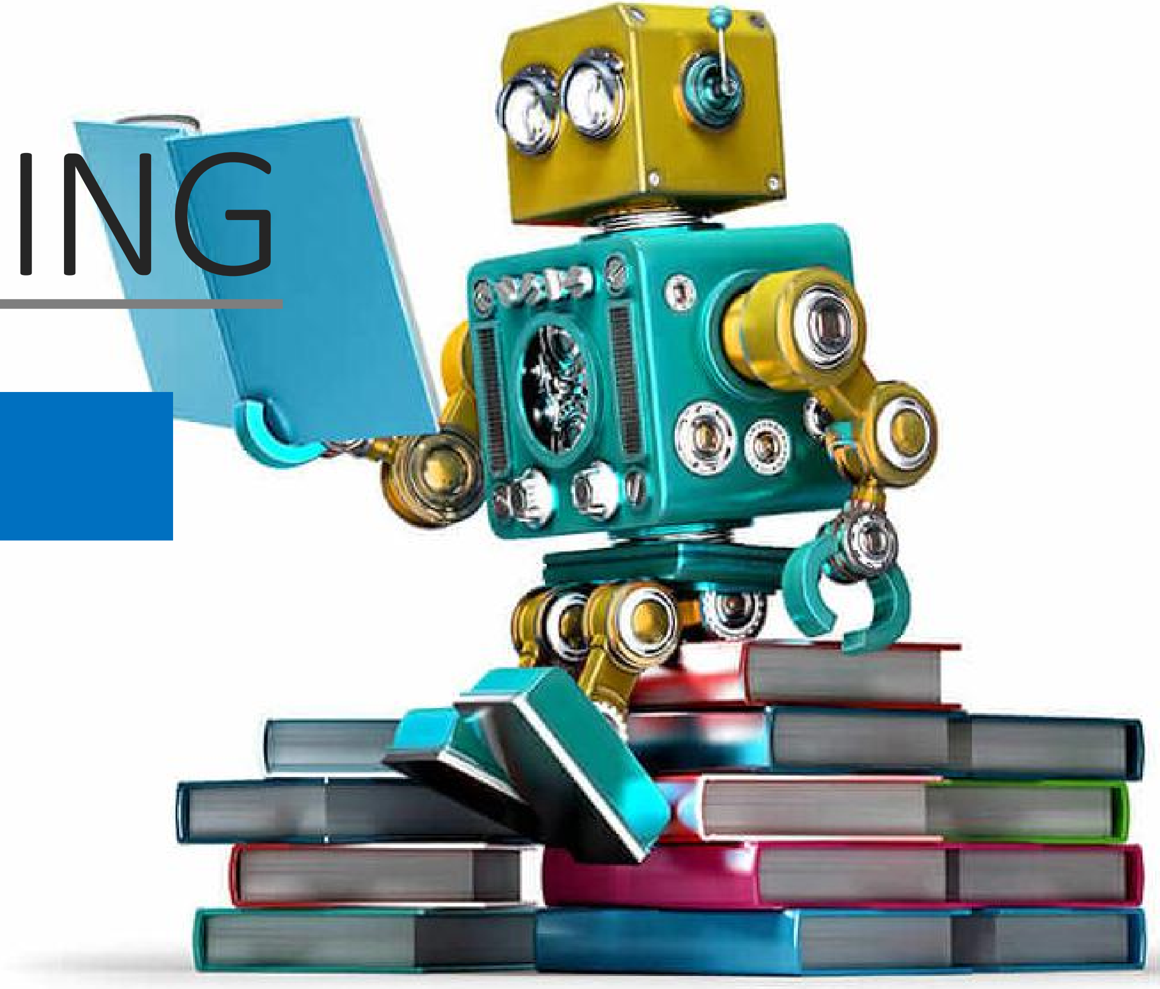
MACHINE LEARNING

LAB7 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
- Application: Pedestrian detection in Computer Vision

Binary Classification

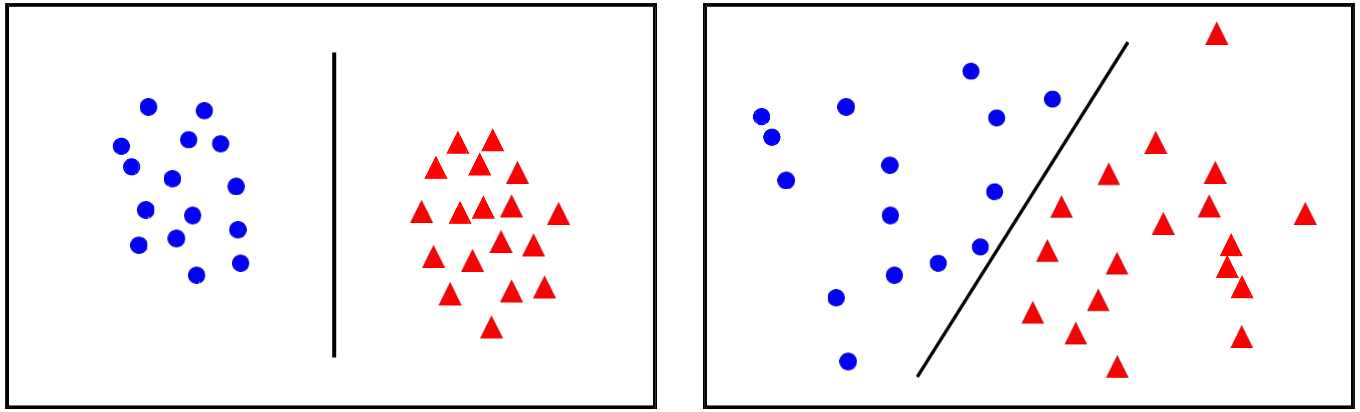
Given training data $(\mathbf{x}_i, y_i)$ for $i = 1 \ldots N$, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, learn a classifier $f(\mathbf{x})$ such that

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

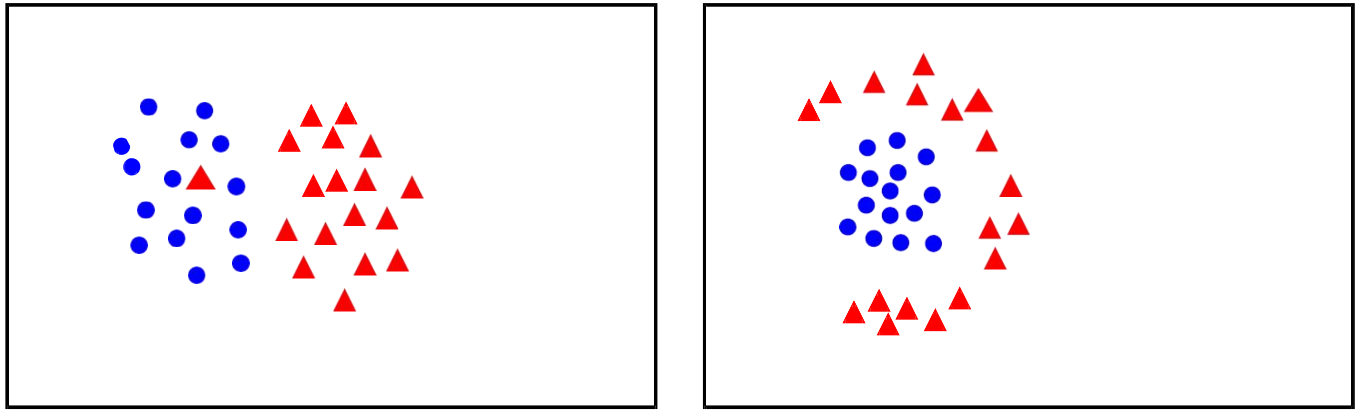
i.e. $y_i f(\mathbf{x}_i) > 0$ for a correct classification.

Linear separability

linearly separable



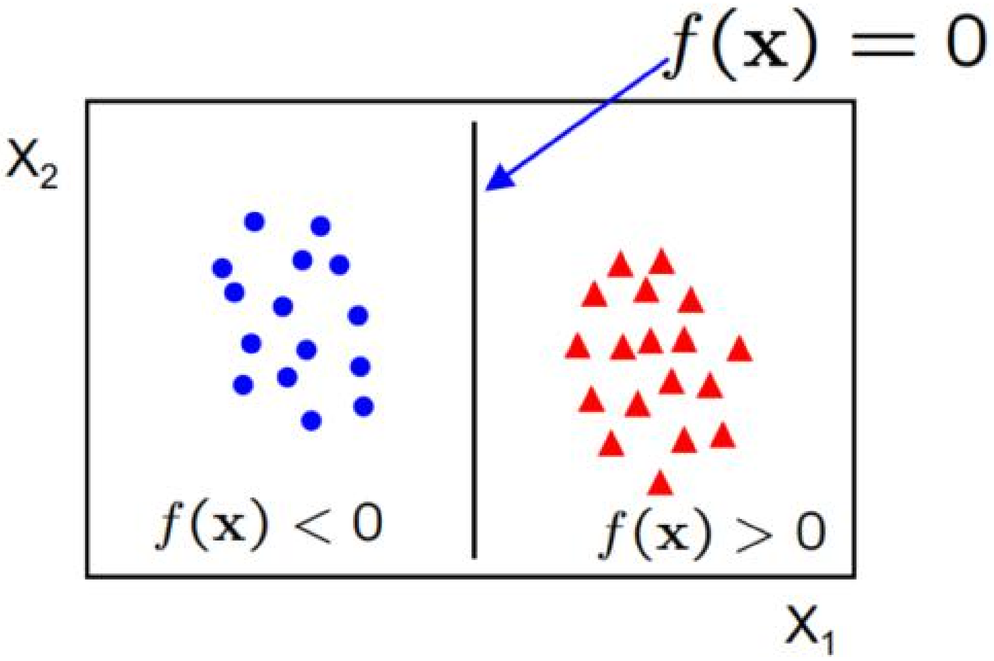
not linearly separable



Linear classifiers

A linear classifier has the form

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

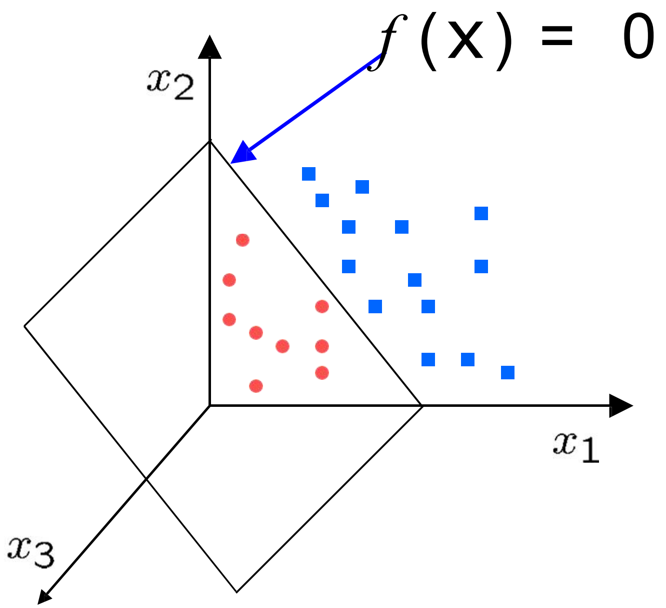


- in 2D the discriminant is a line
- $\mathbf{w}$ is the normal to the line, and b the bias
- $\mathbf{w}$ is known as the weight vector

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

Given linearly separable data $\mathbf{x}_i$ labelled into two categories $y_i = \{-1,1\}$, find a weight vector **w** such that the discriminant function

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

separates the categories for i = 1, .., N

- how can we find this separating hyperplane ? The Perceptron Algorithm

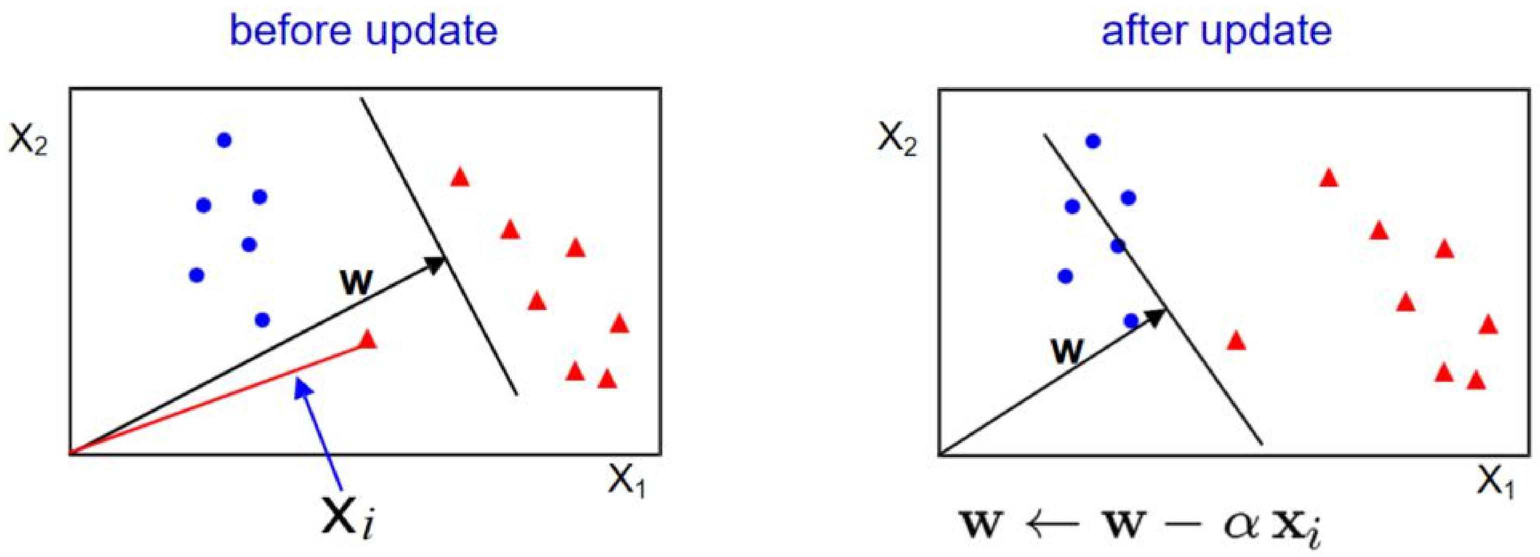
Write classifier as $f(\mathbf{x}_i) = \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_i + w_0 = \mathbf{w}^\top \mathbf{x}_i$

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

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

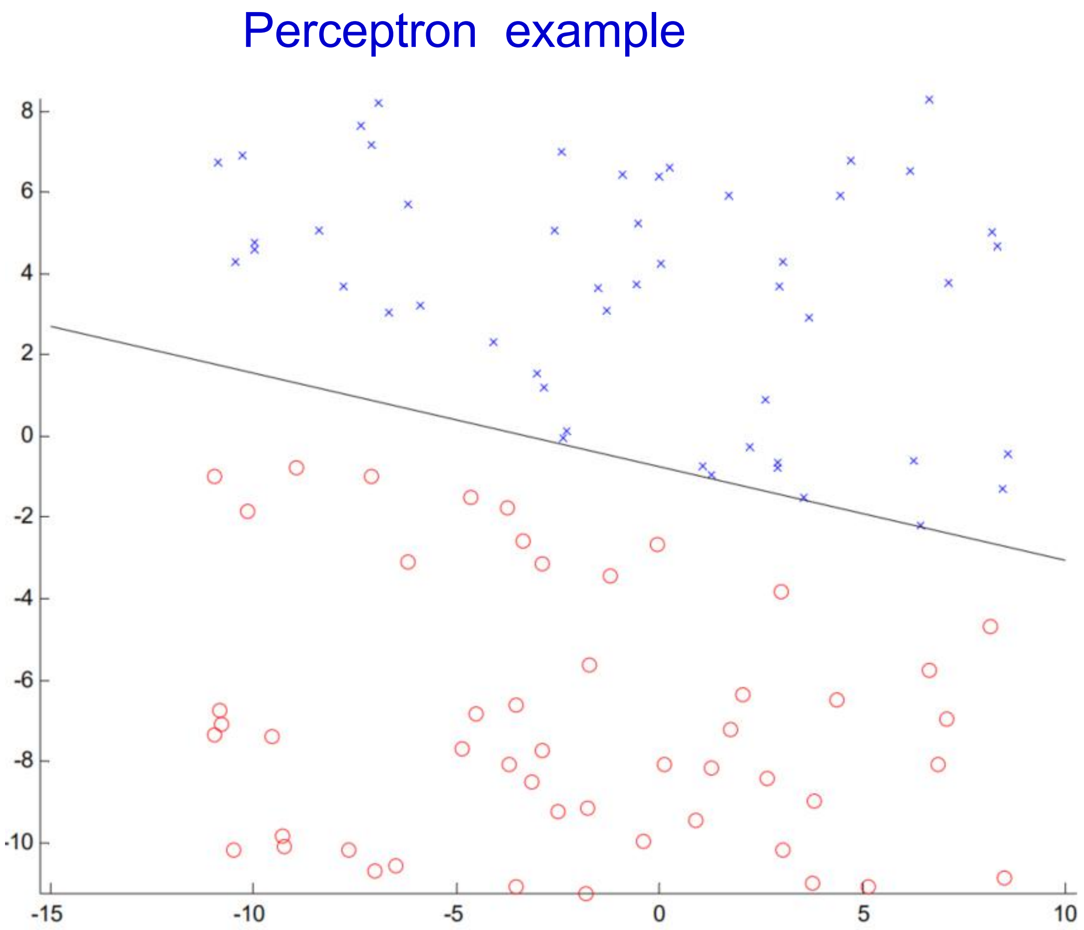
For example in 2D

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



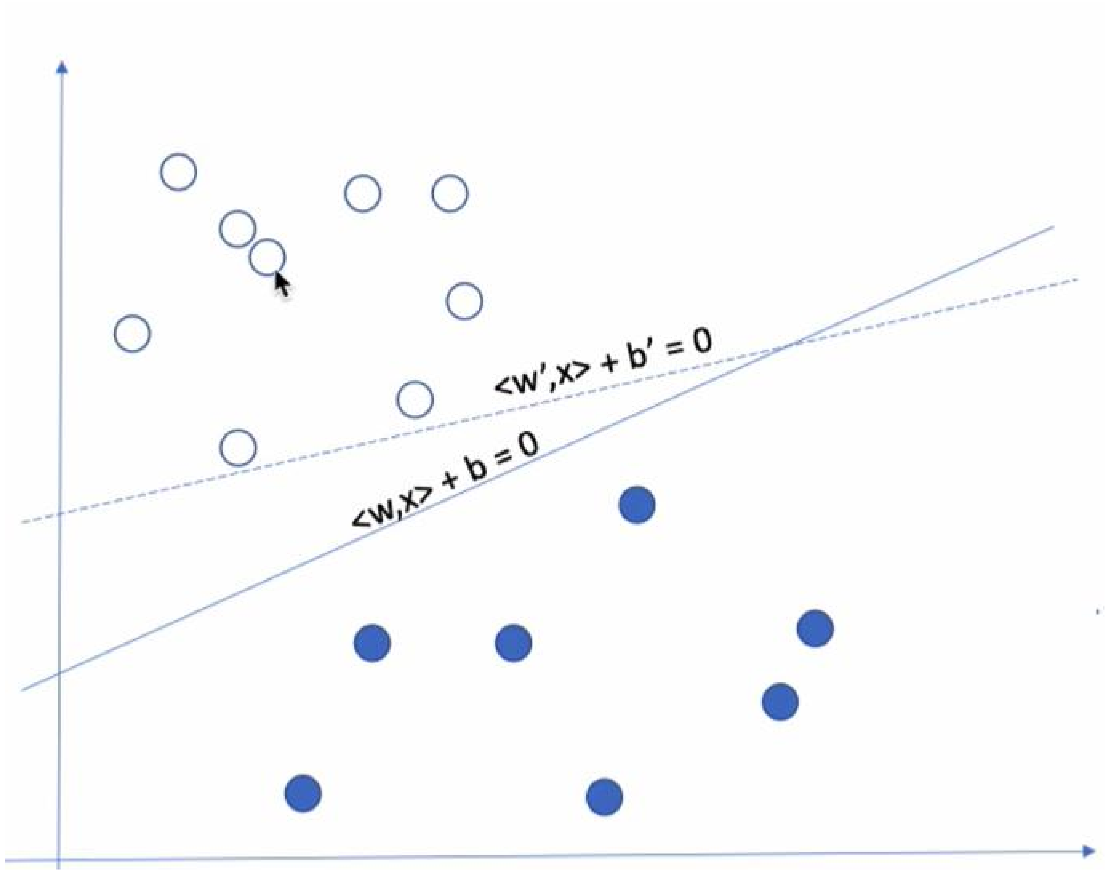
NB after convergence $\mathbf{w} = \sum_i^N \alpha_i \mathbf{x}_i$

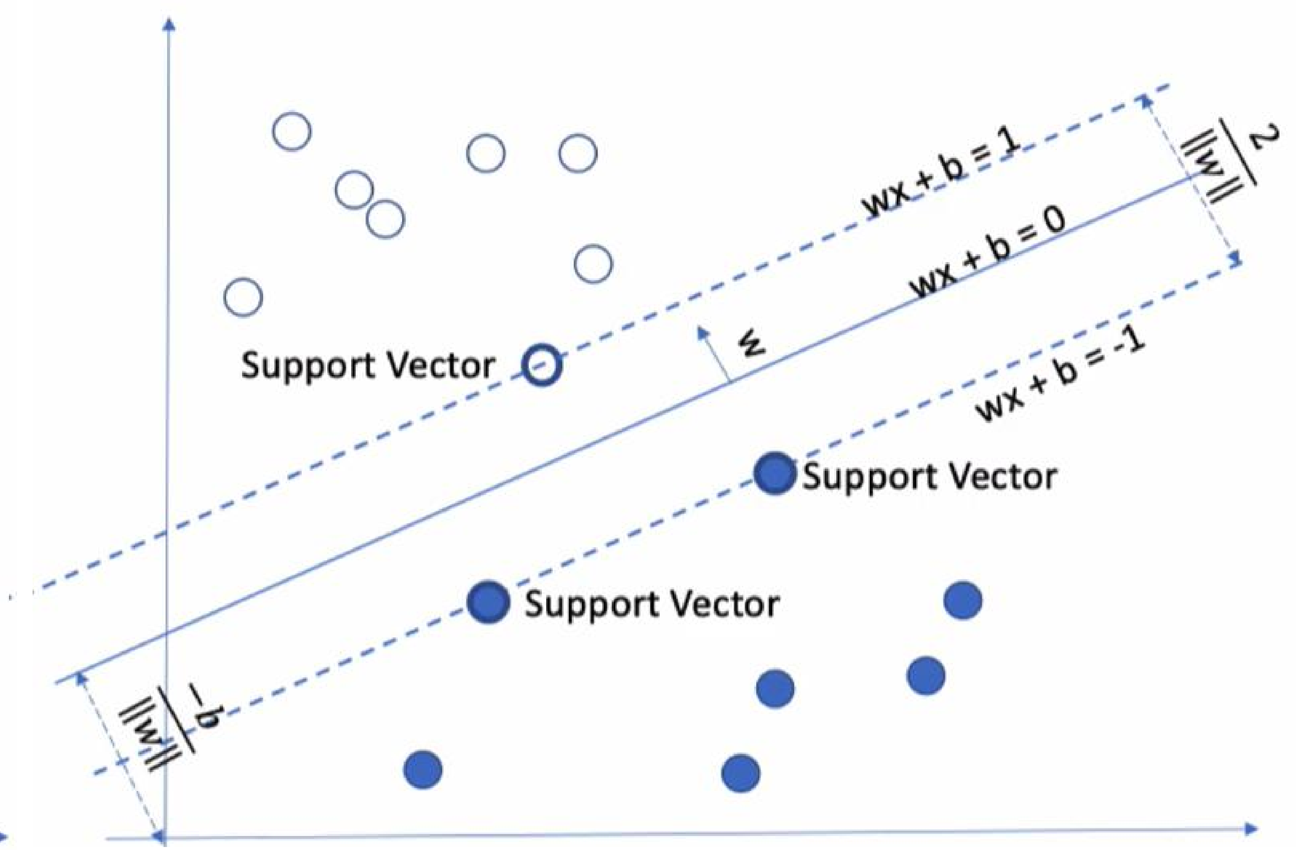
For example in 2D



- 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?

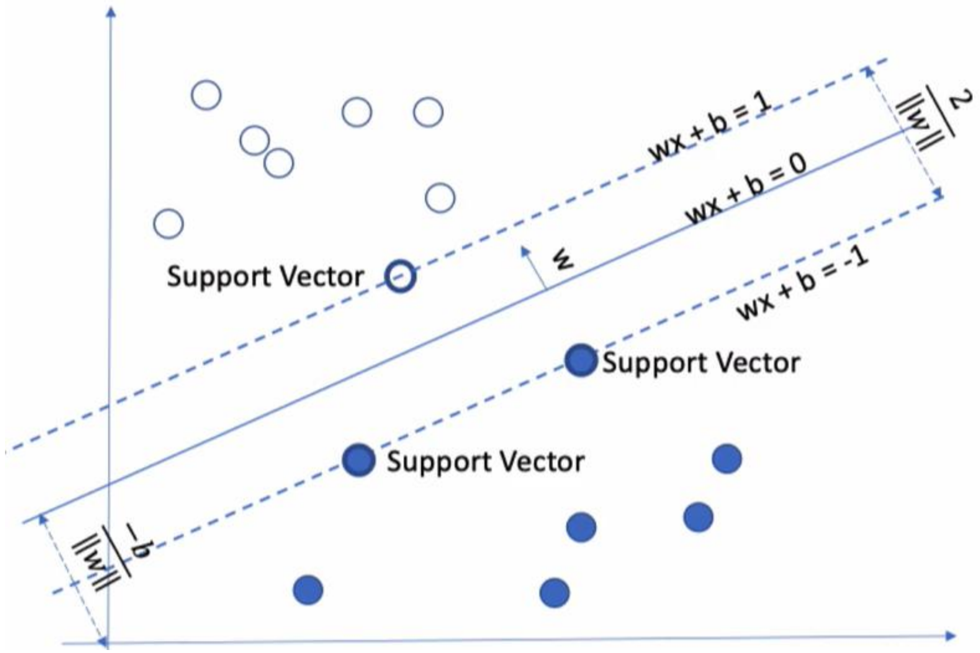




- Linear classifier

svm

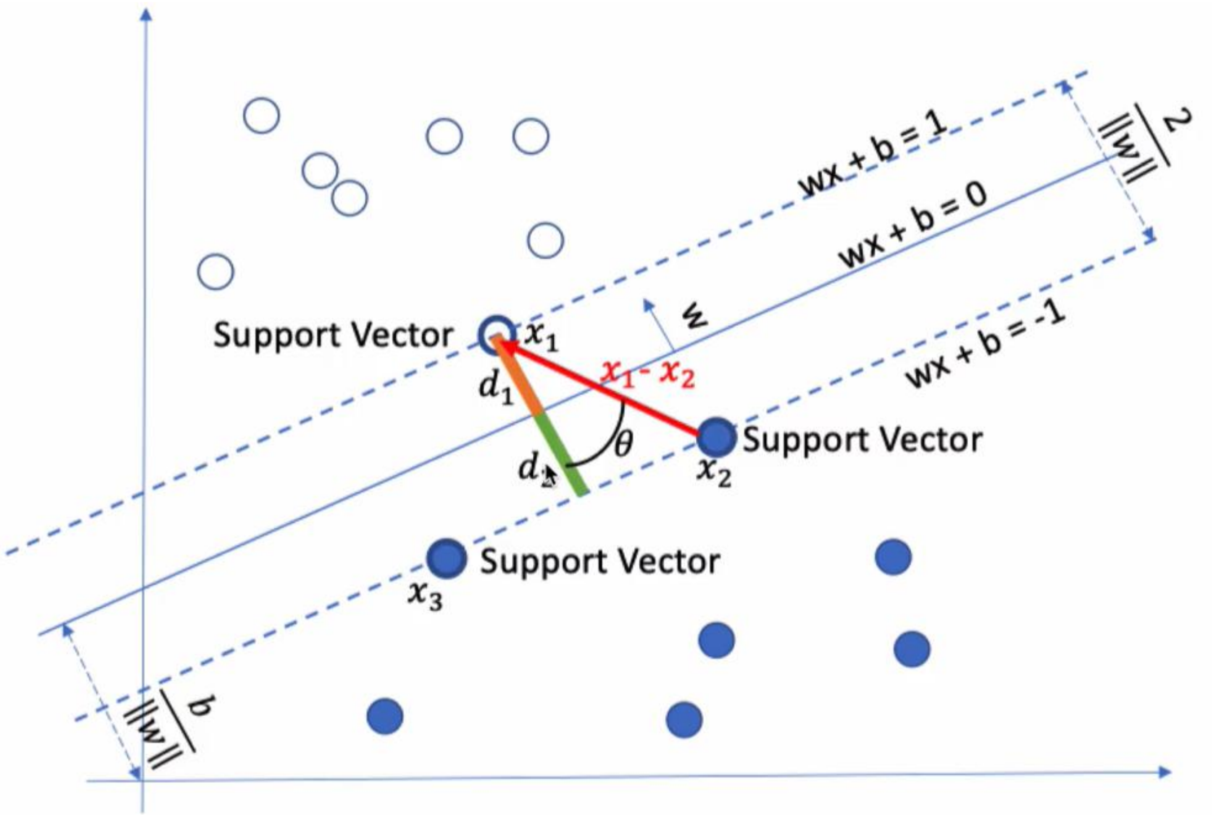
SVM – sketch derivation



- 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



SVM are also called max-Margin Classifer

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

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

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

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

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

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

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

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

SVM – Optimization

- Learning the SVM can be formulated as an optimization:

$$\max_{w,b} \frac{2}{\|w\|_2}$$

$$s.t.\, y^{(i)}(w^T * x^{(i)} + b) \geq 1, i = 1, 2, \ldots, n$$

- Or equivalently

$$\min_{w,b} \quad \frac{1}{2}||w||^2$$

$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, n$$

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

The Optimization Problem Solution

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

- **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:

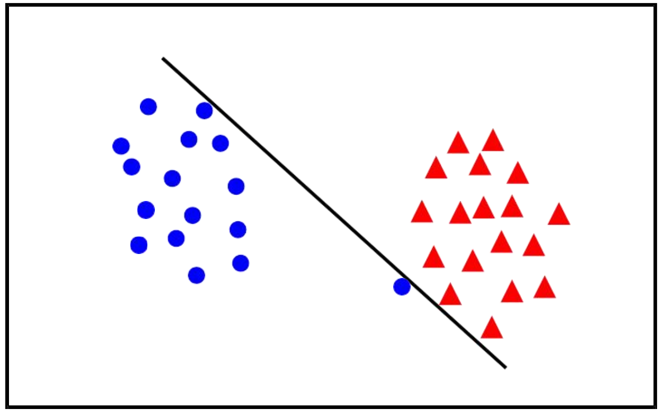
$$\boxed{\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w}^{\mathbf{T}}\mathbf{x_k} \text{ for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0}$$

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

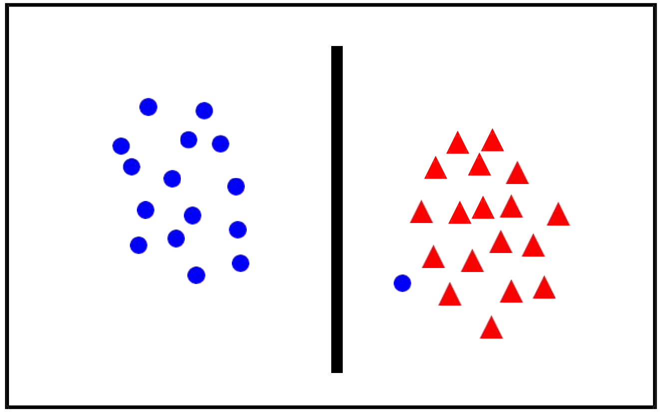
$$\boxed{f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x_i}^{\mathbf{T}}\mathbf{x} + b}$$

- 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



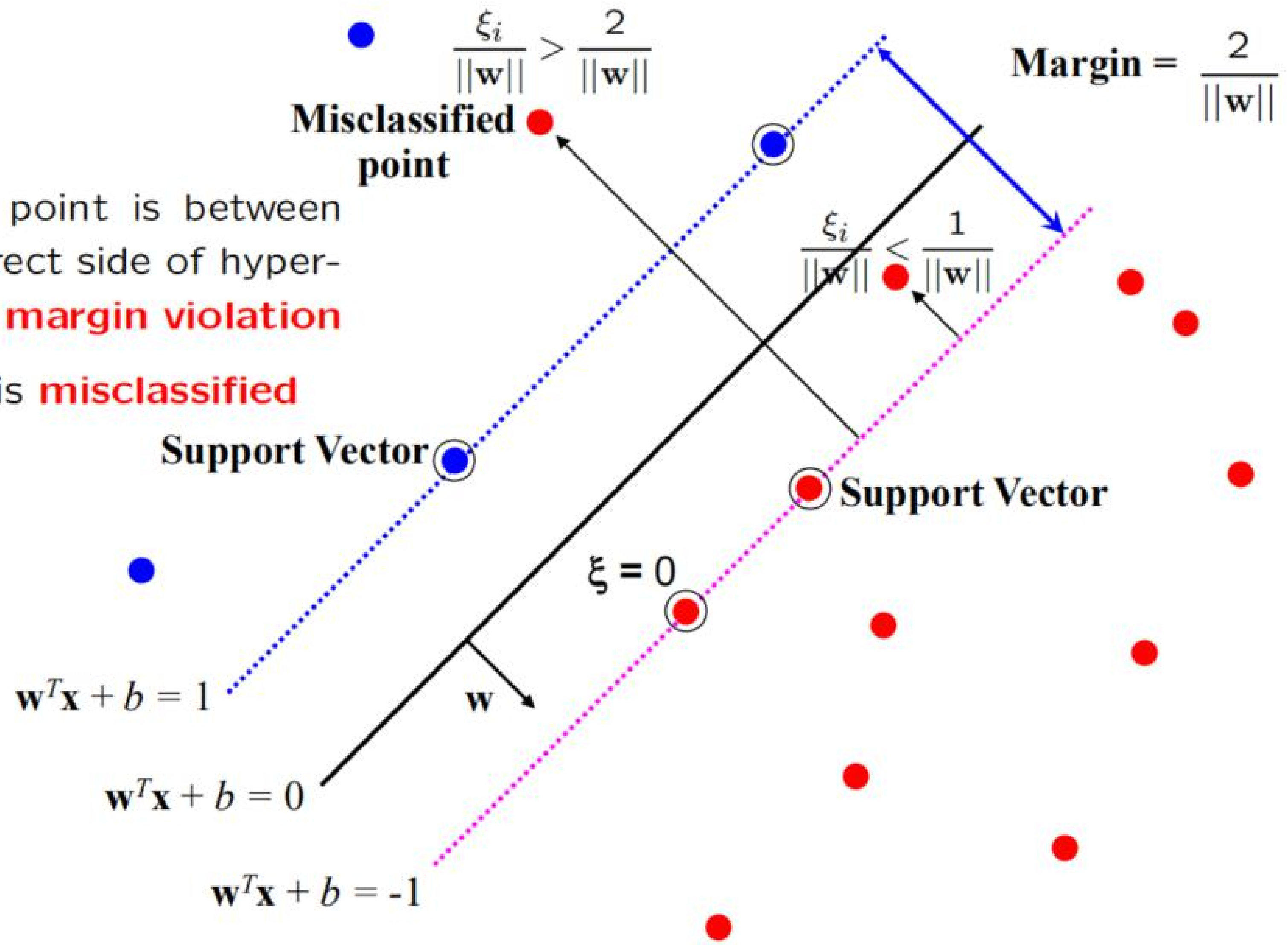
•but possibly the large margin solution is better, even though one constraint is violated

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

Introduce “slack” variables

$$\xi_i \geq 0$$

- for $0 < \xi \leq 1$ point is between margin and correct side of hyper-plane. This is a **margin violation**
- for $\xi > 1$ point is **misclassified**



“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

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

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

$$\xi_i \geq 0$$

The dual problem for soft margin classification:

Find $\alpha_1 \ldots \alpha_N$ such that
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\mathrm{x}_i^T\mathrm{x}_j}$ 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:

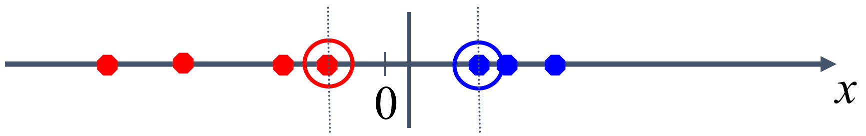
$$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$$
$$b = y_k(1 - \xi_k) - \mathbf{w}^{\mathrm{T}}\mathbf{x}_k \text{ where } k = \underset{k'}{\operatorname{argmax}}\ \alpha_{k'}$$

w is not needed explicitly for classification!

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

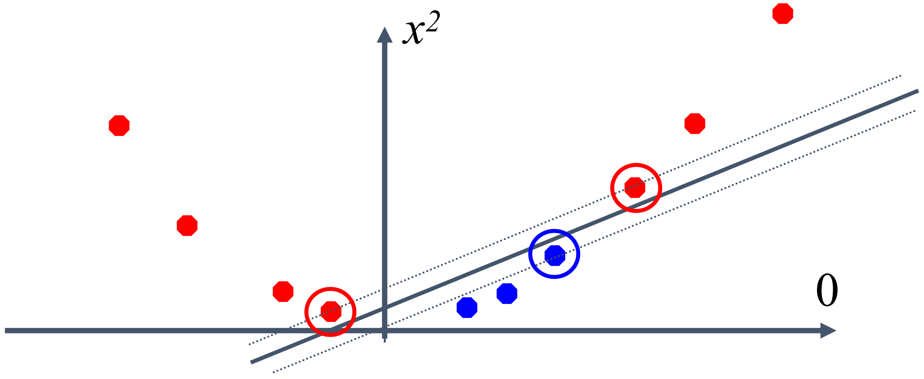
Non-linear SVMs

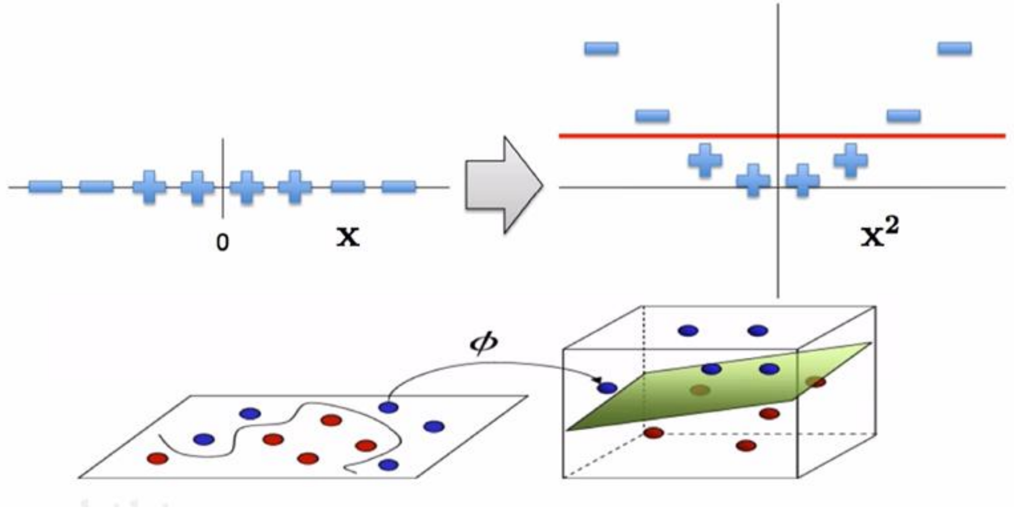
- Datasets that are linearly separable with some noise work out great:



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

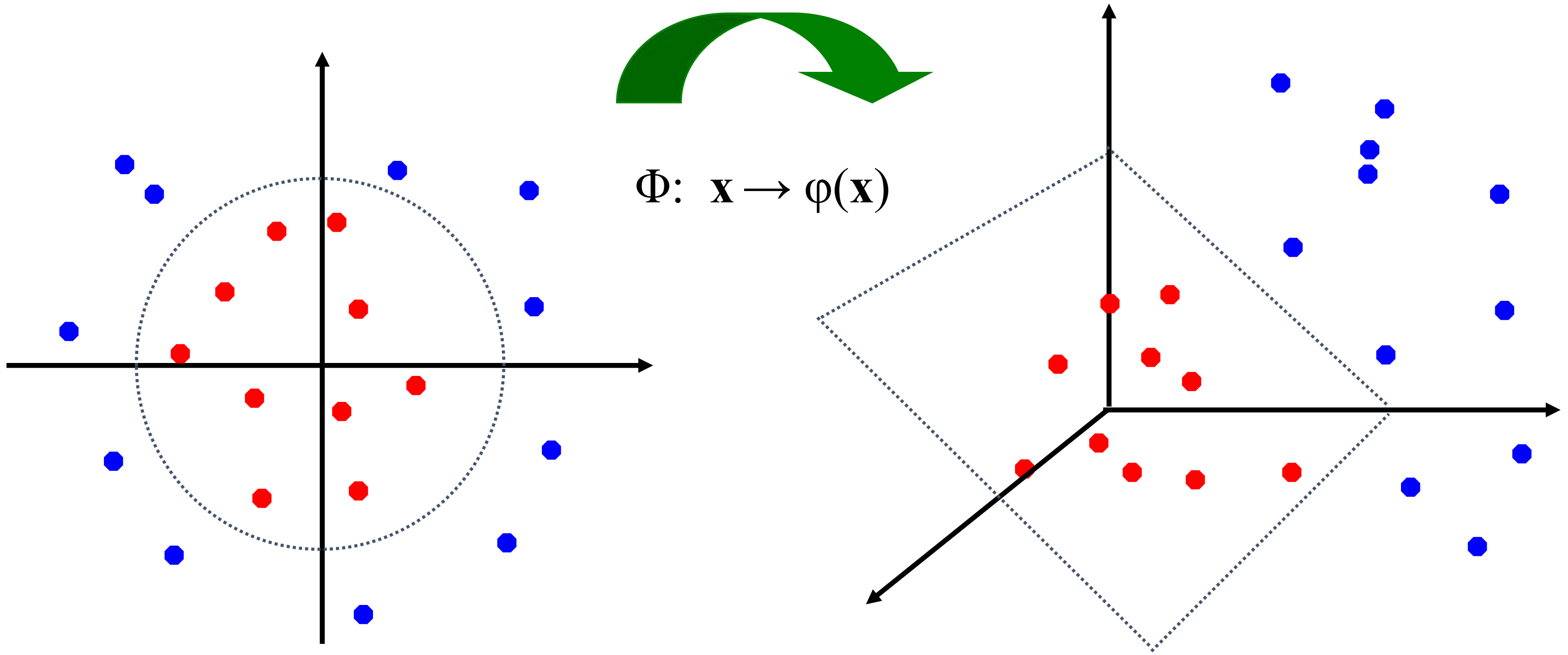
- How about… mapping data to a higher-dimensional space:





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
$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(x_i, x_j)$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

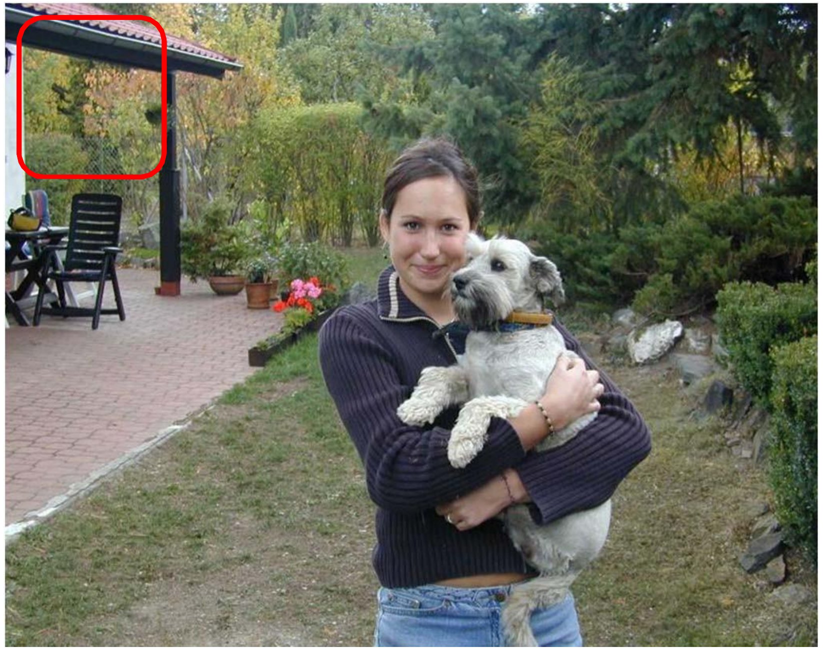
- **The solution is:**

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

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

Application: Pedestrian detection in Computer Vision

Objective: detect (localize) standing humans in an image



•reduces object detection to binary classification

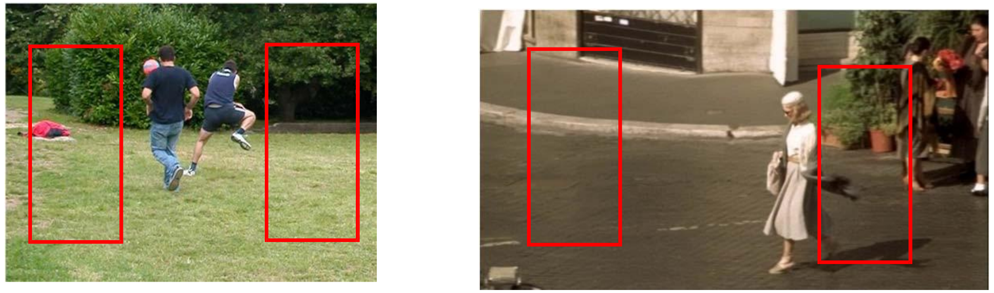
•does an image window contain a person or not?

Method: the HOG detector

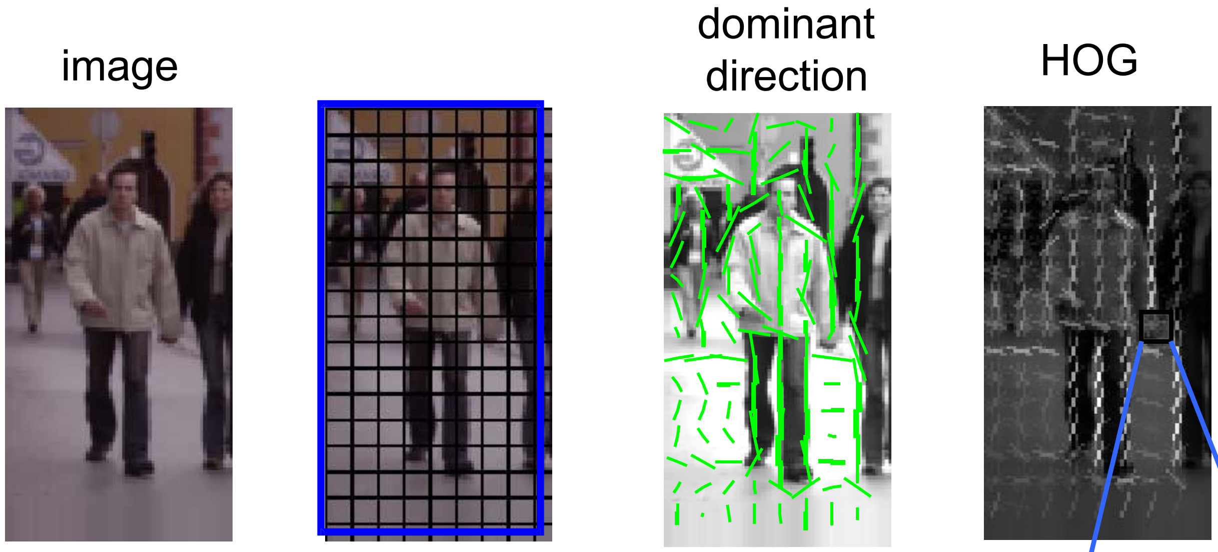
Training data and features

- Positive data – 1208 positive window examples

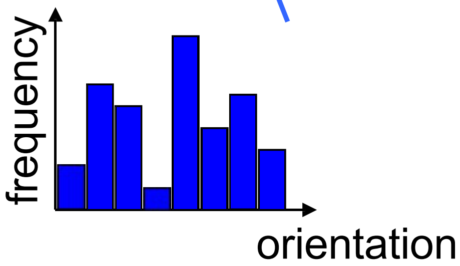
- Negative data – 1218 negative window examples (initially)



Feature: histogram of oriented gradients (HOG)



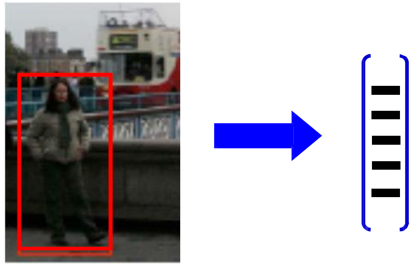
- tile window into 8 x 8 pixel cells
- each cell represented by HOG



Feature vector dimension = 16 x 8 (for tiling) x 8 (orientations) = 1024

Algorithm

- Training (Learning)
  - Represent each example window by a HOG feature vector



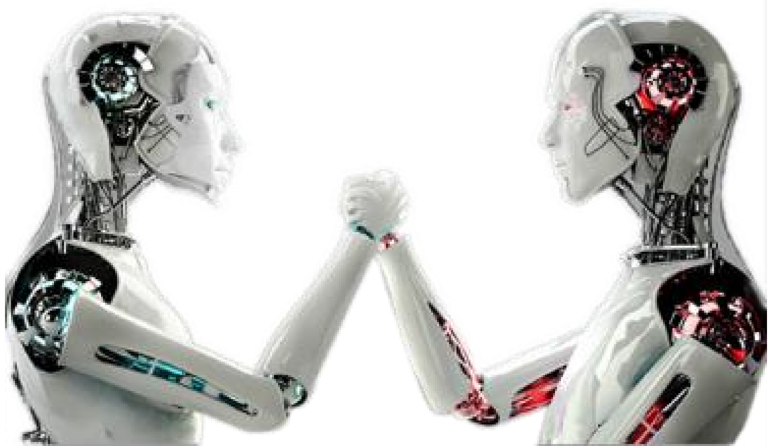
$$x_i \in \mathbb{R}^d, \quad \text{with } d = 1024$$

  - Train a SVM classifier

- Testing (Detection)
  - Sliding window classifier

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

Lab Task



Lab Task

1. Complete the exercises and questions in the Lab07_SVM_guide.pdf

2. Submit two result files with the same content to bb. The extensions of these two files are **ipynb** and **pdf**, respectively.

Lab1: 周三 上午3-4节 荔园6栋408机房
Lab2: 周三 下午7-8节 荔园6栋406机房
Lab3: 周二下午5-6节 荔园6栋409机房

Thanks

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