dealt with this problem by adding transformations of the predictors.
- The original decision boundary is a line:

$$\log\left[\frac{P(Y=1|X)}{P(Y=0|X)}\right] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0.$$

- With a quadratic predictor, we get a quadratic boundary:

$$\log\left[\frac{P(Y=1|X)}{P(Y=0|X)}\right] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 = 0.$$

How to deal with non-linear boundaries?

- With a **support vector classifier** we can apply a similar trick.
- The original decision boundary is the hyperplane defined by:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0.$$

- If we expand the set of predictors to the 3D space (X_1, X_2, X_1^2), the decision boundary becomes:

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

- This is in fact a linear boundary in the 3D space. However, we can classify a point knowing just (X_1, X_2). The boundary in this projection is quadratic in X_1.

How do we expand the space of predictors?

- **Idea:** Add polynomial terms up to degree d:

$$Z = (X_1, X_1^2, \ldots, X_1^d, X_2, X_2^2, \ldots, X_2^d, \ldots, X_p, X_p^2, \ldots, X_p^d).$$

- Does this make the computation more expensive?
- Recall that all we need to compute is the dot product:

$$x_i \cdot x_k = \langle x_i, x_k \rangle = \sum_{j=1}^{p} x_{ij} x_{kj}.$$

- With the expanded set of predictors, we need:

$$z_i \cdot z_k = \langle z_i, z_k \rangle = \sum_{j=1}^{p} \sum_{\ell=1}^{d} x_{ij}^{\ell} x_{kj}^{\ell}.$$

Kernels

A **kernel** is a matrix defined by $K(i,k) = \langle z_i, z_k \rangle$, for a set of linearly independent vectors $z_1, \ldots, z_n$.

K is a **positive definite** matrix, i.e. it is symmetric and has positive eigenvalues.

Theorem:

If K is a positive definite $n \times n$ matrix, there exist vectors $(z_1, \ldots, z_n)$ in some space $\mathbf{Z}$, such that $K(i,k) = \langle z_i, z_k \rangle$.

The kernel trick

Expand the set of predictors:

- Find a mapping Φ which expands the original set of predictors $X_1, \ldots, X_p$. For example,

$$\Phi(X) = (X_1, X_2, X_1^2)$$

- For each pair of samples, compute:

$$K(i,k) = \langle \Phi(x_i), \Phi(x_k) \rangle.$$

Define a kernel:

- Prove that a function $f(\cdot,\cdot)$ is positive definite. For example:

$$f(x_i, x_k) = (1 + \langle x_i, x_k \rangle)^2 .$$

- For each pair of samples, compute:

$$K(i,k) = f(x_i, x_k).$$

- Often much easier!

The kernel trick

Example. The polynomial kernel with $d = 2$:

$$K(x_i, x_k) = f(x_i, x_k) = (1 + \langle x_i, x_k \rangle)^2$$

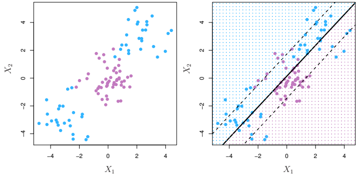
This is equivalent to the expansion:

$$\Phi(X) = (X_1, \ldots, X_p, X_1^2, \ldots, X_p^2, X_1 X_2, X_1 X_3, \ldots, X_{p-1} X_p)$$

- Computing $K(x_i, x_k)$ directly is $O(p)$.
- Computing the kernel using the expansion is $O(p^2)$.

How are kernels defined?

- Proving that a bilinear function $f(\cdot, \cdot)$ is positive definite (PD) is not always easy.
- However, we can easily define PD kernels by combining those we are familiar with:
  - Sums and products of PD kernels are PD.
- Intuitively, a kernel $K(x_i, x_k)$ defines a *similarity* between the samples x_i and x_k. This intuition can guide our choice in different problems.



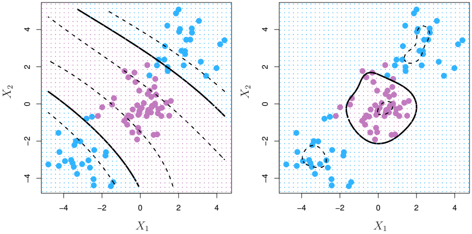
Common kernels

- The polynomial kernel:

$$K(x_i, x_k) = (1 + \langle x_i, x_k \rangle)^d$$

- The radial basis kernel:

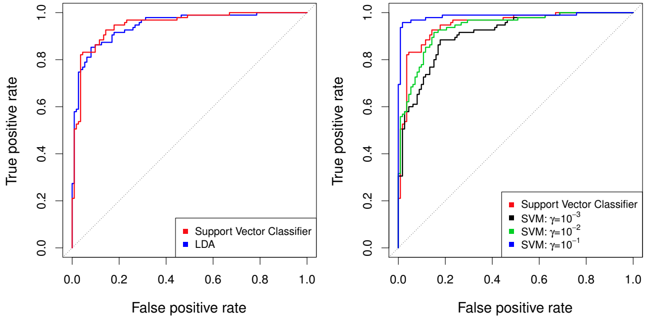
$$K(x_i, x_k) = \exp\left(-\gamma \underbrace{\sum_{j=1}^{p} (x_{ip} - x_{kp})^2}_{\text{Euclidean } d(x_i, x_k)} \right)$$



Dealing with non-standard data types

- Kernels are particularly useful for dealing with data types that cannot easily be represented as vectors, such as:
    1. Strings (gene sequences, search queries)
    2. Graphs (social networks)
    3. Trees
    4. Images, videos.
- It is easier to define a similarity measure which is PD than a set of features that capture the information in each sample.

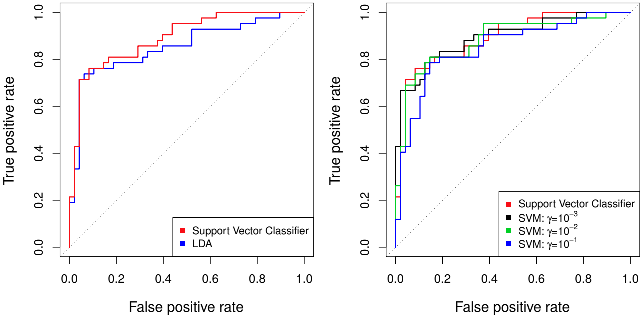
Example. Heart disease dataset



ROC curves computed on the training set.

The SVM uses a radial basis function kernel with different γ's.

Example. Heart disease dataset



ROC curves computed on the test set.