CS4801 : Perceptron and SVM

Sahely Bhadra

19/9/2018

1. Linear separability
2. Perceptron loss
3. Perceptron algorithm

Multi classes case

Choose class K to be the "reference class" and represent each of the other classes as a logistic function of the odds of class *k* versus class K:

$$\log \frac{P(y=1 \mid \mathbf{x})}{P(y=K \mid \mathbf{x})} = \mathbf{w}_1{}^{\mathsf{T}}\mathbf{x}$$

$$\log \frac{P(y=2 \mid \mathbf{x})}{P(y=K \mid \mathbf{x})} = \mathbf{w}_2{}^{\mathsf{T}}\mathbf{x}$$

$$\vdots$$

$$\log \frac{P(y=K-1 \mid \mathbf{x})}{P(y=K \mid \mathbf{x})} = \mathbf{w}_{K-1}{}^{\mathsf{T}}\mathbf{x}$$

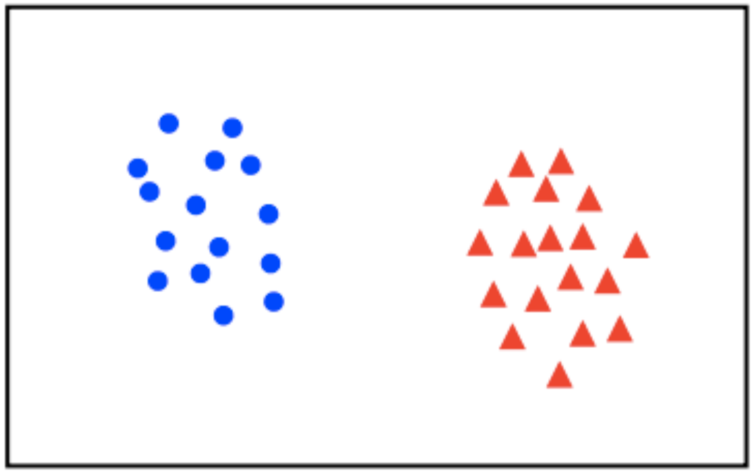
$$P(y=k \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_k{}^{\mathsf{T}}\mathbf{x})}{1+\sum_{l=1}^{K-1}\exp(\mathbf{w}_l{}^{\mathsf{T}}\mathbf{x})} \qquad P(y=K \mid \mathbf{x}) = \frac{1}{1+\sum_{l=1}^{K-1}\exp(\mathbf{w}_l{}^{\mathsf{T}}\mathbf{x})}$$

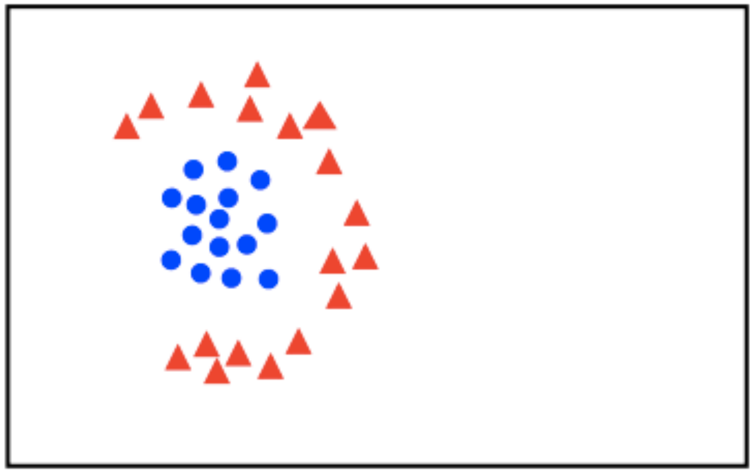
Classification

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.

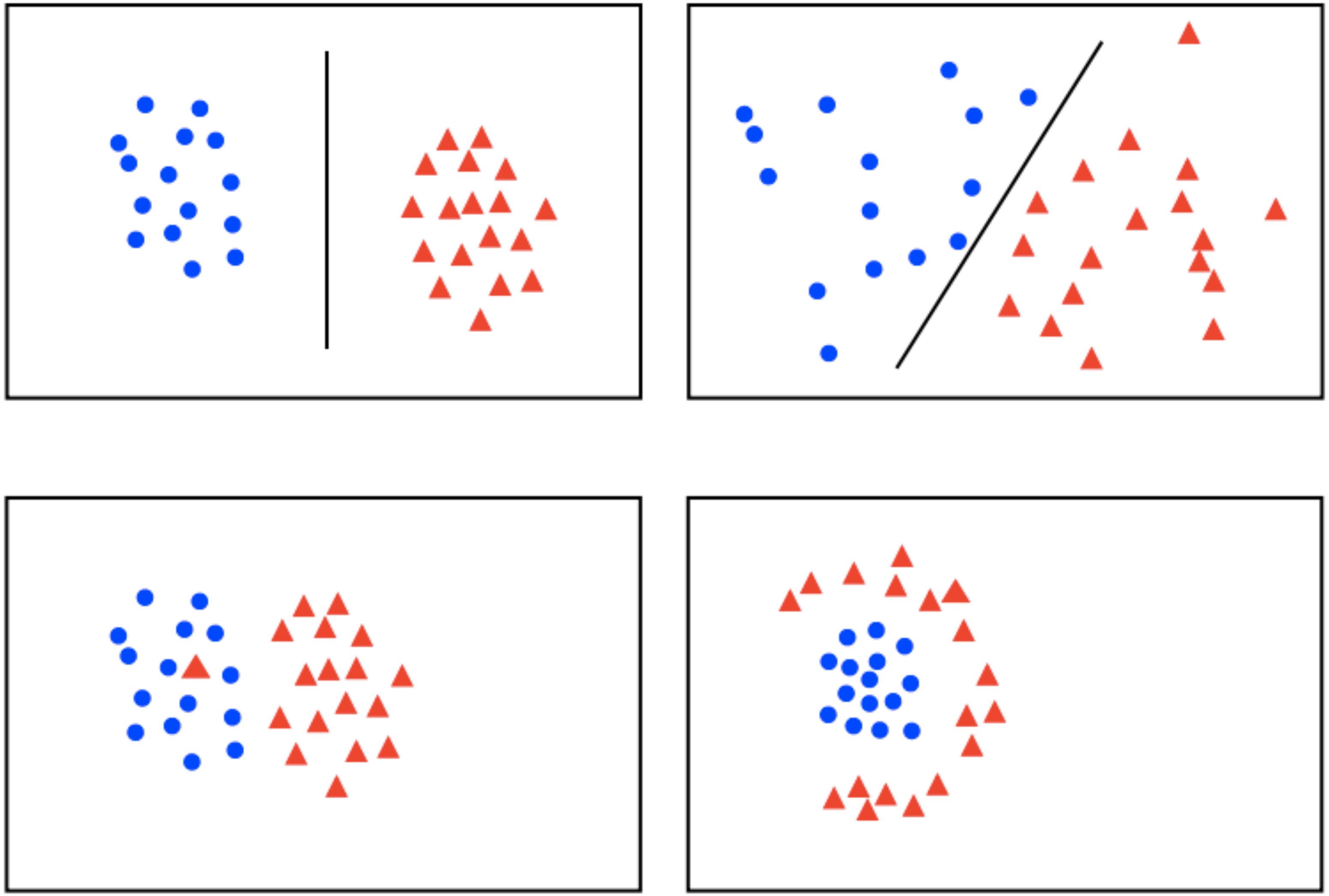




Linear Separability

linearly separable

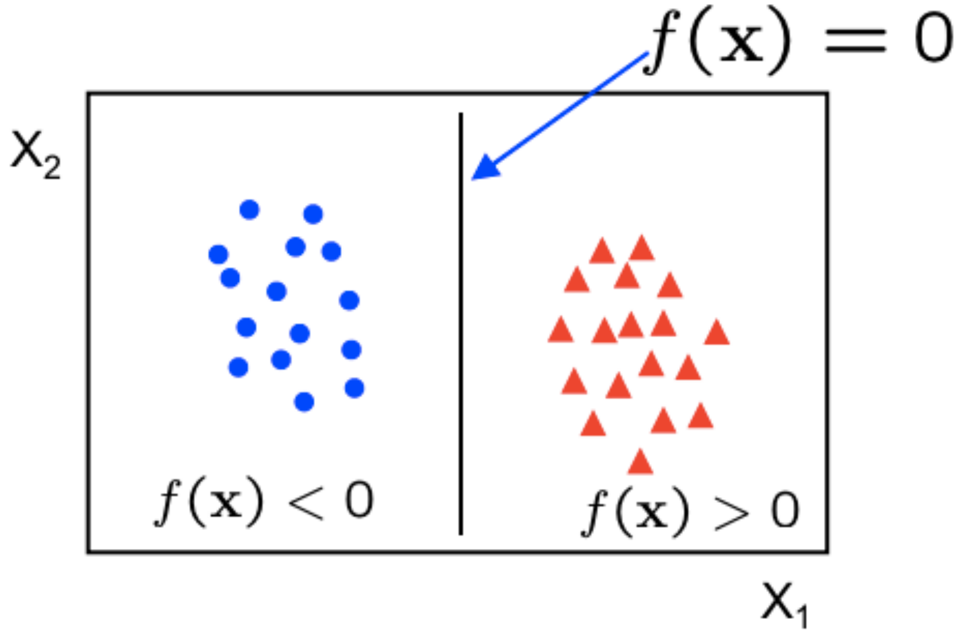
not linearly separable



Linear Classifier

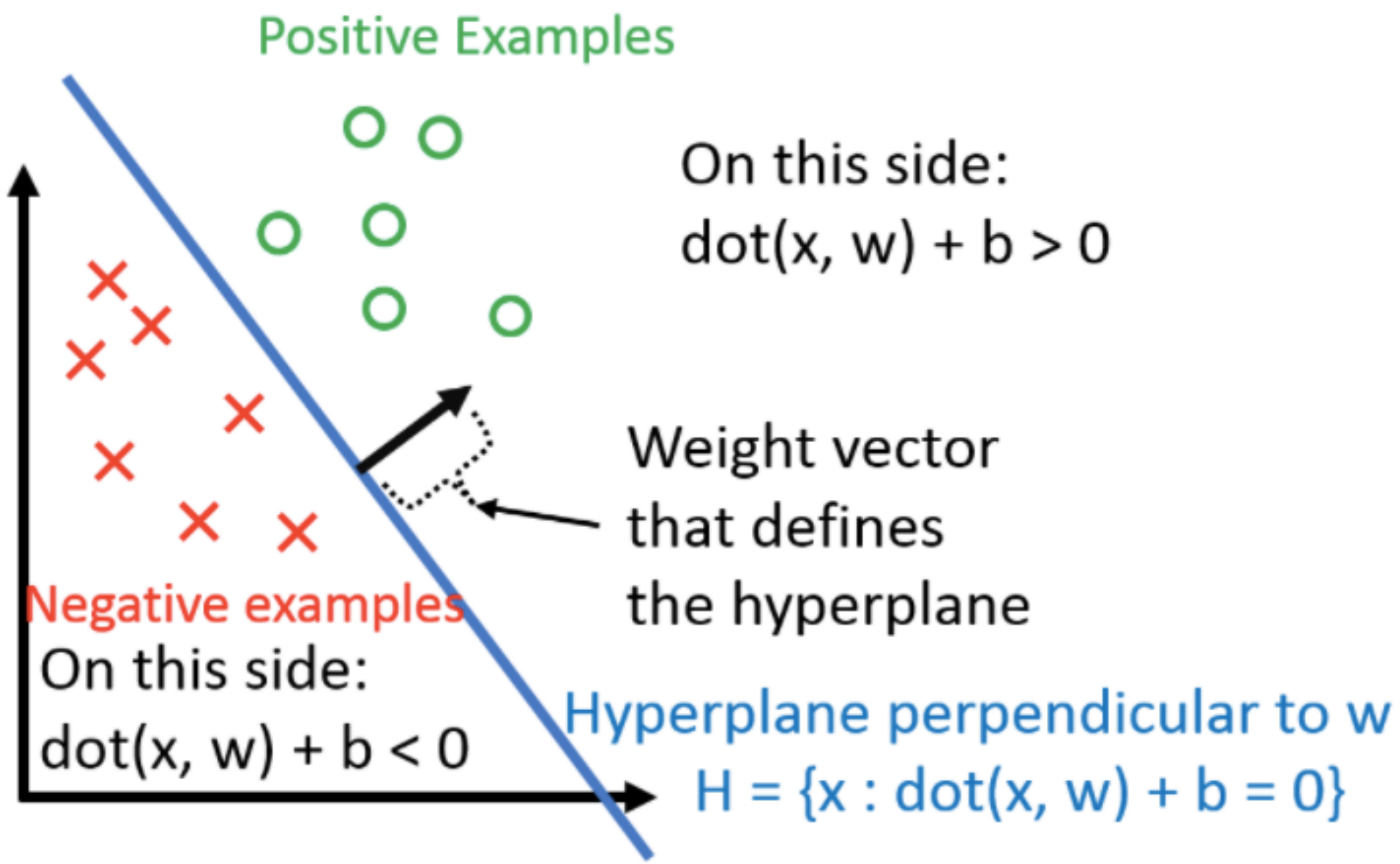
A linear classifier has the form

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



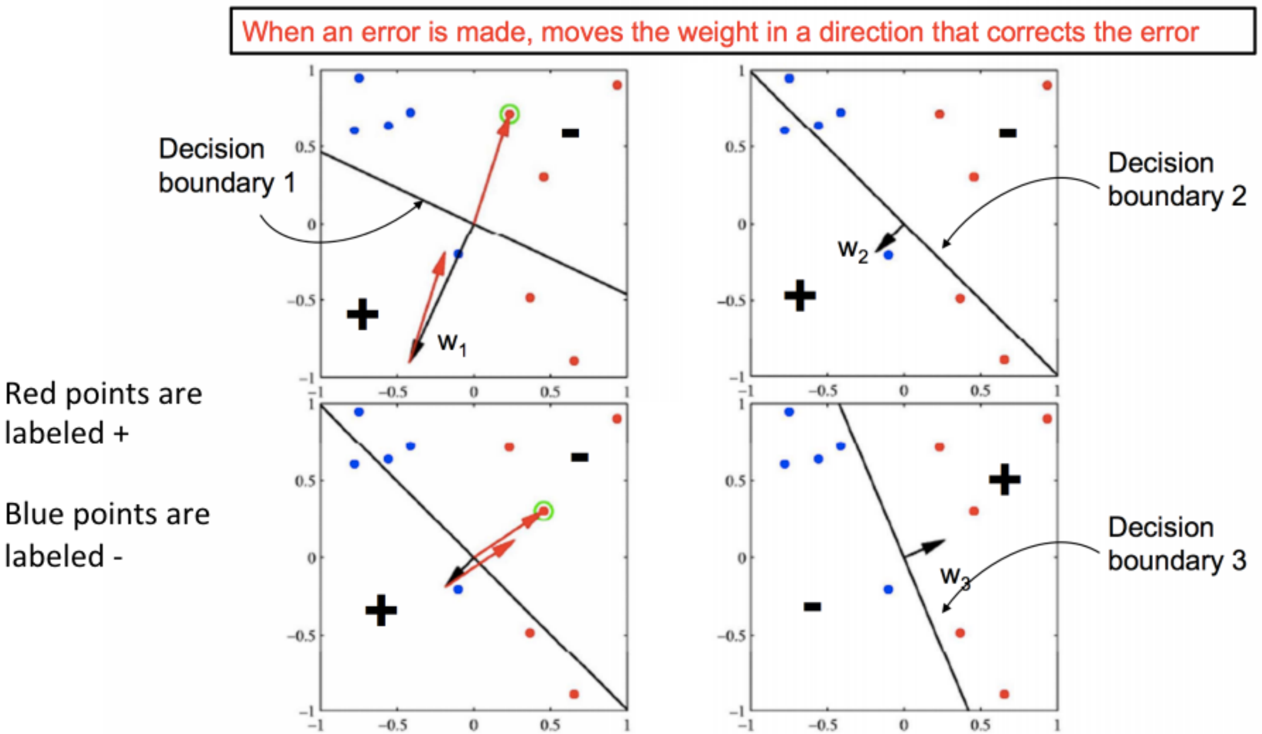
- in 2D the discriminant is a line
- $\mathbf{w}$ is the normal to the line, and b the bias
- $\mathbf{w}$ is known as the weight vector

Linear Classifier :weight vector that define the hyper plane



Perceptron

$$perceptron\ loss = \sum_i \max\{0, -y_i\ w^T x_i\}$$



Perceptron Algorithm

Initialize $\vec{w} = \vec{0}$
while TRUE **do**
  $m = 0$
  for $(x_i, y_i) \in D$ **do**
    if $y_i(\vec{w}^T \cdot \vec{x}_i) \leq 0$ **then**
      $\vec{w} \leftarrow \vec{w} + y\vec{x}$
      $m \leftarrow m + 1$
    end if
  end for
  if $m = 0$ **then**
    break
  end if
end while

Disadvantage : Will not converge for linearly non-separable data
Non-unique solution (few solutions have high generalisation error

Perceptron: convergence

Assume,

For all i, $||x_i|| < R$

There exists a w^* such that for all i ,

$y_i \, w^{*T} x_i >= \gamma > 0$

$|| \, w^* || = 1$

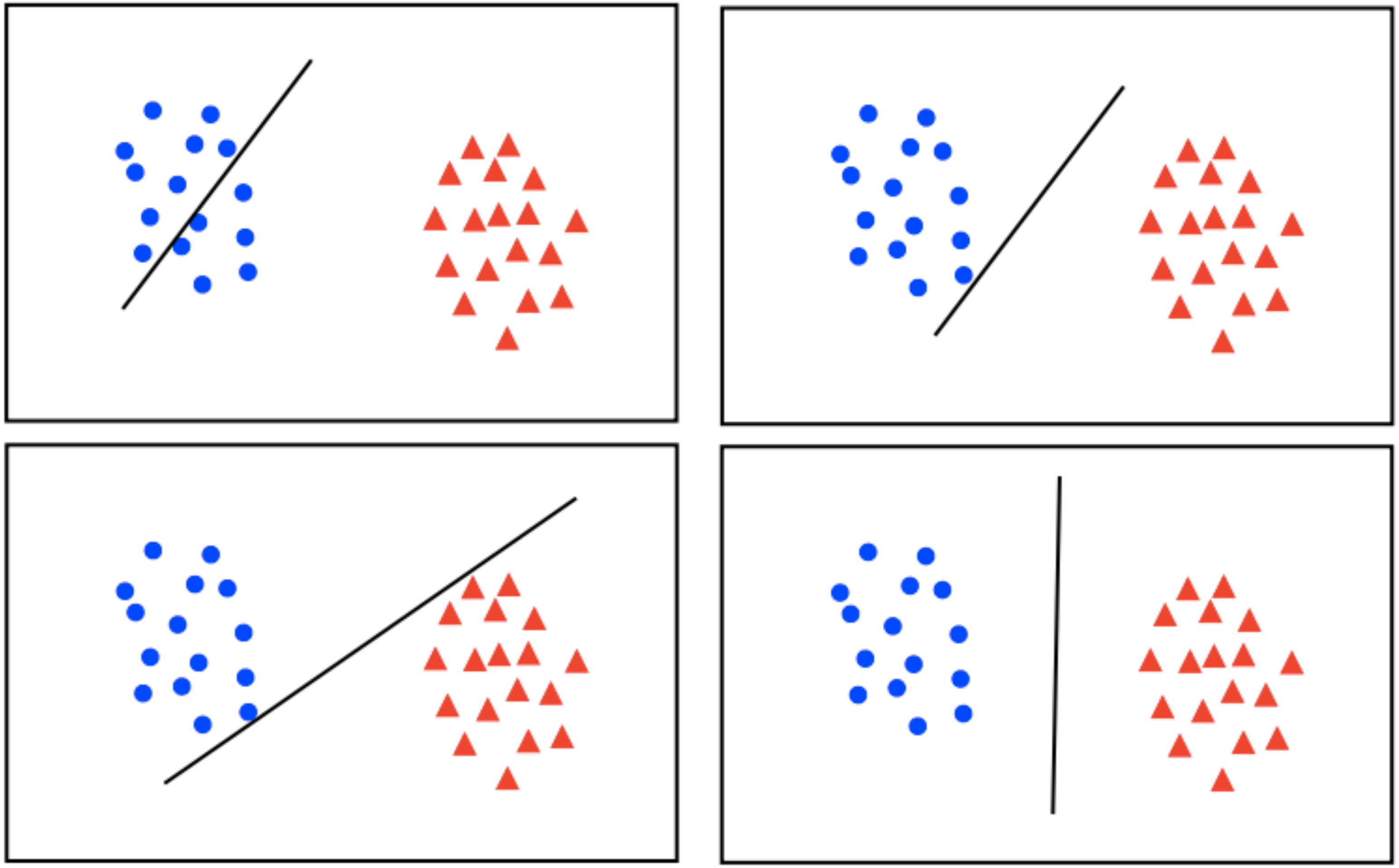
The Perceptron Learning Algorithm makes at most R^2 / γ^2 updates (after which it returns a separating hyperplane).

CS4801 : Support Vector Machine

Sahely Bhadra

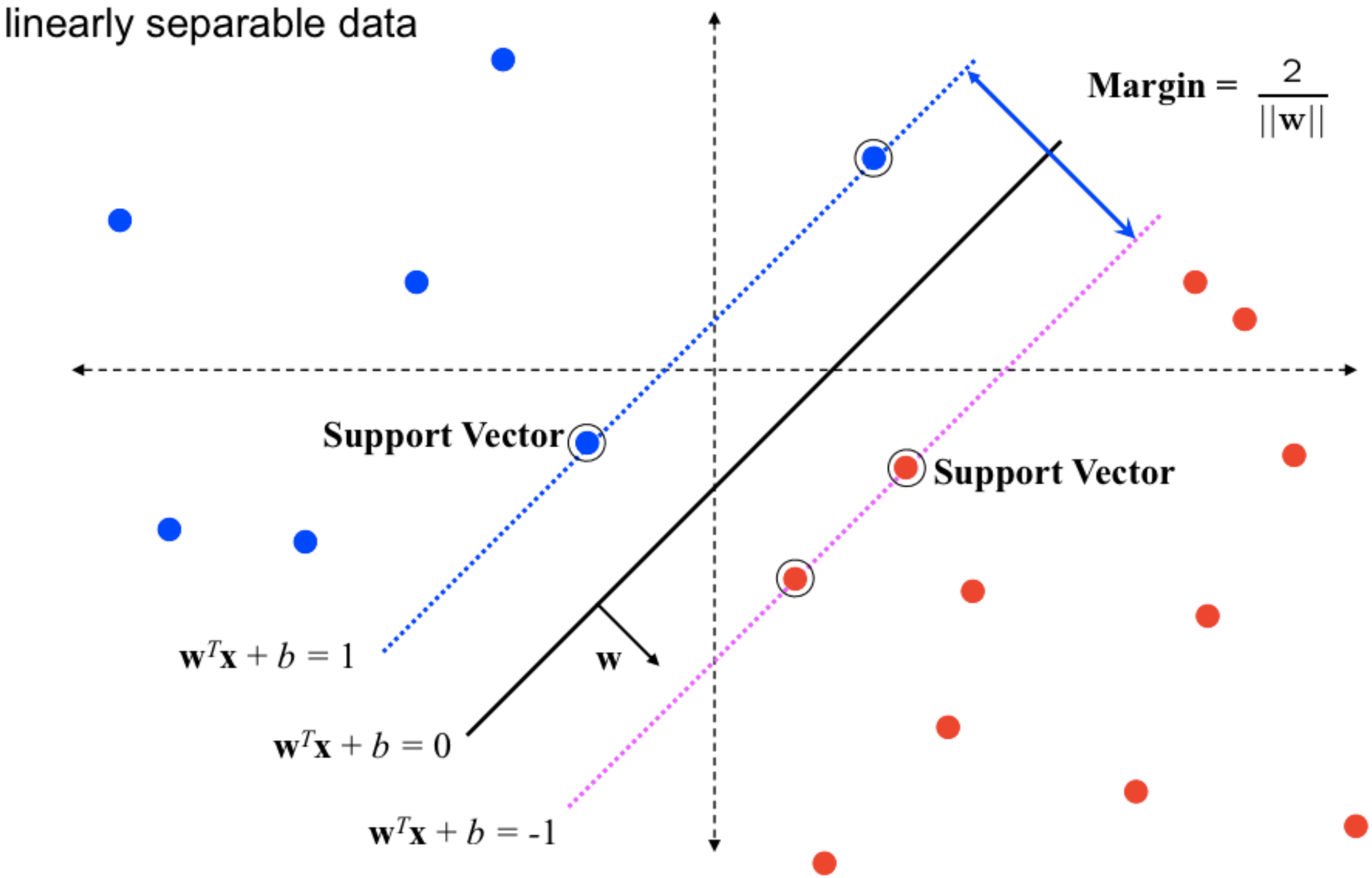
1. Margin for SVM
2. Loss function for SVM
3. Slack : linearly non separable data set
4. Pegasus : gradient based SVM solver

Maximum Margin



• maximum margin solution: most stable under perturbations of the inputs and hence better generalisation performance

Margin


linearly separable data
Margin = $\frac{2}{||\mathbf{w}||}$
Support Vector
Support Vector
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$
$\mathbf{w}$

Margin

- 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

- Then the margin is given by distance between two parallel line

$$\mathbf{w}^\top\mathbf{x}_+ + b = +1 \quad \text{and} \quad \mathbf{w}^\top\mathbf{x}_+ + b = -1$$

$$\frac{|+1 - b - (-1 - b)|}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$$

Perceptron with margin

The optimization problem becomes

$$\min_{\mathbf{w}\in\mathbb{R}^d} \quad ||\mathbf{w}||^2$$

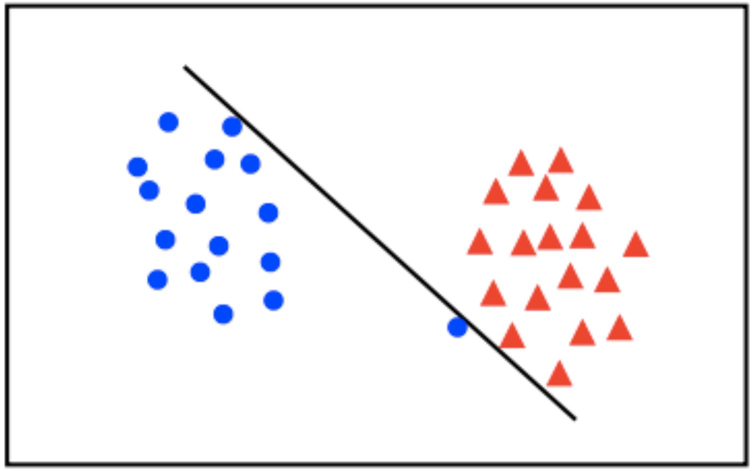
subject to

$$y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 \qquad \text{for } i = 1 \ldots N$$

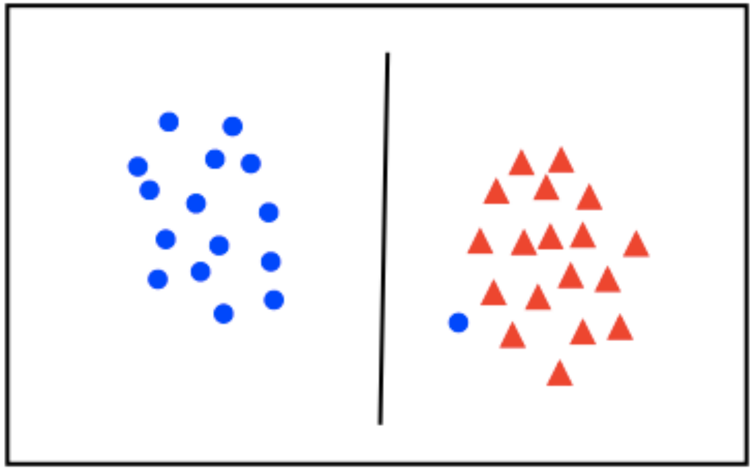
- This is now optimizing a *quadratic* function subject to *linear* constraints
- Quadratic optimization problems are a well-known class of mathematical programming problem, and many (intricate) algorithms exist for solving them (with many special ones built for SVMs)

DisAdvantage : Will not converge for linearly non-separable data

SVM: Trade off



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



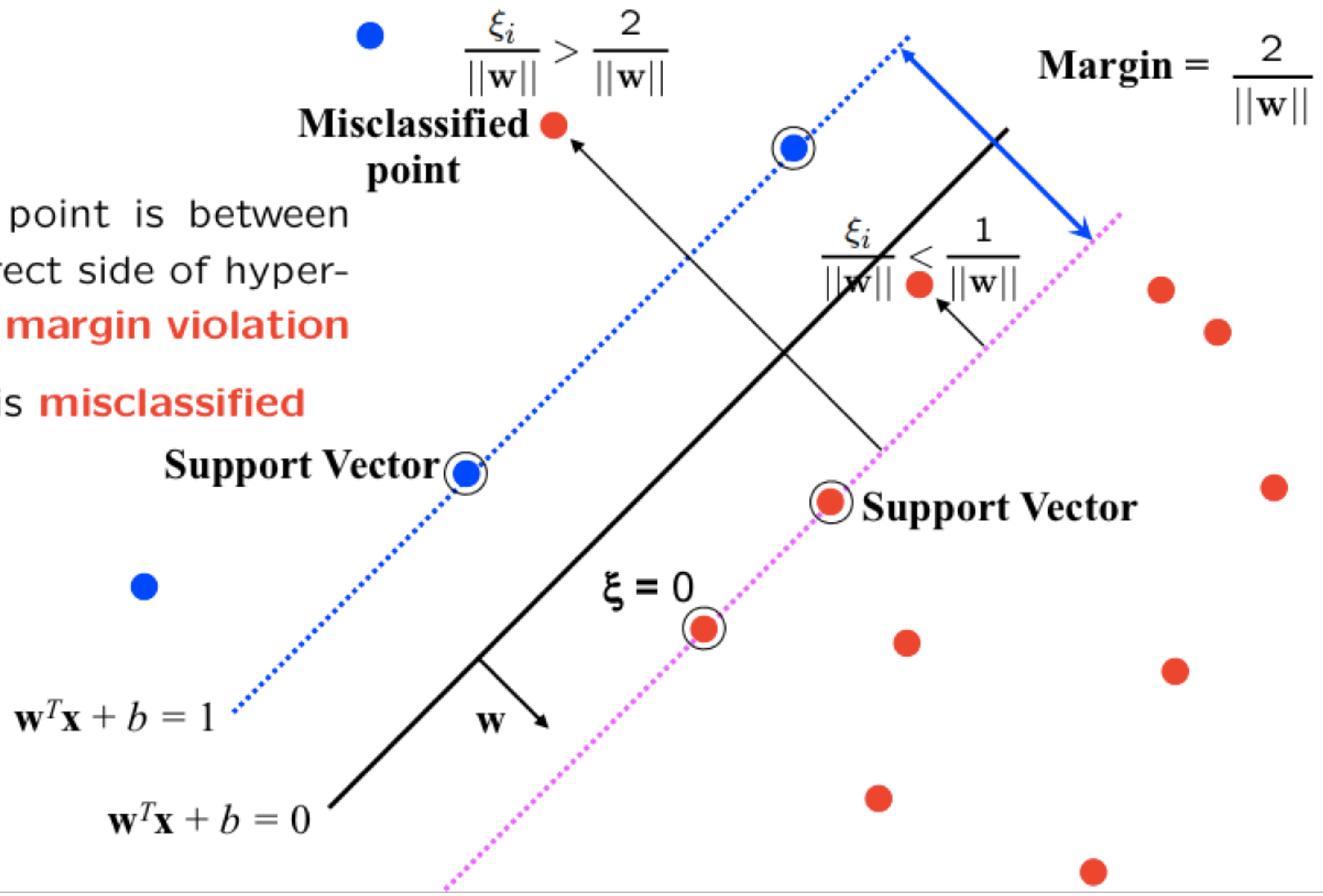
• but possibly the large margin solution is better, even though one constraint is violated

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

SVM: soft margin

$$\xi_i \geq 0$$

- 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**



SVM: Support Vector Machine

The optimization problem becomes

$$\min_{\mathbf{w}\in\mathbb{R}^d, \xi_i \in \mathbb{R}^+} ||\mathbf{w}||^2 + C\sum_i^N \xi_i$$

subject to

$$\begin{aligned} y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) &\geq 1 - \xi_i \text{ for } i = 1 \dots N \\ \xi_i &\geq 0 \end{aligned}$$

- 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.