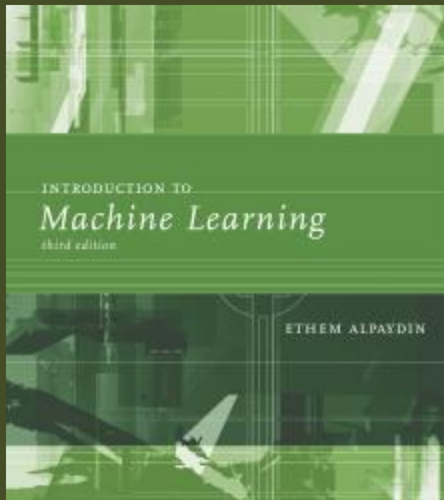
DANİEL GUERREİRO E SİLVA – FT / ENE / UNB

SUPPORT VECTOR MACHİNES



Lecture Slides for

INTRODUCTION TO MACHİNE LEARNİNG

3RD EDİTİON

ETHEM ALPAYDIN

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml3e

Kernel Machines

- 1960s: Vapnik *et al.* develops the Generalized Portrait algorithm.
- Approximately 30 years later, the Support Vector Machine (SVM) is designed by Vapnik and his colleagues at Bell Labs
- SVMs (or Kernel Machines, generally speaking) became widely-used and effective on several classification and function approximation problems.

Kernel Machines: benefits

- Discriminant-based: define the discriminant in terms of support vectors
- The use of kernel functions, application-specific measures of similarity
- No need to represent instances as vectors
- Convex optimization problems with a unique solution
- Good for high dimensional input data

Optimal Separating Hyperplane

Assume $\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^{N}$, where $r^t = \begin{cases} +1, & if\ \mathbf{x}^t \in C_1 \\ -1, & if\ \mathbf{x}^t \in C_2 \end{cases}$

and the classes are linearly separable.

- Find $\mathbf{w}$ and w_0 such that

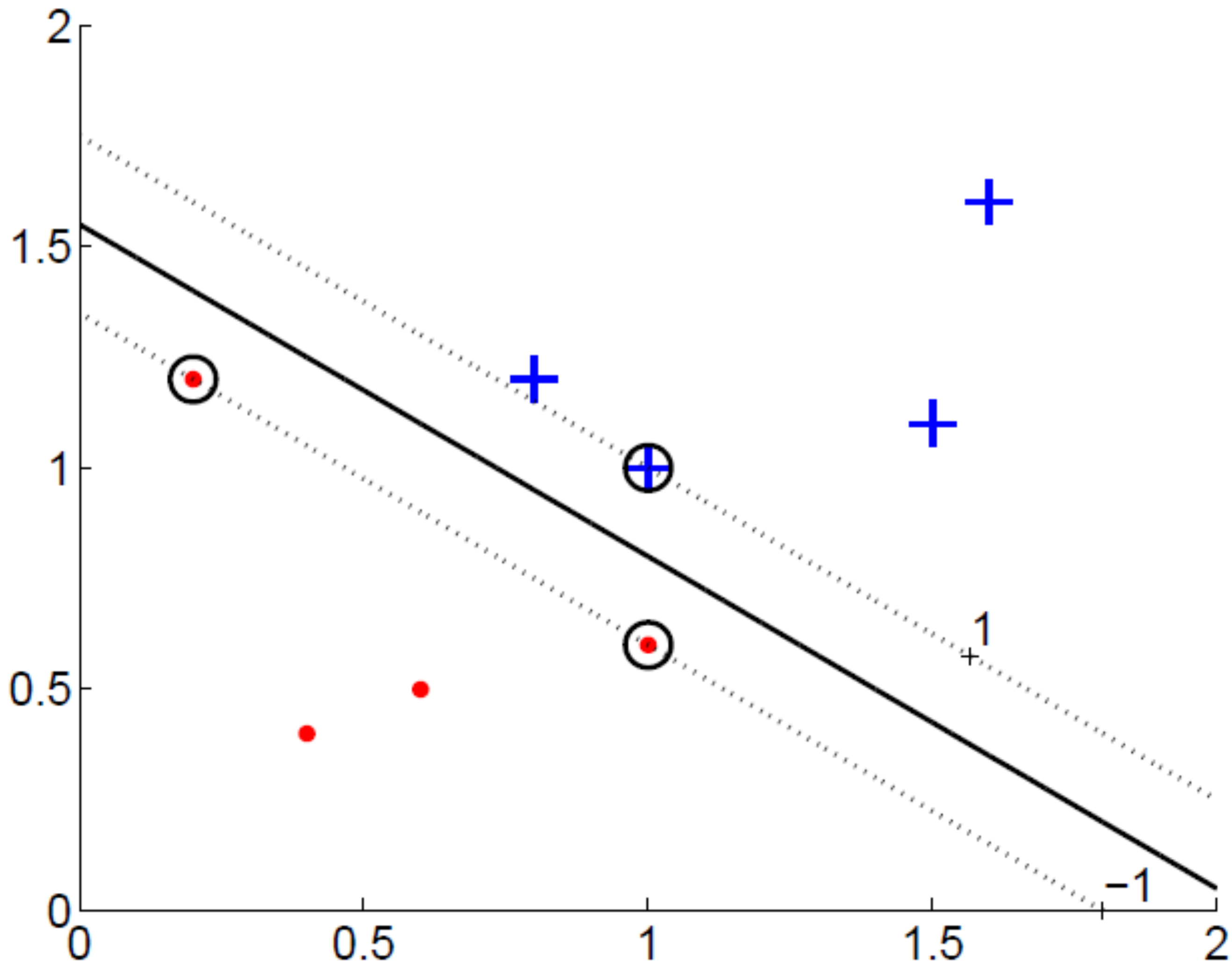
$$g(\mathbf{x}^t) = \mathbf{w}^T\mathbf{x}^t + w_0 \geq +1 \ \text{for}\ r^t = +1$$
$$g(\mathbf{x}^t) = \mathbf{w}^T\mathbf{x}^t + w_0 \leq -1 \ \text{for}\ r^t = -1$$

which can be rewritten as

$$r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1$$

(Cortes and Vapnik, 1995; Vapnik, 1995)

Margin



Lagrangian Multipliers method for optimization

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1, \forall t$$

- Use Lagrance multipliers to write as an unconstrained problem (Primal problem)

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha^t [r^t(\mathbf{w}^T\mathbf{x}^t + w_0) - 1]$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha^t r^t(\mathbf{w}^T\mathbf{x}^t + w_0) + \sum_{t=1}^{N} \alpha^t$$

- L_p should be minimized with respect to $\mathbf{w}$ and w_0 and maximized with respect to α^t

Lagrangian Multipliers method for optimization

- **Dual** problem

$$L_d = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \mathbf{w}^T\sum_{t=1}^{N}\alpha^t r^t \mathbf{x}^t - w_0\sum_{t=1}^{N}\alpha^t r^t + \sum_{t=1}^{N}\alpha^t$$

$$= -\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{t=1}^{N}\alpha^t$$

$$= -\frac{1}{2}\sum_{t=1}^{N}\sum_{s=1}^{N}\alpha^t\alpha^s r^t r^s \left(\mathbf{x}^t\right)^T\mathbf{x}^s + \sum_{t=1}^{N}\alpha^t$$

subject to $\sum_{t=1}^{N}\alpha^t r^t = 0$ and $\alpha^t \geq 0\ \forall t$

Support Vector Machine (SVM)

$$\max_{\alpha^t} -\frac{1}{2}\sum_{t=1}^{N}\sum_{s=1}^{N}\alpha^t\alpha^s r^t r^s\,(\mathbf{x}^t)^T\mathbf{x}^s + \sum_{t=1}^{N}\alpha^t$$

$$\text{subject to} \begin{cases} \displaystyle\sum_{t=1}^{N}\alpha^t r^t = 0 \\ \alpha^t \geq 0\ \forall t \end{cases}$$

- Quadratic programming methods can solve this problem

Support Vector Machine (SVM)

- Most α^t are 0 and only a small number have $\alpha^t \geq 0$; they are the support vectors

$$\{\mathbf{x}^t \colon \mathbf{x}^t \in \mathcal{X} \text{ and } \alpha^t \geq 0\}$$

- $\mathbf{w}$ is written as the weighted sum of the support vectors:

$$\mathbf{w} = \sum_{\alpha^t > 0} \alpha^t r^t \mathbf{x}^t$$

- Testing: calculate $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$ and choose C_1 if $g(x) > 0$ and C_2otherwise

Soft Margin Hyperplane

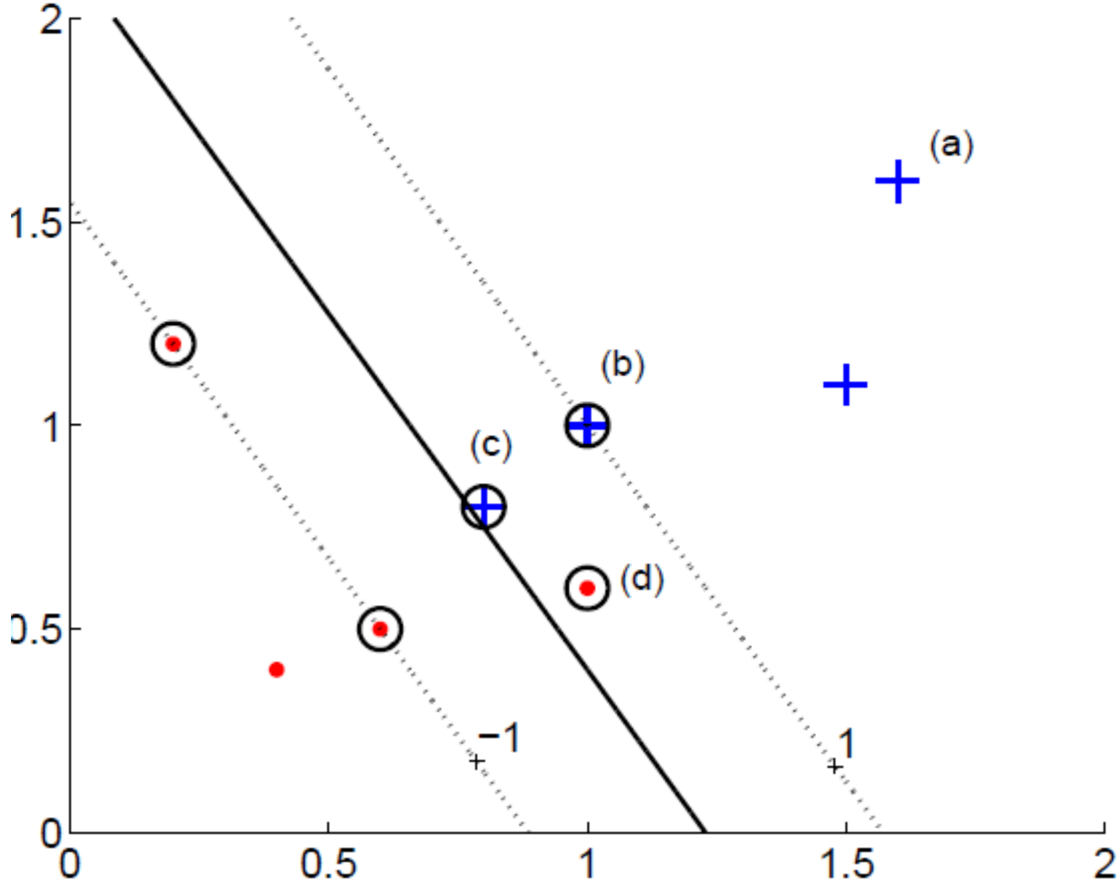
- Classes are not linearly separable: adopt constraints with a slack variable

$$r^t(\mathbf{w}^T\mathbf{x}^t + w_0) \geq 1 - \xi^t, \qquad \forall t$$

- It is included a soft error function as a penalty term, new primal problem is

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{N}\xi^t - \sum_{t=1}^{N}\alpha^t[r^t(\mathbf{w}^T\mathbf{x}^t + w_0) - 1 + \xi^t] - \sum_{t=1}^{N}\mu^t\xi^t$$

μ^t are new Lagrange multipliers to ensure $\xi^t \geq 0$



$\xi^t = 0 \rightarrow$ no problem (a, b)
$0 < \xi^t < 1 \rightarrow$ correctly classified, but in the margin (c)
$\xi^t \geq 1 \rightarrow$ misclassified (d)

Soft Margin SVM

- Dual problem

$$L_d = -\frac{1}{2}\sum_{t=1}^{N}\sum_{s=1}^{N} \alpha^t \alpha^s r^t r^s \left(\mathbf{x}^t\right)^T \mathbf{x}^s + \sum_{t=1}^{N} \alpha^t$$

$$\text{subject to} \begin{cases} \sum_{t=1}^{N} \alpha^t r^t = 0 \\ 0 \le \alpha^t \le C, \forall t \end{cases}$$

- Again, just some values α^t will be nonzero and represent the support vectors, such that

$$\mathbf{w} = \sum_{\alpha^t > 0} \alpha^t r^t \mathbf{x}^t$$

Kernel Trick (Non-linear SVM)

- Preprocess input $\mathbf{x}$ by basis functions

$$\mathbf{z} = \boldsymbol{\varphi}(\mathbf{x}) \;\Rightarrow\; g(\mathbf{z}) = \mathbf{w}^T\mathbf{z} = \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x})$$

- The dual optimization problem is now

$$L_d = -\frac{1}{2}\sum_{t=1}^{N}\sum_{s=1}^{N}\alpha^t\alpha^s r^t r^s\,\boldsymbol{\varphi}(\mathbf{x}^t)^T\boldsymbol{\varphi}(\mathbf{x}^s) + \sum_{t=1}^{N}\alpha^t$$

$$\text{subject to}\begin{cases}\displaystyle\sum_{t=1}^{N}\alpha^t r^t = 0\\ 0 \le \alpha^t \le C, \forall t\end{cases}$$

Kernel Trick (Non-linear SVM)

- It is an equivalent problem if $\boldsymbol{\varphi}(\mathbf{x}^t)^T\boldsymbol{\varphi}(\mathbf{x}^s)$ is replaced by a Kernel Function:

$$\max L_d = -\frac{1}{2}\sum_{t=1}^{N}\sum_{s=1}^{N}\alpha^t\alpha^s r^t r^s K(\mathbf{x}^t,\mathbf{x}^s) + \sum_{t=1}^{N}\alpha^t$$

$$\text{subject to}\begin{cases}\displaystyle\sum_{t=1}^{N}\alpha^t r^t = 0\\ 0 \le \alpha^t \le C, \forall t\end{cases}$$

Kernel Trick (Non-linear SVM)

- The SVM solution becomes

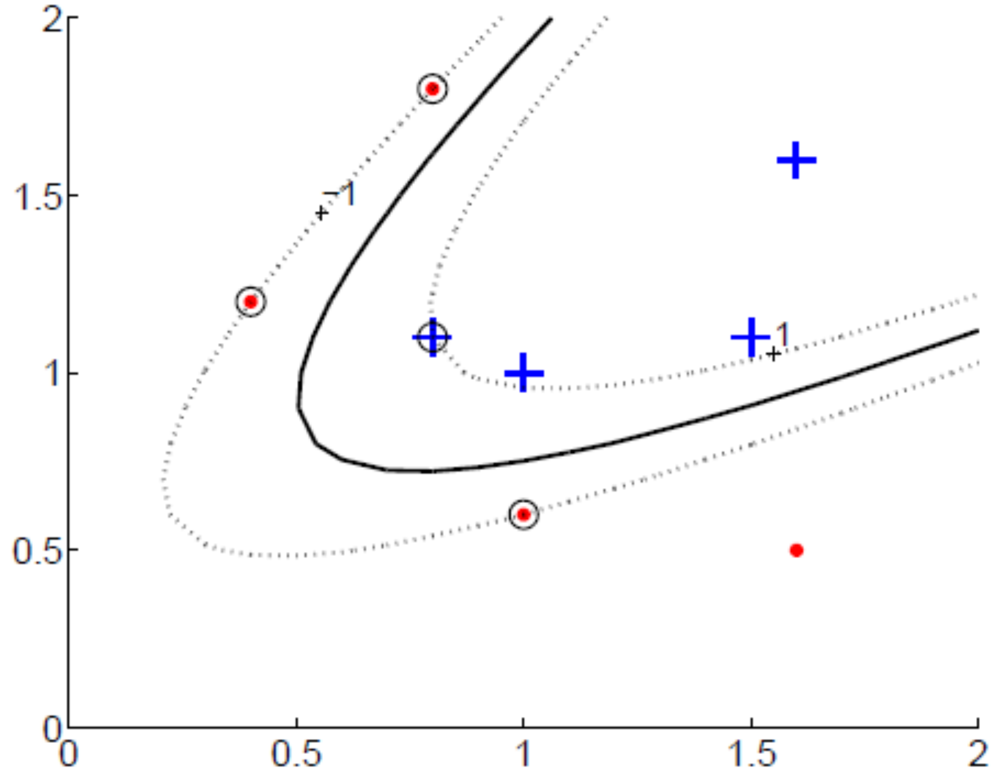
$$\mathbf{w} = \sum_t \alpha^t r^t \mathbf{z}^t = \sum_t \alpha^t r^t \boldsymbol{\varphi}(\mathbf{x}^t)$$

$$\begin{aligned} g(\mathbf{x}) &= \mathbf{w}^T \varphi(\mathbf{x}) \\ &= \sum_t \alpha^t r^t \boldsymbol{\varphi}(\mathbf{x}^t)^T \boldsymbol{\varphi}(\mathbf{x}) \\ &= \sum_t \alpha^t r^t K(\mathbf{x}^t, \mathbf{x}) \end{aligned}$$

Vectorial Kernels

- Polynomials of degree q:

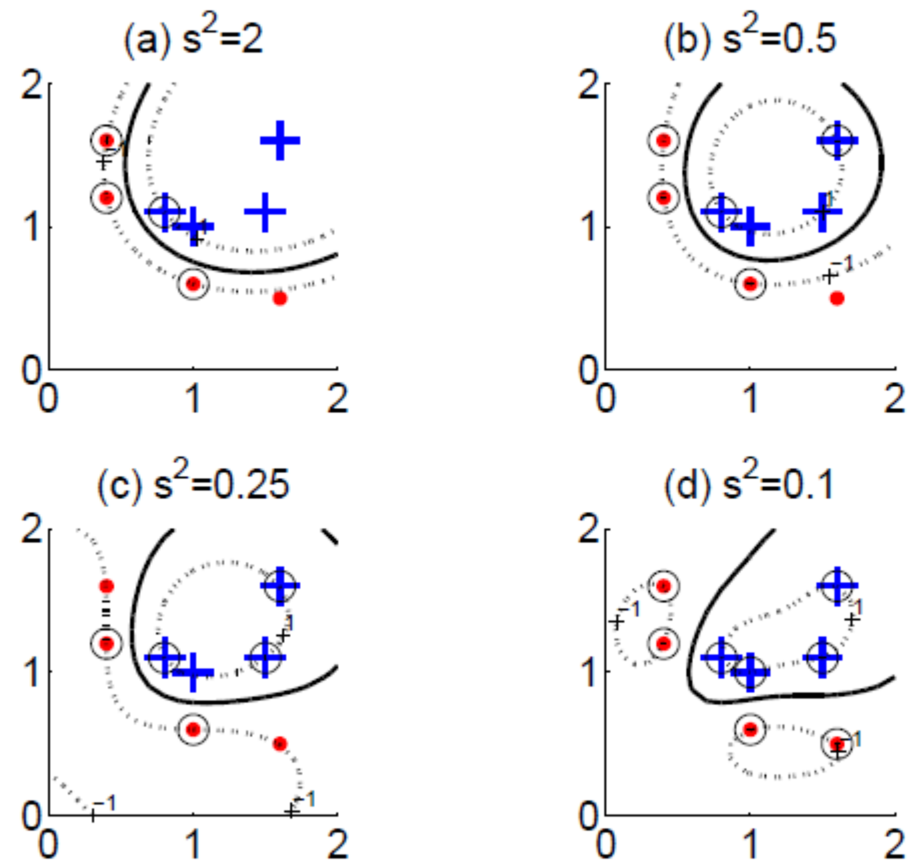
$$K(\mathbf{x}^t, \mathbf{x}) = (\mathbf{x}^T \mathbf{x}^t + 1)^q$$



Vectorial Kernels

- Radial-Basis functions:

$$K(\mathbf{x}^t, \mathbf{x}) = e^{\left(-\frac{\|\mathbf{x}^t - \mathbf{x}\|^2}{2s^2}\right)}$$



Vectorial Kernels

- Mahalanobis

$$K(\mathbf{x}^t, \mathbf{x}) = e^{\left[-\frac{1}{2}(\mathbf{x}^t - \mathbf{x})^T \mathbf{S}^{-1}(\mathbf{x}^t - \mathbf{x})\right]}$$

- Sigmoidal functions

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

Final remarks

- Kernel “engineering”
- Defining good measures of similarity
- String kernels, graph kernels, image kernels, …
- Vert, J., Tsuda, K., & Schölkopf, B. (2004). A primer on kernel methods. *Kernel Methods in Computational Biology*, (1992), 35–70.
- http://www.kernel-machines.org/tutorials