Support Vector Machines

Maria Jose Medina

Universidad de Santiago de Chile

Outline

Introduction

- The support vector machine is a generalization of a simple and classifier called the **maximal margin classifier (MMC)**.

Introduction

- The support vector machine is a generalization of a simple and classifier called the **maximal margin classifier (MMC)**.
- Though it is elegant and simple, MMC cannot be applied to most data sets, since it requires that the classes be separable by a **linear boundary**.

Introduction

- The support vector machine is a generalization of a simple and classifier called the **maximal margin classifier (MMC)**.
- Though it is elegant and simple, MMC cannot be applied to most data sets, since it requires that the classes be separable by a **linear boundary**.
- We'll see the **support vector classifier (SVC)**, an extension of MMC that can be applied in a broader range of cases.

Introduction

- The support vector machine is a generalization of a simple and classifier called the **maximal margin classifier (MMC)**.
- Though it is elegant and simple, MMC cannot be applied to most data sets, since it requires that the classes be separable by a **linear boundary**.
- We'll see the **support vector classifier (SVC)**, an extension of MMC that can be applied in a broader range of cases.
- Then, we'll see the **support vector machine**, which is a further extension of the previous.

Introduction

- The support vector machine is a generalization of a simple and classifier called the **maximal margin classifier (MMC)**.
- Though it is elegant and simple, MMC cannot be applied to most data sets, since it requires that the classes be separable by a **linear boundary**.
- We'll see the **support vector classifier (SVC)**, an extension of MMC that can be applied in a broader range of cases.
- Then, we'll see the **support vector machine**, which is a further extension of the previous.

Table of Contents

Maximal Margin Classifier

Hyperplane

- In a p-dimensional space, a hyperplane is a flat affine subspace of hyperplane dimension $p-1$. It's defined as,

Maximal Margin Classifier

Hyperplane

- In a p-dimensional space, a hyperplane is a flat affine subspace of hyperplane dimension $p-1$. It's defined as,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{1}$$

Maximal Margin Classifier

Hyperplane

- In a p-dimensional space, a hyperplane is a flat affine subspace of hyperplane dimension $p - 1$. It's defined as,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{1}$$

- We can think of the hyperplane as dividing p-dimensional space into two halves:

Maximal Margin Classifier

Hyperplane

- In a p-dimensional space, a hyperplane is a flat affine subspace of hyperplane dimension $p-1$. It's defined as,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{1}$$

- We can think of the hyperplane as dividing p-dimensional space into two halves:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0 \tag{2}$$

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0 \tag{3}$$

Maximal Margin Classifier

Hyperplane

- In a p-dimensional space, a hyperplane is a flat affine subspace of hyperplane dimension $p-1$. It's defined as,

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0 \tag{1}$$

- We can think of the hyperplane as dividing p-dimensional space into two halves:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p > 0 \tag{2}$$
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p < 0 \tag{3}$$

One can determine on which side of the hyperplane a point lies by simply calculating the sign of the left hand side of (1).

Table of Contents

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Supposse that we have a $n \times p$ data matrix $\mathbf{X}$ that consists of n training observations in p-dimensional space,

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Supposse that we have a $n \times p$ data matrix **X** that consists of n training observations in p-dimensional space,

$$x_n = \begin{pmatrix} x_{n1} \\ \dots \\ x_{np} \end{pmatrix}. \tag{4}$$

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Supposse that we have a $n \times p$ data matrix **X** that consists of n training observations in p-dimensional space,

$$x_n = \begin{pmatrix} x_{n1} \\ \dots \\ x_{np} \end{pmatrix}. \tag{4}$$

- These observations fall into two classes: $y_n \in \{-1, 1\}$

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Supposse that we have a $n \times p$ data matrix $\mathbf{X}$ that consists of n training observations in p-dimensional space,

$$x_n = \begin{pmatrix} x_{n1} \\ \dots \\ x_{np} \end{pmatrix}. \tag{4}$$

- These observations fall into two classes: $y_n \in \{-1, 1\}$
- The goal is to develop a classifier based on x_n that will set the test data x^* into one category.

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Supposse that we have a $n \times p$ data matrix $\mathbf{X}$ that consists of n training observations in p-dimensional space,

$$x_n = \begin{pmatrix} x_{n1} \\ \dots \\ x_{np} \end{pmatrix}. \tag{4}$$

- These observations fall into two classes: $y_n \in \{-1, 1\}$
- The goal is to develop a classifier based on x_n that will set the test data x^* into one category.
- Suppose that it is possible to construct a hyperplane such that,

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Supposse that we have a $n \times p$ data matrix $\mathbf{X}$ that consists of n training observations in p-dimensional space,

$$x_n = \begin{pmatrix} x_{n1} \\ \dots \\ x_{np} \end{pmatrix}. \tag{4}$$

- These observations fall into two classes: $y_n \in \{-1, 1\}$
- The goal is to develop a classifier based on x_n that will set the test data x^* into one category.
- Suppose that it is possible to construct a hyperplane such that,

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} > 0 \text{ if } y_1 = 1 \tag{5}$$

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} < 0 \text{ if } y_1 = -1 \tag{6}$$

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Supposse that we have a $n \times p$ data matrix $\mathbf{X}$ that consists of n training observations in p-dimensional space,

$$x_n = \begin{pmatrix} x_{n1} \\ \dots \\ x_{np} \end{pmatrix}. \tag{4}$$

- These observations fall into two classes: $y_n \in \{-1, 1\}$
- The goal is to develop a classifier based on x_n that will set the test data x^* into one category.
- Suppose that it is possible to construct a hyperplane such that,

$$\beta_0 + \beta_1 x_{i1} + \dots + \beta_p X_{ip} > 0 \text{ if } y_1 = 1 \tag{5}$$

$$\beta_0 + \beta_1 x_{i1} + \dots + \beta_p X_{ip} < 0 \text{ if } y_1 = -1 \tag{6}$$

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Suppose that it is possible to construct a hyperplane such that,

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Suppose that it is possible to construct a hyperplane such that,

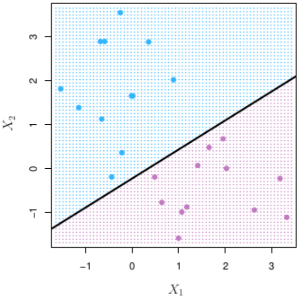
$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} > 0 \text{ if } y_i = 1$$
$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} < 0 \text{ if } y_i = -1$$

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Suppose that it is possible to construct a hyperplane such that,

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} > 0 \text{ if } y_i = 1$$
$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} < 0 \text{ if } y_i = -1$$

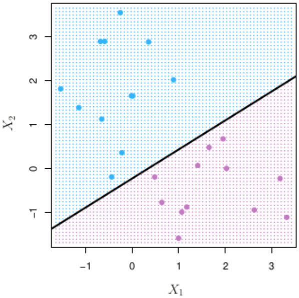


Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Suppose that it is possible to construct a hyperplane such that,

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} > 0 \text{ if } y_i = 1$$
$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} < 0 \text{ if } y_i = -1$$



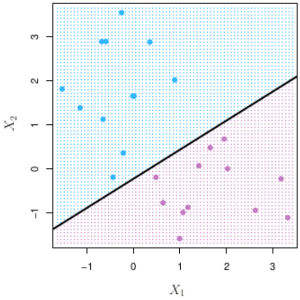
- If $f(x^*) > 0$, then $y^* \in 1$

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Suppose that it is possible to construct a hyperplane such that,

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} > 0 \text{ if } y_i = 1$$
$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} < 0 \text{ if } y_i = -1$$



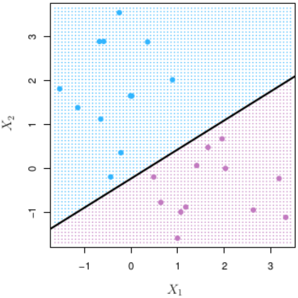
- If $f(x^*) > 0$, then $y^* \in 1$
- If $f(x^*) < 0$, then $y^* \in -1$

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Suppose that it is possible to construct a hyperplane such that,

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} > 0 \text{ if } y_i = 1$$
$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} < 0 \text{ if } y_i = -1$$



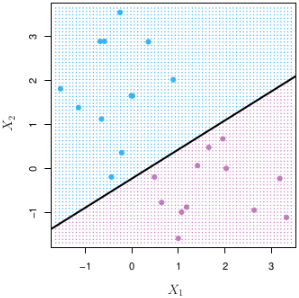
- If $f(x^*) > 0$, then $y^* \in 1$
- If $f(x^*) < 0$, then $y^* \in -1$
- If $\mid f(x^*) \mid$ is far from zero, then this means that x^* lies far from the hyperplane.

Maximal Margin Classifier

Classification Using a Separating Hyperplane

- Suppose that it is possible to construct a hyperplane such that,

$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} > 0 \text{ if } y_i = 1$$
$$\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip} < 0 \text{ if } y_i = -1$$



- If $f(x^*) > 0$, then $y^* \in 1$
- If $f(x^*) < 0$, then $y^* \in -1$
- If $\mid f(x^*) \mid$ is far from zero, then this means that x^* lies far from the hyperplane.
- If $\mid f(x^*) \mid$ is close to zero, then x^* is located near the hyperplane.

Table of Contents

Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.

Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- Which to choose?

Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- Which to choose?
- A natural choice is the **maximal margin hyperplane (MMH)**.

Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- Which to choose?
- A natural choice is the **maximal margin hyperplane (MMH)**.
- MMH is the farthest hyperplane from the training observations.

Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- Which to choose?
- A natural choice is the **maximal margin hyperplane (MMH)**.
- MMH is the farthest hyperplane from the training observations.
- To compute the farthest hyperplane, we follow the next steps:

Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- Which to choose?
- A natural choice is the **maximal margin hyperplane (MMH)**.
- MMH is the farthest hyperplane from the training observations.
- To compute the farthest hyperplane, we follow the next steps:
  - Compute the perpendicular distance from each training observation to a given separating hyperplane.

Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- Which to choose?
- A natural choice is the **maximal margin hyperplane (MMH)**.
- MMH is the farthest hyperplane from the training observations.
- To compute the farthest hyperplane, we follow the next steps:
  - Compute the perpendicular distance from each training observation to a given separating hyperplane.
  - We identify the **margin** as the smallest distance from step 1.

Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- Which to choose?
- A natural choice is the **maximal margin hyperplane (MMH)**.
- MMH is the farthest hyperplane from the training observations.
- To compute the farthest hyperplane, we follow the next steps:
  - Compute the perpendicular distance from each training observation to a given separating hyperplane.
  - We identify the **margin** as the smallest distance from step 1.
  - We identify the **MMH** as the hyperplane that correspond to the largest margin.

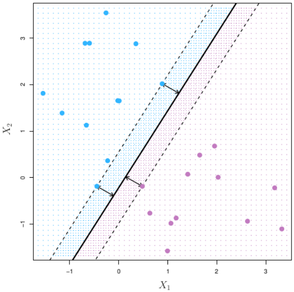
Maximal Margin Classifier

The Maximal Margin Classifier

- In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes.
- Which to choose?
- A natural choice is the **maximal margin hyperplane (MMH)**.
- MMH is the farthest hyperplane from the training observations.
- To compute the farthest hyperplane, we follow the next steps:
  - Compute the perpendicular distance from each training observation to a given separating hyperplane.
  - We identify the **margin** as the smallest distance from step 1.
  - We identify the **MMH** as the hyperplane that correspond to the largest margin.
- Then we classify a test observation based on which side of the maximal margin hyperplane it lies.

Maximal Margin Classifier

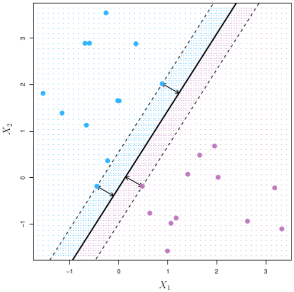
The Maximal Margin Classifier



- There are 3 equidistant observations from the MMH and lie along the dashed lines indicating the width of the margin.

Maximal Margin Classifier

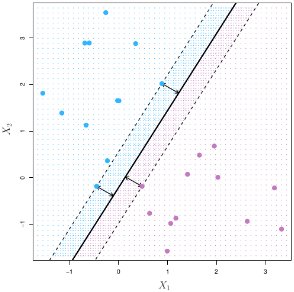
The Maximal Margin Classifier



- There are 3 equidistant observations from the MMH and lie along the dashed lines indicating the width of the margin.
- These are known as **support vectors**.

Maximal Margin Classifier

The Maximal Margin Classifier



- There are 3 equidistant observations from the MMH and lie along the dashed lines indicating the width of the margin.
- These are known as **support vectors**.
- They "support" the MMH: if they were moved slightly then the maximal margin hyperplane would move as well.

Table of Contents

Maximal Margin Classifier

Construction of the Maximal Margin Classifier

- Given n training observations $x_1, \cdots, x_n \in \mathbb{R}^p$ and classes $y_1, \cdots y_n \in \{-1, 1\}$

Maximal Margin Classifier

Construction of the Maximal Margin Classifier

- Given n training observations $x_1, \cdots, x_n \in \mathbb{R}^p$ and classes $y_1, \cdots y_n \in \{-1, 1\}$
- The maximal margin hyperplane is the solution to,

Maximal Margin Classifier

Construction of the Maximal Margin Classifier

- Given n training observations $x_1, \cdots, x_n \in \mathbb{R}^p$ and classes $y_1, \cdots y_n \in \{-1, 1\}$
- The maximal margin hyperplane is the solution to,

$$\max_{\beta_0, \cdots, \beta_p, M} M \tag{7}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{8}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \;\forall\, i = 1, \cdots, n. \tag{9}$$

Maximal Margin Classifier

Construction of the Maximal Margin Classifier

- Given n training observations $x_1, \cdots, x_n \in \mathbb{R}^p$ and classes $y_1, \cdots y_n \in \{-1, 1\}$
- The maximal margin hyperplane is the solution to,

$$\max_{\beta_0, \cdots, \beta_p, M} M \tag{7}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{8}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \ \forall\, i = 1, \cdots, n. \tag{9}$$

- Eq. (9) ensures that each observation will be on the correct side of the hyperplane, provided that M is positive.

Maximal Margin Classifier

Construction of the Maximal Margin Classifier

- Given n training observations $x_1, \cdots, x_n \in \mathbb{R}^p$ and classes $y_1, \cdots y_n \in \{-1, 1\}$
- The maximal margin hyperplane is the solution to,

$$\max_{\beta_0, \cdots, \beta_p, M} M \tag{7}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{8}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \ \forall\, i = 1, \cdots, n. \tag{9}$$

- Eq. (9) ensures that each observation will be on the correct side of the hyperplane, provided that M is positive.
- The l.h.s. of equation (9) is the distance from the ith observation to the hyperplane.

Maximal Margin Classifier

Construction of the Maximal Margin Classifier

- Given n training observations $x_1, \cdots, x_n \in \mathbb{R}^p$ and classes $y_1, \cdots y_n \in \{-1, 1\}$
- The maximal margin hyperplane is the solution to,

$$\max_{\beta_0, \cdots, \beta_p, M} M \tag{7}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{8}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \; \forall\, i = 1, \cdots, n. \tag{9}$$

- Eq. (9) ensures that each observation will be on the correct side of the hyperplane, provided that M is positive.
- The l.h.s. of equation (9) is the distance from the ith observation to the hyperplane.
- So, eq. (9) ensures that each observation is at least a distance M from the hyperplane.

Table of Contents

Maximal Margin Classifier

The Non-separable Case

- The maximal margin classifier is a very natural way to perform classification, if a separating hyperplane exists.

Maximal Margin Classifier

The Non-separable Case

- The maximal margin classifier is a very natural way to perform classification, if a separating hyperplane exists.
- However, in many cases no separating hyperplane exists.

Maximal Margin Classifier

The Non-separable Case

- The maximal margin classifier is a very natural way to perform classification, if a separating hyperplane exists.
- However, in many cases no separating hyperplane exists.
- As we will see in the next section, we can extend the concept of a separating hyperplane in order to develop a hyperplane that almost separates the classes, using a so-called **soft margin**.

Maximal Margin Classifier

The Non-separable Case

- The maximal margin classifier is a very natural way to perform classification, if a separating hyperplane exists.
- However, in many cases no separating hyperplane exists.
- As we will see in the next section, we can extend the concept of a separating hyperplane in order to develop a hyperplane that almost separates the classes, using a so-called **soft margin**.
- The generalization of the maximal margin classifier to the non-separable case is known as the **support vector classifier**.

Support Vector Classifiers (SVC)

- In **SVC** we consider a hyperplane that does not perfectly separate the two classes, in the interest of:

Support Vector Classifiers (SVC)

- In **SVC** we consider a hyperplane that does not perfectly separate the two classes, in the interest of:
  - ✓ Greater robustness to individual observations.

Support Vector Classifiers (SVC)

- In **SVC** we consider a hyperplane that does not perfectly separate the two classes, in the interest of:
  - ✓ Greater robustness to individual observations.
  - ✓ Better classification of most of the training observations.

Support Vector Classifiers (SVC)

- In **SVC** we consider a hyperplane that does not perfectly separate the two classes, in the interest of:
  - ✓ Greater robustness to individual observations.
  - ✓ Better classification of most of the training observations.
- We do not seek the largest possible margin so that **every observation** is classified perfectly.

Support Vector Classifiers (SVC)

- In **SVC** we consider a hyperplane that does not perfectly separate the two classes, in the interest of:
  - ✓ Greater robustness to individual observations.
  - ✓ Better classification of most of the training observations.
- We do not seek the largest possible margin so that **every observation** is classified perfectly.
- We allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.

Support Vector Classifiers (SVC)

- In **SVC** we consider a hyperplane that does not perfectly separate the two classes, in the interest of:
  - ✓ Greater robustness to individual observations.
  - ✓ Better classification of most of the training observations.
- We do not seek the largest possible margin so that **every observation** is classified perfectly.
- We allow some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane.

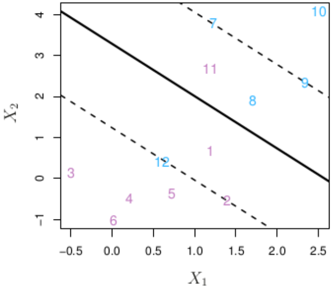


Table of Contents

Support Vector Machines

Classification with Non-Linear Decision Boundaries

- When there is a non-linear relationship between the predictors and the outcome, we enlarging the feature space using quadratic, cubic, and even higher-order polynomial functions of the predictors.

Support Vector Machines

Classification with Non-Linear Decision Boundaries

- When there is a non-linear relationship between the predictors and the outcome, we enlarging the feature space using quadratic, cubic, and even higher-order polynomial functions of the predictors.
- We fit a **support vector classifier** using $2p$ features,

Support Vector Machines

Classification with Non-Linear Decision Boundaries

- When there is a non-linear relationship between the predictors and the outcome, we enlarging the feature space using quadratic, cubic, and even higher-order polynomial functions of the predictors.
- We fit a **support vector classifier** using $2p$ features,

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

Support Vector Machines

Classification with Non-Linear Decision Boundaries

- When there is a non-linear relationship between the predictors and the outcome, we enlarging the feature space using quadratic, cubic, and even higher-order polynomial functions of the predictors.
- We fit a **support vector classifier** using $2p$ features,

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

- Then, the maximization problem is

Support Vector Machines

Classification with Non-Linear Decision Boundaries

- When there is a non-linear relationship between the predictors and the outcome, we enlarging the feature space using quadratic, cubic, and even higher-order polynomial functions of the predictors.
- We fit a **support vector classifier** using $2p$ features,

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

- Then, the maximization problem is

$$\max_{\beta_0, \cdots \beta_{p1}, \beta_{p2}, \epsilon_1, \cdots \epsilon_n, M} M \tag{10}$$

$$\text{s.t. } y_i \left(\beta_0 \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2 \right) \geq M(1 - \epsilon_i) \tag{11}$$

$$\epsilon_i \leq C, \epsilon_i \geq 0, \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1. \tag{12}$$

Support Vector Machines

Classification with Non-Linear Decision Boundaries

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

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

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

Support Vector Machines

Classification with Non-Linear Decision Boundaries

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

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

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

- M is the width of the margin.

Support Vector Machines

Classification with Non-Linear Decision Boundaries

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

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

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

- M is the width of the margin.
- ϵ_i are *slack variables*: allow observations to be on the wrong side of the margin or the hyperplane

Support Vector Machines

Classification with Non-Linear Decision Boundaries

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

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

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

- M is the width of the margin.
- ϵ_i are *slack variables*: allow observations to be on the wrong side of the margin or the hyperplane
  1. If $\epsilon_i = 0$ then the ith observation is on the correct side of the margin.

Support Vector Machines

Classification with Non-Linear Decision Boundaries

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

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

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

- M is the width of the margin.
- ϵ_i are *slack variables*: allow observations to be on the wrong side of the margin or the hyperplane
  1. If $\epsilon_i = 0$ then the ith observation is on the correct side of the margin.
  2. If $\epsilon_i > 0$ then the ith observation is on the wrong side of the margin

Support Vector Machines

Classification with Non-Linear Decision Boundaries

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

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

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

- M is the width of the margin.
- ϵ_i are *slack variables*: allow observations to be on the wrong side of the margin or the hyperplane
  1. If $\epsilon_i = 0$ then the ith observation is on the correct side of the margin.
  2. If $\epsilon_i > 0$ then the ith observation is on the wrong side of the margin
  3. If $\epsilon_i > 1$ then it is on the wrong side of the hyperplane.

Support Vector Machines

Classification with Non-Linear Decision Boundaries

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

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

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

- M is the width of the margin.
- ϵ_i are *slack variables*: allow observations to be on the wrong side of the margin or the hyperplane
  1. If $\epsilon_i = 0$ then the ith observation is on the correct side of the margin.
  2. If $\epsilon_i > 0$ then the ith observation is on the wrong side of the margin
  3. If $\epsilon_i > 1$ then it is on the wrong side of the hyperplane.
- C is a tuning parameter: bounds the sum of the ϵ_i's and it's chosen by Cross-validation.

Table of Contents

Support Vector Machines

The Support Vector Machine (SVM)

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

Support Vector Machines

The Support Vector Machine (SVM)

- **SVM** is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**.
- A **kernel** is a function that quantifies the similarity of two observations and it's a generalization of the inner product of the observations.

Support Vector Machines

The Support Vector Machine (SVM)

- **SVM** is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**.
- A **kernel** is a function that quantifies the similarity of two observations and it's a generalization of the inner product of the observations.

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

Support Vector Machines

The Support Vector Machine (SVM)

- **SVM** is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**.
- A **kernel** is a function that quantifies the similarity of two observations and it's a generalization of the inner product of the observations.

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

- In fact, it can be shown that the linear support vector classifier can be represented as

Support Vector Machines

The Support Vector Machine (SVM)

- **SVM** is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**.
- A **kernel** is a function that quantifies the similarity of two observations and it's a generalization of the inner product of the observations.

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

- In fact, it can be shown that the linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle \tag{13}$$

Support Vector Machines

The Support Vector Machine (SVM)

- **SVM** is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**.
- A **kernel** is a function that quantifies the similarity of two observations and it's a generalization of the inner product of the observations.

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

- In fact, it can be shown that the linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle \tag{13}$$

where,

Support Vector Machines

The Support Vector Machine (SVM)

- **SVM** is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**.
- A **kernel** is a function that quantifies the similarity of two observations and it's a generalization of the inner product of the observations.

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

- In fact, it can be shown that the linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle \tag{13}$$

where,

1. x is the new point and x_i are the training points.

Support Vector Machines

The Support Vector Machine (SVM)

- **SVM** is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using **kernels**.
- A **kernel** is a function that quantifies the similarity of two observations and it's a generalization of the inner product of the observations.

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

- In fact, it can be shown that the linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle \tag{13}$$

where,

1. x is the new point and x_i are the training points.
2. α_i is the coefficient of the supports vectors.
3. S is the collection of indices of the support points.

Support Vector Machines

The Support Vector Machine (SVM)

- If we simply use,

Support Vector Machines

The Support Vector Machine (SVM)

- If we simply use,

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} \tag{14}$$

Support Vector Machines

The Support Vector Machine (SVM)

- If we simply use,

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} \tag{14}$$

in eq. (13) the result would just give us back the support vector classifier.

Support Vector Machines

The Support Vector Machine (SVM)

- If we simply use,

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij}x_{i'j} \tag{14}$$

in eq. (13) the result would just give us back the support vector classifier.

- Then, eq. (14) is known as a **linear kernel** because the support vector classifier is linear in the features.
- But one could instead choose another form for the kernels

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

- It takes the form,

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

- It takes the form,

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d \tag{15}$$

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

- It takes the form,

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d \tag{15}$$

where $d > 0$

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

- It takes the form,

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d \tag{15}$$

  where $d > 0$

- Using such a kernel with $d > 1$, leads to a much more flexible decision boundary.

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

- It takes the form,

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d \tag{15}$$

  where $d > 0$

- Using such a kernel with $d > 1$, leads to a much more flexible decision boundary.
- It essentially amounts to fitting a SVC in a higher-dimensional space, rather than in the original feature space.

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

- It takes the form,

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d \tag{15}$$

  where $d > 0$
- Using such a kernel with $d > 1$, leads to a much more flexible decision boundary.
- It essentially amounts to fitting a SVC in a higher-dimensional space, rather than in the original feature space.
- In this case the function has the form

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

- It takes the form,

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij}x_{i'j})^d \tag{15}$$

  where $d > 0$
- Using such a kernel with $d > 1$, leads to a much more flexible decision boundary.
- It essentially amounts to fitting a SVC in a higher-dimensional space, rather than in the original feature space.
- In this case the function has the form

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i). \tag{16}$$

Support Vector Machines

The Support Vector Machine (SVM)

Polynomial kernel

- It takes the form,

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij}x_{i'j})^d \tag{15}$$

  where $d > 0$

- Using such a kernel with $d > 1$, leads to a much more flexible decision boundary.
- It essentially amounts to fitting a SVC in a higher-dimensional space, rather than in the original feature space.
- In this case the function has the form

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i). \tag{16}$$

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2) \ \gamma > 0. \tag{17}$$

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2)\ \gamma > 0. \tag{17}$$

- Then,

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2)\ \gamma > 0. \tag{17}$$

- Then,

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

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p}(x_{ij} - x_{i'j'})^2)\ \gamma > 0. \tag{17}$$

- Then,

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

- If x^* is far from x_i in terms of Enclidean distance, then

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2)\ \gamma > 0. \tag{17}$$

- Then,

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

- If x^* is far from x_i in terms of Enclidean distance, then
  1. $(x_{ij} - x_{i'j'})^2$ will be large,

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2) \ \gamma > 0. \tag{17}$$

- Then,

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

- If x^* is far from x_i in terms of Enclidean distance, then
  1. $(x_{ij} - x_{i'j'})^2$ will be large,
  2. $K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2)$ will be tiny.

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2) \ \gamma > 0. \tag{17}$$

- Then,

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

- If x^* is far from x_i in terms of Enclidean distance, then
  1. $(x_{ij} - x_{i'j'})^2$ will be large,
  2. $K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2)$ will be tiny.
  3. x_i will play virtually no role in $f(x^*)$

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2) \ \gamma > 0. \tag{17}$$

- Then,

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

- If x^* is far from x_i in terms of Enclidean distance, then
  1. $(x_{ij} - x_{i'j'})^2$ will be large,
  2. $K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j'})^2)$ will be tiny.
  3. x_i will play virtually no role in $f(x^*)$
- In other words, radial kernel has very local behavior.

Support Vector Machines

The Support Vector Machine (SVM)

Radial kernel

- It takes the form,

$$K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p}(x_{ij} - x_{i'j'})^2)\ \gamma > 0. \tag{17}$$

- Then,

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

- If x^* is far from x_i in terms of Enclidean distance, then
  1. $(x_{ij} - x_{i'j'})^2$ will be large,
  2. $K(x_i, x_{i'}) = exp(-\gamma \sum_{j=1}^{p}(x_{ij} - x_{i'j'})^2)$ will be tiny.
  3. x_i will play virtually no role in $f(x^*)$
- In other words, radial kernel has very local behavior.
- Only nearby training observations have an effect on the class label of a test observation.

Table of Contents

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
  1. Construct $K(K-1)/2$ SVMs.

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
    1. Construct $K(K-1)/2$ SVMs.
    2. Compare each SVM with a pair of classes.

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
  1. Construct $K(K-1)/2$ SVMs.
  2. Compare each SVM with a pair of classes.
  3. For example, one SVM might compare the kth class $(+1)$ to the k'th class (-1).

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
  1. Construct $K(K-1)/2$ SVMs.
  2. Compare each SVM with a pair of classes.
  3. For example, one SVM might compare the kth class $(+1)$ to the k'th class (-1).
  4. We count the number of times that x^* is assigned to each of the K classes.

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
  1. Construct $K(K-1)/2$ SVMs.
  2. Compare each SVM with a pair of classes.
  3. For example, one SVM might compare the kth class $(+1)$ to the k'th class (-1).
  4. We count the number of times that x^* is assigned to each of the K classes.
  5. We assign x^* to the most frequent class.

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
  1. Construct $K(K-1)/2$ SVMs.
  2. Compare each SVM with a pair of classes.
  3. For example, one SVM might compare the kth class $(+1)$ to the k'th class (-1).
  4. We count the number of times that x^* is assigned to each of the K classes.
  5. We assign x^* to the most frequent class.
- **One-Versus-All Classification**
  1. We fit K SVMs, each time comparing one of the K classes to the remaining $K-1$ classes.

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
  1. Construct $K(K-1)/2$ SVMs.
  2. Compare each SVM with a pair of classes.
  3. For example, one SVM might compare the kth class $(+1)$ to the k'th class (-1).
  4. We count the number of times that x^* is assigned to each of the K classes.
  5. We assign x^* to the most frequent class.
- **One-Versus-All Classification**
  1. We fit K SVMs, each time comparing one of the K classes to the remaining $K-1$ classes.
  2. Let $\beta_{0k}, \cdots, \beta_{pk}$ denote the parameters that result from fitting an SVM comparing the kth class $(+1)$ to the others (-1).

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
  1. Construct $K(K-1)/2$ SVMs.
  2. Compare each SVM with a pair of classes.
  3. For example, one SVM might compare the kth class $(+1)$ to the k'th class (-1).
  4. We count the number of times that x^* is assigned to each of the K classes.
  5. We assign x^* to the most frequent class.
- **One-Versus-All Classification**
  1. We fit K SVMs, each time comparing one of the K classes to the remaining $K-1$ classes.
  2. Let $\beta_{0k}, \cdots, \beta_{pk}$ denote the parameters that result from fitting an SVM comparing the kth class $(+1)$ to the others (-1).
  3. We assign to x^* to the class for which $\beta_{0k}, \cdots, \beta_{pk} x_p^*$ is largest.

Support Vector Machines

SVMs with More than Two Classes

Suppose that x^* is a test observation. To perform SVM for $K > 2$ classes there are two approaches to follow:

- **One-Versus-One Classification**
  1. Construct $K(K-1)/2$ SVMs.
  2. Compare each SVM with a pair of classes.
  3. For example, one SVM might compare the kth class $(+1)$ to the k'th class (-1).
  4. We count the number of times that x^* is assigned to each of the K classes.
  5. We assign x^* to the most frequent class.
- **One-Versus-All Classification**
  1. We fit K SVMs, each time comparing one of the K classes to the remaining $K-1$ classes.
  2. Let $\beta_{0k}, \cdots, \beta_{pk}$ denote the parameters that result from fitting an SVM comparing the kth class $(+1)$ to the others (-1).
  3. We assign to x^* to the class for which $\beta_{0k}, \cdots, \beta_{pk}x_p^*$ is largest.
  4. The previous amount is the level of level of confidence that x^* belongs to the kth class rather than $(k-1)$.

Thank you!

Any question?