POLITECNICO MILANO 1863

Machine Learning Methods for Communication Networks and Systems

Francesco Musumeci

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)

Politecnico di Milano, Milano, Italy

Part I – 4: Support Vector Machines

Outline

- Introduction
- Large margin classification
- Kernels
- SVM with Kernels
- SVM for multiple classes

Outline

- Introduction
- Large margin classification
- Kernels
- SVM with Kernels
- SVM for multiple classes

Introduction

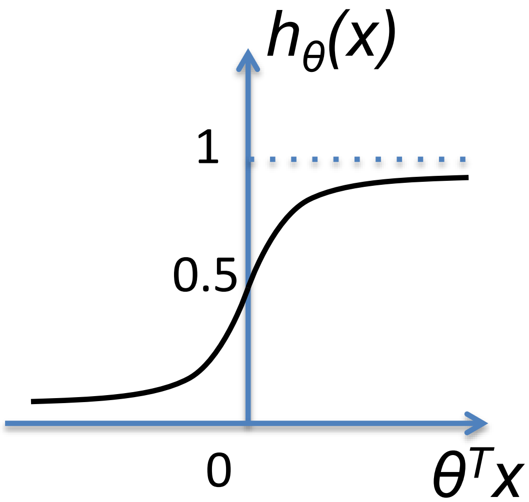
From logistic regression to SVM

- <u>Logistic regression</u>
  - hypotesis

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$\theta^T x \geq 0$ → *predict* $y=1$

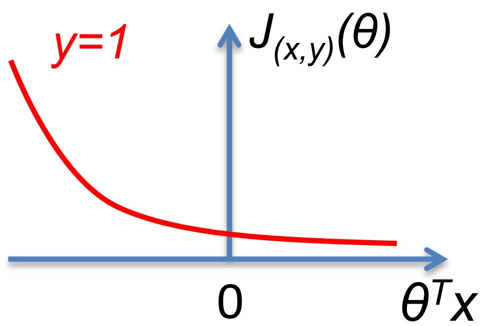
$\theta^T x < 0$ → *predict* $y=0$

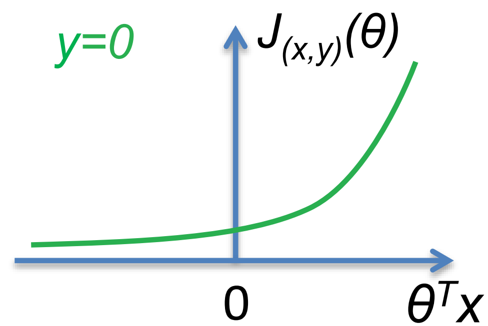


  - optimization objective:

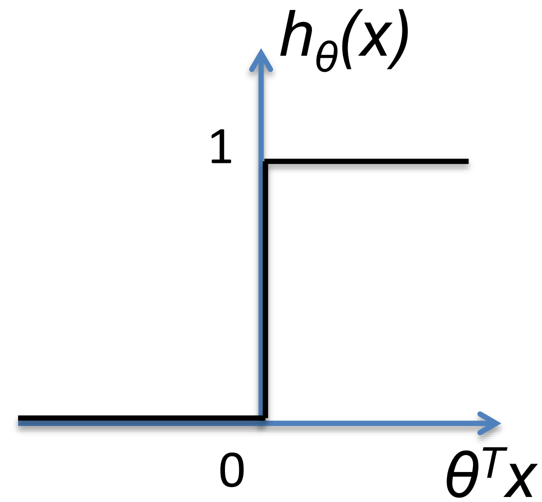
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

  - for a single training example (x,y):





Introduction
From logistic regression to SVM



- <u>Support Vector Machine</u>
  - hypotesis

$$h_\theta(x) = \begin{cases} 1 \text{ if } \theta^T x \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

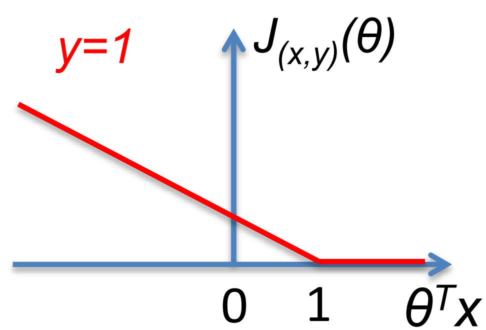
$\theta^T x \geq 0$ → predict y=1

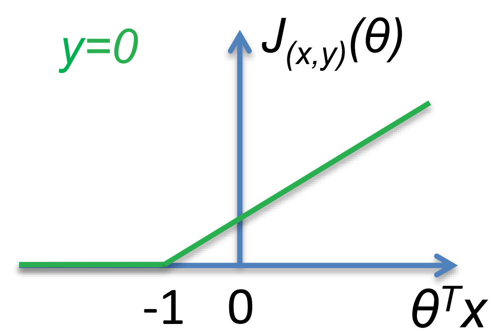
$\theta^T x < 0$ → predict y=0

  - optimization objective:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \boxed{\text{cost}_1(\theta^T x^{(i)})} + (1 - y^{(i)}) \boxed{\text{cost}_0(\theta^T x^{(i)})} \right]$$

  - for a single training example (x,y):





Outline

- Introduction
- Large margin classification
- Kernels
- SVM with Kernels
- SVM for multiple classes

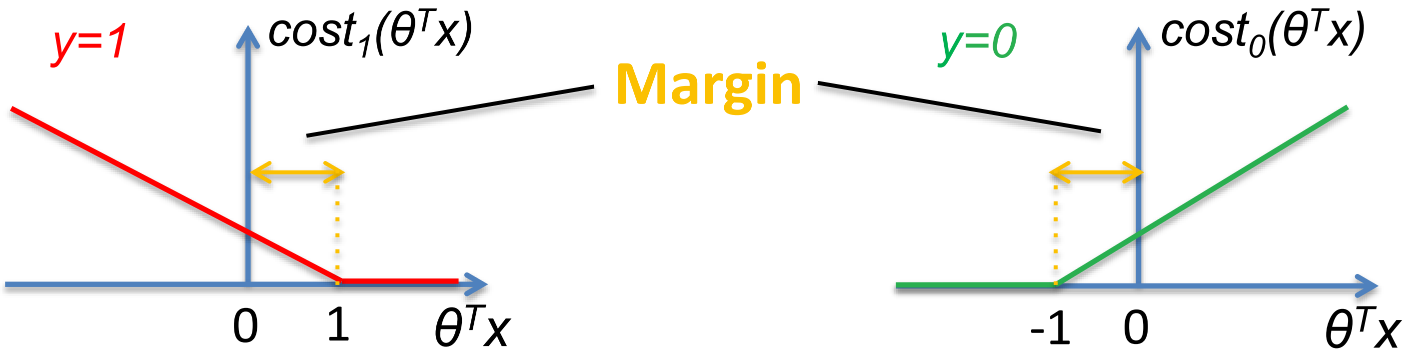
Large margin classification
Intuition

$$h_\theta(x) = \begin{cases} 1 \text{ if } \theta^T x \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

- Optimization objective:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \boxed{\mathrm{cost}_1(\theta^T x^{(i)})} + (1-y^{(i)}) \boxed{\mathrm{cost}_0(\theta^T x^{(i)})} \right]$$

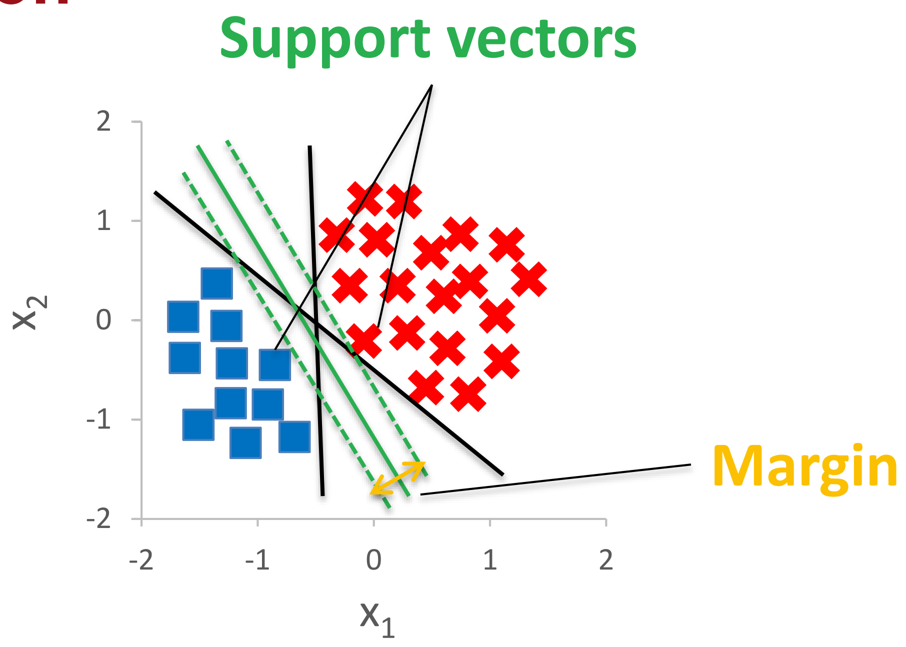
- How do we minimize $J(\theta)$?
  - $y^{(i)}=1$: we want θ s.t. $\theta^T x^{(i)} \geq 1$ (not just $\theta^T x^{(i)} \geq 0$)
  - $y^{(i)}=0$: we want θ s.t. $\theta^T x^{(i)} \leq -1$ (not just $\theta^T x^{(i)} < 0$)



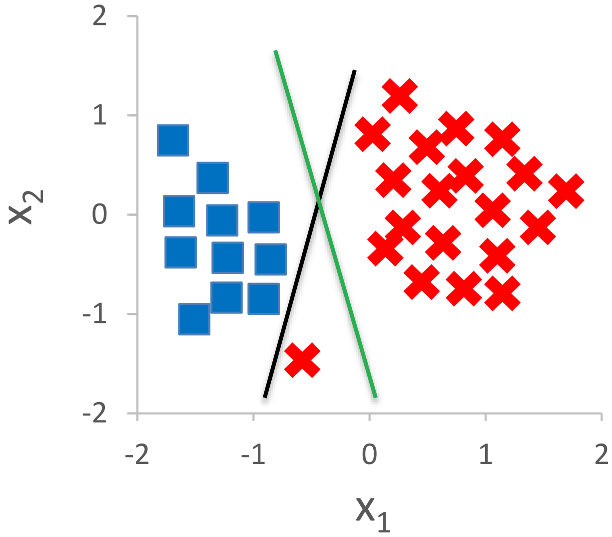
Large margin classification

Graphical view of margin

- Decision boundary
  - Which one is good enough?



- Decision boundary in presence of outliers
  - Take care of overfitting! (we'll see later)



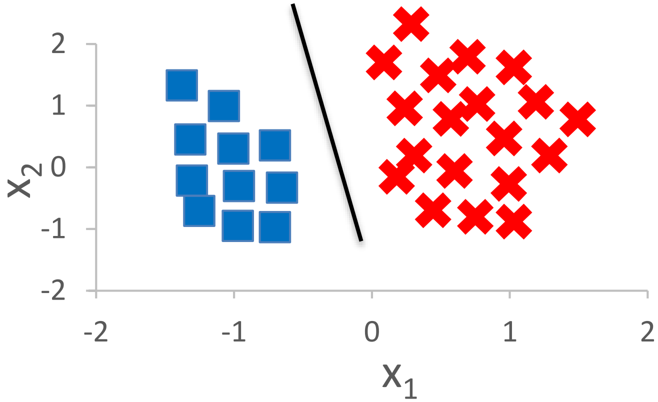
Outline

- Introduction
- Large margin classification
- Kernels
- SVM with Kernels
- SVM for multiple classes

Kernels
Motivation

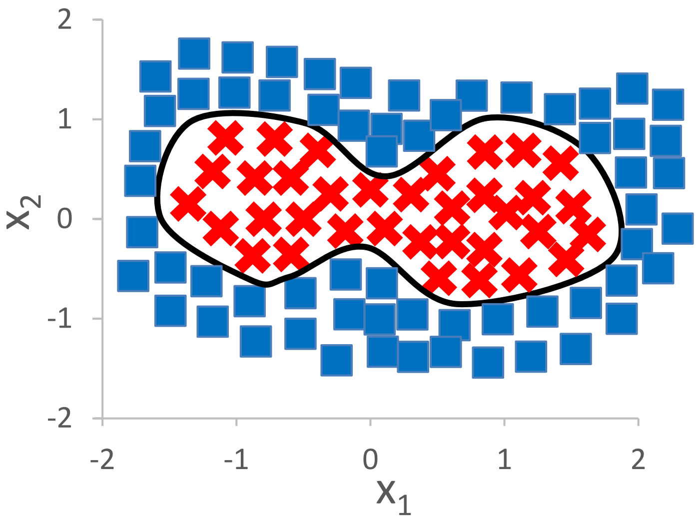
- Linear boundary

$$h_\theta(x)=\begin{cases}1 \text{ if } \theta^T x \geq 0\\ 0 \text{ otherwise}\end{cases}$$



- Non-linear boundary
  - Use polynomial features

$$h_\theta(x)=\begin{cases}1 \text{ if } \theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1 x_2+\theta_4 x_1^2+\theta_5 x_2^2+\ldots \geq 0\\ 0 \text{ otherwise}\end{cases}$$



  - Is it the best option?

→ No! Use *Kernels*

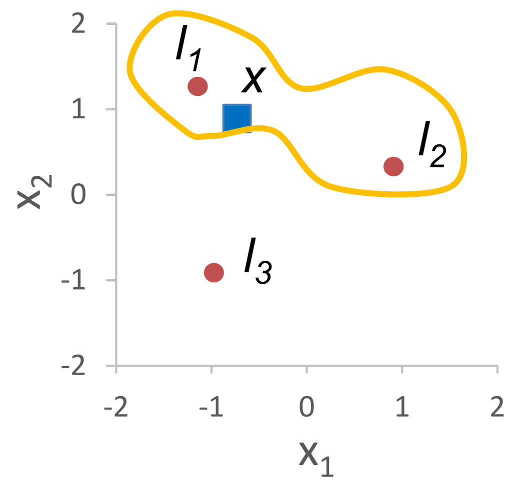
Kernels

Landmarks and similarity function

- Kernels transform "original" features vector *x* into a new set of features f_1, f_2, f_3, …
- New features are obtained measuring the *similarity* between *x* and *Landmarks* $l^{(1)}$, $l^{(2)}$, $l^{(3)}$, …

$$f_1 = \overbrace{similarity}^{\text{Kernel}}(x, l^{(1)})$$
$$f_2 = similarity(x, l^{(2)})$$
$$f_3 = similarity(x, l^{(3)})$$
…

$$h_\theta(x) = \begin{cases} 1 \text{ if } \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0 \\ 0 \text{ otherwise} \end{cases}$$



- Different kernels are possible
  - Gaussian Kernel→ $similarity(x, l^{(i)}) = \exp\left(-\frac{\left\|x - l^{(i)}\right\|^2}{2\sigma^2}\right)$
  - Polynomial, string, chi-square…
- Issues: choice of landmarks, kernel, parmeters tuning (e.g., σ^2)

Outline

- Introduction
- Large margin classification
- Kernels
- SVM with Kernels
- SVM for multiple classes

SVM with Kernels

- Given the training set $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$
  - $x^{(i)}$ is a n-dimensional features vector
- Choose ***landmarks***:
  - Typical setting $l^{(1)}=x^{(1)}$, $l^{(2)}=x^{(2)}$, …, $l^{(m)}=x^{(m)}$
- For each training example $(x^{(i)}, y^{(i)})$, compute features $f_1^{(i)}$, $f_2^{(i)}$, …, $f_m^{(i)}$, where $f_j^{(i)}$ = similarity$(x^{(i)}, l^{(j)})$
  - *N.B. using Gaussian kernel* → $f_i^{(i)}$ = similarity$(x^{(i)}, \underbrace{l^{(i)}}_{=x^{(i)}})=1$
  - Features space is changed from n- to m-dimension
- New hypothesis

$$h_\theta(x)=\begin{cases}1 \text{ if } \theta^T f \geq 0\\ 0 \text{ otherwise}\end{cases}$$

- To train the SVM (i.e., learn params theta), minimize function:

$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\text{cost}_1(\theta^T f^{(i)})+(1-y^{(i)})\text{cost}_0(\theta^T f^{(i)})\right]$$

Outline

- Introduction
- Large margin classification
- Kernels
- SVM with Kernels
- SVM for multiple classes

SVM for multiple classes

- Built-in multiclass SVM in many software packages
- Alternative: *one-vs-all*
  - Train k different SVMs
  - Get $\Theta^{(1)}, \Theta^{(2)}, \ldots, \Theta^{(k)}$
  - Classification for a new element x_{test}:
    - select $y_{test}=i$ s.t. $\Theta^{(i)T}x_{test}$ is maximum