innovate achieve lead

BITS Pilani

Machine Learning (IS ZC464) Session 15: Support Vector Machines (SVM)

Slides adapted from following internet recourses

- http://www.cs.cmu.edu/~awm/tutorials
- http://www-labs.iro.umontreal.ca/~pift6080/H09/documents/papers/svm_tutorial.ppt

Classifiers

- Linear – perceptron model

$$w_1x_1+w_2x_2+\ldots+w_nx_n > T \text{ for class C1}$$

And $$w_1x_1+w_2x_2+\ldots+w_nx_n < T \text{ for class C2}$$

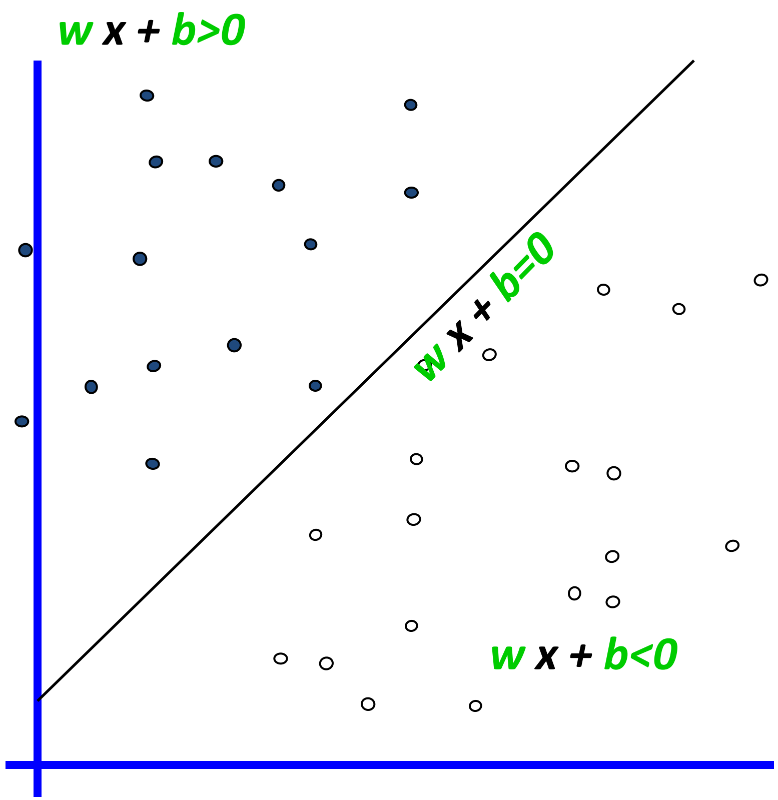
- Non linear – Multi-layer perceptron (MLP) neural network, radial basis function neural network

$$w_1\phi(x_1)+w_2\phi(x_2)+\ldots+w_n\phi(x_n) > T \text{ for class C1}$$

And $$w_1\phi(x_1)+w_2\phi(x_2)+\ldots+w_n\phi(x_n) < T \text{ for class C2}$$

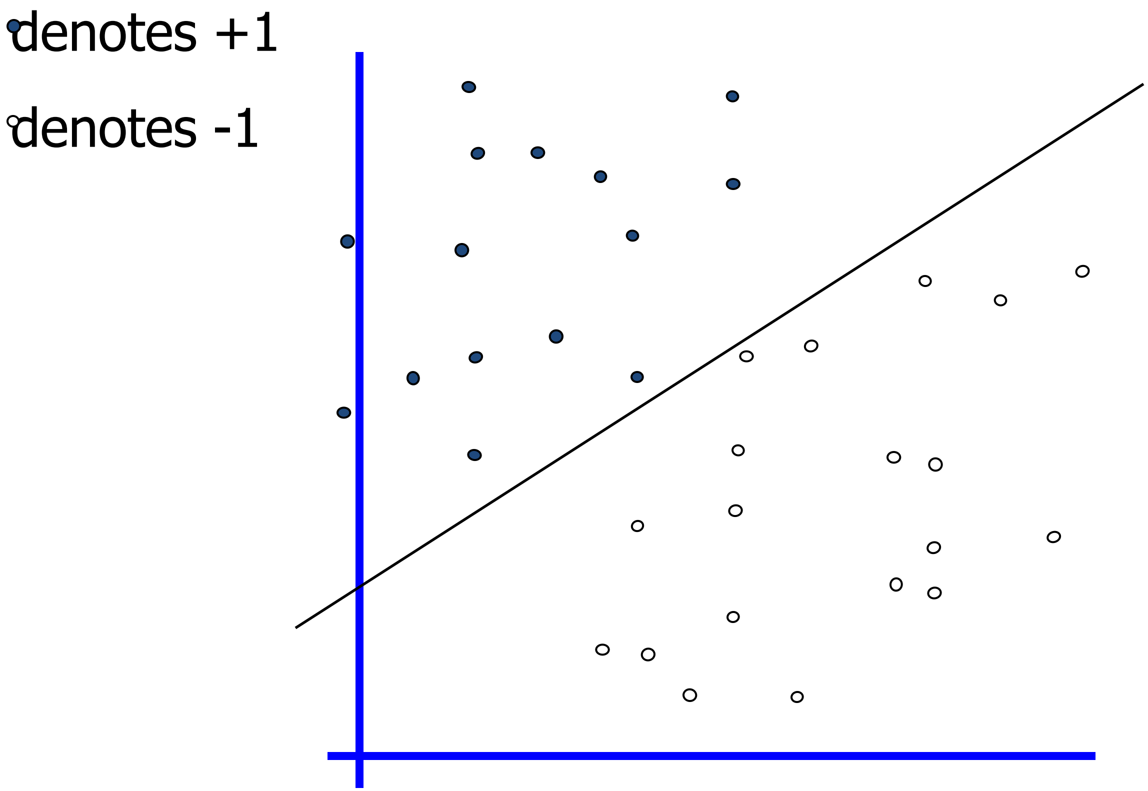
Linear Classifiers

- denotes +1
- denotes -1



How would you classify this data?

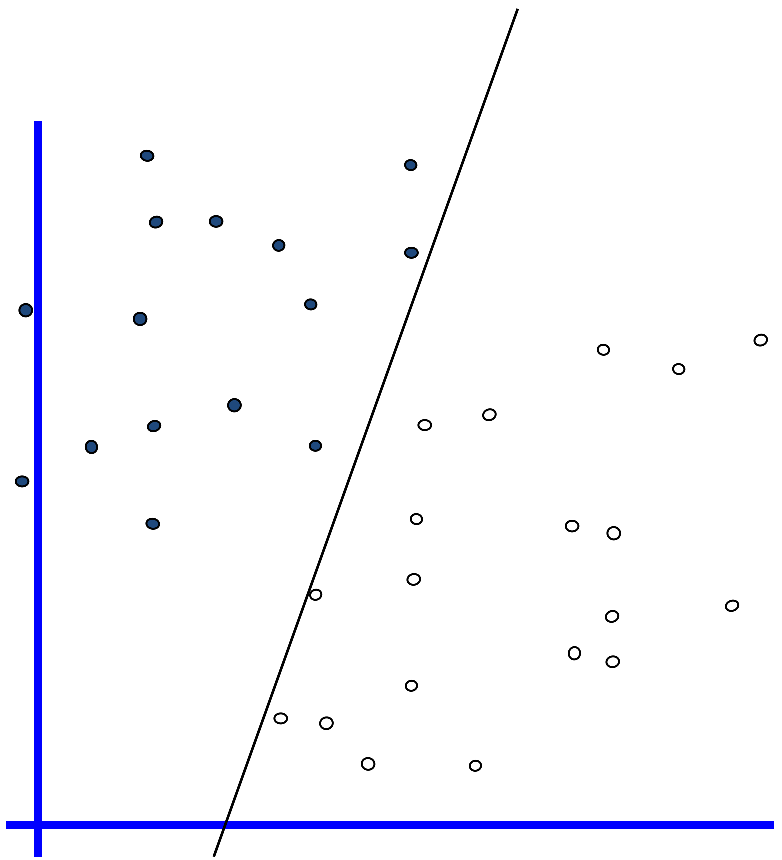
Linear Classifiers



How would you classify this data?

Linear Classifiers

- denotes +1
- denotes -1

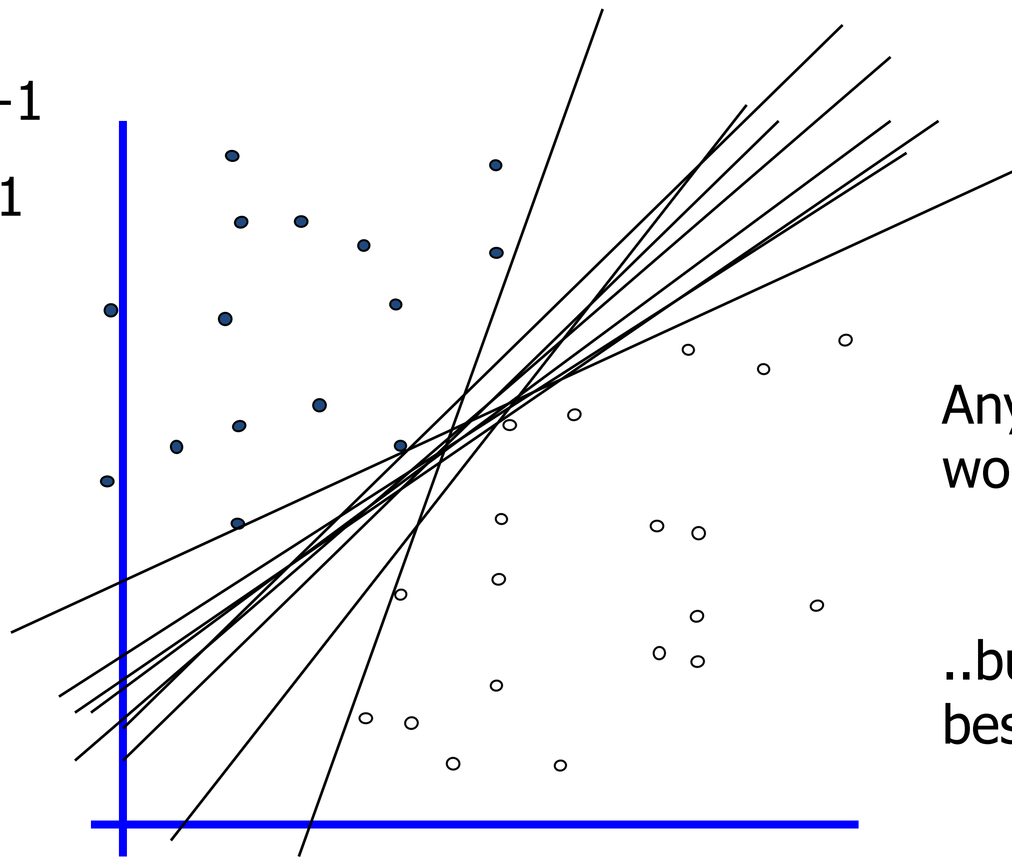


How would you classify this data?

Linear Classifiers

denotes +1

denotes -1

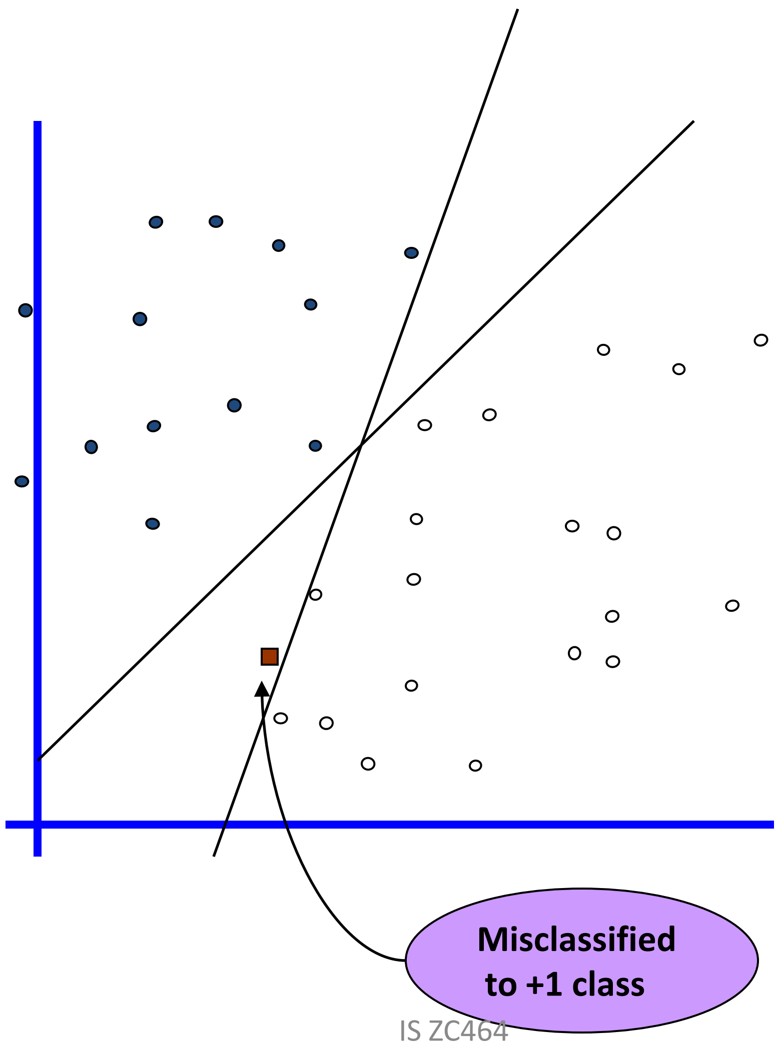


Any of these would be fine..

..but which is best?

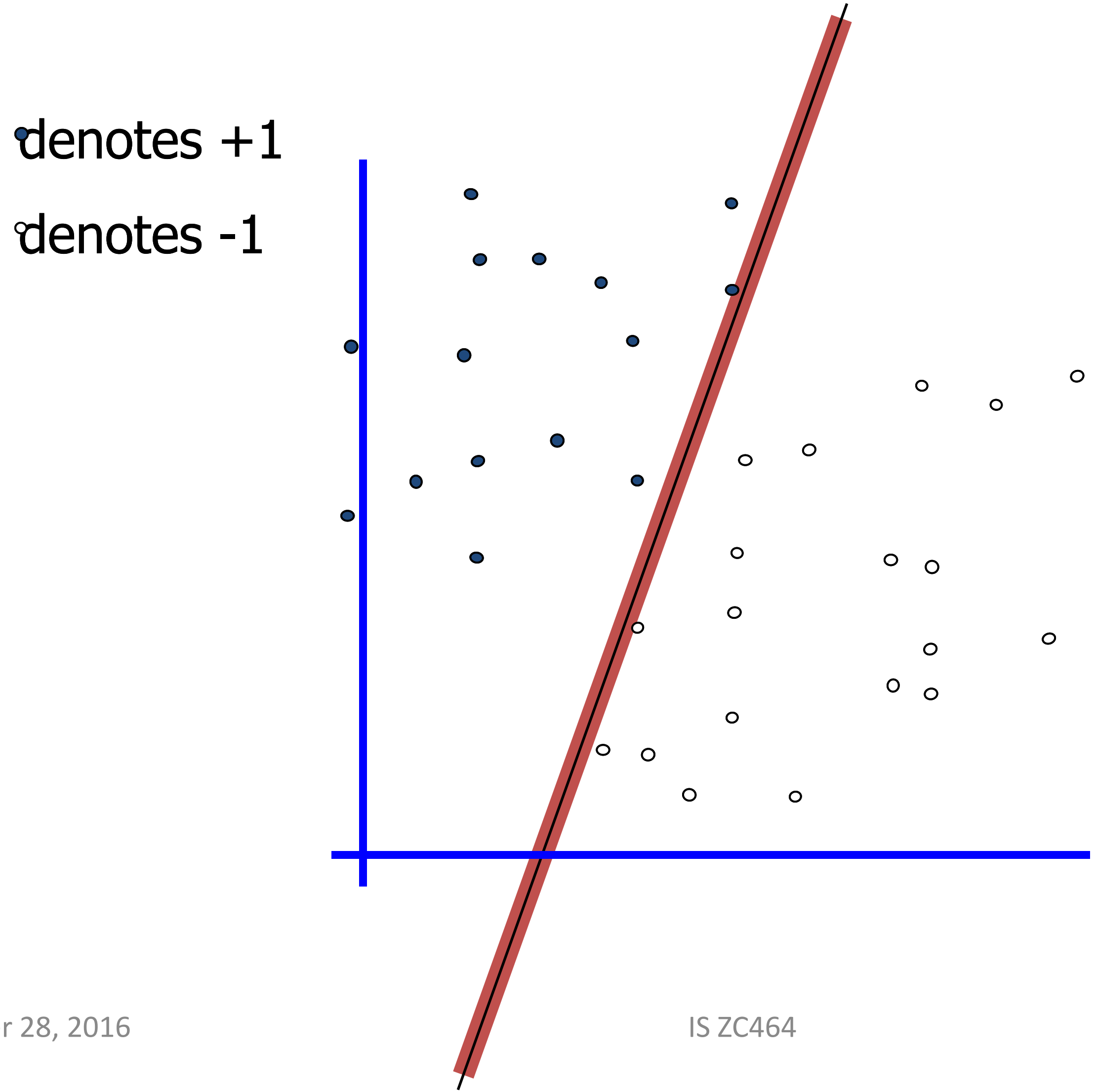
Linear Classifiers

- denotes +1
- denotes -1



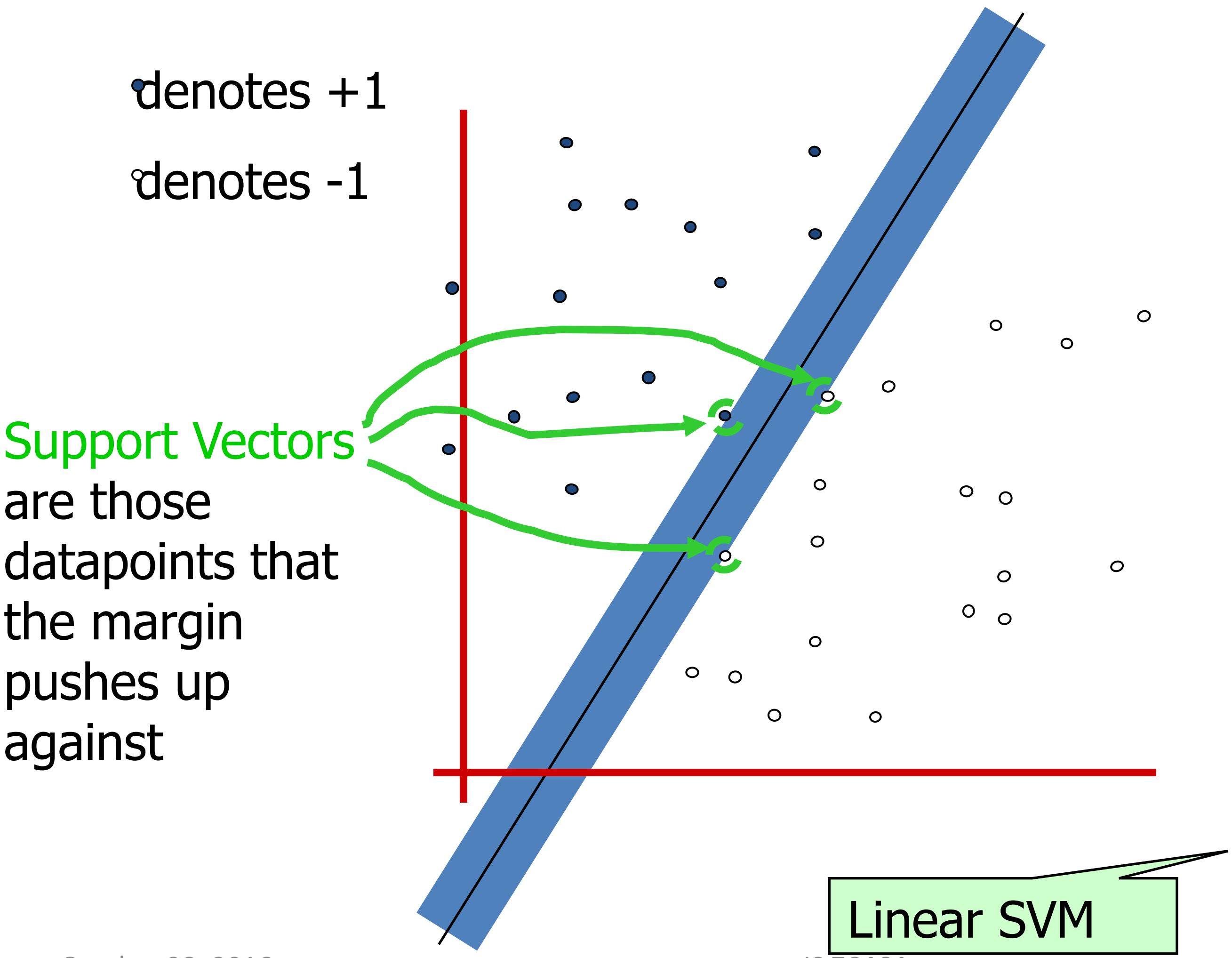
How would you classify this data?

Classifier Margin



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

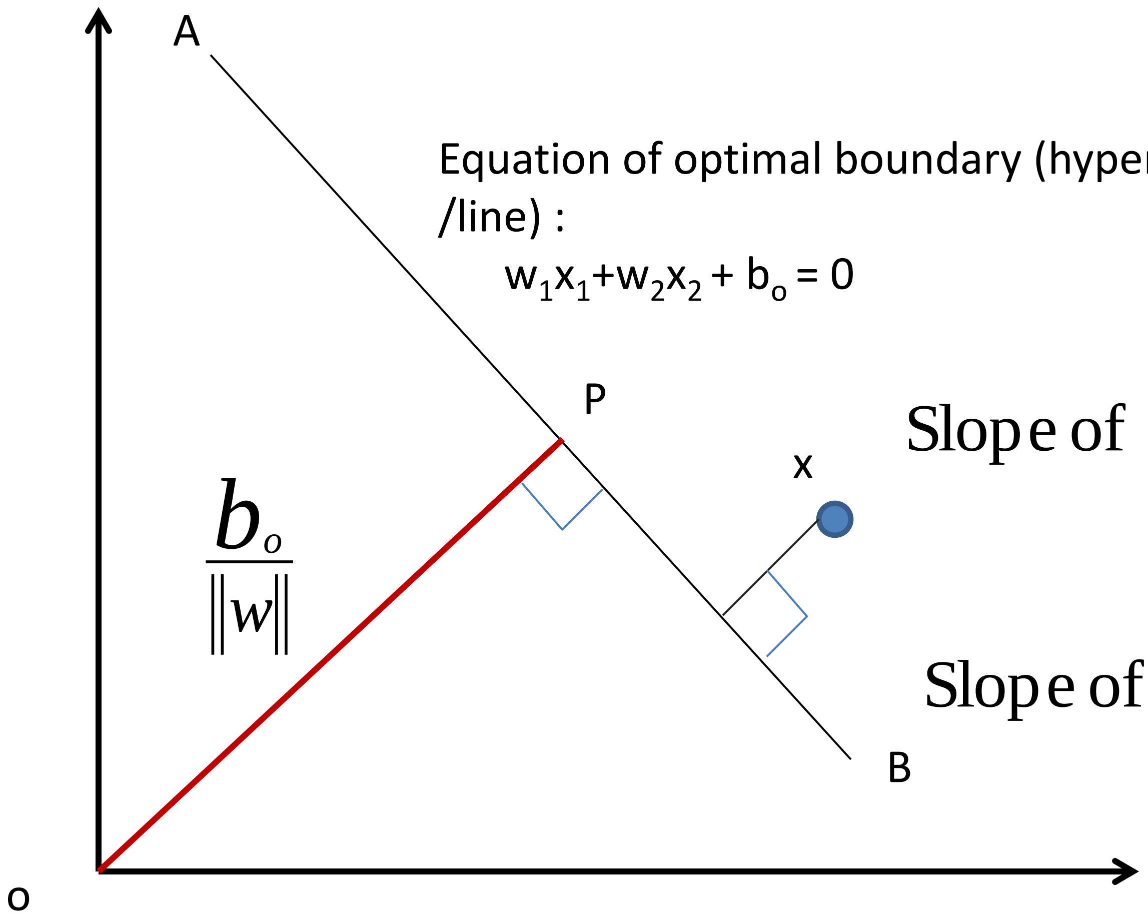
Maximum Margin



The **maximum margin linear classifier** is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Distance of a point on the decision boundary from origin



Equation of optimal boundary (hyperplane /line) :

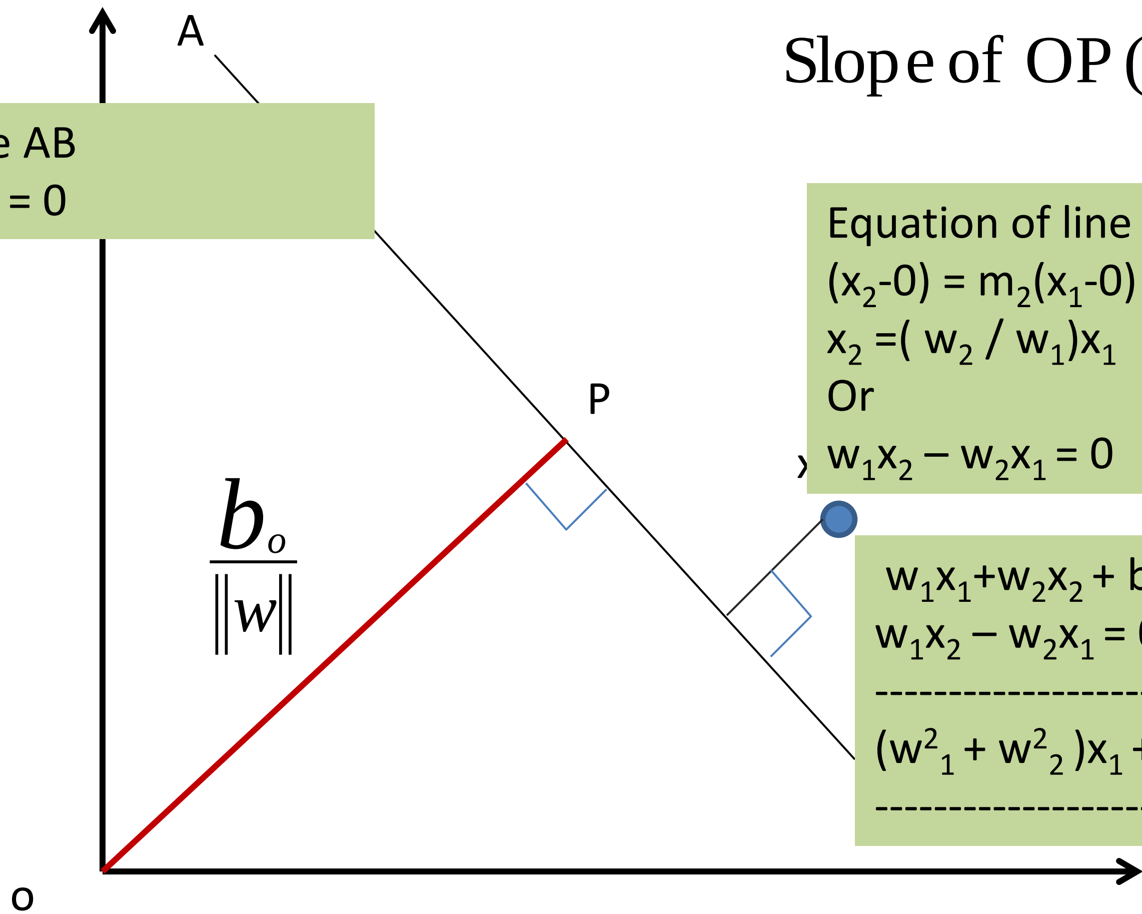
$$w_1x_1+w_2x_2 + b_o = 0$$

$$\text{Slope of AB (m1)} = -\frac{w_1}{w_2}$$

$$\text{Slope of OP (m2)} = \frac{w_2}{w_1}$$

Distance of a point on the decision boundary from origin

$$\text{Slope of OP } (m2) = \frac{w_2}{w_1}$$



Equation of line AB

$w_1x_1+w_2x_2 + b_o = 0$

Equation of line OP

$(x_2-0) = m_2(x_1-0)$

$x_2 =(w_2 / w_1)x_1$

Or

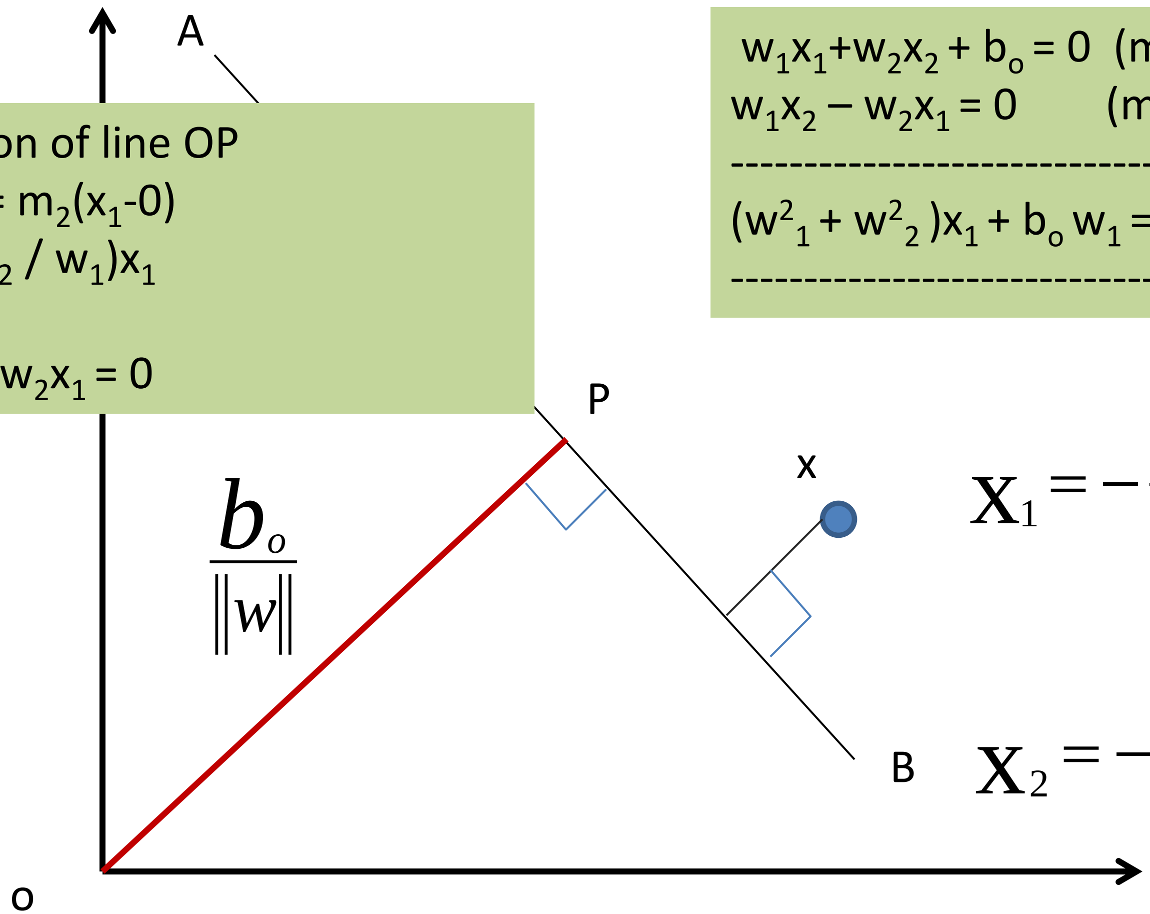
$w_1x_2 – w_2x_1 = 0$

$w_1x_1+w_2x_2 + b_o = 0$ (multiply by w_1)

$w_1x_2 – w_2x_1 = 0$ (multiply by w_2)

$(w^2_1 + w^2_2)x_1 + b_o w_1 = 0$

Distance of a point on the decision boundary from origin



Equation of line OP

$(x_2-0) = m_2(x_1-0)$

$x_2 = (w_2 / w_1)x_1$

Or

$w_1x_2 - w_2x_1 = 0$

$w_1x_1+w_2x_2 + b_o = 0$ (multiply by w_1)

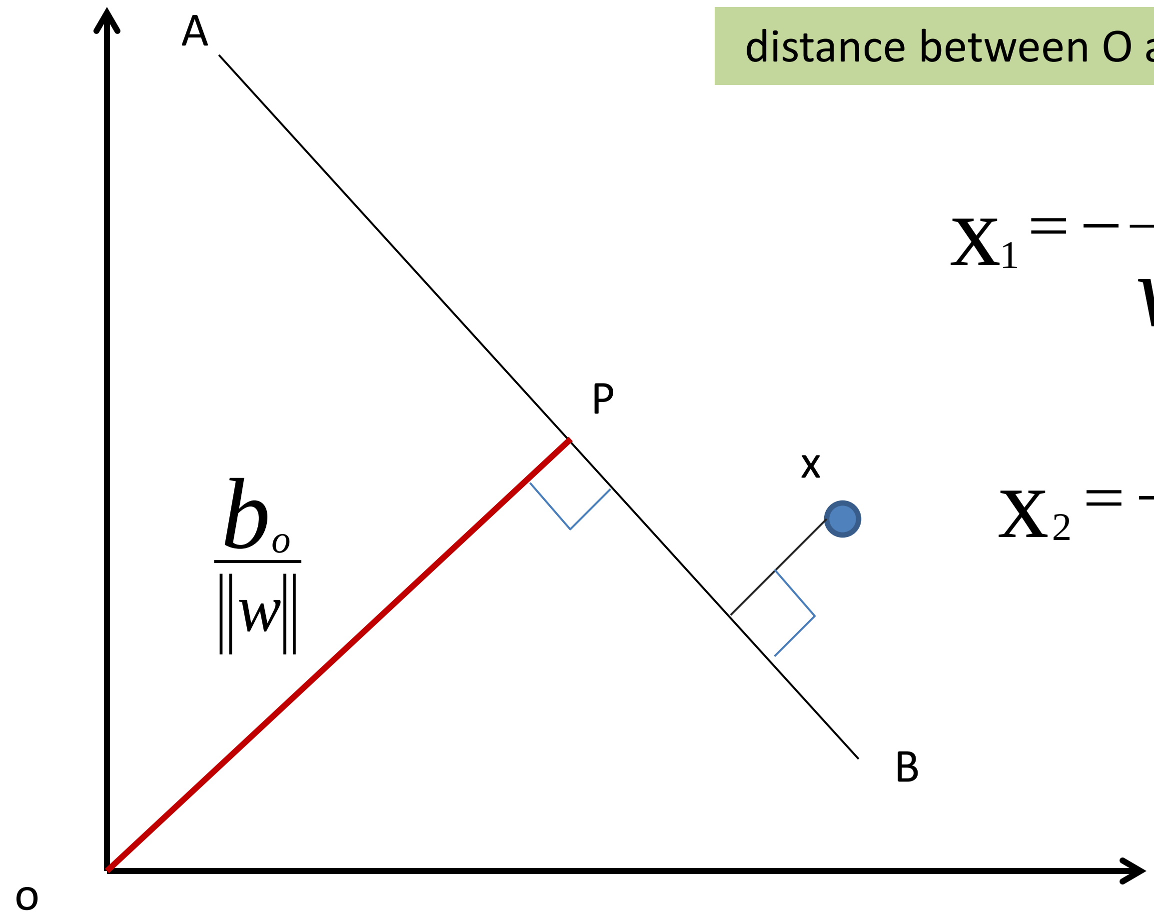
$w_1x_2 - w_2x_1 = 0$ (multiply by w_2)

$(w^2_1 + w^2_2)x_1 + b_o w_1 = 0$

$$X_1 = -\frac{b_o w_1}{w_1^2 + w_2^2}$$

$$X_2 = -\frac{b_o w_2}{w_1^2 + w_2^2}$$

Distance of a point on the decision boundary from origin



distance between O and P = d (say)

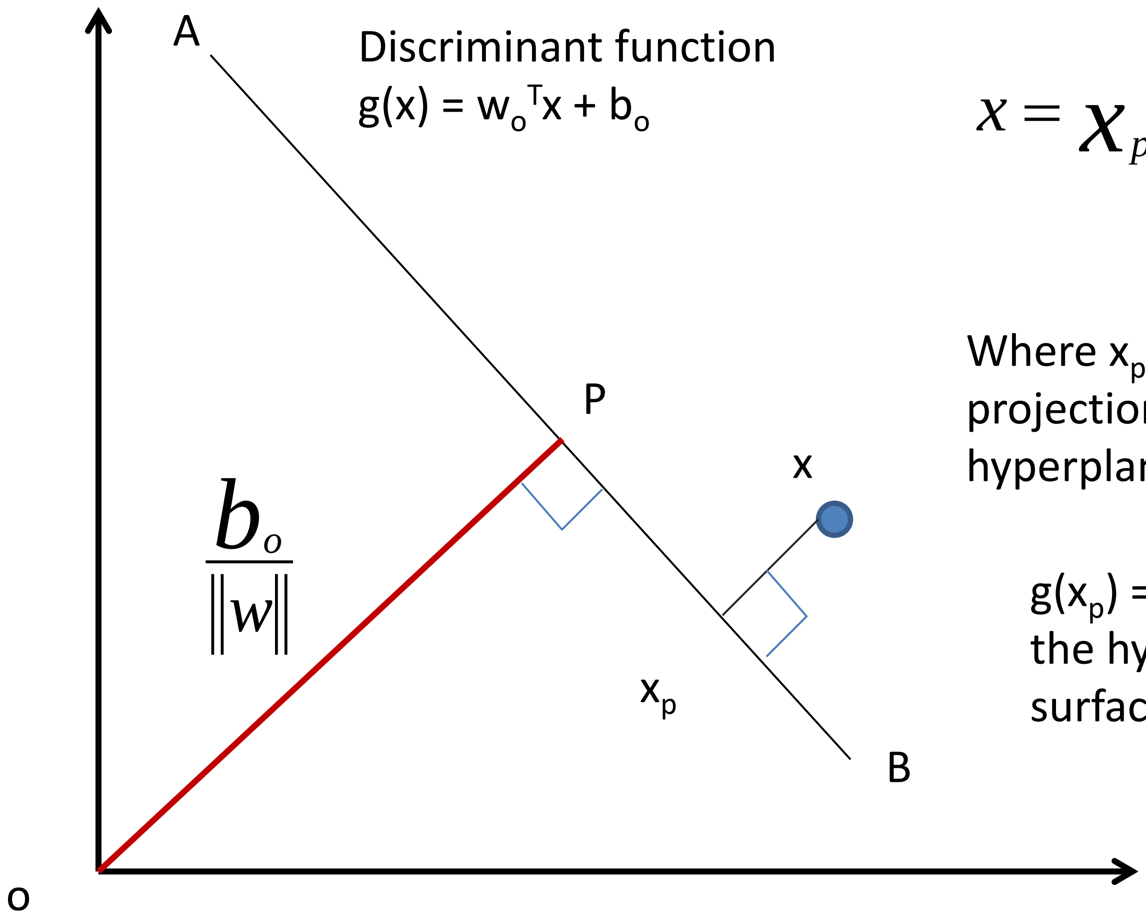
$$x_1 = -\frac{b_o w_1}{w_1^2 + w_2^2}$$

$$x_2 = -\frac{b_o w_2}{w_1^2 + w_2^2}$$

$$d = \frac{b_o}{\sqrt{w_1^2 + w_2^2}}$$

Distance of a point in a class from optimal hyperplane

Discriminant function
$g(x) = w_o^T x + b_o$



$$x = x_p + r\frac{w_o}{\|w\|}$$

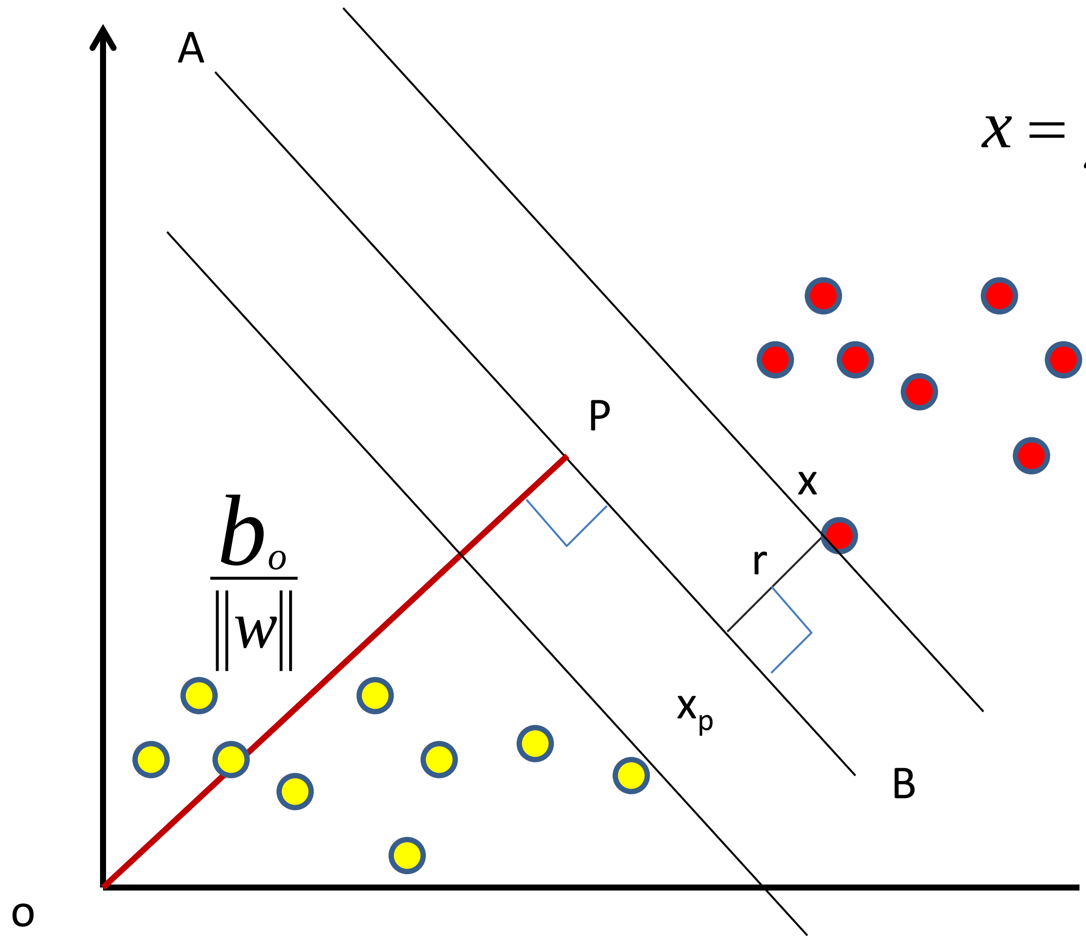
Where x_p is the normal projection of x on the hyperplane

$g(x_p) = w^T x + b_o = 0$ on the hyperplane/Decision surface/line

Margin

$g(x) = w_o^T x + b_o$

$$x = x_p + r\frac{w_o}{\|w\|}$$

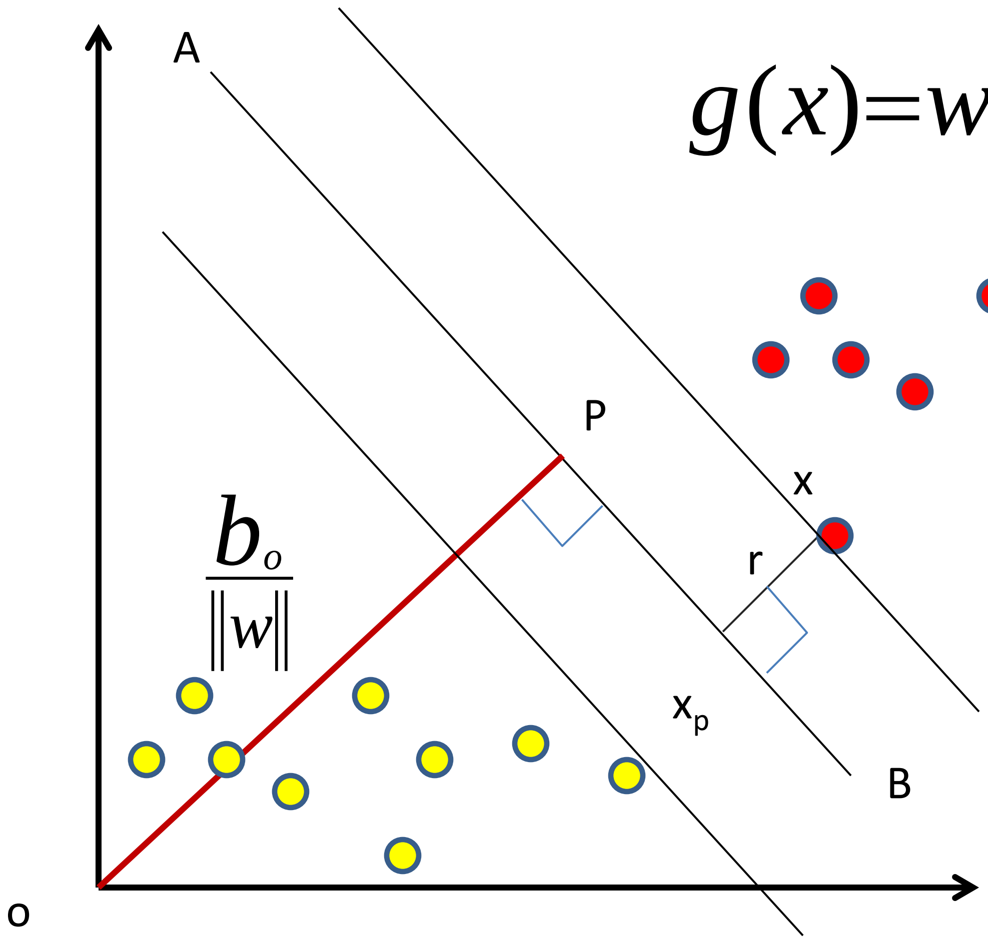


r: positive if x is on the positive side of the hyperplane

r: negative if x is on the negative side of the hyperplane

Margin

$g(x) = w_o^T x + b_o$



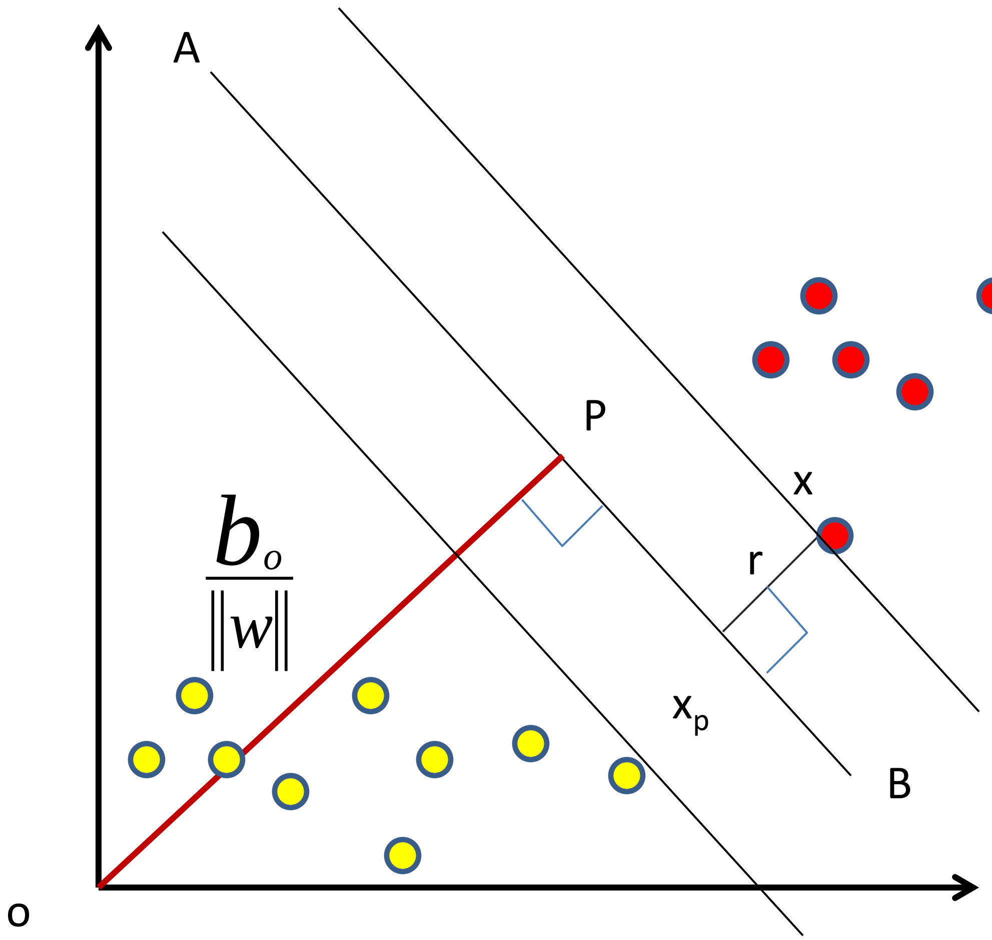
$$g(x) = w_o^T x_p + r\, w_o^T \frac{w_o}{\|w\|} + b_o$$

Since xp is on the hyperplane,
$g(x_p) = w_o^T x_p + b_o = 0$

Therefore

$$g(x) = r\, w_o^T \frac{w_o}{\|w\|}$$

Margin



$$g(x) = r\, w_o^T \frac{w_o}{\|w\|}$$

$$g(x) = r\|w_o\|$$

$$r = \frac{g(x)}{\|w_o\|}$$

Total margin to optimize

Considering $g(x^s) = +1$ for the support vector x^s for which the class is +1

Similarly,

$g(x^s) = -1$ for the support vector x^s for which the class is -1

Therefore, for both support vectors

$$r = \frac{+1}{\|w_o\|} \qquad r = \frac{-1}{\|w_o\|}$$

Total margin to optimize

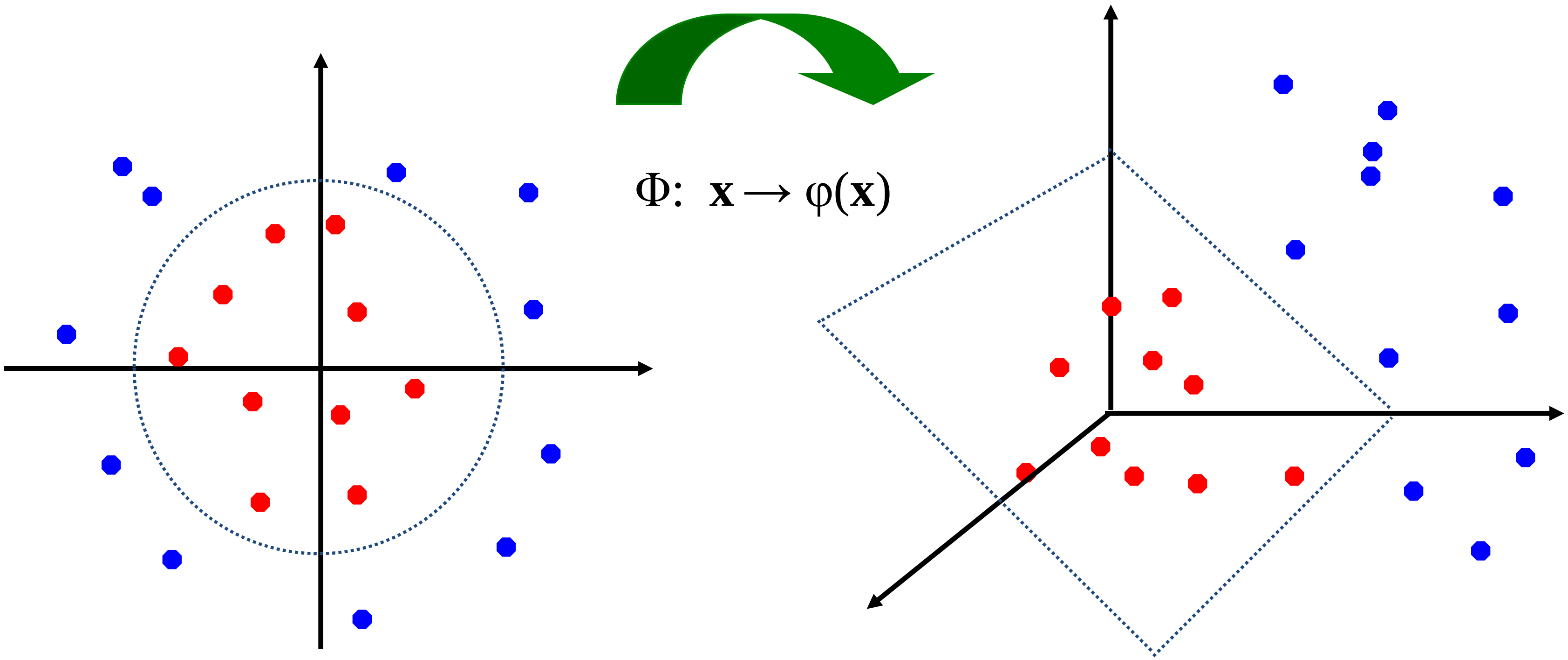
Total margin to optimize is

$$\rho = \frac{2}{\|w_o\|}$$

Maximize ρ
Or equivalently Minimize the Euclidean norm of weight vector w

Non-linear SVMs: Feature spaces

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



$$\Phi: \mathbf{x} \rightarrow \varphi(\mathbf{x})$$

Examples of Kernel Functions

- Linear: $K(\mathbf{x_i},\mathbf{x_j}) = \mathbf{x_i}^T\mathbf{x_j}$

- Polynomial of power p: $K(\mathbf{x_i},\mathbf{x_j}) = (1 + \mathbf{x_i}^T\mathbf{x_j})^p$

- Gaussian (radial-basis function network):

$$K(\mathbf{x_i},\mathbf{x_j}) = \exp\left(-\frac{\left\|\mathbf{x_i} - \mathbf{x_j}\right\|^2}{2\sigma^2}\right)$$

- Sigmoid: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0\mathbf{x_i}^T\mathbf{x_j} + \beta_1)$

SVM Applications

- **SVM has been used successfully in many real-world problems**

  - text (and hypertext) categorization

  - image classification

  - bioinformatics (Protein classification, Cancer classification)

  - hand-written character recognition

Some Issues

- **Choice of kernel**
  - **Gaussian or polynomial kernel is default**
  - **if ineffective, more elaborate kernels are needed**
  - **domain experts can give assistance in formulating appropriate similarity measures**

- **Choice of kernel parameters**
  - **e.g. σ in Gaussian kernel**
  - **σ is the distance between closest points with different classifications**
  - **In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters.**