Support Vector Machine for Non-Linearly Separable Patterns

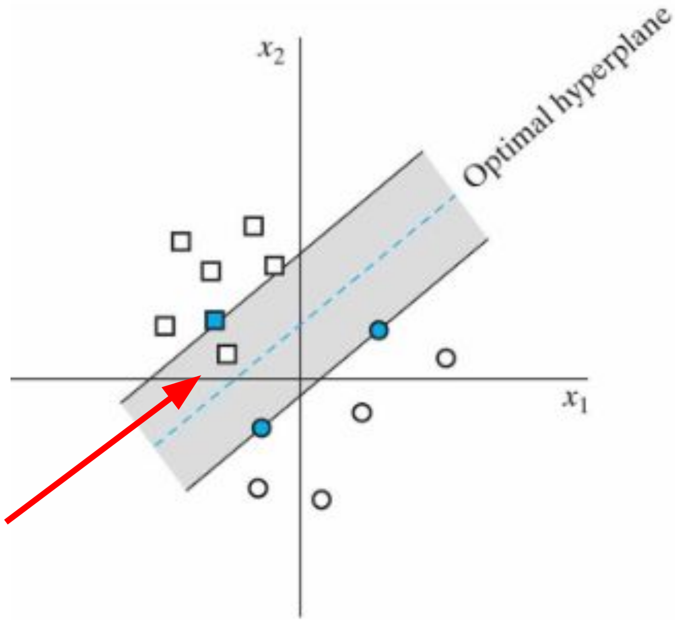
INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Non-linearly Separable Data

- Linear SVM with Soft Margin
- Non-linear SVM
    - Kernel SVM

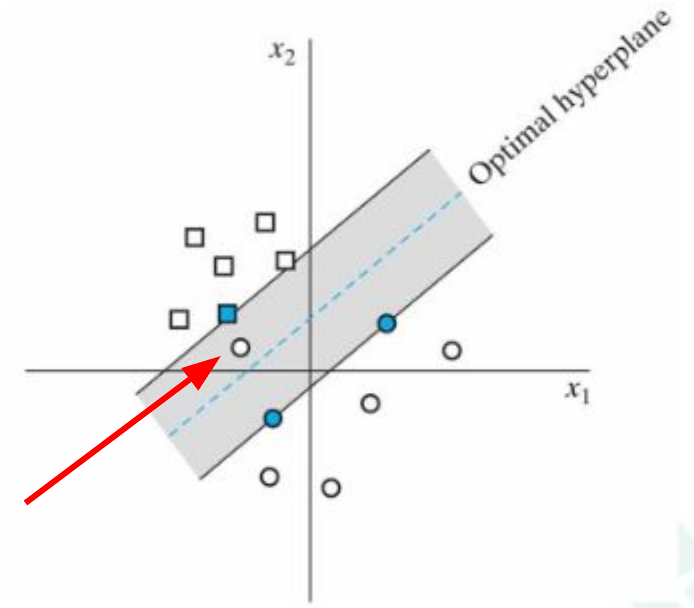
Non-Separable Patterns

- $y_i * (\mathbf{w}^T \mathbf{x_i} + b) >= 1, \forall\ i=1,2, \dots N$
- *Violation:*
  - $\{(x_i, y_i\}$ falls on the right side of the decision surface



Non-Separable Patterns

- $d_i*(\mathbf{w}^T\mathbf{x}_\mathbf{i} + b) >= 1, \forall\ i=1,2, ...N$
- *Violation:*
  - $\{(x_i, y_i\}$ falls on the wrong side of the decision surface



Slack Variables

- Let $\{\varepsilon_i\}^N_{i=1}$ >=0
  - $y_i * (\mathbf{w}^\mathbf{T}\mathbf{x_i} + b) >= 1 - \varepsilon_i, \forall\ i=1,2, \ldots N$
- ε_i are known as slack variables:
  - Deviation of a data point from the ideal condition of pattern separability
    - $0 <= \varepsilon_i <=1$: Data point falls on the right side of the decision surface
    - $\varepsilon_i > 1$: Data point falls on the wrong side of the decision surface
- Support vectors are those particular data points that satisfy the equation precisely even if $\varepsilon_i > 0$

Optimal Hyperplane

- The cost function
  - $\Phi(\mathbf{w}, \varepsilon) = ½\, \mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum^{N}_{i=1}\varepsilon_i$
  - C: Regularization parameter that controls the tradeoff between the complexity of the machine and the number of nonseparable patterns.
- High value of C-> High confidence in the quality of the training data
- Small value of C -> Less emphasis on the training data
- C has to be selected by the user *experimentally*.

Primal Problem

- Given the training sample $\{(x_i, y_i\}^N_{i=1}$, find the optimum values of the weight vector **w** and bias b such that they satisfy the constraints
  - $y_i{*}(\mathbf{w}^T\mathbf{x_i} + b) >= 1 - \varepsilon_i, \forall\ i=1,2, \ldots N$
  - $\varepsilon_i >= 0, \forall\ i$
- and such that the weight vector **w** and the slack variables ε_i minimize the cost functional
  - $\Phi(\mathbf{w}, \varepsilon) = 1/2\mathbf{w}^T\mathbf{w} + C\sum^N_{i=1}\varepsilon_i$

  where C is user-specified positive parameter.

- *Exercise: Apply the Lagrangian multipliers to get the dual*

Dual Problem

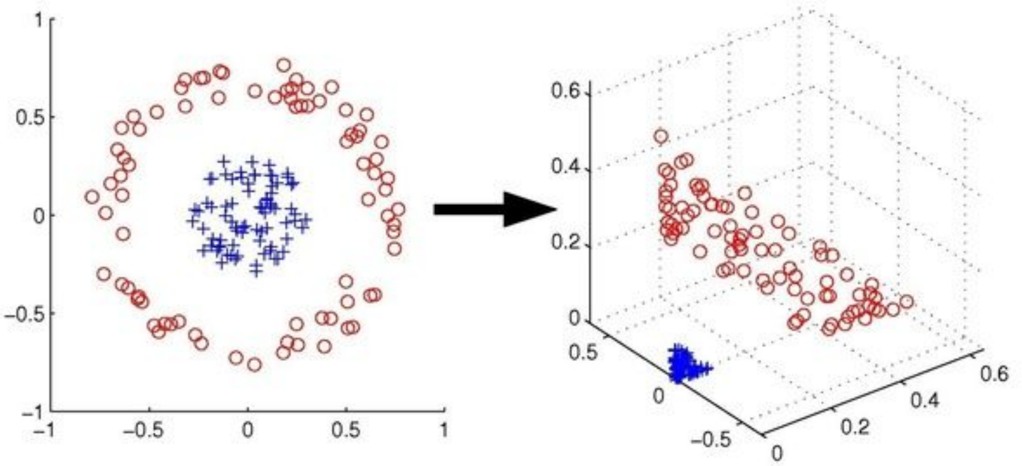
- Given the training sample $\{(x_i, y_i\}^N_{i=1}$, find the Lagrange multipliers α_i that maximize the objective function
  - $Q(\alpha) = \boldsymbol{\Sigma}^N_{i=1}\alpha_i - 1/2\Sigma^N_{i=1}\Sigma^N_{j=1}\alpha_i\alpha_j y_i y_j \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$, S.T.
    - $\Sigma^N_{i=1}\alpha_i y_i = 0$
    - $0 <= \alpha_i <= C, \forall\ i = 1,2, \ldots.N$

  where C is user-specified positive parameter

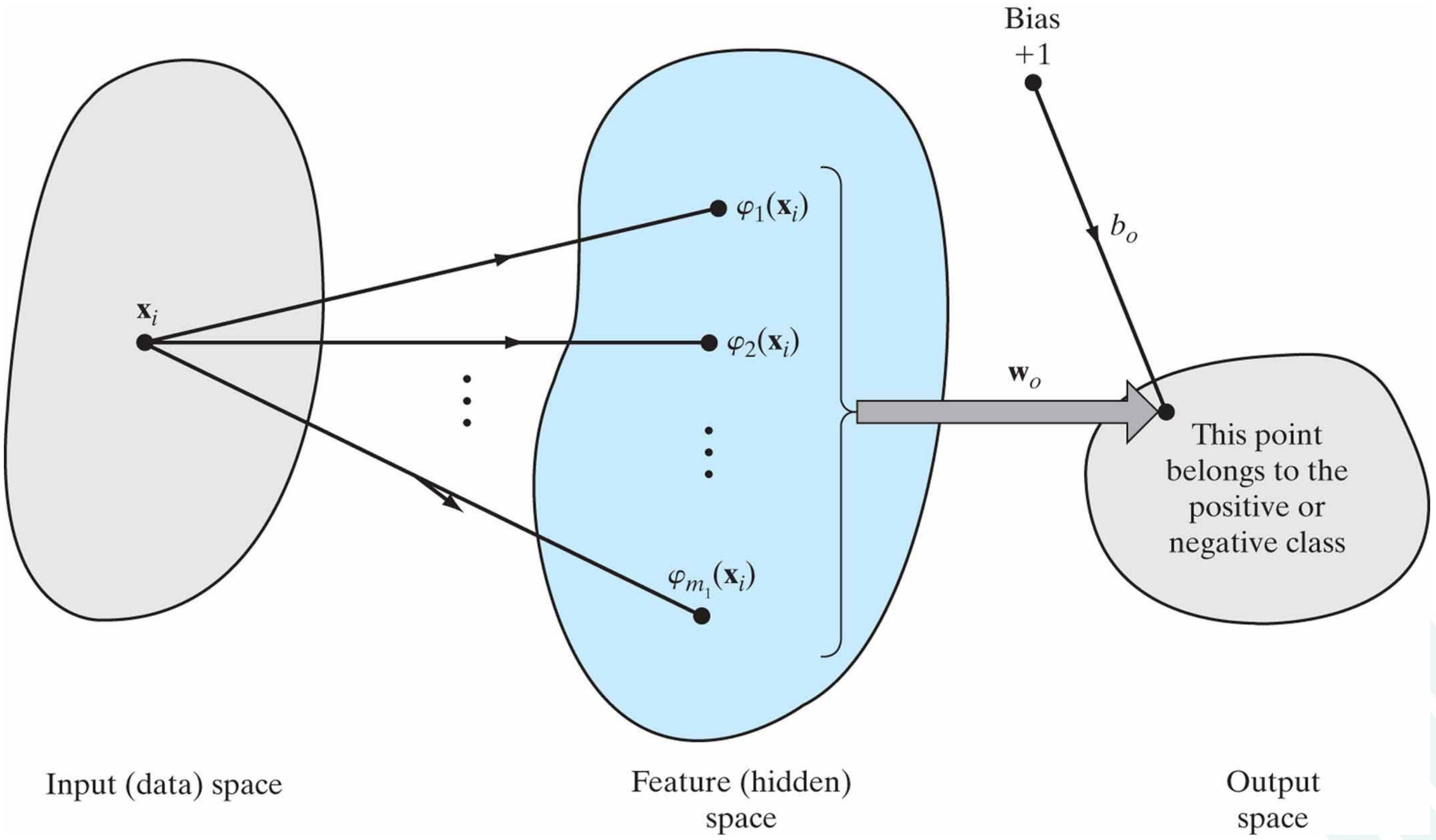
- Neither the slack variables nor their own Langarne multipliers appear in the dual problem!
- Separable vs Non Separable
  - $\alpha_i >= 0$
  - $0 <= \alpha_i <= C, \forall\ i = 1,2, \ldots.N$

Cover's Theorem:(Cover, 1965)

- "A complex pattern-classification problem cast in high-dimensional space nonlinearly is more likely to be linearly separable than in a low dimensional space"
  - Transformation is non-linear
  - Dimensionality of the feature space is high enough



Kernel SVM: Non-linear Mapping



Kernel SVM: Optimal Hyperplane in Feature Space

- Similar to the idea of the non separable patterns
  - Separating hyperplane is defined as a linear function of vectors drawn from the feature space rather than the original input space.
- Non-linear Transformation

$$\Phi(x) = [\Phi_0(x), \Phi_1(x), \ldots, \Phi_m(x)]$$

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i x_i \qquad \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \Phi(x_i)$$

$$\mathbf{w}^T \Phi(x) = 0 \qquad \sum_{i=1}^{N} \alpha_i y_i \Phi^T(x_i) \Phi(x) = 0$$

The Inner Product Kernel

- For a given set of training samples (in a lower-dimensional feature space) and a transformation into a higher-dimensional space, there exists a function (The Kernel Function) which can compute the dot product in the higher-dimensional space without explicitly transforming the vectors into the higher-dimensional space first.

$$K(\mathbf{x}, \mathbf{x}_i) = \Phi(\mathbf{x})^T \Phi(\mathbf{x_i})$$

$$= \sum_{j=0}^{m} \Phi_j(x)^T \Phi_j(x_i), \forall i = 1, 2, \ldots, N$$

$$K(\mathbf{x}, \mathbf{x}_i) = K(\mathbf{x_i}, \mathbf{x}), \forall i$$

The Kernel Trick

- Optimal Hyperplane

$$\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) = 0$$

- Specifying the kernel is *sufficient* for the pattern classification in the output space; we need never explicitly compute the weight vector w_o
- The optimal hyperplane consists of a finite number of terms that is equal to the number of support vector patterns.

Optimum Design of a SVM

- Given the training sample $\{(x_i, y_i\}^N_{i=1}$, find the Lagrange multipliers α_i that maximize the objective function
  - $Q(\alpha) = \boldsymbol{\Sigma}^N_{i=1}\alpha_i - 1/2\Sigma^N_{i=1}\Sigma^N_{j=1}\alpha_i\alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j})$ S.T.
    - $\Sigma^N_{i=1}\alpha_i y_i = 0$
    - $0 <= \alpha_i <= C, \forall\ i = 1,2, \ldots .N$

  where C is user-specified positive parameter
- $K(\mathbf{x_i}, \mathbf{x_j})$ can also be viewed as the ij-th element of a symmetric N-by-N matrix K:
  - $K = \{K(\mathbf{x_i}, \mathbf{x_j})\}^N_{(i, j) = 1}$
- Optimal weight $w_o = \Sigma^{Ns}_{i=1}\alpha_{o,i} y_i \varphi(\mathbf{x_i})$
  - First component of w_o represents the bias b_o
- Feature Space dimensionality is determined by N_s.

Inner-Product Kernels

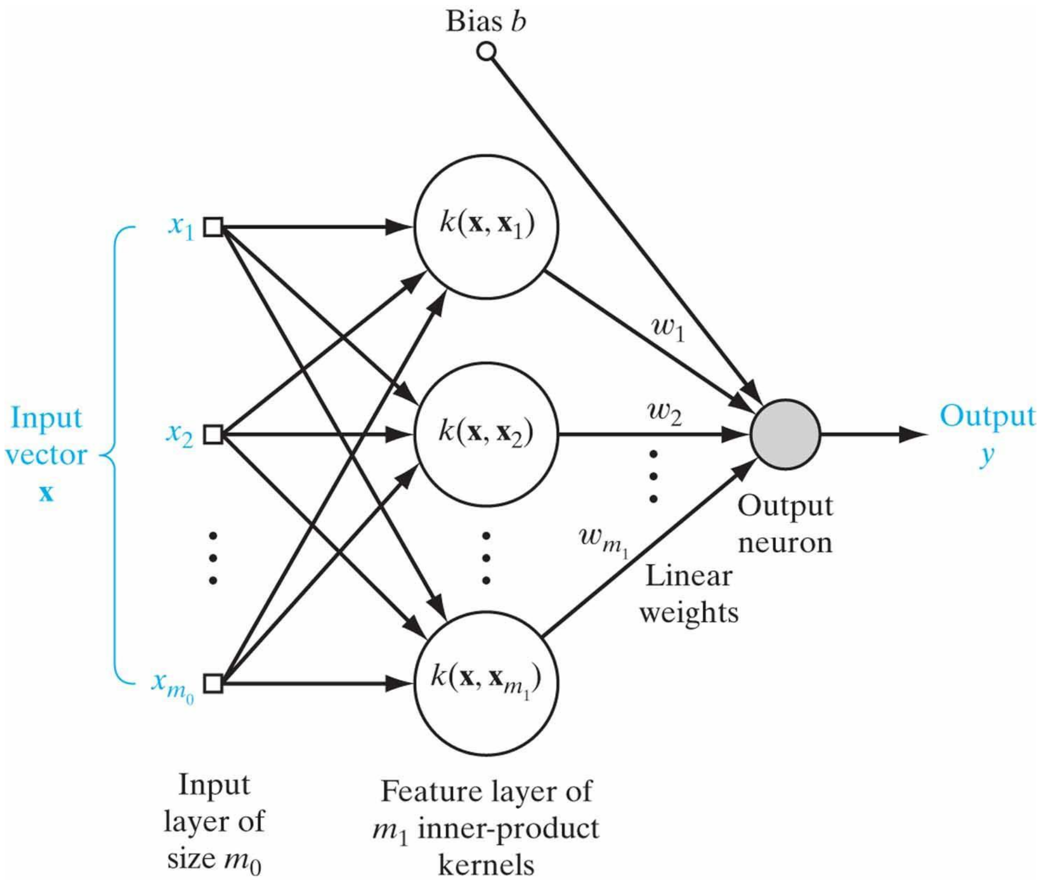
TABLE 6.1 Summary of Mercer Kernels

Type of support vector machine	Mercer kernel $k(\mathbf{x}, \mathbf{x}_i), i = 1, 2, \ldots, N$	Comments
Polynomial learning machine	$(\mathbf{x}^T\mathbf{x}_i + 1)^p$	Power p is specified *a priori* by the user
Radial-basis-function network	$\exp\left(-\frac{1}{2\sigma^2}\lVert\mathbf{x} - \mathbf{x}_i\rVert^2\right)$	The width σ^2, common to all the kernels, is specified *a priori* by the user
Two-layer perceptron	$\tanh(\beta_0\mathbf{x}^T\mathbf{x}_i + \beta_1)$	Mercer's theorem is satisfied only for some values of β_0 and β_1

- *Exercise: Apply the Lagrangian multipliers to get the dual*

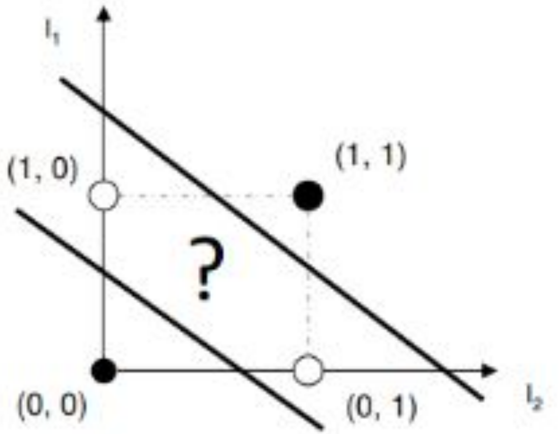
- *Exercise: Read Mercer Theorem*

Architecture of SVM



XOR Problem: Kernel Trick

TABLE 6.2 XOR Problem	
Input vector $\mathbf{x}$	Desired response d
$(-1, -1)$	-1
$(-1, +1)$	$+1$
$(+1, -1)$	$+1$
$(+1, +1)$	-1



- Given the following kernel:
  - $K(\mathbf{x_i}, \mathbf{x_j}) = (1+ \mathbf{x}^T\mathbf{x}_i)^2$

Derive the optimum **w** for a SVM using the above kernel

Example:XOR Problem

- Define a kernel:
  - $K(\mathbf{x_i}, \mathbf{x_j}) = (1 + \mathbf{x}^T\mathbf{x}_i)^2$
- Let $\mathbf{x} = [x_1, x_2]^T$, $\mathbf{x_i} = [x_{i1}, x_{i2}]^T$
- $K(\mathbf{x_i}, \mathbf{x_j})$ = ?
  - $1 + x^2_{1}x^2_{i1} + 2x_1x_2x_{i1}x_{i2} + x^2_{2}x^2_{i2} + 2x_1x_{i1} + 2x_2x_{i2}$
- $\varphi(\mathbf{x}) = [1, x^2_1, \sqrt{2}x_1x_2, x^2_2, \sqrt{2}x_1, \sqrt{2}x_2]$
- $\varphi(\mathbf{x_i}) = [1, x^2_{i1}, \sqrt{2}x_{i1}x_{i2}, x^2_{i2}, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}]$
- Gram

$$\mathbf{K} = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

Example:XOR Problem

- Dual form Objective Function
  - $Q(\alpha) = \boldsymbol{\Sigma}^{N}_{i=1}\alpha_i - 1/2\Sigma^{N}_{i=1}\Sigma^{N}_{j=1}\alpha_i\alpha_j y_i y_i K(\mathbf{x_i x_j})$,

$$Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4 + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2)$$

- *Conditions of optimality*
  - $\partial Q(\alpha)/\partial\alpha = 0$

$$\begin{aligned} 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 &= 1 \\ -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 &= 1 \\ -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 &= 1 \\ \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 &= 1 \end{aligned}$$

Example:XOR Problem

- Dual form Objective Function Solution

$$\alpha_{o,1} = \alpha_{o,2} = \alpha_{o,3} = \alpha_{o,4} = \frac{1}{8}$$

- Optimum Weight
  - $\mathbf{w}_o = \Sigma^{Ns}_{i=1} \alpha_{o,i} d_i \varphi(\mathbf{x}_i)$

$$\mathbf{w}_o = \frac{1}{8}\left[-\varphi(\mathbf{x}_1) + \varphi(\mathbf{x}_2) + \varphi(\mathbf{x}_3) - \varphi(\mathbf{x}_4)\right]$$

Example:XOR Problem

- Optimum Weight

$$\frac{1}{8}\left[-\begin{bmatrix}1\\1\\\sqrt{2}\\1\\-\sqrt{2}\\-\sqrt{2}\end{bmatrix}+\begin{bmatrix}1\\1\\-\sqrt{2}\\1\\-\sqrt{2}\\\sqrt{2}\end{bmatrix}+\begin{bmatrix}1\\1\\-\sqrt{2}\\1\\\sqrt{2}\\-\sqrt{2}\end{bmatrix}-\begin{bmatrix}1\\1\\\sqrt{2}\\1\\\sqrt{2}\\\sqrt{2}\end{bmatrix}\right]$$

$$\begin{bmatrix}0\\0\\-1/\sqrt{2}\\0\\0\\0\end{bmatrix}$$

Example:XOR Problem

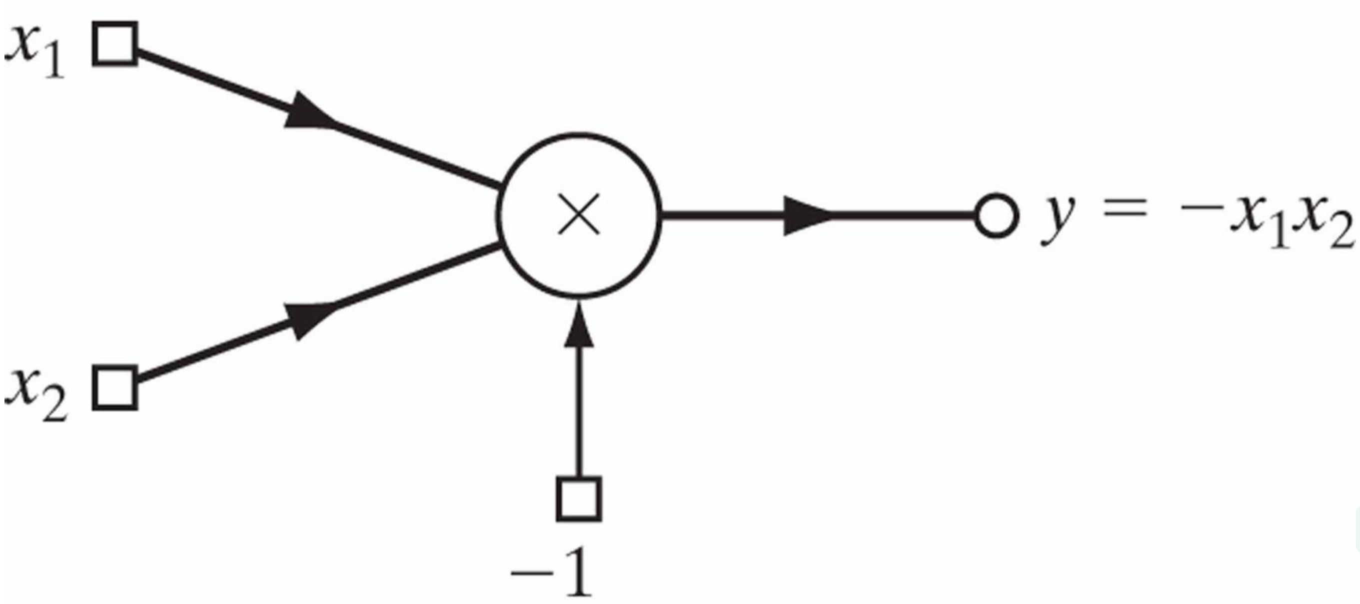
- Optimum Hyperplane
  - $\mathbf{w}_o^T\varphi(\mathbf{x}) = 0$ or

$$\left[0, 0, \frac{-1}{\sqrt{2}}, 0, 0, 0\right] \begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0$$

$$\sum_{i=1}^{N} \alpha_i d_i K(\mathbf{x}, \mathbf{x}_i) = 0$$

$$-x_1x_2 = 0$$

Example:XOR Problem



(a)

References

1. https://www.youtube.com/watch?v=SRVswRH5Q7E
2. http://cs229.stanford.edu/notes/cs229-notes3.pdf
3. Chapter 6, Neural Networks: A Comprehensive Foundation (2nd Edition) 2nd Edition by Simon Haykin
4. Hastie, T., R. Tibshirani, and J. Friedman. The Elements of Statistical Learning, second edition. New York: Springer, 2008.
5. Christianini, N., and J. Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge, UK: Cambridge University Press, 2000.
6. Fan, R.-E., P.-H. Chen, and C.-J. Lin. “Working set selection using second order information for training support vector machines.” Journal of Machine Learning Research, Vol 6, 2005, pp. 1889–1918.