Support Vector Machines

Wenjie Tu

University of Zurich

wenjie.tu@uzh.ch

March 26, 2021

Overview

Maximal Margin Classifier

Classification Using a Separating Hyperplane

Hyperplane

In a p-dimensional space, a *hyperplane* is a flat affine subspace of dimension $p - 1$.

Exmaples

- In two dimensions, a hyperplane is defined by

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

- In p dimensions, a hyperplane is defined by

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

Maximal Margin Classifier

Classification Using a Separating Hyperplane

Suppose that there exists a point $X = (X_1, X_2, \cdots, X_p)^T$ in the p dimensional space

- If X satisifes $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$, then X lies on the hyperplane.
- If X satisfies $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0$, then X lies on the one side of the hyperplane.
- If X satisfies $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0$, then X lies on the other side of the hyperplane.

Maximal Margin Classifier

Classification Using a Separating Hyperplane

In binary classification, we can label the observations from one class as $y_i = 1$ and those from the other class as $y_i = -1$.

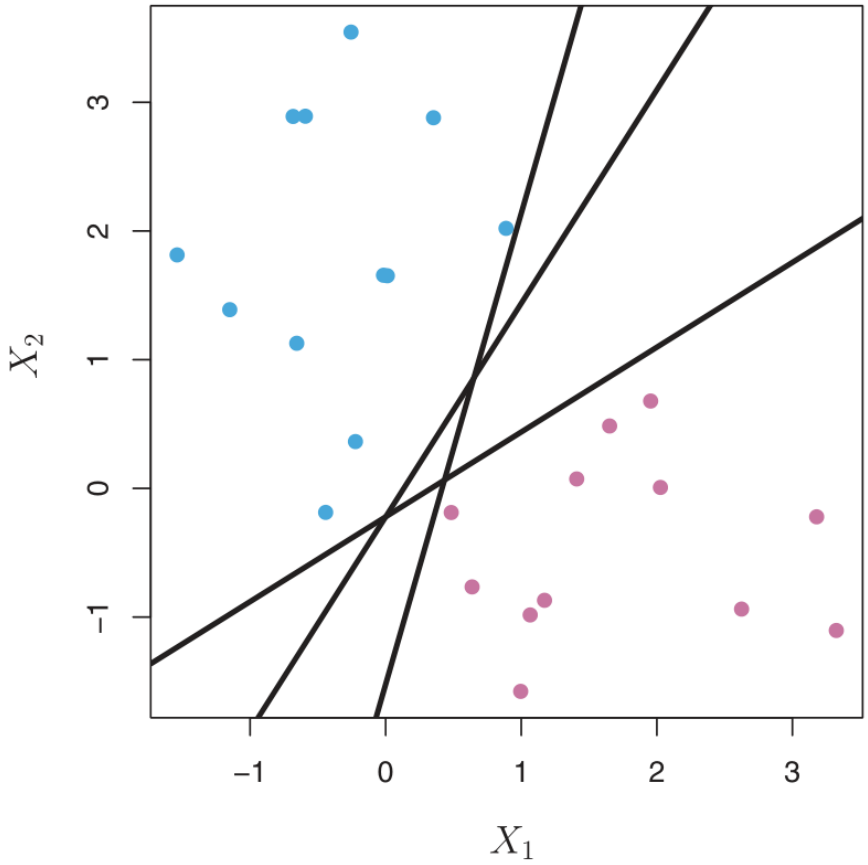
$$\begin{cases} \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} > 0 & y_i = 1 \\ \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} < 0 & y_i = -1 \end{cases}$$

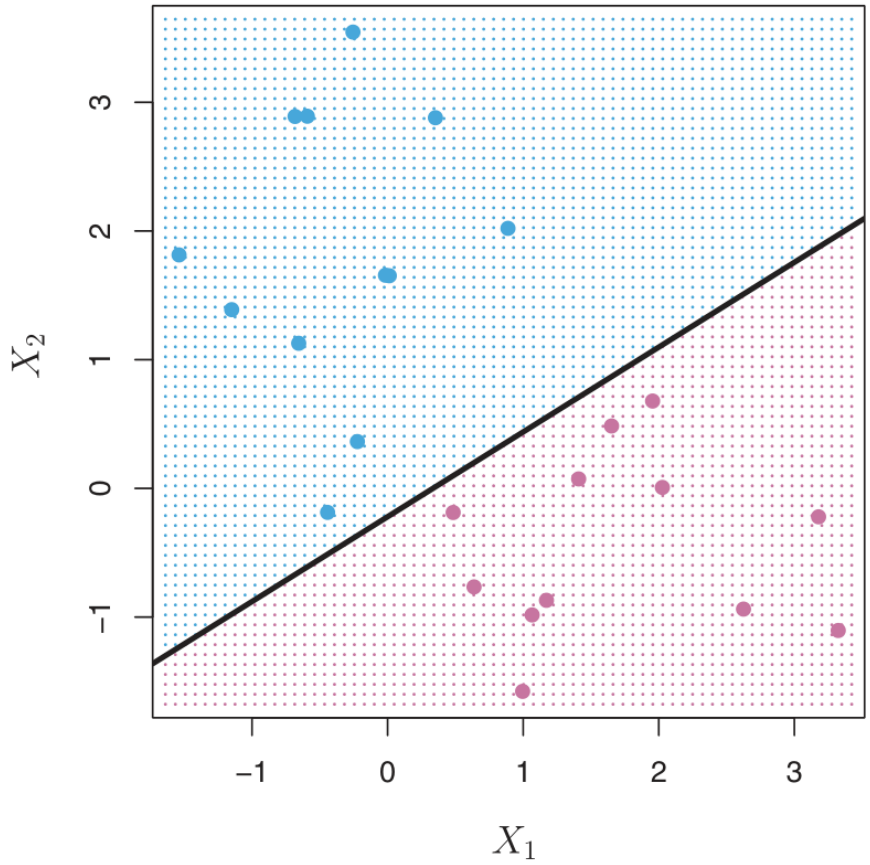
Equivalently,

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

Maximal Margin Classifier

Classification Using a Separating Hyperplane





Question

If our data can be perfectly separated using a hyperplane, then there will be an infinite number of such hyperplanes. Which hyperplane would be the most appropriate decision boundary?

Maximal Margin Classifier

Classification Using the Maximal Margin Classifier

Margin

The minimal distance from observations to a given separating hyperplane.

The Maximal Margin Hyperplane

The maximal margin hyperplane is the separating hyperplane for which the margin is maximized.

Support Vectors

Observations that are closest to the separating hyperplane are called support vectors.

Maximal Margin Classifier

Construction of the Maximal Margin Classifier

We are now constructing the maximal margin hyperplane based on a set of n training observations $x_1, \cdots, x_n \in \mathbb{R}^p$ and associated class labels $y_1, \cdots, y_n \in \{-1, 1\}$.

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

$$\text{s.t.} \begin{cases} \sum_{j=1}^{p} \beta_j^2 = 1 \\ y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M, \forall i = 1, \cdots, n \end{cases}$$

Note:
$\sum_{j=1}^{p} \beta_j^2 = 1$ is not really a constraint on the hyperplane. However, with this constraint, we can easily show that the perpendicular distance from the ith observation to the hyperplane is given by

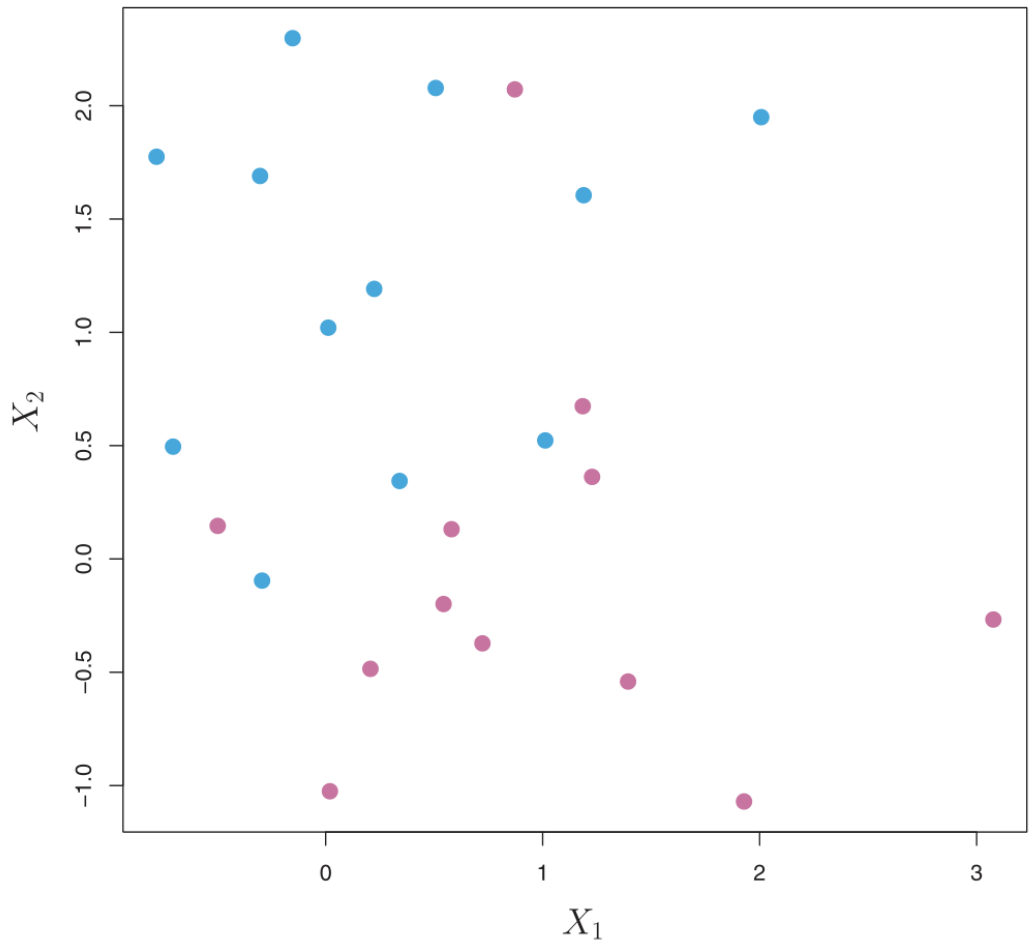
$$d = \frac{y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})}{\sqrt{\sum_{j=1}^{p} \beta_j^2}}$$

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

Maximal Margin Classifier

The Non-separable Case

In many cases, no separating hyperplane exists and there is no maximal margin classifier. The optimization problem has no solution with $M > 0$.



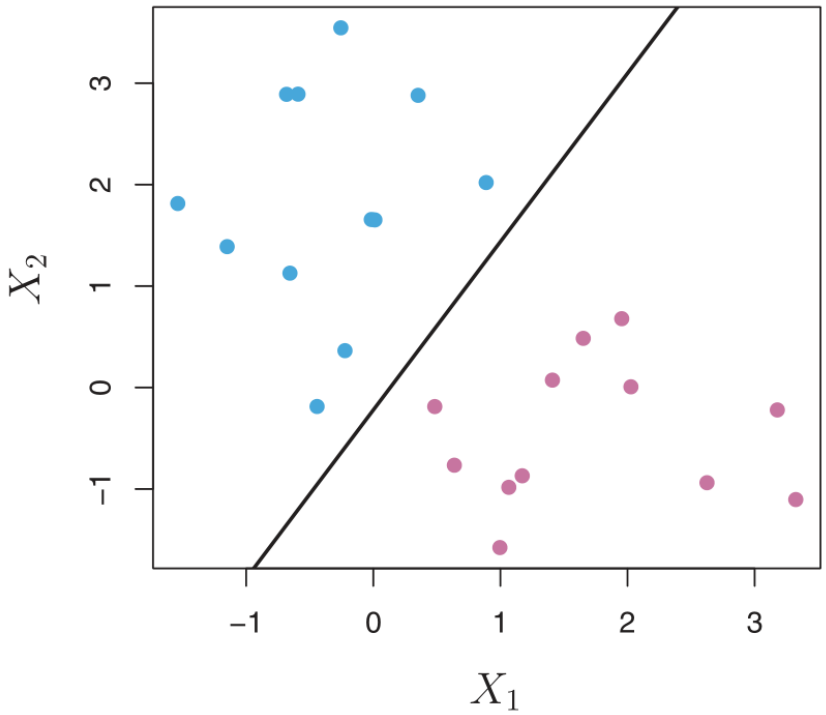
Maximal Margin Classifier

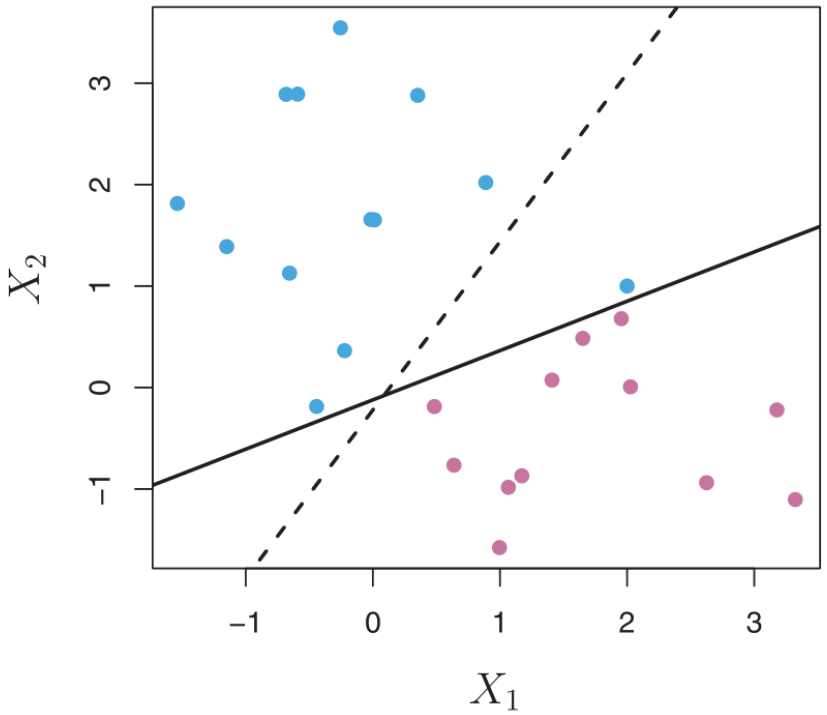
The *Maximal Margin Classifier* has some limitations:

- It is sensitive to outliers.
- It is more likely to overfit the training data.

Maximal Margin Classifier

In order to solve the problems, sometimes it could be worthwhile to misclassify a few training observations.





Support Vector Classifiers

Overview of the Support Vector Classifier

The Support Vector Classifier

The *Support Vector Classifier* is a *soft margin classifier* that allows for some misclassified observations in order to improve generalization ability.

The Support Vector Classifier ensures

- Greater robustness to individual observations
- Better classification of *most* of the training observations

Support Vector Classifiers

Classification Using the Support Vector Classifier

Support vector classifier comes down to an optimization problem

$$\max_{\beta_0,\beta_1,\cdots,\beta_p,\epsilon_1,\cdots,\epsilon_n} M$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^{p} \beta_j^2 = 1 \\ y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \\ \epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \end{cases}$$

- C is a nonnegative parameter
- M is the width of the margin
- $\epsilon_1, \cdots, \epsilon_n$ are slack variables

Support Vector Classifiers

Classification Using the Support Vector Classifier

Nonnegative tuning parameter

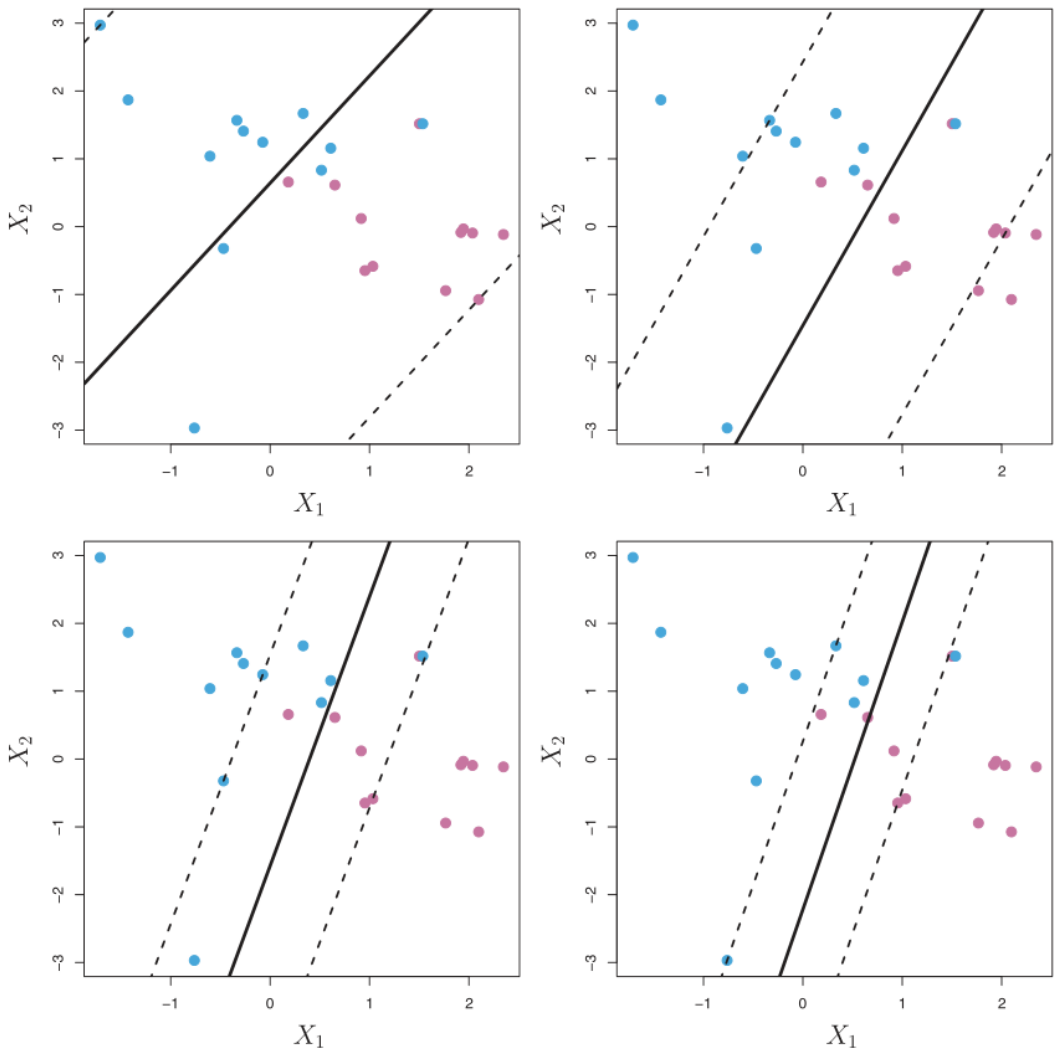
C bounds the sum of the ϵ_i's so it determines the number and severity of the violations to the margin that we will tolerate. We can think of C as a *budegt* for the amount that the margin can be violated by the n observations.

C controls the bias-variance trade-off of the support vector classifier.

- When C is smaller, we seek narrow margins that are rarely violated. In this case, the classifier may overfit the data and have low bias and high variance.
- When C is larger, the margin is wider and we allow more violations. In this case, the classifier may underfit the data and have high bias and low variance.

Support Vector Classifiers

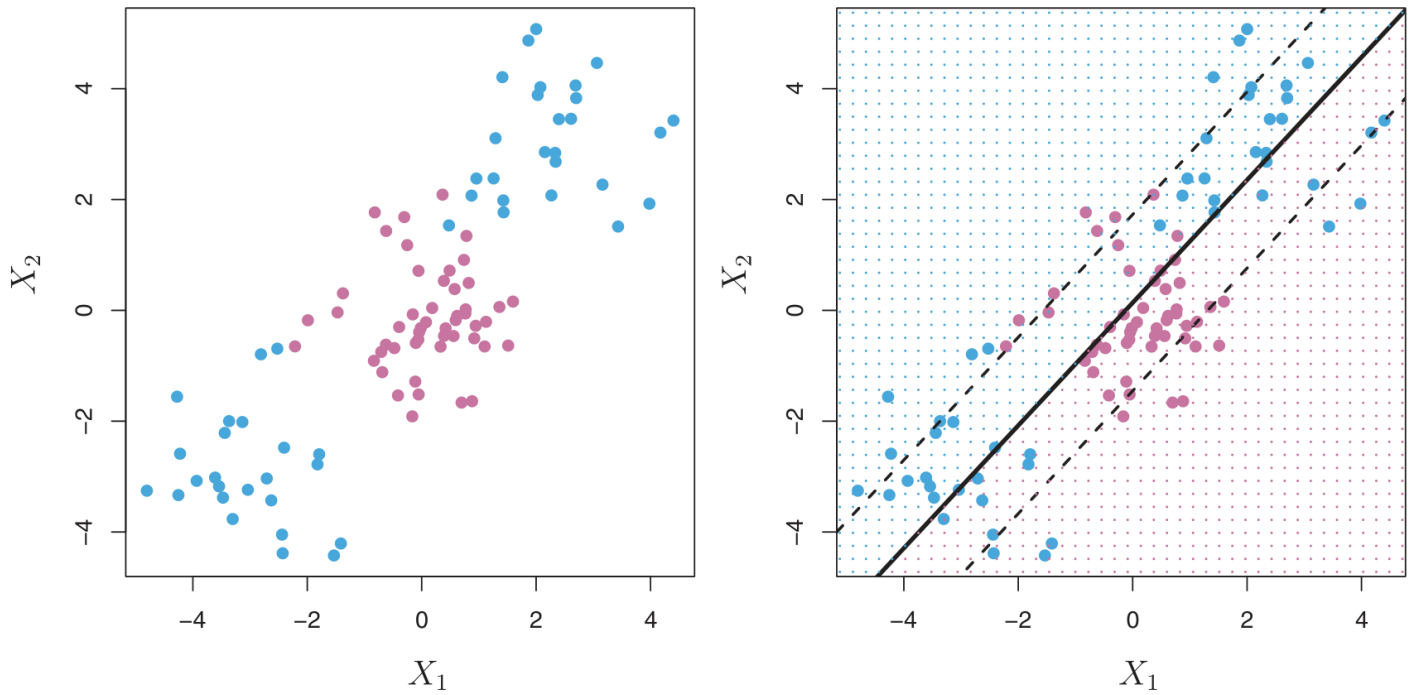
Classification Using the Support Vector Classifier



Support Vector Machines

Classification with Non-linear Decision Boundaries

What if the data is non-linearly separable? The *support vector classifier* performs poorly in this setting.



Support Vector Machines

Classification with Non-linear Decision Boundaries

Rather than fitting a support vector classifier using p features, we could instead fit a support vector classifier using $2p$ features

$$X_1, X_1^2, X_2, X_2^2, \cdots, X_p, X_p^2$$

Then the optimization problem would become

$$\max_{\beta_0, \beta_{11}, \beta_{12}, \cdots, \beta_{p1}, \beta_{p2}, \epsilon_1, \cdots, \epsilon_n} M$$

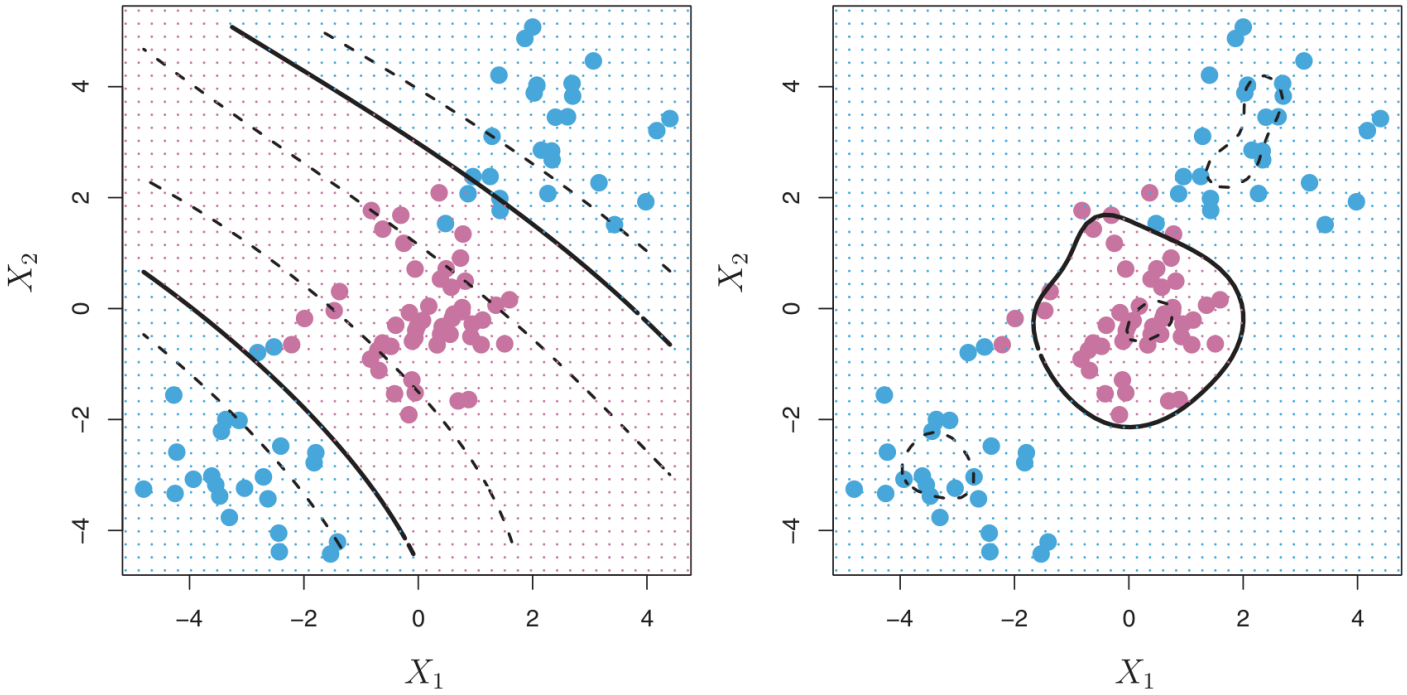
$$\text{s.t.} \begin{cases} \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1 \\ y_i(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2) \geq M(1 - \epsilon_i) \\ \sum_{i=1}^{n} \epsilon_i \leq C, \epsilon_i \geq 0 \end{cases}$$

Support Vector Machines

Classification Using the Suport Vector Machine

SVM

The *support vector machine* (SVM) is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using *kernels*.



Why are we interested in kernels?

Our goal is to enlarge the feature space so that the data can be linearly separated. Since the solution to the optimization problem only involves the *inner product* of the observations, we therefore want to find a more efficient way in computing the inner product between each pair of data in the feature space or each pair of its expansion. *Kernels* are such functions that can give us the inner product of two inputs without even knowing each input.

Support Vector Machines

Classification Using the Suport Vector Machine

Kernel

A *kernel* is a function that quantifies the similarity of two observations.

$$K(x_i, x_{i'})$$

- Linear kernel: $K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$.
- Polynomial kernel: $K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d$.
- Radial kernel: $K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$

SVMs with More than Two Classes

Question

The SVM can only deal with the binary classification. What if we have multiple classes? How can we extend SVMs to a more general setting?

Two most popular approaches:

- One-Versus-One Classification
- One-Versus-All Classification

SVMs with More than Two Classes

Suppose there are $K > 2$ classes

One-Versus-One Classification

A *one-versus-one* approach constructs $\binom{K}{2}$ SVMs, each of which compares a pair of classes.

- Training $\frac{K(K-1)}{2}$ classifiers.
- Each training procedure uses on average $\frac{2}{K}$ of the training data.

One-Versus-All Classification

A *one-versus-all* approach constructs K SVMs, each of which compares one of the K classes to the remaining $K-1$ classes.

- Training K classifiers.
- Each training procedure uses the entire training data.

Reference

G. James, D. Witten, R. Tibshirani

An Introduction to Statistical Learning with Application in R

Support Vector Machines 337-356

The End