9.1 Maximal Margin Classifier

Support Vector Machines

In this chapter, we approach the two-class classification problem in a direct way: we try to find a plane that separates the classes in the feature space.

If we cannot, we make some adjustments:

- Soften what we mean by "separates".
- Enrich and enlarge the feature space to make separation possible.

Hyperplanes

- In a p-dimensional space, a *hyperplane* is a flat, affine subspace of dimension $p - 1$.
  - *Affine* indicates the subspace does not need to pass through the origin.
- In two dimensions, this is a one-dimensional subspace (a line).
- In three dimensions, a plane.

Hyperplanes

In two dimensions, a hyperplane is defined by

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

- Any $X = (X_1, X_2)^T$ for which this holds is a point on the hyperplane.

In p dimensions,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

defines a p-dimensional hyperplane.

Hyperplanes

Now, suppose that X does not satisfy this equation. That is,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0$$

- then X is to one side of the hyperplane

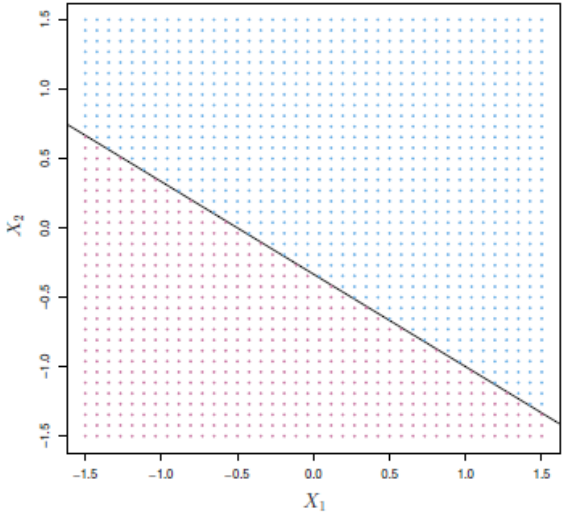
or if

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0$$

- then X is to the other side of the hyperplane.

So we can think of hyperplanes as dividing a p-dimensional space into two parts.

Hyperplanes



Classification Using Hyperplanes

- Suppose we have some $n \times p$ data matrix X where observations fall into one of two classes.
- We also have a test observation x^*
- Our goal is to develop a classifier based on the training data that will correctly classify the test observation using its feature measurements.
- We have several such approaches already: logistic regression, LDA, classification trees, etc.
- We now consider an approach using a *separating hyperplane*.

Classification Using Hyperplanes

- Now suppose it's possible to construct a hyperplane that fully separates the training observations by their class levels.
- If we label the classes as -1 and 1, we can say that

$$\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} > 0$$

if $y_1 = 1$ and

$$\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} < 0$$

if $y_1 = -1$ or that a separating hyperplane is such that

$$y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip}) > 0$$

for all $i = 1, \ldots, n$.

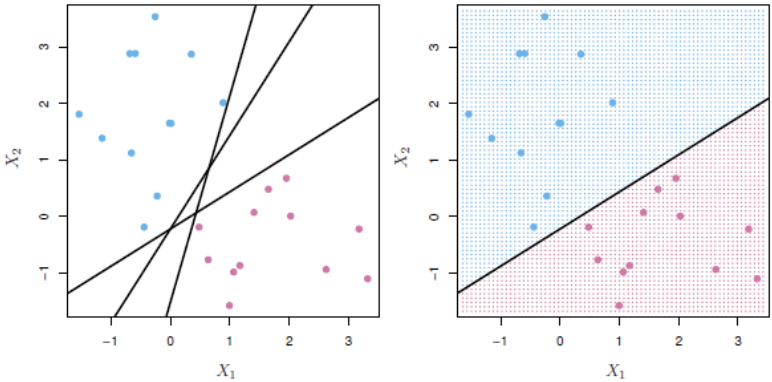
Classification Using Hyperplanes

- If a separating hyperplane exists, there is a natural classifier:
  - a test observation is assigned a class depending on which side of the hyperplane it is located.
  - Note: separating hyperplanes will lead to linear decision boundaries.
- The *magnitude* of $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_p x_p^*$ acts as a measure of how confident we are in the class assignment
  - If $f(x^*)$ is far from 0, the point is far from the hyperplane.

The Maximal Margin Classifier

- If our data can be perfectly separated using a hyperplane, there there are an infinite number of such hyperplanes.
  - We would like to choose the best one.

Example



Three possible separating hyperplanes are shown on the left.

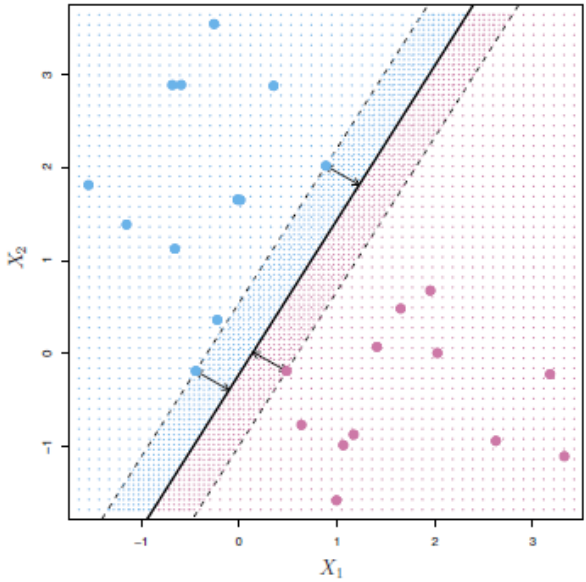
The Maximal Margin Classifier

- A natural choice of "best" hyperplane is the *maximal margin hyperplane* (or the *optimal separating hyperplane*)
  - This is the one that is farthest from the training observations.
- We find the (perpendicular) distance from each training observation to a given separating hyperplane
  - The smallest such distance is known as the *margin*
- The *maximal margin classifier* utilizes the hyperplane for which the margin is largest.
  - That is, the hyperplane has the farthest minimum distance to the training observations.

The Maximal Margin Classifier

- We hope that a classifier with a large margin on the training data will also have a large margin on the test data.
  - This should tend to classify correctly.
  - However, these can lead to overfitting when p is large.

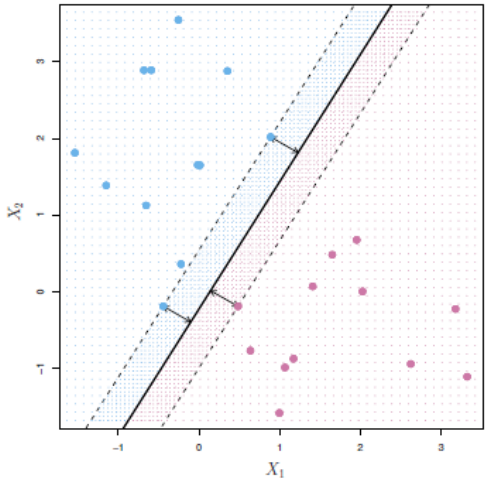
Example



Example

- The three points which are equidistant from the hyperplane (on the dashed lines) are known as *support vectors*
  - If any were to be moved, the hyperplane would shift.
  - The maximal margin hyperplane depends directly on these points (and not the other points).

Construction of the Maximal Margin Classifier



Constrained optimization problem

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize}}\, M$$

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

$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M$$

$$\text{for all } i = 1, \ldots, N.$$

Construction of the Maximal Margin Classifier

This turns out to be simpler than it looks!

- The constraint that

$$y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip}) \geq M; \quad M > 0$$

  guarantees complete and correct separation of the training data.
- The constraint

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

  forces the perpendicular distance from the ith observation to the hyperplane to be

$$y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip})$$

- Then M is the margin for the hyperplane, and the optimization problem maximizes this value.
  - (The actual optimization problem is outside the scope of this course.)

Maximal Margin Classifiers in R

```
library(e1071)
?svm
```

- We will see this in detail in the next section.

The Non-Separable Case

- The maximal marginal classifier is a natural way to perform classification, but it does require the classes be fully separable.
- In many (most?) cases, no such hyperplane exists and the optimization problem has no solution.
- We will need to generalize these ideas a bit to continue to generate support vector machine classifiers in this scenario.