Neural Networks: Support Vector Machines (SVM)

Sven Lončarić
Doc. dr. sc. Marko Subašić

Faculty of Electrical Engineering and Computing
Univerity of Zagreb

http://www.fer.hr/predmet/neumre_b

Overview

- Classification of linearly separable classes
- Separation margin
- Support vectors
- Classification of linearly non separable classes
- Nonlinear mapping to feature space

Introduction

- Similarities to MLP and RBN
  - Feed forward network
  - Applications in <u>classification</u> and nonlinear regression
  - Inherently good generalization properties
- Differences
  - Training of SVMs is not iterative on selected training samples
  - SVM minimizes number of training samples within the separation margin – MLP minimizes mean square error
- General algorithm for training of feed-forward networks
- Feed-forward network with one hidden layer

What is the goal?

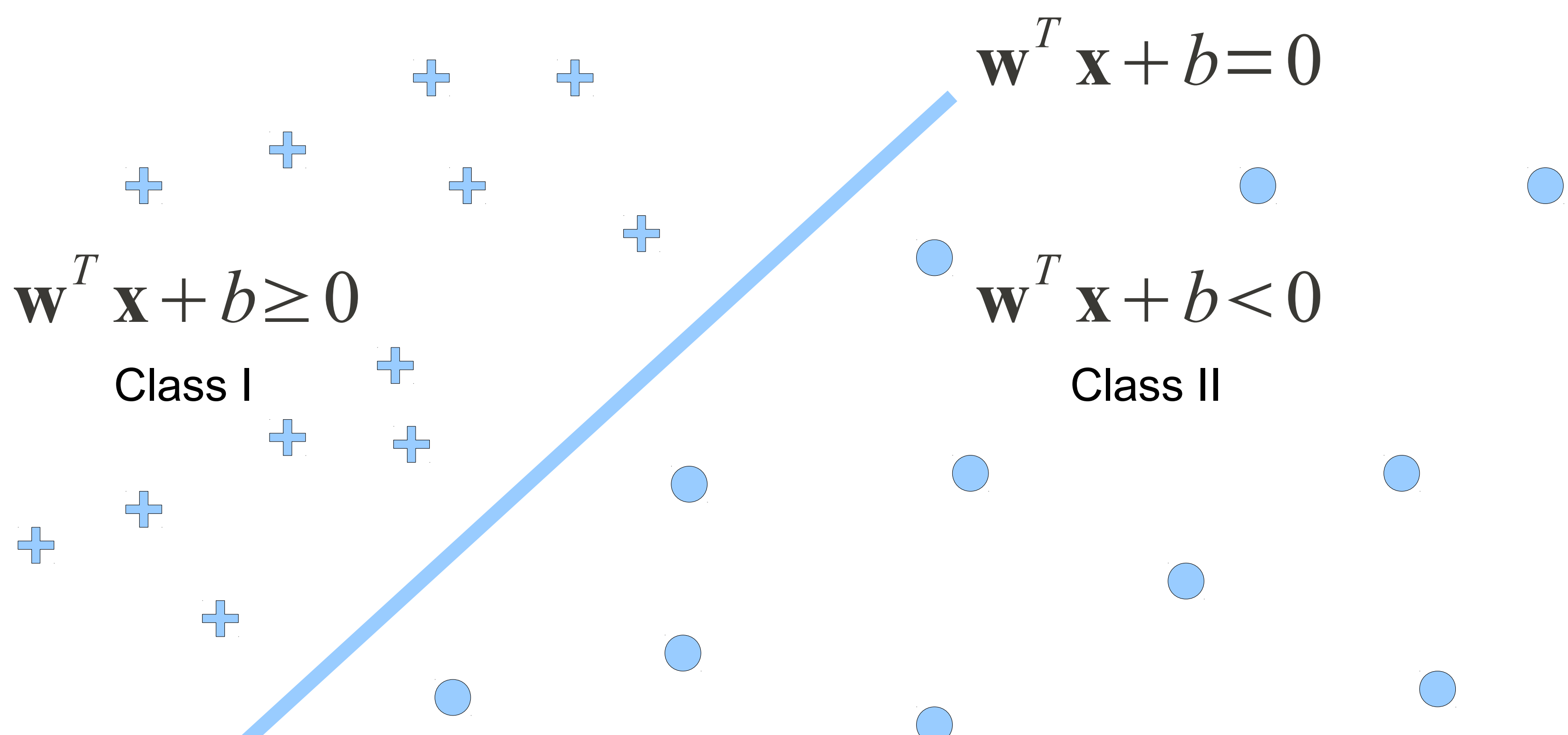
- Classification of samples in two classes
- To find a separation plane for two classes that maximizes the separation, or margin
- Margin is defined using the “key” training samples – support vectors

Separation of two linearly separable classes

- A simple problem
- Sometimes more complex problems can be reduced to such simple problems...

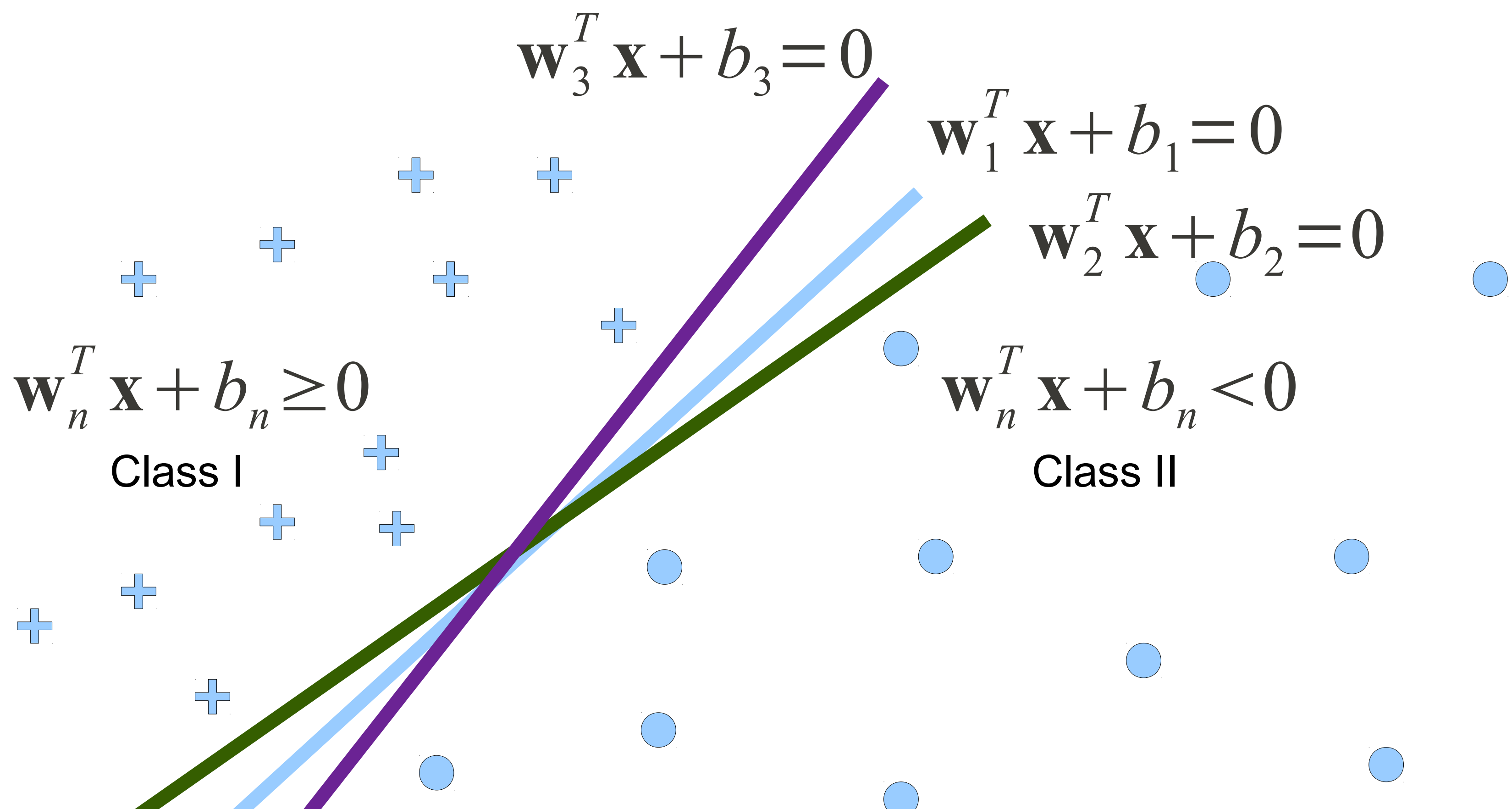
Separation of two linearly separable classes

- Hyperplane equation
  - **w** – weight vector, **x** – input vector, *b* - bias



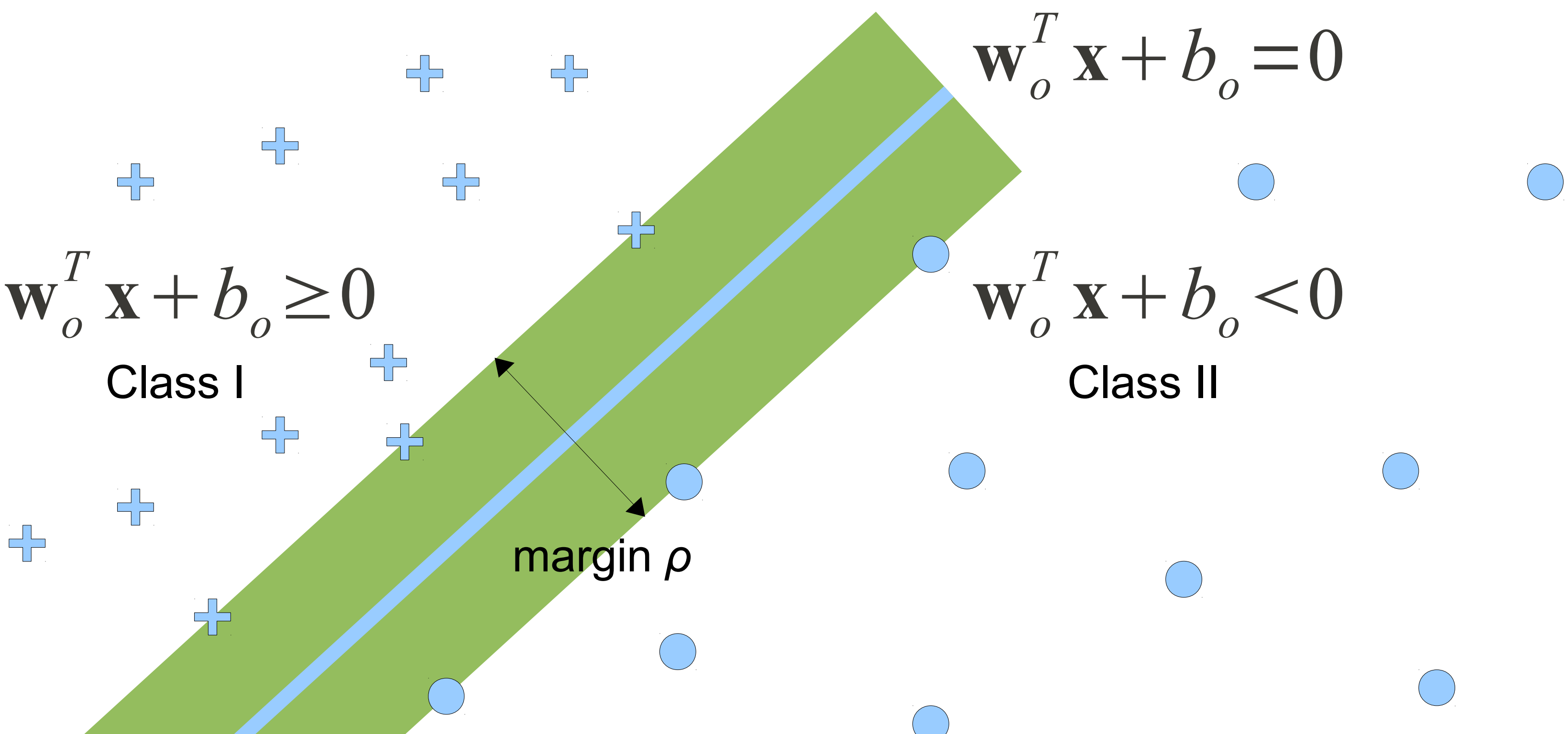
Separation of two linearly separable classes

- There are many possible separation hyperplanes
- Which one is optimal?



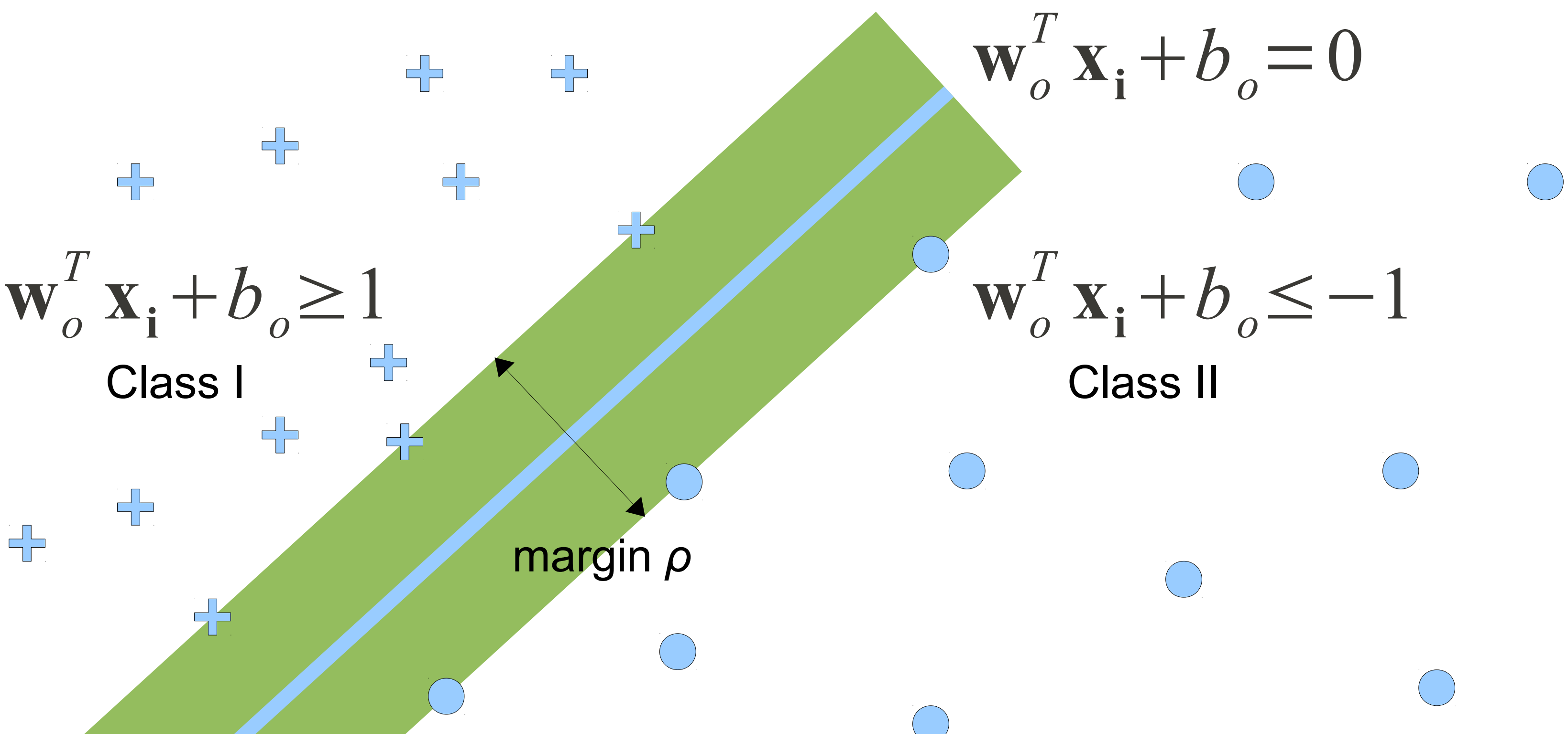
Separation Margin

- Distance from hyperplane to the closest sample $\mathbf{x}_i$ of any class
- SVM seeks the optimal separation hyperplane ($\mathbf{w}_o$ i b_o) that maximizes separation margin ρ



Support vectors

- Support vectors $\mathbf{x}_i$ such that $\mathbf{w}_o^T \mathbf{x}_i + b_o = \pm 1$
  - Closest to the separation hyperplane
  - Most difficult to classify
  - Most relevant for estimation of $\mathbf{w}_o$ i b_o



Separation hyperplane

- Separation hyperplane

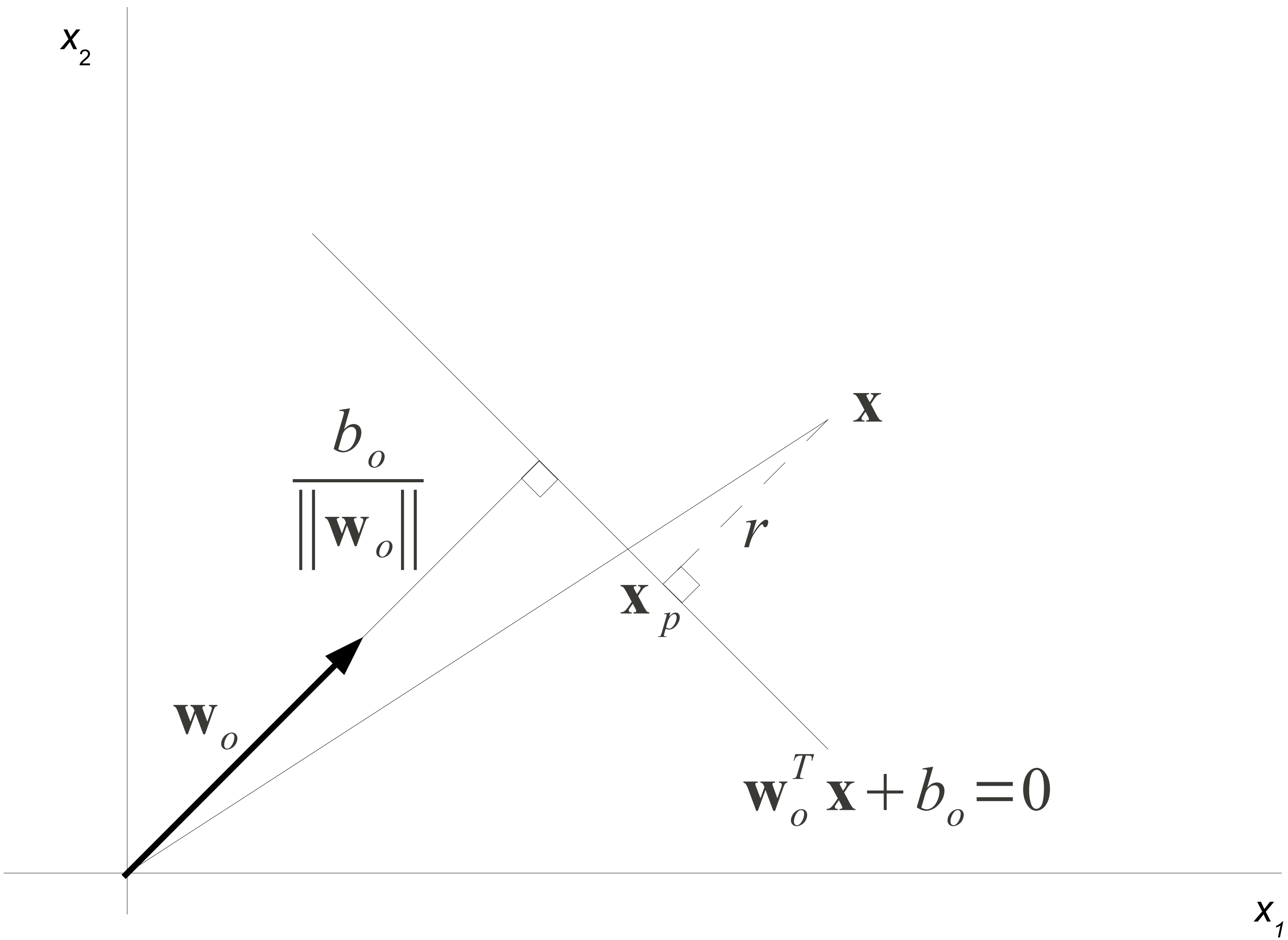
$$\mathbf{w}_o^T \mathbf{x} + b_o = 0$$

- Discrimination function

$$g(\mathbf{x}) = \mathbf{w}_o^T \mathbf{x} + b_o$$

- Determines the class of the sample **x** on the basis of the sign of the corresponding value **g**(**x**)

Separation hyperplane location



Distance to the separation hyperplane

- Sample **x** position expressed trough the projection on the separation hyperplane $\mathbf{x}_p$ and distance r
- r determines the amplitude of vector whose direction is determined by $\mathbf{w}_o$ – perpendicular to the projection hyperplane (normal)

$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}_o}{\left\|\mathbf{w}_o\right\|}$$

- Sign of r depends on the side of the hyperplane that sample is located in

Distance to the separation hyperplane

- Lets express the distance using the discriminating function

$$g(\mathbf{x}) = \mathbf{w}_o^T \mathbf{x} + b_o = r \|\mathbf{w}_o\|$$

$$g(\mathbf{x}_p) = 0$$

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}_o\|}$$

Separation hyperplane distance to the origin

- Separation hyperplane distance to the origin

$$\frac{b_o}{\left\|\mathbf{w}_o\right\|}$$

- Scaling of $\mathbf{w}_o$ and b_o together does not change the separation hyperplane
- Orientation of vector $\mathbf{w}_o$ stays the same

"Choosing" the support vectors

- Selecting the support vectors $\mathbf{x}^{(s)}$ such that

$$g\left(\mathbf{x}^{(s)}\right)=\mathbf{w}_o^T\mathbf{x}^{(s)}+b_o=\pm 1 \qquad \text{for} \qquad d=\pm 1$$

- Their distance to the separation hyperplane

$$r=\frac{\mathbf{g}\left(\mathbf{x}^{(s)}\right)}{\left\|\mathbf{w}_o\right\|}=\frac{\pm 1}{\left\|\mathbf{w}_o\right\|}$$

- Width of the separation margin ρ is equal to

$$\rho=2r=\frac{2}{\left\|\mathbf{w}_o\right\|}$$

Maximizing the separation margin

- Maximizing the margin is equivalent to minimizing of the Euclidean norm of $\mathbf{w}_o$

$$\rho = 2r = \frac{2}{\|\mathbf{w}_o\|}$$

- This results in the optimal separation hyperplane that maximizes the margin

Optimization procedure

- Requirement for all training samples

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

- The goal is to find the minimum of the optimization function (weight vector norm)

$$\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

Optimization procedure

- Solution using Lagrangian multipliers (α_i)

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i [d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1]$$

$$\alpha_i \geq 0$$

$$\min_{\mathbf{w}, b} \max_{\alpha_i} J(\mathbf{w}, b, \alpha)$$

- The solution is in the saddle point

Optimization procedure

- Partial derivations of **w** and *b* equal to 0

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i [d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1]$$

$$\frac{\partial J(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{0} \qquad \frac{\partial J(\mathbf{w}, b, \alpha)}{b} = 0$$

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x}_i \qquad \sum_{i=1}^{N} \alpha_i d_i = 0$$

Optimization procedure

- Determination of Lagrangian multipliers (α_i)

$$\min_{\mathbf{w},b} \max_{\alpha_i} J(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i [d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1]$$

$$\max_{\alpha_i} J(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{N} \alpha_i [d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1]$$

$$\alpha_i \geq 0 \qquad d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0$$

- The maximum is reached when all components are equal to zero
  - *α_i will not be zero only when*

$$d_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 = 0$$

Optimization procedure

- Equation describes support vectors

$$d_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 = 0$$

- Lagrangian multipliers α_i that are not zero "automatically select" support vectors

Optimization procedure

- To calculate the Lagrangian multipliers we use the dual problem

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

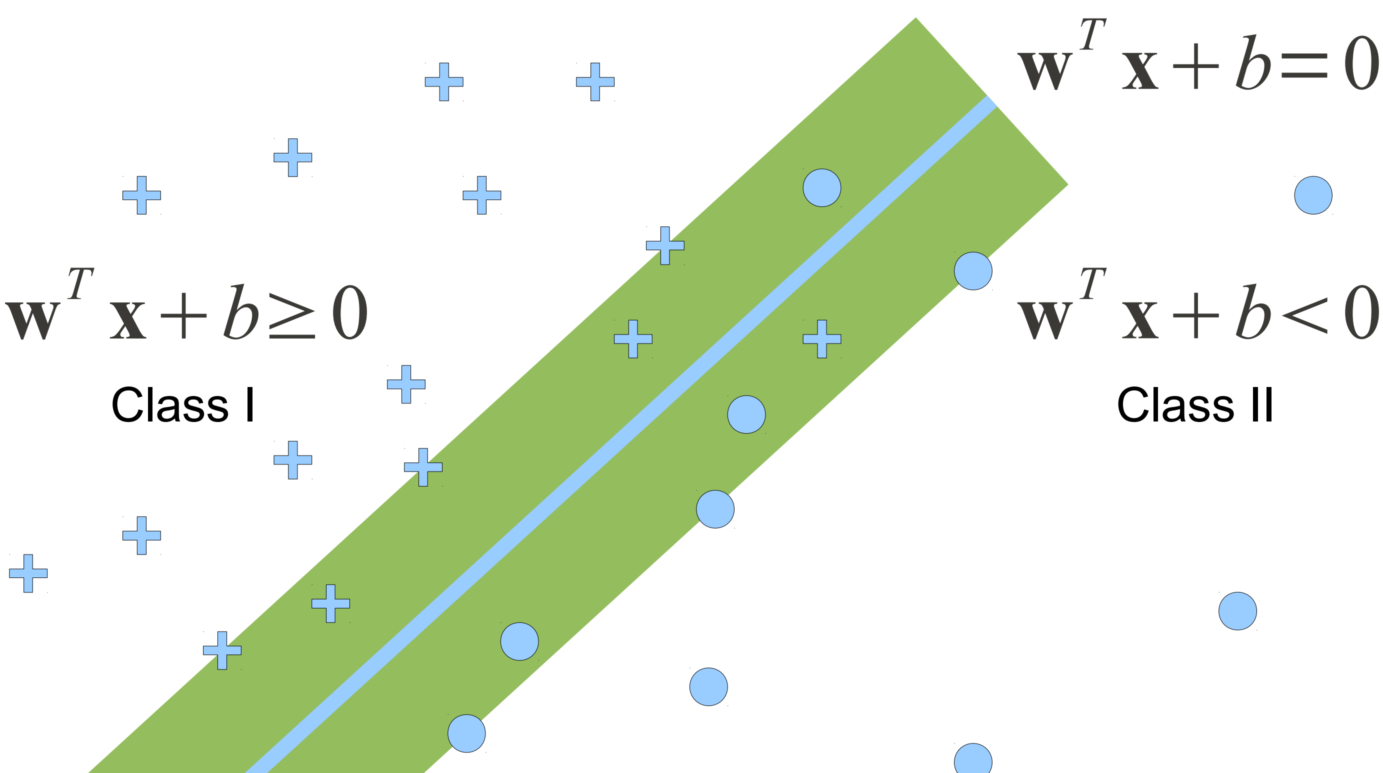
$$\sum_{i=1}^{N} \alpha_i d_i = 0$$

$$\alpha_i \geq 0$$

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_{o,i} d_i \mathbf{x}_i \qquad b_o = 1 - \mathbf{w}_o^T \mathbf{x}^{(s)}, \quad d^{(s)} = 1$$

Separation of linearly non separable classes

- Final procedure is practically identical to the case of linearly separable classes



Separation of linearly non separable classes

$$d_i(\mathbf{w}^T \mathbf{x_i} + b) \geq 1 - \xi$$

- The goal is to reduce the average classification error

$$\Phi(\xi) = \sum_{i=1}^{N} I(\xi_i - 1)$$

$$I(\xi) = \begin{cases} 0 & if\ \xi \leq 0 \\ 1 & if\ \xi > 0 \end{cases}$$

Optimization procedure

- We can simplify the problem by approximation

$$\Phi(\xi) = \sum_{i=1}^{N} \xi_i$$

- And expand it with the minimization of the Euclidean norm of **w**

$$\Phi(\xi, \mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{N} \xi_i$$

Optimization procedure

- The solution is again calculated using the dual problem

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\sum_{i=1}^{N} \alpha_i d_i = 0$$

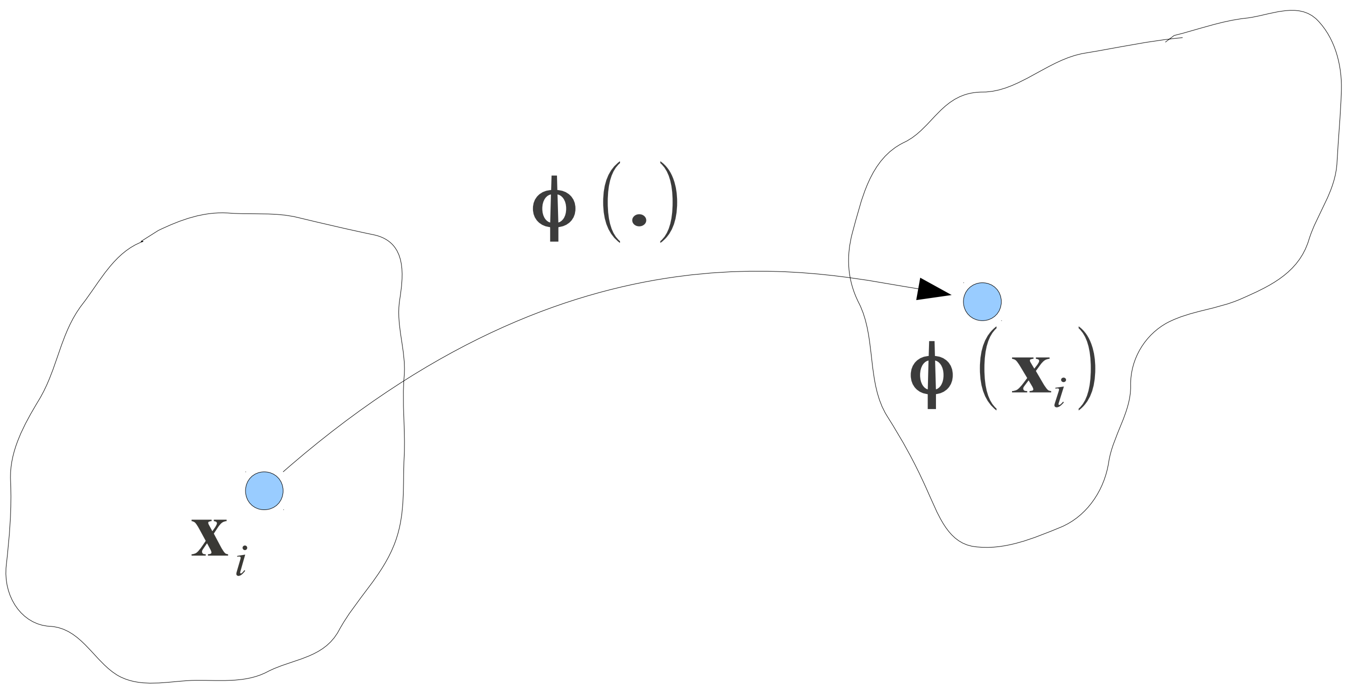
$$0 \le \alpha_i \le C$$

$$\mathbf{w_o} = \sum_{i=1}^{N} \alpha_{o,i} d_i \mathbf{x}_i \qquad b_o = \frac{1}{N_s} \sum_{i=1}^{N_s} d_i \left(1 - \mathbf{w}_i^T \mathbf{x}_i^{(s)}\right)$$

Classification using SVM

- If samples are not linearly separable, it would be nice if we could make them to be
- Then we could just apply previous classification algorithm
- By transitioning to the higher dimensional spaces we increase the chance of obtaining the linear separability (Cover's theorem)
- Basic idea consists of:
    - Nonlinear mapping of input space to the higher dimensional feature space
    - Construction of optimal separation hyperplane in the new feature space

Nonlinear mapping



$$\boldsymbol{\phi}(\mathbf{x}) = [\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \ldots \phi_m(\mathbf{x})]$$

Linear separation in the feature space

- Optimal separation hyperplane is constructed in the higher dimensional feature space

$$\sum_{j=1}^{m} w_j \varphi_j(\mathbf{x}) + b = 0$$

- $\varphi_j(\mathbf{x})$ are m transformation functions
- m is the number of dimensions in the new feature space

Linear separation in the feature space

- The bias b can be included in the weight vector **w** as the first element

$$\sum_{j=1}^{m} w_j \varphi_j(\mathbf{x}) = 0$$

$$\varphi_0(\mathbf{x}) = 1$$

$$w_0 = b_o$$

Separation hyperplane

- Separation hyperplane

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) = 0$$

- The missing **w** can be expressed as

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \boldsymbol{\varphi}(\mathbf{x}_i)$$

- By combining the two above expressions we get

$$\sum_{i=1}^{N} \alpha_i d_i \boldsymbol{\varphi}^T(\mathbf{x}_i) \boldsymbol{\varphi}(\mathbf{x}) = 0$$

Inner product kernel function

- Inner product of two vectors in the new feature space

$$\boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{x})$$

- We introduce the new kernel function *K*

$$K(\mathbf{x}_i, \mathbf{x}) = \boldsymbol{\varphi}^T(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{x})$$

- We obtain the new separation hyperplane equation

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

Mercer's theorem

- Let K(**x**,**x**') be a symmetric kernel function defined on the closed intervals for **x** i **x**'
- Such kernel can be expanded in the series:

$$K(\mathbf{x}_i, \mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{x}')$$

- If this is true than the kernel K is the inner product kernel
  - Number of dimensions can theoretically be infinite

Optimization

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\sum_{i=1}^{N} \alpha_i d_i = 0$$

$$0 \le \alpha_i \le C$$

$$\mathbf{w_o} = \sum_{i=1}^{N} \alpha_{o,i} d_i \boldsymbol{\varphi}(\mathbf{x}_i)$$

Examples of inner product kernels

- There is some freedom in choosing the kernel but it must satisfy Mercer's theorem
- Typical examples include:
  - Polynomial kernel
  - Radial-basis function
  - Two-layer perceptron
- Feature space dimensionality depends on the number of support vectors

Polynomial kernel

$$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + 1)^p$$

- Parameter *p* is set a priori by the user
- $\mathbf{x}_i$ are support vectors

Radial-basis function

$$K(\mathbf{x}, \mathbf{x}_i) = e^{\left(-\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{x}_i\|^2\right)}$$

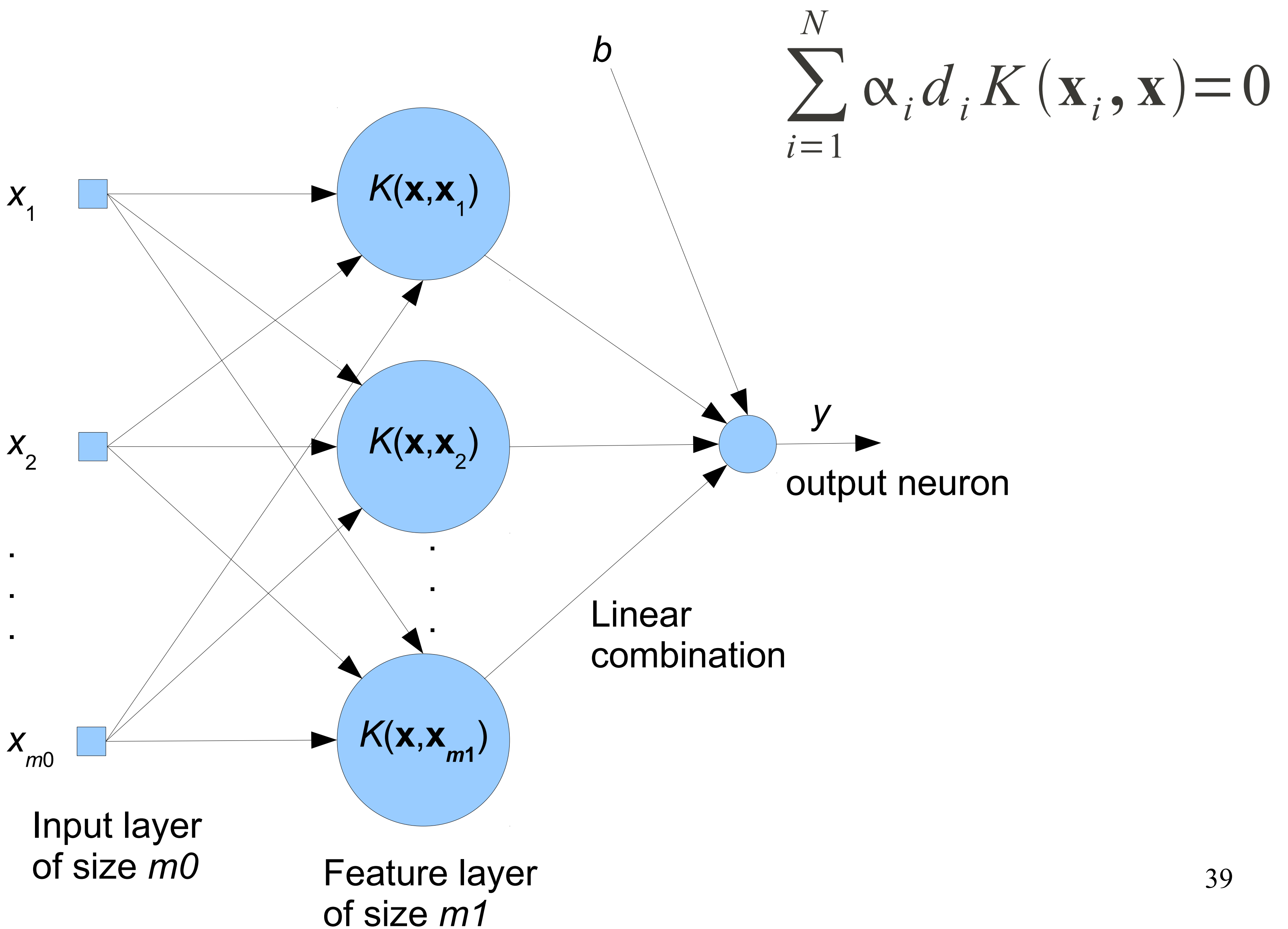
- The width σ^2 for all kernels is set a priori by the user
- The number RBFs and their centers is determined by the choice of support vectors

Two-layer perceptron

$$K(\mathbf{x}, \mathbf{x}_i) = \tanh(\beta_0 \mathbf{x}^T \mathbf{x}_i + \beta_1)$$

- Mercer's theorem is satisfied only for some values of β_0 and β_1

SVM Architecture



Example: XOR problem

XOR problem	
Input **x**	Desired response d_i
(-1,-1)	- 1
(-1,+1)	+1
(+1,-1)	+1
(+1,+1)	-1

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

$$\mathbf{x}=[x_{1,}x_2]^T$$

$$K(\mathbf{x},\mathbf{x}_i)=1+x_1^2x_{i1}^2+2x_1x_2x_{i1}x_{i2}+x_2^2x_{i2}^2+2x_1x_{i1}+2x_2x_{i2}$$

$$\boldsymbol{\varphi}(\mathbf{x})=\left[1,x_1^2,\sqrt{2}x_1x_2,x_2^2,\sqrt{2}x_1,\sqrt{2}x_2\right]$$

Example: XOR problem

$$Q(\alpha) = \sum_{i=1}^{4} \alpha_i - \frac{1}{2} \sum_{i=1}^{4} \sum_{j=1}^{4} \alpha_i \alpha_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j)$$

- Optimization procedure provides following Lagrange multipliers

$$\alpha_{o1} = \alpha_{o2} = \alpha_{o3} = \alpha_{o4} = \frac{1}{8}$$

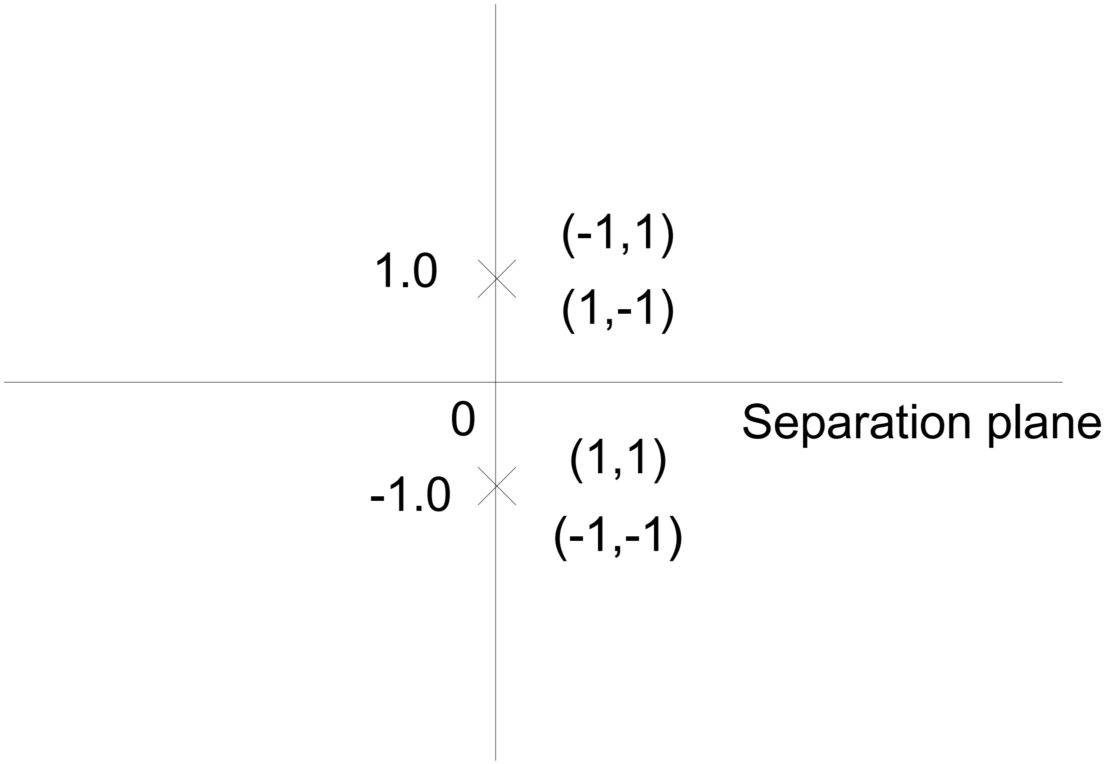
$$\mathbf{w_o} = \sum_{i=1}^{N} \alpha_{o,i} d_i \boldsymbol{\varphi}(\mathbf{x}_i) = \begin{bmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Example: XOR problem

$$0 = \mathbf{w}_{\mathbf{o}}^{\mathbf{T}} \boldsymbol{\varphi}(\mathbf{x}) = \left[0, 0, -1/\sqrt{2}, 0, 0, 0\right] \begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \\ \sqrt{2}\, x_1 \\ \sqrt{2}\, x_2 \end{bmatrix} = -x_1 x_2$$

Example: XOR problem

$$-x_1 x_2 = 0$$



How does it all work

1. Prepare training samples
2. Select the inner product kernel function *K* that satisfies Mercer's theorem
3. Calculate optimal α_i

$$Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j K(\mathbf{x}_i,\mathbf{x}_j),\quad \sum_{i=1}^{N}\alpha_i d_i=0,\quad 0\le\alpha_i\le C$$

   (this determines the support vectors)

4. Classification by the following discrimination function

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

SVM: pros and cons

- Advantages
  - Optimal solution regarding the goal function is always obtainable
  - Efficient optimization implementation solutions are possible
  - We achieve separation in the higher dimensional feature space without ever visiting the space
- Disadvantages
  - Execution speed – there is no direct control over the number of support vectors
  - It is not possible to adjust the algorithm based on the a priori knowledge of the problem (this can be viewed as an advantage)
    - Solution: construction of "artificial" training samples based on the a priori knowledge
    - Solution: introduction of new constraints to the goal function

Overview

- Classification of linearly separable classes
- Separation margin
- Support vectors
- Classification of linearly non separable classes
- Nonlinear mapping to feature space

Tasks

1. Show that the margin is equal to $2/\|\mathbf{w}_o\|$ if the separation hyperplane $\mathbf{w}_o^T\mathbf{x}+b_o=0$ satisfies the additional condition $\min_{i=1,2,\ldots,N} |\mathbf{w}_o^T\mathbf{x}+b_o|=1$

2. For polynomial kernel in XOR example, determine minimal value for the positive power *p* that still enables the solution to the problem.