Support Vector Machines and Kernel based Learning

Johan Suykens

K.U. Leuven, ESAT-SCD-SISTA
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium

Tel: 32/16/32 18 02 - Fax: 32/16/32 19 70
Email: johan.suykens@esat.kuleuven.ac.be
http://www.esat.kuleuven.ac.be/sista/members/suykens.html
http://www.esat.kuleuven.ac.be/sista/lssvmlab/

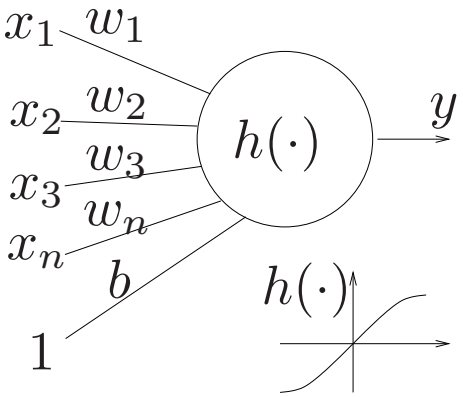
IJCNN 2005 Tutorial - July 2005

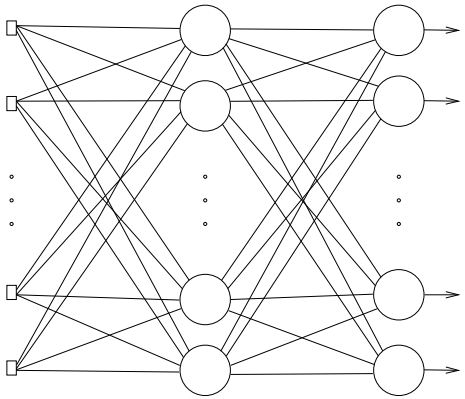
◇

Contents - Part I: Basics

- Motivation
- Basics of support vector machines
- Use of the "kernel trick"
- Kernelbased learning
- Least squares support vector machines
- Applications

Classical MLPs





Multilayer Perceptron (MLP) properties:

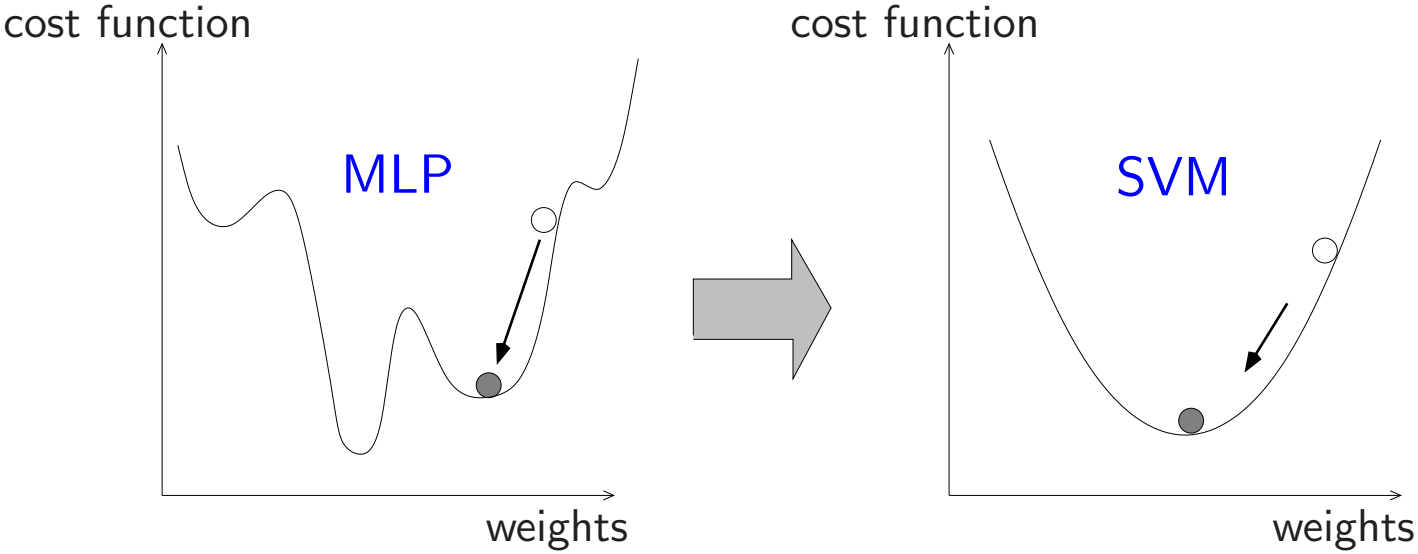
- Universal approximation of continuous nonlinear functions
- Learning from input-output patterns; either off-line or on-line learning
- Parallel network architecture, multiple inputs and outputs

Use in feedforward and recurrent networks

Use in supervised and unsupervised learning applications

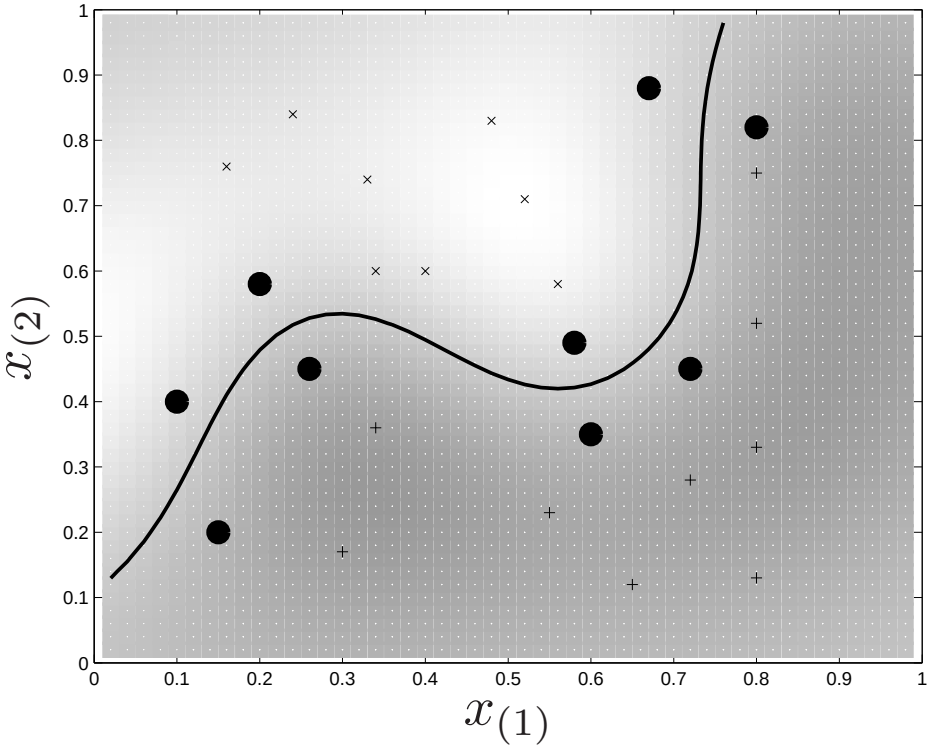
Problems: Existence of many local minima!
How many neurons needed for a given task?

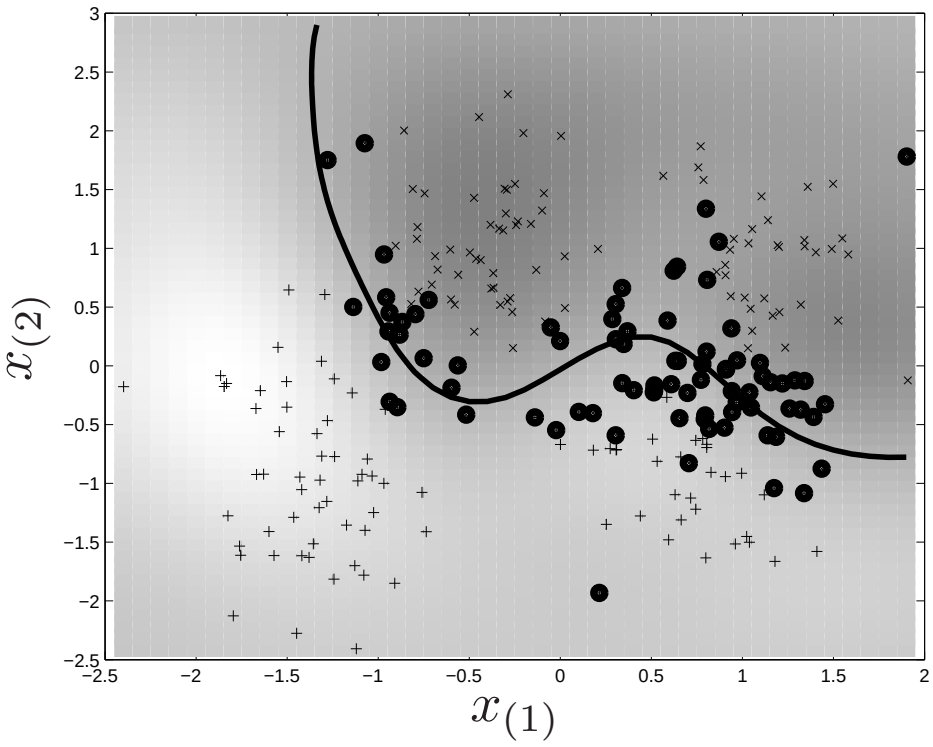
Support Vector Machines (SVM)



- 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 huge dimensional input spaces (able to avoid the curse of dimensionality!).
- Use of kernels (e.g. linear, polynomial, RBF, MLP, splines, ...). Application-specific kernels possible (e.g. textmining, bioinformatics)

SVM: support vectors

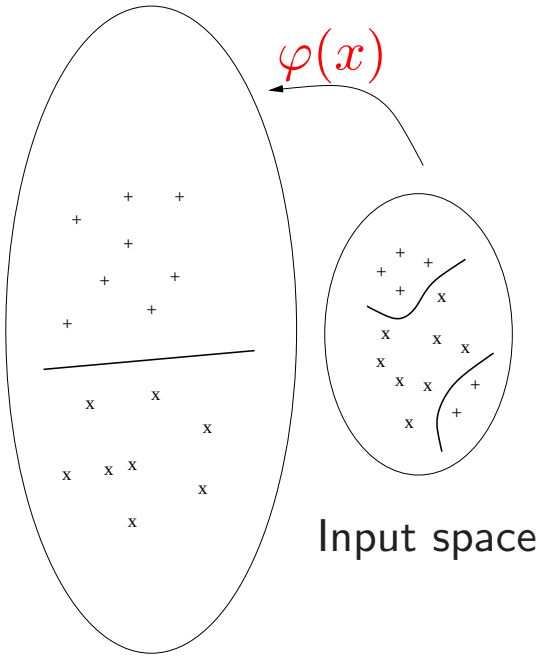




- Decision boundary can be expressed in terms of a limited number of support vectors (subset of given training data); sparseness property

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

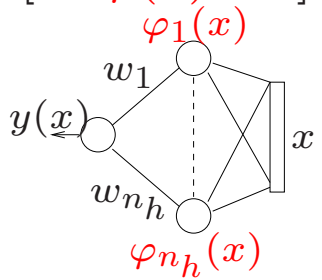
SVMs: living in two worlds ...



Primal space: ($\rightarrow$ large data sets)

Parametric: estimate $w \in \mathbb{R}^{n_h}$

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

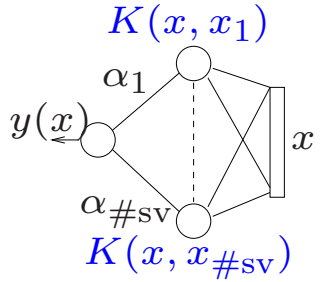


$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ **("Kernel trick")**

Dual space: ($\rightarrow$ high dimensional inputs)

Non-parametric: estimate $\alpha \in \mathbb{R}^N$

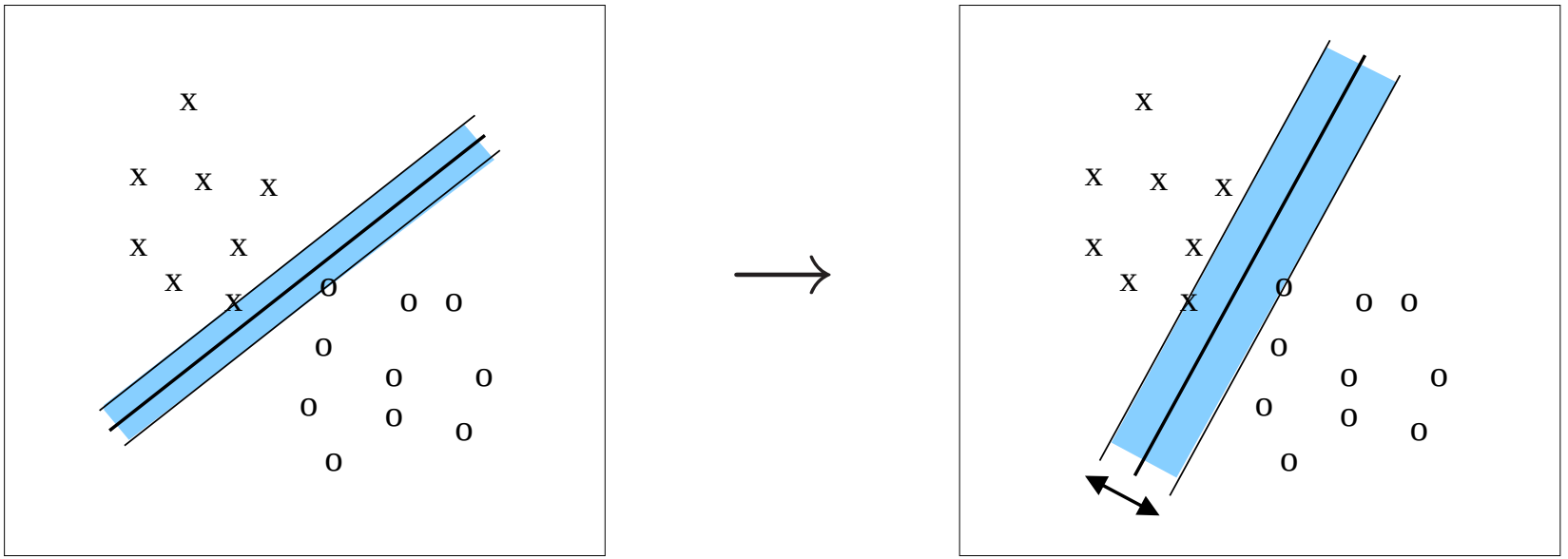
$y(x) = \text{sign}[\sum_{i=1}^{\#\text{sv}} \alpha_i y_i K(x, x_i) + b]$



Classifier with maximal margin

- Training set $\{(x_i, y_i)\}_{i=1}^N$: inputs $x_i \in \mathbb{R}^n$; class labels $y_i \in \{-1, +1\}$
- Classifier: $y(x) = \text{sign}[w^T \varphi(x) + b]$

  with $\varphi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$ the mapping to a high dimensional feature space (which can be infinite dimensional!)
- Maximize the margin for good generalization ability (margin $= \frac{2}{\|w\|_2}$):



Standard SVM classifier (1)

[Vapnik, 1995]

- For separable data, assume

$$\begin{cases} w^T\varphi(x_i)+b \geq +1, & \text{if } y_i=+1 \\ w^T\varphi(x_i)+b \leq -1, & \text{if } y_i=-1 \end{cases} \Rightarrow y_i[w^T\varphi(x_i)+b] \geq 1, \forall i$$

- Optimization problem (non-separable case):

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

Trade-off between margin maximization and tolerating misclassifications

Standard SVM classifier (2)

- 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

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

Standard SVM classifier (3)

- Dual problem: QP problem

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

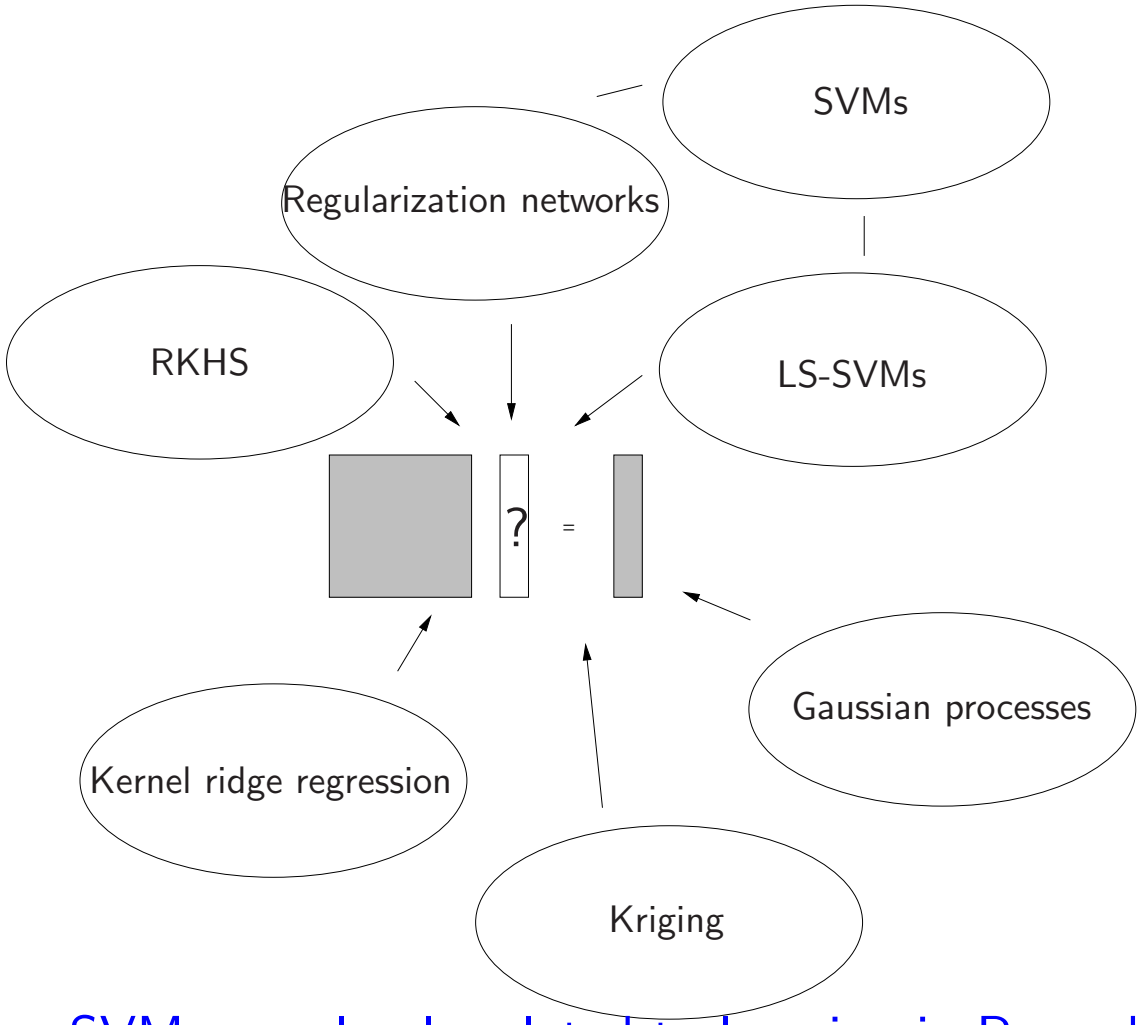
  with kernel trick (Mercer Theorem): $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$

- Obtained classifier: $y(x) = \text{sign}[\sum_{i=1}^{N} \alpha_i\, y_i\, K(x, x_i) + b]$

  Some possible kernels $K(\cdot, \cdot)$:

  $K(x, x_i) = x_i^T x$ (linear SVM)
  $K(x, x_i) = (x_i^T x + \tau)^d$ (polynomial SVM of degree d)
  $K(x, x_i) = \exp(-\|x - x_i\|_2^2 / \sigma^2)$ (RBF kernel)
  $K(x, x_i) = \tanh(\kappa\, x_i^T x + \theta)$ (MLP kernel)

Kernelbased learning: many related methods and fields



Some early history on RKHS:

1910-1920: Moore
1940: Aronszajn
1951: Krige
1970: Parzen
1971: Kimeldorf & Wahba

SVMs are closely related to learning in Reproducing Kernel Hilbert Spaces

Reproducing Kernel Hilbert Space (RKHS) view

- Variational problem: [Wahba, 1990; Poggio & Girosi, 1990]
  find function f such that

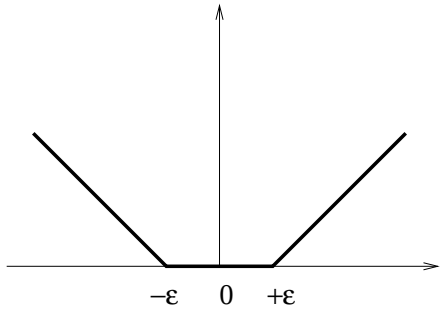
$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda \|f\|_K^2$$

  with $L(\cdot, \cdot)$ the loss function. $\|f\|_K$ is norm in RKHS $\mathcal{H}$ defined by K.

- Representer theorem: for any convex loss function the solution is of the form

$$f(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i)$$

- Some special cases:



$L(y, f(x)) = (y - f(x))^2$ regularization network
$L(y, f(x)) = |y - f(x)|_\epsilon$ SVM regression with ϵ-insensitive loss function

Wider use of the kernel trick

- **Angle between vectors:**
  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:

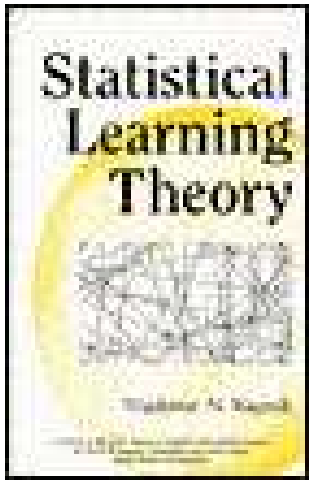
  $$\|\varphi(x) - \varphi(z)\|_2^2 = K(x,x) + K(z,z) - 2K(x,z)$$

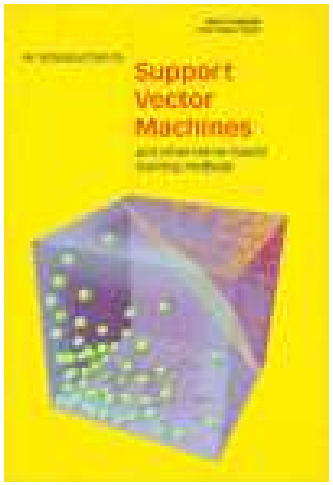
Books, software, papers ...

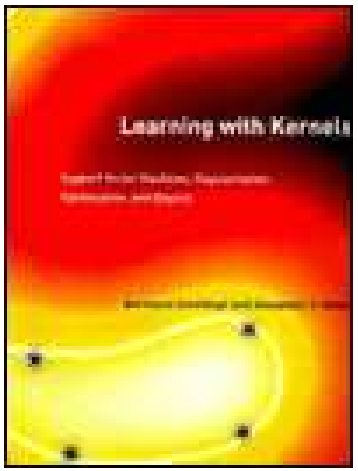
Websites:

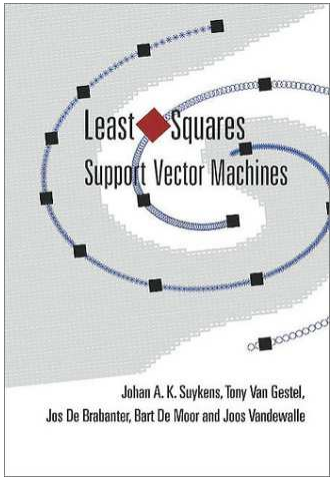
www.kernel-machines.org
www.esat.kuleuven.ac.be/sista/lssvmlab/

Books:









Least Squares Support Vector Machines

- RR

$$\min_{w,b,e} w^T w + \gamma \sum_i e_i^2 \quad \text{s.t.} \quad y_i = w^T x_i + b + e_i, \quad \forall i$$

- FDA

$$\min_{w,b,e} w^T w + \gamma \sum_i e_i^2 \quad \text{s.t.} \quad y_i (w^T x_i + b) = 1 - e_i, \quad \forall i$$

- PCA

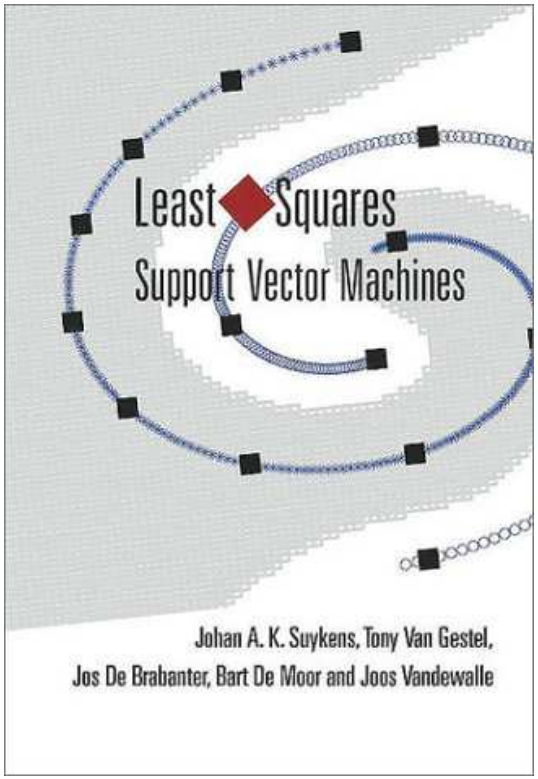
$$\min_{w,b,e} w^T w - \gamma \sum_i e_i^2 \quad \text{s.t.} \quad e_i = w^T x_i + b, \quad \forall i$$

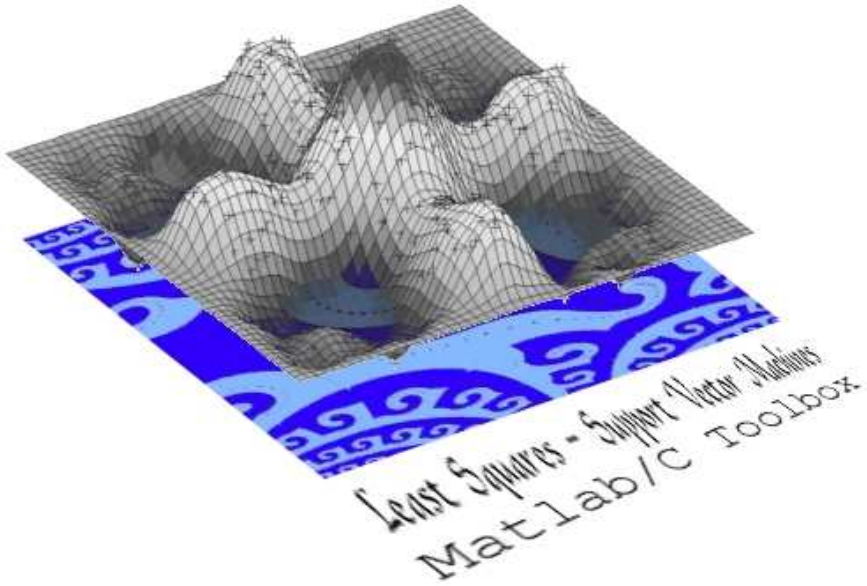
- CCA/PLS

$$\min_{w,v,b,d,e,r} w^T w + v^T v + \nu_1 \sum_i e_i^2 + \nu_2 \sum_i r_i^2 - \gamma \sum_i e_i r_i \text{ s.t. } \begin{cases} e_i &= w^T x_i + b \\ r_i &= v^T y_i + d \end{cases}$$

Primal problem in w, b, e and dual problem in Lagrange multipliers
Kernelize via kernel trick with use of feature map in primal problem

Least Squares Support Vector Machines





J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle,
Least Squares Support Vector Machines, World Scientific, Singapore, 2002
http://www.esat.kuleuven.ac.be/sista/lssvmlab/

LS-SVM classifier (1)

- Modifications w.r.t. standard SVM classifier [Suykens, 1999]:
  - Use *target values* instead of threshold values in the constraints
  - Simplify the problem via *equality constraints* and *least squares*.

- Optimization problem:

$$\min_{w,b,e} \mathcal{J}(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{N} e_i^2 \quad \text{s.t.} \quad y_i \left[w^T \varphi(x_i) + b \right] = 1 - e_i, \ \forall i$$

- Lagrangian:

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

  with Lagrange multipliers α_i (support values).

LS-SVM classifier (2)

- Conditions for optimality:

$$
\begin{cases}
\frac{\partial \mathcal{L}}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{i=1}^{N} \alpha_i y_i \varphi(x_i) \\
\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \\
\frac{\partial \mathcal{L}}{\partial e_i} = 0 \quad \rightarrow \quad \alpha_i = \gamma e_i, \qquad\qquad i = 1, ..., N \\
\frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \quad \rightarrow \quad y_i [w^T \varphi(x_i) + b] - 1 + e_i = 0, \quad i = 1, ..., N
\end{cases}
$$

- Dual problem (after elimination of w, e)

$$
\left[\begin{array}{c|c} 0 & y^T \\ \hline y & \Omega + I/\gamma \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_v \end{array}\right]
$$

where $\Omega_{ij} = y_i y_j \, \varphi(x_i)^T \varphi(x_j) = y_i y_j \, K(x_i, x_j)$ for $i, j = 1, ..., N$ and $y = [y_1; ...; y_N]$, $1_v = [1; ...; 1]$.

Benchmarking LS-SVM classifiers (1)

LS-SVM classifiers perform very well on 20 UCI benchmark data sets (10 binary, 10 multiclass) in comparison with many other methods.

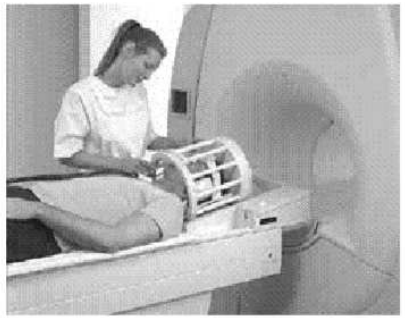
	bal	cmc	ims	iri	led	thy	usp	veh	wav	win
N_{CV}	416	982	1540	100	2000	4800	6000	564	2400	118
N_{test}	209	491	770	50	1000	2400	3298	282	1200	60
N	625	1473	2310	150	3000	7200	9298	846	3600	178
n_{num}	4	2	18	4	0	6	256	18	19	13
n_{cat}	0	7	0	0	7	15	0	0	0	0
n	4	9	18	4	7	21	256	18	19	13
M	3	3	7	3	10	3	10	4	3	3
$n_{y,\mathrm{MOC}}$	2	2	3	2	4	2	4	2	2	2
$n_{y,1\mathrm{vs}1}$	3	3	21	3	45	3	45	6	2	3

[Van Gestel et al., 2004]

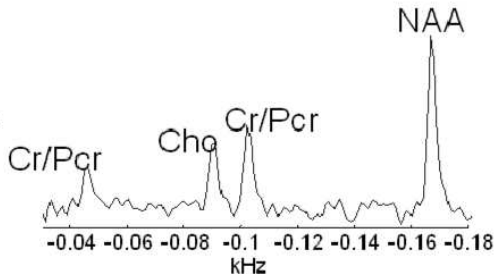
Benchmarking LS-SVM classifiers (2)

	acr	bld	gcr	hea	ion	pid	snr	ttt	wbc	adu	AA	AR	P_{ST}
N_{test}	230	115	334	90	117	256	70	320	228	12222			
n	14	6	20	13	33	8	60	9	9	14			
RBF LS-SVM	**87.0**(2.1)	**70.2**(4.1)	**76.3**(1.4)	**84.7**(4.8)	**96.0**(2.1)	**76.8**(1.7)	*73.1*(4.2)	*99.0*(0.3)	*96.4*(1.0)	*84.7*(0.3)	**84.4**	**3.5**	**0.727**
RBF LS-SVM$_{\text{F}}$	**86.4**(1.9)	*65.1*(2.9)	*70.8*(2.4)	83.2(5.0)	*93.4*(2.7)	*72.9*(2.0)	*73.6*(4.6)	*97.9*(0.7)	96.8(0.7)	*77.6*(1.3)	*81.8*	8.8	**0.109**
Lin LS-SVM	**86.8**(2.2)	*65.6*(3.2)	75.4(2.3)	**84.9**(4.5)	*87.9*(2.0)	76.8(1.8)	*72.6*(3.7)	*66.8*(3.9)	*95.8*(1.0)	*81.8*(0.3)	79.4	7.7	**0.109**
Lin LS-SVM$_{\text{F}}$	**86.5**(2.1)	*61.8*(3.3)	*68.6*(2.3)	*82.8*(4.4)	*85.0*(3.5)	*73.1*(1.7)	73.3(3.4)	*57.6*(1.9)	**96.9**(0.7)	*71.3*(0.3)	*75.7*	12.1	**0.109**
Pol LS-SVM	**86.5**(2.2)	**70.4**(3.7)	**76.3**(1.4)	**83.7**(3.9)	*91.0*(2.5)	**77.0**(1.8)	**76.9**(4.7)	**99.5**(0.5)	*96.4*(0.9)	*84.6*(0.3)	**84.2**	**4.1**	**0.727**
Pol LS-SVM$_{\text{F}}$	**86.6**(2.2)	*65.3*(2.9)	*70.3*(2.3)	*82.4*(4.6)	*91.7*(2.6)	*73.0*(1.8)	**77.3**(2.6)	*98.1*(0.8)	**96.9**(0.7)	*77.9*(0.2)	82.0	**8.2**	**0.344**
RBF SVM	*86.3*(1.8)	**70.4**(3.2)	**75.9**(1.4)	**84.7**(4.8)	95.4(1.7)	**77.3**(2.2)	**75.0**(6.6)	*98.6*(0.5)	*96.4*(1.0)	*84.4*(0.3)	**84.4**	**4.0**	**1.000**
Lin SVM	**86.7**(2.4)	*67.7*(2.6)	75.4(1.7)	*83.2*(4.2)	*87.1*(3.4)	77.0(2.4)	*74.1*(4.2)	*66.2*(3.6)	*96.3*(1.0)	*83.9*(0.2)	*79.8*	**7.5**	0.021
LDA	*85.9*(2.2)	*65.4*(3.2)	**75.9**(2.0)	**83.9**(4.3)	*87.1*(2.3)	76.7(2.0)	*67.9*(4.9)	*68.0*(3.0)	*95.6*(1.1)	*82.2*(0.3)	*78.9*	*9.6*	*0.004*
QDA	*80.1*(1.9)	*62.2*(3.6)	*72.5*(1.4)	*78.4*(4.0)	*90.6*(2.2)	*74.2*(3.3)	*53.6*(7.4)	*75.1*(4.0)	*94.5*(0.6)	*80.7*(0.3)	*76.2*	*12.6*	*0.002*
Logit	**86.8**(2.4)	*66.3*(3.1)	**76.3**(2.1)	*82.9*(4.0)	*86.2*(3.5)	**77.2**(1.8)	*68.4*(5.2)	*68.3*(2.9)	*96.1*(1.0)	*83.7*(0.2)	79.2	7.8	**0.109**
C4.5	*85.5*(2.1)	*63.1*(3.8)	*71.4*(2.0)	*78.0*(4.2)	*90.6*(2.2)	*73.5*(3.0)	*72.1*(2.5)	*84.2*(1.6)	*94.7*(1.0)	**85.6**(0.3)	*79.9*	*10.2*	0.021
oneR	*85.4*(2.1)	*56.3*(4.4)	*66.0*(3.0)	*71.7*(3.6)	*83.6*(4.8)	*71.3*(2.7)	*62.6*(5.5)	*70.7*(1.5)	*91.8*(1.4)	*80.4*(0.3)	*74.0*	*15.5*	*0.002*
IB1	*81.1*(1.9)	*61.3*(6.2)	*69.3*(2.6)	*74.3*(4.2)	*87.2*(2.8)	*69.6*(2.4)	**77.7**(4.4)	*82.3*(3.3)	*95.3*(1.1)	*78.9*(0.2)	*77.7*	12.5	0.021
IB10	**86.4**(1.3)	*60.5*(4.4)	*72.6*(1.7)	*80.0*(4.3)	*85.9*(2.5)	*73.6*(2.4)	*69.4*(4.3)	*94.8*(2.0)	*96.4*(1.2)	*82.7*(0.3)	*80.2*	*10.4*	0.039
NB$_{\text{k}}$	*81.4*(1.9)	*63.7*(4.5)	*74.7*(2.1)	83.9(4.5)	*92.1*(2.5)	75.5(1.7)	*71.6*(3.5)	*71.7*(3.1)	**97.1**(0.9)	*84.8*(0.2)	*79.7*	**7.3**	**0.109**
NB$_{\text{n}}$	*76.9*(1.7)	*56.0*(6.9)	74.6(2.8)	**83.8**(4.5)	*82.8*(3.8)	75.1(2.1)	*66.6*(3.2)	*71.7*(3.1)	*95.5*(0.5)	*82.7*(0.2)	*76.6*	*12.3*	*0.002*
Maj. Rule	*56.2*(2.0)	*56.5*(3.1)	*69.7*(2.3)	*56.3*(3.8)	*64.4*(2.9)	*66.8*(2.1)	*54.4*(4.7)	*66.2*(3.6)	*66.2*(2.4)	*75.3*(0.3)	*63.2*	*17.1*	*0.002*

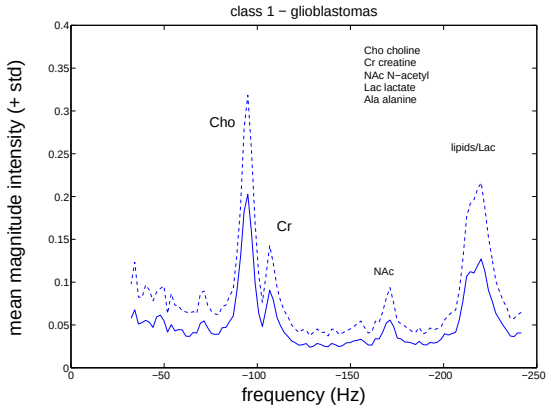
Classification of brain tumors from MRS data (1)



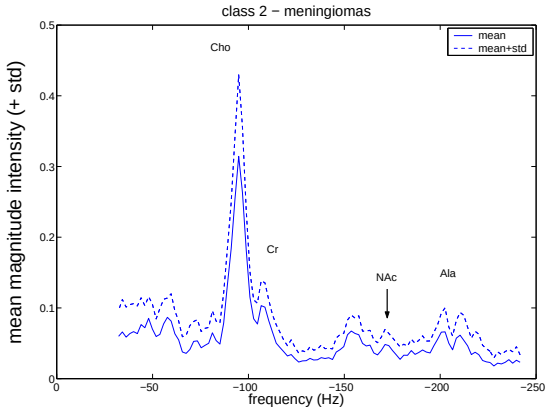
MR scanner



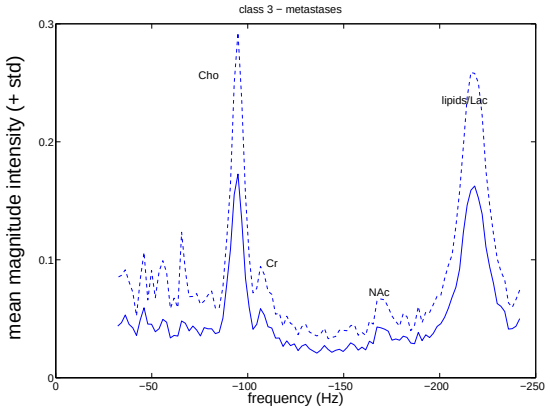
Feature Vector



Class 1



Class 2



Class 3

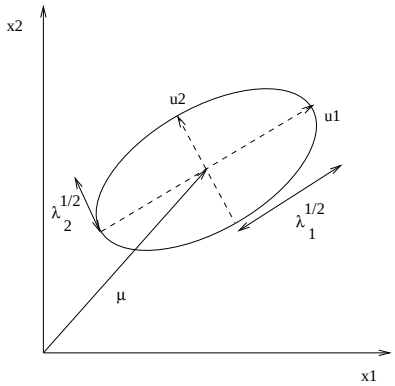
Classification of brain tumors from MRS data (2)

	$\overline{e_{train}} \pm std(e_{train})$	mean % correct	$\overline{e_{test}} \pm std(e_{test})$	mean % correct
RBF12	0.0800 $\pm$ 0.2727	99.8621	2.8500 $\pm$ 1.9968	90.1724
	0.0600 $\pm$ 0.2387	99.8966	2.6800 $\pm$ 1.6198	90.7586
RBF13	1.6700 $\pm$ 1.1106	96.7255	8.1200 $\pm$ 1.2814	67.5200
	1.7900 $\pm$ 1.0473	96.4902	7.7900 $\pm$ 1.2815	68.8400
RBF23	0 $\pm$ 0	100	2.0000 $\pm$ 1.1976	90.4762
	0 $\pm$ 0	100	2.0200 $\pm$ 1.2632	90.3810
Lin12, γ=1	6.2000 $\pm$ 1.3333	89.3100	3.8900 $\pm$ 1.8472	86.586
	6.1300 $\pm$ 1.4679	89.4310	3.6800 $\pm$ 1.7746	87.3103
Lin13, γ=1	15.6400 $\pm$ 1.7952	69.333	7.6800 $\pm$ 0.8863	69.280
	15.3700 $\pm$ 1.8127	69.8627	7.9200 $\pm$ 1.0316	68.3200
Lin23, γ=1	4.0100 $\pm$ 1.3219	90.452	3.4400 $\pm$ 1.2253	83.619
	4.0000 $\pm$ 1.1976	90.4762	2.9600 $\pm$ 1.3478	85.9048

Comparison of LS-SVM classification with LOO using RBF and linear kernel, with additional bias term correction ($N_1 = 50, N_2 = 37, N_3 = 26$).

[Devos et al., 2004]

Classical PCA analysis



- Given zero mean data $\{x_i\}_{i=1}^N$ with $x \in \mathbb{R}^n$
- Find projected variables $w^T x_i$ with maximal variance

$$\max_w \mathrm{Var}(w^T x) = \mathrm{Cov}(w^T x, w^T x) \simeq \frac{1}{N}\sum_{i=1}^N (w^T x_i)^2$$
$$= w^T C\, w$$

where $C = \frac{1}{N}\sum_{i=1}^N x_i x_i^T$. Consider additional constraint $w^T w = 1$.

- Resulting eigenvalue problem:

$$Cw = \lambda w$$

with $C = C^T \geq 0$, obtained from the Lagrangian $\mathcal{L}(w;\lambda) = \frac{1}{2}w^T C w - \lambda(w^T w - 1)$ and setting $\partial\mathcal{L}/\partial w = 0$, $\partial\mathcal{L}/\partial\lambda = 0$.

SVM formulation to PCA (1)

- Primal problem: [Suykens et al., 2003]

$$\max_{w,e} \mathcal{J}(w,e) = \gamma \frac{1}{2} \sum_{i=1}^{N} e_i^2 - \frac{1}{2} w^T w \quad \text{s.t.} \quad e_i = w^T x_i, \quad i = 1, ..., N$$

- Lagrangian $\mathcal{L}(w, e; \alpha) = \gamma \frac{1}{2} \sum_{i=1}^{N} e_i^2 - \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left(e_i - w^T x_i \right)$

- Conditions for optimality

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow \quad w = \sum_{i=1}^{N} \alpha_i x_i \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 & \rightarrow \quad \alpha_i = \gamma e_i, & i = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 & \rightarrow \quad e_i - w^T x_i = 0, & i = 1, ..., N \end{cases}$$

SVM formulation to PCA (2)

- By elimination of e, w one obtains the eigenvalue problem

$$\begin{bmatrix} x_1^T x_1 & \dots & x_1^T x_N \\ \vdots & & \vdots \\ x_N^T x_1 & \dots & x_N^T x_N \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} = \lambda \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}$$

  as the dual problem (with eigenvalues $\lambda = 1/\gamma$).

- The score variables become $z(x) = w^T x = \sum_{j=1}^N \alpha_j x_j^T x$.
  The optimal solution corresponding to largest eigenvalue has

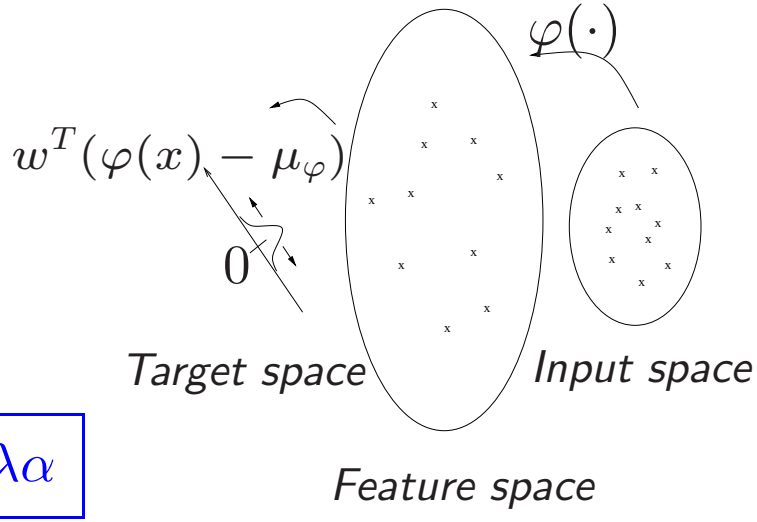
$$\sum_{i=1}^N (w^T x_i)^2 = \sum_{i=1}^N e_i^2 = \sum_{i=1}^N \frac{1}{\gamma^2} \alpha_i^2 = \lambda_{max}^2$$

  where $\sum_{i=1}^N \alpha_i^2 = 1$ for the normalized eigenvector.

Kernel PCA: SVM formulation

- Primal problem:

$$\max_{w,b,e} \mathcal{J}(w,e) = \gamma \frac{1}{2}\sum_{i=1}^{N} e_i^2 - \frac{1}{2} w^T w$$
$$\text{s.t. } e_i = w^T \varphi(x_i) + b, \; i = 1, ..., N.$$



- Dual problem = **kernel PCA:** $\boxed{\Omega_c \alpha = \lambda \alpha}$

  with centered kernel matrix $\Omega_{c,ij} = (\varphi(x_i) - \hat{\mu}_\varphi)^T (\varphi(x_j) - \hat{\mu}_\varphi), \; \forall i,j$
  [Schölkopf et al., 1998]

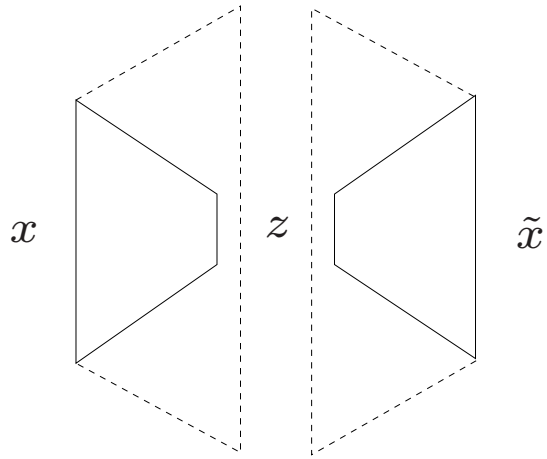
- Score variables (note: $\hat{\mu}_\varphi = (1/N)\sum_{i=1}^{N} \varphi(x_i)$)

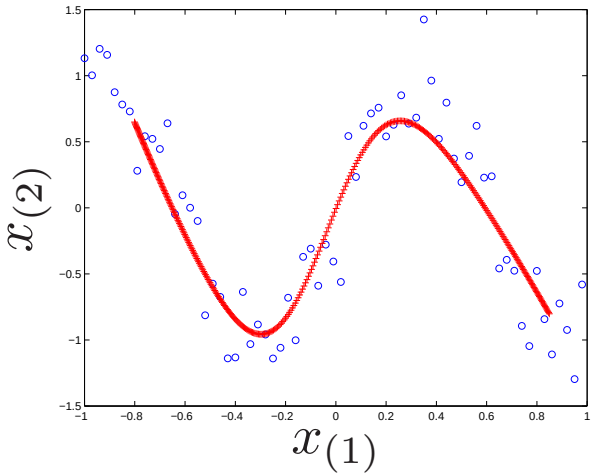
$$\begin{aligned} z(x) &= w^T(\varphi(x) - \hat{\mu}_\varphi) \\ &= \sum_{j=1}^{N} \alpha_j (K(x_j, x) - \frac{1}{N}\sum_{r=1}^{N} K(x_r, x) - \frac{1}{N}\sum_{r=1}^{N} K(x_r, x_j) \\ &\quad + \frac{1}{N^2}\sum_{r=1}^{N}\sum_{s=1}^{N} K(x_r, x_s)) \end{aligned}$$

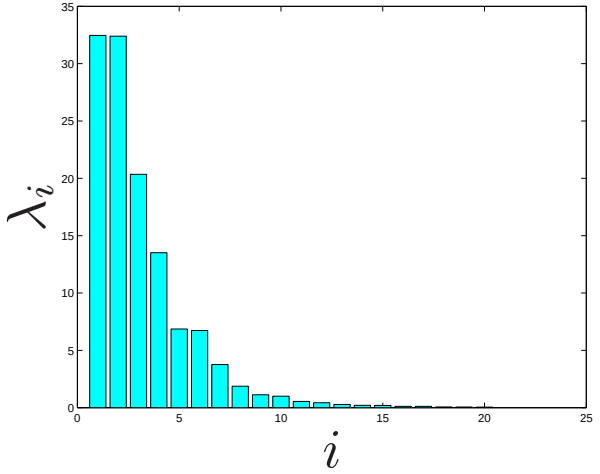
Kernel PCA: reconstruction problem

Reconstruction error:

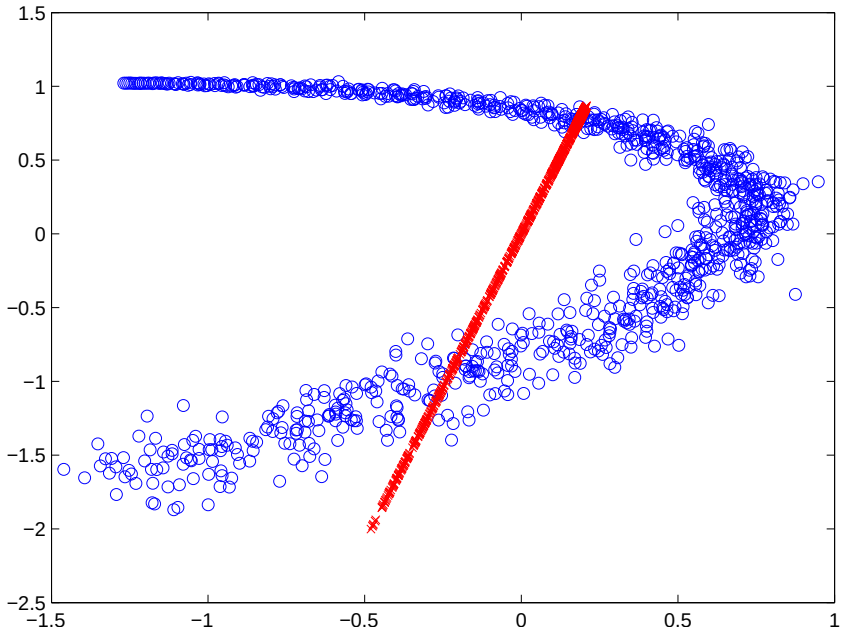
$$\min \sum_{i=1}^{N} \|x_i - \tilde{x}_i\|_2^2$$



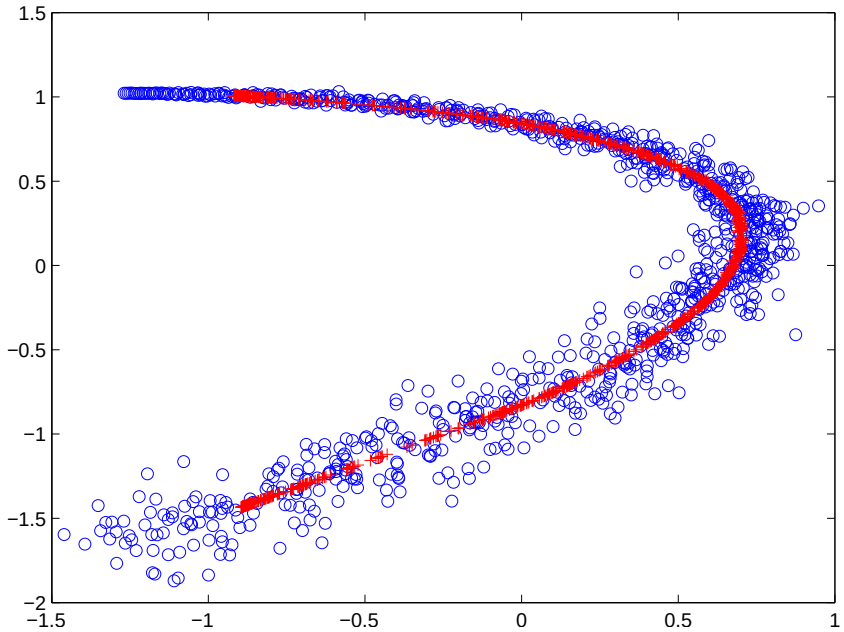




Reconstruction: linear versus nonlinear



linear PCA

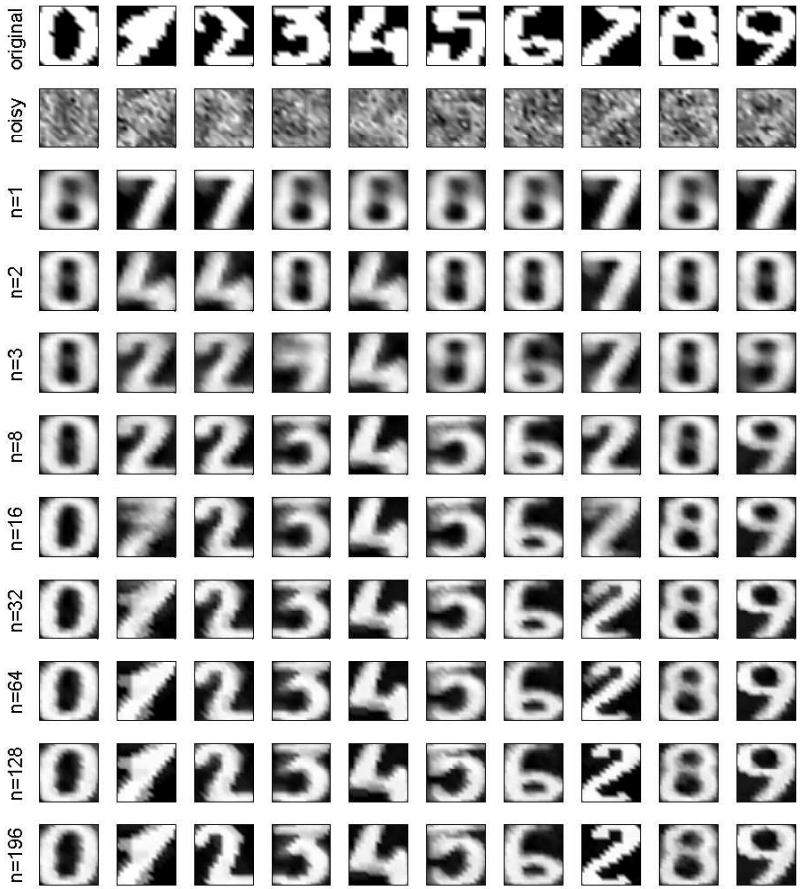


kernel PCA (RBF kernel)

[Schölkopf et al., 1998]

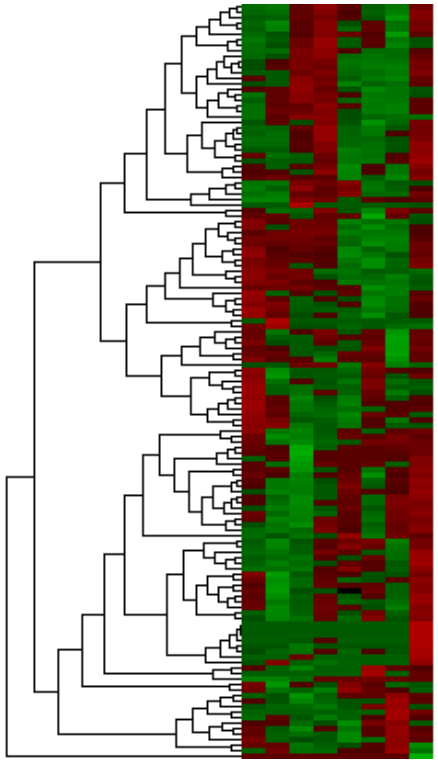
Application: denoising of images

linear PCA



kernel PCA (RBF kernel)

Microarray data analysis



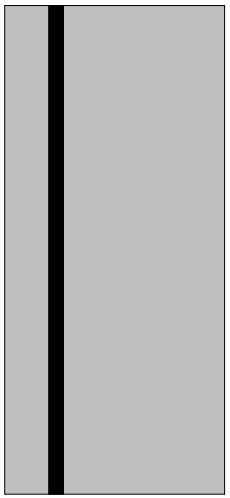
FDA
LS-SVM classifier (linear, RBF)
Kernel PCA + FDA
(unsupervised selection of PCs)
(supervised selection of PCs)

Use regularization for linear classifiers

Systematic benchmarking study in [Pochet et al., 2004]
Webservice: http://www.esat.kuleuven.ac.be/MACBETH/

Primal versus dual problem

Example 1: microarray data (10,000 genes & 50 training data)



Classifier model:

$\text{sign}(w^T x + b)$ (primal)

$\text{sign}(\sum_i \alpha_i y_i x_i^T x + b)$ (dual)

linear FDA primal: $w \in \mathbb{R}^{10,000}$ (but only 50 training data!)

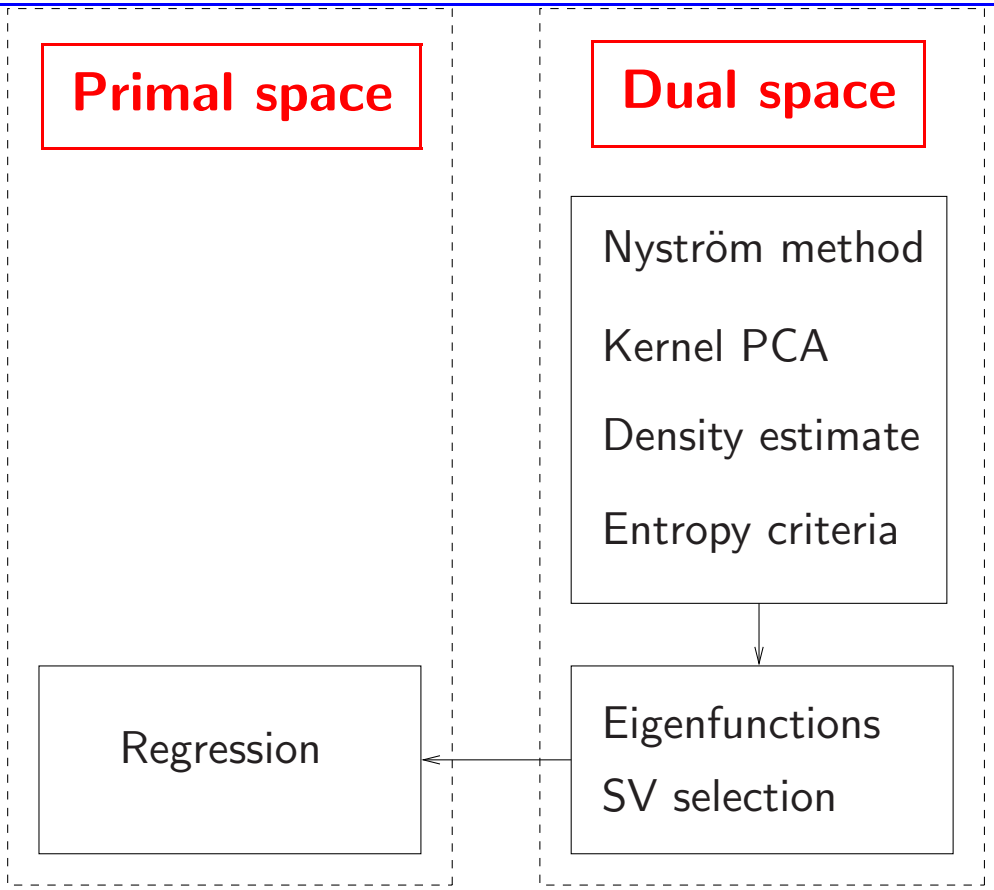
linear FDA dual: $\alpha \in \mathbb{R}^{50}$

Example 2: datamining problem (1,000,000 training data & 20 inputs)

linear FDA primal: $w \in \mathbb{R}^{20}$

linear FDA dual: $\alpha \in \mathbb{R}^{1,000,000}$ (kernel matrix: $1,000,000 \times 1,000,000$!)

Fixed-size LS-SVM: primal-dual kernel machines



Modelling in view of primal-dual representations

Link Nyström approximation (GP) - kernel PCA - density estimation

[Suykens et al., 2002]: primal space estimation, sparse, large scale

Nyström method (Gaussian processes)

[Williams, 2001 *Nyström method*; Girolami, 2002 *KPCA, density estimation*]

- *"big"* matrix: $\Omega_{(N,N)} \in \mathbb{R}^{N \times N}$, *"small"* matrix: $\Omega_{(M,M)} \in \mathbb{R}^{M \times M}$ (based on random subsample, in practice often $M \ll N$)
- Eigenvalue decompositions: $\Omega_{(N,N)}\, \tilde{U} = \tilde{U}\, \tilde{\Lambda}$ and $\Omega_{(M,M)}\, \overline{U} = \overline{U}\, \overline{\Lambda}$
- Relation to eigenvalues and eigenfunctions of the integral equation

$$\int K(x, x')\phi_i(x)p(x)dx = \lambda_i \phi_i(x')$$

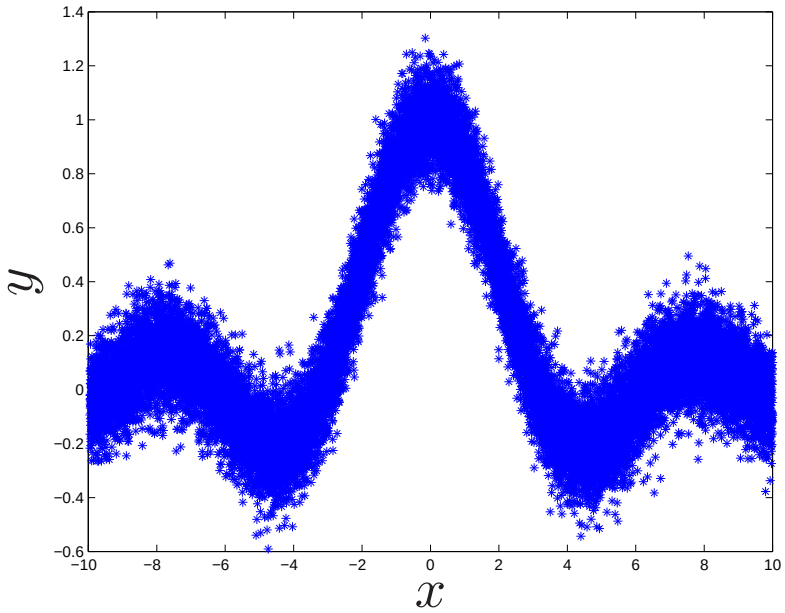
with

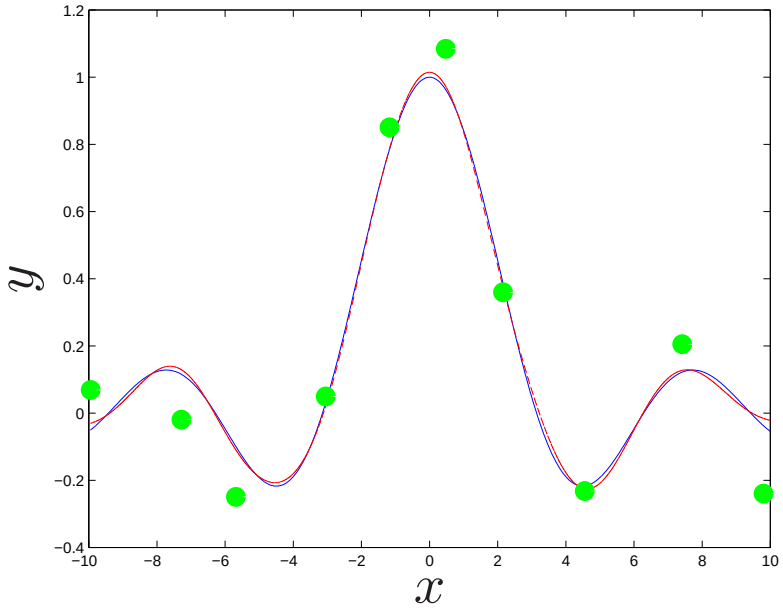
$$\hat{\lambda}_i = \frac{1}{M}\overline{\lambda}_i, \quad \hat{\phi}_i(x_k) = \sqrt{M}\, \overline{u}_{ki}, \quad \hat{\phi}_i(x') = \frac{\sqrt{M}}{\overline{\lambda}_i} \sum_{k=1}^{M} \overline{u}_{ki} K(x_k, x')$$

Fixed-size LS-SVM: examples (1)

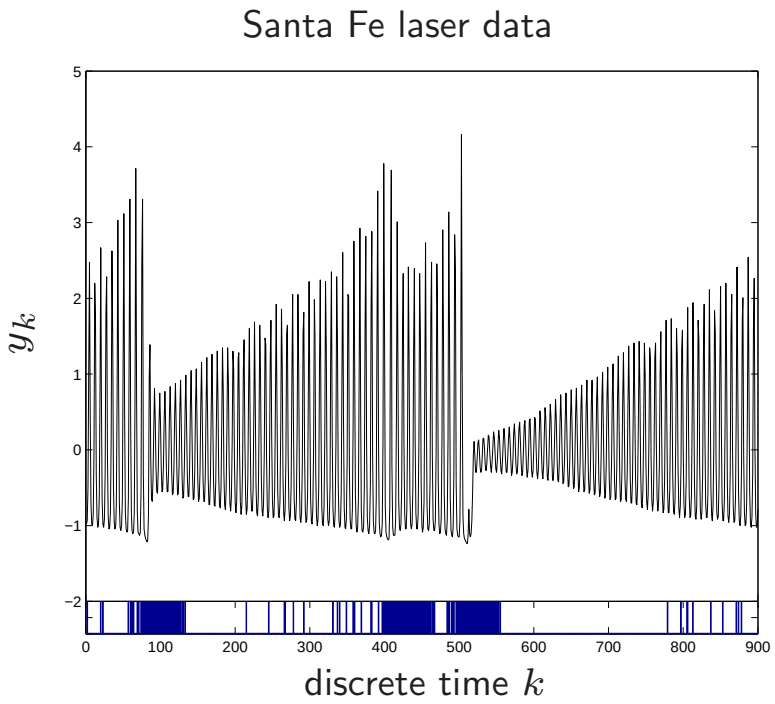
high dimensional inputs, large data sets, adaptive learning machines (using LS-SVMlab)

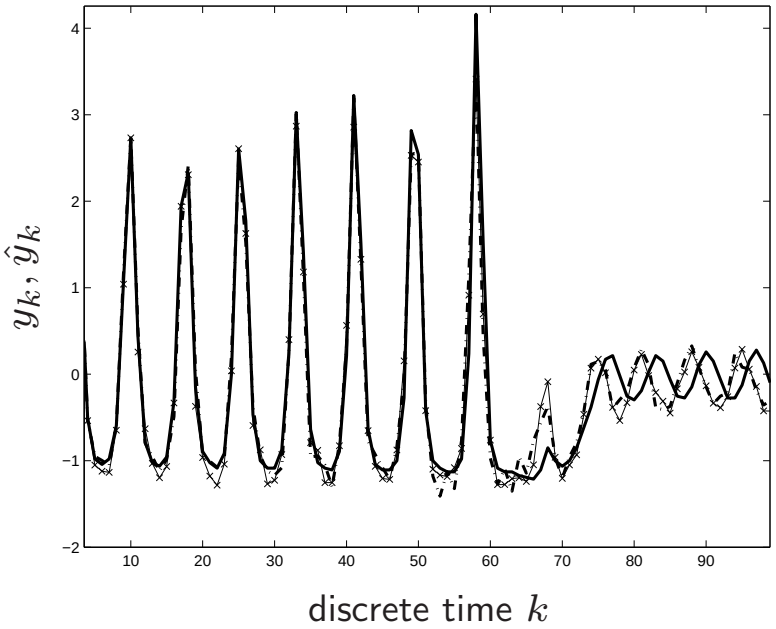
Sinc function (20.000 data, 10 SV)





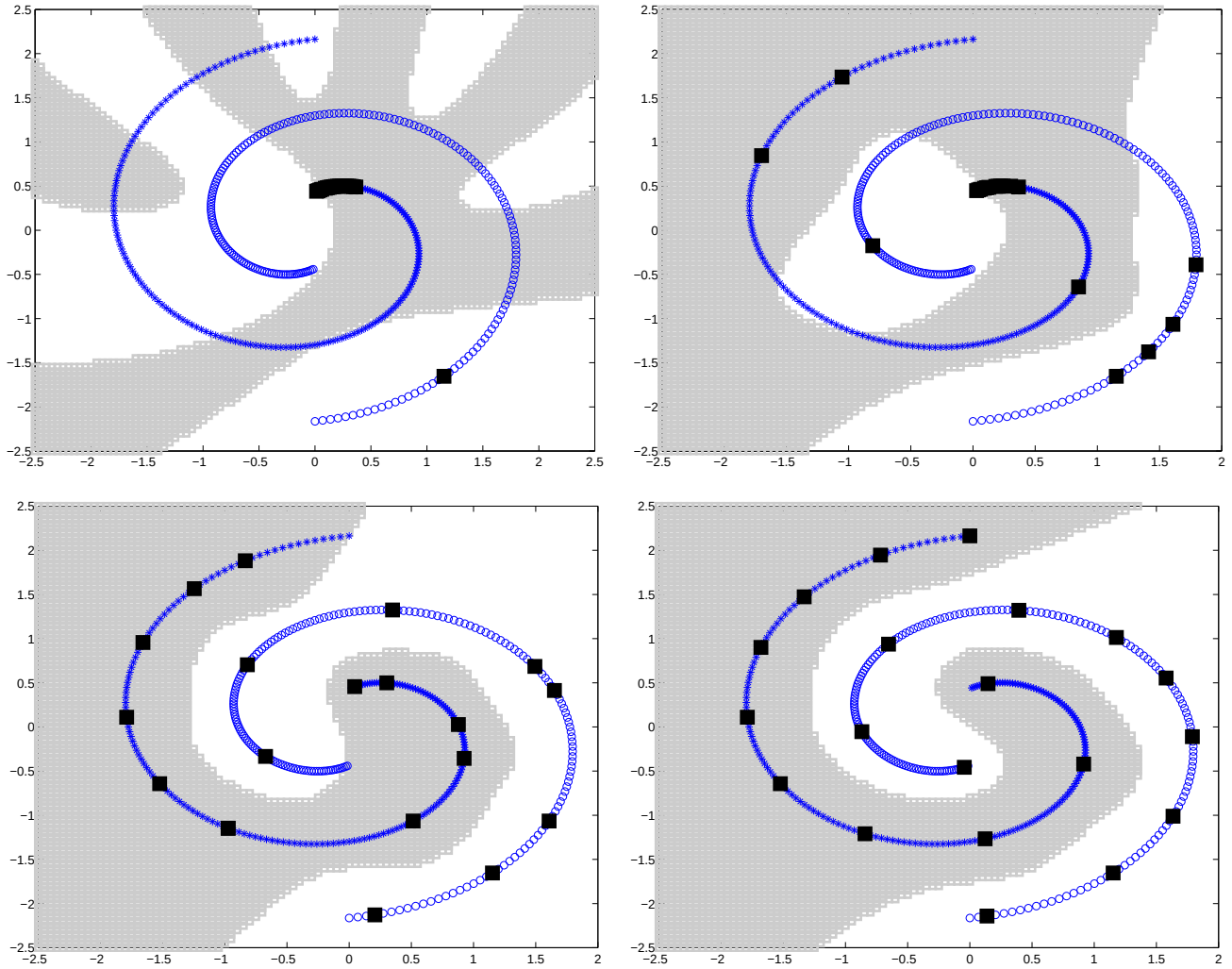
Fixed-size LS-SVM: examples (2)





Training: $\hat{y}_{k+1} = f(y_k, y_{k-1}, ..., y_{k-p})$
Iterative prediction: $\hat{y}_{k+1} = f(\hat{y}_k, \hat{y}_{k-1}, ..., \hat{y}_{k-p})$
(works well for p large, e.g. $p = 50$)

Fixed-size LS-SVM: examples (3)



Conclusions of Part I

- Support vector machines: learning and generalization in high dimensional input spaces
- Convex optimization, primal and dual problem and representations
- Kernel based learning
- "Kernelize" many classical statistical techniques
- Core problems and extensions via LS-SVMs
- Many successful applications in different areas

Research directions

- "Kernelizing" classical methods (FDA, PCA, CCA, ICA, subspace ...)
  Customizing kernels towards specific application areas

- Learning and generalization (model selection, CV, stability, ...)
  Robust statistics for kernel methods

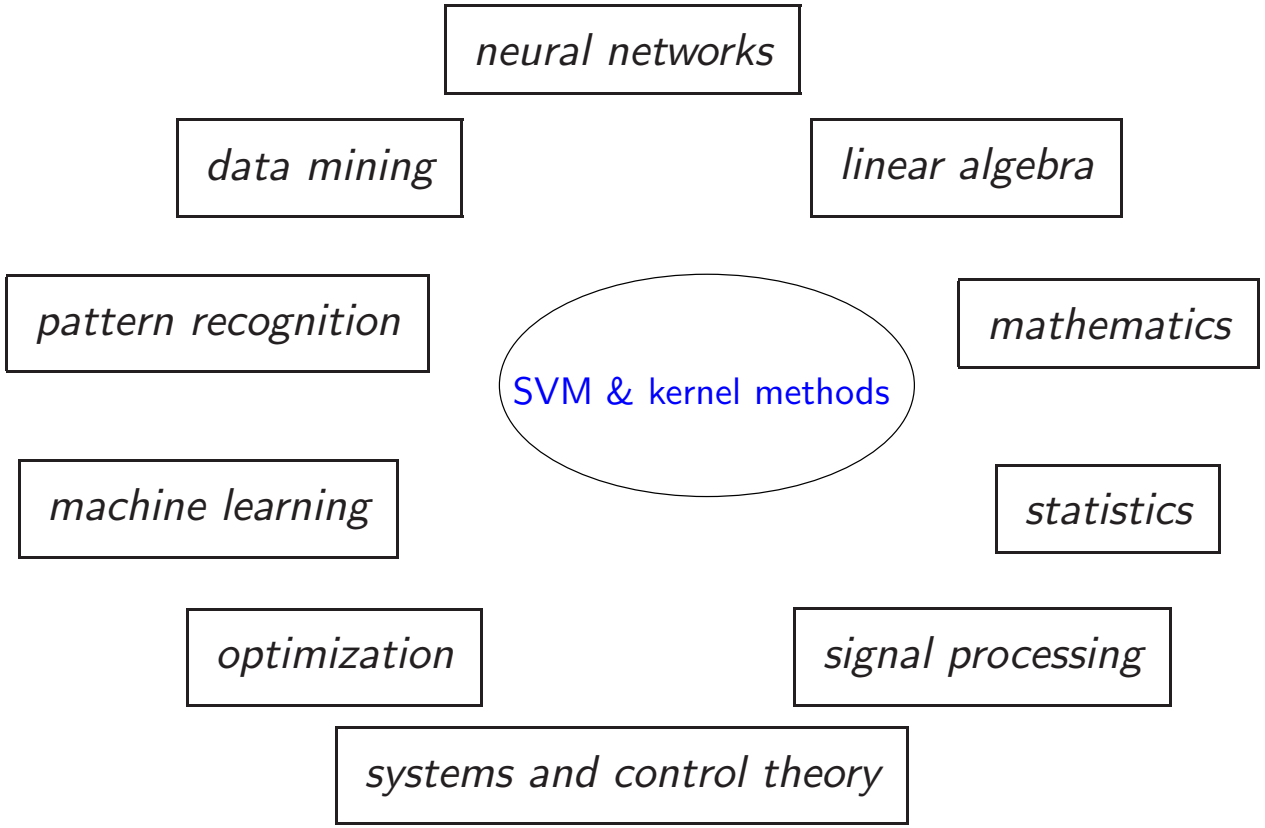
- Learning paradigms: (un)-supervised, transductive, reinforcement
  System aspects: static, dynamic, control problems, adaptive algorithms

- Convex optimization approaches, stable numerical algorithms

- Large data sets, high dimensional input data

- Generic models with many applications
  Incorporation of prior knowledge, hardware implementations

Interdisciplinary challenges



NATO Advanced Study Institute on Learning Theory and Practice

http://www.esat.kuleuven.ac.be/sista/natoasi/ltp2002.html

Contents - Part II: Advanced topics

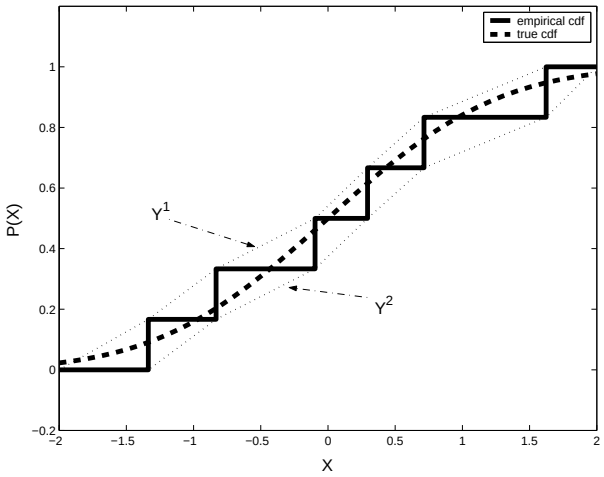
- Incorporation of structure and prior knowledge
- Kernels and graphical models
- Iterative weighting and robustness
- Bayesian inference and relevance determination
- Hierarchical kernel machines
- Input selection, stability, sparseness

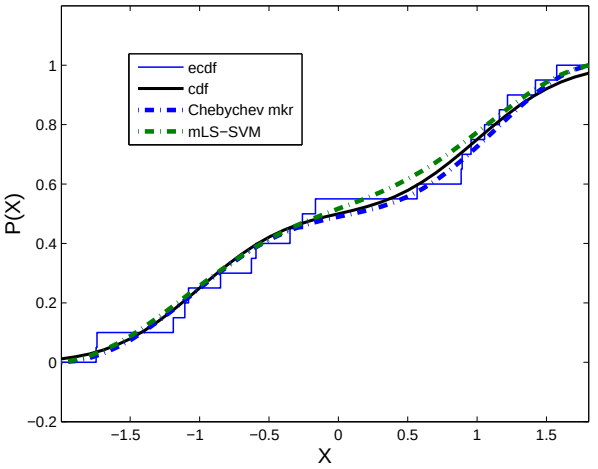
Incorporation of prior knowledge

- Support vector machine formulations allow to incorporate additional constraints that express prior knowledge about the problem, e.g. monoticity, symmetry, positivity, ...
- Example: LS-SVM regression with monoticity constraint

$$\min_{w,b,e} \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{N} e_i^2 \quad \text{s.t.} \quad \begin{cases} y_i = w^T \varphi(x_i) + b + e_i, & \forall i = 1, ..., N \\ w^T \varphi(x_i) \leq w^T \varphi(x_{i+1}), & \forall i = 1, ..., N-1 \end{cases}$$

- Application: estimation of cdf [Pelckmans et al., 2005]





Partially linear models

- Partially linear regression with regressors z^a, z^b [Espinoza et al., 2005]:

$$\min_{w,b,e,\beta} \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{N} e_i^2 \quad \text{s.t.} \quad y_i = \beta^T z_i^a + w^T \varphi(z_i^b) + b + e_i, \ \forall i$$

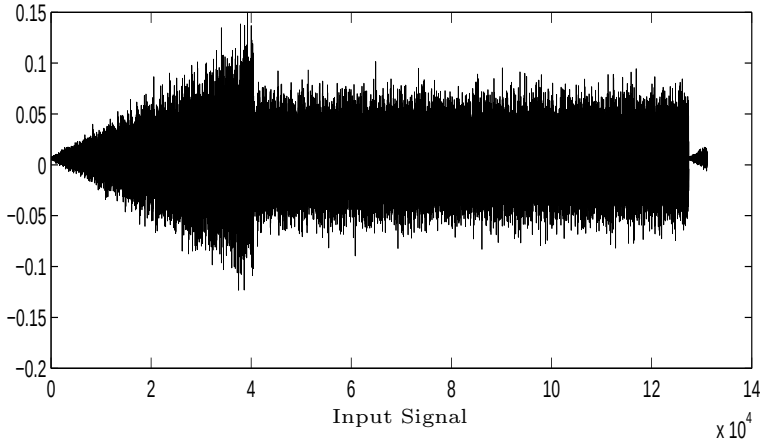
- Dual problem:

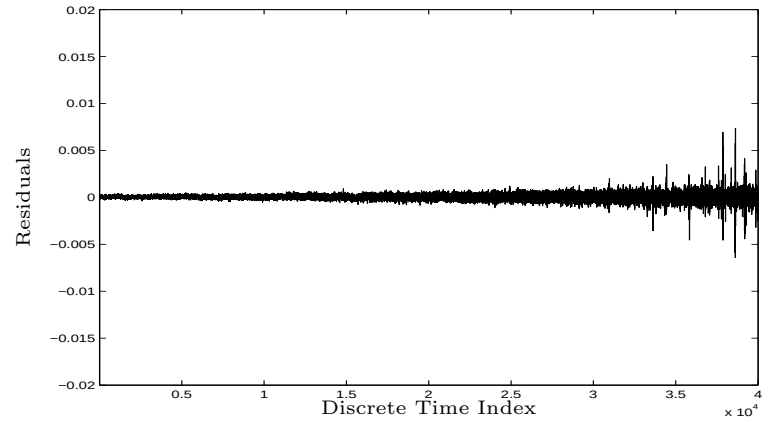
$$\begin{bmatrix} \Omega + I/\gamma & 1_N & Z \\ 1_N^T & 0 & 0 \\ z^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ b \\ \beta \end{bmatrix} = \begin{bmatrix} y \\ 0 \\ 0 \end{bmatrix}$$

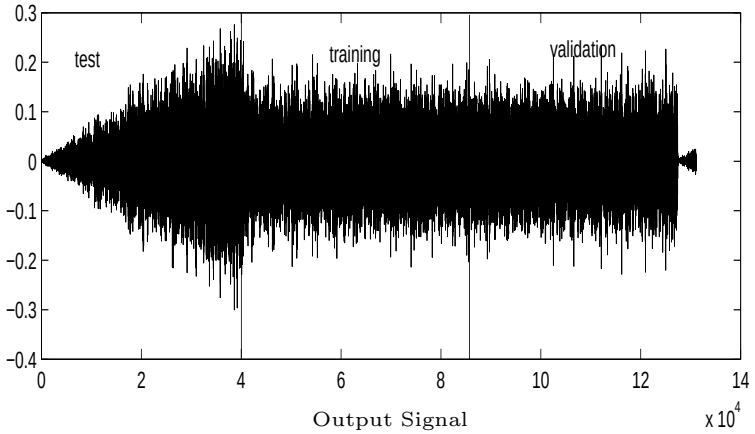
  with $\Omega_{ij} = K(z_i^b, z_j^b)$, $Z = [z_1^{aT}; ...; z_N^{a\,T}]$.

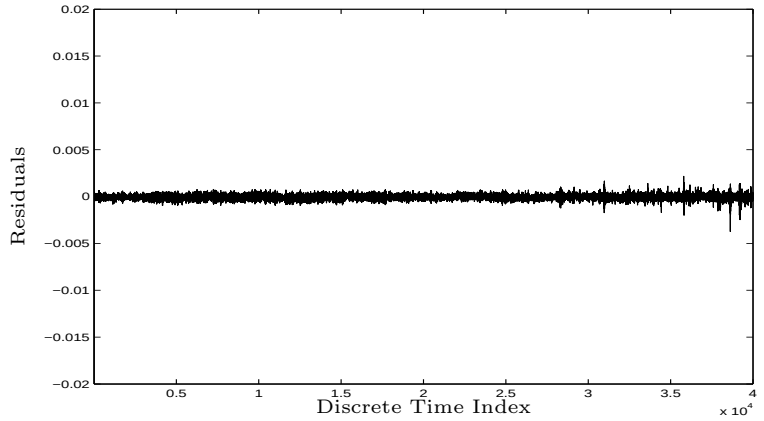
- Reduces the effective number of parameters.

Partially linear models: nonlinear system identification









Silver box benchmark study:
(top-right) full black-box, (bottom-right) partially linear

Kernels and graphical models (1)

- Probability product kernel [Jebara et al., 2004]

$$K(p, p') = \int_{\mathcal{X}} p(x)^{\rho} \, p'(x)^{\rho} dx$$

- Case $\rho = 1/2$: Bhattacharyya kernel

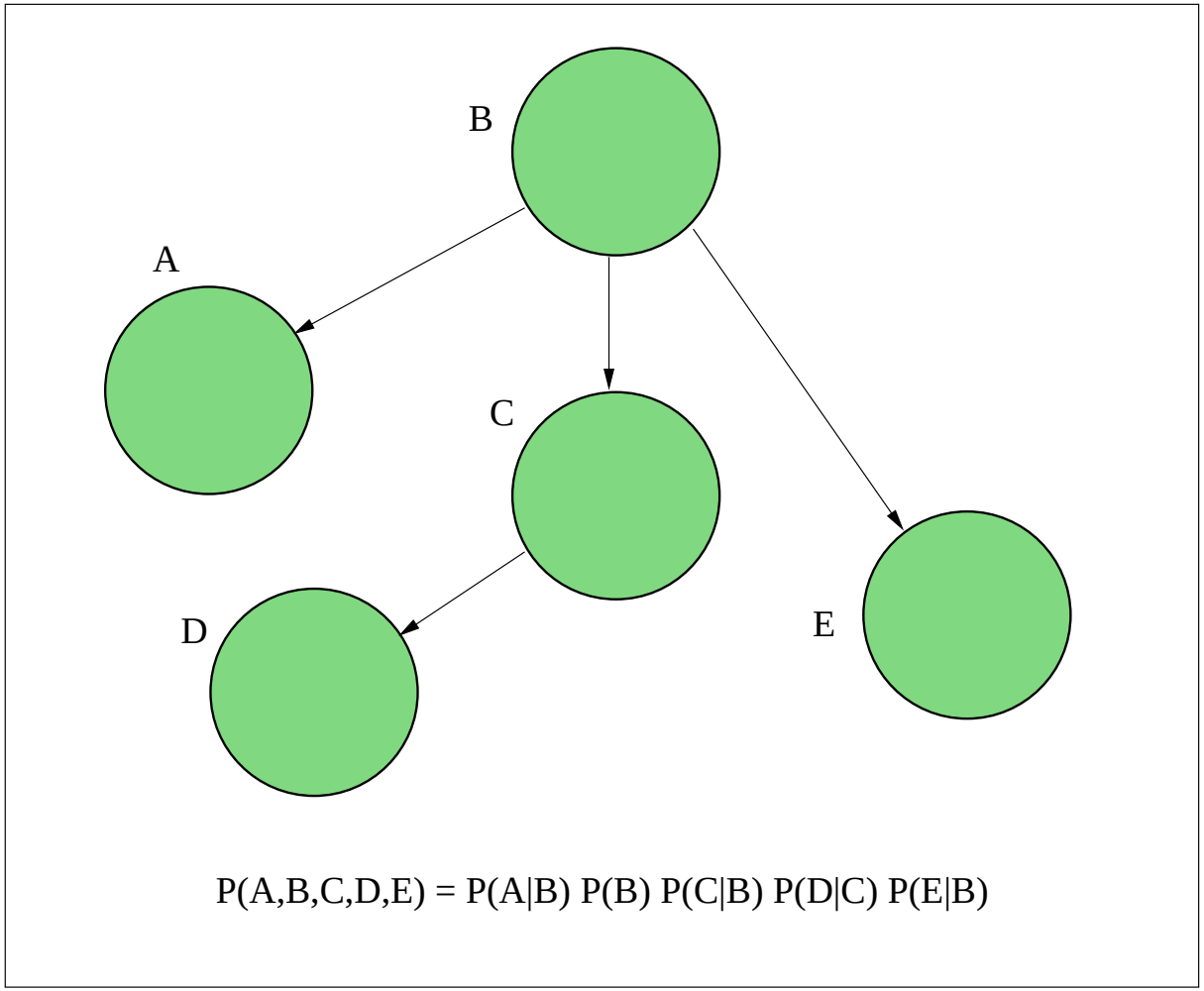
$$K(p, p') = \int_{\mathcal{X}} \sqrt{p(x)} \sqrt{p'(x)} dx$$

(related to Hellinger distance $H(p, p') = \frac{1}{2} \int_{\mathcal{X}} (\sqrt{p(x)} - \sqrt{p'(x)})^2 dx$ by $H(p, p') = \sqrt{2 - 2K(p, p')}$, which is a symmetric approximation to Kullback-Leibler divergence and a bound on it).

- Case $\rho = 1$: expected likelihood kernel

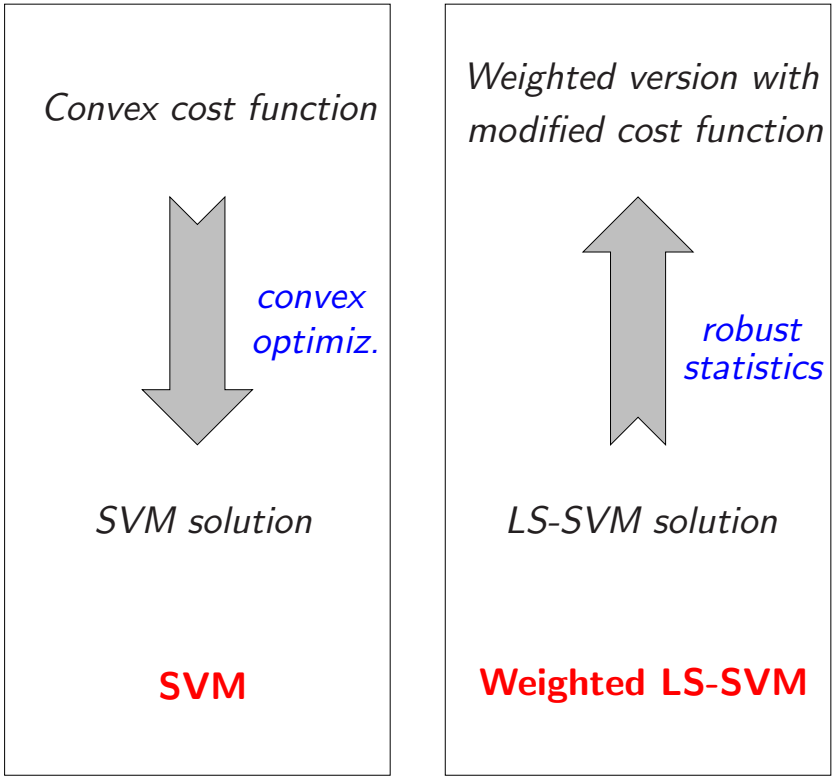
$$K(p, p') = \int_{\mathcal{X}} p(x) p'(x) dx = \mathbb{E}_p[p'(x)] = \mathbb{E}_{p'}[p(x)]$$

Kernels and graphical models (2)



Extract kernels from graphical models, Bayesian networks, HMMs

Robustness

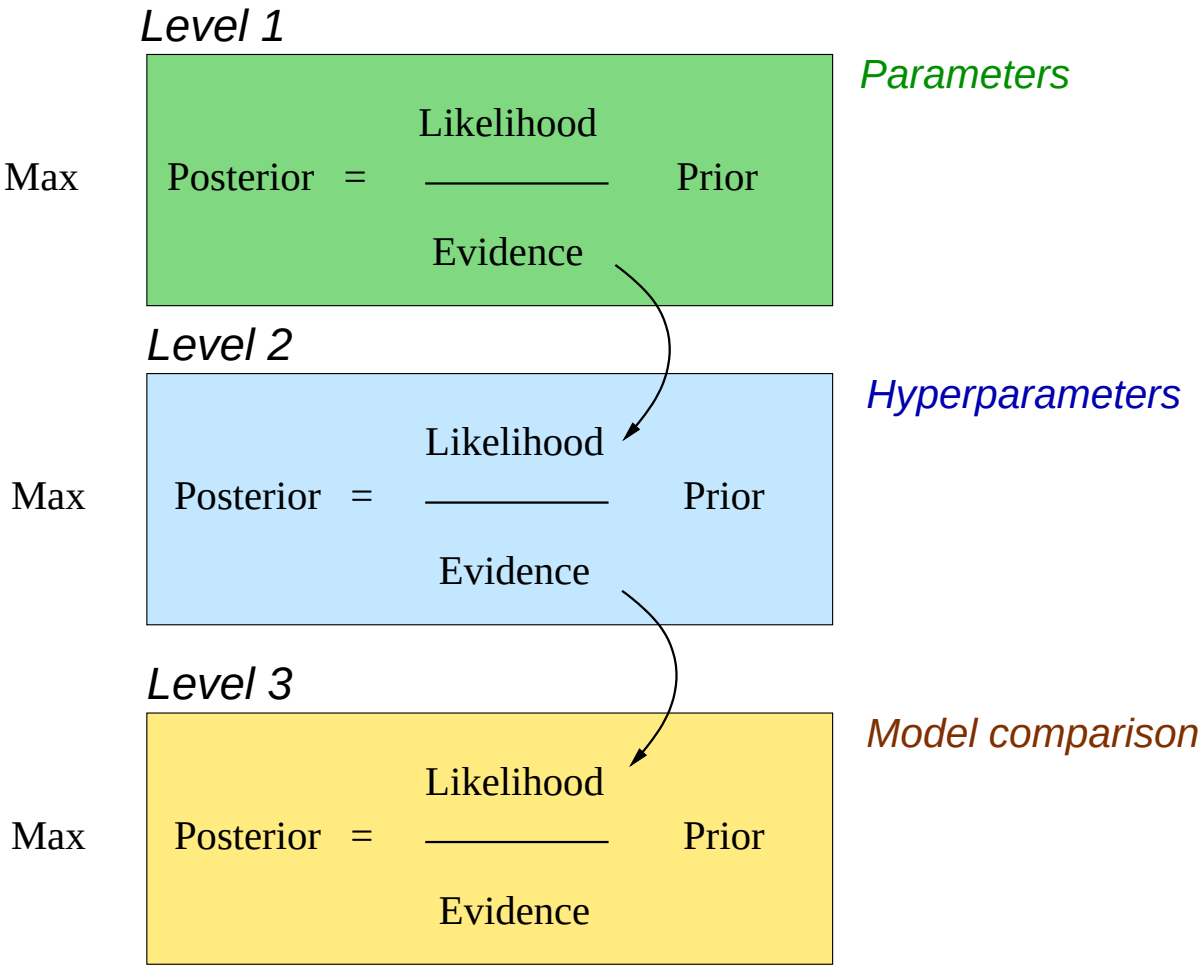


- Weighted LS-SVM: $\min_{w,b,e} \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{N} v_i e_i^2 \ \text{s.t.} \ y_i = w^T \varphi(x_i) + b + e_i, \ \forall i$

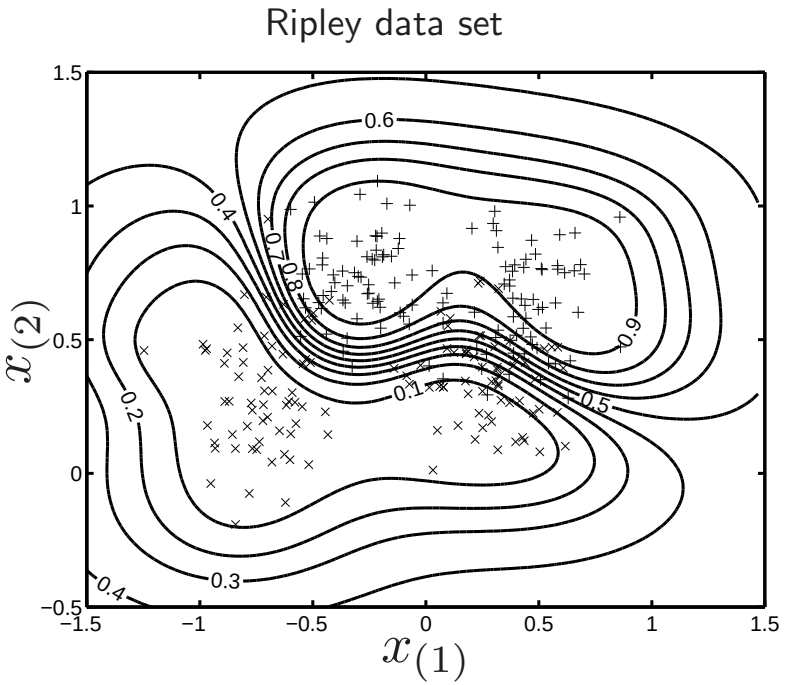
  with v_i determined from $\{e_i\}_{i=1}^N$ of unweighted LS-SVM [Suykens et al., 2002]

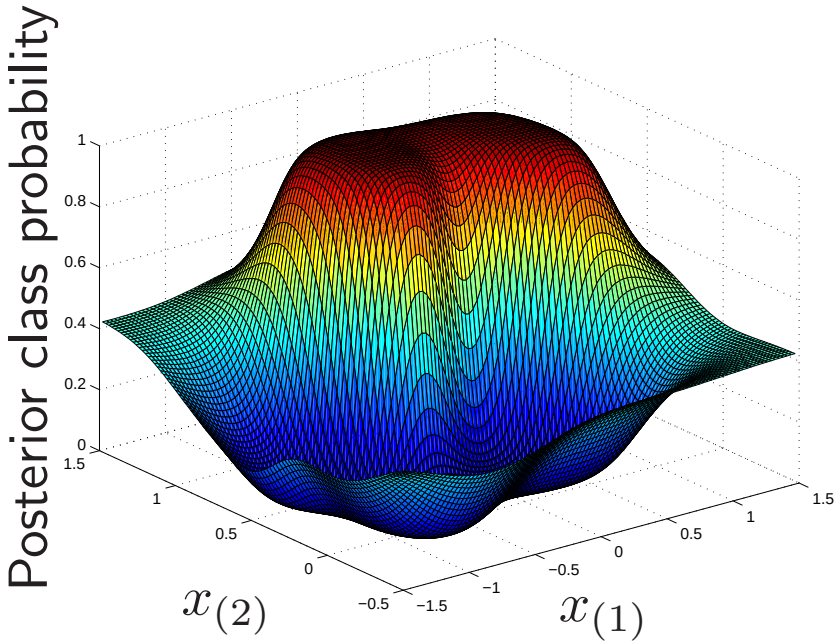
- SVM solution by applying weighted LS iteratively [Perez-Cruz et al., 2005]

Bayesian inference



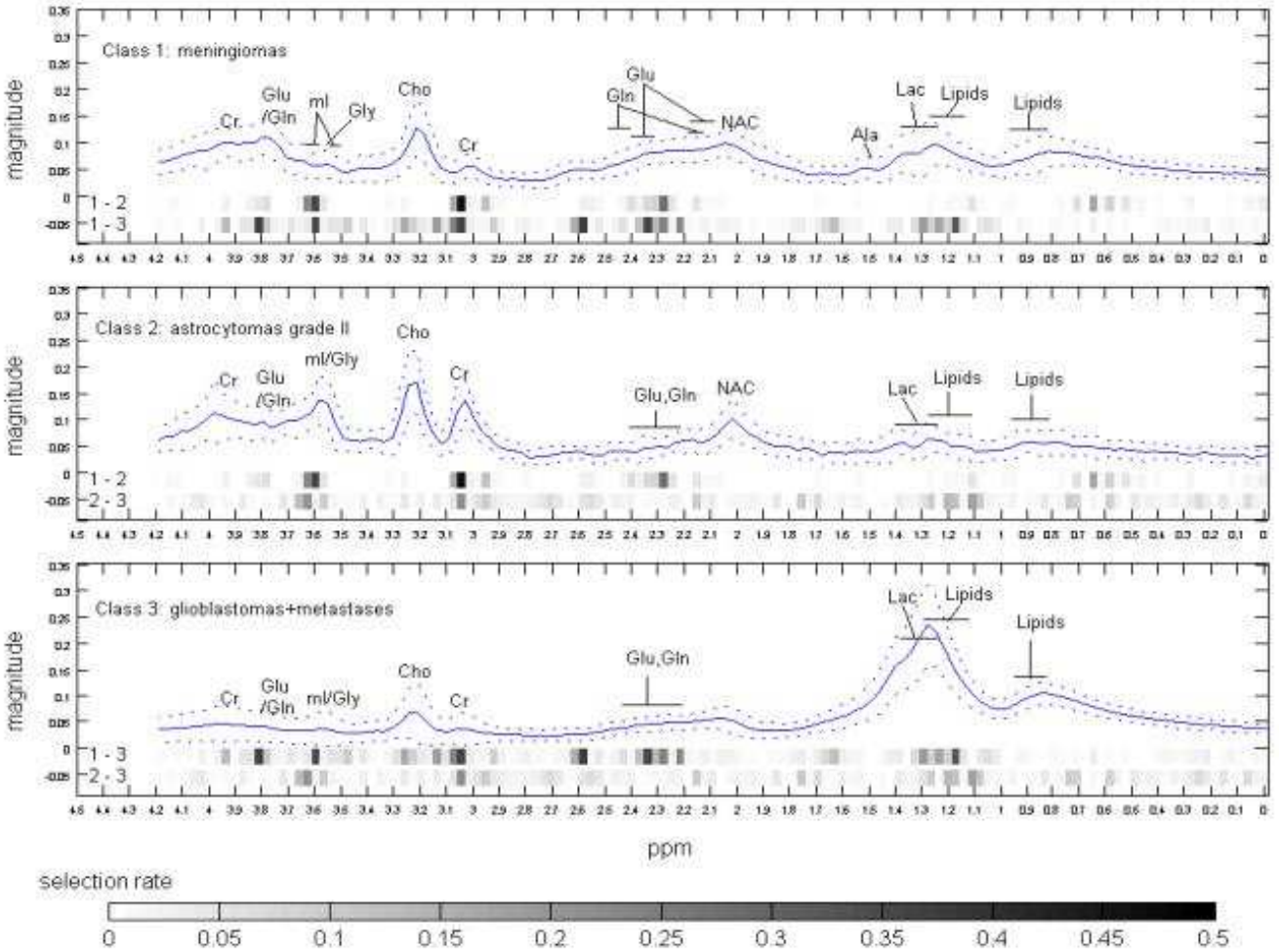
Bayesian inference: classification





- Probabilistic interpretation with moderated output
- Bias term correction for unbalanced and/or small data sets

Classification of brain tumors using ARD



Bayesian learning (automatic relevance determination) of most relevant frequencies

[Lu et al.]

Additive regularization trade-off

- Traditional Tikhonov regularization scheme:

$$\min_{w,e} w^T w + \gamma \sum_i e_i^2 \quad \text{s.t.} \quad e_i = y_i - w^T \varphi(x_i), \quad \forall i = 1, ..., N$$

Training solution for fixed value of γ:

$$(K + I/\gamma)\alpha = y$$

$\longrightarrow$ Selection of γ via validation set: non-convex problem

- Additive regularization trade-off [Pelckmans et al., 2005]:

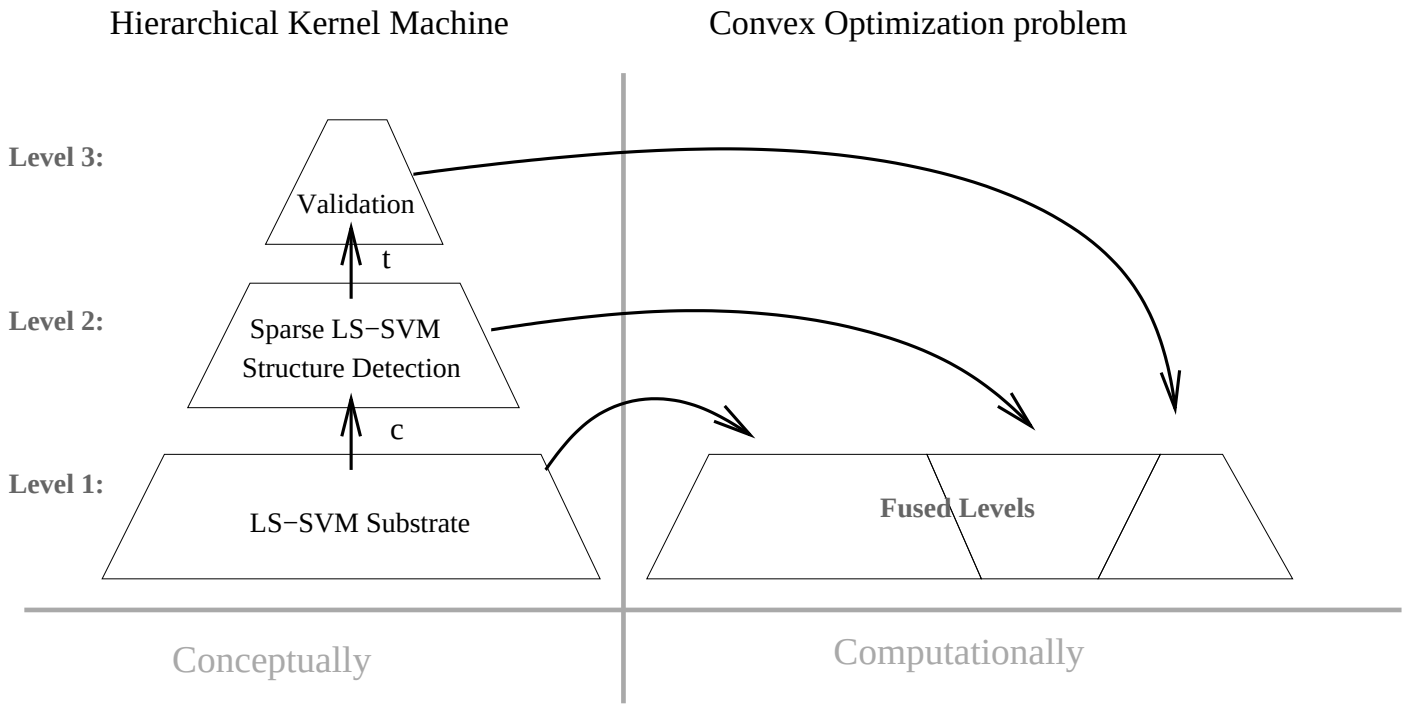
$$\min_{w,e} w^T w + \sum_i (e_i - c_i)^2 \quad \text{s.t.} \quad e_i = y_i - w^T \varphi(x_i), \quad \forall i = 1, ..., N$$

Training solution for fixed value of $c = [c1; ...; c_N]$:

$$(K + I)\alpha = y - c$$

$\longrightarrow$ Selection of c via validation set: can be convex problem

Hierarchical Kernel Machines



Building sparse representations, stable models, additive models for input selection on LS-SVM substrates. Fuse training, validation and hyperparameter selection into one single convex optimization problem.

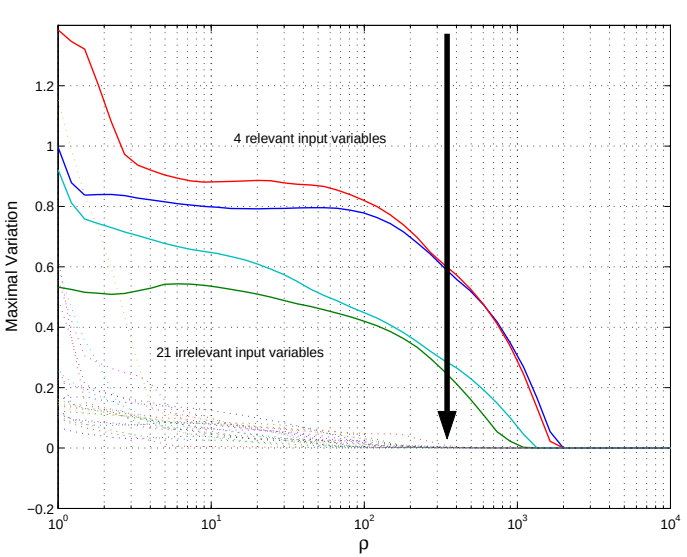
[Pelckmans et al., 2005]

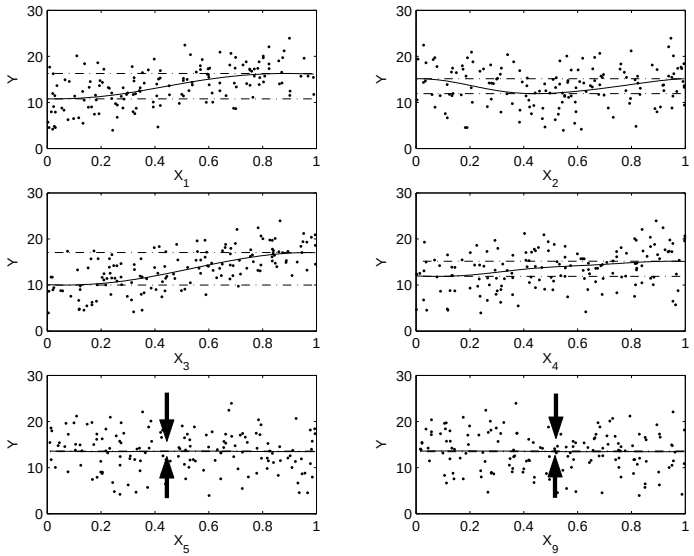
Additive models and structure detection

- Additive models: $\hat{y}(x) = \sum_{p=1}^{P} w^{(p)^T} \varphi^{(p)}(x^{(p)})$ with $x^{(p)}$ the p-th input variable and a feature map $\varphi^{(p)}$ for each variable. This leads to the kernel $K(x_i, x_j) = \sum_{p=1}^{P} K^{(p)}(x_i^{(p)}, x_j^{(p)})$.
- Structure detection [Pelckmans et al., 2005]:

$$\min_{w,e,t} \quad \mu \sum_{p=1}^{P} t_p + \sum_{p=1}^{P} w^{(p)^T} w^{(p)} + \gamma \sum_{i=1}^{N} e_i^2$$

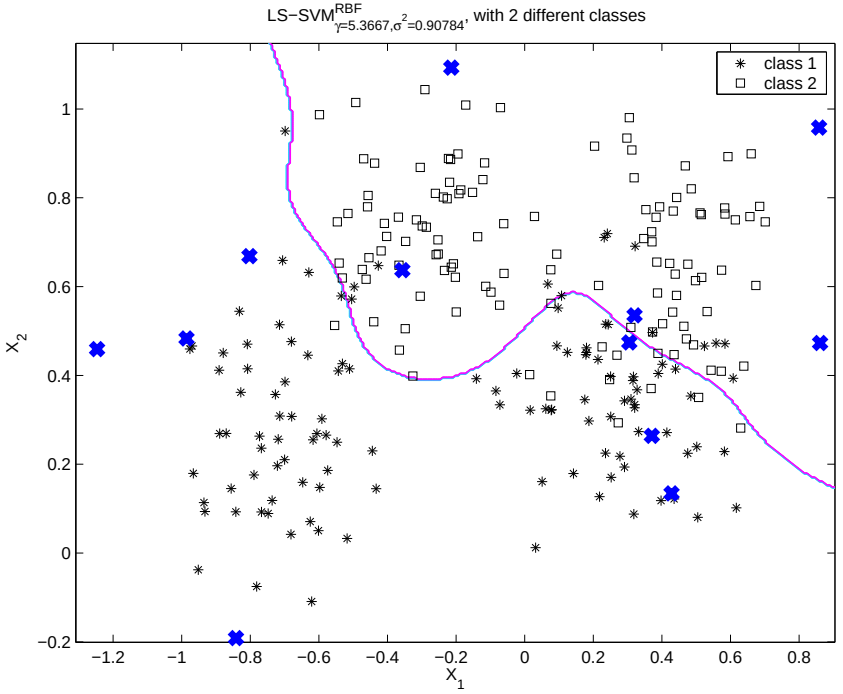
$$\text{s.t.} \begin{cases} y_i = \sum_{p=1}^{P} w^{(p)^T} \varphi^{(p)}(x_i^{(p)}) + e_i, & \forall i = 1, ..., N \\ -t_p \leq w^{(p)^T} \varphi^{(p)}(x_i^{(p)}) \leq t_p, & \forall i = 1, ..., N, \forall p = 1, ..., P \end{cases}$$





Sparse models

- SVM classically: sparse solution from QP problem at training level
- Hierarchical kernel machine: fused problem with sparseness obtained at the validation level



Conclusions of Part II

- Support vector machines: straightforward to add additional constraints and preserving convexity
- Customizing kernels towards different types of data and applications
- Hierarchical kernel machines: several objectives at different levels fused to one single convex optimization problem
- Challenges: robustness, input selection, on-line learning, large scale problems, ...

References: books

- Cristianini N., Shawe-Taylor J., *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

- Schölkopf B., Smola A., *Learning with Kernels*, MIT Press, Cambridge, MA, 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, New York, 1998.

- Wahba G., *Spline Models for Observational Data*, Series in Applied Mathematics, **59**, SIAM, Philadelphia, 1990.

Related references

- Brown M., Grundy W., Lin D., Cristianini N., Sugnet C, Furey T., Ares M., Haussler D., "Knowledge-based analysis of microarray gene expression data using support vector machines", *Proceedings of the National Academy of Science*, **97**(1), 262–267, 2000.
- Burges C.J.C., "A tutorial on support vector machines for pattern recognition", *Knowledge Discovery and Data Mining*, **2**(2), 121-167, 1998.
- Cawley G.C., Talbot N.L.C., "Fast exact leave-one-out cross-validation of sparse least-squares support vector machines", *Neural Networks*, Vol. 17, 10, pp. 1467-1475, 2004.
- Cortes C., Vapnik V., "Support vector networks", *Machine Learning*, **20**, 273–297, 1995.
- Devos A., Lukas L., Suykens J.A.K., Vanhamme L., Tate A.R., Howe F.A., Majos C., Moreno-Torres A., Van der Graaf M., Arus C., Van Huffel S., "Classification of brain tumours using short echo time 1H MRS spectra", *Journal of Magnetic Resonance*, vol. 170 , no. 1, Sep. 2004, pp. 164-175.
- Espinoza M., Suykens J., De Moor B., "Kernel Based Partially Linear Models and Nonlinear Identification", *IEEE Transactions on Automatic control, special issue (System identification : linear vs nonlinear)*, to appear.
- Girolami M., "Orthogonal series density estimation and the kernel eigenvalue problem", *Neural Computation*, **14**(3), 669–688, 2002.
- Girosi F., "An equivalence between sparse approximation and support vector machines", *Neural Computation*, **10**(6), 1455–1480, 1998.
- Guyon I., Weston J., Barnhill S., Vapnik V., "Gene selection for cancer classification using support vector machines", *Machine Learning*, **46**, 389–422, 2002.
- Hoegaerts L., Suykens J.A.K., Vandewalle J., De Moor B., "Subset based least squares subspace regression in RKHS", *Neurocomputing*, vol. 63, Jan. 2005, pp. 293-323.

- Jebara T., Kondor R., Howard A., "Probability Product Kernels", *Journal of Machine Learning Research*, 5(Jul):819–844, 2004.
- Kwok J.T., "The evidence framework applied to support vector machines", *IEEE Transactions on Neural Networks*, **10**, 1018–1031, 2000.
- Lin C.-J., "On the convergence of the decomposition method for support vector machines", *IEEE Transactions on Neural Networks*. **12**, 1288–1298, 2001.
- Lu C., Van Gestel T., Suykens J.A.K., Van Huffel S., Vergote I., Timmerman D., "Preoperative prediction of malignancy of ovarium tumor using least squares support vector machines", *Artificial Intelligence in Medicine*, vol. 28, no. 3, Jul. 2003, pp. 281-306.
- MacKay D.J.C., "Bayesian interpolation", *Neural Computation,* **4**(3), 415–447, 1992.
- Mercer J., "Functions of positive and negative type and their connection with the theory of integral equations", *Philos. Trans. Roy. Soc. London*, **209**, 415–446, 1909.
- Mika S., Rätsch G., Weston J., Schölkopf B., Müller K.-R., "Fisher discriminant analysis with kernels", In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, 41-48. IEEE, 1999.
- Müller K.R., Mika S., Ratsch G., Tsuda K., Schölkopf B., "An introduction to kernel-based learning algorithms", *IEEE Transactions on Neural Networks*, 2001. 12(2): 181-201, 2001.
- Pelckmans K., Suykens J.A.K., De Moor B., "Building Sparse Representations and Structure Determination on LS-SVM Substrates", *Neurocomputing*, vol. 64, Mar. 2005, pp. 137-159.
- Pelckmans K., Suykens J.A.K., De Moor B., "Additive regularization trade-off: fusion of training and validation levels in kernel Methods", Internal Report 03-184, ESAT-SISTA, K.U.Leuven (Leuven, Belgium).
- Pelckmans K., Espinoza M., De Brabanter J., Suykens J.A.K., De Moor B., "Primal-Dual Monotone Kernel Regression", *Neural processing letters*, to appear.

- Perez-Cruz F. Bousono-Calzon C., Artes-Rodriguez A., "Convergence of the IRWLS Procedure to the Support Vector Machine Solution", *Neural Computation*, 17: 7-18, 2005.
- Platt J., "Fast training of support vector machines using sequential minimal optimization", In Schölkopf B., Burges C.J.C., Smola A.J. (Eds.) *Advances in Kernel methods - Support Vector Learning*, 185–208, MIT Press, 1999.
- Pochet N., De Smet F., Suykens J.A.K., De Moor B., "Systematic benchmarking of micorarray data classification : assessing the role of nonlinearity and dimensionality reduction", *Bioinformatics*, vol. 20, no. 17, Nov. 2004, pp. 3185-3195.
- Poggio T., Girosi F., "Networks for approximation and learning", *Proceedings of the IEEE*, **78**(9), 1481–1497, 1990.
- Principe J., Fisher III, Xu D., "Information theoretic learning", in S. Haykin (Ed.), *Unsupervised Adaptive Filtering*, John Wiley & Sons, New York, 2000.
- Rosipal R., Trejo L.J., "Kernel partial least squares regression in reproducing kernel Hilbert space", *Journal of Machine Learning Research*, **2**, 97–123, 2001.
- Saunders C., Gammerman A., Vovk V., "Ridge regression learning algorithm in dual variables", *Proc. of the 15th Int. Conf. on Machine Learning (ICML-98)*, Madison-Wisconsin, 515–521, 1998.
- Schölkopf B., Smola A., Müller K.-R., "Nonlinear component analysis as a kernel eigenvalue problem", *Neural Computation,* **10**, 1299–1319, 1998.
- Schölkopf B., Mika S., Burges C., Knirsch P., Müller K.-R., Rätsch G., Smola A., "Input space vs. feature space in kernel-based methods", *IEEE Transactions on Neural Networks*, **10**(5), 1000–1017, 1999.
- Suykens J.A.K., Vandewalle J., "Least squares support vector machine classifiers", *Neural Processing Letters*, vol. 9, no. 3, Jun. 1999, pp. 293-300.
- Suykens J.A.K., Vandewalle J., De Moor B., "Optimal control by least squares support vector machines", *Neural Networks*, vol. 14, no. 1, Jan. 2001, pp. 23-35.

- Suykens J.A.K., De Brabanter J., Lukas L., Vandewalle J., "Weighted least squares support vector machines : robustness and sparse approximation", *Neurocomputing, Special issue on fundamental and information processing aspects of neurocomputing*, vol. 48, no. 1-4, Oct. 2002, pp. 85-105.
- Suykens J.A.K., Van Gestel T., Vandewalle J., De Moor B., "A support vector machine formulation to PCA analysis and its kernel version", *IEEE Transactions on Neural Networks*, vol. 14, no. 2, Mar. 2003, pp. 447-450.
- Tsuda K., Kin T., Asai K. "Marginalized kernels for biological sequences", *Bioinformatics,* 18(Suppl.1): S268-S275, 2002.
- Van Gestel T., Suykens J., Baesens B., Viaene S., Vanthienen J., Dedene G., De Moor B., Vandewalle J., "Benchmarking Least Squares Support Vector Machine Classifiers", *Machine Learning*, vol. 54, no. 1, Jan. 2004, pp. 5-32.
- Van Gestel T., Suykens J., Lanckriet G., Lambrechts A., De Moor B., Vandewalle J., "Bayesian Framework for Least Squares Support Vector Machine Classifiers, Gaussian Processes and Kernel Fisher Discriminant Analysis", *Neural Computation*, vol. 15, no. 5, May 2002, pp. 1115-1148.
- Williams C.K.I., Rasmussen C.E., "Gaussian processes for regression", In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems* **8**, 514–520. MIT Press, 1996.
- Williams C.K.I., Seeger M., "Using the Nyström method to speed up kernel machines", In T.K. Leen, T.G. Dietterich, and V. Tresp (Eds.), *Advances in neural information processing systems*, **13**, 682–688, MIT Press, 2001.