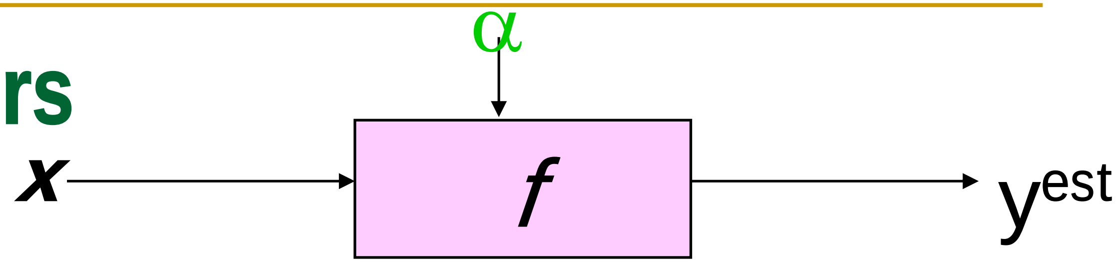
Support Vector Machines (SVM)

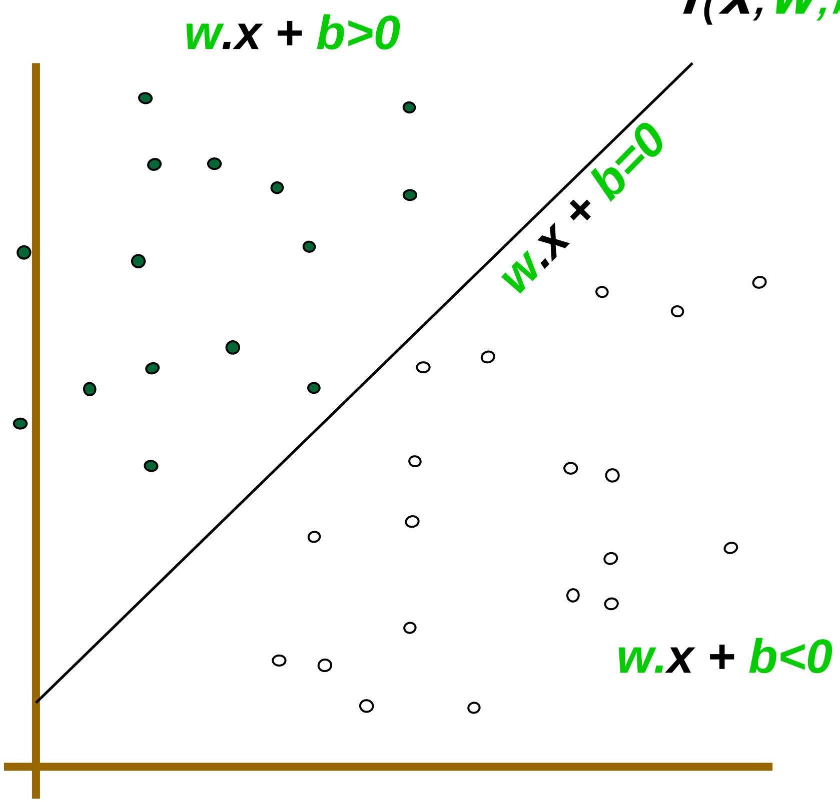
Part of the slides are taken from
Prof. Andrew Moore's SVM tutorial

Linear Classifiers



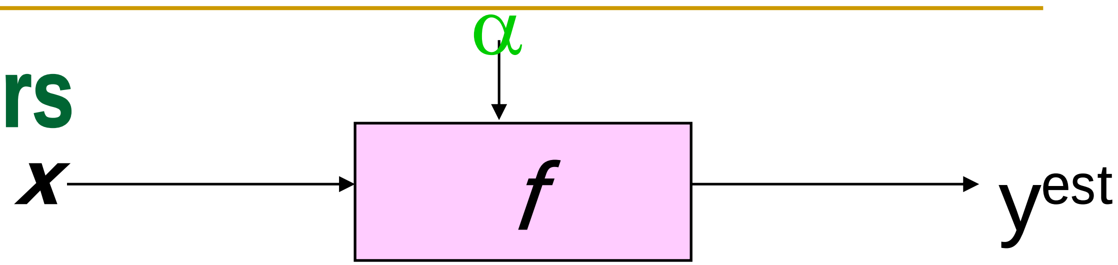
$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}.\mathbf{x} + b)$

- denotes +1
- denotes -1



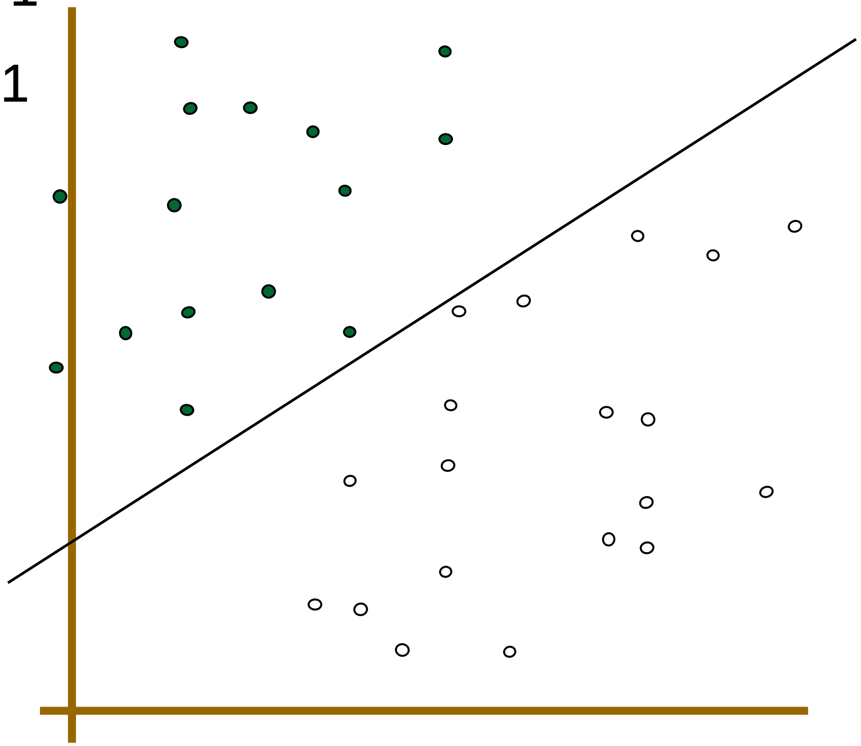
How would you classify this data?

Linear Classifiers



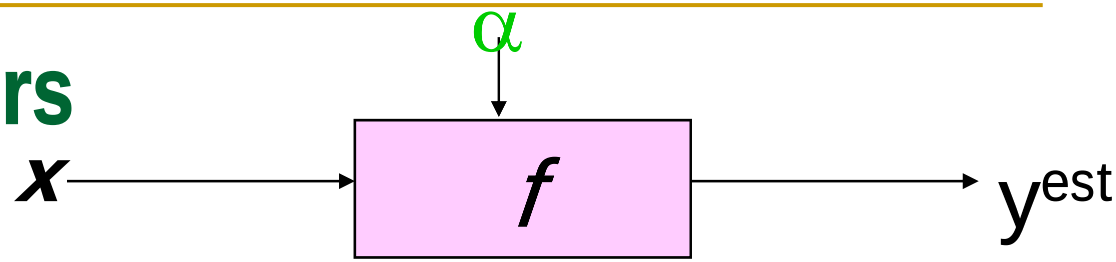
$$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} + b)$$

- denotes +1
- denotes -1



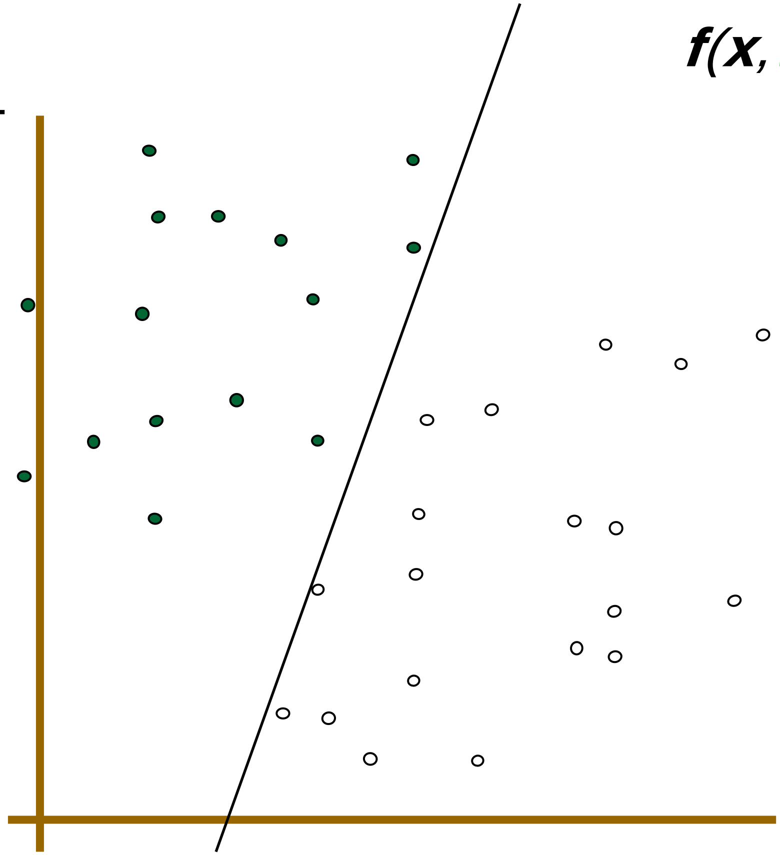
How would you classify this data?

Linear Classifiers



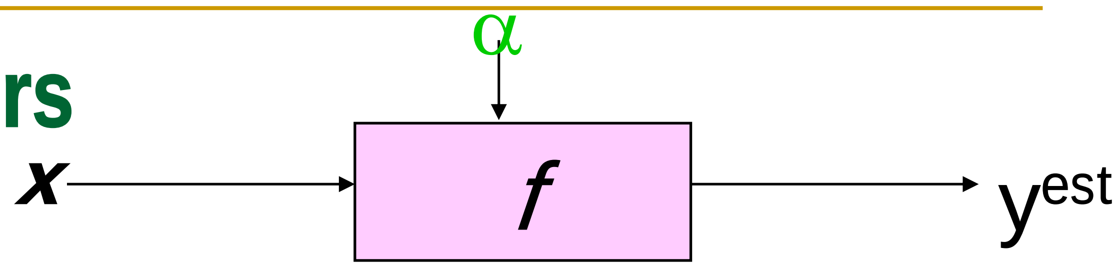
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} + b)$

- denotes +1
- denotes -1



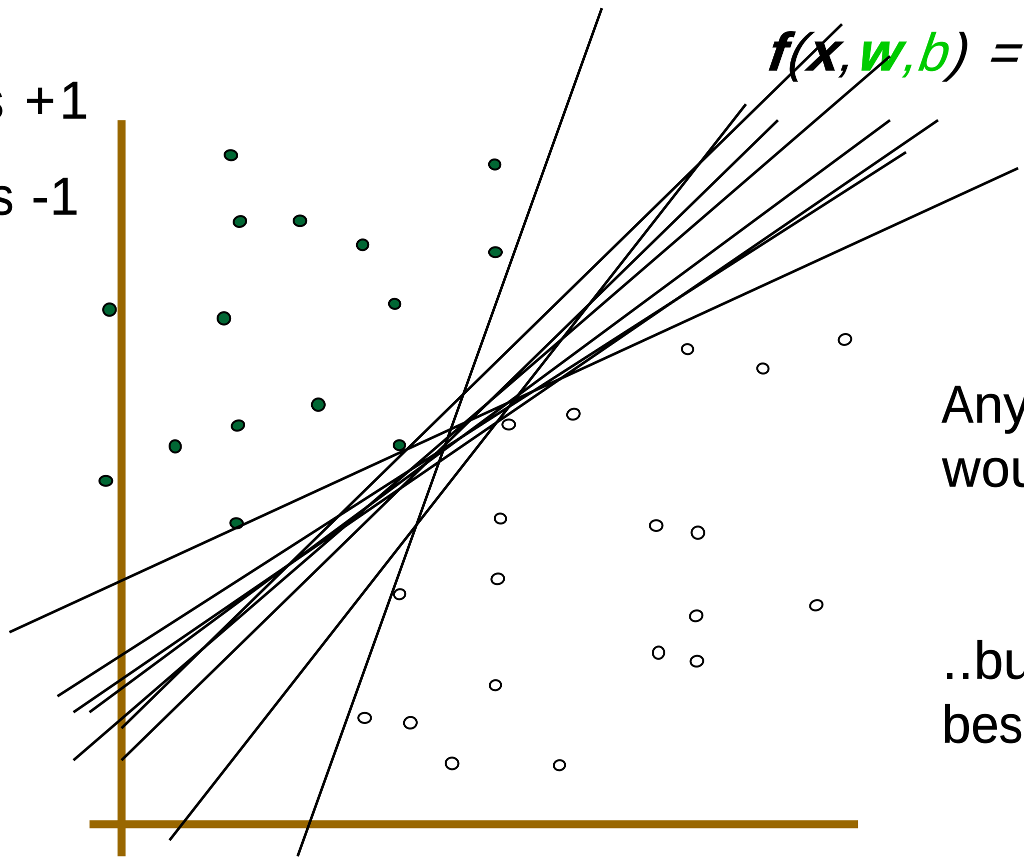
How would you classify this data?

Linear Classifiers



$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w} . \mathbf{x} + b)$

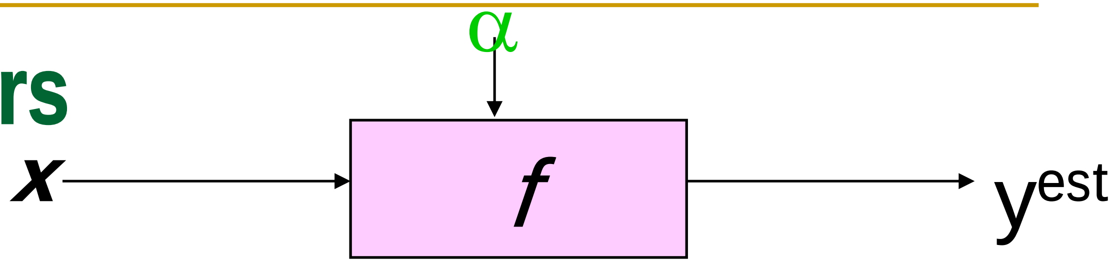
- denotes +1
- denotes -1



Any of these would be fine..

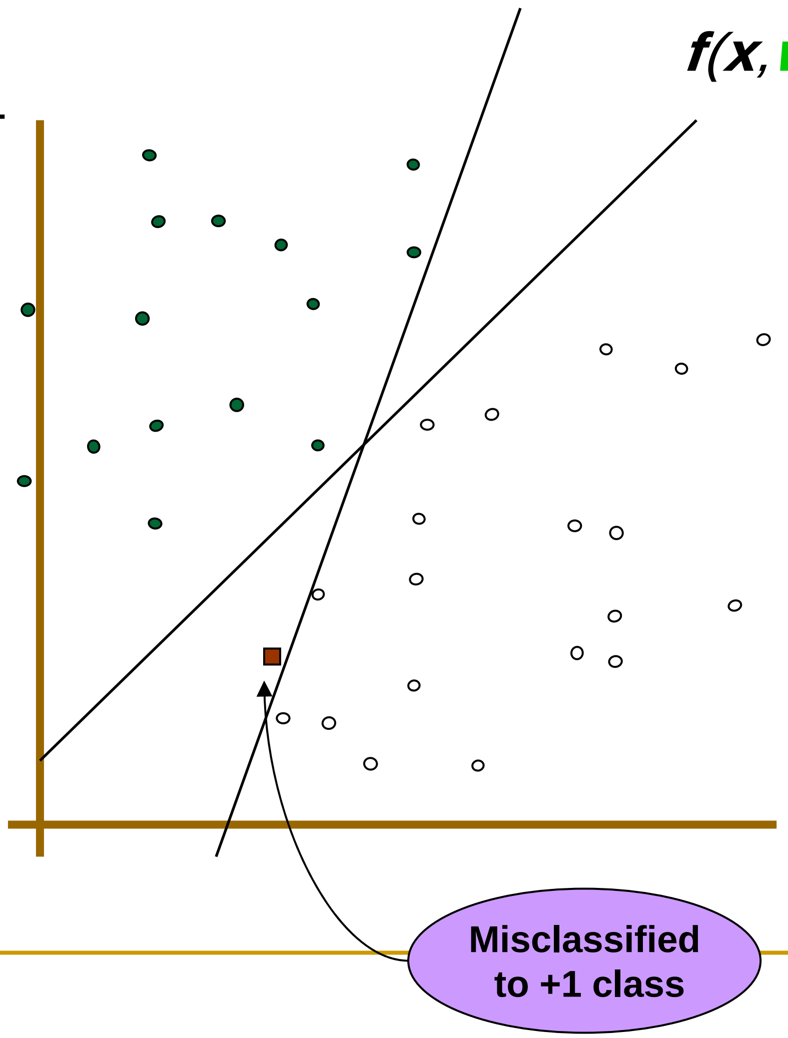
..but which is best?

Linear Classifiers



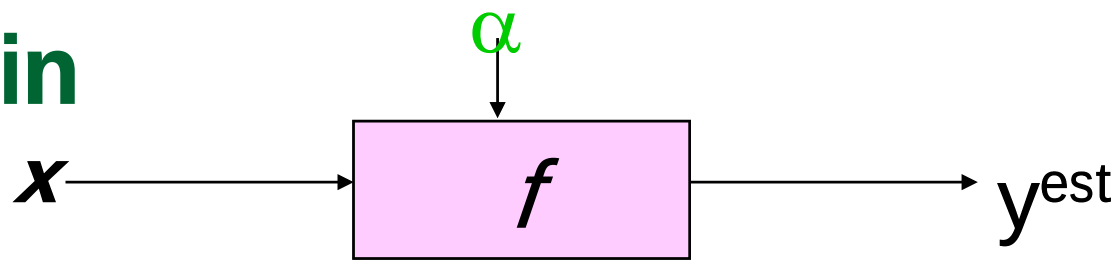
$$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} + b)$$

- denotes +1
- denotes -1

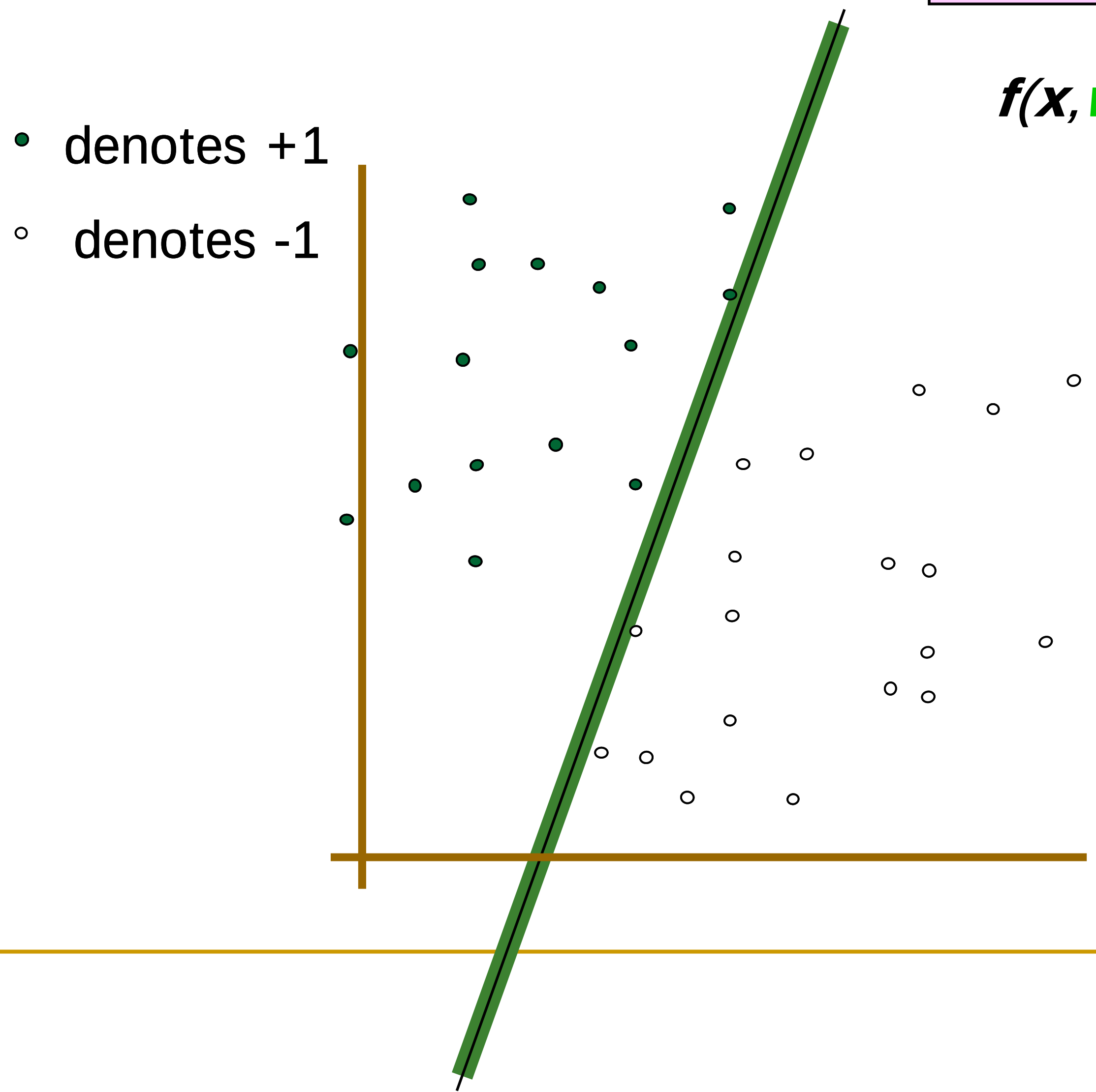


How would you classify this data?

Classifier Margin

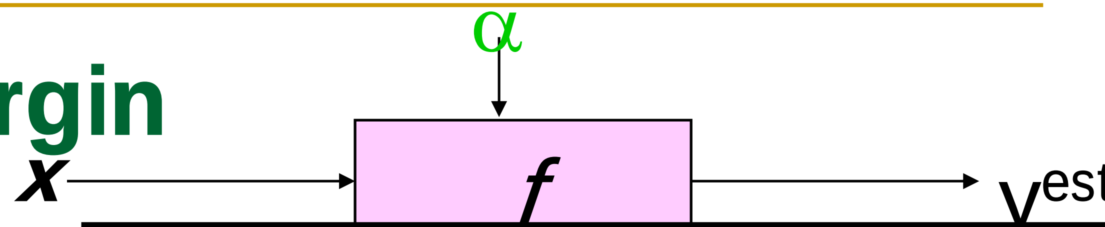


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} + b)$



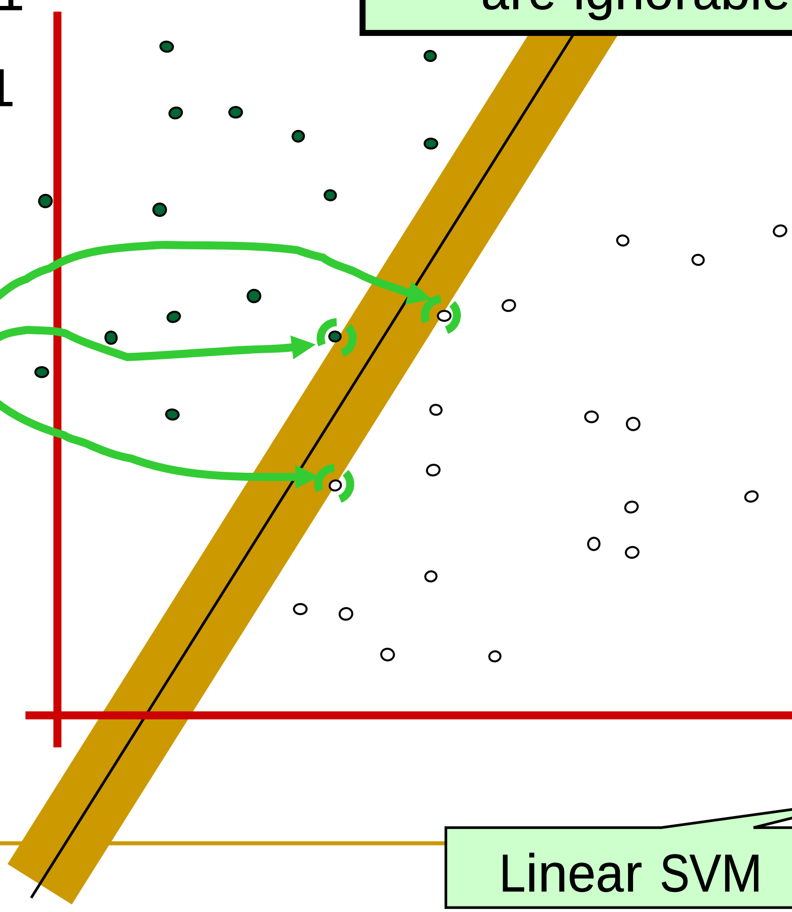
Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a data point.

Maximum Margin



1. Implies that only support vectors are important; other training examples are ignorable.

- denotes +1
- denotes -1



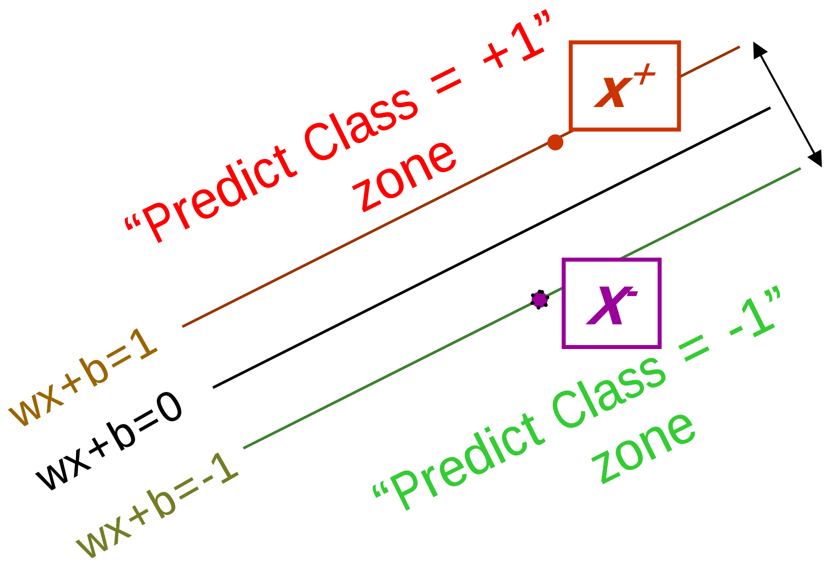
Support Vectors are those datapoints that the margin pushes up against

The maximum margin linear classifier is the linear classifier with the, a maximum margin.

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

Linear SVM

Linear SVM Mathematically



M=Margin Width

What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{w} \cdot (\mathbf{x}^+ - \mathbf{x}^-) = 2$

$$M = \frac{(\mathbf{x}^+ - \mathbf{x}^-) \cdot \mathbf{w}}{|\mathbf{w}|} = \frac{2}{|\mathbf{w}|}$$

Linear SVM Mathematically

- **Goal: 1) Correctly classify all training data**

$$\mathbf{w}.\mathbf{x}_i + b \geq 1 \qquad \textit{if } y_i = +1$$

$$\mathbf{w}.\mathbf{x}_i + b \leq 1 \qquad \textit{if } y_i = -1$$

$$y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1 \qquad \text{for all i}$$

2) Maximize the Margin $M = \dfrac{2}{|\mathbf{w}|}$

same as minimize $\dfrac{1}{2}\mathbf{w}^t\mathbf{w}$

- **We can formulate a Quadratic Optimization Problem and solve for w and b**

- **Minimize** $\Phi(\mathbf{w}) = \dfrac{1}{2}\mathbf{w}^t\mathbf{w}$

  subject to $y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1 \qquad \forall i$

Solving the Optimization Problem

Find **w** and b such that
$\Phi(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x}_i, y_i)\}$: $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1$

- **Need to optimize a *quadratic* function subject to *linear* constraints.**
- **Many algorithms exist for solving the optimization problem.**
- **Constrained Optimization problem with Lagrangian :**

$$L(\mathbf{w},b,\alpha)=\tfrac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\alpha_i(y_i\cdot((\mathbf{x}_i\cdot\mathbf{w})+b)-1)$$

$$\frac{\partial}{\partial b}L(\mathbf{w},b,\alpha)=0 \quad \frac{\partial}{\partial \mathbf{w}}L(\mathbf{w},b,\alpha)=0$$

Solving the Optimization Problem

Primal variables vanish **:**

$$\sum_{i=1}^{l} a_i y_i = 0 \qquad \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$$

Support Vectors whose α_i are nonzero

The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:

Find $\alpha_1 \dots \alpha_l$ such that

$\mathbf{Q}(\alpha) = \Sigma\, \alpha_i - ½ \Sigma \Sigma\, \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\mathbf{T}} \mathbf{x}_j$ is maximized and

(1) $\Sigma\, \alpha_i y_i = 0$

(2) $\alpha_i \geqslant 0$ for all α_i

The Optimization Problem Solution

- The solution has the form:

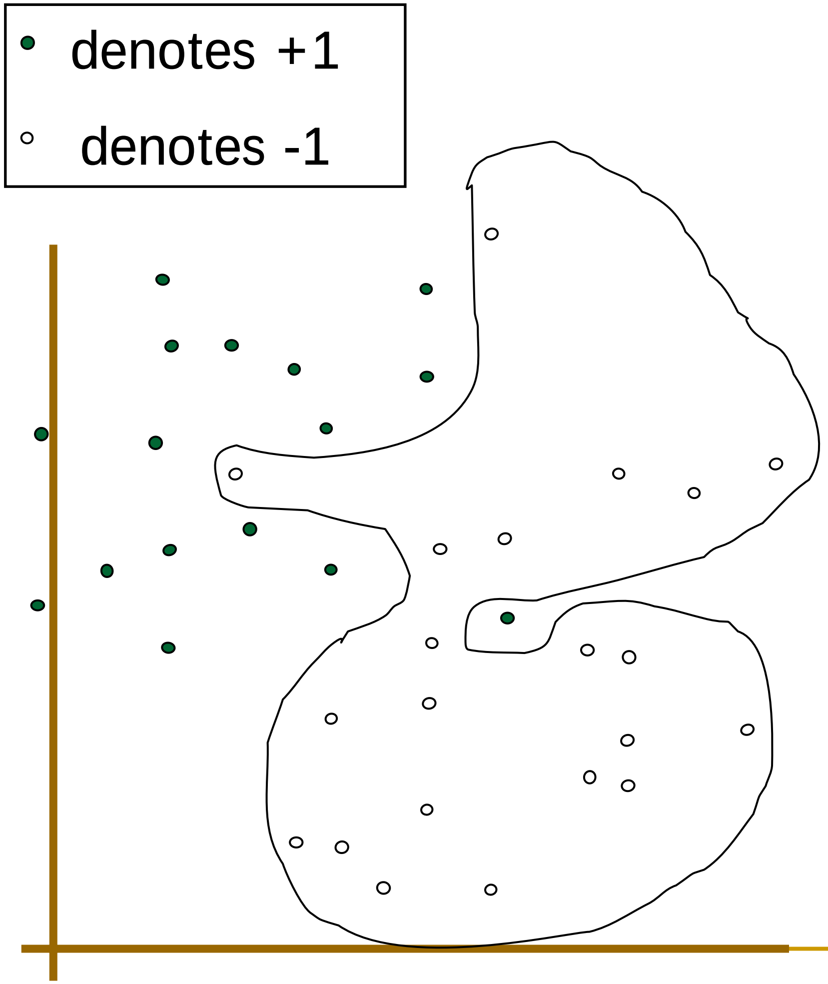
$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x}_i \qquad b = y_k - \mathbf{w}^T\mathbf{x}_k \text{ for any } \mathbf{x}_k \text{ such that } \alpha_k \neq 0$$

- Each non-zero α_i indicates that corresponding $\mathbf{x}_i$ is a support vector.
- Then the classifying function will have the form:

$$f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

- Notice that it relies on an *inner product* between the test point $\mathbf{x}$ and the support vectors $\mathbf{x}_i$.
- Notice that solving the optimization problem involves computing the inner products $\mathbf{x}_i^T\mathbf{x}_j$ between all pairs of training points.

Data set with noise



- **Hard Margin:** **So far we require all data points be classified correctly**

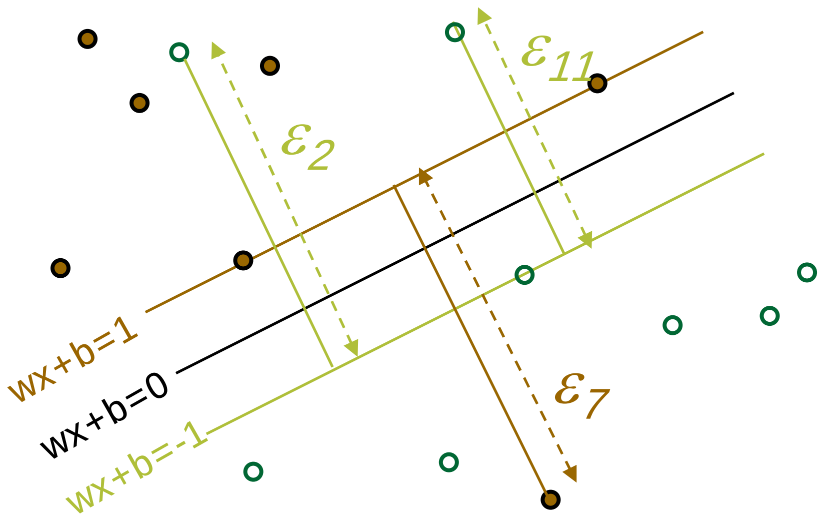
  - No training error

- **What if the training set is noisy?**

OVERFITTING!

Soft Margin Classification

Slack variables ξ_i **can be added to allow misclassification of difficult or noisy examples.**



What should our quadratic optimization criterion be?

Minimize

$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{l}\varepsilon_k$$

Hard Margin v.s. Soft Margin

- **The old formulation:**

> Find $\mathbf{w}$ and b such that
> $\Phi(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x}_i, y_i)\}$
> $y_i (\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1$

- **The new formulation incorporating slack variables:**

> Find $\mathbf{w}$ and b such that
> $\Phi(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w} + C \sum \xi_i$ is minimized and for all $\{(\mathbf{x}_i, y_i)\}$
> $y_i (\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1 - \xi_i$ and $\xi_i \geqslant 0$ for all i

- **Parameter *C* can be viewed as a way to control overfitting.**

Linear SVM: Overview

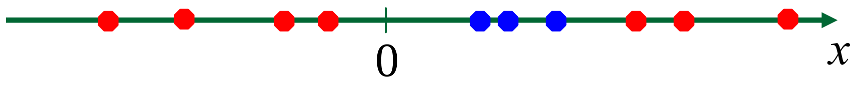
- The classifier is a *separating hyper-plane.*
- Most "important" training points are support vectors; they define the hyper-plane.
- Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i
- Training points appear only inside dot products

Find $\alpha_1 \ldots \alpha_N$ **such that**
$Q(\alpha) = \Sigma \alpha_i - \frac{1}{2} \Sigma \Sigma \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i^T \mathbf{x}_j}$ **is maximized and**
(1) $\Sigma \alpha_i y_i = 0$
(2) $0 \leqslant \alpha_i \leqslant C$ **for all** α_i

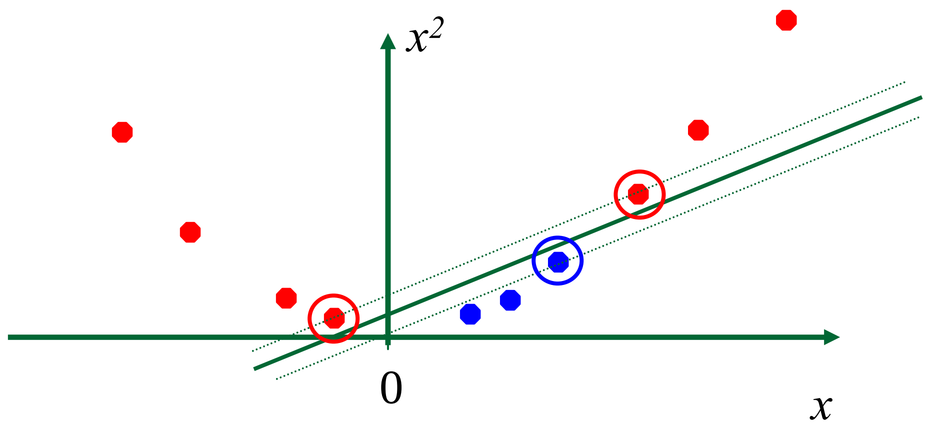
$$f(\mathbf{x}) = \Sigma \alpha_i y_i \boxed{\mathbf{x}_i^T \mathbf{x}} + b$$

Non-linear SVM

- What are we going to do if the dataset is not linearly separable?

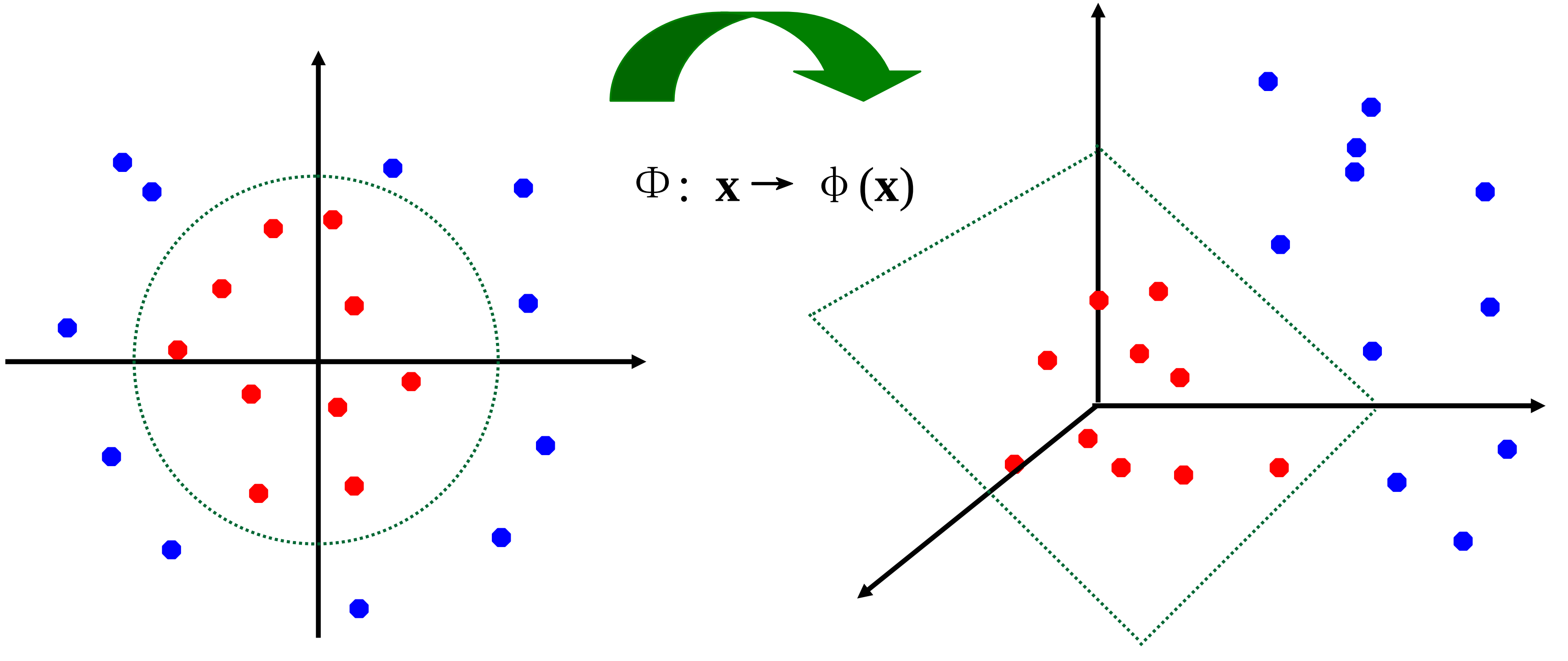


- How about… mapping data to a higher-dimensional space:



Non-linear SVM: Feature spaces

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



The "Kernel Trick"

- **The linear classifier relies on dot product between vectors $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$**
- **If every data point is mapped into a high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \phi(\mathbf{x})$, the dot product becomes:**

$$K(\mathbf{x}_i,\mathbf{x}_j)= \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

- **A *kernel function* is some function that corresponds to an inner product in some expanded feature space.**
- **Example:**

2-dimensional vectors $\mathbf{x}=[x_1\ \ x_2]$; let $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2$,

Need to show that $K(\mathbf{x}_i,\mathbf{x}_j)= \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$:

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2,$$

$$= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1\ \ x_{i1}^2\ \ \sqrt{2}\, x_{i1}x_{i2}\ \ x_{i2}^2\ \ \sqrt{2}x_{i1}\ \ \sqrt{2}x_{i2}]^T [1\ \ x_{j1}^2\ \ \sqrt{2}\, x_{j1}x_{j2}\ \ x_{j2}^2\ \ \sqrt{2}x_{j1}\ \ \sqrt{2}x_{j2}]$$

$$= \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \quad \text{where } \phi(\mathbf{x}) = [1\ \ x_1^2\ \ \sqrt{2}\, x_1x_2\ \ x_2^2\ \ \sqrt{2}x_1\ \ \sqrt{2}x_2]$$

What Functions are Kernels?

- For some functions $K(\mathrm{x}_i,\mathrm{x}_j)$ checking that

  $K(\mathrm{x}_i,\mathrm{x}_j) = \phi(\mathrm{x}_i)^T \phi(\mathrm{x}_j)$ can be cumbersome.
- Mercer's theorem:

  Every semi-positive definite symmetric function is a kernel
- Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

K=

$K(\mathbf{x}_1,\mathbf{x}_1)$	$K(\mathbf{x}_1,\mathbf{x}_2)$	$K(\mathbf{x}_1,\mathbf{x}_3)$	…	$K(\mathbf{x}_1,\mathbf{x}_N)$
$K(\mathbf{x}_2,\mathbf{x}_1)$	$K(\mathbf{x}_2,\mathbf{x}_2)$	$K(\mathbf{x}_2,\mathbf{x}_3)$		$K(\mathbf{x}_2,\mathbf{x}_N)$
…	…	…	…	…
$K(\mathbf{x}_N,\mathbf{x}_1)$	$K(\mathbf{x}_N,\mathbf{x}_2)$	$K(\mathbf{x}_N,\mathbf{x}_3)$	…	$K(\mathbf{x}_N,\mathbf{x}_N)$

Examples of Kernel Functions

- Linear: $K(\mathbf{x}_i,\mathbf{x}_j)= \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j$

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

- 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: $K(\mathbf{x}_i,\mathbf{x}_j)= \tanh(\beta_0 \mathbf{x}_i^{\mathbf{T}}\mathbf{x}_j + \beta_1)$

Non-linear SVM Mathematically

- **Dual problem formulation:**

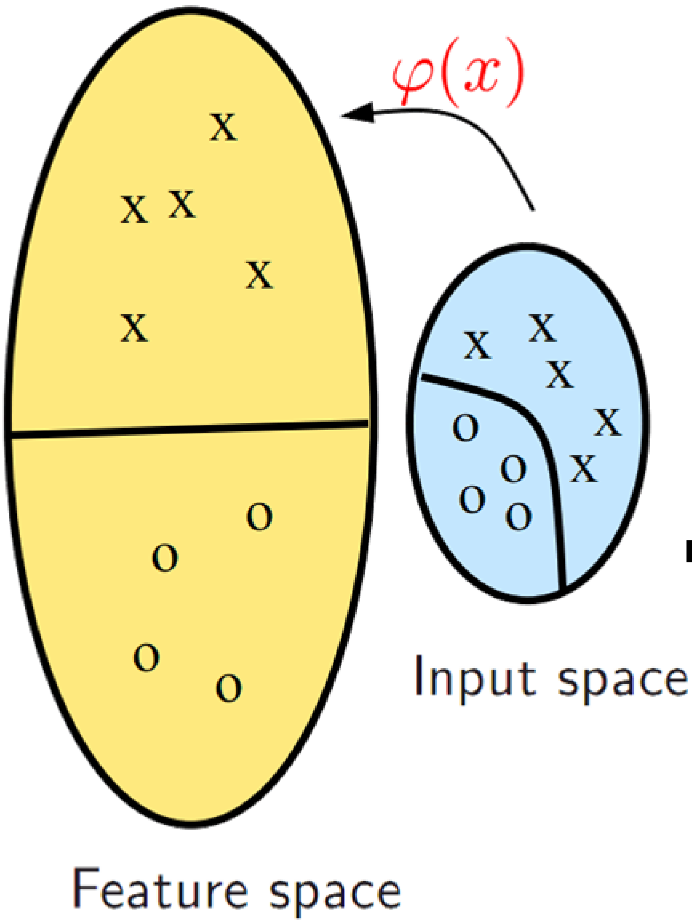
Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\, \alpha_i - \frac{1}{2} \Sigma\, \Sigma\, \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ is maximized and
(1) $\Sigma\, \alpha_i y_i = 0$
(2) $\alpha_i \geqslant 0$ for all α_i

- **The solution is:**

$$f(\mathbf{x}) = \Sigma\, \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b$$

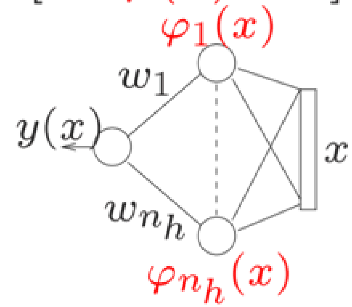
- **Optimization techniques for finding α_i's remain the same!**

Kernel Trick in SVM



- **Primal representation (inputs)**

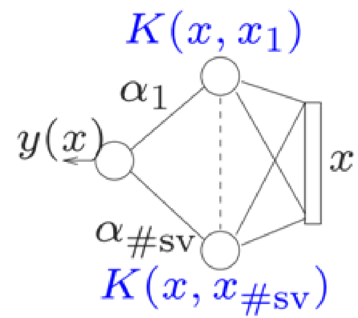
$$y(x) = \text{sign}[w^T \varphi(x) + b]$$



$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$$

- **Dual representation (features)**

$$y(x) = \text{sign}[\sum_{i=1}^{\#\text{sv}} \alpha_i y_i K(x, x_i)]$$



Primal & Dual Problems in SVM

- Primal problem (margin maximization)

$$\min_{w,b,\xi} \mathcal{J}(w,\xi) = \frac{1}{2}w^T w + c\sum_{i=1}^{N}\xi_i \ \ \text{s.t.} \ \ \begin{cases} y_i[w^T\varphi(x_i)+b] \geq 1-\xi_i \\ \xi_i \geq 0, \quad i=1,\ldots,N \end{cases}$$

- Dual problem (quadratic programming)

$$\max_{\alpha} \mathcal{Q}(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{N} y_i y_j K(x_i,x_j)\,\alpha_i\alpha_j + \sum_{j=1}^{N}\alpha_j \ \ \text{s.t.} \ \ \begin{cases} \sum_{i=1}^{N}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c, \ \forall i \end{cases}$$

- Primal representation / Dual representation

$$y(x) = \text{sign}[w^T\varphi(x)+b] \qquad y(x) = \text{sign}[\sum_i \alpha_i\, y_i\, K(x,x_i) + b]$$

Nonlinear SVM - Overview

- **SVM locates a separating hyper-plane in the feature space and classifies points in that space.**
- **It does not need to represent the space explicitly, simply by defining a kernel function.**
- **The kernel function plays the role of the dot product in the feature space.**

Properties of SVM

- **Sparseness of solution when dealing with large data sets**

  - only support vectors are used to specify the separating hyper-plane
- **Ability to handle large feature spaces**

  - complexity does not depend on the dimensionality of the feature space
- **Overfitting can be controlled by soft margin approach**
- **Nice math property**:

  - a simple convex optimization problem which is guaranteed to converge to a single global solution