More on SVM, VC theory and kernels

Johan Suykens

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

Lecture 3

Contents

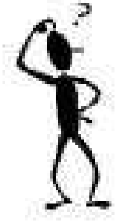
- Tuning parameters in SVM and ways to tune it
- VC theory, risk and empirical risk
- Consistency of learning process, VC dimension
- Bound on generalization error, VC dimension for SVMs
- SVM for linear and nonlinear function estimation
- More about kernels
- Wider use of the kernel trick
- Application-specific kernels

Tuning parameters in SVMs

- SVM classifier with **RBF kernel**

$$y(x) = \mathrm{sign}[\sum_{k=1}^{N} \alpha_k y_k \exp\left(-\frac{\|x - x_k\|_2^2}{\sigma^2}\right) + b]$$

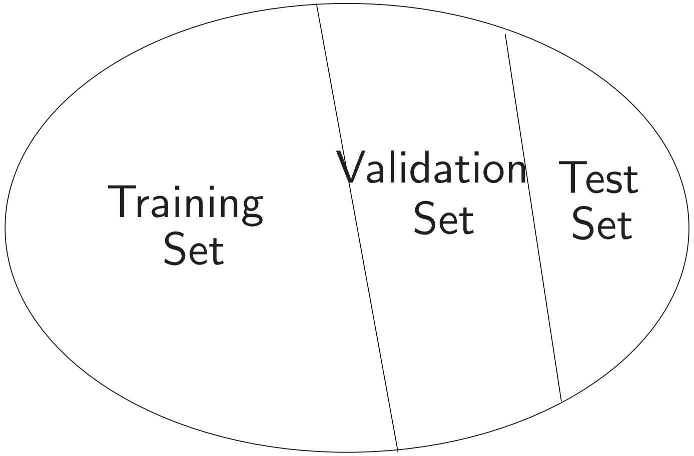
- Unknowns to be determined:
  - α, b by solving QP problem
  - tuning parameter σ → **how ?**
    (and any other tuning parameters)



- Also note that:
  σ large: decision boundary tends to become linear
  σ small: decision boundary becomes highly nonlinear

... a simple way to determine σ

- Define a training set, validation set and test set



- Find α, b based on the given **training data** and try this for several values of σ. Select the value of σ for which the error on the **validation set** is minimal. Finally check the selected model on a **test set** (this set may not be used in any sense to determine parameters of the model!).

Tuning parameter selection methods

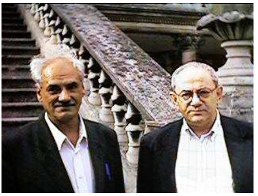
- *Optimization on a separate validation set:*
  In this case the designer is responsible for defining a meaningful training and validation set. The generalization performance should be checked on a completely independent test set.

- *Cross-validation:*
  Tuning parameters are optimized on the sum of the parts of the training set that were left out in the several runs of the cross-validation process.

- *Bayesian inference*

- *Generalization bounds from statistical learning theory (VC theory)*

Statistical learning theory (VC theory)

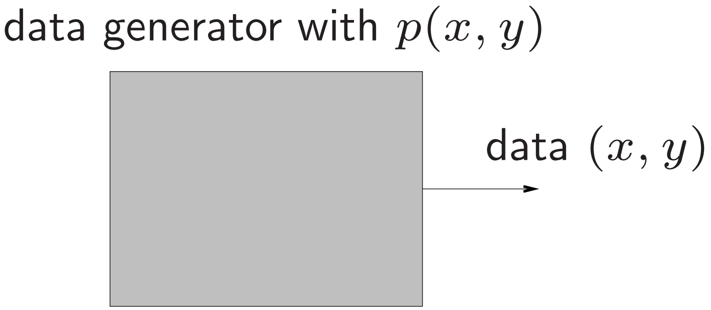


- Binary classification problem:
  Consider set of functions with adjustable parameters θ

$$\{f(x;\theta) : \theta \in \Theta\}, \; f(x;\theta) : \mathbb{R}^n \to \{-1, +1\}$$

and a set of training examples, i.e. pairs of patterns x_k and labels y_k:

$$(x_1, y_1), ..., (x_N, y_N) \in \mathbb{R}^n \times \{-1, +1\}$$



In VC theory **no assumptions** are made about $p(x, y)$, except that it generates data that are **i.i.d.** (independently identically distributed).

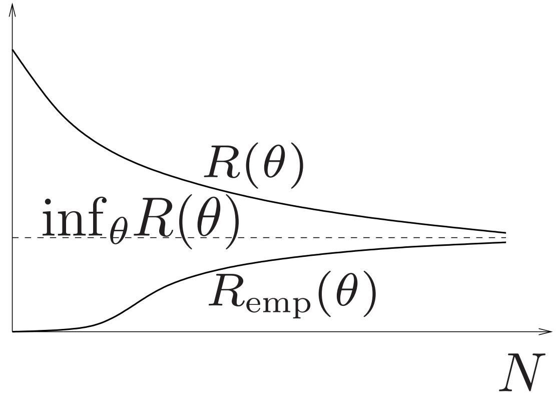
Statistical learning theory (2)

- Definition of **risk** (i.e. generalization error):

$$R(\theta) = \int \frac{1}{2}|y - f(x;\theta)|\, p(x,y)\, dxdy$$

Empirical risk (training set error): $R_{\mathrm{emp}}(\theta) = \frac{1}{2N}\sum_{k=1}^{N}|y_k - f(x_k;\theta)|$

- **Consistency** of a learning process



If the expected risk $R(\theta)$ and the empirical risk $R_{\mathrm{emp}}(\theta)$ converge to $\inf_\theta R(\theta)$ for $N \to \infty$, then the learning process is consistent.

Statistical learning theory (3)

- *Theorem* [Vapnik, 1979]: For any $\theta \in \Theta$ and $N > h$, the upper bound

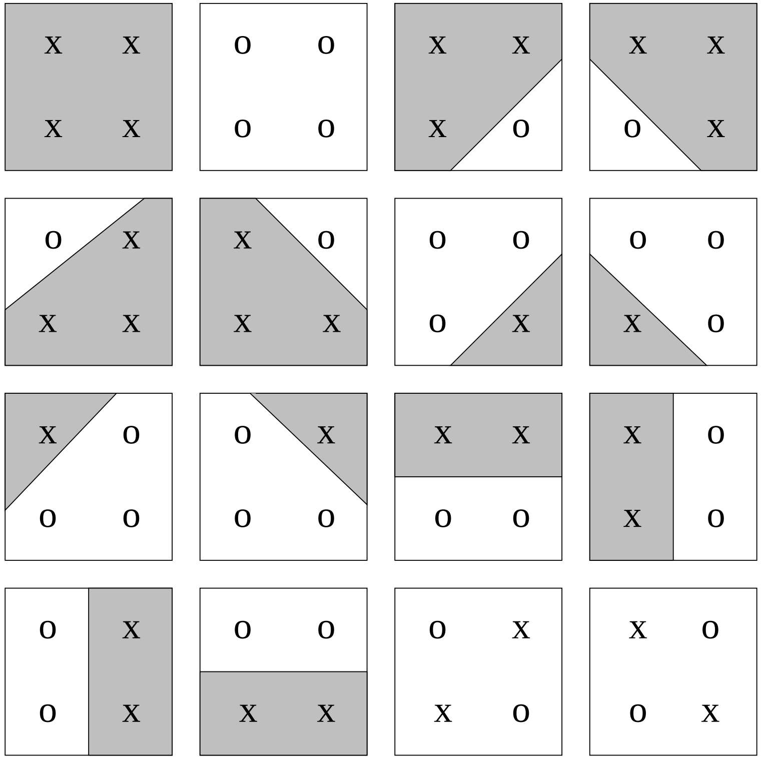
  $$R(\theta) \leq R_{\mathrm{emp}}(\theta) + \sqrt{\frac{h(\log(2N/h) + 1) - \log(\eta/4)}{N}}$$

  holds **with probability** $1 - \eta$, where the second term is a confidence term which depends on the VC dimension h.

- **VC (Vapnik-Chervonenkis) dimension** h of a set of functions: Describes the capacity of a set of functions. It is a combinatorial measure for the model complexity.

- **Without restricting** the set of admissible functions, empirical risk minimization is **not consistent**. For the empirical risk minimization principle to be consistent, it is **necessary and sufficient** that the empirical risk $R_{\mathrm{emp}}(\theta)$ converges uniformly to the risk $R(\theta)$.

VC dimension for linear separating hyperplanes



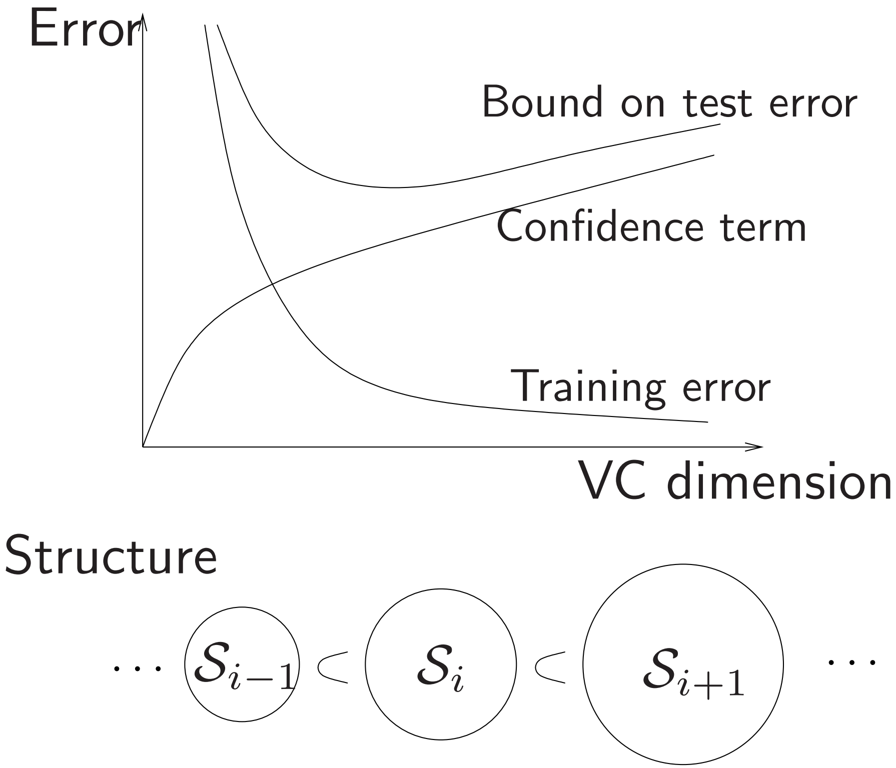
Consider $N = 4$ points in an $n = 2$ dimensional space. The points can be labelled then in $2^4 = 16$ possible ways. At most 3 points can be separated by straight lines (remember the XOR problem). The VC dimension equals the maximum number of points that can be separated, i.e. $h = 3$ (in general $h = n + 1$ for hyperplanes). (link with Popper's work in philosophy: four points can falsify any linear law in this example).

Structural risk minimization

- Consider the structure (nested sets of functions):

$$S_i = \{w^T \varphi(x) + b : \|w\|_2^2 \leq c_i\}, \ c_1 < c_2 < c_3 ...$$

- Interpretation of bound on generalization error:


Error
Bound on test error
Confidence term
Training error
VC dimension
Structure
$\cdots \ \mathcal{S}_{i-1} \subset \mathcal{S}_i \subset \mathcal{S}_{i+1} \ \cdots$

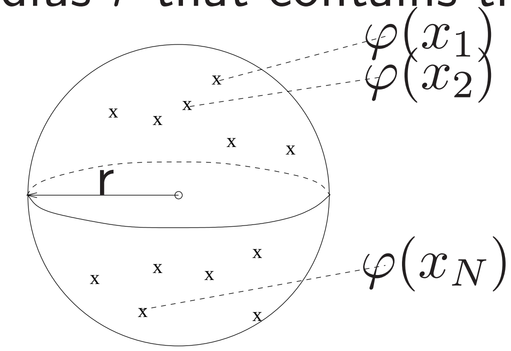
VC dimension for SVMs (1)

- For SVMs compute

$$
\begin{aligned}
w^T w &= (\sum_{k=1}^{N} \alpha_k y_k \varphi(x_k))^T \sum_{l=1}^{N} \alpha_l y_l \varphi(x_l) \\
&= \alpha^T \Omega \alpha
\end{aligned}
$$

  with $\Omega_{kl} = y_k y_l K(x_k, x_l)$ for $k, l = 1, ..., N$ after application of the kernel trick.

- Then consider the points $\varphi(x_1), ..., \varphi(x_N)$ in the feature space and find the smallest ball with radius r that contains these points



Feature space

VC dimension for SVMs (2)

- *How to find this ball ?* Optimization problem:

$$\min\ r\ \ \text{s.t.}\ \ \|\varphi(x_k) - q\|_2 < r,\ \ k = 1, 2, ..., N$$

with q a point inside the ball to be determined. The Lagrangian is $\mathcal{L}(r, q, \lambda) = r^2 - \sum_{k=1}^{N} \lambda_k (r^2 - \|\varphi(x_k) - q\|_2^2)$. The solution follows from the QP problem:

$$\max_{\lambda} \mathcal{Q}(\lambda) = -\sum_{k,l=1}^{N} K(x_k, x_l)\, \lambda_k \lambda_l + \sum_{k=1}^{N} \lambda_k K(x_k, x_k)$$

such that

$$\sum_{k=1}^{N} \lambda_k = 1\ \ \text{and}\ \ \lambda_k \geq 0\ , k = 1, ..., N$$

with $K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l)$. Furthermore $q = \sum_{k=1}^{N} \lambda_k \varphi(x_k)$.

VC dimension for SVMs (3)

- Given $w^T w \leq a^2$ and radius r, one can compute an **upper bound on the VC dimension** h which is given by

  $$h \leq \min([r^2 a^2], n) + 1$$

  where $[\cdot]$ denotes the integer part. Hence, minimize radius r and maximize the margin (note that the margin equals $2/\|w\|_2$) for getting a small VC dimension.

- This bound can be used to determine e.g. the σ value of RBF kernel and the c value in the SVM formulation.

- Although this bound is not sharp it can often be a good indication. One chooses the tuning parameters in such away that this upper bound is minimized.

SVM - linear function estimation

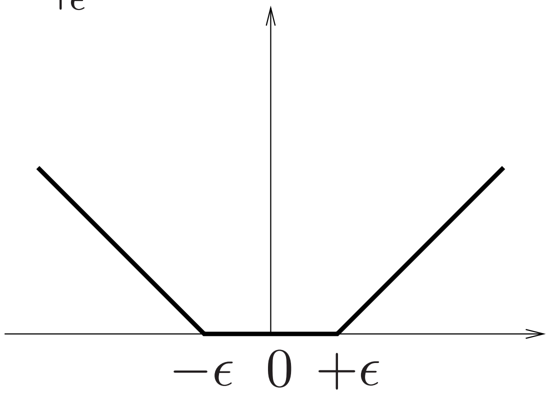
- Consider regression in the set of **linear functions** $f(x) = w^T x + b$ with N training data $x_k \in \mathbb{R}^m$ and output values $y_k \in \mathbb{R}$.

- Empirical risk minimization with

$$R_{\text{emp}} = \frac{1}{N} \sum_{k=1}^{N} |y_k - w^T x_k - b|_\epsilon$$

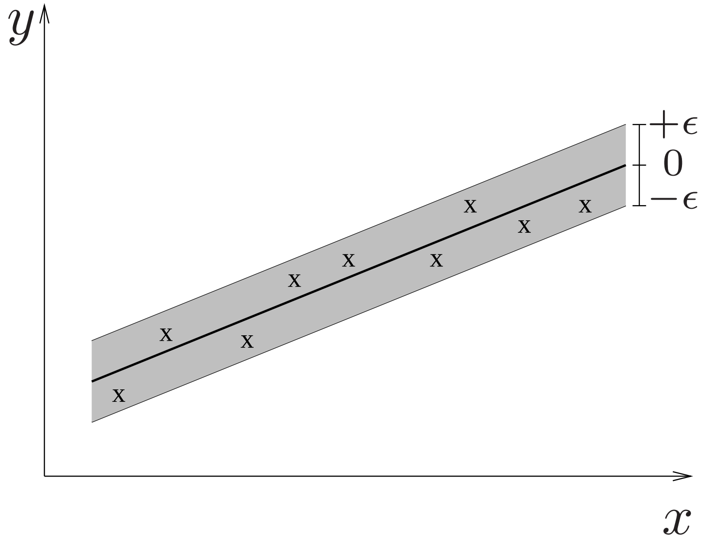
  subject to a structure S_i (with $\|w\|_2^2 \le c_i$).

- **Vapnik's ϵ-insensitive loss function**:



$$|y - f(x)|_\epsilon = \begin{cases} 0 & , \quad \text{if } |y - f(x)| \le \epsilon \\ |y - f(x)| - \epsilon & , \quad \text{otherwise} \end{cases}$$

Function estimation: ϵ-tube

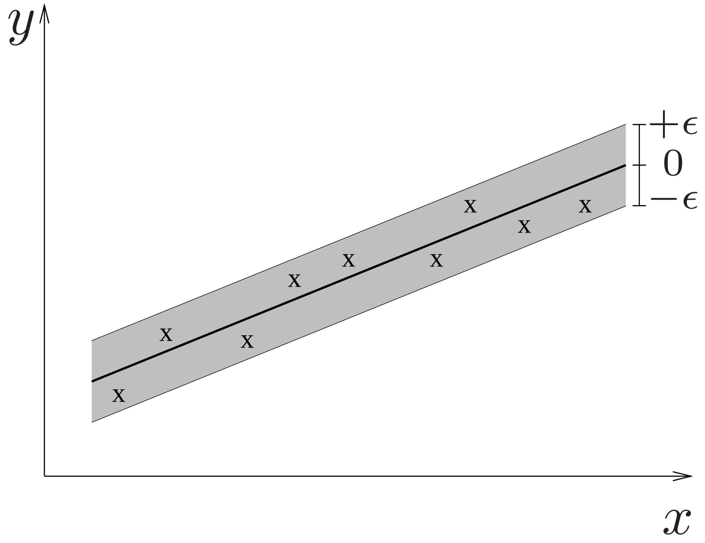


Model: $\hat{y} = w^T x + b$

Require that $\{(x_i, y_i)\}$ are contained in ϵ-tube: $|y_i - \hat{y}_i| \le \epsilon$ or

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

Function estimation: ϵ-tube

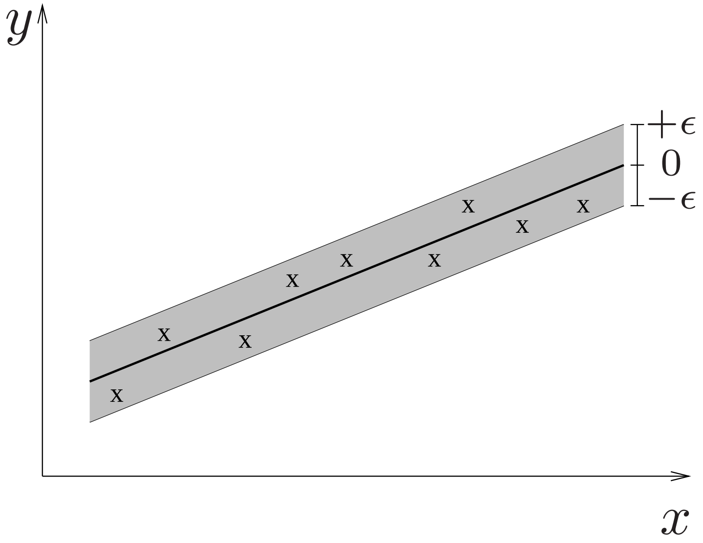


Model: $\hat{y} = w^T x + b$
Require that $\{(x_i, y_i)\}$ are contained in ϵ-tube: $|y_i - \hat{y}_i| \leq \epsilon$ or

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

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

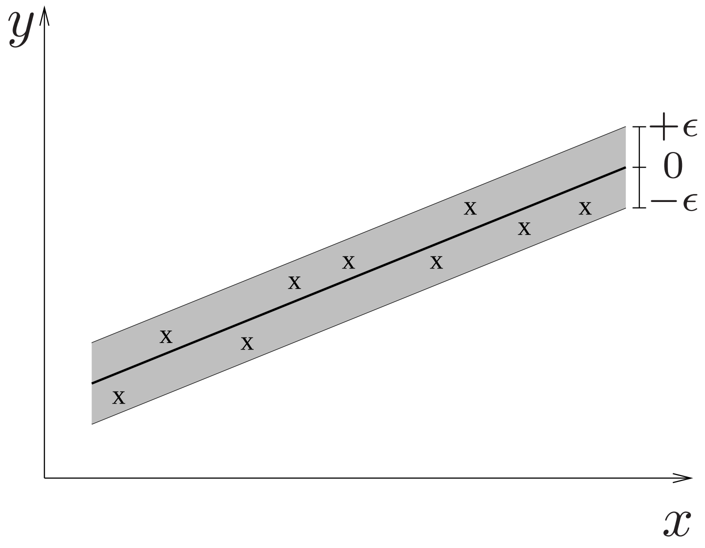
Function estimation: ϵ-tube



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

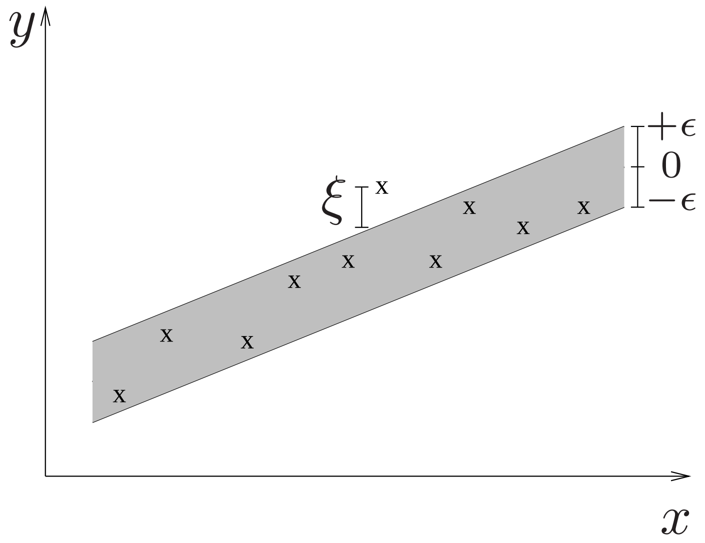
$$\begin{array}{rl} & |y_i - w^T x_i - b| \leq \epsilon, \ \ \forall i \\ \Leftrightarrow & -\epsilon \leq y_i - w^T x_i - b \leq \epsilon, \ \ \forall i \\ \Leftrightarrow & y_i - w^T x_i - b \leq \epsilon, \ \ \forall i \\ & w^T x_i + b - y_i \leq \epsilon, \ \ \forall i \end{array}$$

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)



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

SVM for function estimation (linear)

- **Lagrangian**:

$$
\begin{aligned}
&\mathcal{L}(w,b,\xi,\xi^*;\alpha,\alpha^*,\eta,\eta^*) = \\
&\frac{1}{2}w^Tw + c\sum_{k=1}^{N}(\xi_k+\xi_k^*) - \sum_{k=1}^{N}\alpha_k(\epsilon+\xi_k-y_k+w^Tx_k+b) \\
&-\sum_{k=1}^{N}\alpha_k^*(\epsilon+\xi_k^*+y_k-w^Tx_k-b) - \sum_{k=1}^{N}(\eta_k\xi_k+\eta_k^*\xi_k^*)
\end{aligned}
$$

  with Lagrange multipliers $\alpha_k, \alpha_k^*, \eta_k, \eta_k^* \geq 0$.

- **Saddle point** of Lagrangian:

$$
\max_{\alpha,\alpha^*,\eta,\eta^*}\ \min_{w,b,\xi,\xi^*}\ \mathcal{L}(w,b,\xi,\xi^*;\alpha,\alpha^*,\eta,\eta^*)
$$

SVM for function estimation (linear)

- **Conditions for optimality**:

$$\left\{\begin{array}{lcl}
\frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow & w = \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) x_k \\
\frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) = 0 \\
\frac{\partial \mathcal{L}}{\partial \xi_k} = 0 & \rightarrow & c - \alpha_k - \eta_k = 0 \\
\frac{\partial \mathcal{L}}{\partial \xi_k^*} = 0 & \rightarrow & c - \alpha_k^* - \eta_k^* = 0
\end{array}\right.$$

SVM for function estimation (linear)

- Resulting **dual problem**:

$$\max_{\alpha,\alpha_*} \mathcal{Q}(\alpha,\alpha_*) = -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)\,x_k^T x_l$$
$$-\epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)$$

subject to

$$\left\{\begin{array}{lcl} \sum_{k=1}^{N}(\alpha_k-\alpha_k^*) & = & 0 \\ \alpha_k,\alpha_k^* & \in & [0,c] \end{array}\right.$$

This is a **QP problem** to be solved in the unknowns α_k, α_k^*.

SVM for function estimation (linear)

- Computation of b term from KKT conditions:

$$\begin{array}{rcl} \alpha_k(\epsilon + \xi_k - y_k + w^T x_k + b) & = & 0 \\ \alpha_k^*(\epsilon + \xi_k^* + y_k - w^T x_k - b) & = & 0 \end{array}$$

and

$$\begin{array}{rcl} \eta_k \xi_k = (c - \alpha_k)\xi_k & = & 0 \\ \eta_k^* \xi_k^* = (c - \alpha_k^*)\xi_k^* & = & 0 \end{array}$$

Hence

$$\alpha_k \alpha_k^* = 0$$

and

$$\begin{array}{rcl} b & = & y_k - w^T x_k - \epsilon \quad \text{for} \quad \alpha_k \in (0, c) \\ b & = & y_k - w^T x_k + \epsilon \quad \text{for} \quad \alpha_k^* \in (0, c) \end{array}$$

SVM - nonlinear function estimation

- Consider again a nonlinear mapping to the **feature space**:

$$f(x) = w^T \varphi(x) + b$$

with given training data $\{x_k, y_k\}_{k=1}^N$.

- **Optimization problem:**

$$\min \ \frac{1}{2} w^T w + c \sum_{k=1}^N (\xi_k + \xi_k^*)$$

subject to

$$\left\{ \begin{array}{rcl} y_k - w^T \varphi(x_k) - b & \leq & \epsilon + \xi_k \\ w^T \varphi(x_k) + b - y_k & \leq & \epsilon + \xi_k^* \\ \xi_k, \xi_k^* & \geq & 0 \end{array} \right.$$

SVM - nonlinear function estimation

- **Dual problem**:

$$\max_{\alpha,\alpha_*} \mathcal{Q}(\alpha,\alpha_*) = -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)\,K(x_k,x_l)$$
$$-\epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\sum_{k=1}^{N}y_k(\alpha_k-\alpha_k^*)$$

subject to

$$\left\{\begin{array}{lcl}\sum_{k=1}^{N}(\alpha_k-\alpha_k^*) & = & 0\\ \alpha_k,\alpha_k^* & \in & [0,c]\end{array}\right.$$

with application of the Mercer theorem:

$$K(x_k,x_l)=\varphi(x_k)^T\varphi(x_l)$$

SVM model in primal and dual

- in the **primal**:

$$y(x) = w^T \varphi(x) + b$$

with unknowns w, b. Note that w can be infinite dimensional.

- in the **dual space**:

$$y(x) = \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) K(x_k, x) + b$$

with unknowns α, α^*, b

(this follows from the fact that $w = \sum_{k=1}^{N} (\alpha_k - \alpha_k^*) \varphi(x_k)$)

SVM function estimation - example (1)

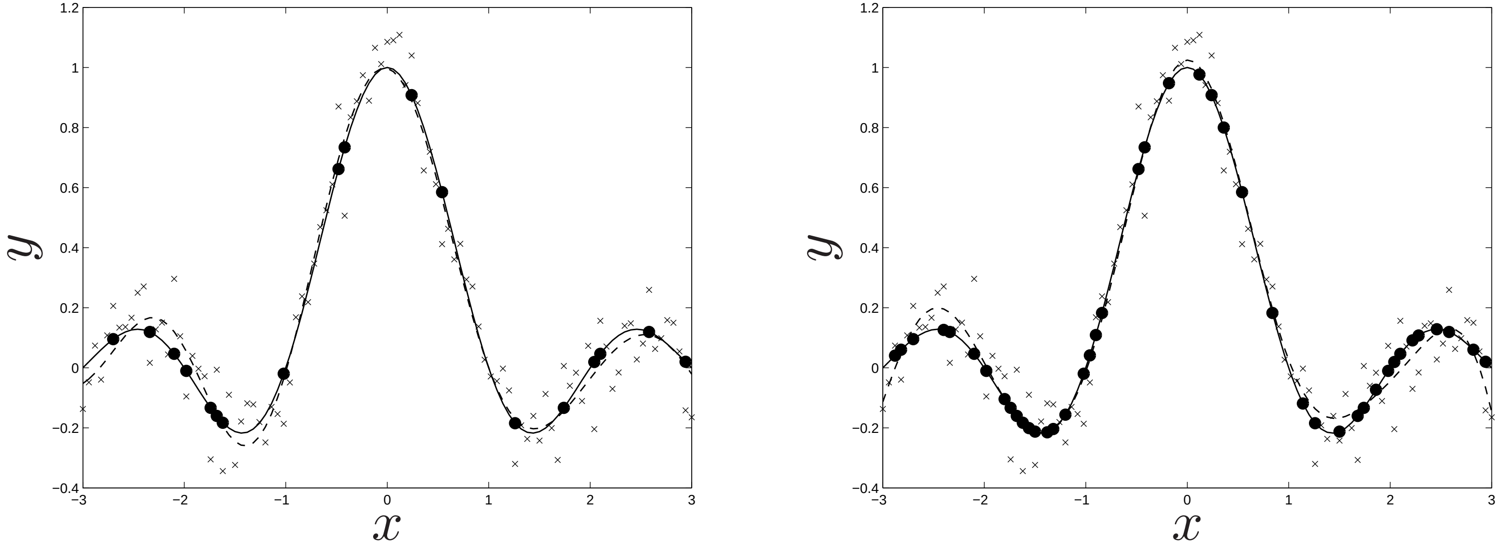


Illustration of SVM with Vapnik ϵ-insensitive loss function on a noisy sinc function (zero mean Gaussian noise with standard dev. equal to $\mathbf{0.01}$) with (Left) $c = 100$ and $\epsilon = 0.15$ and RBF kernel with $\sigma = 0.8$ giving 18 support vectors and (Right) $\epsilon = 0.15/2$ giving 43 support vectors. True sinc function (solid line); SVM output (dashed line).

SVM function estimation - example (2)

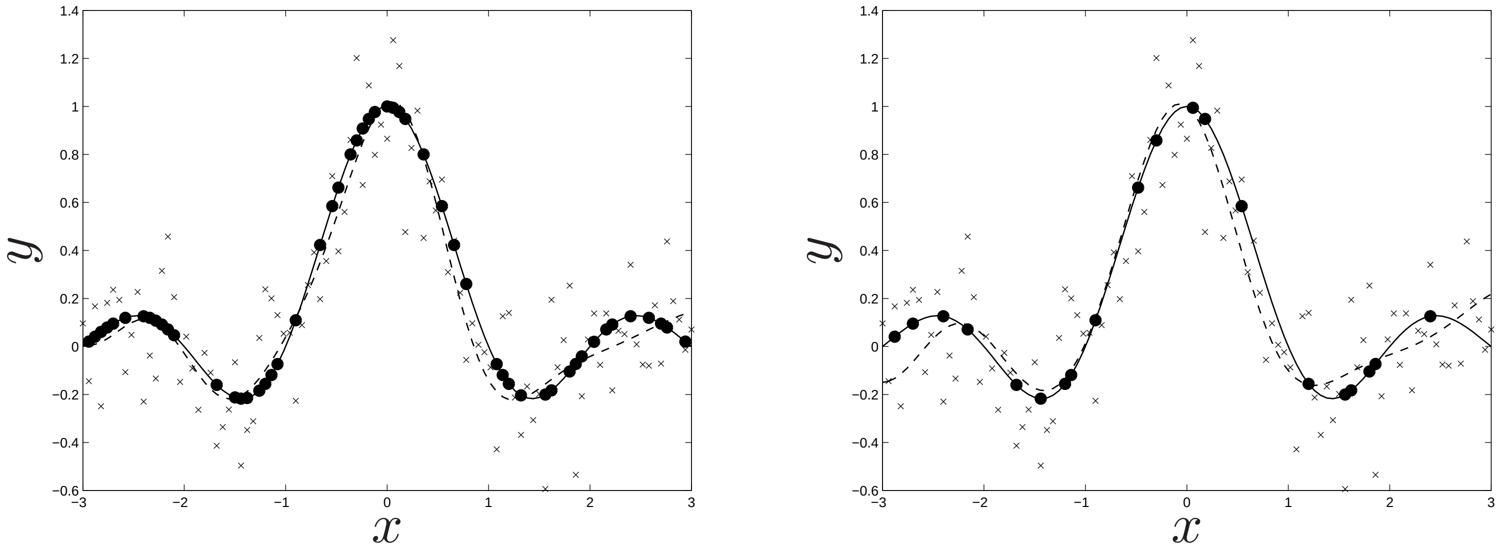


Illustration of SVM with Vapnik ϵ-insensitive loss function on a noisy sinc function (zero mean Gaussian noise with standard dev. equal to **0.05**) with (Left) $c = 100$ and $\epsilon = 0.15$ and RBF kernel with $\sigma = 0.8$ resulting into 54 support vectors and (Right) $\epsilon = 0.3$ giving 20 support vectors. True sinc function (solid line); SVM output (dashed line).

VC theory for function estimation (1)

- **Empirical risk** with squared loss function

$$R_{\mathrm{emp}}(\theta) = \frac{1}{N}\sum_{k=1}^{N}\left(y_k - f(x_k;\theta)\right)^2 .$$

Predicted risk (**generalization error**)

$$R(\theta) = \int \left(y - f(x;\theta)\right)^2 p(x,y)dxdy$$

- **VC bound**

$$R(\theta) \leq R_{\mathrm{emp}}(\theta)\left(1 - c\sqrt{\frac{h(\ln(aN/h)+1) - \ln\eta}{N}}\right)_{+}^{-1}$$

with h the VC dimension of the set of approximating functions. Typically $a = c = 1$ is chosen. This bound holds with probability $1-\eta$.

VC theory for function estimation (2)

- The estimated risk can be expressed as

$$R_{\text{est}} = g(h, N) \frac{1}{N} \sum_{k=1}^{N} (y_k - f(x_k; \theta))^2$$

  where $g(h, N)$ is a correcting function.

- Other model selection criteria:
  - *Finite prediction error* (FPE) (Akaike): $g(d, N) = \frac{1+d/N}{1-d/N}$
  - *Generalized CV* (GCV) (Craven & Wahba): $g(d, N) = \frac{1}{\left(1-\frac{d}{N}\right)^2}$
  - *Shibata's model selector* (SMS): $g(d, N) = 1 + 2\frac{d}{N}$
  - *Schwarz criteria* (MDL criteria): $g(d, N) = 1 + \frac{\frac{d}{N}\log N}{2\left(1-\frac{d}{N}\right)}$

  with d the number of free parameters for a model which is linear in the parameters.

Kernels from kernels

- $K(x,z) = aK_1(x,z) \quad (a > 0)$
- $K(x,z) = K_1(x,z) + b \quad (b > 0)$
- $K(x,z) = x^T P z \quad (P = P^T > 0)$
- $K(x,z) = K_1(x,z) + K_2(x,z)$
- $K(x,z) = K_1(x,z) K_2(x,z)$
- $K(x,z) = f(x) f(z)$
- $K(x,z) = K_3(\phi(x), \phi(z))$
- $K(x,z) = p_+(K_4(x,z))$
- $K(x,z) = \exp(K_4(x,z))$

where $a, b \in \mathbb{R}^+$ and K_1, K_2, K_3, K_4 symmetric positive definite kernel functions, $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$, $\phi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$ and $p_+(\cdot)$ is a polynomial with positive coefficients.

Normalization of kernels (1)

- **Linear kernel** $K(x,z) = x^T z$.
  Normalized linear kernel:

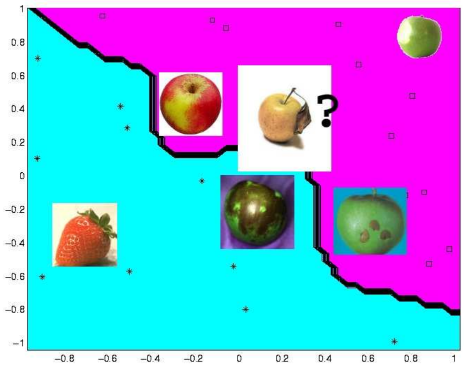
$$K(x,z) = \frac{x^T z}{\|x\|_2 \|z\|_2} = \cos(\theta_{\{x,z\}})$$

- **Nonlinear kernels** with feature map

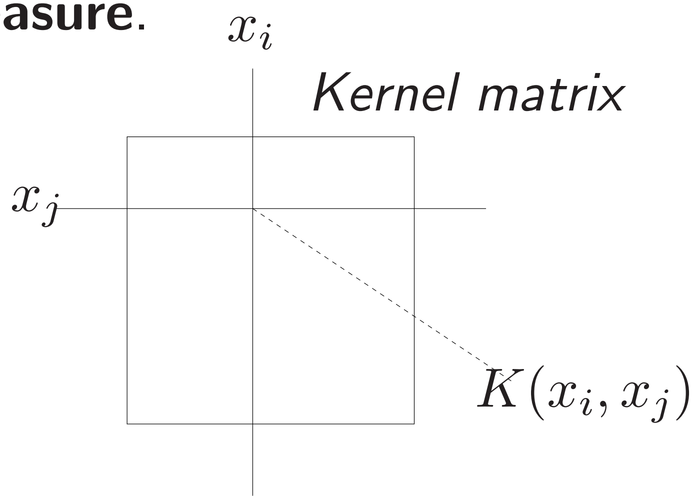
$$\tilde{K}(x,z) \quad = \quad \frac{K(x,z)}{\sqrt{K(x,x)}\sqrt{K(z,z)}} = \cos(\theta_{\{\varphi(x),\varphi(z)\}})$$

with $\tilde{K}$ the normalized kernel and $\theta_{\{\varphi(x),\varphi(z)\}}$ the angle between $\varphi(x)$ and $\varphi(z)$ in the feature space.

Normalization of kernels (2)



Elements $K(x_i, x_j)$ of a (normalized) kernel matrix look at the angle between $\varphi(x_i)$ and $\varphi(x_j)$ vectors in the feature space. Hereby $K(x_i, x_j)$ serves as a **similarity measure**.



Wider use of the kernel trick

- In many knowledge discovery and clustering algorithms one computes **distances between vectors**.

- Instead of considering a distance between vectors x, z

$$\|x - z\|_2$$

in the original space, one could also compute distances between $\varphi(x)$ and $\varphi(z)$ in a feature space (even when $\varphi(\cdot)$ is infinite dimensional):

$$\begin{aligned} \|\varphi(x) - \varphi(z)\|_2^2 &= \varphi(x)^T \varphi(x) + \varphi(z)^T \varphi(z) - 2\varphi(x)^T \varphi(z) \\ &= K(x, x) + K(z, z) - 2K(x, z) \end{aligned}$$

after application of the kernel trick.

Application specific kernels

Any positive definite kernel can be used within the SVM context. Furthermore, one can also develop kernels that are tailored towards the application itself.

Examples:

- string kernels in *textmining*
- special kernels for *bioinformatics* applications

Classical MLPs via SVM

For the MLP

$$y(x) = \mathrm{sign}[w^T \tanh(Vx + \beta)]$$

one can **explicitly define as feature map**

$$\varphi(x) = \tanh(Vx + \beta)$$

where the feature space corresponds to the hidden layer space.

The kernel function becomes then

$$\begin{aligned} K(x,z) &= \varphi(x)^T \varphi(z) \\ &= \tanh(Vx + \beta)^T \tanh(Vz + \beta) \end{aligned}$$

In this case it is clear that the kernel has all the elements of the hidden layer matrix and bias vector as tuning parameters of the kernel (instead of only one in the case of the RBF kernel).