Lecture 10: Support vector classifier

Reading: Sections 9.1-9.2

STATS 202: Data mining and analysis

Rajan Patel

Hyperplanes and normal vectors

- Consider a p-dimensional space of predictors.
- A **hyperplane** is an affine space which separates the space into two regions.
- The normal vector $\beta = (\beta_1, \ldots, \beta_p)$, is a unit vector $\sum_{j=1}^{p} \beta_j^2 = 1$ which is perpendicular to the hyperplane.
- If the hyperplane goes through the origin, the deviation between a point $(x_1, \ldots, x_p)$ and the hyperplane is the dot product:

$$x \cdot \beta = x_1\beta_1 + \cdots + x_p\beta_p.$$

- The sign of the dot product tells us on which side of the hyperplane the point lies.

Hyperplanes and normal vectors

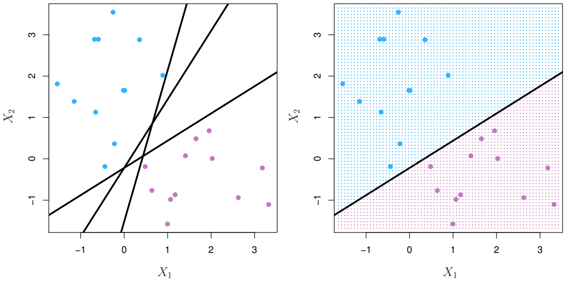
- Consider a p-dimensional space of predictors.
- A **hyperplane** is an affine space which separates the space into two regions.
- The normal vector $\beta = (\beta_1, \ldots, \beta_p)$, is a unit vector $\sum_{j=1}^p \beta_j^2 = 1$ which is perpendicular to the hyperplane.
- If the hyperplane goes through a point $-\beta_0\beta$, i.e. it is displaced from the origin by $-\beta_0$ along the normal vector, the deviation of a point $(x_1, \ldots, x_p)$ from the hyperplane is:

$$\beta_0 + x_1\beta_1 + \cdots + x_p\beta_p.$$

- The sign tells us on which side of the hyperplane the point lies.

Maximal margin classifier

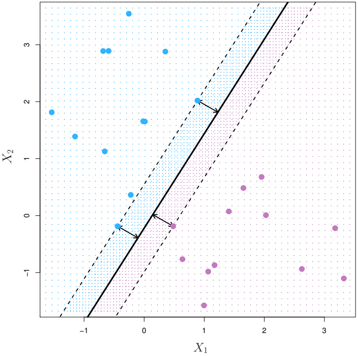
- Suppose we have a classification problem with response $Y = -1$ or $Y = 1$.
- If the classes can be separated, most likely, there will be an infinite number of hyperplanes separating the classes.



Maximal margin classifier

Idea:

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



Maximal margin classifier

This can be written as an optimization problem:

$$\max_{\beta_0,\beta_1,\ldots,\beta_p} \quad M$$

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

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

M is simply the width of the margin in either direction.

Finding the maximal margin classifier

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

$$\min_{\beta_0, w} \quad \frac{1}{2}\|w\|^2$$

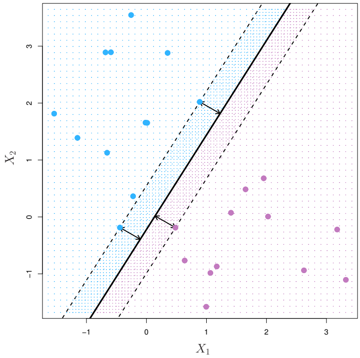
subject to

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

This is a quadratic optimization problem.

Support vectors

The vectors that fall on the margin and determine the solution are called **support vectors**:

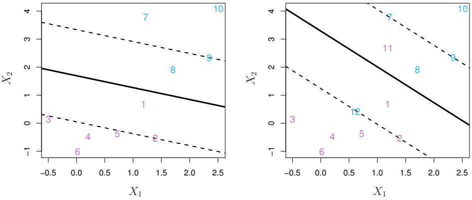


Support vector classifier

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 even the hyperplane.



Support vector classifier

This can be written as an optimization problem:

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

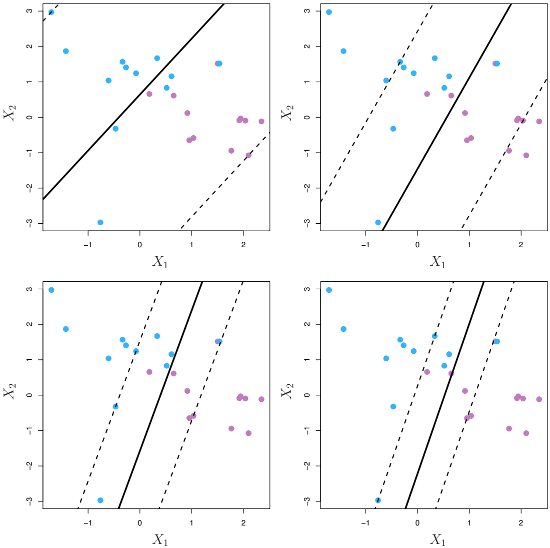
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

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

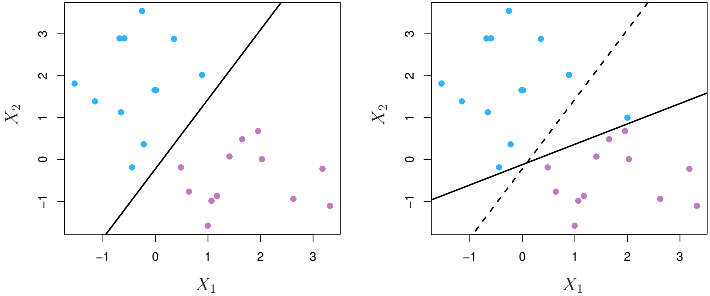
$$\epsilon_i \geq 0 \text{ for all } i = 1, \ldots, n, \quad \sum_{i=1}^{n} \epsilon_i \leq C.$$

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 $w = (w_1, \ldots, w_p) = \beta/M$:

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

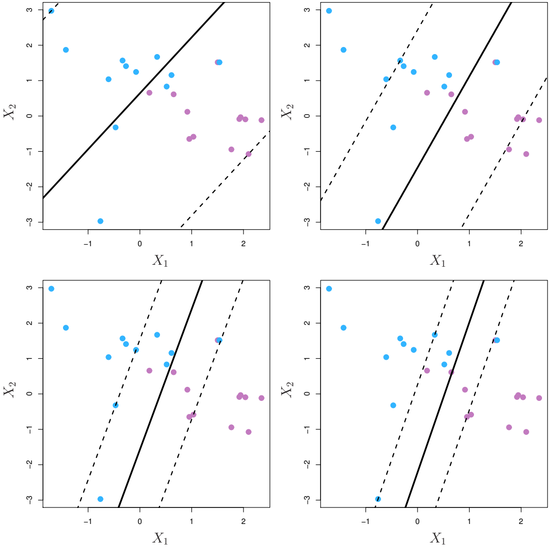
subject to

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

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

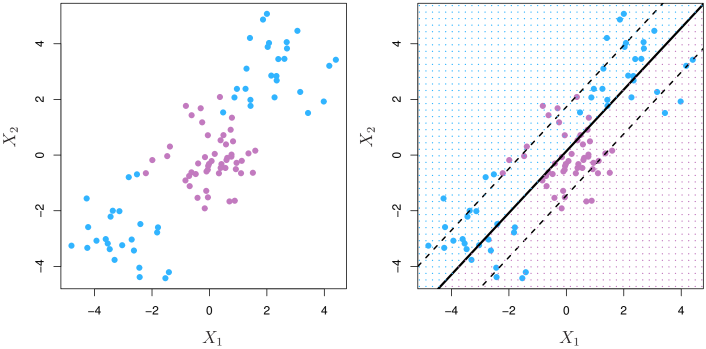
The penalty $D \geq 0$ serves a function similar to the budget C, but is inversely related to it.

Support vectors



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.

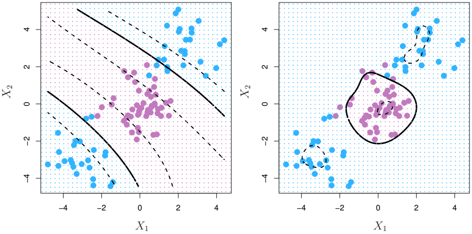
Common kernels

- The polynomial kernel:

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

- The radial basis kernel:

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



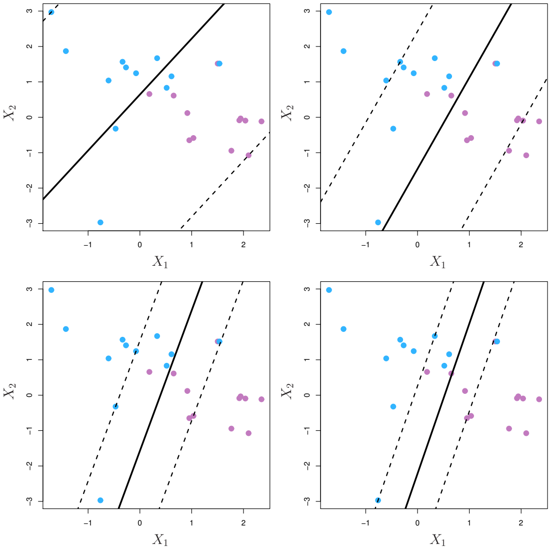
Review of support vector classifier

- The **support vector classifier** defines a hyperplane and two margins.
- **Goal:** to maximize the width of the margins, with some budget C for "violations of the margins", i.e.

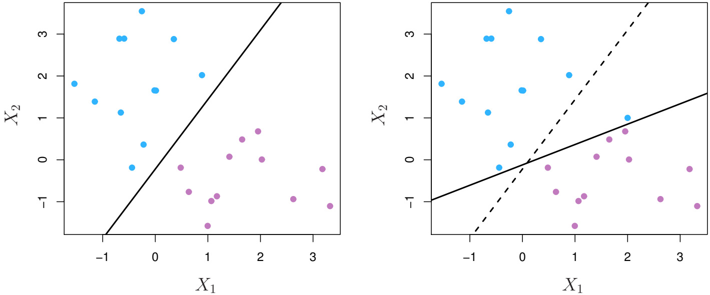
$$\sum_{\substack{x_i \text{ on the wrong} \\ \text{side of the margin}}} \text{Distance from } x_i \text{ to the margin} \ \leq \ C.$$

- The only points that affect the orientation of the hyperplane are those at the margin or on the wrong side of it.
- Low budget $C \iff$ Few samples used $\iff$ High variance $\iff$ Tendency to overfit.

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.