Support vector machines

Johan Suykens

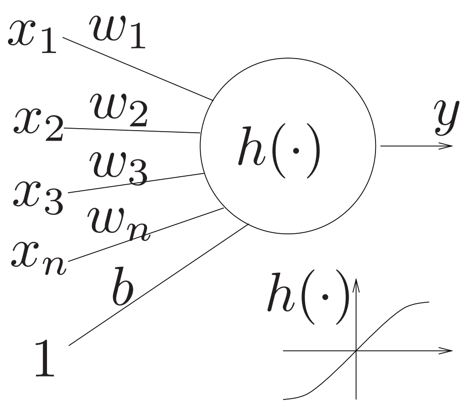
K.U. Leuven, ESAT-STADIUS
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium
Email: johan.suykens@esat.kuleuven.be
http://www.esat.kuleuven.be/stadius

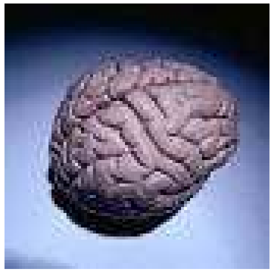
Lecture 9

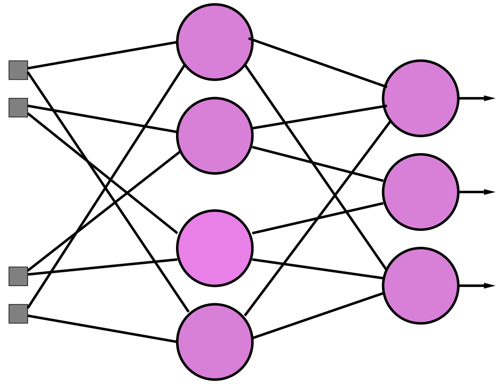
Overview

- Disadvantages of classical neural networks
- Linear support vector machine
- Nonlinear support vector machine, kernel trick
- Primal and dual problem
- Support vector regression

Classical MLPs







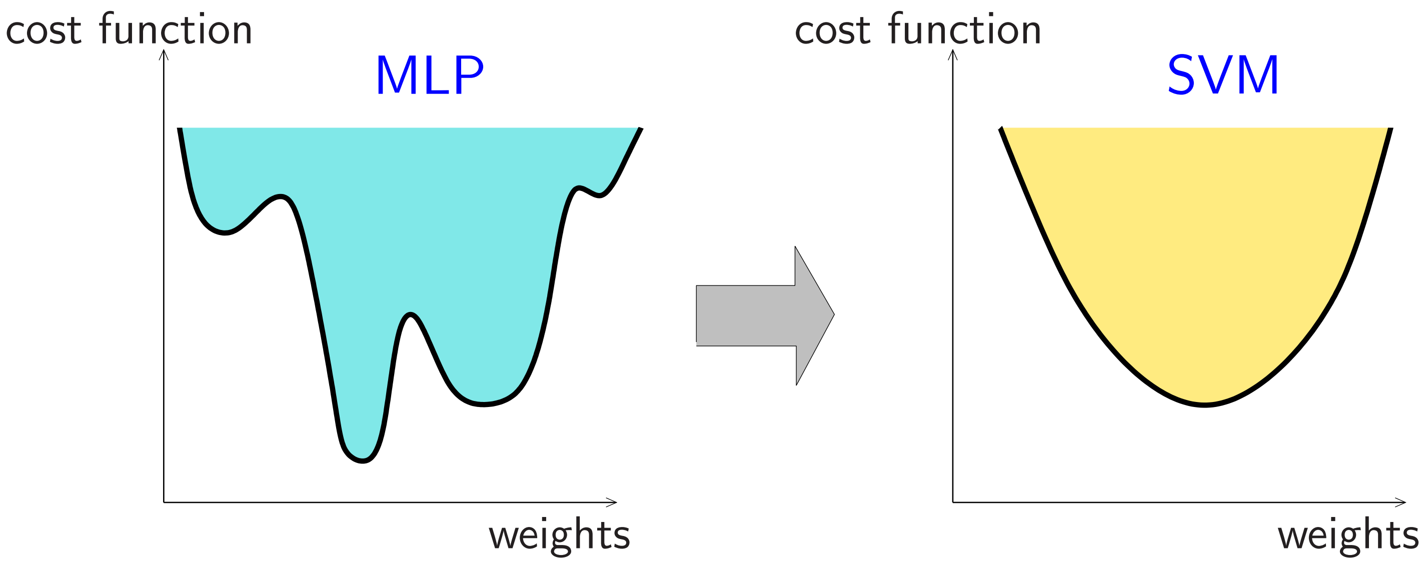
Multilayer Perceptron (MLP) properties:

- Universal approximation of continuous nonlinear functions
- Learning from input-output patterns: off-line/on-line
- Parallel network architecture, multiple inputs and outputs

$\boxed{+}$ Flexible and widely applicable:
Feedforward/recurrent networks, supervised/unsupervised learning

$\boxed{-}$ Many local minima, trial and error for determining number of neurons

Support Vector Machines

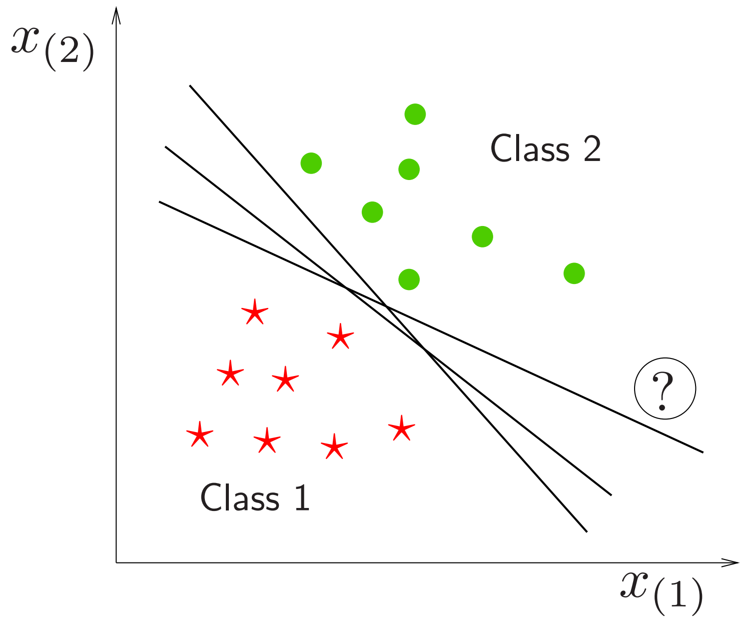


- Nonlinear classification and function estimation by **convex optimization** with a unique solution and primal-dual interpretations.
- **Number of neurons** automatically follows from a convex program.
- Learning and generalization in **high dimensional** input spaces (coping with the curse of dimensionality).
- Use of **kernels** (e.g. linear, polynomial, RBF, MLP, splines, kernels from graphical models, ...), application-specific kernels (e.g. bioinformatics)

Linear classifier

Training set $\{(x_i, y_i)\}_{i=1}^N$:
input data $x_i \in \mathbb{R}^d$
class labels $y_i \in \{-1, +1\}$

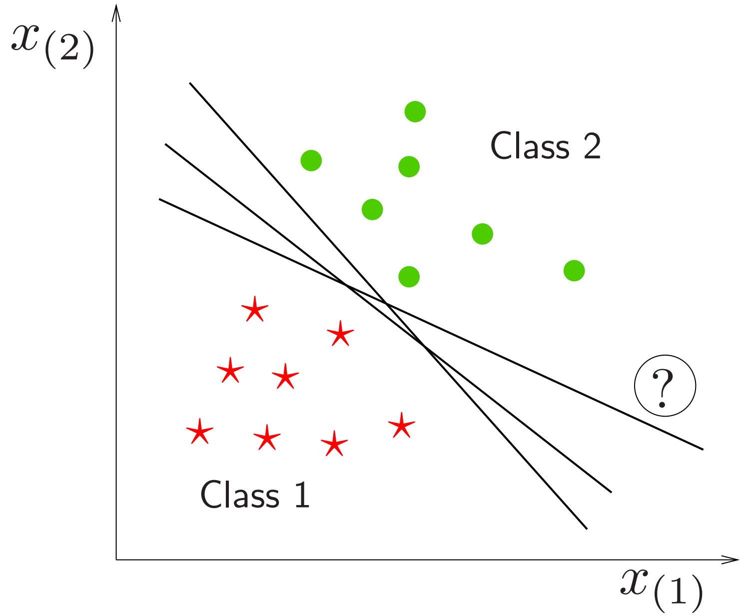
Classifier: $\hat{y} = \text{sign}[w^T x + b]$



Linear classifier

Training set $\{(x_i, y_i)\}_{i=1}^{N}$:
input data $x_i \in \mathbb{R}^d$
class labels $y_i \in \{-1, +1\}$

Classifier: $\hat{y} = \text{sign}[w^T x + b]$



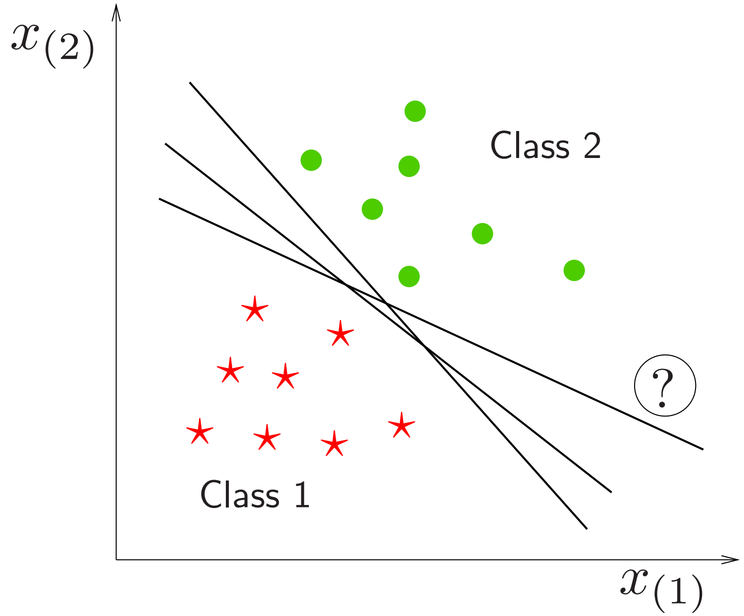
Requirement that all training data are correctly classified:

$$\begin{array}{ll} w^T x_i + b \geq +1, & \text{if} \;\; y_i = +1 \\ w^T x_i + b \leq -1, & \text{if} \;\; y_i = -1 \end{array}$$

Linear classifier

Training set $\{(x_i, y_i)\}_{i=1}^N$:
input data $x_i \in \mathbb{R}^d$
class labels $y_i \in \{-1, +1\}$

Classifier: $\hat{y} = \text{sign}[w^T x + b]$

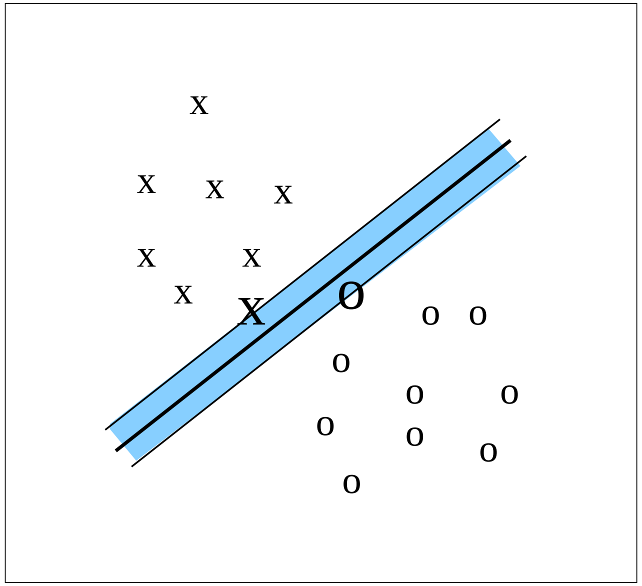


Requirement that all training data are correctly classified:

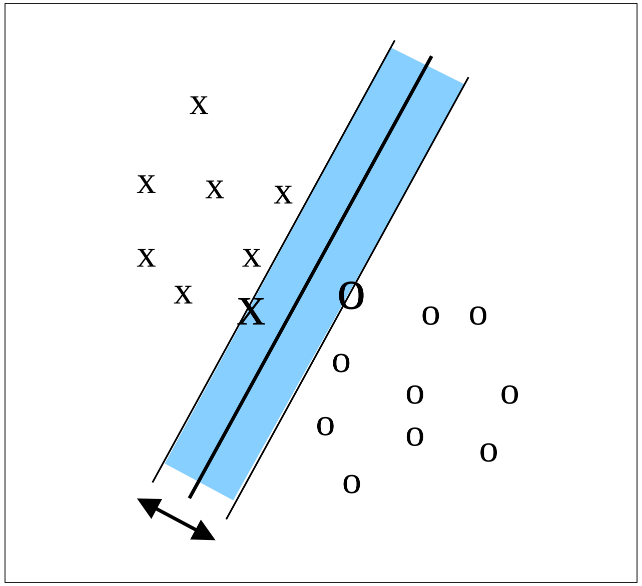
$$\begin{aligned} w^T x_i + b &\geq +1, \quad \text{if} \quad y_i = +1 \\ w^T x_i + b &\leq -1, \quad \text{if} \quad y_i = -1 \end{aligned}$$

$$\Leftrightarrow \quad y_i [w^T x_i + b] \geq 1, \quad \forall i$$

Maximize the margin



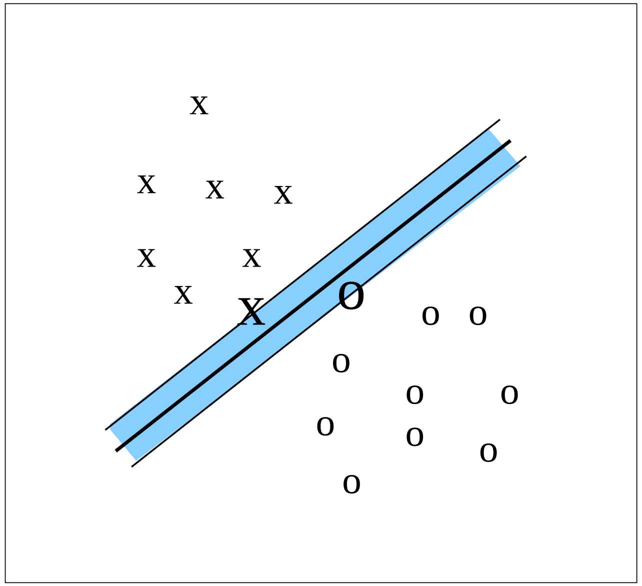
$\longrightarrow$



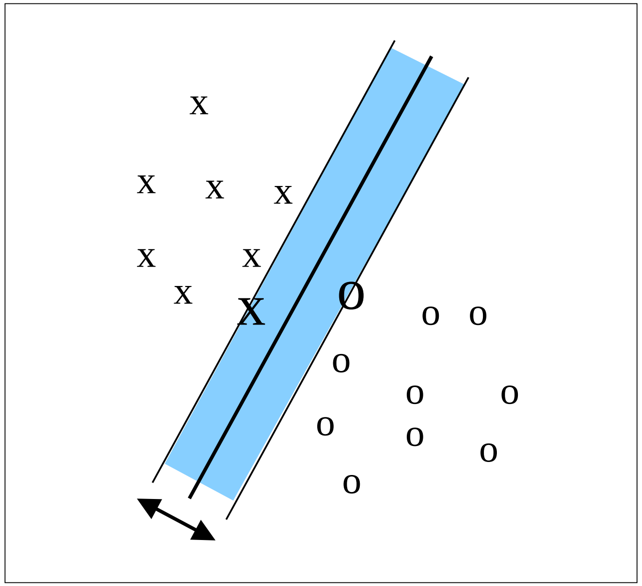
Margin $= \frac{2}{\|w\|}$

$$\begin{array}{ll} \min\limits_{w,b} & \frac{1}{2} w^T w \\ \text{subject to} & y_i [w^T x_i + b] \geq 1 \qquad , \; i = 1, ..., N \end{array}$$

Maximize the margin



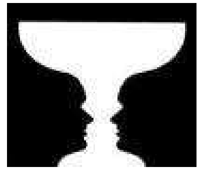
$\longrightarrow$

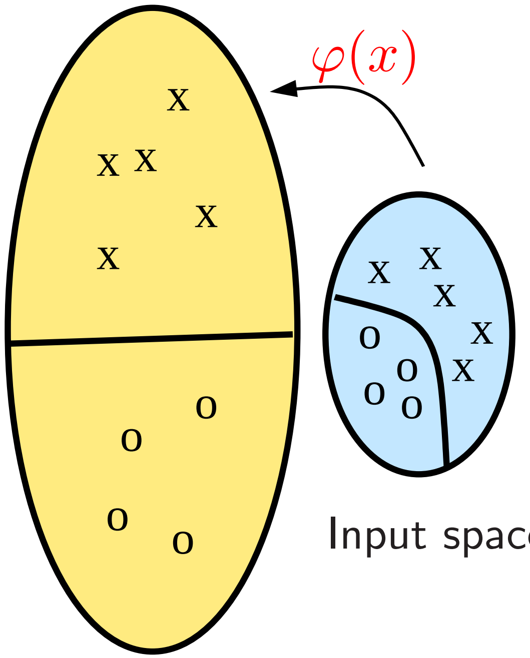


Margin $= \frac{2}{\|w\|}$

$$\begin{array}{ll} \min\limits_{w,b,\xi_i} & \frac{1}{2}w^T w + c \sum_{i=1}^{N} \xi_i \\ \text{subject to} & y_i[w^T x_i + b] \geq 1 - \xi_i, \;\; i = 1, ..., N \\ & \xi_i \geq 0, \;\; i = 1, ..., N \end{array}$$

SVMs: living in two worlds ...



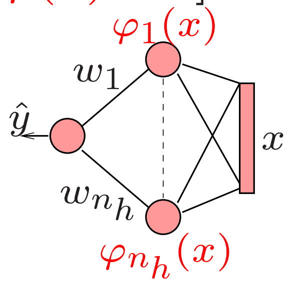


Input space

Feature space

Primal space

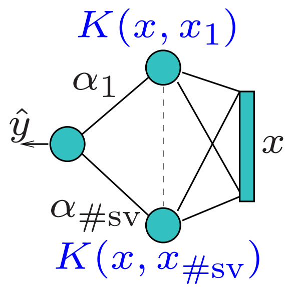
$$\hat{y} = \text{sign}[w^T \varphi(x) + b]$$



$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$$ **("Kernel trick")**

Dual space

$$\hat{y} = \text{sign}[\sum_{i=1}^{\#\text{sv}} \alpha_i y_i K(x, x_i) + b]$$



SVM classifier: primal and dual problem

- **Primal problem:** [Vapnik, 1995]

$$\min_{w,b,\xi} \mathcal{J}(w,\xi) = \frac{1}{2} w^T w + c \sum_{i=1}^{N} \xi_i \quad \text{s.t.} \quad \begin{cases} y_i[w^T \varphi(x_i) + b] \geq 1 - \xi_i \\ \xi_i \geq 0, \quad i = 1, ..., N \end{cases}$$

Trade-off between margin maximization and tolerating misclassifications

SVM classifier: primal and dual problem

- **Primal problem:** [Vapnik, 1995]

$$\min_{w,b,\xi} \mathcal{J}(w,\xi) = \frac{1}{2} w^T w + c \sum_{i=1}^{N} \xi_i \ \text{ s.t. } \left\{ \begin{array}{c} y_i [w^T \varphi(x_i) + b] \geq 1 - \xi_i \\ \xi_i \geq 0, \quad i = 1, ..., N \end{array} \right.$$

  Trade-off between margin maximization and tolerating misclassifications

- Conditions for optimality from Lagrangian.
  Express the solution in the Lagrange multipliers.

- **Dual problem**: QP problem (convex problem)

$$\max_{\alpha} \mathcal{Q}(\alpha) = -\frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \, K(x_i, x_j) \, \alpha_i \alpha_j + \sum_{j=1}^{N} \alpha_j \ \text{ s.t. } \left\{ \begin{array}{l} \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq c, \ \forall i \end{array} \right.$$

Obtaining solution via Lagrangian

- Lagrangian:

$$\mathcal{L}(w, b, \xi; \alpha, \nu) = \mathcal{J}(w, \xi) - \sum_{i=1}^{N} \alpha_i \{ y_i [w^T \varphi(x_i) + b] - 1 + \xi_i \} - \sum_{i=1}^{N} \nu_i \xi_i$$

- Find saddle point: $\max_{\alpha,\nu} \min_{w,b,\xi} \mathcal{L}(w, b, \xi; \alpha, \nu)$, one obtains

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \quad \rightarrow \quad 0 \leq \alpha_i \leq c, \ i = 1, ..., N \end{cases}$$

Finally, write the solution in terms of α (Lagrange multipliers).

SVM classifier: model representations

- Classifier: Primal representation: $\hat{y} = \text{sign}[w^T \varphi(x) + b]$

  Kernel trick (Mercer Theorem):

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) = \sum_{l=1}^{n_h} \varphi_l(x_i)\varphi_l(x_j)$$

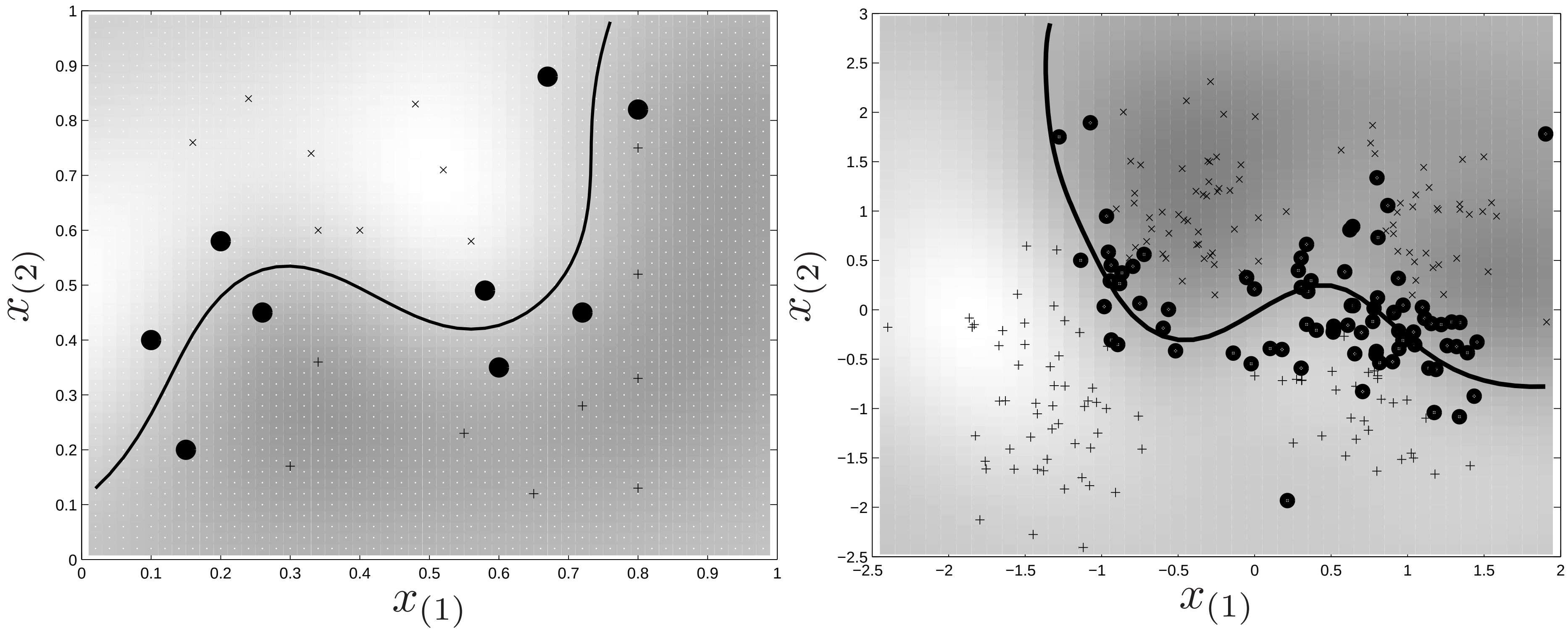
- Dual representation: (sparse model: many $\alpha_i = 0$)

$$\hat{y} = \text{sign}[\sum_i \alpha_i \, y_i \, K(x, x_i) + b]$$

  Some possible kernels:

  $K(x, x_i) = x_i^T x$ (linear)
  $K(x, x_i) = (x_i^T x + \tau)^d$ with $\tau \geq 0$ (polynomial)
  $K(x, x_i) = \exp(-\|x - x_i\|_2^2 / \sigma^2)$ (RBF Gaussian)
  $K(x, x_i) = \tanh(\kappa \, x_i^T x + \theta)$ (MLP)

Support vectors



- Decision boundary can be expressed in terms of a limited number of support vectors (subset of given training data: $\alpha_i \neq 0$); sparseness property

- Classifier follows from the solution to a convex QP problem.

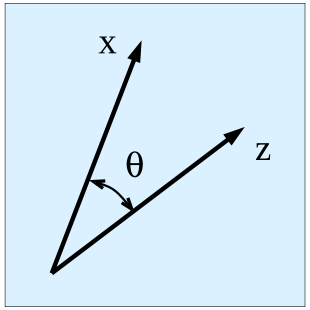
Selection of tuning parameters

- a **careful tuning** is needed to determine all tuning parameters (e.g. by using a validation set or 10-fold cross-validation)

- linear kernel: value c

- RBF kernel: value c and kernel tuning parameter σ

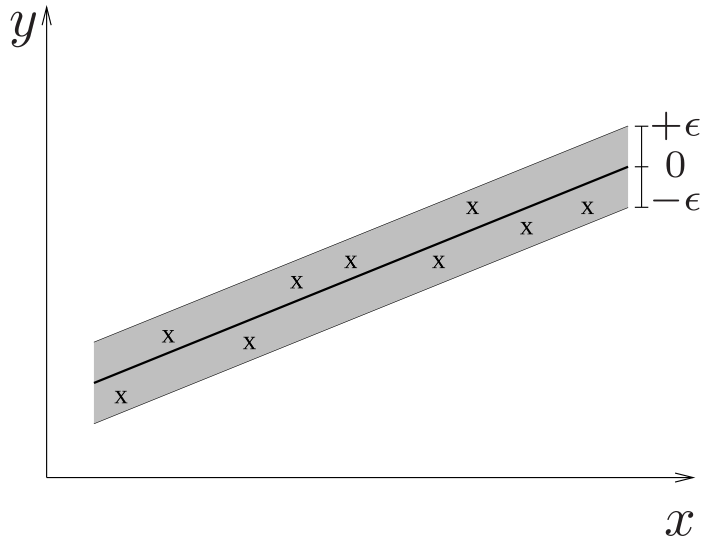
Wider use of the "kernel trick"



- **Angle between vectors:** (e.g. correlation analysis)
  Input space:
  $$\cos\theta_{xz} = \frac{x^T z}{\|x\|_2 \|z\|_2}$$
  Feature space:
  $$\cos\theta_{\varphi(x),\varphi(z)} = \frac{\varphi(x)^T \varphi(z)}{\|\varphi(x)\|_2 \|\varphi(z)\|_2} = \frac{K(x,z)}{\sqrt{K(x,x)}\sqrt{K(z,z)}}$$

- **Distance between vectors:** (e.g. for "kernelized" clustering methods)
  Input space:
  $$\|x - z\|_2^2 = (x-z)^T(x-z) = x^T x + z^T z - 2x^T z$$
  Feature space:
  $$\|\varphi(x) - \varphi(z)\|_2^2 = K(x,x) + K(z,z) - 2K(x,z)$$

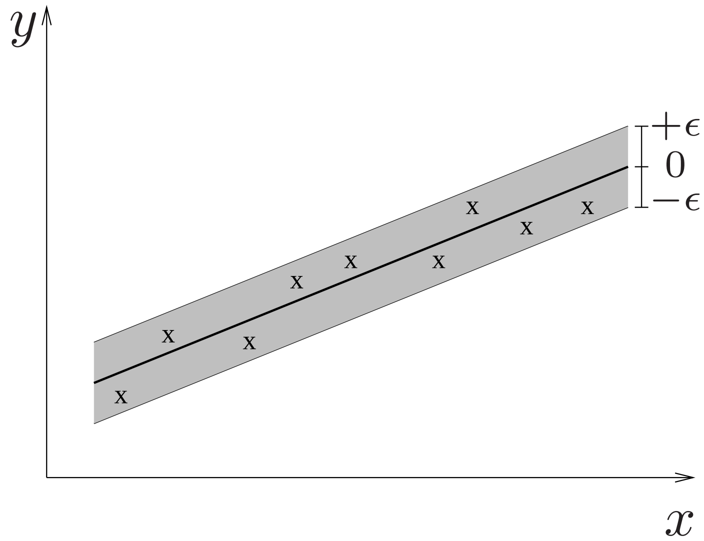
ϵ-tube



Model: $\hat{y} = w^T x + b$
Require that $\{(x_i, y_i)\}$ are contained in ϵ-tube: $|y_i - \hat{y}_i| \leq \epsilon$ or

$$|y_i - w^T x_i - b| \leq \epsilon, \;\; \forall i$$

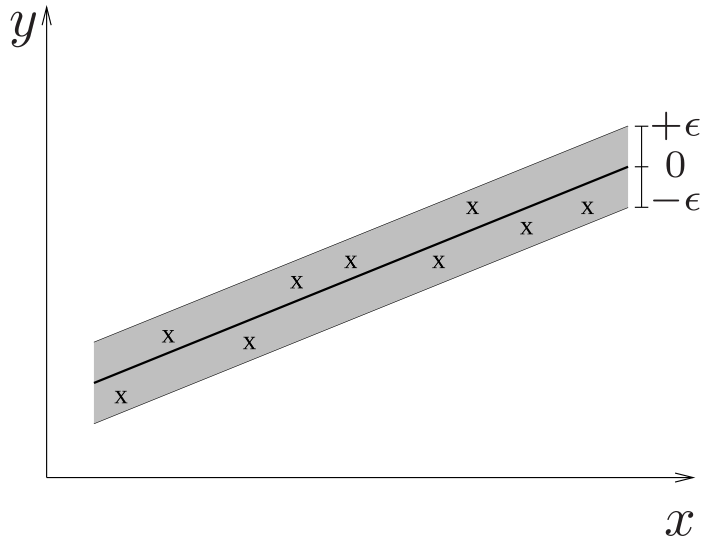
ϵ-tube



Model: $\hat{y} = w^T x + b$
Require that $\{(x_i, y_i)\}$ are contained in ϵ-tube: $|y_i - \hat{y}_i| \leq \epsilon$ or

$$
\begin{array}{rl}
 & |y_i - w^T x_i - b| \leq \epsilon, \ \ \forall i \\
\Leftrightarrow & -\epsilon \leq y_i - w^T x_i - b \leq \epsilon, \ \ \forall i
\end{array}
$$

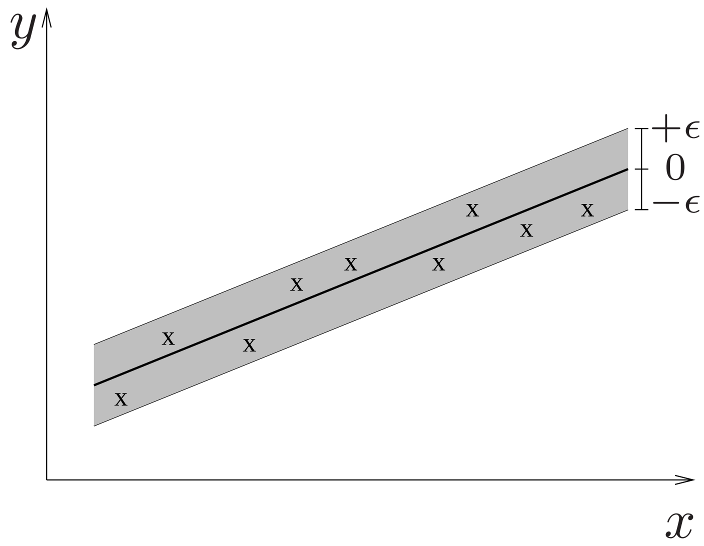
ϵ-tube



Model: $\hat{y} = w^T x + b$
Require that $\{(x_i, y_i)\}$ are contained in ϵ-tube: $|y_i - \hat{y}_i| \leq \epsilon$ or

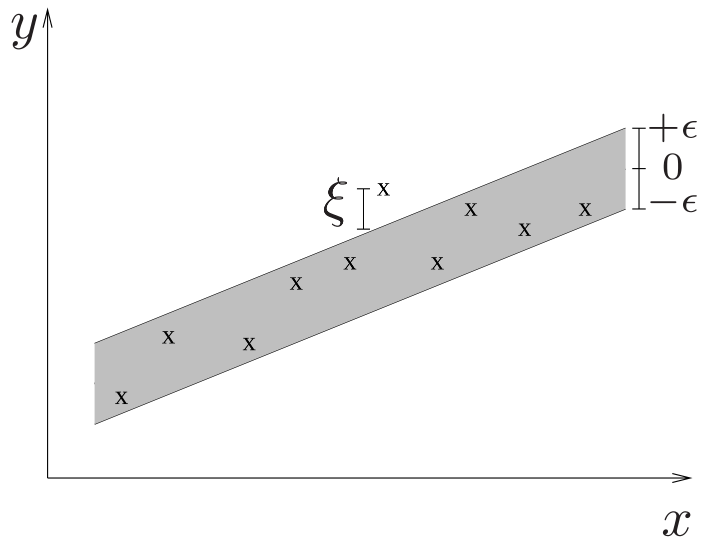
$$\begin{array}{ll} & |y_i - w^T x_i - b| \leq \epsilon, \;\; \forall i \\ \Leftrightarrow & -\epsilon \leq y_i - w^T x_i - b \leq \epsilon, \;\; \forall i \\ \Leftrightarrow & y_i - w^T x_i - b \leq \epsilon, \;\; \forall i \\ & w^T x_i + b - y_i \leq \epsilon, \;\; \forall i \end{array}$$

SVM for function estimation (linear)



$$
\begin{array}{ll}
\min\limits_{w,b} & \frac{1}{2} w^T w \\
\text{subject to} & y_i - w^T x_i - b \leq \epsilon \quad , \quad i = 1, ..., N \\
& w^T x_i + b - y_i \leq \epsilon \quad , \quad i = 1, ..., N
\end{array}
$$

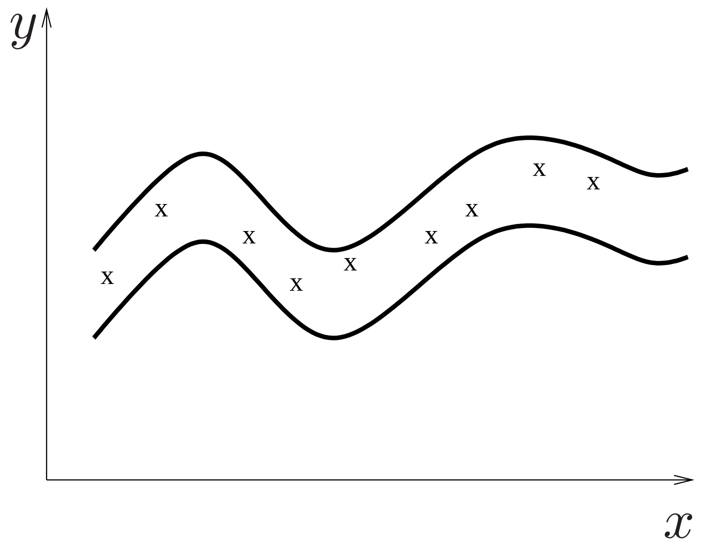
SVM for function estimation (linear)



$$\min_{w,b,\xi_i,\xi_i^*} \quad \frac{1}{2}w^T w + c\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

$$\begin{aligned} \text{subject to} \quad & y_i - w^T x_i - b \leq \epsilon + \xi_i, \quad i = 1, ..., N \\ & w^T x_i + b - y_i \leq \epsilon + \xi_i^*, \quad i = 1, ..., N \\ & \xi_i, \xi_i^* \geq 0, \quad i = 1, ..., N \end{aligned}$$

SVM for function estimation (non-linear)



$$\begin{array}{ll} \min_{w,b,\xi_i,\xi_i^*} & \frac{1}{2}w^T w + c\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ \text{subject to} & y_i - w^T\varphi(x_i) - b \leq \epsilon + \xi_i, \;\; i = 1, ..., N \\ & w^T\varphi(x_i) + b - y_i \leq \epsilon + \xi_i^*, \;\; i = 1, ..., N \\ & \xi_i, \xi_i^* \geq 0, \;\; i = 1, ..., N \end{array}$$

Additional optional material

- Burges C.J.C., "A tutorial on support vector machines for pattern recognition", *Knowledge Discovery and Data Mining*, **2**(2), 121-167, 1998.
- Cortes C., Vapnik V., "Support vector networks", *Machine Learning*, **20**, 273-297, 1995.
- Cristianini N., Shawe-Taylor J., *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.
- Schölkopf B., Smola A., *Learning with Kernels*, MIT Press, 2002.
- Suykens J.A.K., Van Gestel T., De Brabanter J., De Moor B., Vandewalle J., *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.
- Vapnik V., *Statistical Learning Theory*, John Wiley & Sons, 1998.