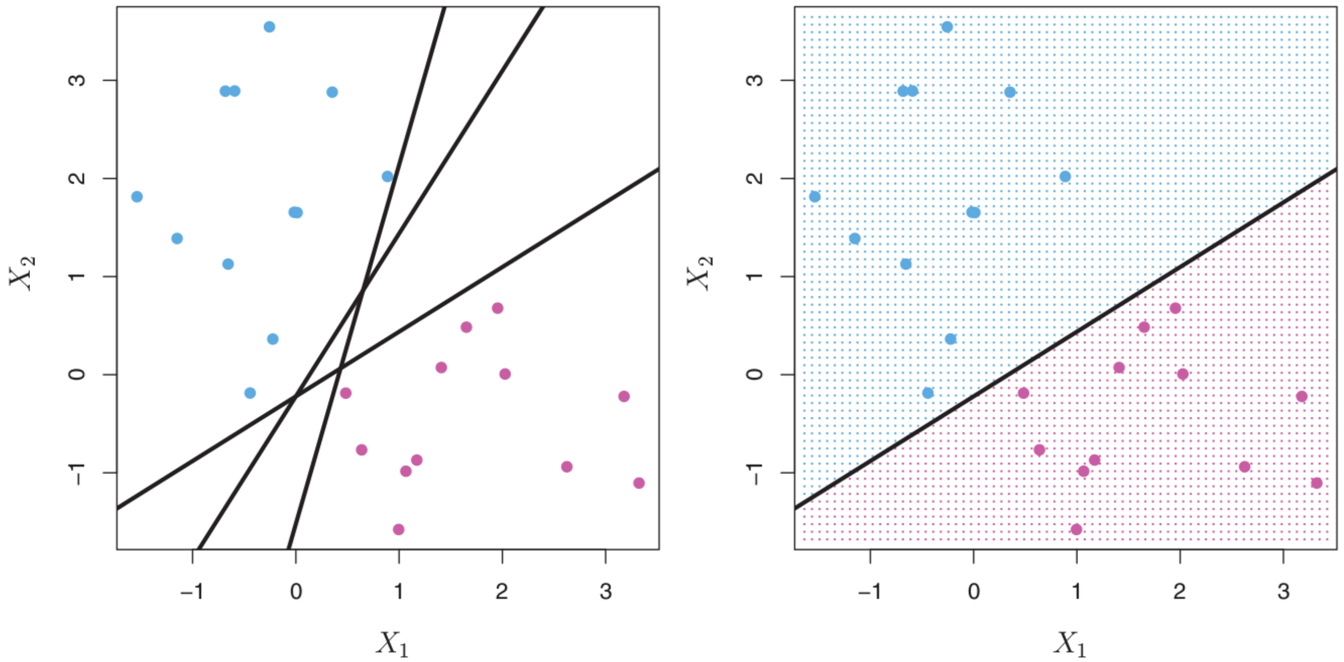
JIGSAW ACADEMY

Analytics for Professionals

Support Vector Machines with R

Separating Hyperplane



A hyperplane is defined by $\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p = 0$

All points on either side of the plane satisfy

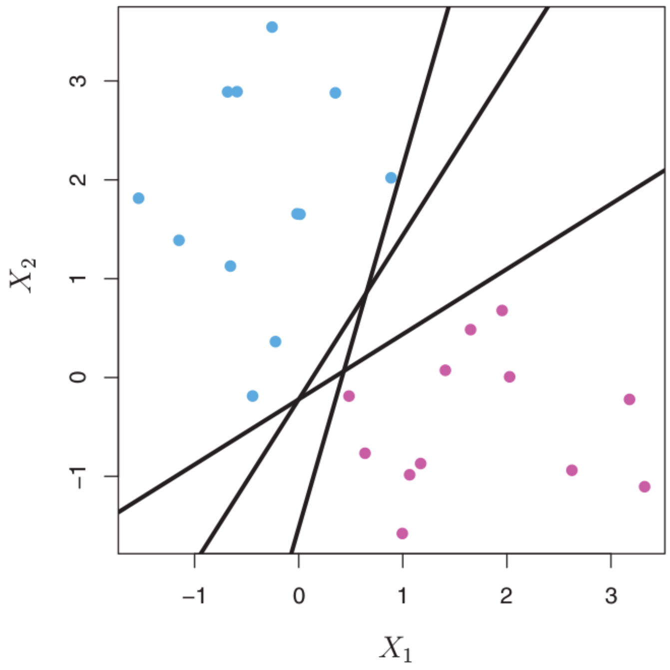
$$\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \begin{cases} < 0 & \text{(same side as origin)} \\ > 0 & \text{(other side of origin)} \end{cases}$$

Separating Hyperplane...

If we assign the purple points $y_i = -1$ and the blue points $y_i = 1$ Then the separating hyperplanes satisfy

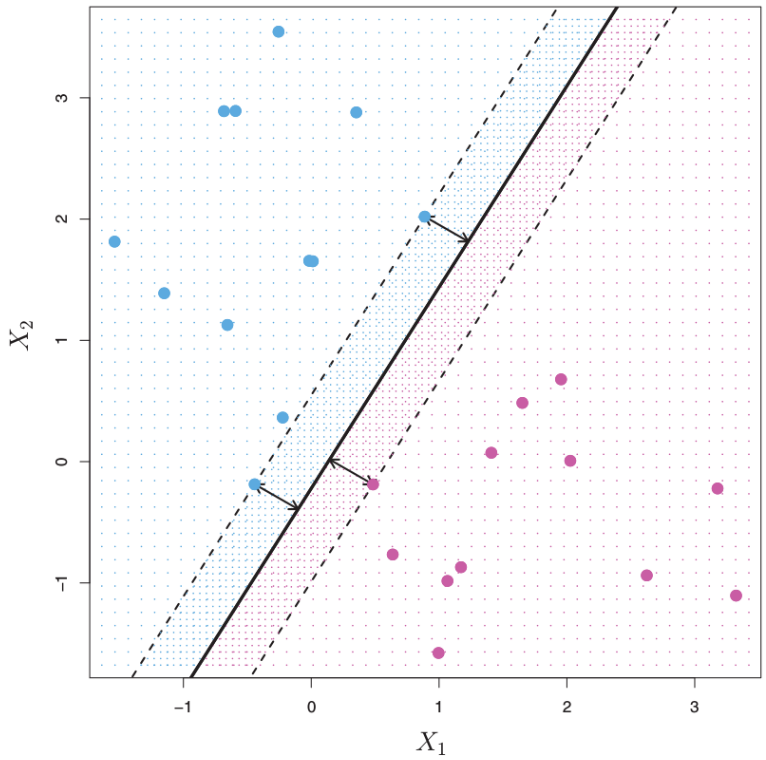
$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) > 0 \quad \forall i \in 1, \ldots, n$$

- If such natural classifier exists, we can use the distance of a point from the plane as a measure of confidence in classification
- Unfortunately, if it does exist, infinitely many exist!



Maximal Margin Classifier

- A natural choice is the plane 'farthest' from training data points in the two classes.
- A maximal margin separating hyperplane is the mid-line of the widest separating slab that may be inserted between the two classes.
- The training points lying on the boundaries of the slab are known as the supporting vectors



Maximal Margin Classifier...

Distance of a point $\vec{x}^*$ from a plane is given by

$$\frac{(\beta_0 + \beta_1 x_1^* + \ldots + \beta_p x_p^*)}{(\sum_{i=1}^{p} \beta_i^2)^{1/2}}$$

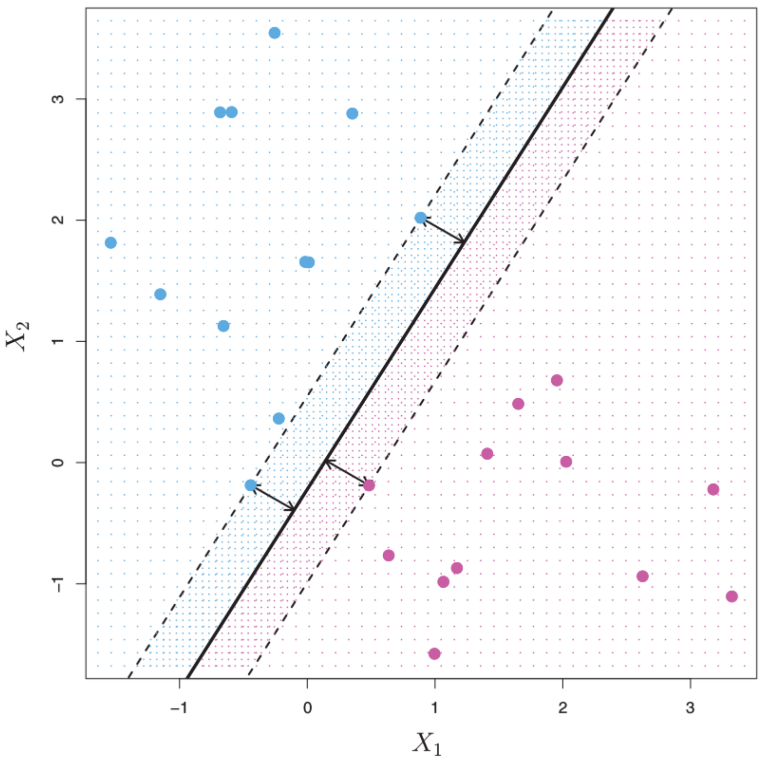
Hence the *maximal margin hyperplane* is the solution to

$$\max_{\beta_0 \ldots \beta_p} M$$

$$\text{subject to} \sum_{i=1}^{p} \beta_i^2 = 1$$

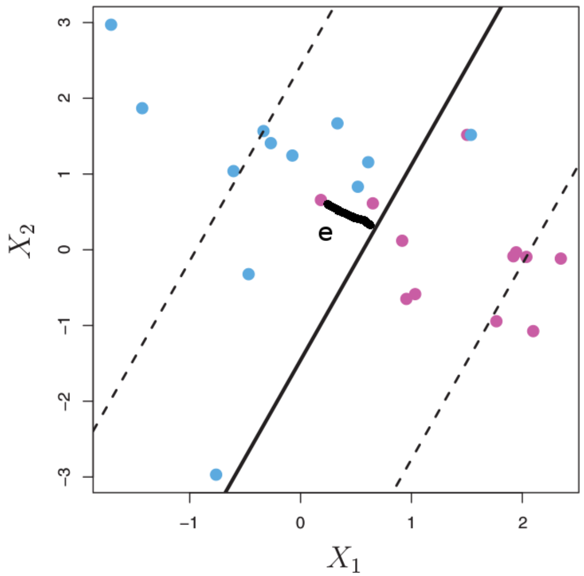
$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M, \quad i = 1, \ldots, n$$

Provided such hyperplane exists (i.e. complete separation is possible)



Support Vector Classifier

If no separating hyperplane exists, we can relax the constraint $\geq M$ with slack (error) terms to give a support vector classifier Specifically, support vector classifier is the solution to:



$$\max_{\beta_0 \ldots \beta_p} M$$

$$\text{subject to} \sum_{i=1}^{p} \beta_i^2 = 1$$

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

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

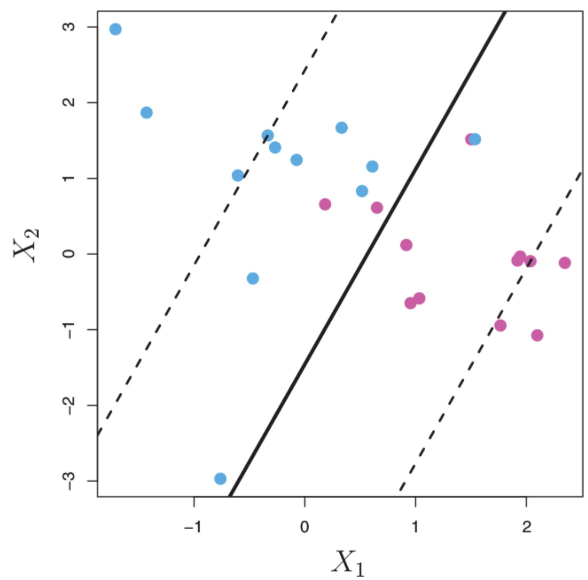
where C is the tuning parameter, identified through cross-validation

Support Vector Classifier...

Observe

- $\epsilon_i = 0$ for all points on the 'right' side of the classifier
- These points outside the margin have no role in determining the hyperplane
- $\epsilon_i > 0$, point has violated the margin
- $\epsilon_i > 1$, point is on the wrong side of the hyperplane
- C is also called budget, and at most C points can be on the wrong side of the hyperplane

- These two sets of points are together known as support vectors
- Only support vectors play a role in determining the hyperplane



Support Vector Classifier Solution

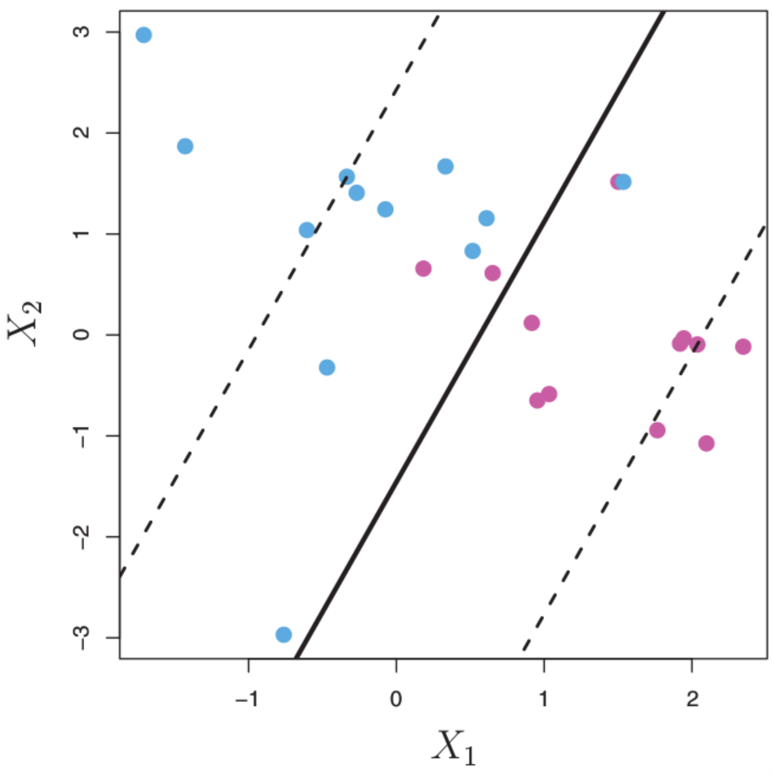
- Uses really just the inner product of the support vector

$$\langle x_i, x_j \rangle = \sum_{k=1}^{p} x_{ik} x_{jk}$$

- Let S be the support vectors. Then the linear support vector is given by

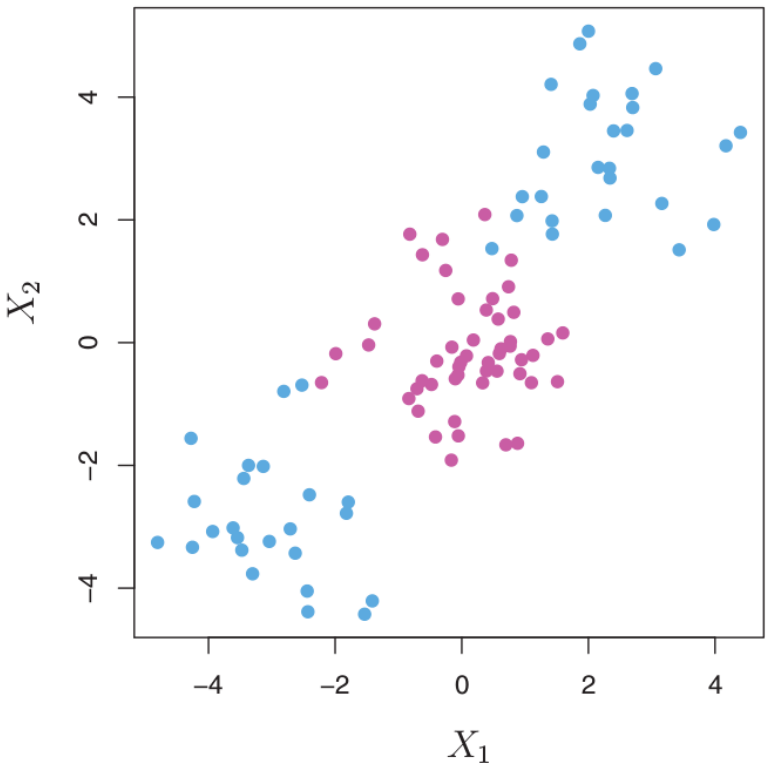
$$f(x^*) = \beta_0 + \sum_{i \in S} \alpha_i \langle x_i, x^* \rangle$$

- If the cardinality of S is n (number of points in S), then one needs only $n(n-1)/2$ inner products to estimate the β_0 and α_is



Support Vector Machines

- Linear boundaries in p dimension, called p-feature space is defined using $X_1, X_2, \ldots, X_p$,
- This may be extended by including quadratic terms to a $2p$-feature space:
  i.e. $X_1, X_1^2, X_2, X_2^2, \ldots, X_p, X_p^2$
- This is expensive in terms of increasing computational complexity
- Support vector machines enlarge the feature space using kernels
- Kernels are generalizations of the inner product



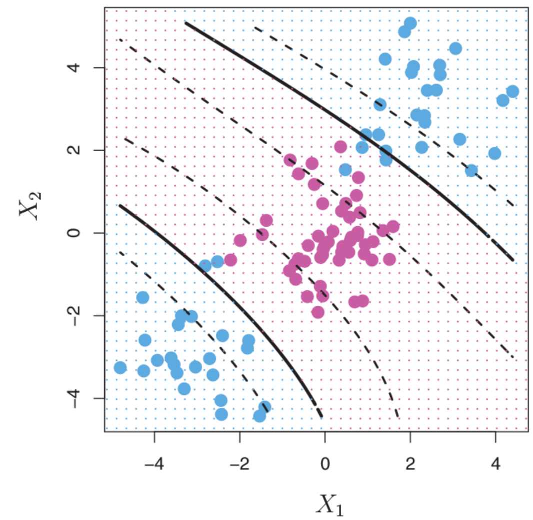
Examples of Kernels

- Polynomial Kernel of degree d

$$K(x_i, x_j) = (1 + \sum_{k=1}^{p} x_{ik} x_{jk})^d$$

$d > 1$, will have a solution

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



- Radial Kernel takes the form

$$K(x_i, x_j) = exp(-\gamma \sum_{k=1}^{p} (x_{ik} - x_{jk})^2)$$

$\gamma > 0$