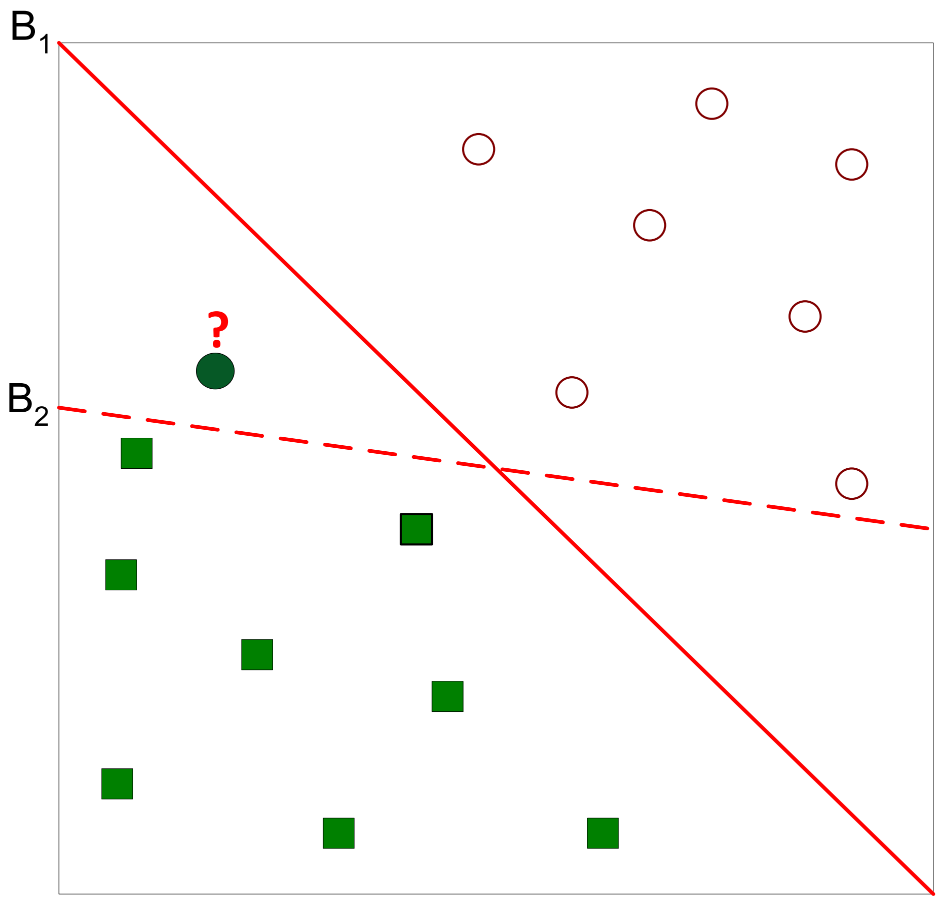
SUPPORT VECTOR MACHINES

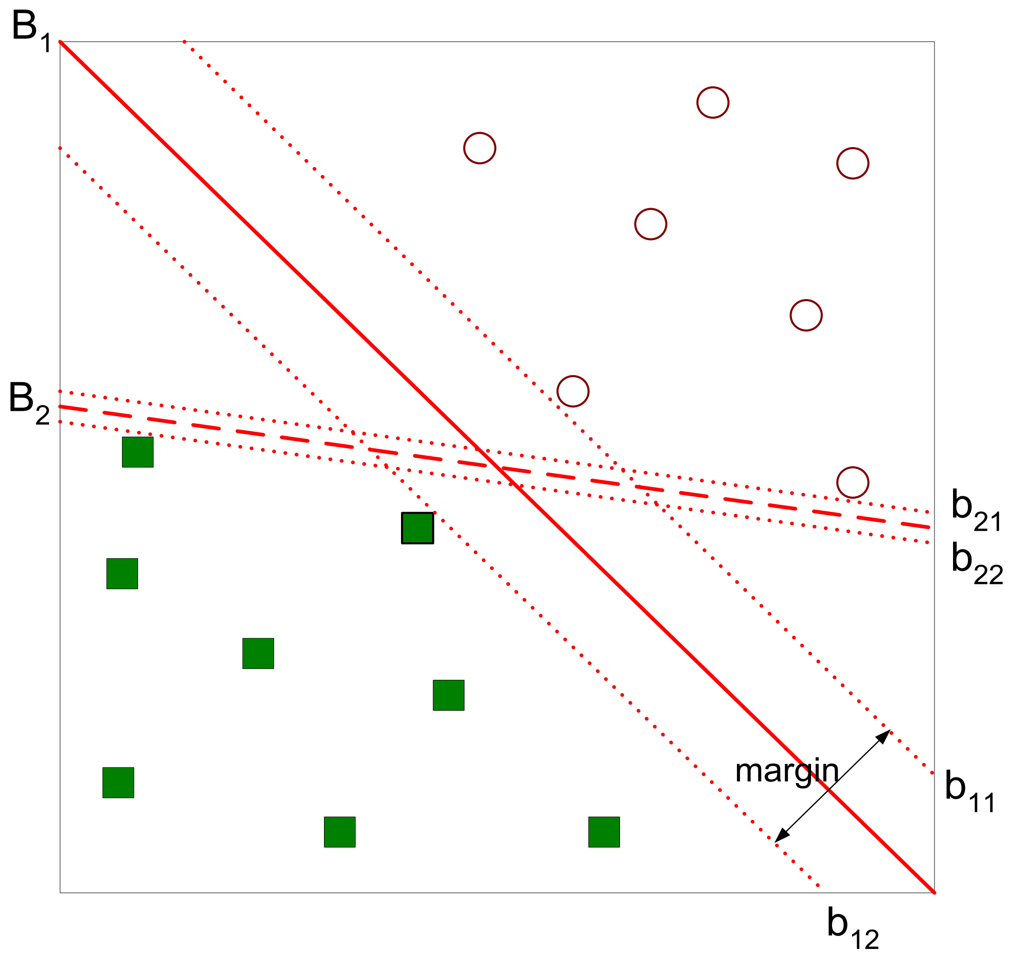
Support Vector Machines

- Both decision boundaries classify all training points correctly
- Which decision boundary is better?
- Which one is more likely to classify correctly unseen test tuples?



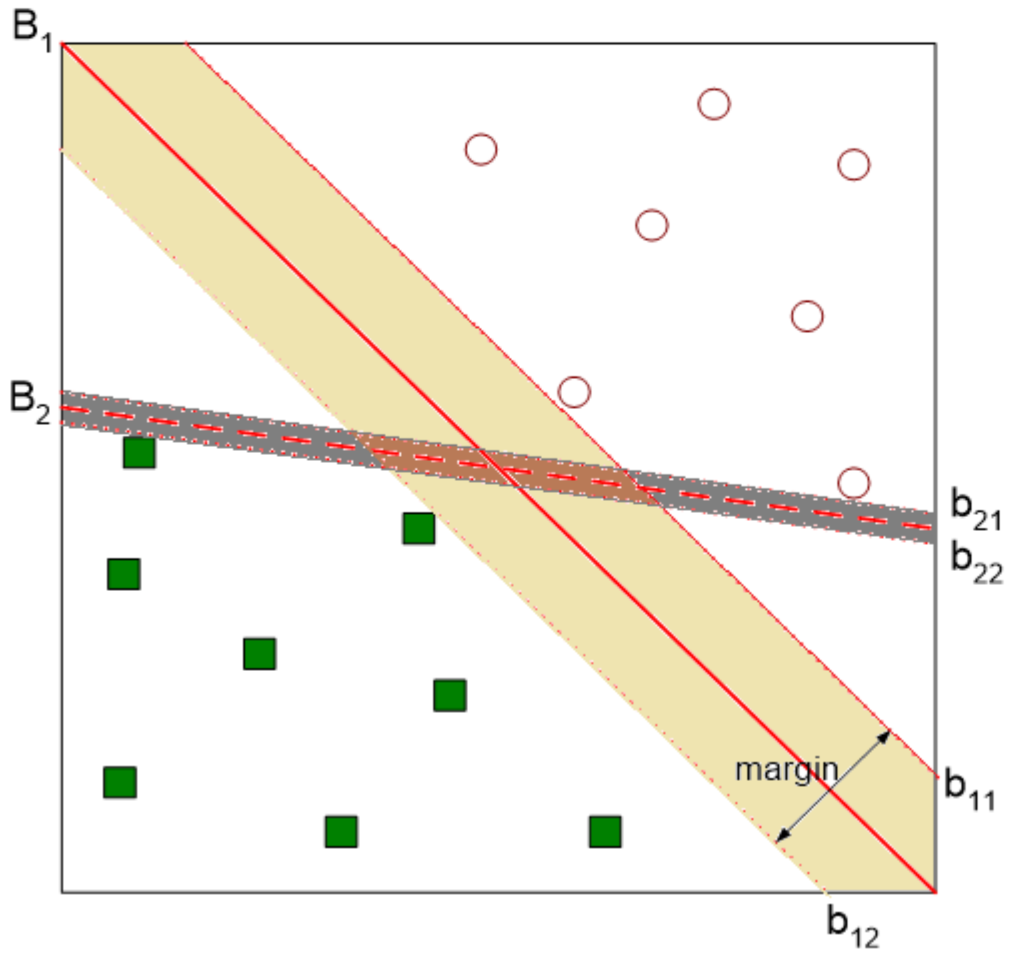
Maximum Marginal Hyperplane

- Each separating hyperplane has a margin
- The hyperplane with the largest margin is expected to be more accurate
- During the learning phase, the SVM searches for the hyperplane with the largest margin (MMH)



Maximum Marginal Hyperplane

- Each separating hyperplane has a margin
- The hyperplane with the largest margin is expected to be more accurate
- During the learning phase, the SVM searches for the hyperplane with the largest margin (MMH)



Linear SVM

Any separating hyperplane:

$$\omega^T x + b = 0$$

where:

$\omega^T = \{w_1, w_2, \ldots, w_n\}$ is a weight vector

n is the number of attributes

b is a scalar

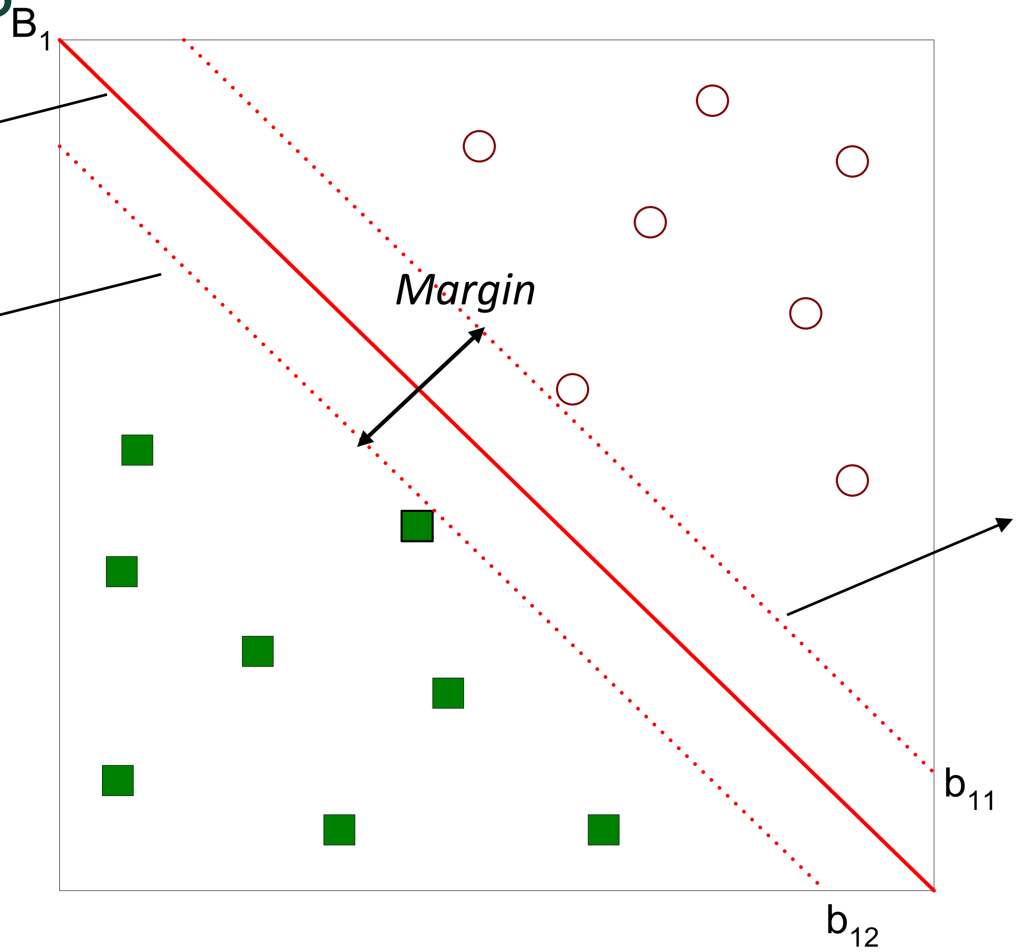
When ω and b are determined, classify using:

$$\hat{\boldsymbol{y}} = \begin{cases} 1 & \text{if } \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x} + \mathrm{b} \geq 1 \\ -1 & \text{if } \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x} + \mathrm{b} \leq -1 \end{cases}$$

Hyperplane Margin

$\boldsymbol{\omega}^T \boldsymbol{x} + \boldsymbol{b} = 0$

$\omega^T x + b = -1$



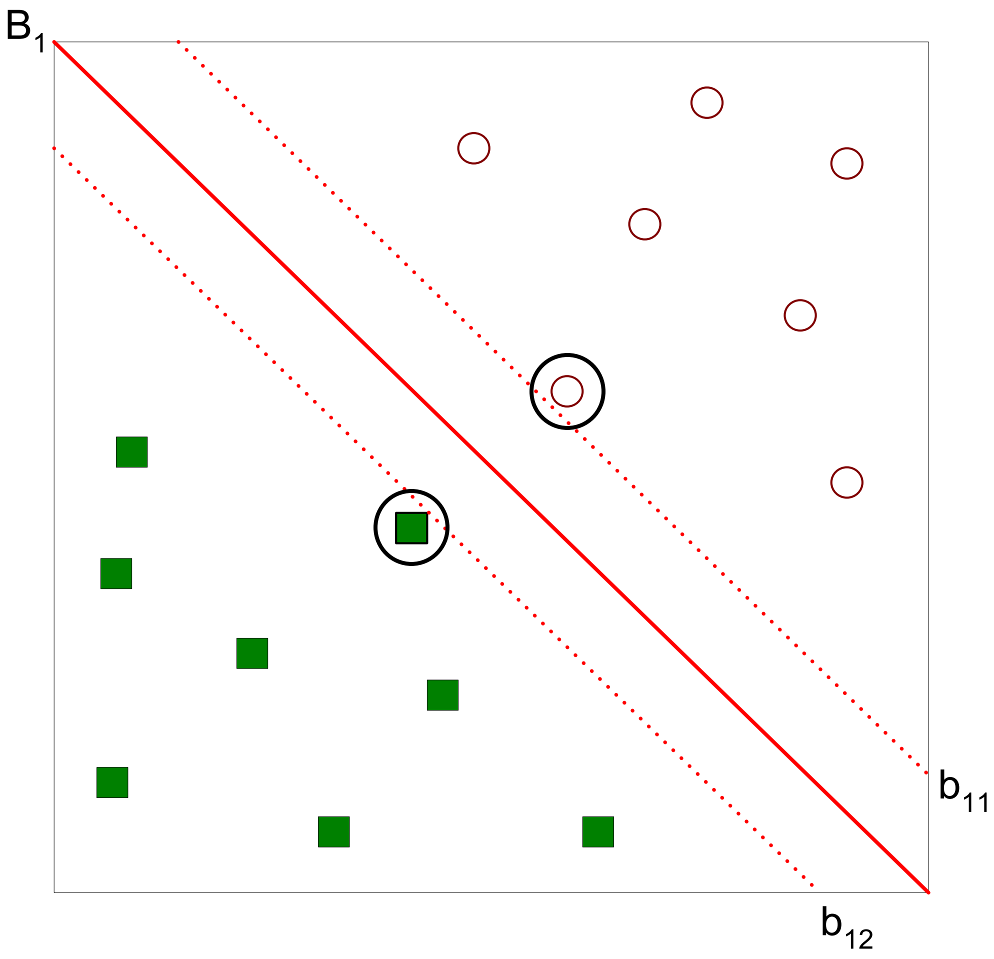
$\omega^T x + b = 1$

$$\text{Margin} = \frac{|(1-b)-(-1-b)|}{\|\omega\|}$$

$$\text{Margin} = \frac{2}{\|\omega\|}$$

Support Vectors

- Tuples that fall on the hyperplanes defining the margin b_{12} and b_{11}
- **Support vectors** are the samples that lie on the decision boundary
- Determine the maximum marginal hyperplane



How to find MMH?

- Goal is to find w and b that maximize the margin: $\frac{2}{\|\omega\|}$
- Optimization problem:

Minimize $$E(\omega) = \frac{\|\omega\|^2}{2}$$

Subject to:

$$y_i = \begin{cases} 1 & \text{if } \omega^T x_i + b \geq 1 \\ -1 & \text{if } \omega^T x_i + b \leq -1 \end{cases}$$

or

$$\boldsymbol{y}_i(\boldsymbol{\omega}^T \boldsymbol{x}_i + \boldsymbol{b}) \geq 1, \quad \boldsymbol{i} = 1, 2, \ldots, \boldsymbol{N}$$

Each training tuple adds one constraint

How to find MMH?

Minimize $E(\omega) = \frac{\| \omega \|^2}{2}$

Subject to:

$$y_i = \begin{cases} 1 & \text{if } \omega^T x_i + b \geq 1 \\ -1 & \text{if } \omega^T x_i + b \leq -1 \end{cases}$$

- Constrained quadratic optimization problem
- Solved using a Lagrangian formulation and KKT conditions

$$\max_{\lambda_i} \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j$$

How to find MMH?

- Once you get the Lagrange multipliers λ_i , you can compute the optimal weight vector w and b as follows:
- The weight vector w is a linear combination of the training examples that correspond to non-zero Lagrange multipliers λ_i:

$$w = \sum_{i=1}^{n} \lambda_i y_i x_i$$

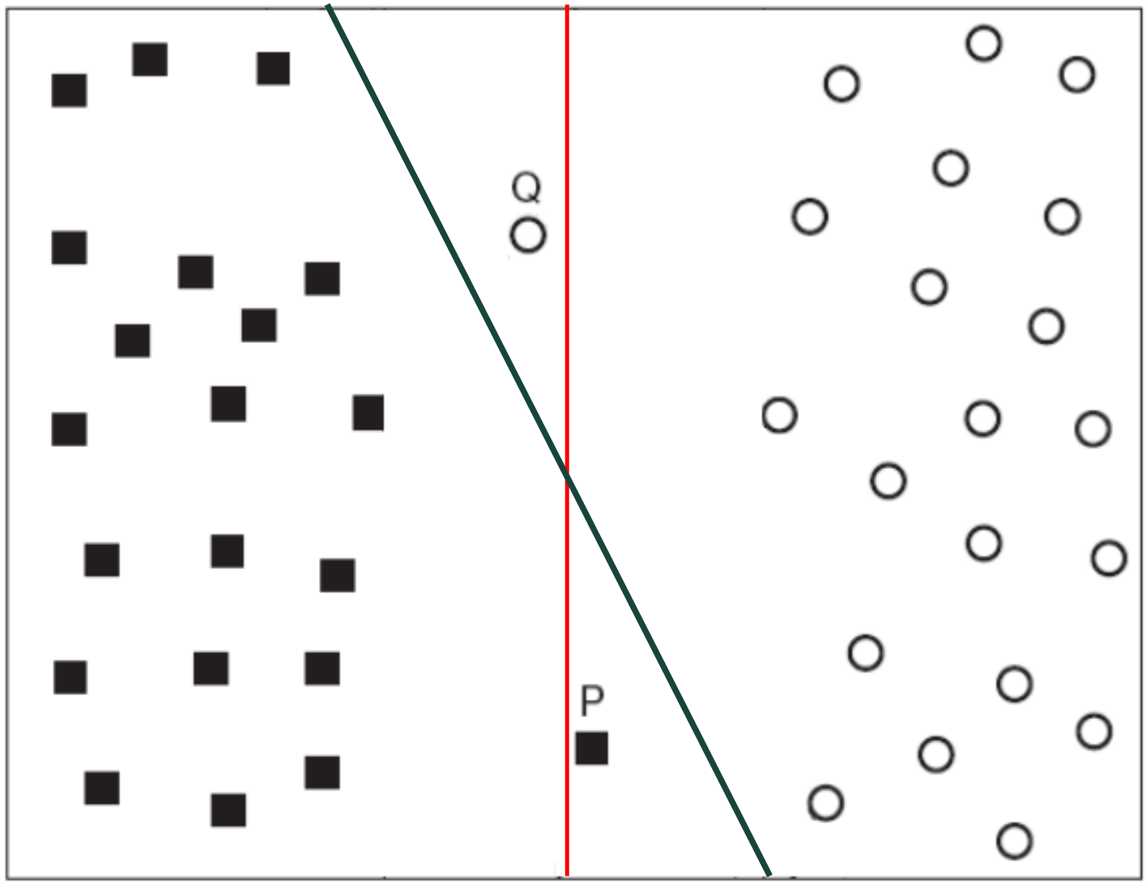
Only the support vectors (those with non-zero λ_i) contribute to the weight vector w.

- Once you have w, you can compute the bias term b by using any of the support vectors x_s

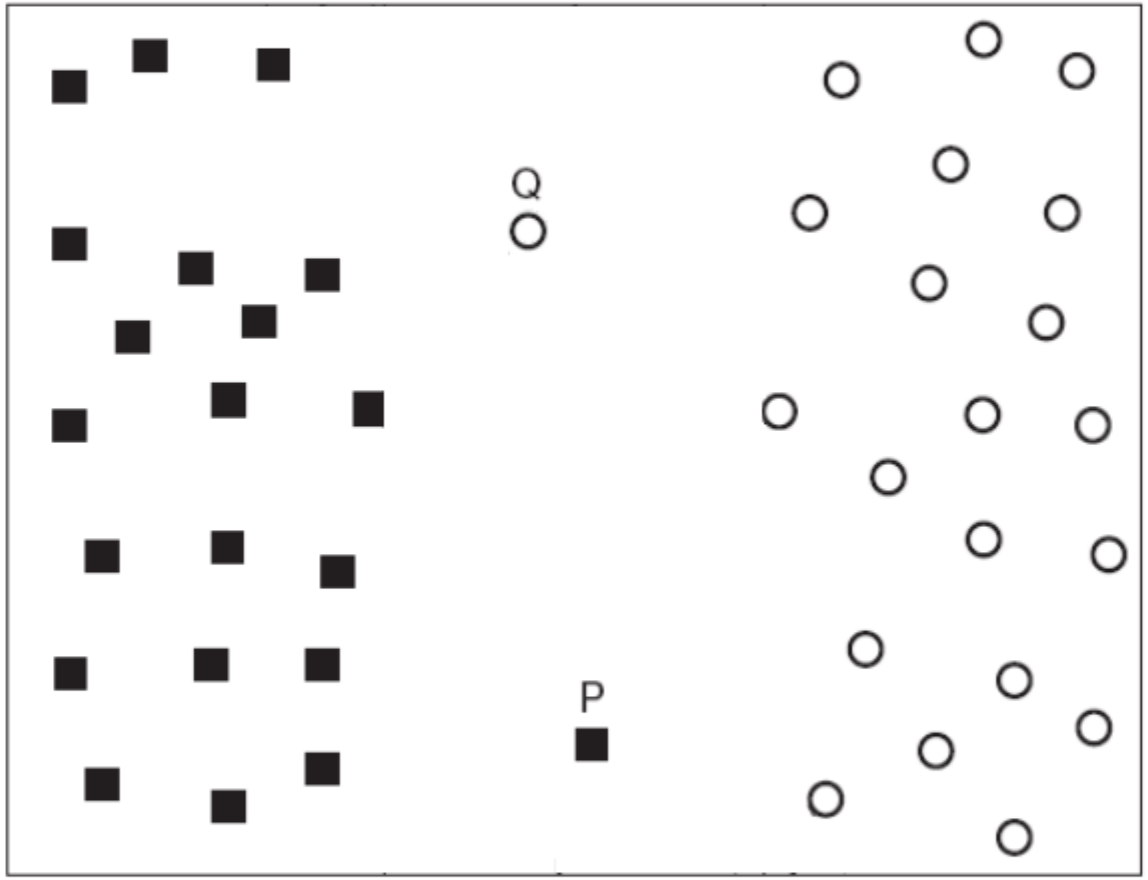
$$y_s\,(w^T x_s + b) = 1$$

Can compute b for each support vector and average them to get a more robust estimate of b

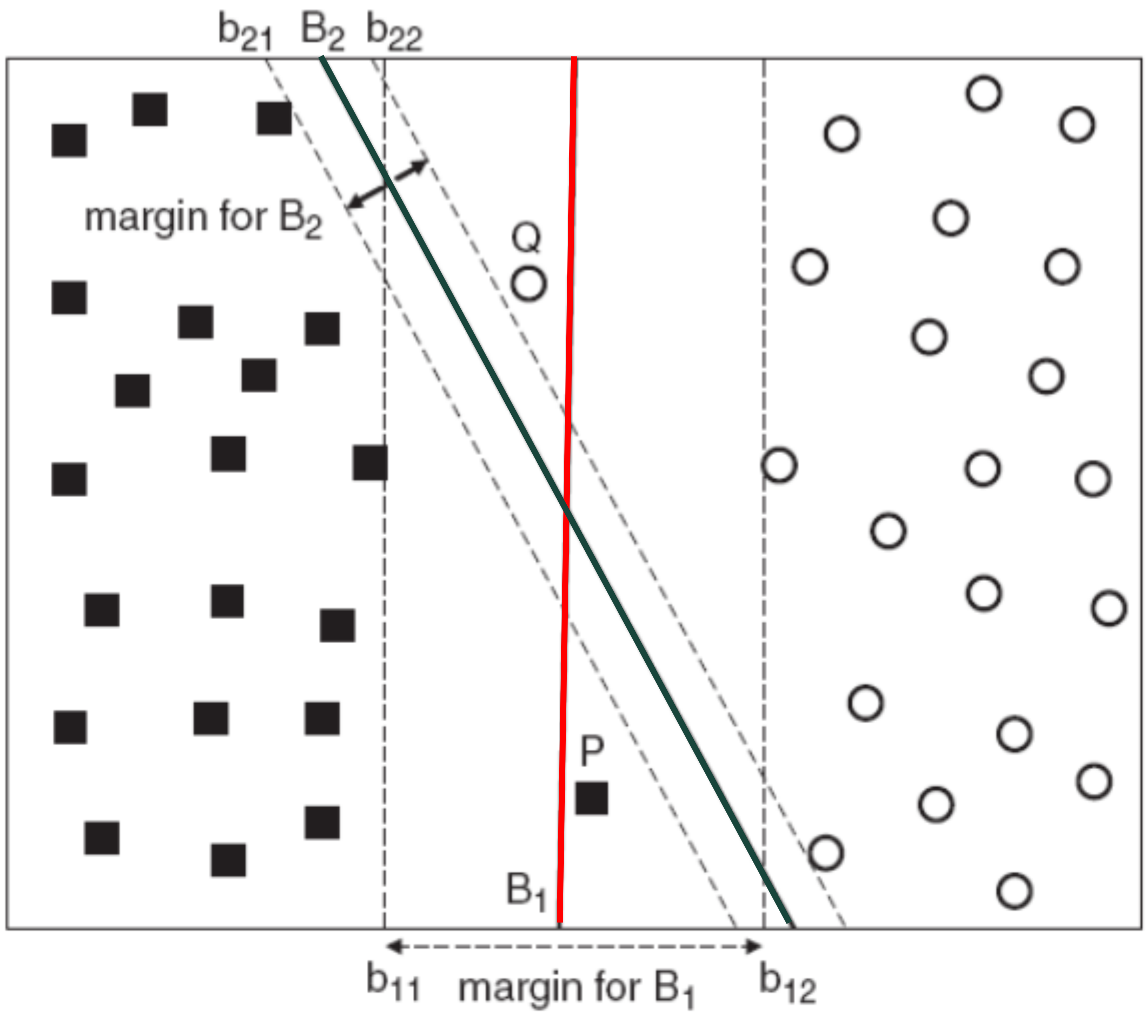
Linear SVM: Nonseparable Case



Linear SVM: Nonseparable Case

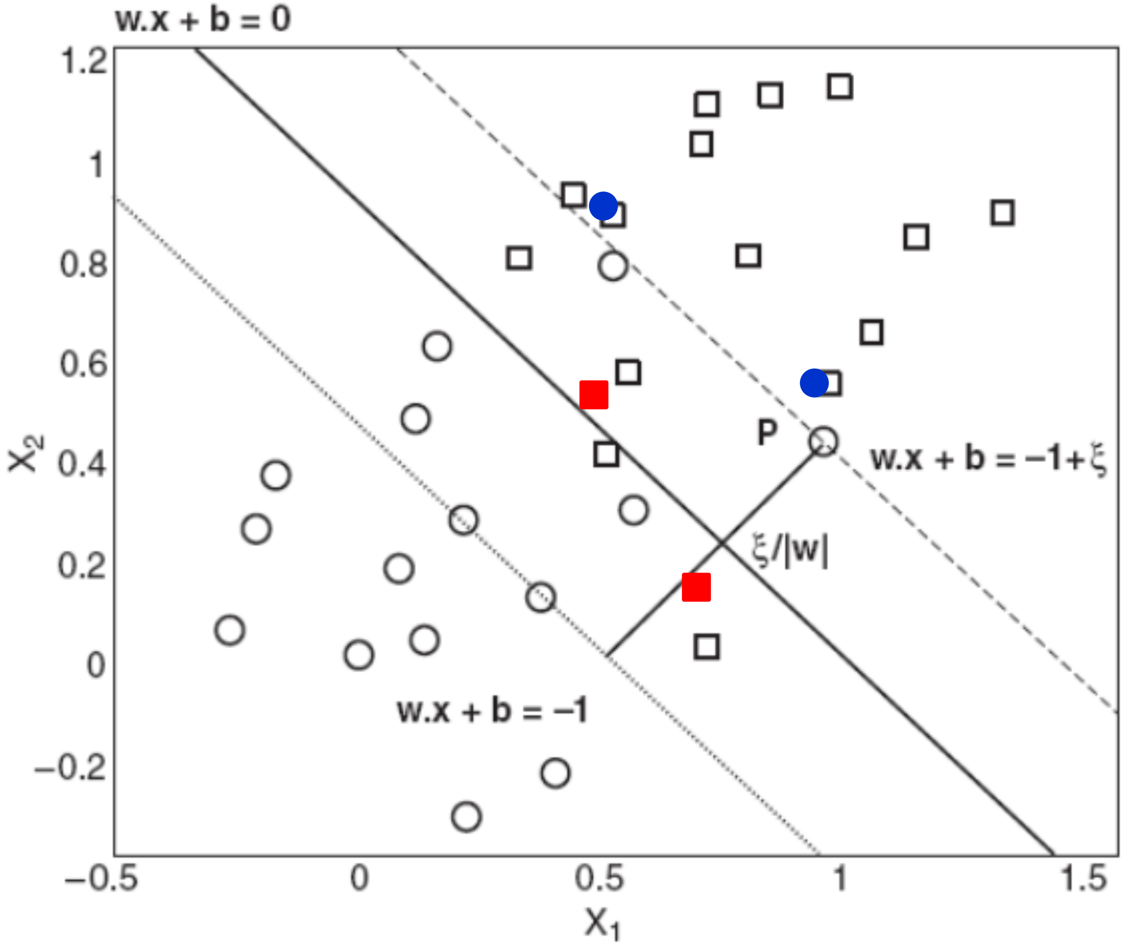


Linear SVM: Nonseparable Case



Consider a tradeoff between margin width and training errors

Linear SVM: Nonseparable Case



Linear SVM: Nonseparable Case

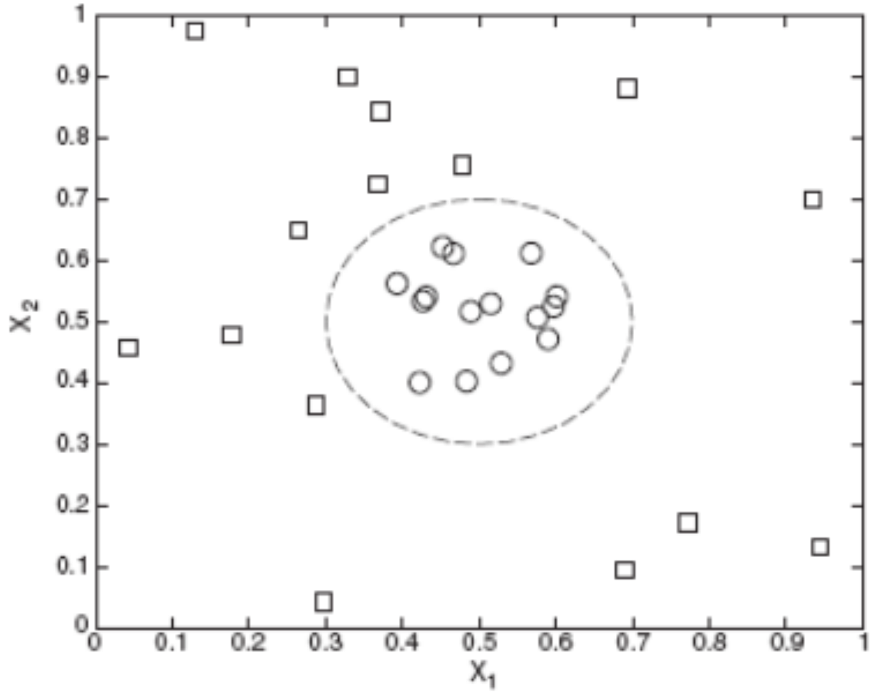
Minimize $E(\omega) = \frac{\| \omega \|^2}{2} + C(\sum_{i=1}^{N} \zeta_i)^k$

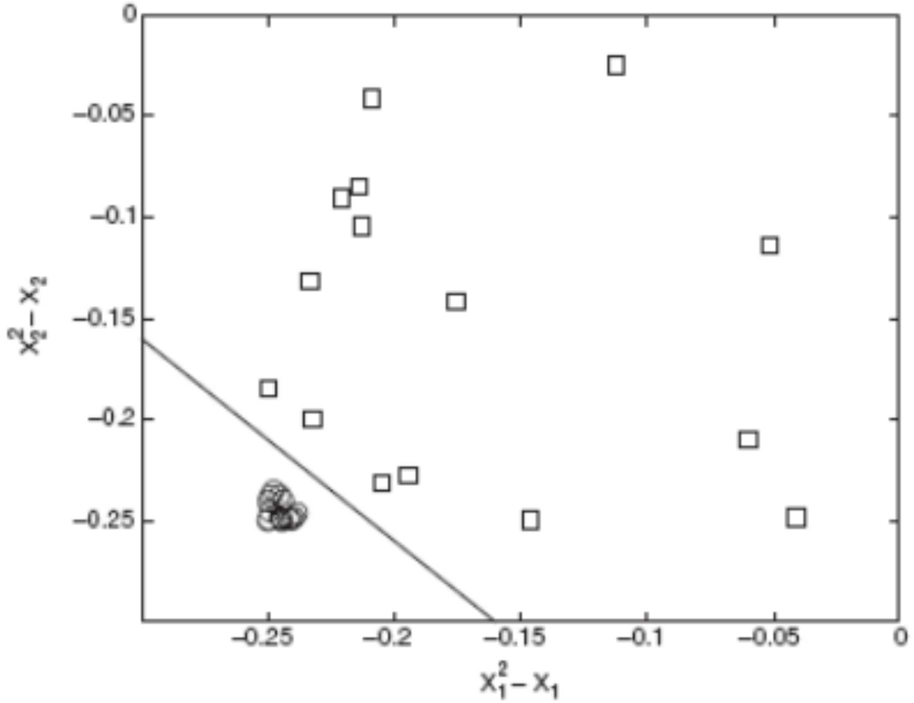
Subject to:

$$y_i = \begin{cases} 1 & \text{if } \omega^{\mathrm{T}} x_i + \mathrm{b} \geq 1 - \zeta_{\mathrm{i}} \\ -1 & \text{if } \omega^{\mathrm{T}} x_i + \mathrm{b} \leq -1 + \zeta_{\mathrm{i}} \end{cases}$$

- Relax constraints
- Add penalty to objective function

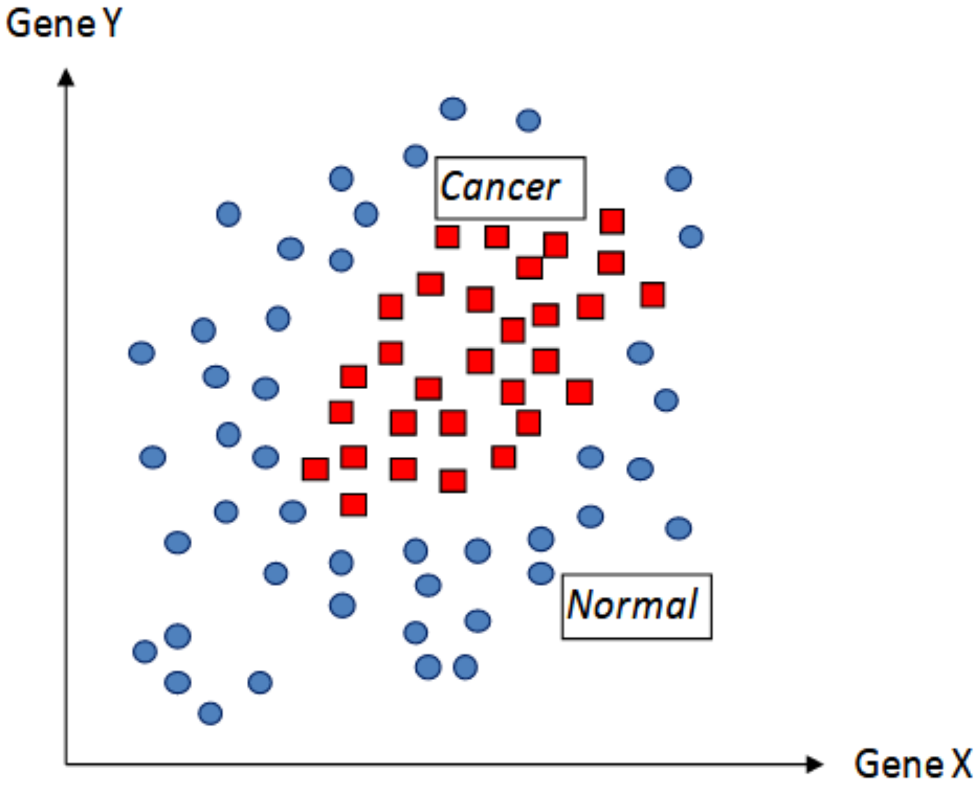
Nonlinear SVM

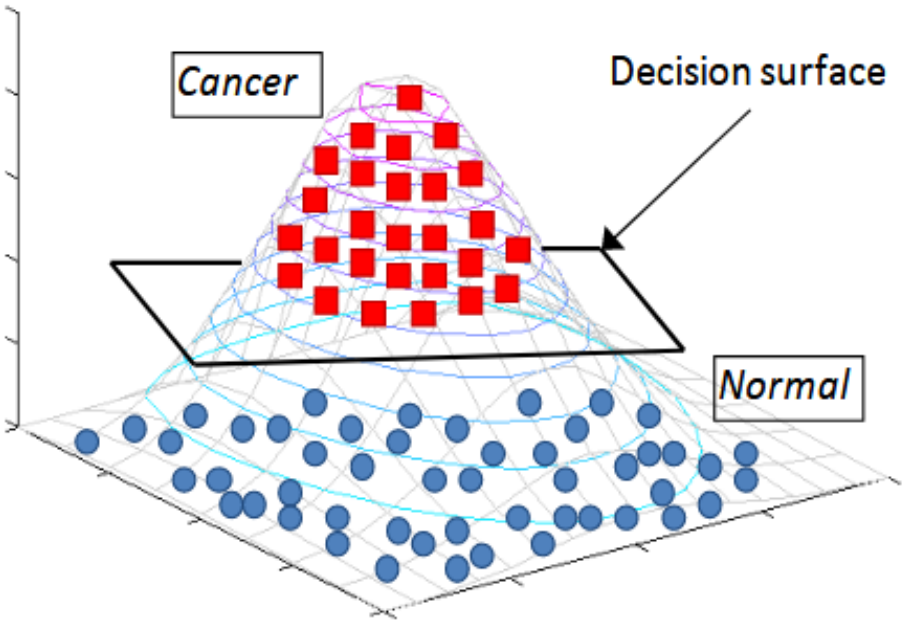




Linear SVM cannot solve this case
Transform data set from original space into a new space such that data has linear boundary

Nonlinear SVM – Predicting cancer risk





https://med.nyu.edu/chibi/sites/default/files/chibi/Final.pdf

Attribute Transformation

- Given features x_1, x_2
- Learn a transformation function ϕ
- Degree 2 polynomial
  - $\{x_1^2,\ \ x_1 x_2,\ \ x_2^2\}$
  - $\{x_1^2 - x_1, x_2^2 - x_2\}$
- Degree 3 polynomial
  - $\{x_1^3,\ \ x_2^3,\ \ x_1 x_2^2,\ \ x_1^2 x_2\}$
- Challenges:
  - Find hyperplane in transformed space

Transformations

Finding hyperplane in transformed space

- Original space
  - $\omega^T x + b = 0$
  - $\max_{\lambda_i} \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j$
- Transformed space
  - $w^\top \phi(x) + b = 0$
  - $\max_{\lambda_i} \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \phi(x_i) \cdot \phi(x_j)$

Expensive to compute $\phi(x_i) \cdot \phi(x_j)$

Kernel Trick

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

- Polynomial
  - $k(x_i, x_j) = (x_i \cdot x_j + 1)^d$
- Radial Basis Function (RBF)
  - $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$
- Sigmoid / Hyperbolic tangent
  - $k(x_i, x_j) = \tanh(k\, x_i \cdot x_j - \delta)$

SVM - Characteristics

- Training time can be slow
- Can be formulated as convex optimization problem
- Highly accurate
- Less prone to overfitting than other methods
- Can be used for numeric prediction as well as classification
- Can be used with categorical attributes by transforming each possible value to a binary attribute