Predictive Modeling

Support Vector Classifiers

Mirko Birbaumer

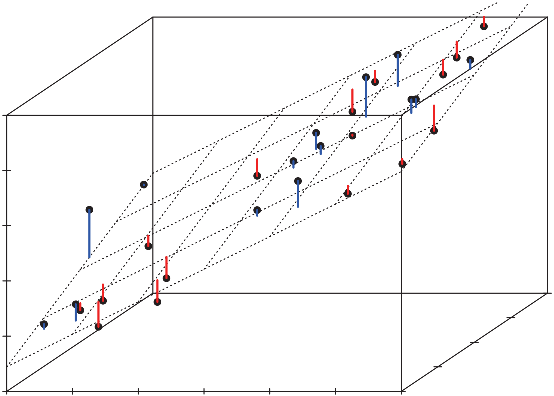
HSLU T&A

Support Vector Machines

- **Support vector machine** (SVM) was developed in the computer science community in the 1990s is a generalization of a simple and intuitive classifier called the **maximal margin classifier**
- **Support vector classifier**, an extension of the maximal margin classifier that can be applied in a broader range of cases
- **Support vector machine** is a further extension of the support vector classifier in order to accommodate non-linear class boundaries
- SVMs have been shown to perform well in a variety of settings, and are often considered one of the best „out of the box" classifiers

Maximal Margin Classifier - What Is a Hyperplane?

- A **hyperplane** is a generalized plane
- A plane separates the space in regions „above" and „below"
- Each point in space lies either above, below, or on the plane



Hyperplane

- In p-dimensional space, a **hyperplane** is a flat affine subspace of dimension $p - 1$
- In two dimensions, a hyperplane is a one-dimensional subspace – in other words: a **straight line**
- In three dimensions, a **hyperplane** is a flat two-dimensional subspace, i.e. a plane

Hyperplane

- In two dimensions, a **hyperplane** is defined by the equation

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

for parameters β_0, β_1 and β_2

- When we say that (1) „defines" the hyperplane, we mean that any $X = (X_1, X_2)^T$ for which (1) holds is a **point** on the hyperplane

- Note that (1) is simply the equation of a straight line

Hyperplane

- Generalization to p-dimensional space:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0 \tag{2}$$

- Equation (2) defines a $(p-1)$-dimensional hyperplane

- If a point $X = (X_1, X_2, \ldots, X_p)^T$ in p-dimensional space (i.e., a vector of length p) satisfies (2), then X lies on the hyperplane

Hyperplane

- Suppose that X satisfy

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

then this tells us that X lies to **one side** of the hyperplane

- If

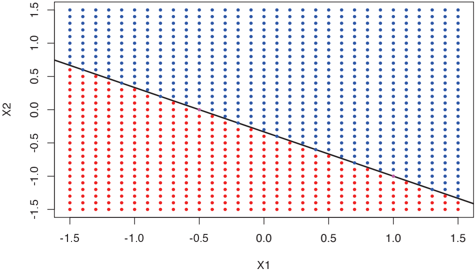
$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0$$

then X lies on the **other side** of the hyperplane.

Example : Hyperplane

- Consider the following hyperplane in two-dimensional space:

$$1 + 2X_1 + 3X_2 = 0$$



- **Red points** below the hyperplane satisfy

$$1 + 2X_1 + 3X_2 < 0$$

- **Blue points** above the hyperplane satisfy

$$1 + 2X_1 + 3X_2 > 0$$

Classification Using a Separating Hyperplane

- $n \times p$ data matrix $\mathbf{X}$ consisting of n training observations in p-dimensional space $\mathbb{R}^p$,

$$x_1 = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1p} \end{pmatrix} \quad x_2 = \begin{pmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2p} \end{pmatrix} \quad \ldots \quad x_n = \begin{pmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{np} \end{pmatrix}$$

  That is,

$$\mathbf{X} = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix}$$

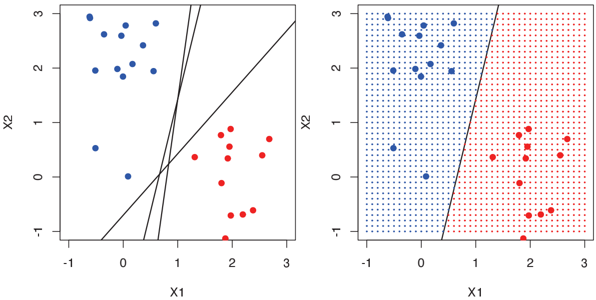
- Assume that these observations fall into two classes – that is,

$$y_1, y_2, \ldots, y_n \in \{-1, 1\}$$

Classification Using a Separating Hyperplane

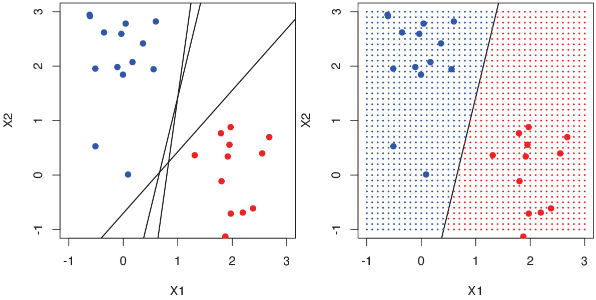
- We also have a **test observation** : a p-vector of observed features $x^* = (x_1^*, x_2^*, \ldots, x_p^*)^T$
- **Our goal** : develop a classifier based on the training data that will correctly classify the test observation using its feature measurements
- New approach that is based upon the concept of a **separating hyperplane**

Example : Separating Hyperplane



- *Left-hand panel :* three separating hyperplanes, out of many possible
- *Right-hand panel :* one of these separating hyperplanes and the decision rule made by a classifier based on this separating hyperplane

Example : Separating Hyperplane



- Test observation that falls in the **blue** portion of the grid will be assigned to the blue class
- Test observation that falls into the **red** portion of the grid will be assigned to the red class

Example : Separating Hyperplane

- Label the observations from the **blue class** as $y_i = 1$ and those from the **purple class** as $y_i = -1$

- **Separating hyperplane** has the property that

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0 \quad \text{if} \quad y_i = 1$$

  and

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0 \quad \text{if} \quad y_i = -1$$

- Equivalently, a **separating hyperplane** has the property that

$$y_i \cdot (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) > 0 \quad \text{for all} \quad i = 1, \ldots n$$

Separating Hyperplane

- We classify the **test observation** x^* based on the *sign* of

$$f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \ldots + \beta_p x_p^*$$

- If $f(x^*)$ is **positive**, then we assign the test observation to class 1

- If $f(x^*)$ is **negative**, then we assign it to class -1

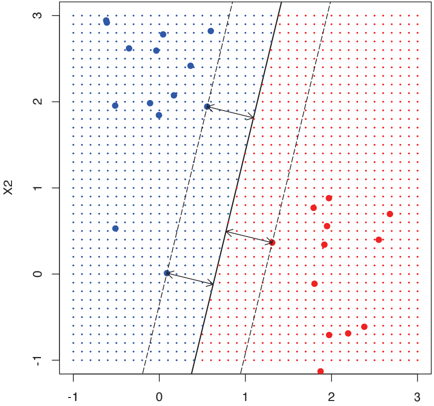
- If $f(x^*)$ is **far** from zero, then this means that x^* lies far from the hyperplane, and so we can be **confident** about our class assignment for x^*

Maximal Margin Classifier

- **Problem** of separating hyperplanes : there will in fact exist an infinite number of such hyperplanes
- We must have a reasonable way to decide **which** of the infinite possible separating hyperplanes to use
- Natural choice : **maximal margin hyperplane** is the separating hyperplane that is farthest from the training observations
- **Maximal margin hyperplane** is the separating hyperplane for which the margin is largest – that is, it is the hyperplane that has the *farthest minimum distance* to the training observations

Maximal Margin Classifier

- In a sense, the **maximal margin hyperplane** represents the mid-line of the widest „slab" we can insert between the two classes
- Three training observations are equidistant from the maximal margin hyperplane and lie along the dashed lines ⟶ **support vectors**



Support Vectors

- **Support vectors** are vectors in p-dimensional space : they **support** the maximal margin hyperplane in the sense that if these points were moved slightly then the maximal margin hyperplane would move as well

- Maximal margin hyperplane depends directly on the support vectors, but not on the other observations

- Check example 1.4 of the Support Vector Machines chapter

Construction of the Maximal Margin Classifier

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

- Briefly, the **maximal margin hyperplane** is the solution to the following *optimization problem*

$$\max_{\beta_0,\beta_1,\ldots,\beta_p,M} M \tag{3}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1 \tag{4}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M \quad \text{for} \quad i = 1, 2, \ldots, n \tag{5}$$

Construction of the Maximal Margin Classifier

- Actually, for each observation to be on the correct side of the hyperplane we would simply need

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

- Constraint

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

in fact requires that each observation be **on the correct side of the hyperplane**, with some cushion given by M

Construction of the Maximal Margin Classifier

- Coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are not uniquely defined by the hyperplane. I.e, the two equations

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = 0 \quad \text{and}$$
$$k\beta_0 + k\beta_1 x_1 + k\beta_2 x_2 + \cdots + k\beta_p x_p = 0$$

  define the same hyperplane, provided that $k \neq 0$.

- Geometrically, the vector $(\beta_1, \ldots, \beta_p)^T$ is **perpendicular** to the hyperplane, and the constraint

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

  restricts this normal vector to unit length

Construction of the Maximal Margin Classifier

- It can be shown that

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

  is the **perpendicular distance** from the *i*-th observation to the hyperplane

- Thus, constraints (4) and (5) ensure that each observation is **at least a distance** M on the correct side from the hyperplane

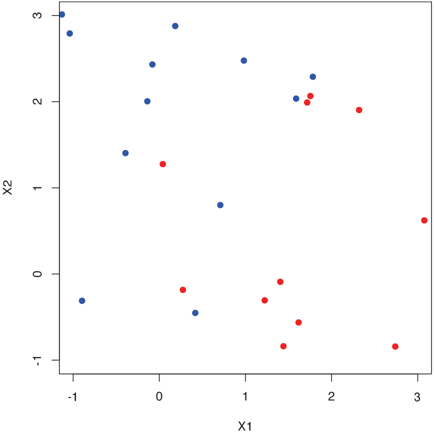
- Hence, M represents the **margin of our hyperplane**

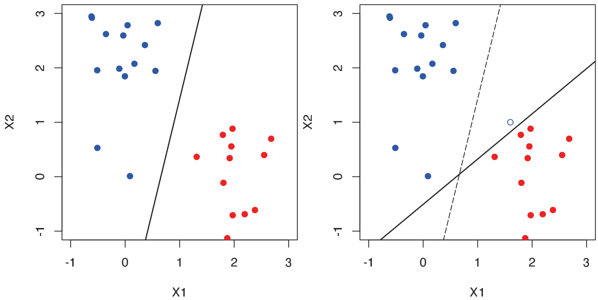
- The optimization problem chooses the coefficients β_i to **maximize** M.

- This is exactly the definition of the maximal margin hyperplane!

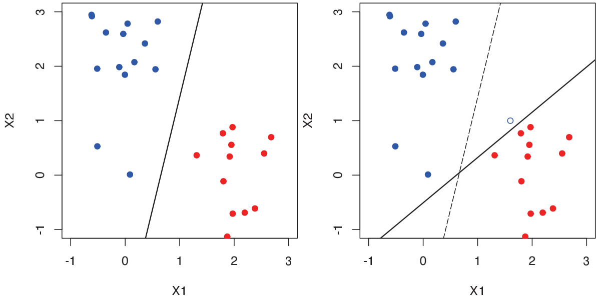
Support Vector Classifiers

- Observations belonging to two classes are not necessarily separable by a hyperplane



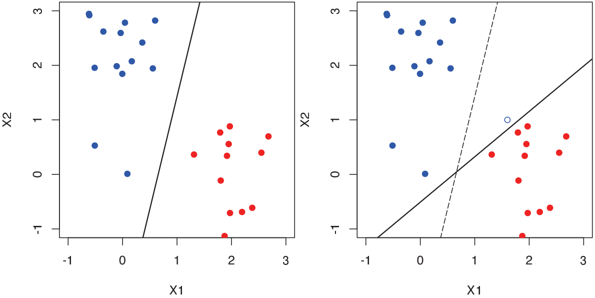


- **Wanted** : Classifier based on a hyperplane that does *not* perfectly separate the two classes, in the interest of
  - greater robustness to individual observations, and
  - better classification of *most* of the training observations.



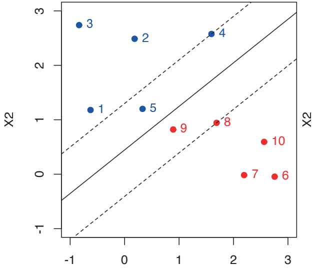
- It could be worthwhile to misclassify **a few** training observations in order to do a better job in classifying the **majority of** observations

Support Vector Classifier



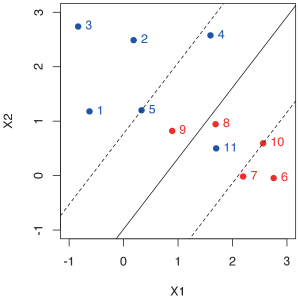
- It could be worthwhile to misclassify **a few** training observations in order to do a better job in classifying the **majority of** observations

Support Vector Classifier (Soft Margin Classifier)



- Most of the observations are on the **correct** side of the margin : blue points 1, 2, 3, 4 as well as the red points 6, 7, 8, and 10
- A small subset of the observations are on the **wrong** side of the margin: only the points 5 and 9
- All points are still on the right side of the hyperplane.

Support Vector Classifier (Soft Margin Classifier)



- Blue observation 11 has been added
- Hyperplane has changed: two observations 9 and 11 are now on the **wrong** side of the hyperplane

Support Vector Classifier (Soft Margin Classifier)

- **Optimization problem** for a hyperplane that separates most of the training observations into the two classes, but may **misclassify a few observations**

$$\max_{\beta_0,\beta_1,\ldots,\beta_p,\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n,M} M \tag{6}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1 \tag{7}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \varepsilon_i) \tag{8}$$

$$\varepsilon_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{n} \varepsilon_i \leq C \tag{9}$$

- where variables $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ are **slack variables** and $C \geq 0$ is called a **tuning parameter**

Support Vector Classifier

- M is the **width of the soft margin**; we seek to make this quantity as large as possible
- $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ are **slack variables** that allow individual observations to be on the wrong side of the margin or the hyperplane
- Quantity ε_i tells us where the *i*-th observation is located, **relative to the hyperplane and to the margin**:
  - If $\varepsilon_i = 0$, then the *i*-th observation is on the correct side of the margin
  - If $\varepsilon_i > 0$, then the *i*-th observation is on the wrong side of the margin, and we say that it *violates* the margin
  - If $\varepsilon_i > 1$, then it is even on the wrong side of the hyperplane

Support Vector Classifier

- C is *budget* for the amount that the margin can be violated by the n observations
  - If $C = 0$: no budget for violations to the margin :

    $$\varepsilon_1 = \varepsilon_2 = \cdots = \varepsilon_n = 0 \tag{10}$$

    in which case (6) – (9) simply amounts to the maximal margin hyperplane optimization problem
  - For $C > 0$ no more than C observations can be on the wrong side of the hyperplane, because if observation i is on the wrong side of the hyperplane then $\varepsilon_i > 1$, so the sum in (9) is at least the number of violations to the hyperplane.
  - As the budget C increases, we become more tolerant to violations to the margin, and so the margin will widen.

Support Vector Classifier : Example

- Please check examples 2.4 and 2.5 of the Support Vector Machines chapter