Support Vector Machines

Nayel Bettache

What is SVM?

- The support vector machine is a direct method in a two-class classification problem.
- We want to find a hyperplane that separates the classes in feature space
- A hyperplane in p dimensions is a flat affine subspace of dimension $p - 1$.
- In general the equation for a hyperplane has the form

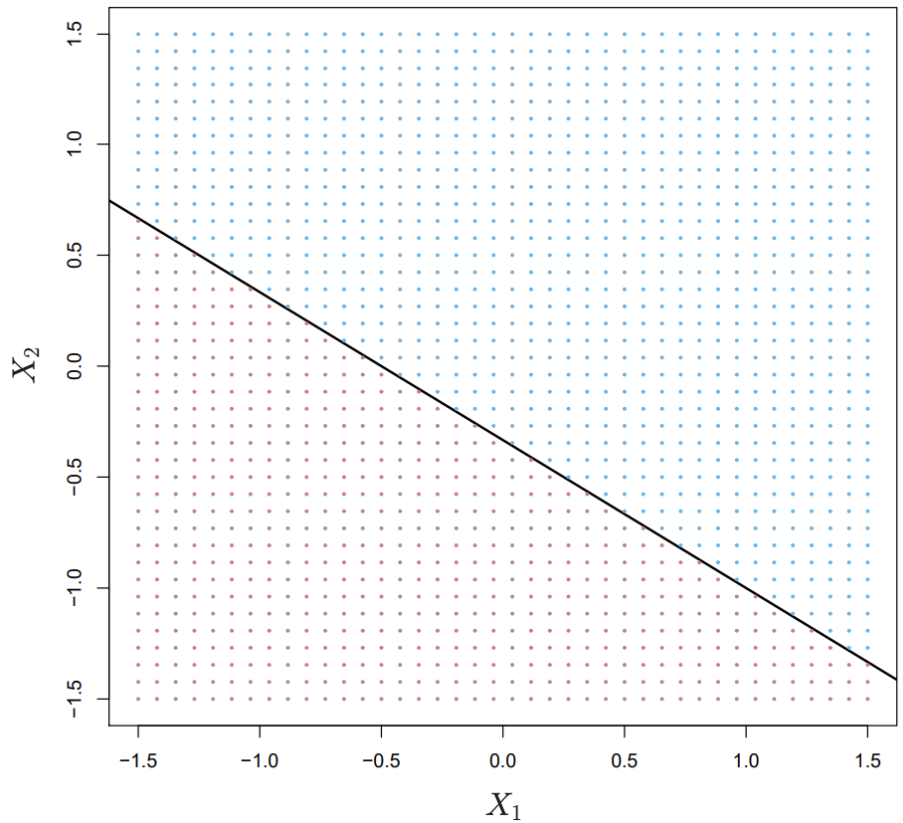
$$\beta_0 + \beta_1 X_1 + ... + \beta_p X_p = 0$$

- In p $= 2$ dimensions a hyperplane is a line.

Objective

- Goal: develop a classifer based on the training data that will correctly classify the test observation using its feature measurements.
- We have seen a number of approaches for this task
  - Linear discriminant analysis (LDA)
  - Logistic regression
  - Classifcation trees
- We will now see a new approach that is based upon the concept of a separating hyperplane.

Example

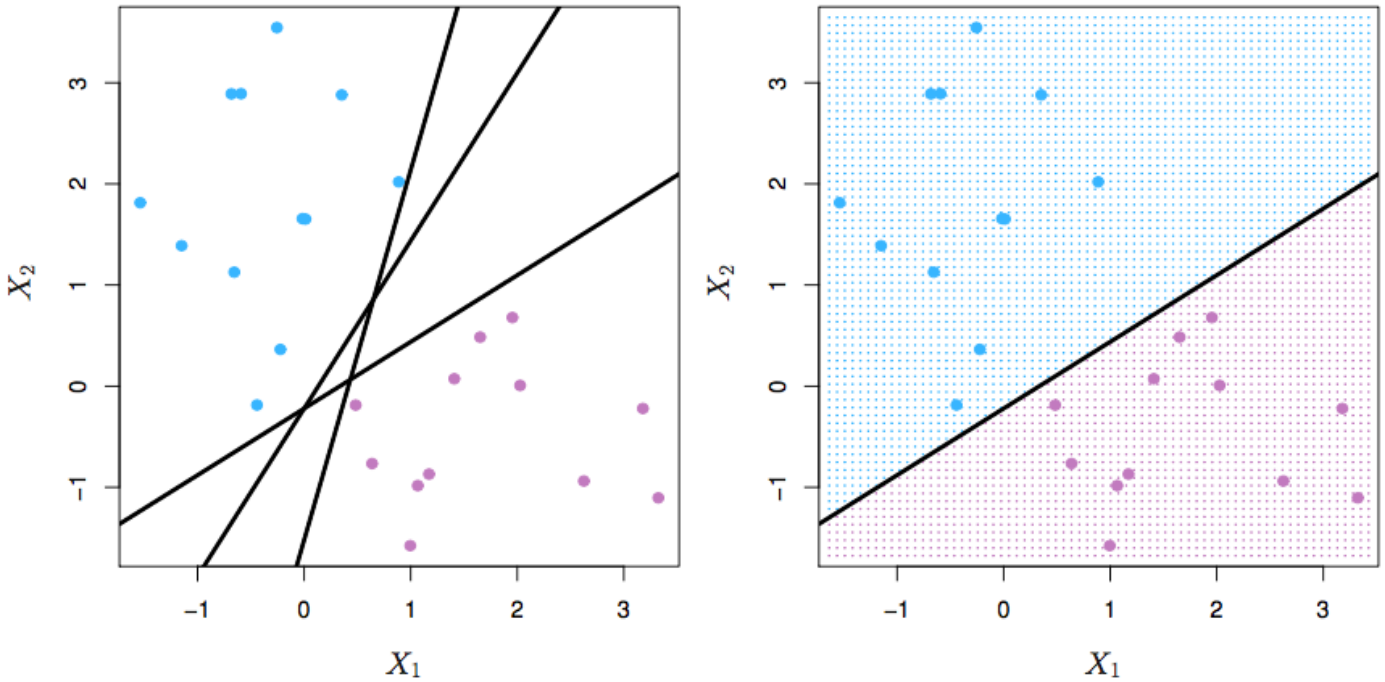


The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$, and the purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.

Hyperplane

- If $f(X) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$, then $f(X) > 0$ for points on one side of the hyperplane, and $f(X) < 0$ for points on the other.
- If we predict points as $Y_i = +1$ if $f(X) > 0$, and $Y_i = -1$ if $f(X) < 0$, then we have $Y_i f(X_i) > 0$ for all i, $f(X) = 0$ defines a separating hyperplane.

Hyperplane



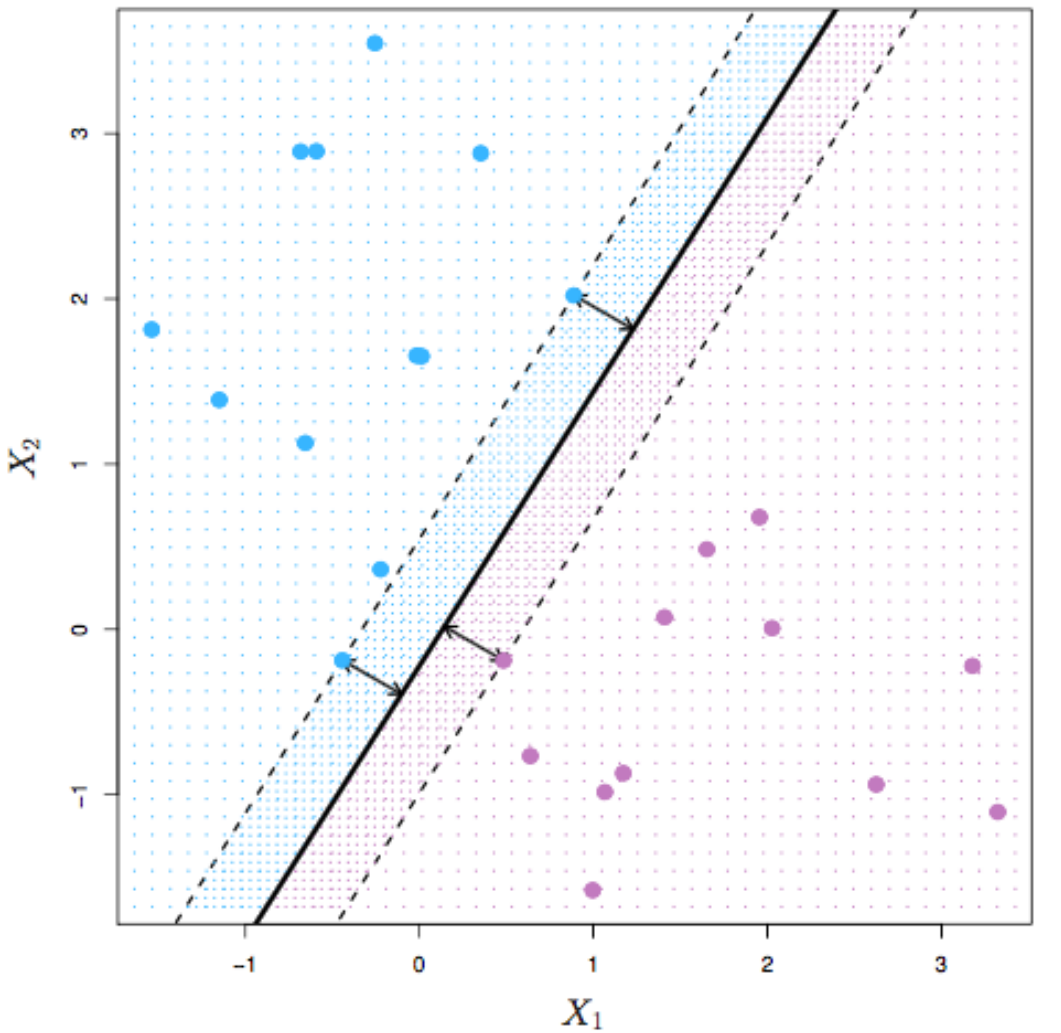
- Left: There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.
- Right: A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifer based on this separating hyperplane.

Details

- Assumption: Our data can be perfectly separated using a hyperplane
- Problem: There will in fact exist an infinite number of such hyperplanes.
- Solution: A natural choice is the maximal margin hyperplane.
- Details: It is the separating hyperplane that is farthest from the training observations.
- Idea: We can compute the (perpendicular) distance from each training observation to a given separating hyperplane; the smallest such distance is the minimal distance from the observations to the hyperplane, and is known as the margin.

Maximal Margin Classifier

- Among all separating hyperplanes, find the one that makes the biggest gap or margin between the two classes.



Construction of the Maximal Margin Classifier

Constrained optimization problem

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

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

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

$$\text{for all} \quad i = 1, \ldots, N.$$

This is a quadratic constrained programming problem that can be efficiently solved with Lagrangian multipliers .

Separating hyperplane

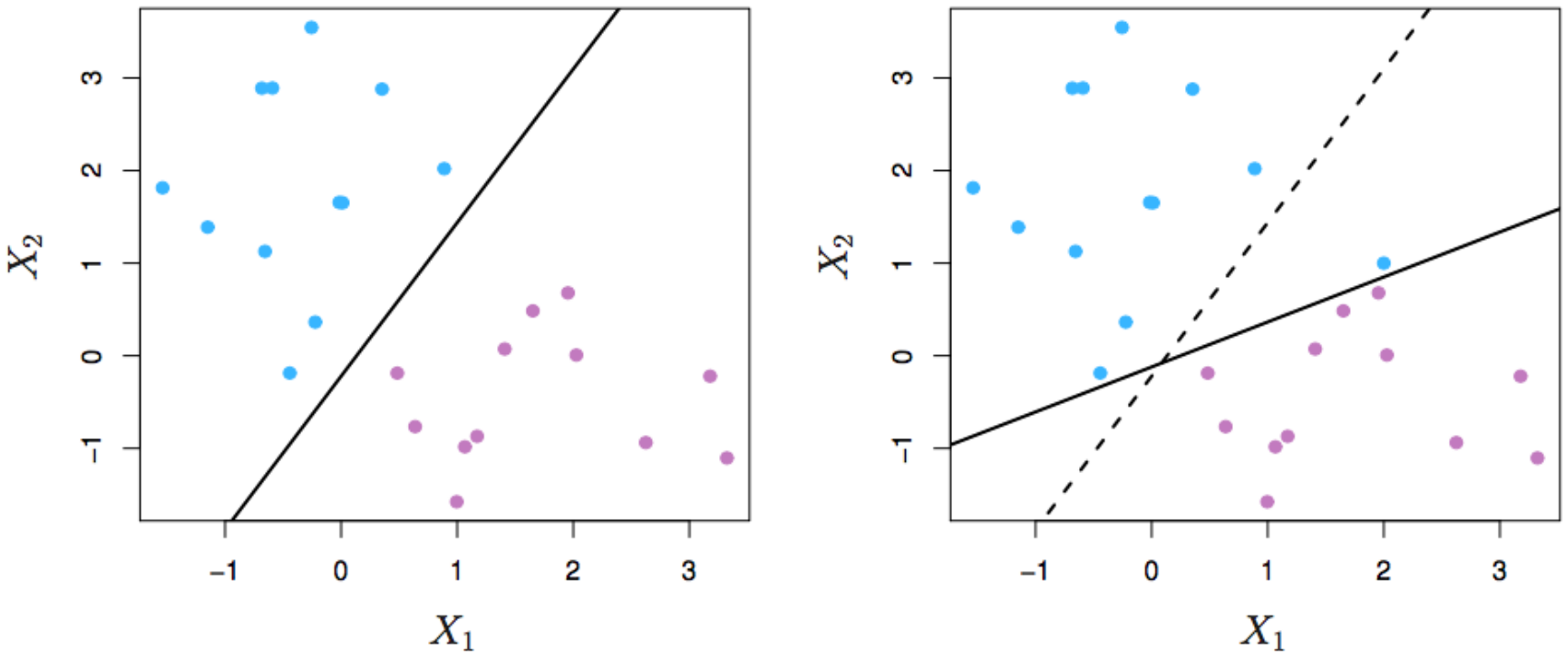
The maximal margin classifer is a very natural way to perform classification,

if a separating hyperplane exists.

Noisy Data

- Sometimes the data are separable, but noisy.
- This can lead to a poor solution for the maximal-margin classifier.
- In this case, we might be willing to consider a classifier based on a hyperplane that does **NOT** perfectly separate the two classes.

Example

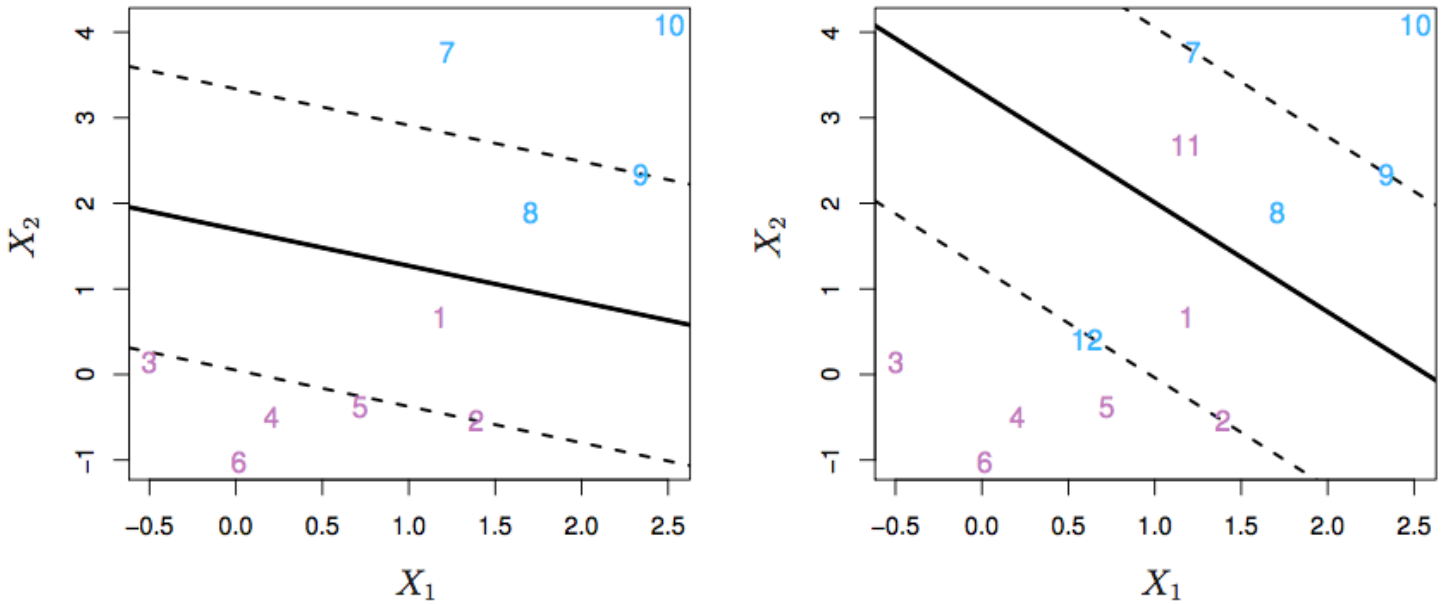


- Left: Two classes of observations are shown in blue and in purple, along with the MMH.
- Right: An additional blue observation has been added.
- The MMH is extremely sensitive to a change in a single observation – overfitting ?

Support Vector Classifier

- **Soft Margin Classifier**
- We allow some observations to be on the incorrect side of the hyperplane.
- The margin is **soft** because it can be violated by some of the training observations.

Example and construction



$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \; M \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

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

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

Read section 9.2.2 of the textbook for technical details.

Kernel trick

- Sometime a linear boundary simply won't work.
- Map the data into a higher-dimensional space where a linear separation becomes possible.
- The separation is LINEAR in the high-dimensional space but NON LINEAR in the original feature space.
- Kernel SVM: $f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(x, x_i)$, where $K(x, x_i)$ is a kernel.
  - Polynomial kernel: $K(x_i, x_{i'}) = (x_i \cdot x_{i'})^d$
  - Gaussian radial kernel: $K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$.

SVM or Logistic Regression

- When classes are (nearly) separable, SVM does better than LR. So does LDA.
- When not, LR (with ridge penalty) and SVM very similar.
- If you wish to estimate probabilities, LR is the choice.
- For nonlinear boundaries, kernel SVMs are popular. Can use kernels with LR and LDA as well, but computations are more expensive.