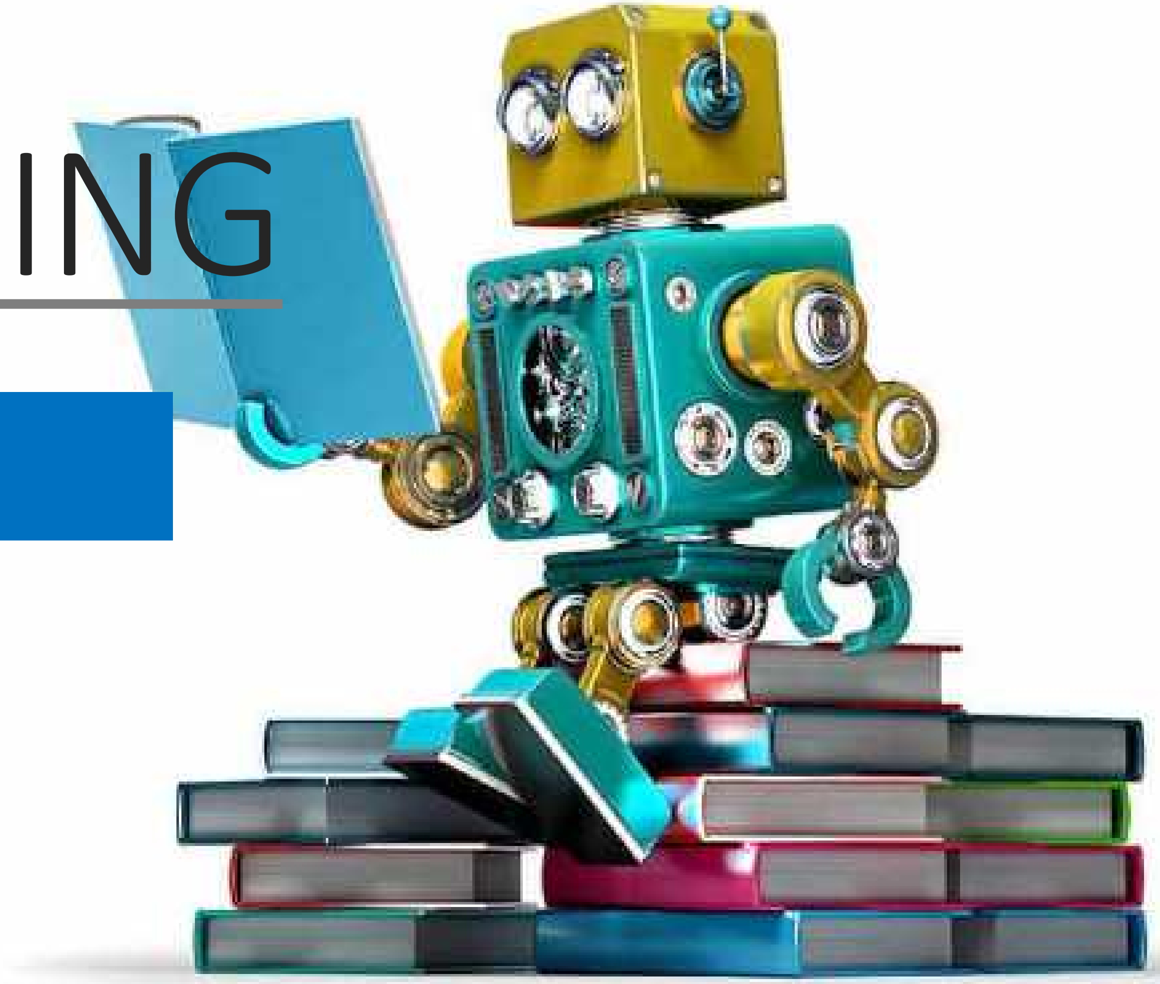
MACHINE LEARNING

LAB10 SVM



贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn

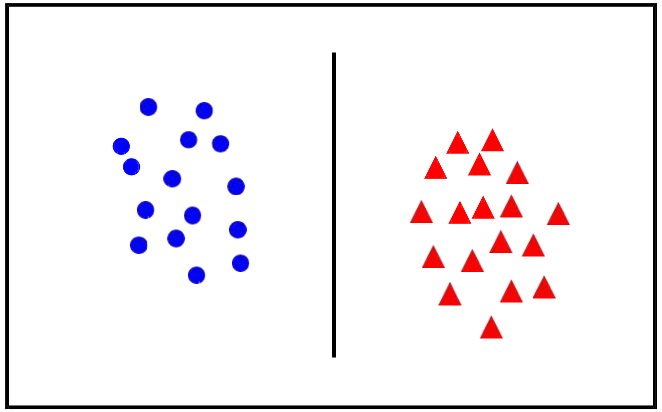
Outline

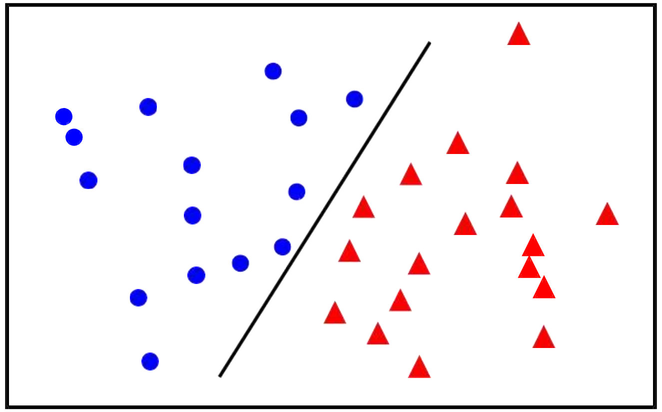
- Intro. to Linear separability and Perceptron
- Intro. to Support Vector Machine (svm) classifier
- LAB Assignment

Linear separability

A dataset is said to be linearly separable if it is possible to draw a line that can separate the red and green points from each other.

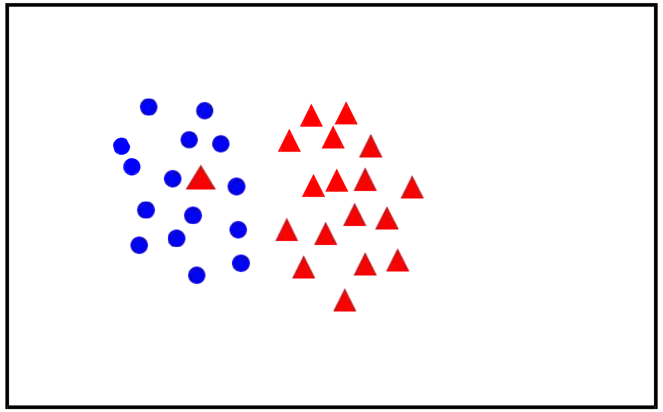
linearly separable

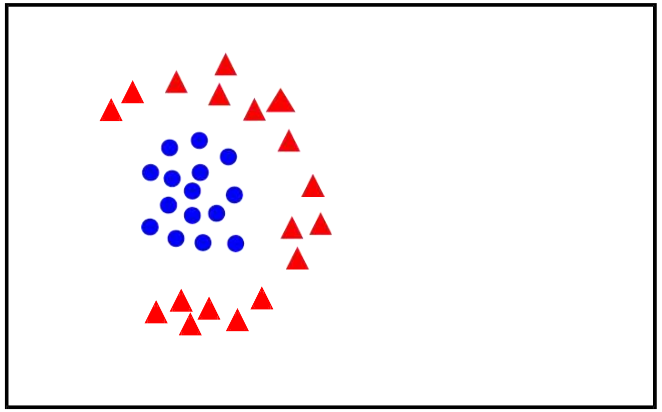




The items are completely linearly separable. All items fall to one side or the other.

not linearly separable





What if you can't find it?

Binary Classification

Given training data $(\mathbf{x}_i, y_i)$ for $i = 1 \ldots N$, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, learn a classifier $f(\mathbf{x})$ such that

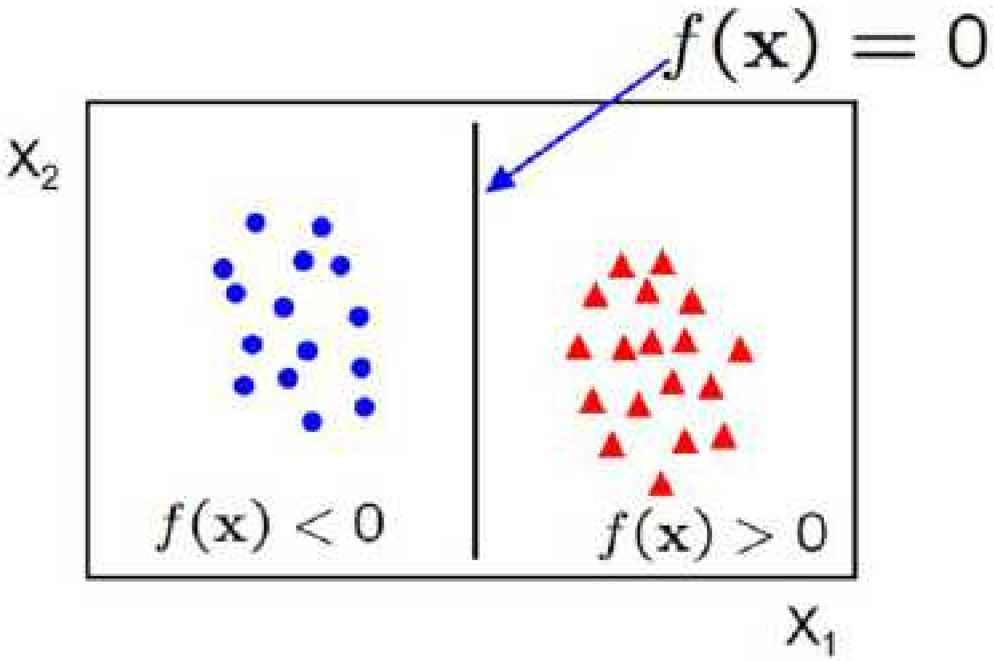
$$f(\mathbf{x}_i) \begin{cases} \geq 0 & y_i = +1 \\ < 0 & y_i = -1 \end{cases}$$

i.e. $y_i f(\mathbf{x}_i) > 0$ for a correct classification.

Linear classifiers

A linear classifier has the form

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

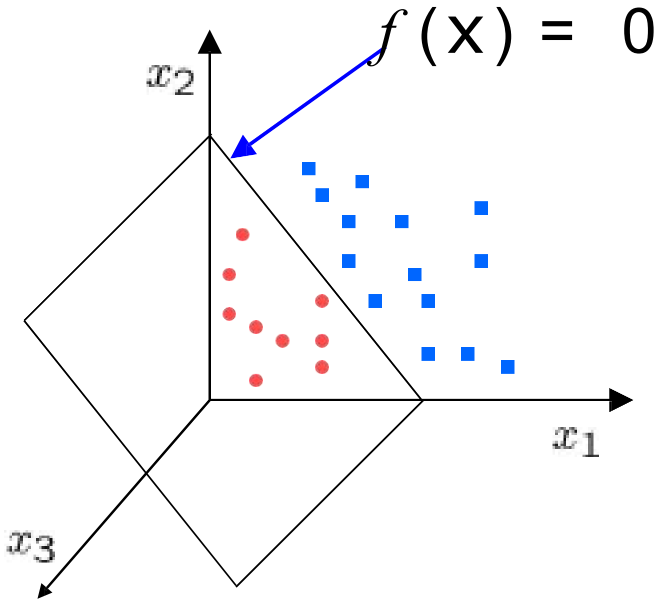


- in 2D the discriminant is a line
- $\mathbf{w}$ is the normal to the line, and b the bias
- $\mathbf{w}$ is known as the weight vector

Linear classifiers

A linear classifier has the form

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$



- in 3D the discriminant is a plane, and in nD it is a hyperplane
- For a linear classifier, the training data is used to learn **w** and then discarded
- Only **w** is needed for classifying new data

The Perceptron Classifier

How can we find this separating hyperplane?

- The Perceptron Algorithm

Write classifier as $f(\mathbf{x}_i) = \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_i + w_0 = \mathbf{w}^\top \mathbf{x}_i$

where $\mathbf{w} = (\tilde{\mathbf{w}}, w_0), \mathbf{x}_i = (\tilde{\mathbf{x}}_i, 1)$

- Loss Function

$$L(\vec{w}) = \sum_{\vec{x_i} \epsilon \in M} y_i \vec{w}^T \vec{x_i}$$

- Take the derivative:

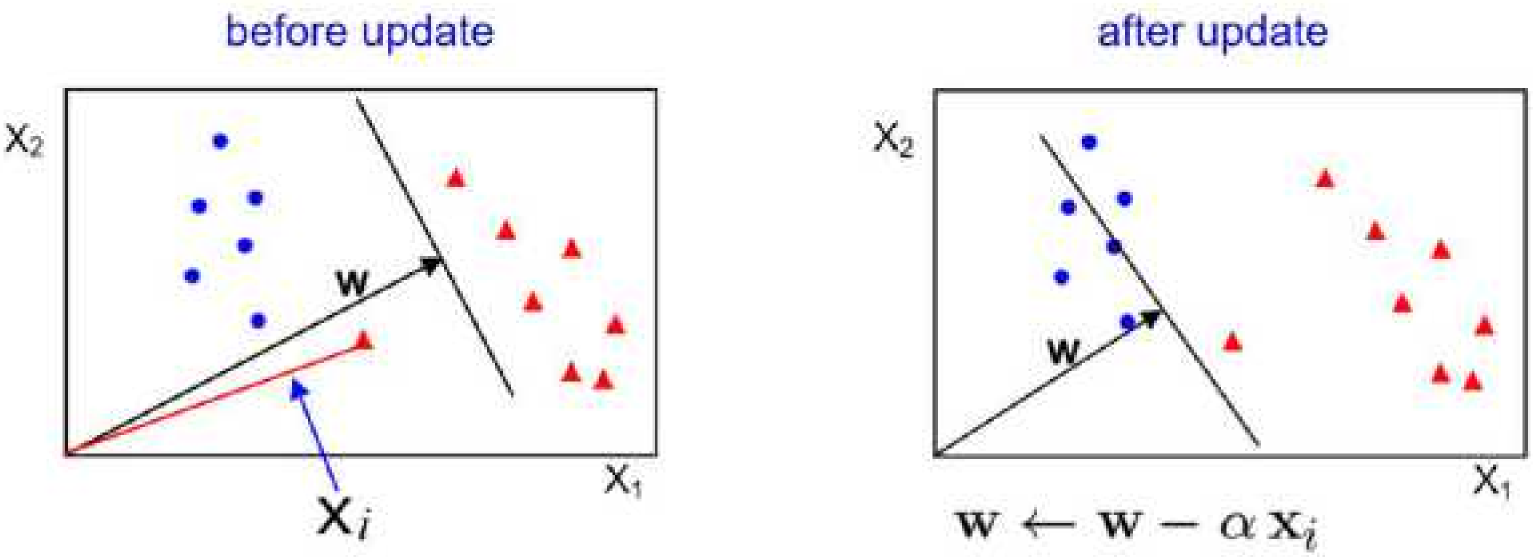
$$\Delta_{\vec{w}} L(\vec{w}) = \sum_{\vec{x_i} \epsilon \in M} y_i \vec{x_i}$$

- Gradient descent:

$$\vec{w} \leftarrow \vec{w_0} + \alpha y_i \vec{x_i}$$

For example in 2D

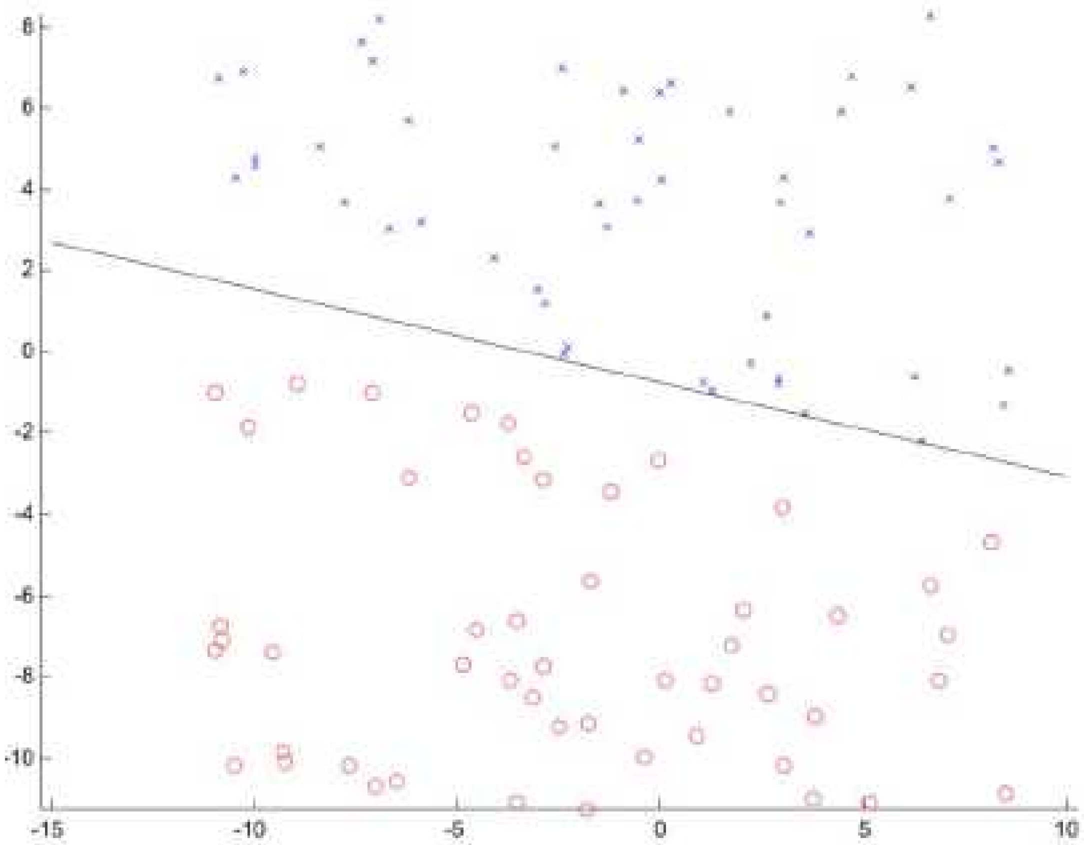
- Initialize **w** = 0
- Cycle though the data points { $\mathbf{x}_i$, y_i }
  - if $\mathbf{x}_i$ is misclassified then $\mathbf{w} \leftarrow \mathbf{w} + \alpha\, \mathrm{sign}(f(\mathbf{x}_i))\, \mathbf{x}_i$
- Until all the data is correctly classified



NB after convergence $\mathbf{w} = \sum_i^N \alpha_i \mathbf{x}_i$

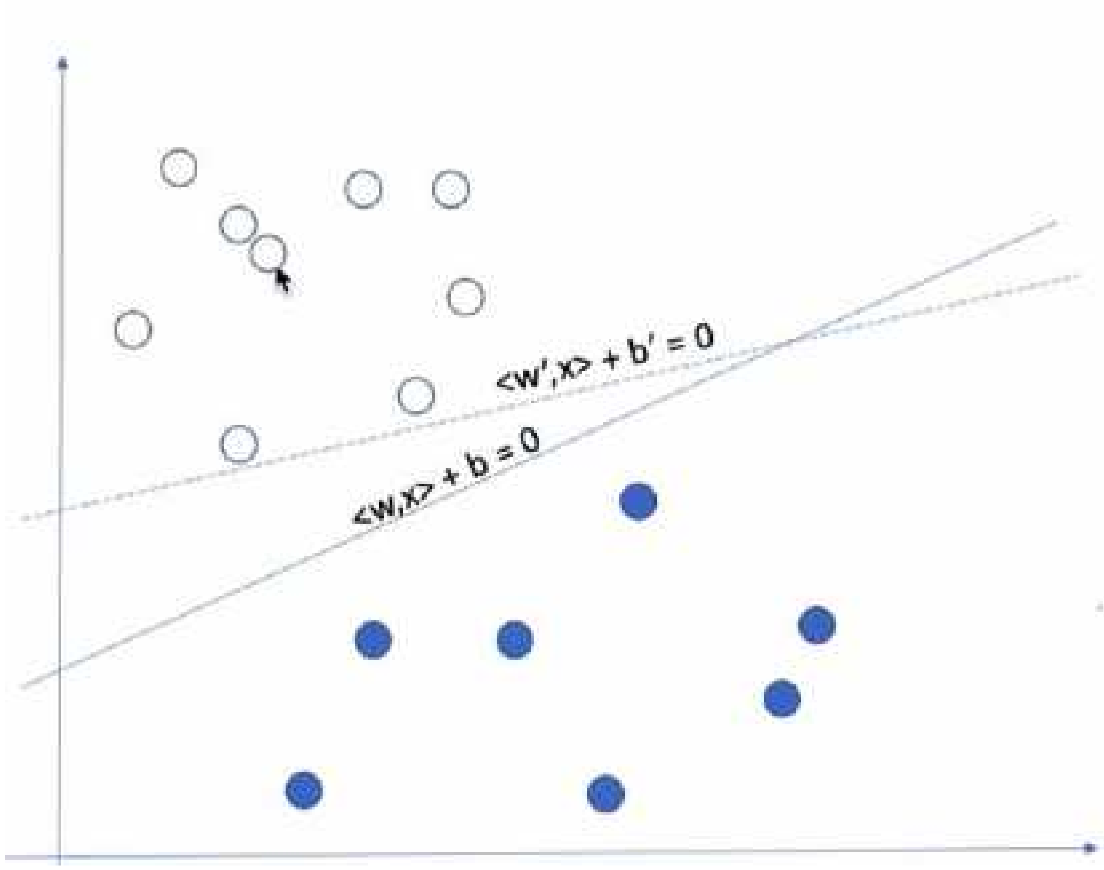
For example in 2D

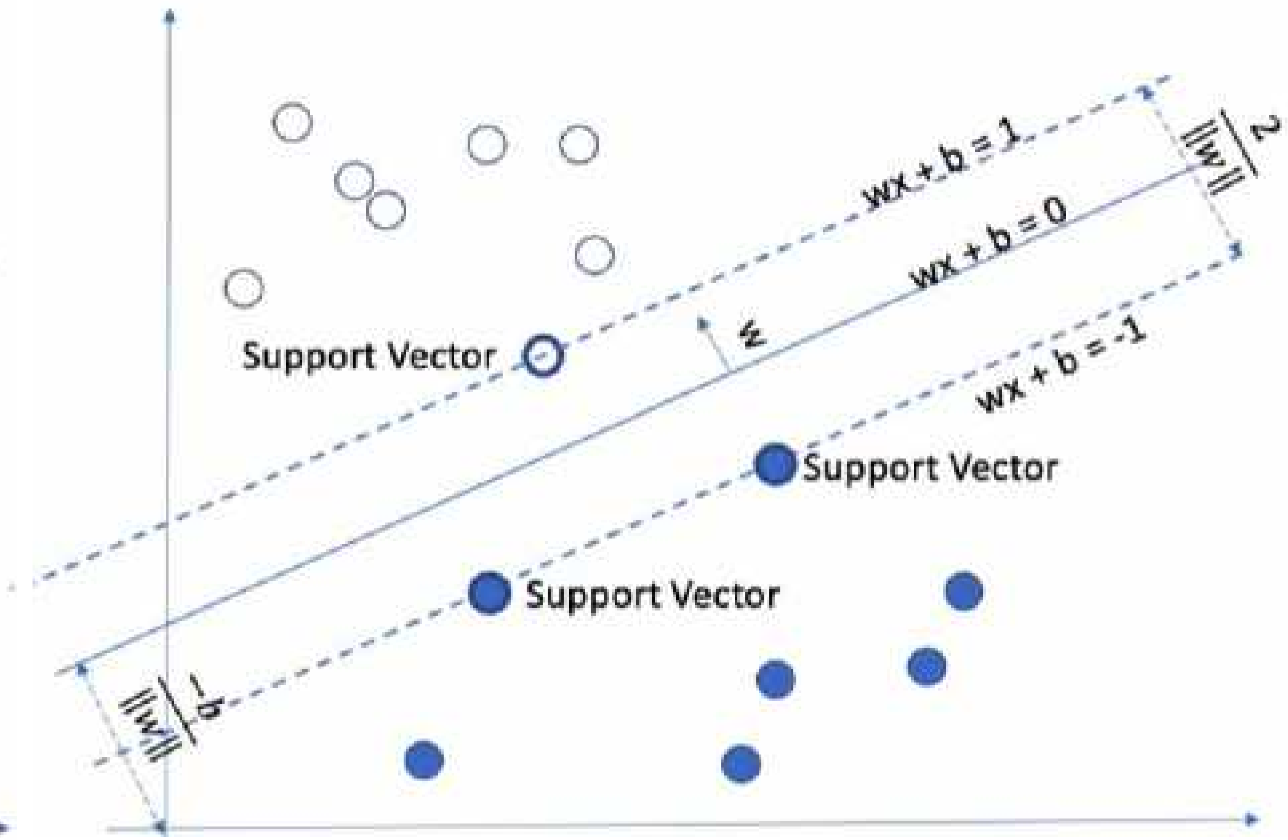
Perceptron example



- if the data is linearly separable, then the algorithm will converge
- convergence can be slow …
- separating line close to training data
- we would prefer a larger margin for generalization

What is the best w?

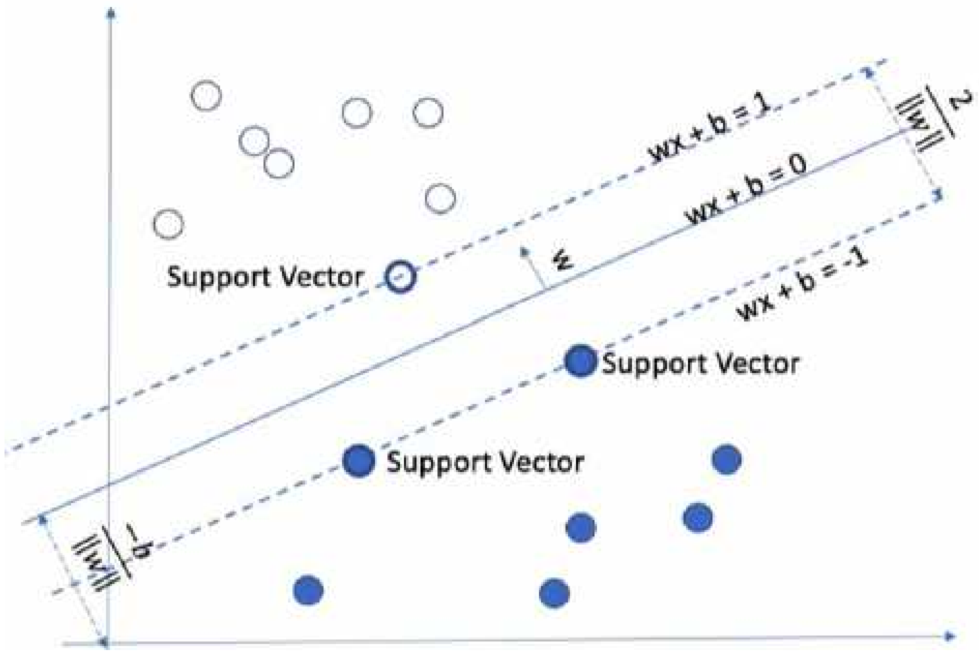




- Linear classifier

svm

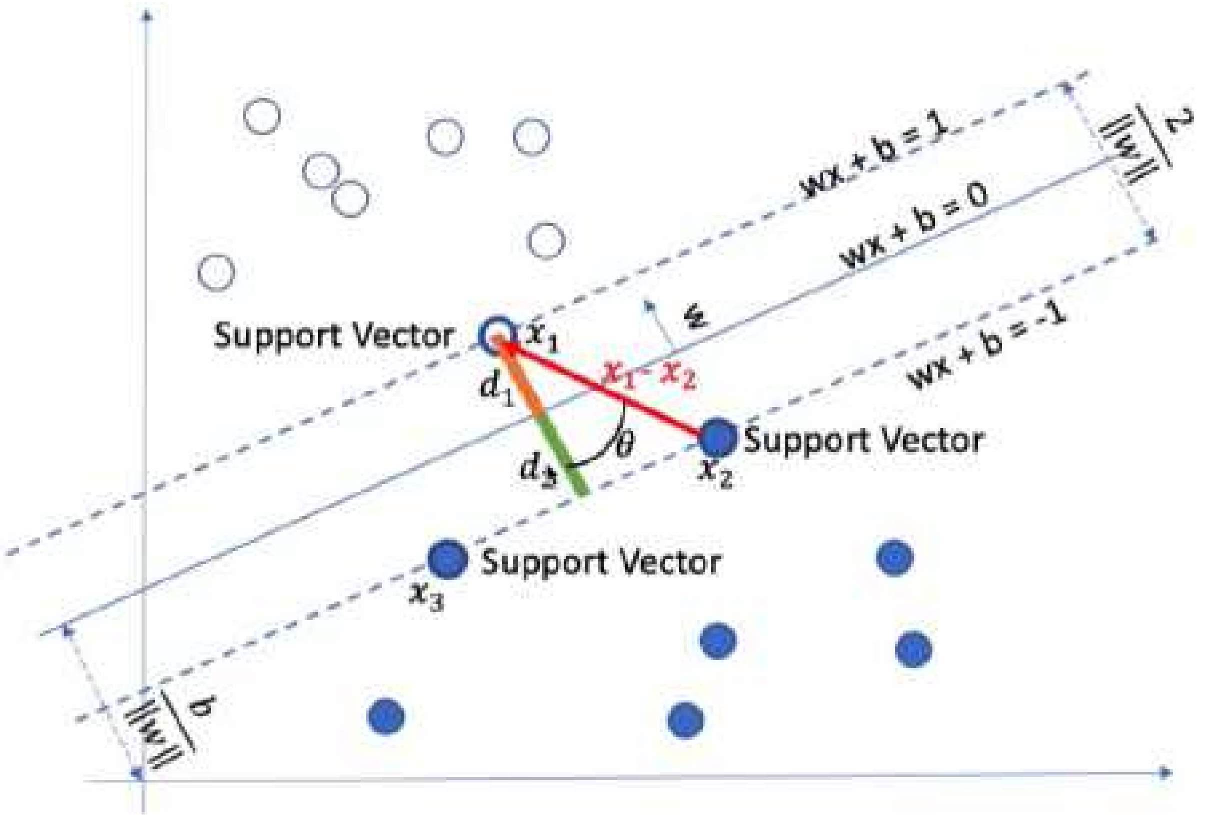
SVM – sketch derivation



- Since $\mathbf{w}^\top \mathbf{x} + b = 0$ and $c(\mathbf{w}^\top \mathbf{x} + b) = 0$ define the same plane, we have the freedom to choose the normalization of $\mathbf{w}$

- Choose normalization such that $\mathbf{w}^\top \mathbf{x}_+ + b = +1$ and $\mathbf{w}^\top \mathbf{x}_- + b = -1$ for the positive and negative support vectors respectively

SVM – sketch derivation



$$w^T x_1 + b = 1$$

$$w^T x_2 + b = -1$$

$$(w^T x_1 + b) - (w^T x_2 + b) = 2$$

$$w^T (x_1 - x_2) = 2$$

$$w^T (x_1 - x_2) = \|w\|_2 \|x_1 - x_2\|_2 cos\theta = 2$$

$$\|x_1 - x_2\|_2 cos\theta = \frac{2}{\|w\|_2}$$

$$d_1 = d_2 = \frac{\|x_1 - x_2\|_2 cos\theta}{2} = \frac{\frac{2}{\|w\|_2}}{2} = \frac{1}{\|w\|_2}$$

$$d_1 + d_2 = \frac{2}{\|w\|_2}$$

SVM are also called max-Margin Classifer

SVM – Optimization

- Learning the SVM can be formulated as an optimization:

$$\max_{w,b} \quad \frac{1}{||w||}$$
$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, ..., n$$

- Or equivalently

$$\min_{w,b} \quad \frac{1}{2}||w||^2$$
$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, ..., n$$

- This is a quadratic optimization problem subject to linear constraints and there is a unique minimum

The Optimization Problem Solution

Lagrange function:

$$L(w,b,\alpha)=\frac{1}{2}||w||^2-\sum_{i=1}^{n}\alpha_i(y_i(w^T x_i+b)-1)$$

α_i* is *Lagrange multiplier* , *$\alpha_i \geq 0$

The original problem is equivalent to

$$\min_{w,b}\max_{\alpha} L(w,b,\alpha)$$

According to Lagrangian duality

$$\min_{w,b}\max_{\alpha} L(w,b,\alpha)=\max_{\alpha}\min_{w,b} L(w,b,\alpha)$$

So you can solve the original problem by solving a simpler dual problem.

$$\begin{cases}\nabla_w L(w,b,\alpha)=w-\sum_{i=1}^{n}\alpha_i y_i x_i=0\Rightarrow \boxed{w=\sum_{i=1}^{n}\alpha_i y_i x_i}\\ \nabla_b L(w,b,\alpha)=-\sum_{i=1}^{n}\alpha_i y_i=0\Rightarrow \boxed{\sum_{i=1}^{n}\alpha_i y_i=0}\end{cases}$$

$$\min_{w,b} L(w,b,\alpha)=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{n}\alpha_i$$

The Optimization Problem Solution

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{n}\alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n}\alpha_i y_i = 0, \quad \alpha_i \geq 0, i = 1, \ldots, n$$

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

Find $\alpha_1 \ldots \alpha_N$ such that

$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\mathbf{x_i}^T\mathbf{x_j}}$ is maximized and

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

(2) $\alpha_i \geq 0$ for all α_i

The Optimization Problem Solution

- The solution has the form:

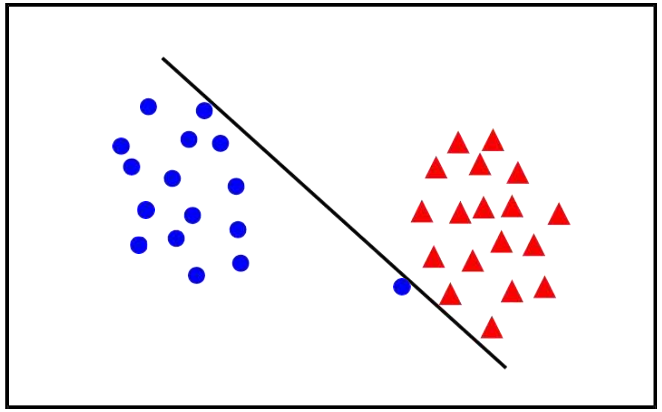
$$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w}^{\mathbf{T}}\mathbf{x_k} \text{ for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0$$

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

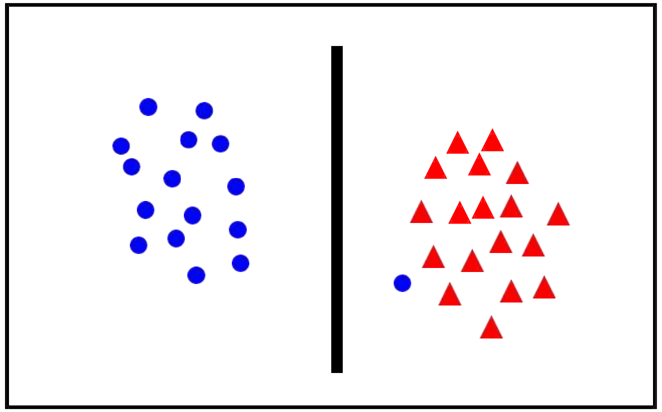
$$f(\mathbf{x}) = \Sigma\alpha_i y_i \mathbf{x_i}^{\mathbf{T}}\mathbf{x} + b$$

- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$ between all pairs of training points.

Linear separability again: What is the best w?



•the points can be linearly separated but there is a very narrow margin



•but possibly the large margin solution is better, even though one constraint is violated

In general there is a trade off between the margin and the number of mistakes on the training data

Introduce "slack" variables

- **By introducing a slack variable, the constraint becomes**

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

- **The learning problem becomes**

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, ..., n \\ & \xi_i \geq 0, \quad i = 1, 2, ..., n \end{aligned}$$

ξ_j - "slack" variables
= (>1 if x_j misclassifed)

pay linear penalty if mistake

C - tradeoff parameter (C = ∞ recovers hard margin SVM)

"Soft" margin solution

The optimization problem becomes

$$\min_{\mathbf{w},b,\xi\geq 0} \quad \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

$$\xi_i \geq 0$$

- Every constraint can be satisfied if ξ_i is sufficiently large
- C is a regularization parameter:
  - small C allows constraints to be easily ignored $\rightarrow$ large margin
  - large C makes constraints hard to ignore $\rightarrow$ narrow margin
  - $C = \infty$ enforces all constraints: hard margin
- This is still a quadratic optimization problem and there is a unique minimum. Note, there is only one parameter, C.

“Soft” margin solution

The optimization problem becomes

$$\min_{\mathbf{w},b,\xi\geq 0} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + C\sum_i \xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$

$$\xi_i \geq 0$$

The dual problem for soft margin classification:

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{x_i^T x_j}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

“Soft” margin solution

Solution to the dual problem is:

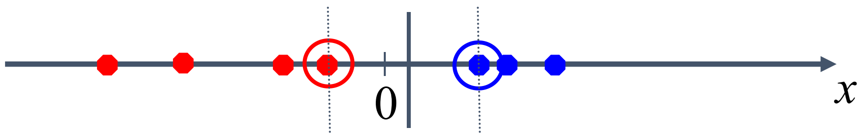
$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x_i}$$
$$b = y_k(1 - \xi_k) - \mathbf{w}^{\mathbf{T}}\mathbf{x}_k \text{ where } k = \underset{k'}{\operatorname{argmax}}\ \alpha_{k'}$$

w is not needed explicitly for classification!

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

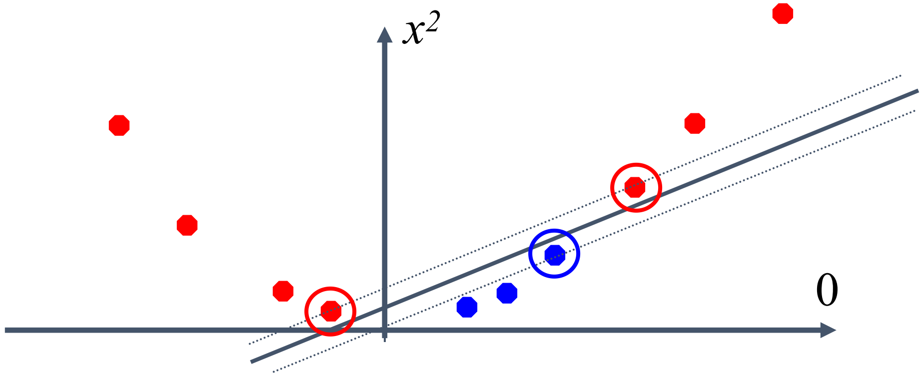
Non-linear SVMs

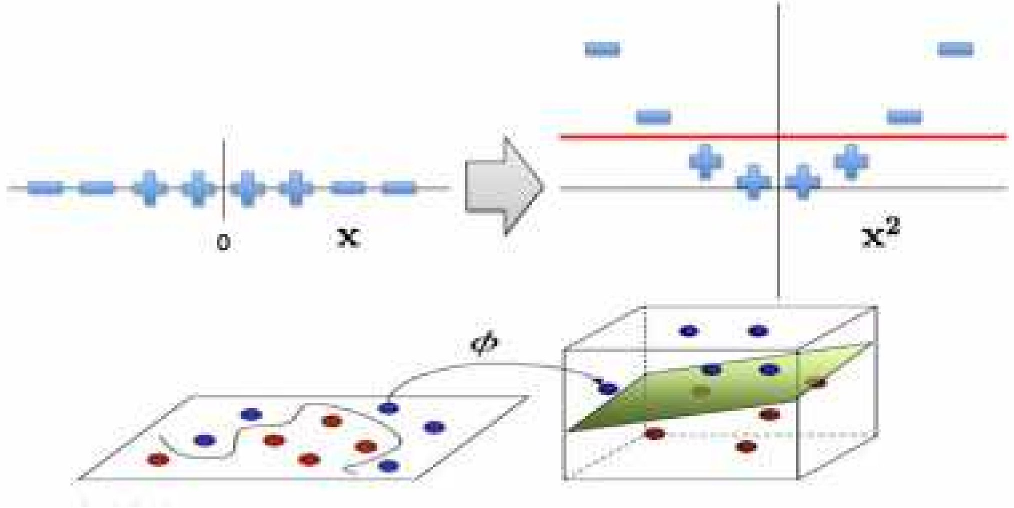
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

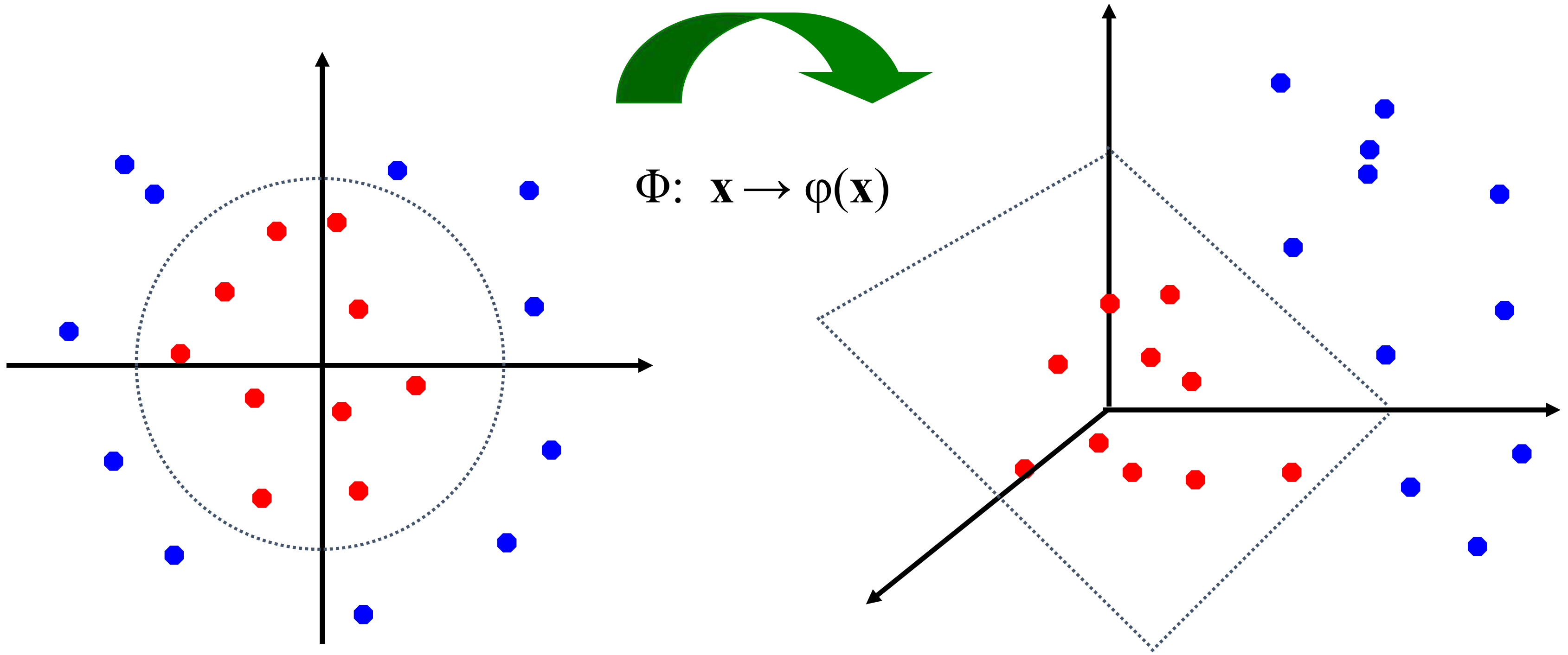
- How about… mapping data to a higher-dimensional space:





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:



The “Kernel Trick”

两个数据点属于同一类别使值增加，否则减小

衡量两个数据之间的相似性

$$\max_{\alpha\geq 0,\lambda\geq 0} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j \boxed{y_i y_j} \boxed{\mathbf{x}_i^\top \mathbf{x}_j}$$

$$\text{s.t.} \quad \boxed{\sum_i \alpha_i y_i = 0}, \quad C - \alpha_i - \lambda_i = 0$$

不同数据点的权重不同，不同的类别的权重一致

两个数据点属于同一类别使值增加，否则减小

衡量两个数据之间的相似性

$$\max_{\alpha\geq 0} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j \boxed{y_i y_j} \boxed{k(\mathbf{x}_i, \mathbf{x}_j)}$$

$$\text{s.t.} \quad \boxed{\sum_i \alpha_i y_i = 0}, \quad \alpha_i \leq C, \quad i = 1, \ldots, n$$

What Functions are Kernels?

- For some functions $K(x_i,x_j)$ checking that

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

Non-linear SVMs Mathematically

- **Dual problem formulation:**

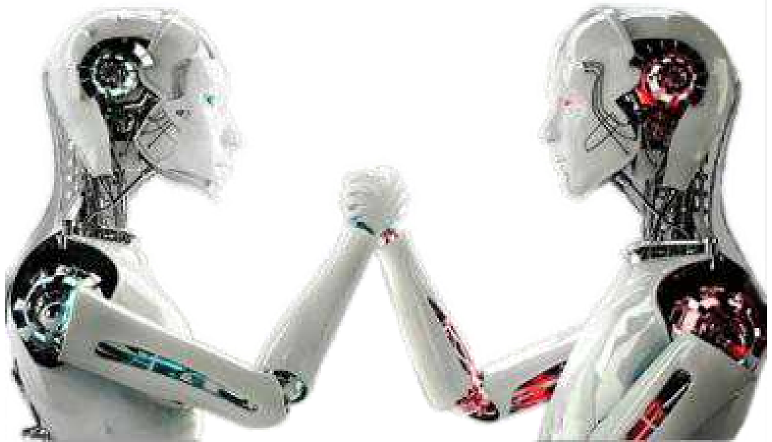
Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(\mathrm{x}_i, \mathrm{x}_j)$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- **The solution is:**

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

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

Lab Assignment



Lab Task

Complete the exercises and questions in the Lab10.svm.md

Thanks

贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn