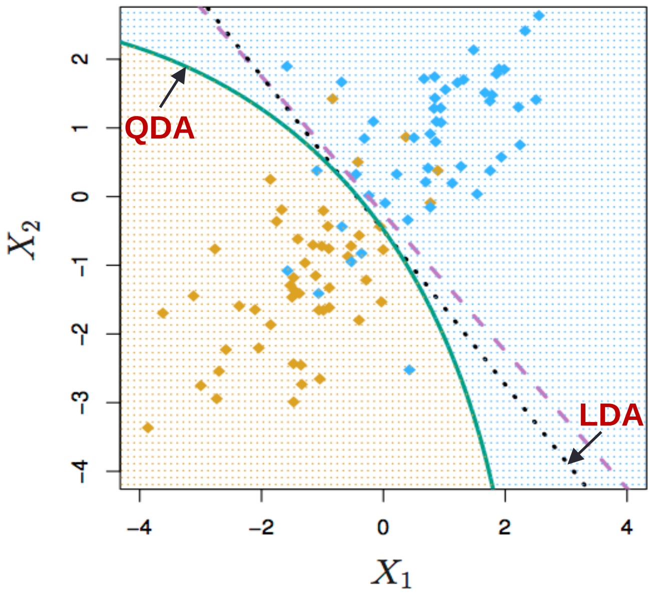
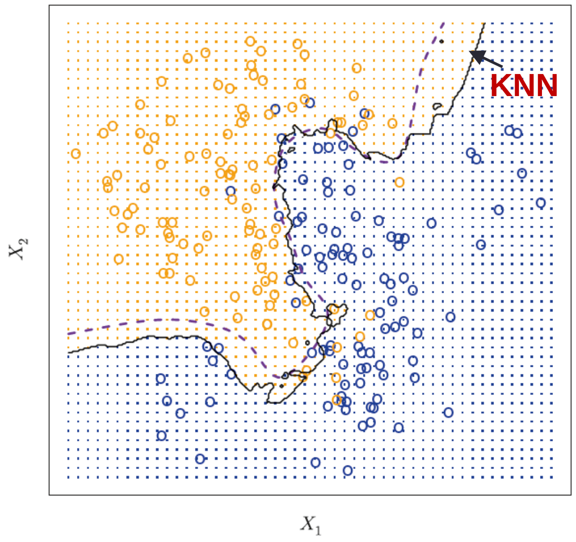
Topic 7. Support Vector Machines

Support Vector Machines (SVM)

- Approach for classification developed in computer science
- One of the best "out of the box" classifiers

- Recall classifiers covered in Chapter 4: logistic regression, LDA, QDA, KNN





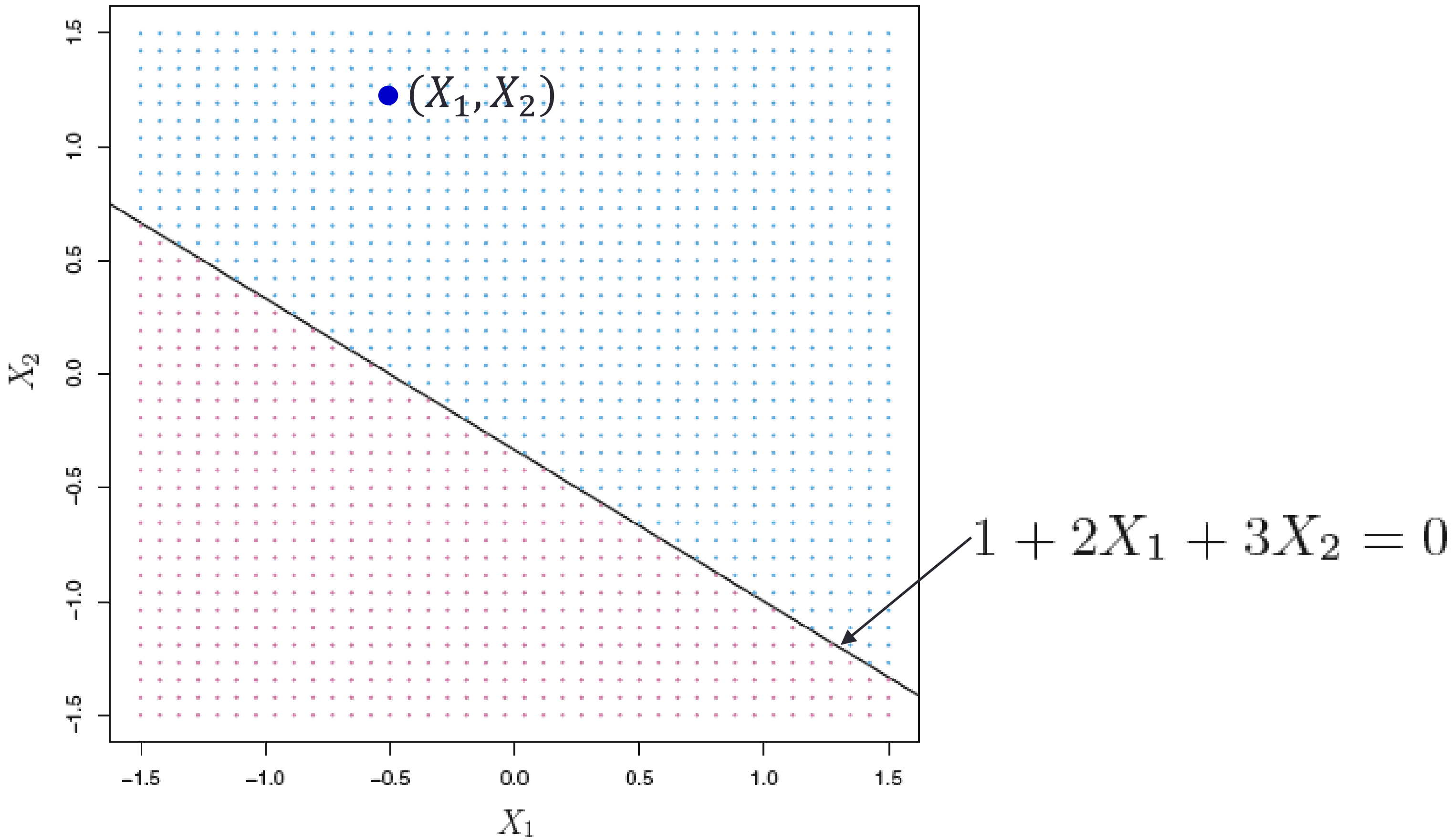
Outline

- Maximal Margin Classifier: classes are separable by a linear boundary
- Support Vector Classifier: extend to cases where classes are not separable
- Support Vector Machine: extend to non-linear class boundaries

Maximal Margin Classifier

Hyperplane

- In a p-dimensional space, an affine *hyperplane* is a flat affine subspace of dimension $p-1$. For example, in two dimensions, a hyperplane is a line.



Classification Problem

- **Feature space**
  - Space formed by the predictors
  - Also referred to as the state space, input space
  - p-dimensional (p predictors), n-points (n observations)

- **Training data**
  - Inputs determine its location in the feature space

$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix}$$

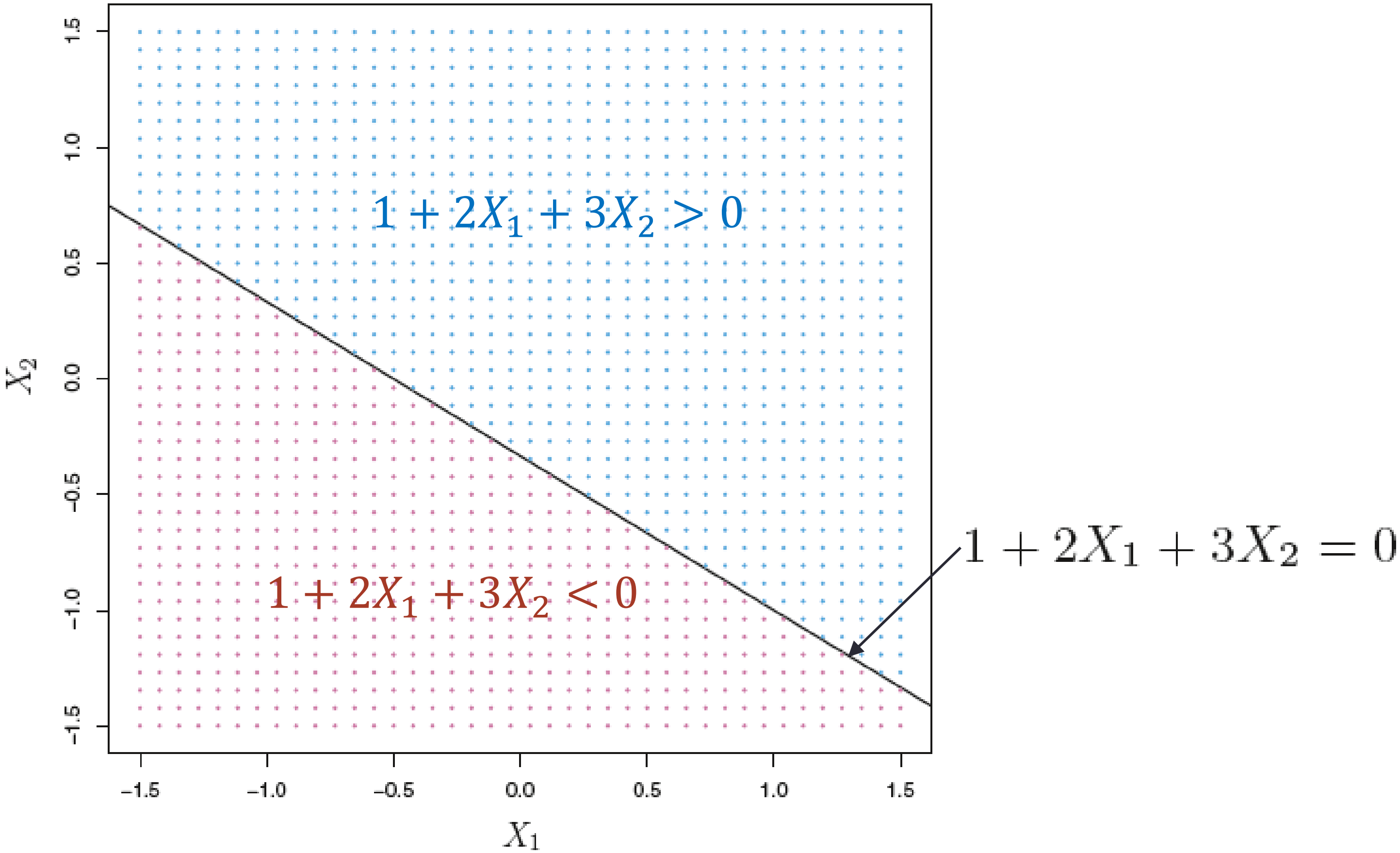
  - Outputs $y_1, \ldots, y_n$ determine the **color** (i.e., **classes**)

- **Classification**: <u>Find the hyperplane </u>such that a test point

$$x^* = \begin{pmatrix} x_1^* & \ldots & x_p^* \end{pmatrix}^T$$

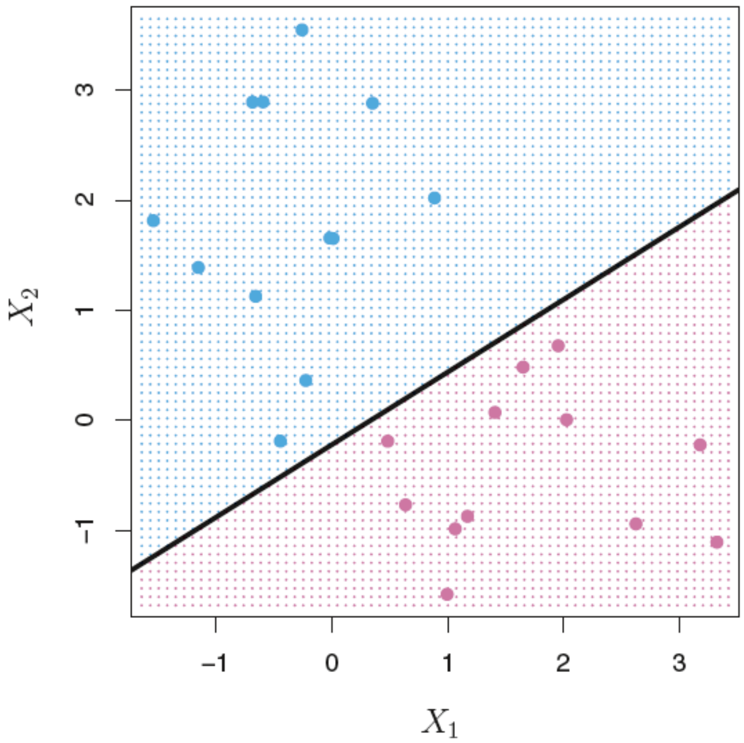
is assigned the correct class.

Classifier as Feature Space Coloring



Separating Hyperplane

- **Separating hyperplane**: separates the training observations perfectly according to their class labels
- Blue: class 1 ($y = 1$)
  Purple: class –1 ($y = -1$)
- $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$
- class 1: $f(x) > 0$
  class –1: $f(x) < 0$



- Property: $\boldsymbol{y_i f(x_i) > 0}$, for all training points $x_1, x_2, \ldots, x_n$

Prediction

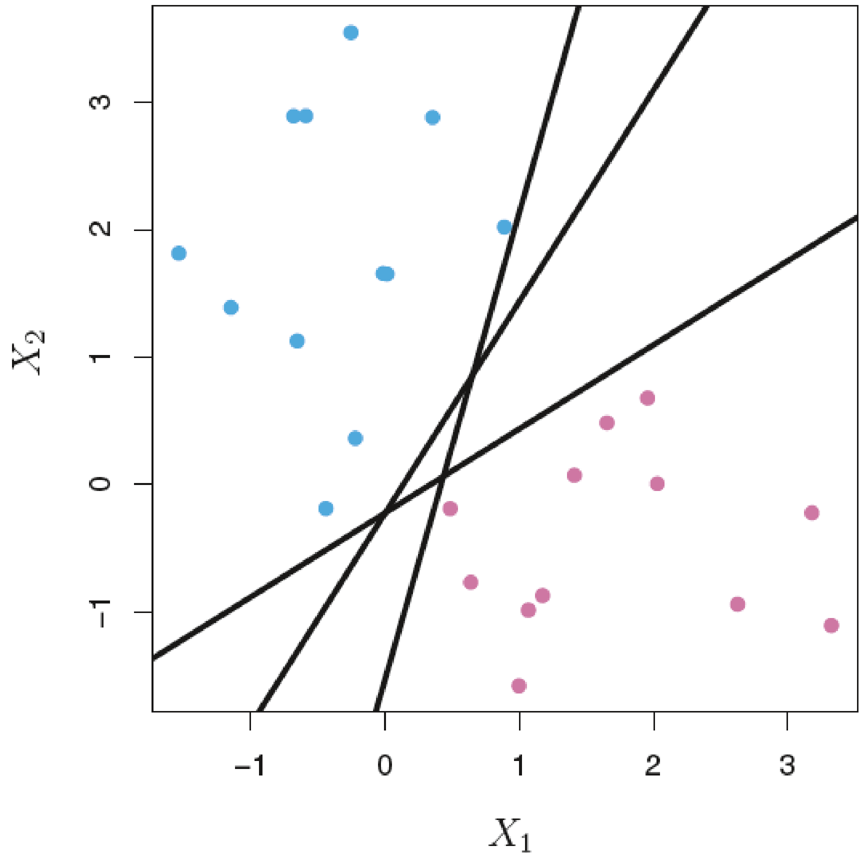
- Given a test point x^*, we will assign it to
  class 1 ($y^* = 1$), if $f(x^*) > 0$
  class –1 ($y^* = -1$), if $f(x^*) < 0$

- If $\boldsymbol{y^* f(x^*)}$ is far from zero, that means the test point lies far from the hyperplane, and so we can be confident about our class assignment for it.

- If $\boldsymbol{y^* f(x^*)}$ is close to zero, that means the test point is located near the hyperplane, and so we are less certain about the class assignment for it.

How to Do It Right?

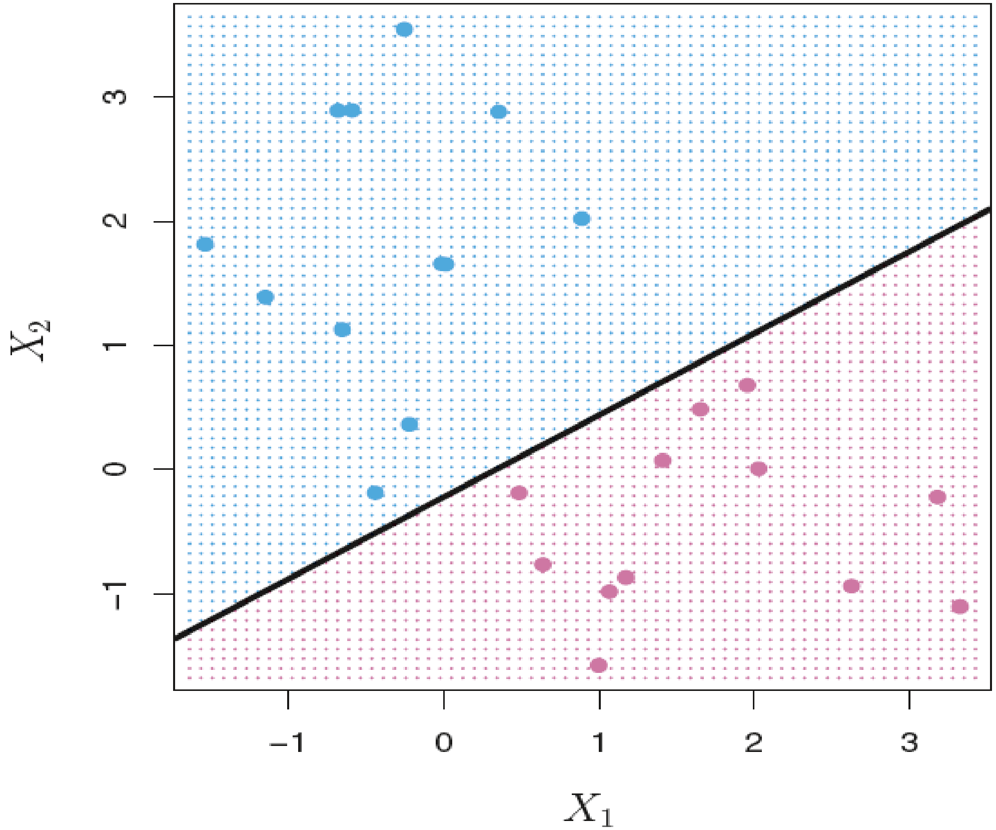
- There may be an infinite number of hyperplanes that separates the training observations perfectly.
- We need to decide which hyperplane to use.



Margin

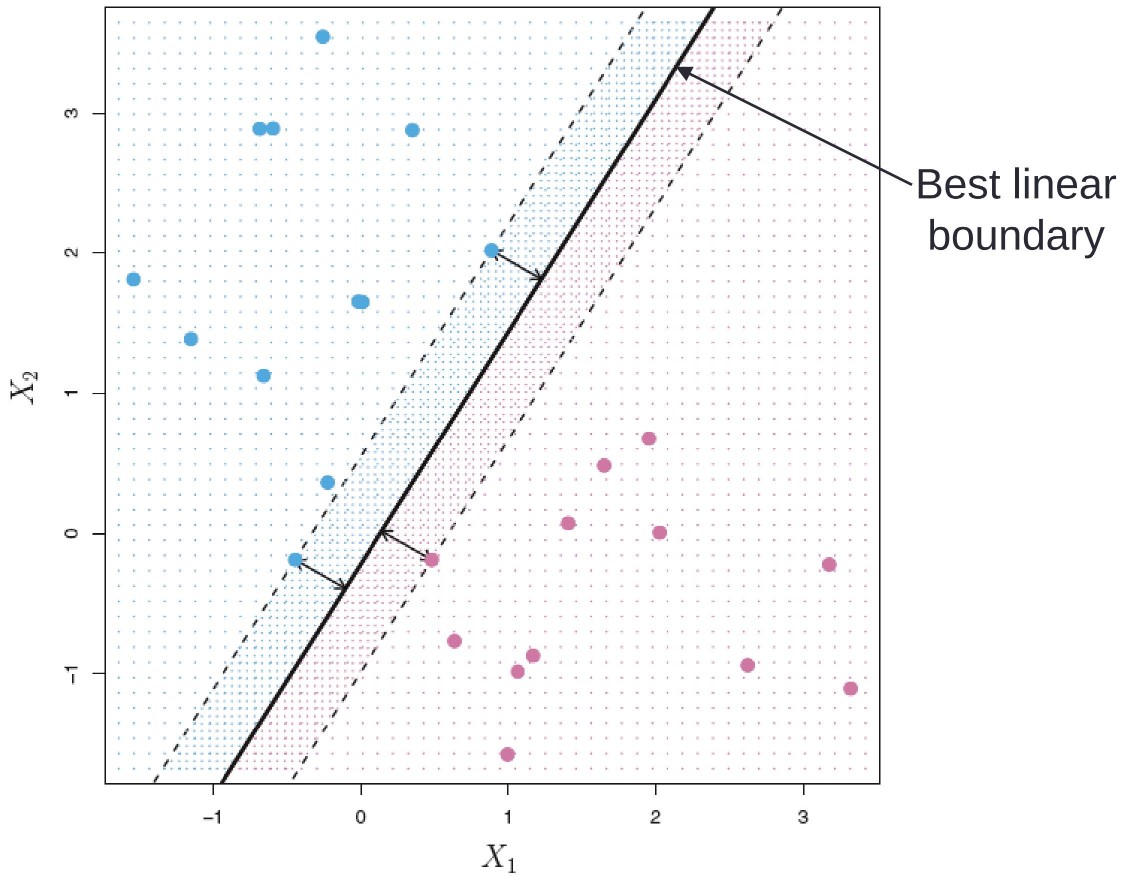
- Suppose we have a separating hyperplane
- Find perpendicular distance from every point to the hyperplane
- Margin: The smallest of such distances, M

i.e., minimum distance from the observations to the hyperplane



Maximal Margin Classifier

- Best separating hyperplane
- Maximize min-distance (max margin)
- Represent the mid-line of the widest “slab” inserted between the two classes



Support Vectors

- Support vectors: the 3 training points equidistant from the maximal margin hyperplane
  - "vectors": each point is a vector in p-dimensional space
  - "support": they support the maximal margin hyperplane: if they were moved slightly then the maximal margin hyperplane would move as well.

- The maximal margin hyperplane depends directly on the support vectors, but not on the other observations: a movement of any of those observations would not affect the separating hyperplane.

Optimization Formulation

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

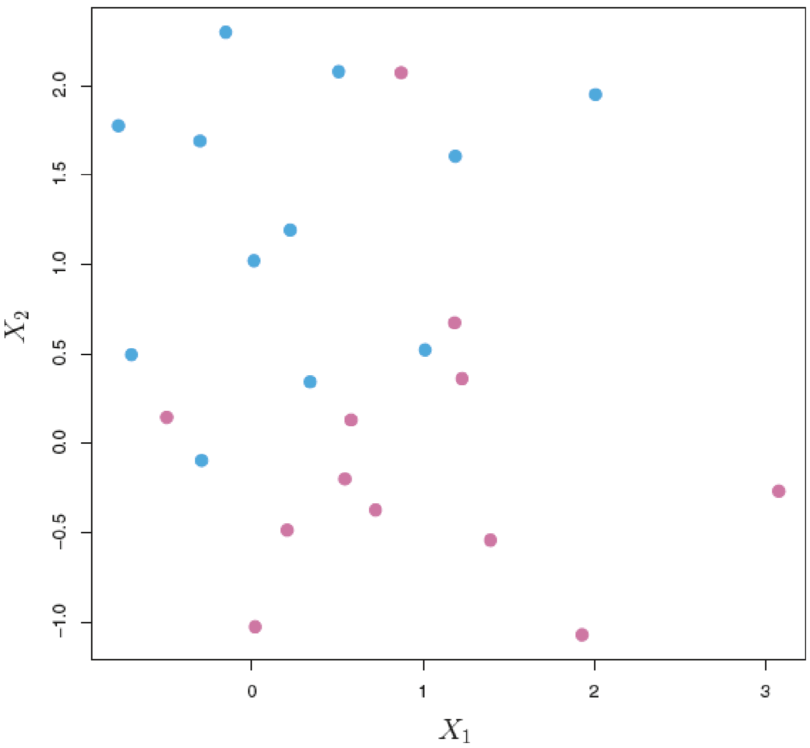
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \quad \forall\, i = 1, \ldots, n$$

Question: for each point to be on the correct side of the hyperplane, we only need $y_i f(x_i) > 0$. Why here it is required that $y_i f(x_i) > M$ $(M > 0)$?

Non-separable Case

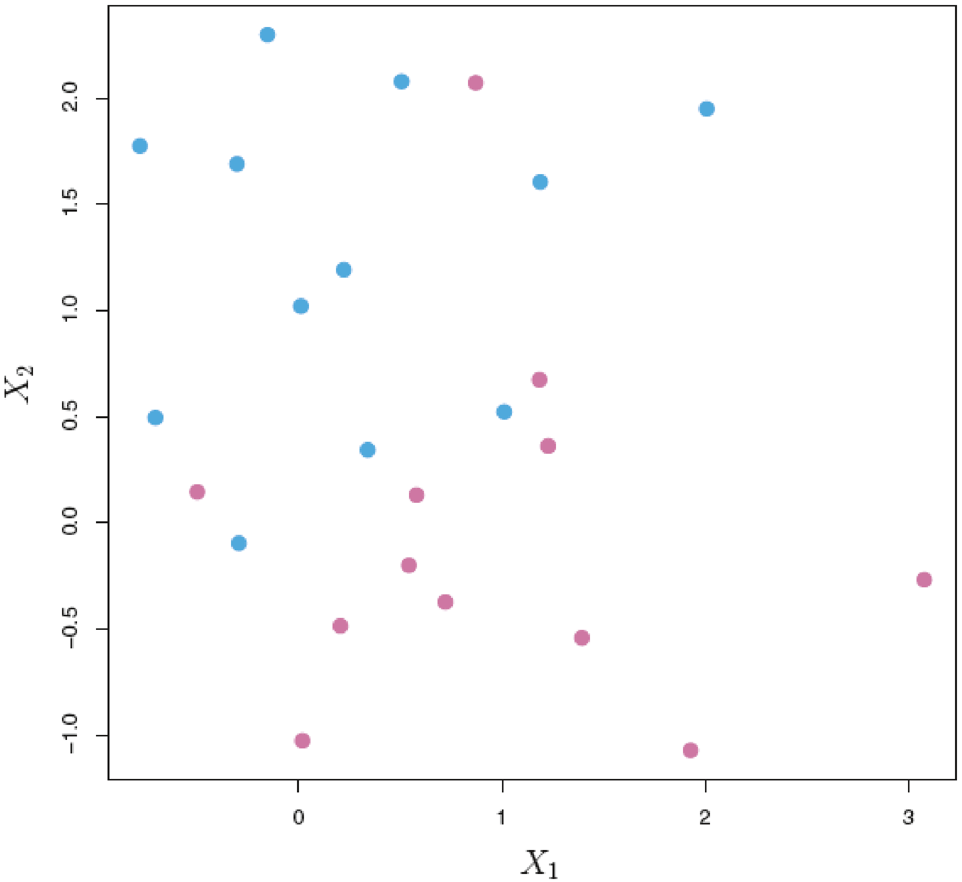
- No solution if the observation classes are mixed (*non-separable* case)
- We can extend the concept of a separating hyperplane to develop a hyperplane that almost separate the classes, using the idea of "soft margin"
- This generalization to the non-separable case is known as the **support vector classifier**



Support Vector Classifier

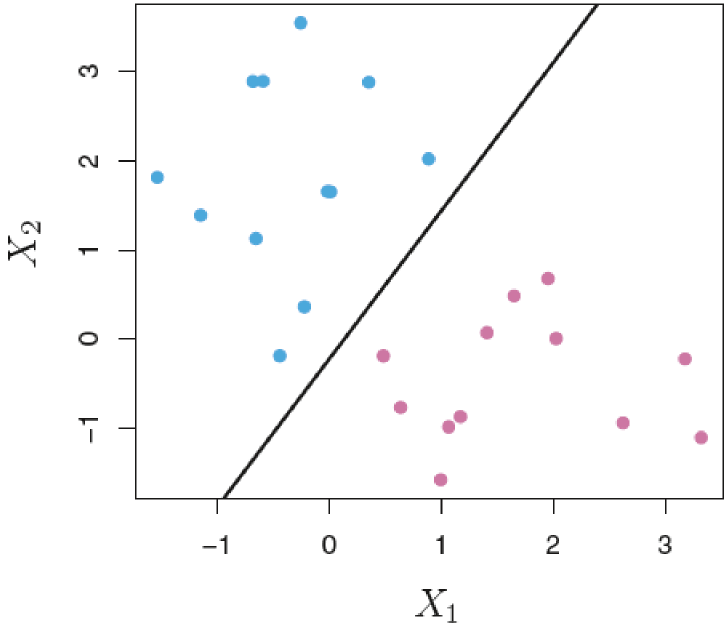
Motivation

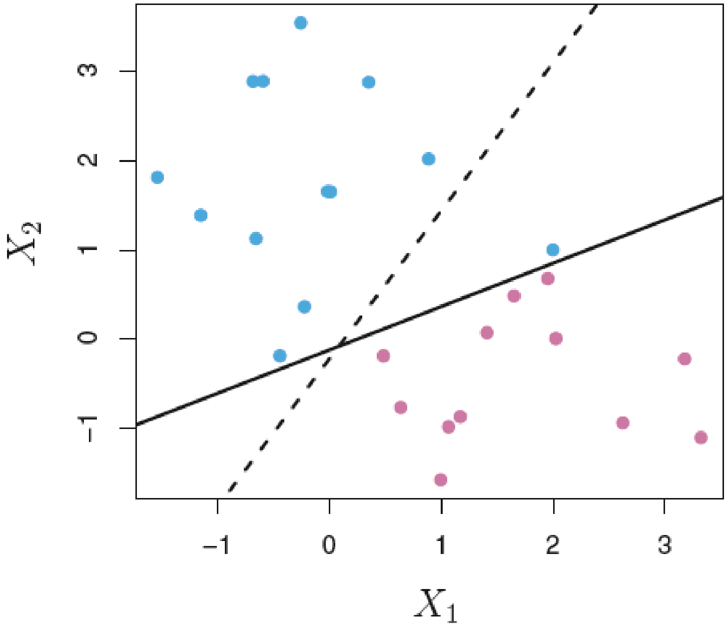
- Maximal margin classifier only works for separable cases
- Need to extend it to the non-separable case



Motivation

- In the separable case, the separating hyperplane is sensitive to individual observations.
- The addition of a single observation may lead to dramatic change in the maximal margin hyperplane.
- Reason: margin is tiny (recall that $yf(x)$ is a measure of our confidence of correct classification)





Idea of Support Vector Classifier

- Consider a hyperplane that does not perfectly separate the two classes, in the interest of

  Greater robustness to individual observations, and

  Better classification of most of the training observations

- It could be worthwhile to misclassify a few training points in order to do a better job in classifying the remaining points.

- Soft margin classifier: the margin is “soft” because it can be violated by some of the training points.

Relax the Constraint of Perfect Separation

- Support vector classifier

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

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

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

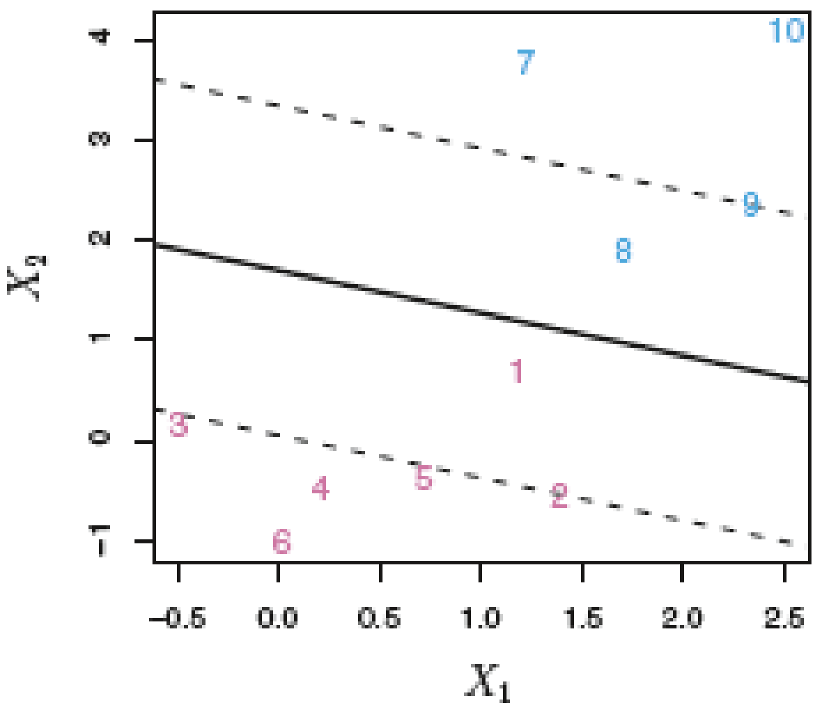
$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

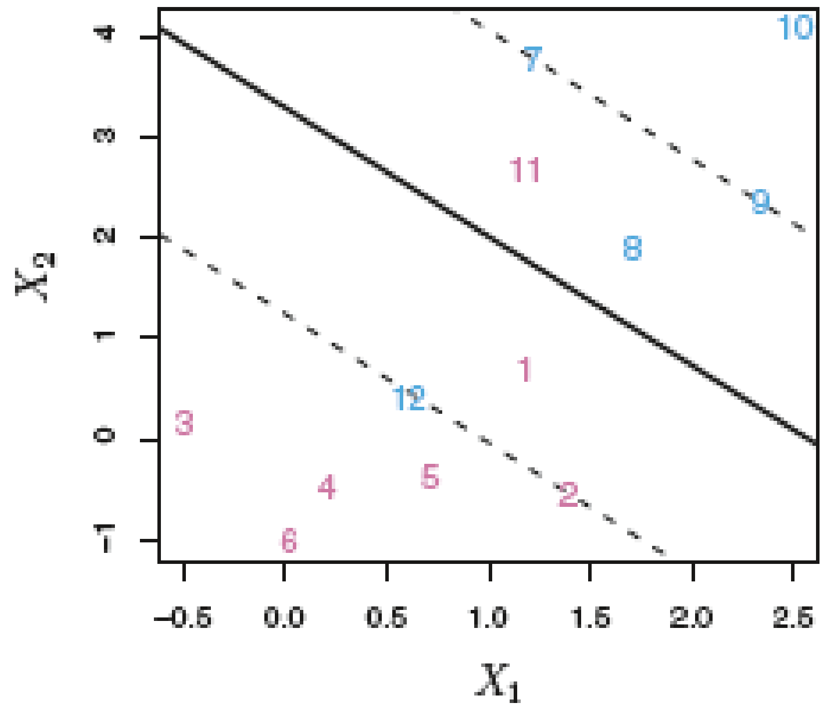
Slack Variables and Tuning Parameter

- For a sample i,
  - $\epsilon_i = 0$ margin not violated (on the correct side of the margin)
  - $\epsilon_i > 0$ margin violated (on the wrong side of the margin)
  - $\epsilon_i > 1$ hyperplane boundary crossed (on the wrong side of the hyperplane, or leakage)

C: tuning parameter, a budget for the amount that the margin can be violated

Effect of ϵ





3,4,5,6: on the correct side of the margin
2: on the margin
1: on the wrong side of the margin

7,10: on the correct side of the margin
9: on the margin
8: on the wrong side of the margin

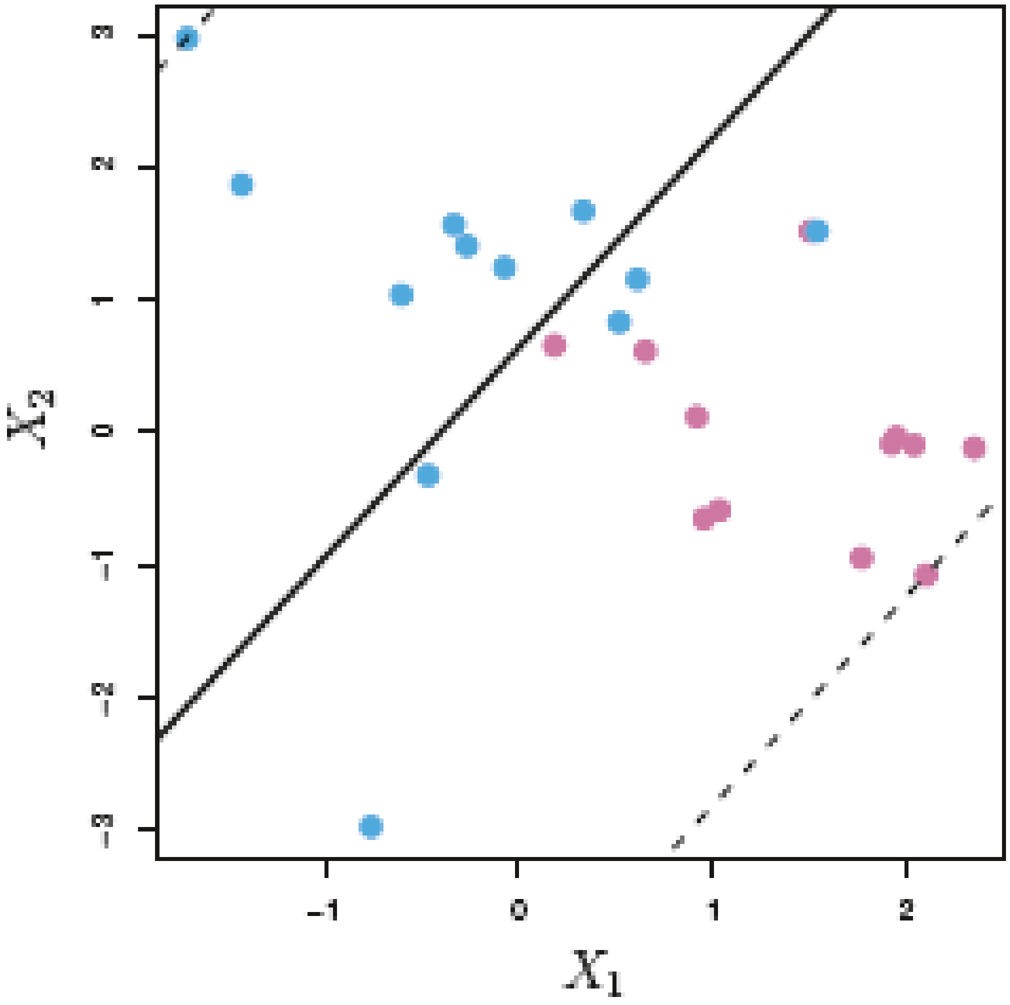
11,12: on the wrong side of the margin and the wrong side of the hyperplane

Effect of *C* — Bias-Variance Trade-off

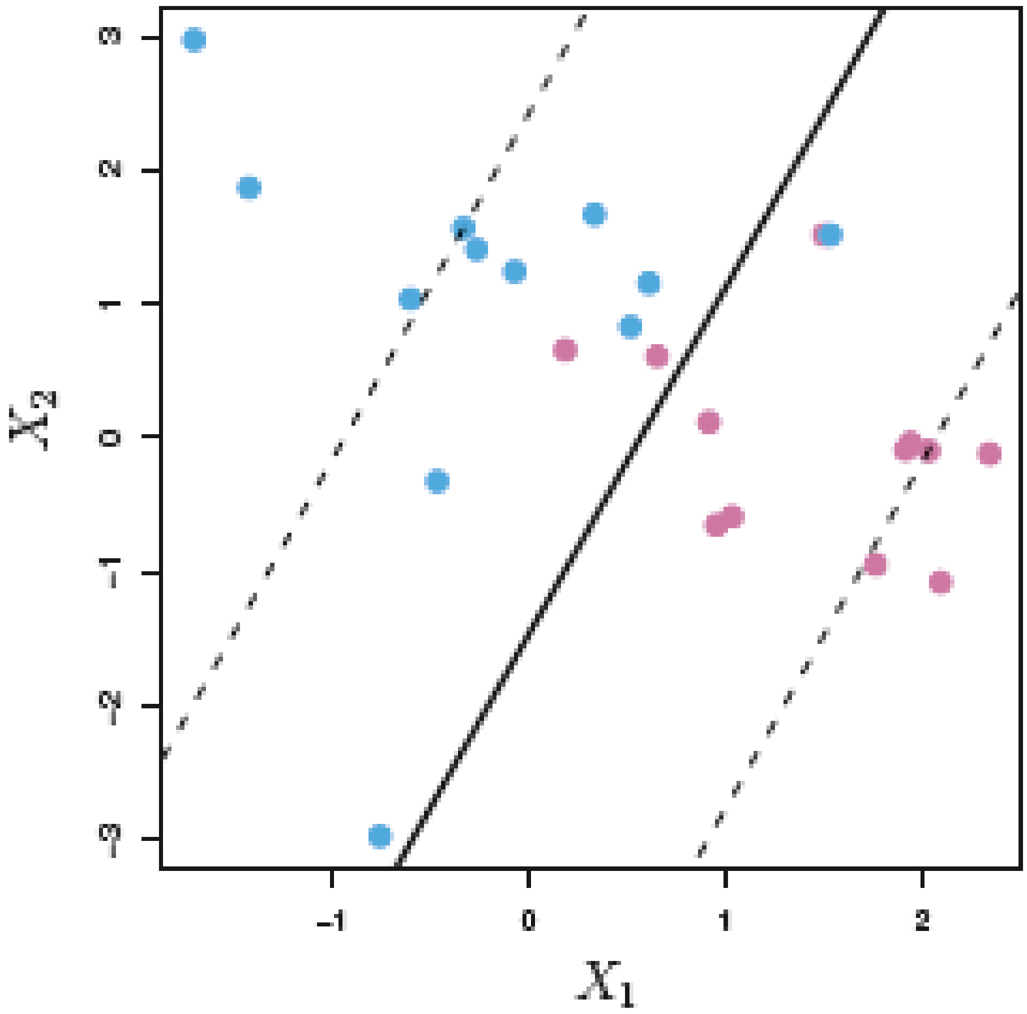
- Small C
  - Narrow margins
  - Rarely violated
  - Fitting data well
  - Low bias but high variance

- Large C
  - Wide margins
  - Allow more violations
  - Fitting data less hard
  - Low variance but high bias

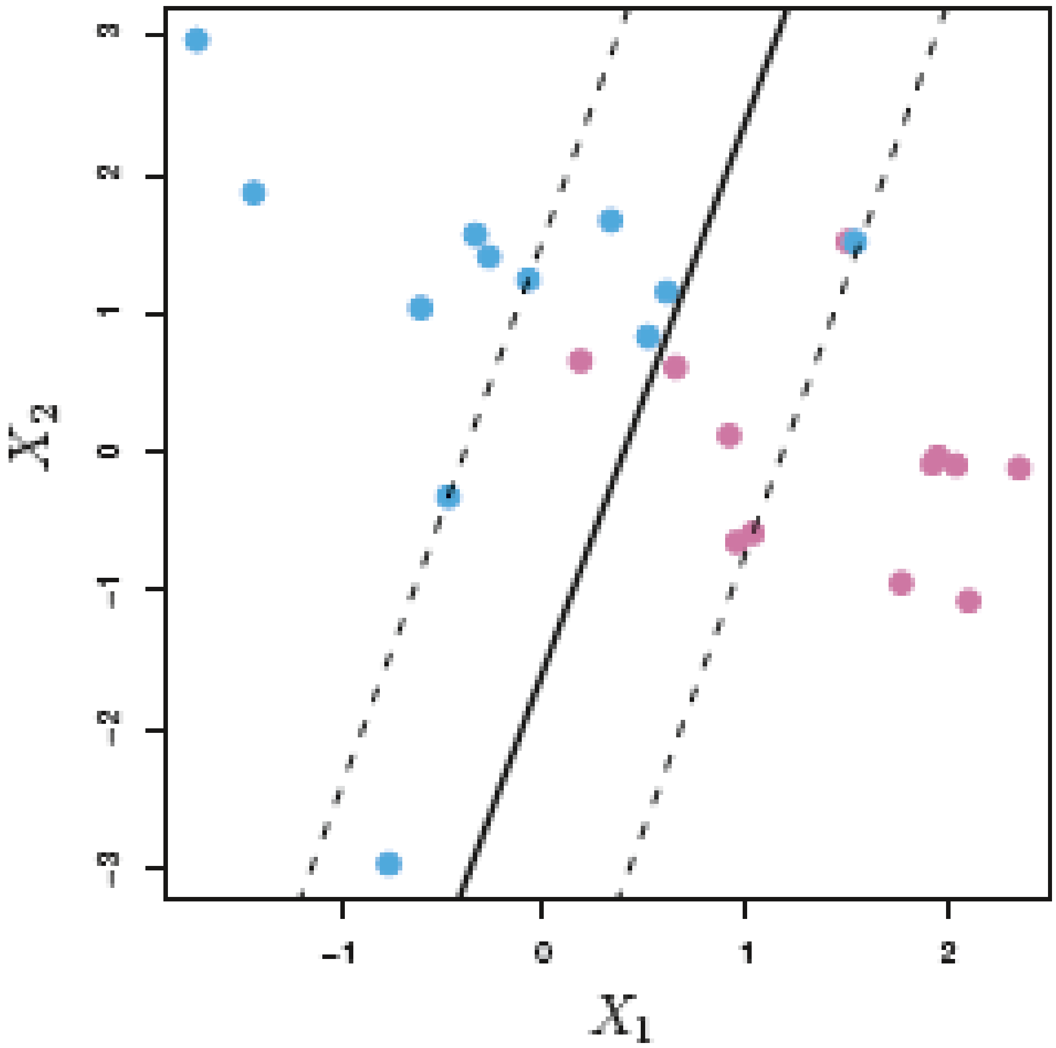
C = 25



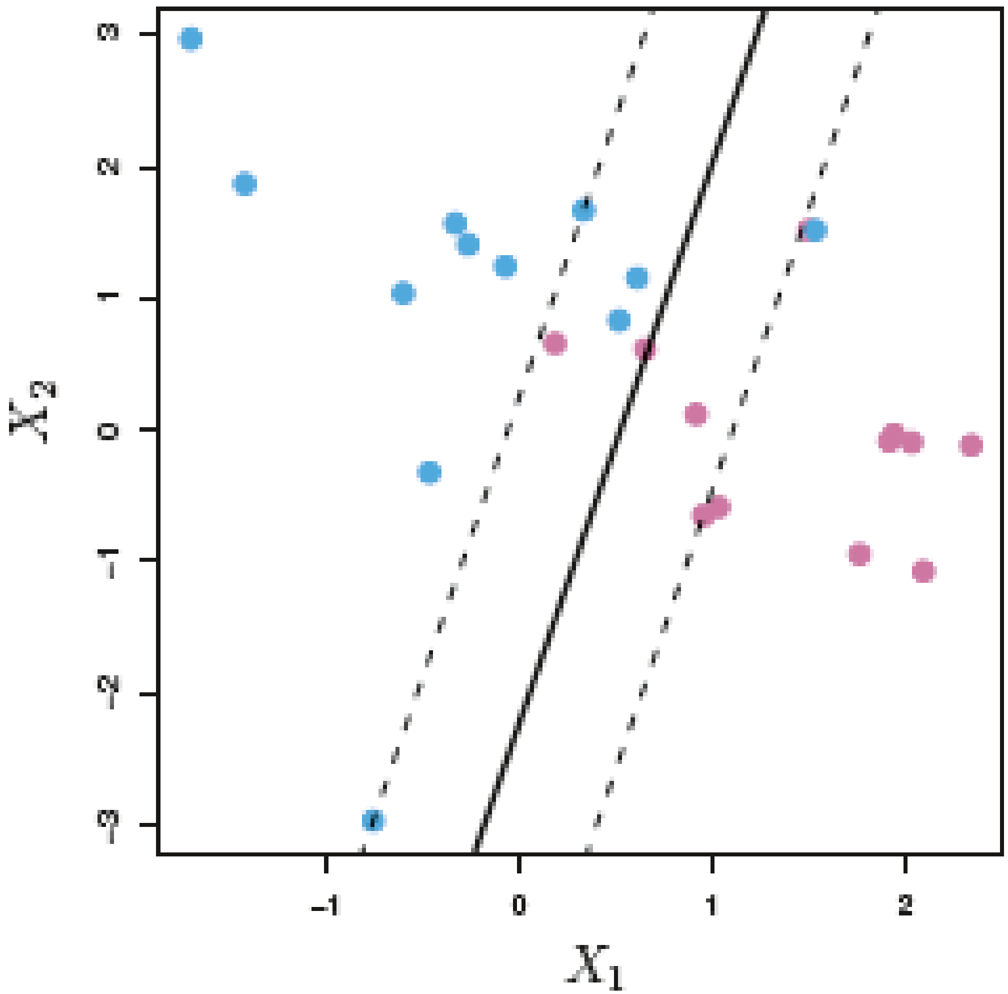
C = 15



$C = 10$



$C = 5$



Properties of the Solution to the Relaxed Formulation

- For the support vector classifier formulation, we have n constraints

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i)$$

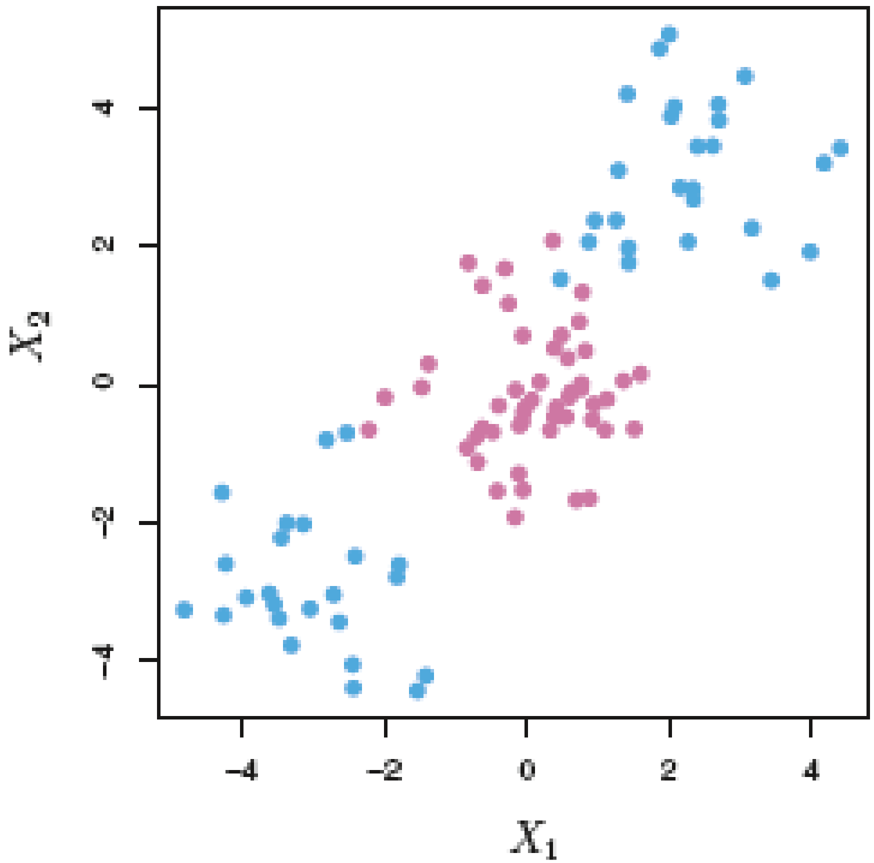
- Support vectors: Points that lie on the margin or on the wrong side of the margin for their class (including those on the wrong side of the hyperplane)

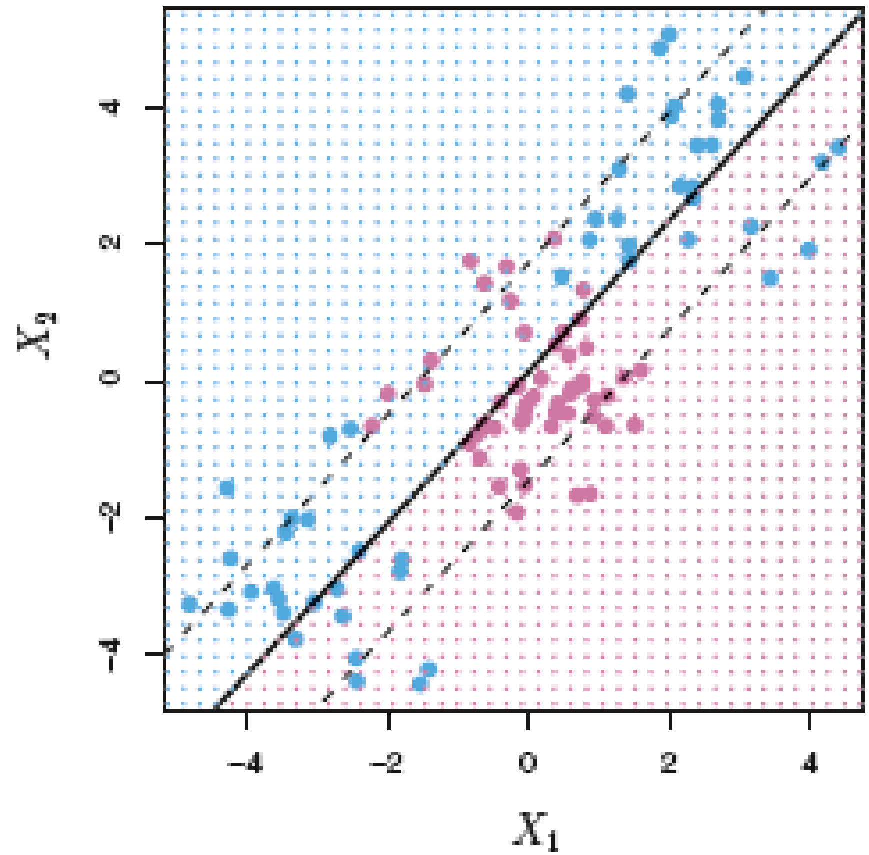
- Observations that lie strictly on the correct side of the margin do not affect the support vector classifier.

Support Vector Machine

Motivation

- What would you use to classify?
- Linear classifier performs poorly.





Accommodating Nonlinearity

- Recall Chapter 3: linear regression may suffer when there is nonlinear relationship between predictors and the response. The solution is to add *transformations* (e.g., quadratic and cubic terms) of predictors into the model. Similar idea can be applied here.

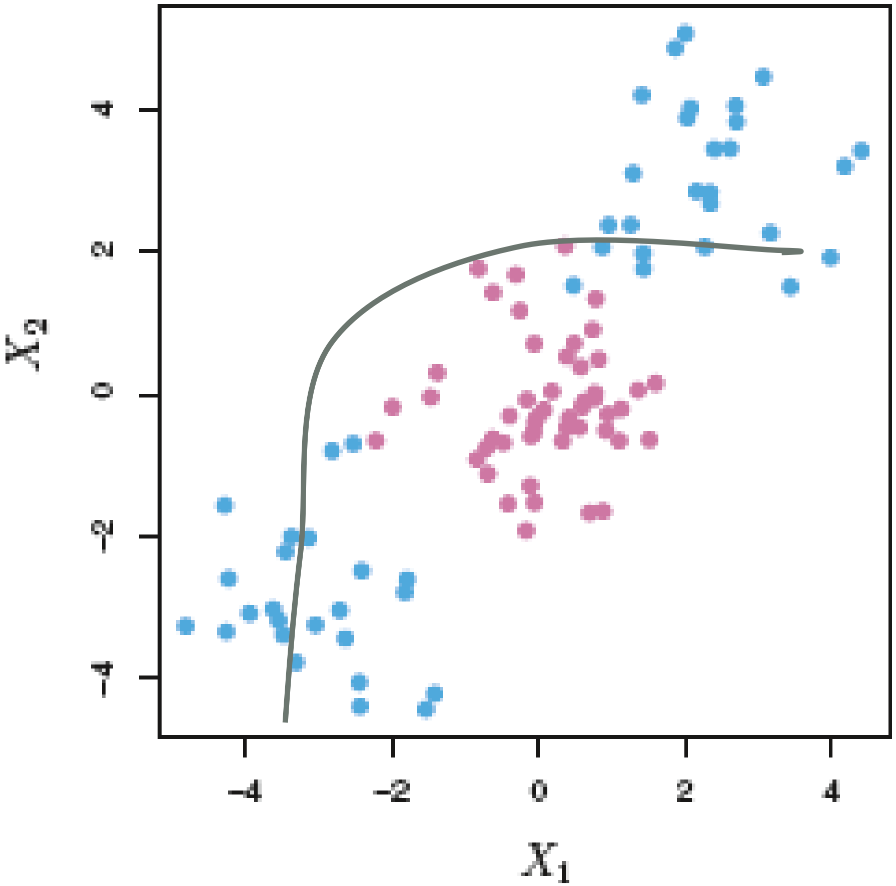
- Instead of fitting a support vector classifier using the p features $X_1, X_2, \ldots, X_p$, we can enlarge the feature space by using $X_1, X_1^2, X_2, X_2^2 \ldots, X_p, X_p^2$.

Modified Optimization Formulation

$$\underset{\beta_0,\beta_{11},\beta_{12},\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M$$

$$\text{subject to } y_i\left(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right) \geq M(1-\epsilon_i)$$

$$\sum_{i=1}^{n}\epsilon_i \leq C, \quad \epsilon_i \geq 0, \quad \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 = 1.$$



However…

- Including quadratic terms is only one way to enlarge the feature space in order to accommodate nonlinearity.
- There are many possible ways to enlarge the feature space. Unless we are careful, we could end up with a huge number of features. Then computations would become unmanageable.
- The support vector machine allows us to enlarge the feature space in a way that leads to efficient computations.

Properties of Optimal Hyperplane

- Let us define an inner product (dot product) of two vectors:

$$\langle a, b \rangle = \sum_{i=1}^{r} a_i b_i$$

For real vectors, it is simply *a*^{T}*b*

- The optimal linear hyperplane *f(x) = 0* can be written as

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle$$

It turns out that $\alpha_i \neq 0$ only for support vectors

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

- **Decision rule is based on the inner product**.

Support Vector Machine (SVM)

- Let us generalize the support vector classifier

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

$$\downarrow$$

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

K is called the *kernel* function

Choices of Kernel Function

- Linear kernel (i.e., support vector classifier)

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$$

- Polynomial kernel

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$$

tuning parameter: d

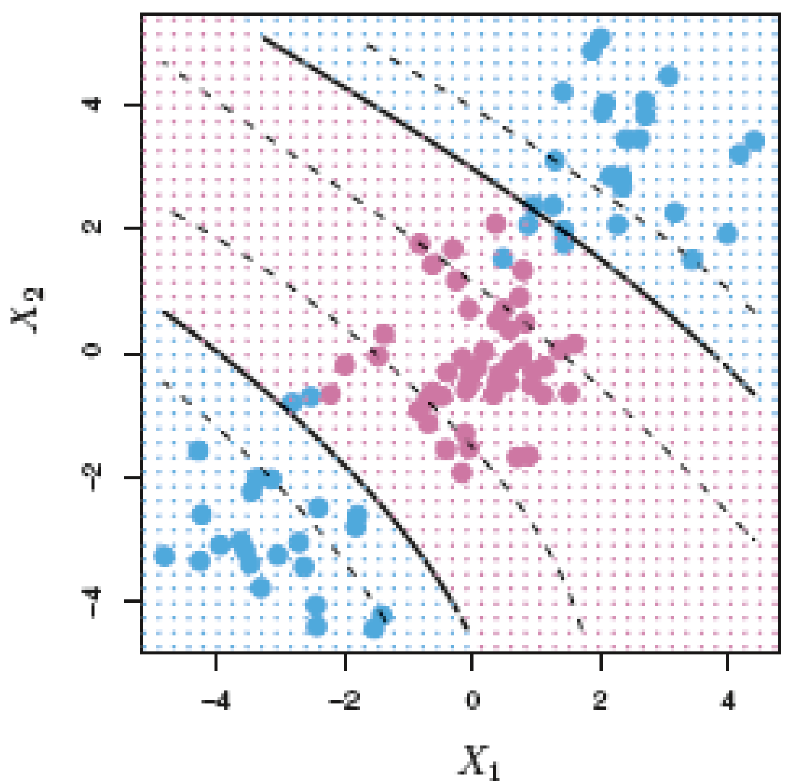
- Radial kernel

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$

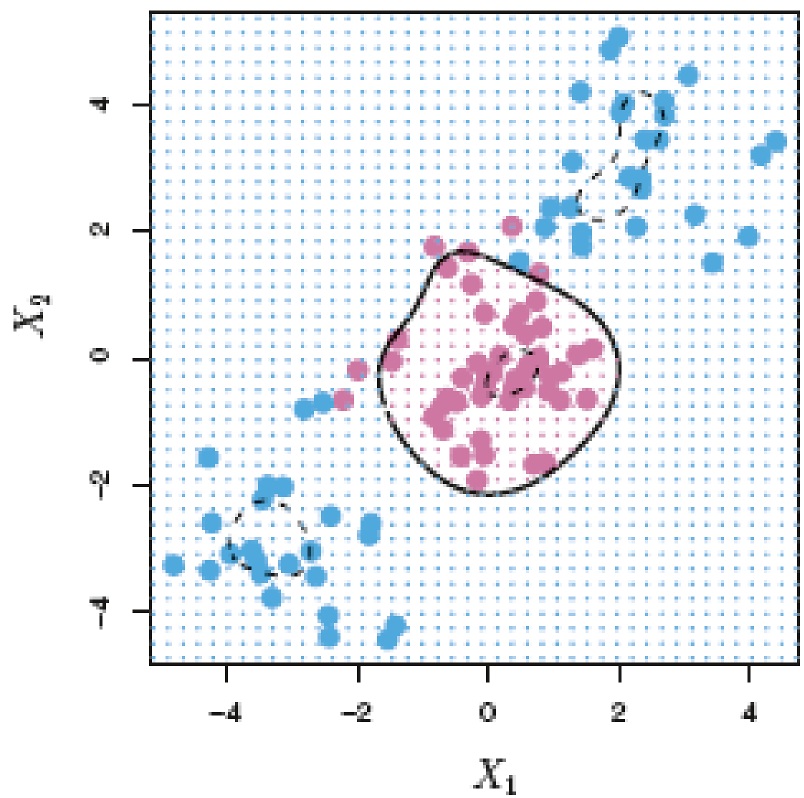
tuning parameter: γ

SVM Boundaries

Degree 3 polynomial kernel



Radial kernel



Summary

- Maximal Margin Classifier
  linear boundary
  separable cases

- Support Vector Classifier
  linear boundary
  separable or non-separable cases
  (A special case of SVM when linear kernel is used.)

- Support Vector Machine
  linear or nonlinear boundary
  separable or non-separable cases