Least Squares Support Vector Machines

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

NATO-ASI Learning Theory and Practice
Leuven July 2002
http://www.esat.kuleuven.ac.be/sista/natoasi/ltp2002.html

Main reference:

J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, in press (ISBN 981-238-151-1)

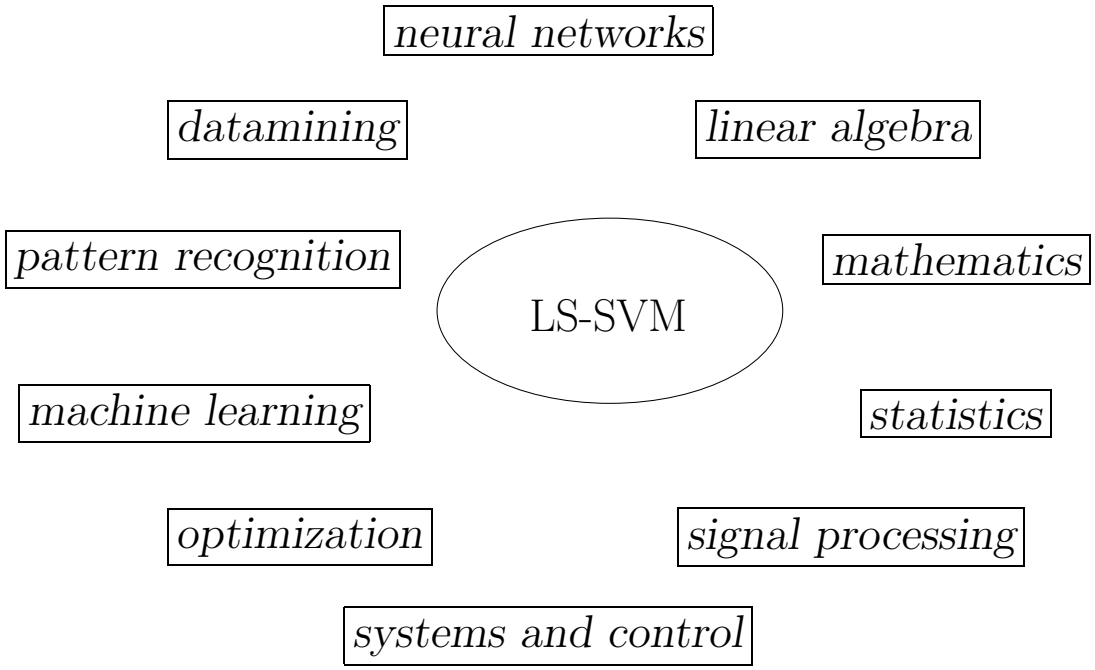
Related software *LS-SVMlab* (Matlab/C toolbox):

http://www.esat.kuleuven.ac.be/sista/lssvmlab/

(software + publications)

... with thanks to Kristiaan Pelckmans, Bart Hamers, Lukas, Luc Hoegaerts, Chuan Lu, Lieveke Ameye, Sabine Van Huffel, Gert Lanckriet, Tijl De Bie and many others

Interdisciplinary challenges



One of the original dreams in the neural networks area is to make a universal class of models (such as MLPs) generally applicable to a wide range of applications.

Presently, standard SVMs are mainly available only for classification, function estimation and density estimation. However, several extensions are possible in terms of least squares and equality constraints (and exploiting primal-dual interpretations).

Overview of this talk

- LS-SVM for classification and link with kernel Fisher discriminant analysis
- Links to regularization networks and Gaussian processes
- Bayesian inference for LS-SVMs
- Sparseness and robustness
- Large scale methods:
  Fixed Size LS-SVM
  Extensions to committee networks
- New formulations to kernel PCA, kernel CCA, kernel PLS
- Extensions of LS-SVM to recurrent networks and optimal control

Vapnik's SVM classifier

- Given a training set $\{x_k, y_k\}_{k=1}^N$
  Input patterns $x_k \in \mathbb{R}^n$
  Class labels $y_k \in \mathbb{R}$ where $y_k \in \{-1, +1\}$

- Classifier:
$$y(x) = \text{sign}[w^T \varphi(x) + b]$$
  with $\varphi(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ mapping to high dimensional feature space (can be infinite dimensional)

- For separable data, assume
$$\begin{cases} w^T \varphi(x_k) + b \geq +1 \ , & \text{if } \ y_k = +1 \\ w^T \varphi(x_k) + b \leq -1 \ , & \text{if } \ y_k = -1 \end{cases}$$
  which is equivalent to
$$y_k[w^T \varphi(x_k) + b] \geq 1, \quad k = 1, ..., N$$

- Optimization problem (non-separable case):
$$\min_{w,\xi} \mathcal{J}(w, \xi) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k$$
  subject to
$$\begin{cases} y_k[w^T \varphi(x_k) + b] \geq 1 - \xi_k, \quad k = 1, ..., N \\ \xi_k \geq 0, \quad k = 1, ..., N. \end{cases}$$

- Construct Lagrangian:

$$\mathcal{L}(w, b, \xi; \alpha, \nu) = \mathcal{J}(w, \xi_k) - \sum_{k=1}^{N} \alpha_k \{y_k [w^T \varphi(x_k) + b] - 1 + \xi_k\} - \sum_{k=1}^{N} \nu_k \xi_k$$

  with Lagrange multipliers $\alpha_k \geq 0$, $\nu_k \geq 0$ $(k = 1, ..., N)$.

- Solution given by saddle point of Lagrangian:

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

  One obtains

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \ \rightarrow \ w = \sum_{k=1}^{N} \alpha_k y_k \varphi(x_k) \\ \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \ \rightarrow \ \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \\ \frac{\partial \mathcal{L}}{\partial \xi_k} = 0 \ \rightarrow \ 0 \leq \alpha_k \leq c, \ k = 1, ..., N \end{cases}$$

- Quadratic programming problem (Dual problem):

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

  such that

$$\begin{cases} \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \\ 0 \leq \alpha_k \leq c, \ k = 1, ..., N. \end{cases}$$

Note: w and $\varphi(x_k)$ are not calculated.

- Mercer condition:

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

- Obtained classifier:

$$y(x) = \text{sign}[\sum_{k=1}^{N} \alpha_k \, y_k \, K(x, x_k) + b]$$

with α_k positive real constants, b real constant, that follow as solution to the QP problem.

Non-zero α_k are called support values and the corresponding data points are called support vectors.
The bias term b follows from KKT conditions.

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

$K(x, x_k) = x_k^T x$ (linear SVM)

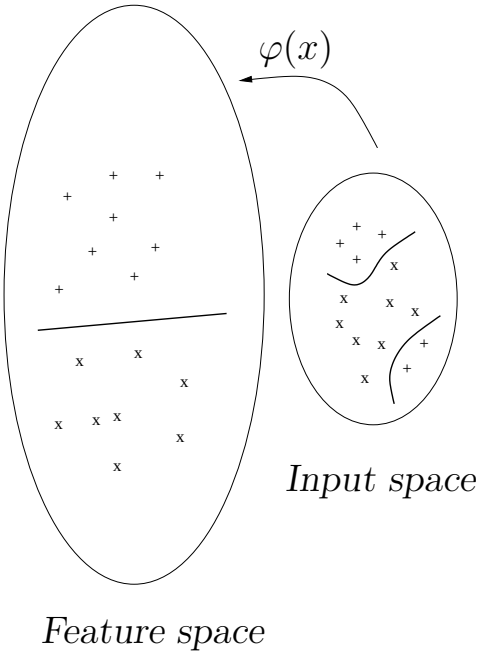
$K(x, x_k) = (x_k^T x + 1)^d$ (polynomial SVM of degree d)

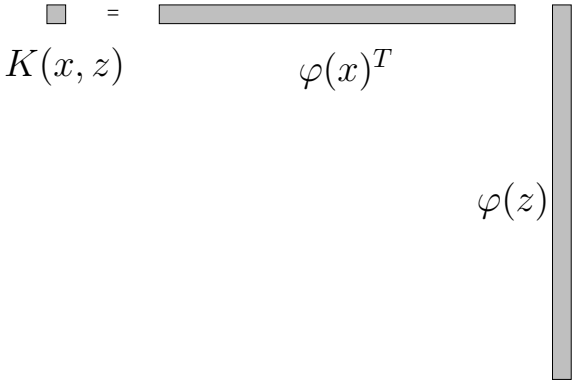
$K(x, x_k) = \exp\{-\|x - x_k\|_2^2 / \sigma^2\}$ (RBF SVM)

$K(x, x_k) = \tanh(\kappa \, x_k^T x + \theta)$ (MLP SVM)

- In the case of RBF and MLP kernel, the number of hidden units corresponds to the number of support vectors.

Feature space and kernel trick



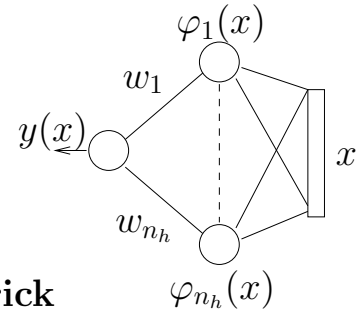


Primal-dual interpretations of SVMs

Primal problem P

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

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



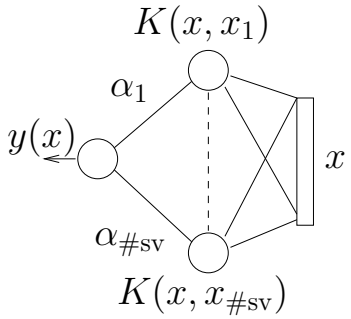
Kernel trick

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

Dual problem D

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

$y(x) = \text{sign}[\sum_{k=1}^{\#\text{sv}} \alpha_k y_k K(x, x_k) + b]$



Least Squares SVM classifiers

- LS-SVM classifiers (Suykens, 1999): close to Vapnik's SVM formulation but solves linear system instead of QP problem.
- Optimization problem:

$$\min_{w,b,e} \mathcal{J}(w,b,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2$$

  subject to the equality constraints

$$y_k \left[w^T \varphi(x_k) + b \right] = 1 - e_k, \;\; k = 1, ..., N.$$

- Lagrangian

$$\mathcal{L}(w,b,e;\alpha) = \mathcal{J}(w,b,e) - \sum_{k=1}^{N} \alpha_k \{ y_k [w^T \varphi(x_k) + b] - 1 + e_k \}$$

  where α_k are Lagrange multipliers.

- Conditions for optimality:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \;\rightarrow\; w = \sum_{k=1}^{N} \alpha_k y_k \varphi(x_k) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \;\rightarrow\; \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\rightarrow\; \alpha_k = \gamma e_k, & k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\rightarrow\; y_k [w^T \varphi(x_k) + b] - 1 + e_k = 0, & k = 1, ..., N \end{cases}$$

- Set of linear equations (instead of QP):

$$\left[\begin{array}{ccc|c} I & 0 & 0 & -Z^T \\ 0 & 0 & 0 & -Y^T \\ 0 & 0 & \gamma I & -I \\ \hline Z & Y & I & 0 \end{array}\right] \left[\begin{array}{c} w \\ b \\ e \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline \vec{1} \end{array}\right]$$

with

$$\begin{array}{l} Z = [\varphi(x_1)^T y_1; ...; \varphi(x_N)^T y_N] \\ Y = [y_1; ...; y_N] \\ \vec{1} = [1; ...; 1] \\ e = [e_1; ...; e_N] \\ \alpha = [\alpha_1; ...; \alpha_N]. \end{array}$$

After elimination of w, e one obtains

$$\left[\begin{array}{c|c} 0 & Y^T \\ \hline Y & \Omega + \gamma^{-1} I \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline \vec{1} \end{array}\right].$$

where

$$\Omega = Z Z^T$$

and Mercer's condition is applied

$$\begin{aligned} \Omega_{kl} &= y_k y_l \, \varphi(x_k)^T \varphi(x_l) \\ &= y_k y_l \, K(x_k, x_l). \end{aligned}$$

- Related work: Saunders (1998), Smola & Schölkopf (1998), Cristianini & Taylor (2000)

Large scale LS-SVMs

- For the binary class case:
  problem involves matrix of size $(N+1) \times (N+1)$.

  Large data sets $\rightarrow$ iterative methods needed (CG, SOR, ...)

- Problem: solving

$$\mathcal{A}x = \mathcal{B} \qquad \mathcal{A} \in \mathbb{R}^{n \times n}, \mathcal{B} \in \mathbb{R}^n$$

  by CG requires that $\mathcal{A}$ is symmetric positive definite.

- Represent the original problem of the form

$$\begin{bmatrix} 0 & Y^T \\ Y & H \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

  with $H = \Omega + \gamma^{-1} I$, $\xi_1 = b$, $\xi_2 = \alpha$, $d_1 = 0$, $d_2 = \vec{1}$ as

$$\begin{bmatrix} s & 0 \\ 0 & H \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 + H^{-1} Y \xi_1 \end{bmatrix} = \begin{bmatrix} -d_1 + Y^T H^{-1} d_2 \\ d_2 \end{bmatrix}$$

  with $s = Y^T H^{-1} Y > 0$ ($H = H^T > 0$).

  Iterative methods can be applied to the latter problem.

- Convergence of CG depends on condition number
  (hence it depends on γ, σ)

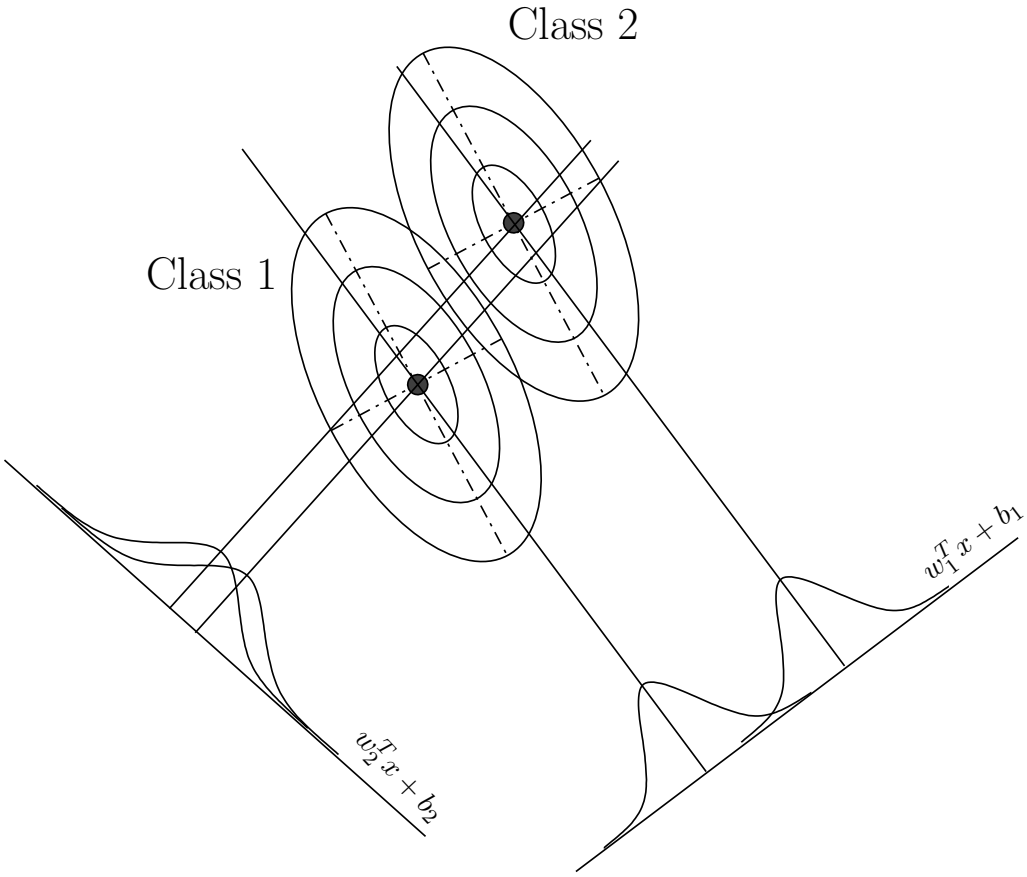
Fisher Discriminant Analysis

- Projection of data:

$$z = f(x) = w^T x + b$$

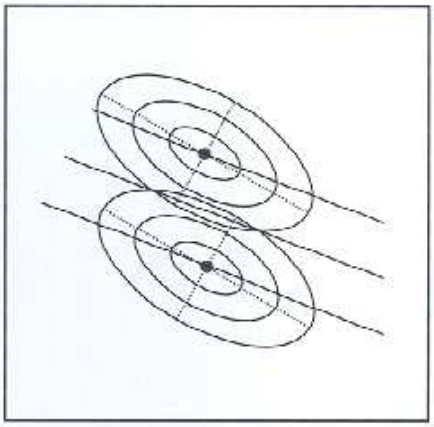
- Optimize Rayleigh quotient:

$$\max_{w,b} J_{\mathrm{FD}}(w, b) \;=\; \frac{w^T \Sigma_{\mathcal{B}} w}{w^T \Sigma_{\mathcal{W}} w}$$

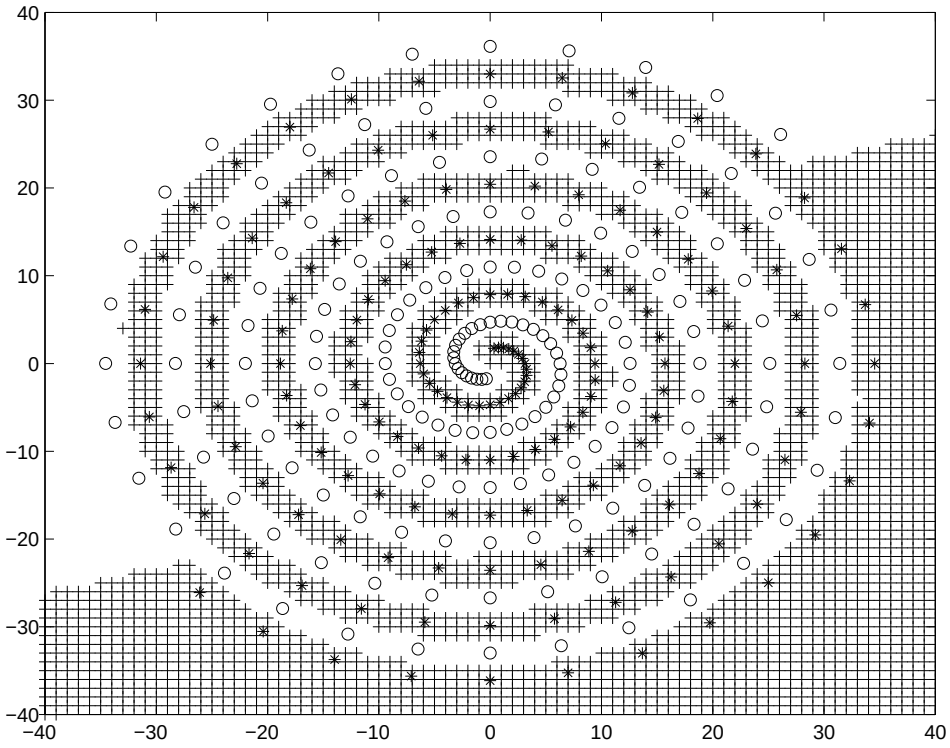


NEURAL COMPUTATION

Volume 14 Number 5 May 2002



A two spiral classification problem



Difficult problem for MLPs, not for SVM or LS-SVM with RBF kernel (but easy in the sense that the two classes are separable)

Benchmarking LS-SVM classifiers

	acr	bld	gcr	hea	ion	pid	snr	ttt	wbc	adu
N_{CV}	460	230	666	180	234	512	138	638	455	33000
N_{test}	230	115	334	90	117	256	70	320	228	12222
N	690	345	1000	270	351	768	208	958	683	45222
n_{num}	6	6	7	7	33	8	60	0	9	6
n_{cat}	8	0	13	6	0	0	0	9	0	8
n	14	6	20	13	33	8	60	9	9	14

	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
$n_{y,\text{MOC}}$	2	2	3	2	4	2	4	2	2	2
$n_{y,\text{1vs1}}$	3	3	21	3	45	3	45	6	2	3

(Van Gestel *et al.*, Machine Learning, in press)

Successful LS-SVM applications

- 20 UCI benchmark data sets (binary and multiclass problems)
- Ovarian cancer classification
- Classification of brain tumours from magnetic resonance spectroscopy signals
- Prediction of mental development of preterm newborns
- Prediction of air polution and chaotic time series
- Marketting and financial engineering studies
- Modelling the Belgian electricity consumption
- Softsensor modelling in the chemical process industry

	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_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_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_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_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_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*

	bal	cmc	ims	iri	led	thy	usp	veh	wav	win	AA	AR	P_{ST}
N_{test}	209	491	770	50	1000	2400	3298	282	1200	60			
n	4	9	18	4	7	21	256	18	19	13			
RBF LS-SVM (MOC)	*92.7*(1.0)	**54.1**(1.8)	*95.5*(0.6)	*96.6*(2.8)	*70.8*(1.4)	*96.6*(0.4)	*95.3*(0.5)	81.9(2.6)	**99.8**(0.2)	**98.7**(1.3)	**88.2**	**7.1**	**0.344**
RBF LS-SVM$_F$ (MOC)	*86.8*(2.4)	*43.5*(2.6)	*69.6*(3.2)	**98.4**(2.1)	*36.1*(2.4)	*22.0*(4.7)	*86.5*(1.0)	*66.5*(6.1)	*99.5*(0.2)	*93.2*(3.4)	*70.2*	17.8	**0.109**
Lin LS-SVM (MOC)	*90.4*(0.8)	*46.9*(3.0)	*72.1*(1.2)	*89.6*(5.6)	*52.1*(2.2)	*93.2*(0.6)	*76.5*(0.6)	*69.4*(2.3)	*90.4*(1.1)	**97.3**(2.0)	*77.8*	*17.8*	*0.002*
Lin LS-SVM$_F$ (MOC)	*86.6*(1.7)	*42.7*(2.0)	*69.8*(1.2)	*77.0*(3.8)	*35.1*(2.6)	*54.1*(1.3)	*58.2*(0.9)	*69.1*(2.0)	*55.7*(1.3)	*85.5*(5.1)	*63.4*	*22.4*	*0.002*
Pol LS-SVM (MOC)	*94.0*(0.8)	53.5(2.3)	*87.2*(2.6)	**96.4**(3.7)	*70.9*(1.5)	*94.7*(0.2)	*95.0*(0.8)	*81.8*(1.2)	*99.6*(0.3)	**97.8**(1.9)	87.1	**9.8**	**0.109**
Pol LS-SVM$_F$ (MOC)	*93.2*(1.9)	*47.4*(1.6)	*86.2*(3.2)	96.0(3.7)	*67.7*(0.8)	*69.9*(2.8)	*87.2*(0.9)	*81.9*(1.3)	*96.1*(0.7)	*92.2*(3.2)	*81.8*	*15.7*	*0.002*
RBF LS-SVM (1vs1)	*94.2*(2.2)	**55.7**(2.2)	**96.5**(0.5)	**97.6**(2.3)	**74.1**(1.3)	*96.8*(0.3)	94.8(2.5)	83.6(1.3)	*99.3*(0.4)	**98.2**(1.8)	**89.1**	**5.9**	**1.000**
RBF LS-SVM$_F$ (1vs1)	*71.4*(15.5)	*42.7*(3.7)	*46.2*(6.5)	*79.8*(10.3)	*58.9*(8.5)	*92.6*(0.2)	*30.7*(2.4)	*24.9*(2.5)	*97.3*(1.7)	*67.3*(14.6)	*61.2*	*22.3*	*0.002*
Lin LS-SVM (1vs1)	*87.8*(2.2)	*50.8*(2.4)	*93.4*(1.0)	**98.4**(1.8)	**74.5**(1.0)	*93.2*(0.3)	*95.4*(0.3)	*79.8*(2.1)	*97.6*(0.9)	**98.3**(2.5)	**86.9**	**9.7**	**0.754**
Lin LS-SVM$_F$ (1vs1)	*87.7*(1.8)	*49.6*(1.8)	*93.4*(0.9)	**98.6**(1.3)	**74.5**(1.0)	*74.9*(0.8)	*95.3*(0.3)	*79.8*(2.2)	*98.2*(0.6)	**97.7**(1.8)	85.0	**11.1**	**0.344**
Pol LS-SVM (1vs1)	*95.4*(1.0)	*53.2*(2.2)	*95.2*(0.6)	96.8(2.3)	72.8(2.6)	88.8(14.6)	**96.0**(2.1)	82.8(1.8)	*99.0*(0.4)	**99.0**(1.4)	**87.9**	**8.9**	**0.344**
Pol LS-SVM$_F$ (1vs1)	*56.5*(16.7)	*41.8*(1.8)	*30.1*(3.8)	*71.4*(12.4)	*32.6*(10.9)	*92.6*(0.7)	95.8(1.7)	*20.3*(6.7)	*77.5*(4.9)	*82.3*(12.2)	*60.1*	*21.9*	0.021
RBF SVM (MOC)	**99.2**(0.5)	*51.0*(1.4)	*94.9*(0.9)	96.6(3.4)	*69.9*(1.0)	*96.6*(0.2)	*95.5*(0.4)	*77.6*(1.7)	**99.7**(0.1)	**97.8**(2.1)	**87.9**	**8.6**	**0.344**
Lin SVM (MOC)	*98.3*(1.2)	*45.8*(1.6)	*74.1*(1.4)	**95.0**(10.5)	*50.9*(3.2)	*92.5*(0.3)	*81.9*(0.3)	*70.3*(2.5)	*99.2*(0.2)	97.3(2.6)	80.5	16.1	0.021
RBF SVM (1vs1)	*98.3*(1.2)	**54.7**(2.4)	*96.0*(0.4)	97.0(3.0)	*64.6*(5.6)	*98.3*(0.3)	**97.2**(0.2)	83.8(1.6)	*99.6*(0.2)	**96.8**(5.7)	**88.6**	**6.5**	**1.000**
Lin SVM (1vs1)	*91.0*(2.3)	*50.8*(1.6)	*95.2*(0.7)	**98.0**(1.9)	**74.4**(1.2)	*97.1*(0.3)	*95.1*(0.3)	*78.1*(2.4)	*99.6*(0.2)	**98.3**(3.1)	**87.8**	**7.3**	**0.754**
LDA	*86.9*(2.1)	*51.8*(2.2)	*91.2*(1.1)	**98.6**(1.0)	73.7(0.8)	*93.7*(0.3)	*91.5*(0.5)	*77.4*(2.7)	*94.6*(1.2)	**98.7**(1.5)	85.8	**11.0**	**0.109**
QDA	*90.5*(1.1)	*50.6*(2.1)	*81.8*(9.6)	**98.2**(1.8)	**73.6**(1.1)	*93.4*(0.3)	*74.7*(0.7)	**84.8**(1.5)	*60.9*(9.5)	**99.2**(1.2)	**80.8**	**11.8**	**0.344**
Logit	*88.5*(2.0)	*51.6*(2.4)	*95.4*(0.6)	**97.0**(3.9)	**73.9**(1.0)	*95.8*(0.5)	*91.5*(0.5)	*78.3*(2.3)	**99.9**(0.1)	*95.0*(3.2)	*86.7*	**9.8**	0.021
C4.5	*66.0*(3.6)	*50.9*(1.7)	96.1(0.7)	96.0(3.1)	73.6(1.3)	**99.7**(0.1)	*88.7*(0.3)	*71.1*(2.6)	**99.8**(0.1)	*87.0*(5.0)	82.9	**11.8**	**0.109**
oneR	*59.5*(3.1)	*43.2*(3.5)	*62.9*(2.4)	*95.2*(2.5)	*17.8*(0.8)	*96.3*(0.5)	*32.9*(1.1)	*52.9*(1.9)	*67.4*(1.1)	*76.2*(4.6)	*60.4*	*21.6*	*0.002*
IB1	*81.5*(2.7)	*43.3*(1.1)	**96.8**(0.6)	*95.6*(3.6)	**74.0**(1.3)	*92.2*(0.4)	*97.0*(0.2)	*70.1*(2.9)	*99.7*(0.1)	*95.2*(2.0)	**84.5**	**12.9**	**0.344**
IB10	*83.6*(2.3)	*44.3*(2.4)	*94.3*(0.7)	*97.2*(1.9)	**74.2**(1.3)	*93.7*(0.3)	*96.1*(0.3)	*67.1*(2.1)	*99.4*(0.1)	*96.2*(1.9)	84.6	**12.4**	**0.344**
NB_k	*89.9*(2.0)	*51.2*(2.3)	*84.9*(1.4)	**97.0**(2.5)	**74.0**(1.2)	*96.4*(0.2)	*79.3*(0.9)	*60.0*(2.3)	*99.5*(0.1)	**97.7**(1.6)	*83.0*	12.2	0.021
NB_n	*89.9*(2.0)	*48.9*(1.8)	*80.1*(1.0)	**97.2**(2.7)	**74.0**(1.2)	*95.5*(0.4)	*78.2*(0.6)	*44.9*(2.8)	*99.5*(0.1)	97.5(1.8)	*80.6*	13.6	0.021
Maj. Rule	*48.7*(2.3)	*43.2*(1.8)	*15.5*(0.6)	*38.6*(2.8)	*11.4*(0.0)	*92.5*(0.3)	*16.8*(0.4)	*27.7*(1.5)	*34.2*(0.8)	*39.7*(2.8)	*36.8*	*24.8*	*0.002*

LS-SVMs for function estimation

- LS-SVM model as feature space representation:

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

with $x \in \mathbb{R}^n, y \in \mathbb{R}$.

The nonlinear mapping $\varphi(\cdot)$ is similar to the classifier case.

Given training set $\{x_k, y_k\}_{k=1}^N$.

- Optimization problem

$$\min_{w,e} \mathcal{J}(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2$$

subject to equality constraints

$$y_k = w^T \varphi(x_k) + b + e_k, \quad k = 1, ..., N$$

This is a form of ridge regression (see Saunders, 1998).

- Lagrangian:

$$\mathcal{L}(w, b, e; \alpha) = \mathcal{J}(w, e) - \sum_{k=1}^{N} \alpha_k \left\{ w^T \varphi(x_k) + b + e_k - y_k \right\}$$

with α_k Lagrange multipliers.

- Conditions for optimality

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \;\rightarrow\; w = \sum_{k=1}^{N} \alpha_k \varphi(x_k) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \;\rightarrow\; \sum_{k=1}^{N} \alpha_k = 0 \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\rightarrow\; \alpha_k = \gamma e_k, \qquad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\rightarrow\; w^T \varphi(x_k) + b + e_k - y_k = 0, \;\; k = 1, ..., N \end{cases}$$

- Solution

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

with

$$y = [y_1; ...; y_N],\ \vec{1} = [1; ...; 1],\ \alpha = [\alpha_1; ...; \alpha_N]$$

and by applying Mercer's condition

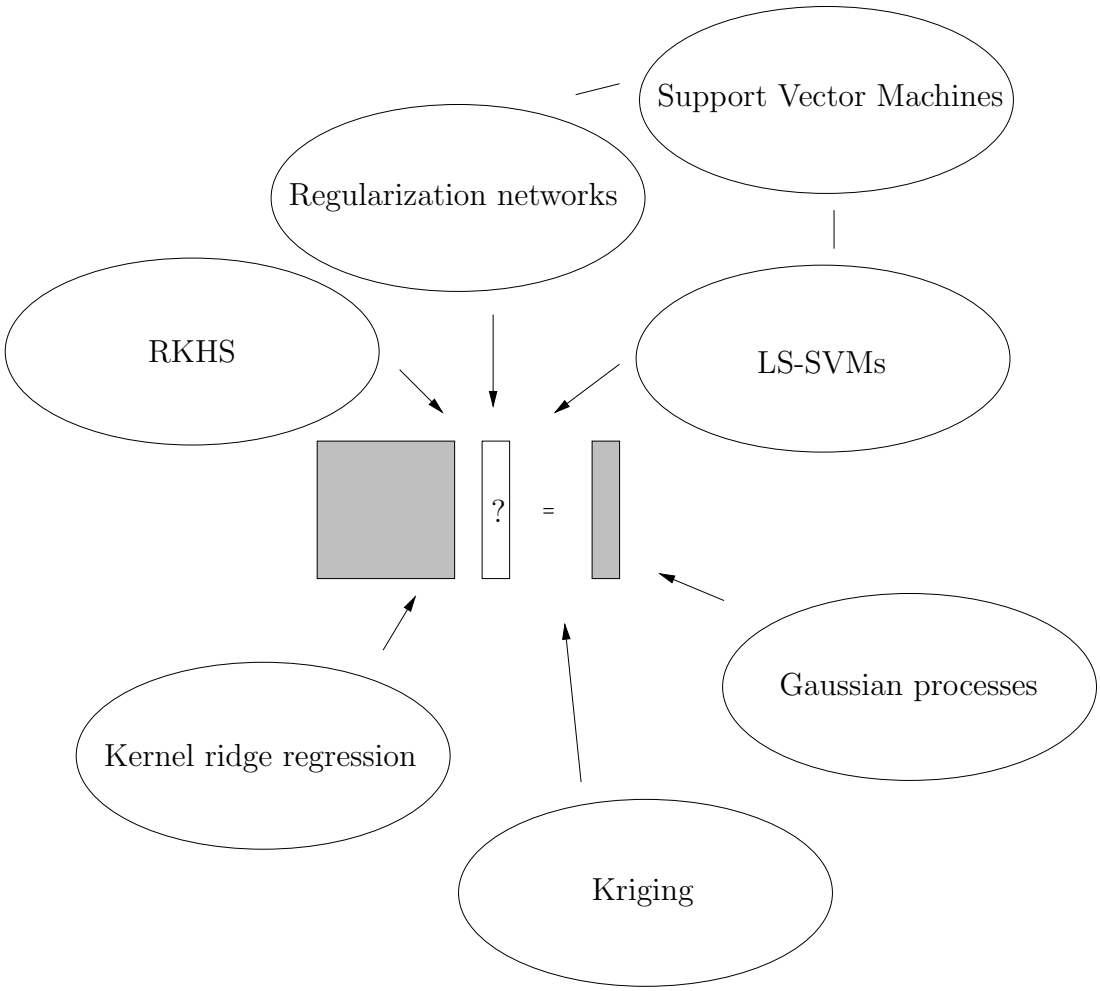
$$\begin{aligned} \Omega_{kl} &= \varphi(x_k)^T \varphi(x_l), \quad k, l = 1, ..., N \\ &= K(x_k, x_l) \end{aligned}$$

- Resulting LS-SVM model for function estimation

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

- Solution mathematically equivalent to regularization networks and Gaussian processes (usually without bias term) (see e.g. Poggio & Girosi (1990)).

SVM, RN, GP, LS-SVM, ...



Some early history:
1910-1920: Moore - RKHS
1940: Aronszajn - systematic development of RKHS
1951: Krige
1970: Parzen - RKHS for statistical problems
1971: Kimeldorf & Wahba

In LS-SVM primal-dual interpretations are emphasized (and often exploited).

Bayesian inference of LS-SVM models

Level 1 (w, b)

$p(w, b|\mathcal{D}, \mu, \zeta, \mathcal{H}_\sigma)$ $\quad$ $p(\mathcal{D}|w, b, \mu, \zeta, \mathcal{H}_\sigma)$ $\quad$ $p(w, b|\mu, \zeta, \mathcal{H}_\sigma)$

Maximize Posterior $\quad$ Likelihood $\quad$ Prior

Evidence

Level 2 (μ, ζ)

$p(\mathcal{D}|\mu, \zeta, \mathcal{H}_\sigma) =$

$p(\mu, \zeta|\mathcal{D}, \mathcal{H}_\sigma)$ $\quad$ $p(\mu, \zeta|\mathcal{H}_\sigma)$

Likelihood

Maximize Posterior $\quad$ Prior

Evidence

Level 3 (σ)

$p(\mathcal{D}|\mathcal{H}_\sigma) \quad =$

$p(\mathcal{H}_\sigma|\mathcal{D})$ $\quad$ $p(\mathcal{H}_\sigma)$

Likelihood

Maximize Posterior $\quad$ Prior

Evidence

$p(\mathcal{D})$

Bayesian LS-SVM framework

(Van Gestel, Suykens *et al.*, Neural Computation, 2002)

- *Level 1 inference:*
  Inference of w, b
  Probabilistic interpretation of outputs
  Moderated outputs
  Additional correction for prior probabilities and bias term

- *Level 2 inference:*
  Inference of hyperparameter γ
  Effective number of parameters ($< N$)
  Eigenvalues of centered kernel matrix are important

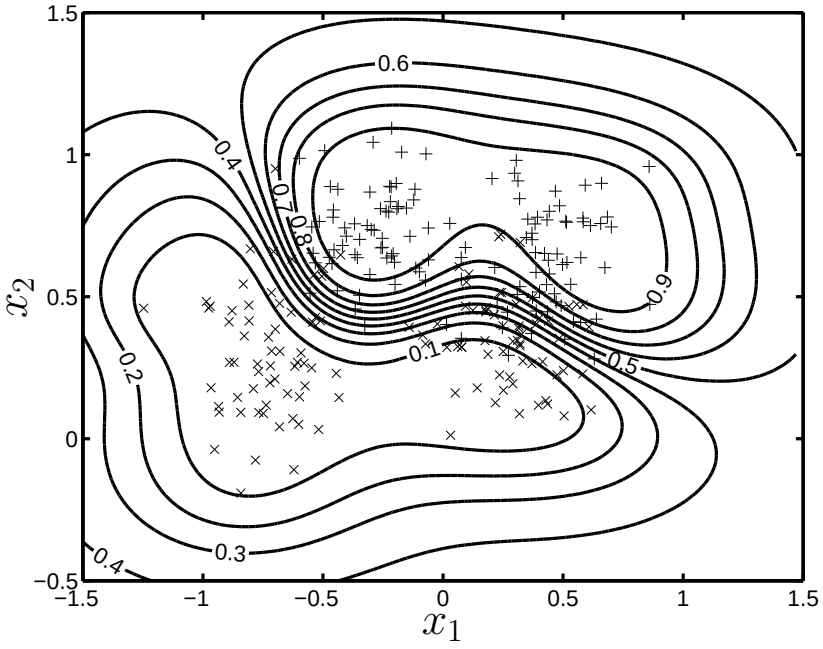
- *Level 3:*
  Model comparison - Occam's razor
  Moderated output with uncertainty on hyperparameters
  Selection of σ width of kernel
  Input selection (ARD at level 3 instead of level 2)

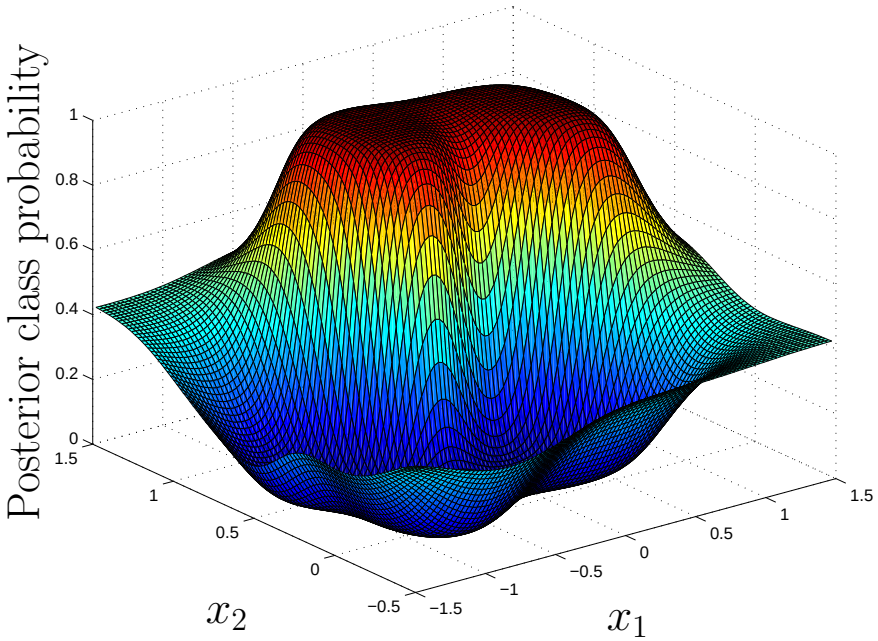
- *Related work:*
  MacKay's Bayesian interpolation for MLPs, GP
- *Difference with GP:*
  bias term contained in the formulation
  classifier case treated as a regression problem
  σ of RBF kernel determined at Level 3

Benchmarking results

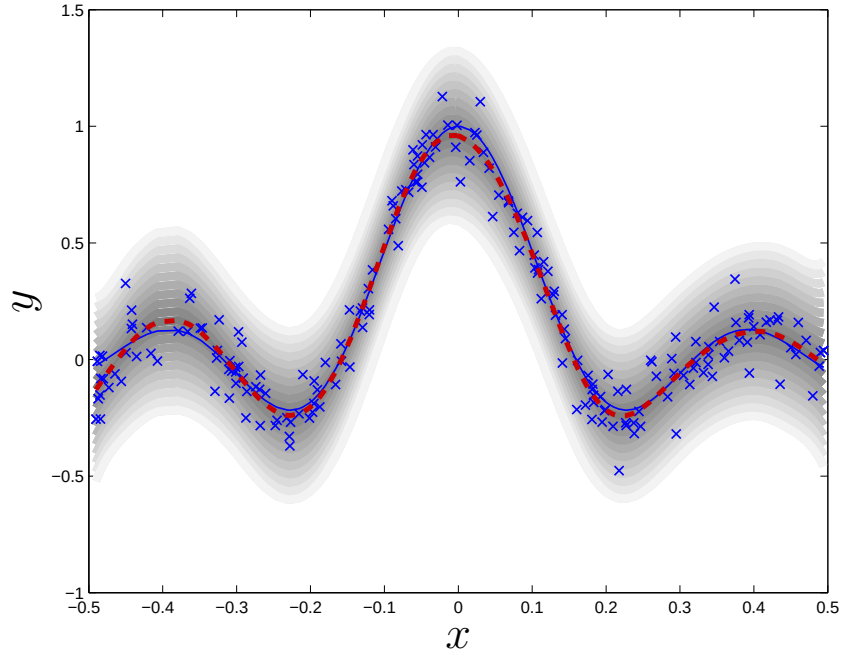
	n	N	N_{test}	N_{tot}	LS-SVM (BayM)	LS-SVM (Bay)	LS-SVM (CV10)	SVM (CV10)	GP (Bay)	GP_{b} (Bay)	GP (CV10)
bld	6	230	115	345	**69.4**(2.9)	**69.4**(3.1)	**69.4**(3.4)	**69.2**(3.5)	**69.2**(2.7)	**68.9**(3.3)	**69.7**(4.0)
cra	6	133	67	200	**96.7**(1.5)	**96.7**(1.5)	**96.9**(1.6)	**95.1**(3.2)	**96.4**(2.5)	94.8(3.2)	**96.9**(2.4)
gcr	20	666	334	1000	*73.1*(3.8)	73.5(3.9)	**75.6**(1.8)	*74.9*(1.7)	**76.2**(1.4)	**75.9**(1.7)	**75.4**(2.0)
hea	13	180	90	270	**83.6**(5.1)	**83.2**(5.2)	**84.3**(5.3)	**83.4**(4.4)	**83.1**(5.5)	**83.7**(4.9)	**84.1**(5.2)
ion	33	234	117	351	95.6(0.9)	**96.2**(1.0)	**95.6**(2.0)	95.4(1.7)	*91.0*(2.3)	*94.4*(1.9)	*92.4*(2.4)
pid	8	512	256	768	**77.3**(3.1)	**77.5**(2.8)	**77.3**(3.0)	**76.9**(2.9)	**77.6**(2.9)	**77.5**(2.7)	**77.2**(3.0)
rsy	2	250	1000	1250	**90.2**(0.7)	**90.2**(0.6)	89.6(1.1)	**89.7**(0.8)	**90.2**(0.7)	**90.1**(0.8)	**89.9**(0.8)
snr	60	138	70	208	76.7(5.6)	**78.0**(5.2)	**77.9**(4.2)	76.3(5.3)	**78.6**(4.9)	**75.7**(6.1)	**76.6**(7.2)
tit	3	1467	734	2201	**78.8**(1.1)	**78.7**(1.1)	**78.7**(1.1)	**78.7**(1.1)	**78.5**(1.0)	77.2(1.9)	**78.7**(1.2)
wbc	9	455	228	683	95.9(0.6)	*95.7*(0.5)	**96.2**(0.7)	**96.2**(0.8)	95.8(0.7)	*93.7*(2.0)	**96.5**(0.7)
AP					**83.7**	**83.9**	**84.1**	83.6	**83.7**	83.2	**83.7**
AR					**2.3**	**2.5**	**2.5**	**3.8**	**3.2**	**4.2**	**2.6**
P_{ST}					**1.000**	**0.754**	**1.000**	**0.344**	**0.754**	**0.344**	**0.508**

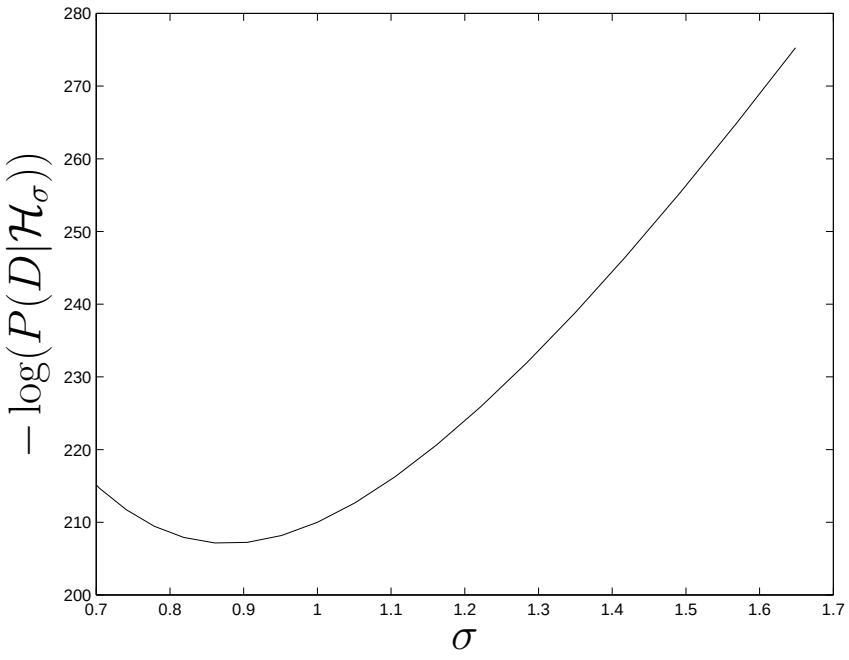
Bayesian inference of LS-SVMs: example Ripley data set



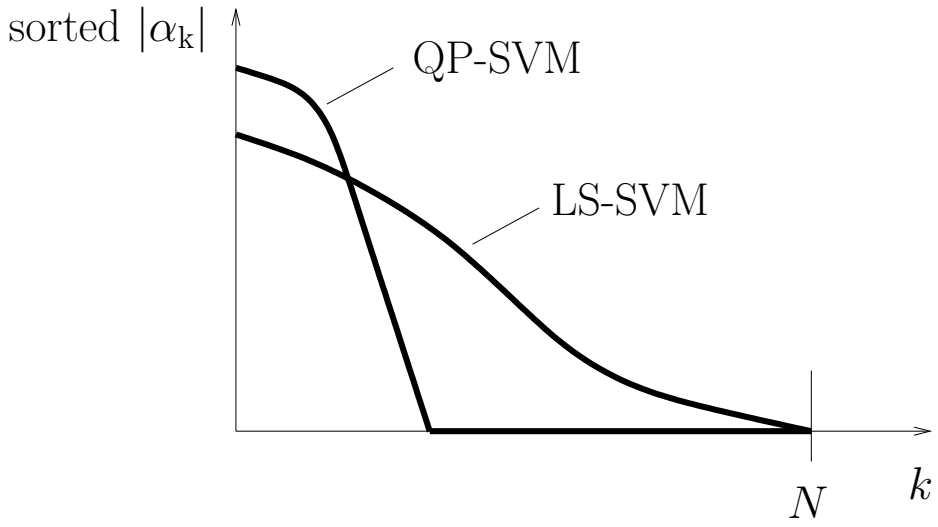


Bayesian inference of LS-SVMs: sinc example



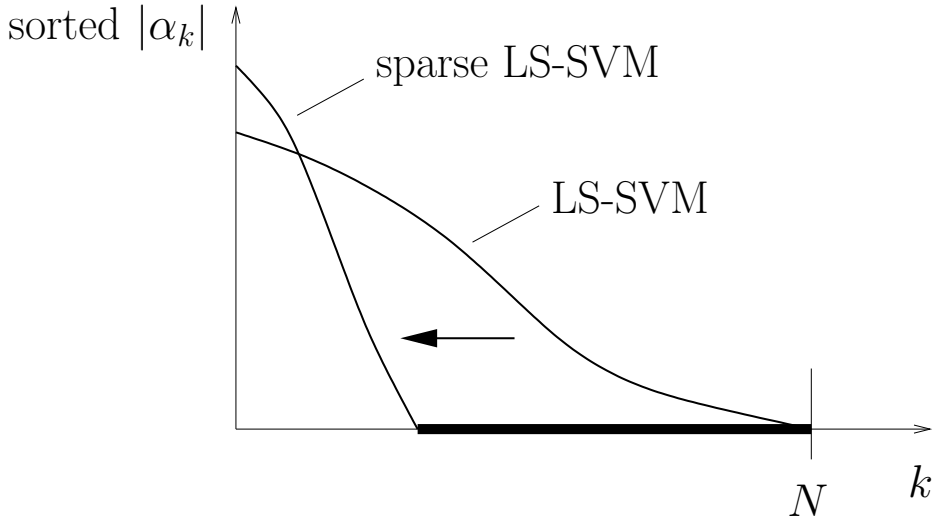


Sparseness

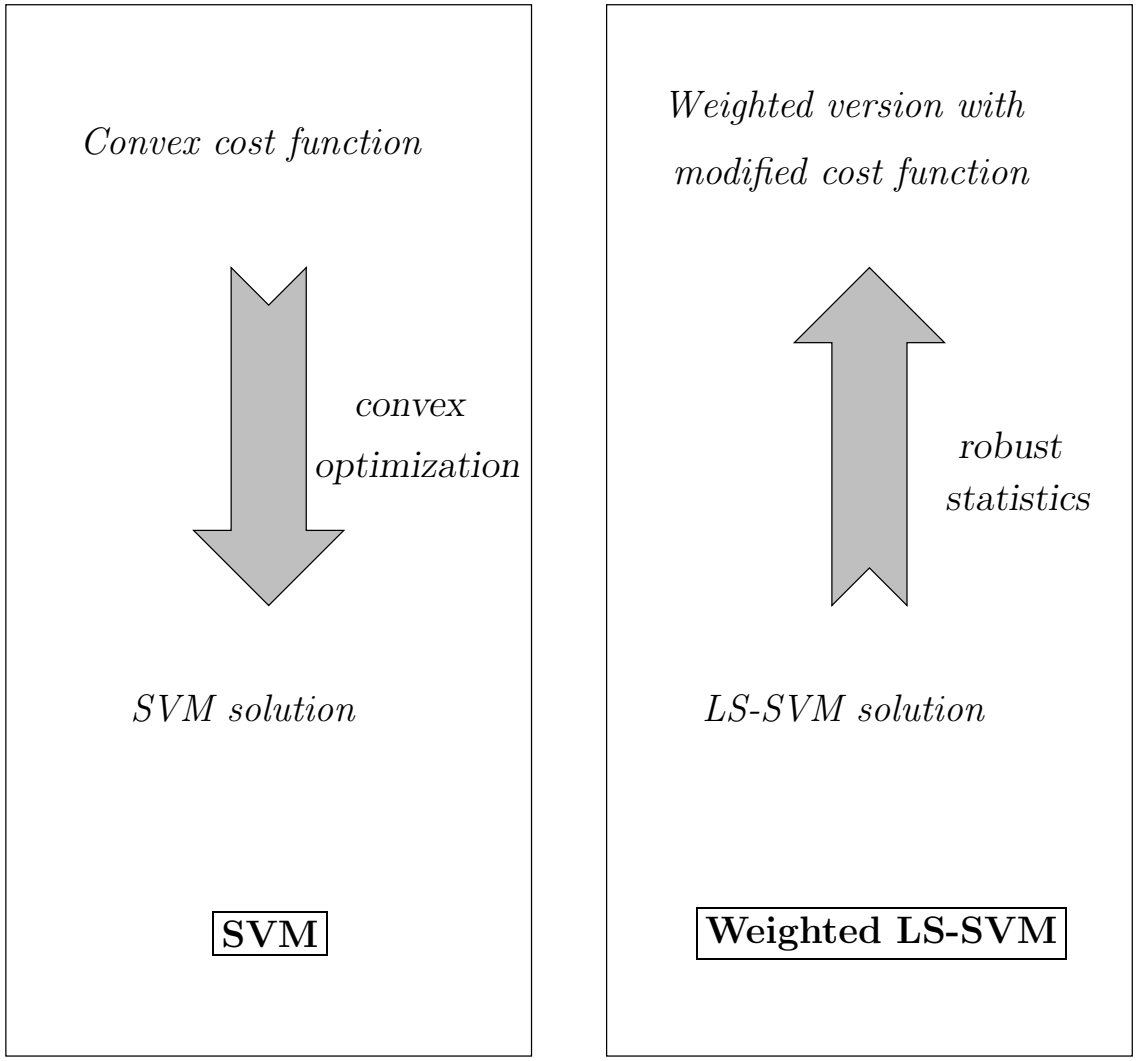


Lack of sparseness in the LS-SVM case

but ... sparseness can be imposed by applying pruning techniques existing in the neural networks area (e.g. optimal brain damage, optimal brain surgeon etc.)



Robustness



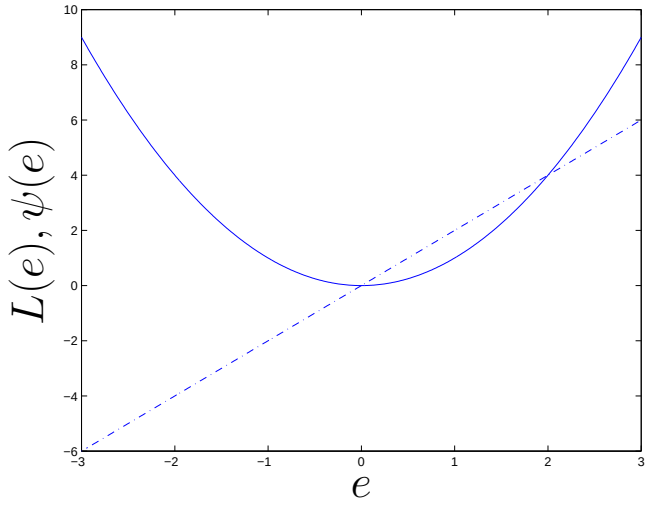
Weighted LS-SVM:

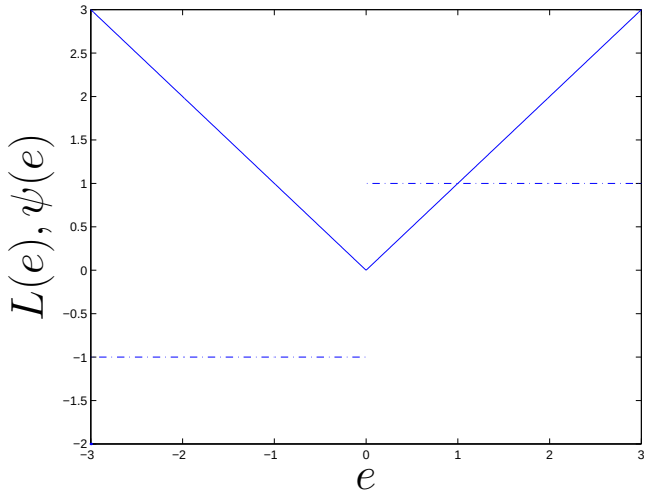
$$\min_{w,b,e} J(w,e) = \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{k=1}^{N} v_k e_k^2$$

$$\text{such that} \quad y_k = w^T\varphi(x_k) + b + e_k,\ k = 1,...,N$$

with v_k determined by the distribution of $\{e_k\}_{k=1}^N$ obtained from the unweighted LS-SVM.

L_2 and L_1 estimators: score function

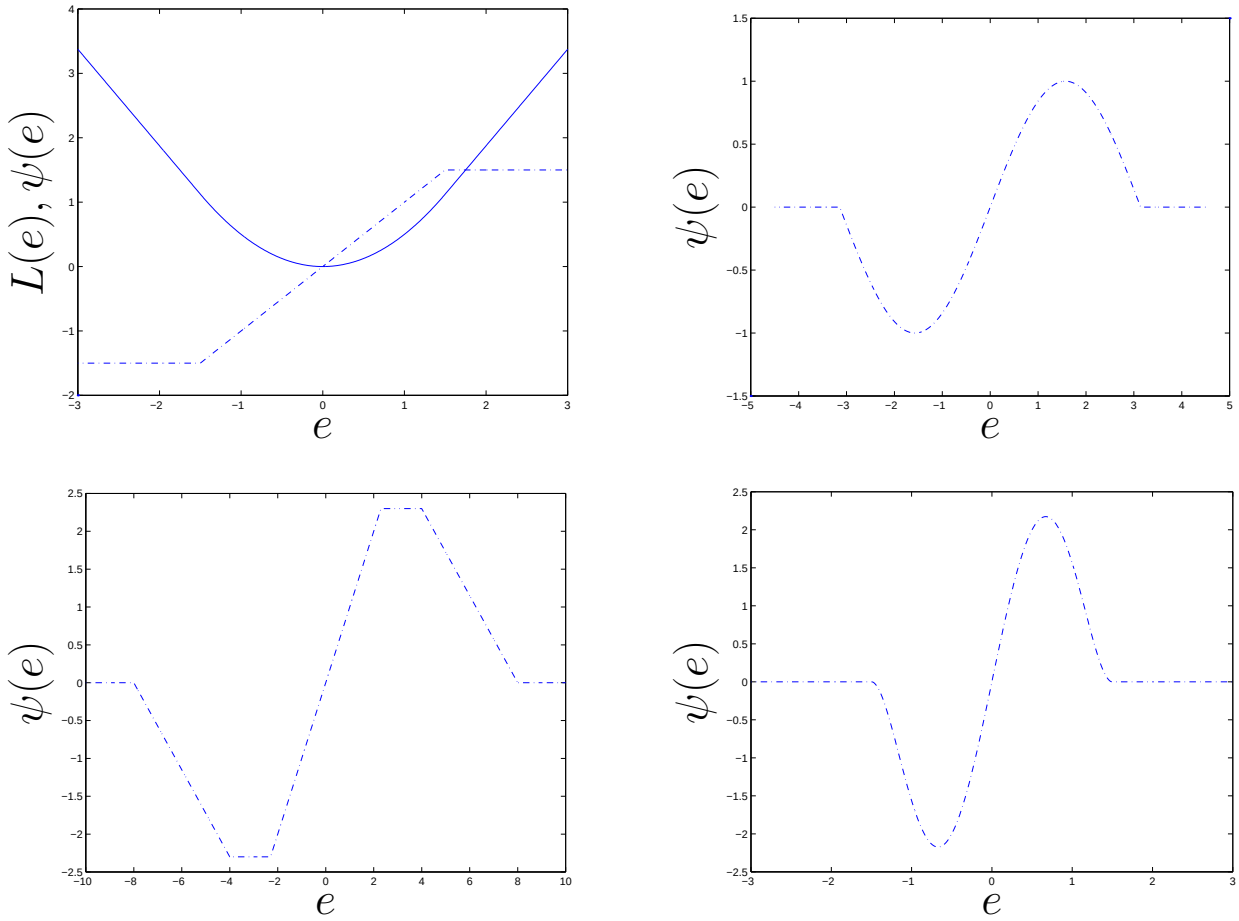




L_1 estimator: influence of outliers is reduced (the score function in robust statistics means derivative of the loss function)

Loss function and score function

Examples of score function in robust statistics:
Huber, Andrews, Hampel, Tukey



In practice: iterative weighting accrording to the score function. Note that all the score functions (except Huber's) lead to non-convex loss functions. Further application of robust statistics to robust cross-validation (De Brabanter *et al.*, 2002) with good performance in terms of efficiency-robustness trade-off.

Large scale methods

- Nyström method (GP)
- Fixed size LS-SVM
- Committee networks and extensions

Nyström method

- Reference in GP: Williams & Seeger (2001)

- *"big"* kernel matrix: $\Omega_{(N,N)} \in \mathbb{R}^{N \times N}$
  "small" kernel matrix: $\Omega_{(M,M)} \in \mathbb{R}^{M \times M}$
  (based on random subsample, in practice often $M \ll N$)

- Eigenvalue decomposition of $\Omega_{(M,M)}$:

$$\Omega_{(M,M)} \, \overline{U} = \overline{U} \, \overline{\Lambda}$$

- Relation to eigenvalues and eigenfunctions of the integral equation

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

  is given by

$$\begin{aligned} \hat{\lambda}_i &= \tfrac{1}{M} \overline{\lambda}_i \\ \hat{\phi}_i(x_k) &= \sqrt{M} \, \overline{u}_{ki} \\ \hat{\phi}_i(x') &= \tfrac{\sqrt{M}}{\overline{\lambda}_i} \sum_{k=1}^{M} \overline{u}_{ki} K(x_k, x') \end{aligned}$$

  where $\hat{\lambda}_i$ and $\hat{\phi}_i$ are estimates to λ_i and ϕ_i, respectively, and $\overline{u}_{ki}$ denotes the ki-th entry of the matrix $\overline{U}$.

- For the big matrix:

$$\Omega_{(N,N)}\,\tilde{U} = \tilde{U}\,\tilde{\Lambda}$$

Furthermore, one has

$$\begin{aligned}\tilde{\lambda}_i &= \tfrac{N}{M}\overline{\lambda}_i \\ \tilde{u}_i &= \sqrt{\tfrac{N}{M}}\tfrac{1}{\overline{\lambda}_i}\Omega_{(N,M)}\overline{u}_i\end{aligned}$$

One can show then that

$$\Omega_{(N,N)} \simeq \Omega_{(N,M)}\Omega^{-1}_{(M,M)}\Omega_{(M,N)}$$

where $\Omega_{(N,M)}$ is the $N \times M$ block matrix taken from $\Omega_{(N,N)}$.

- Solution to the big linear system

$$(\Omega_{(N,N)} + I/\gamma)\alpha = y$$

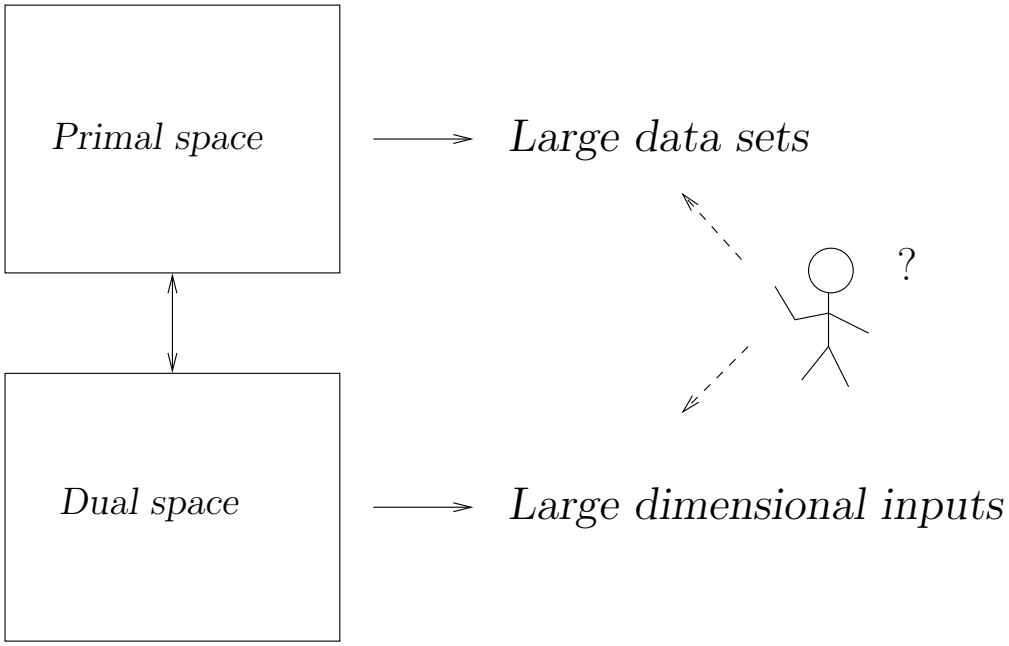
can be written as

$$\alpha = \gamma\left(y - \tilde{U}(\frac{1}{\gamma}I + \tilde{\Lambda}\tilde{U}^T\tilde{U})^{-1}\tilde{\Lambda}\tilde{U}^T y\right)$$

by applying Sherman-Morrison-Woodbury formula.

- Some numerical difficulties as pointed out by Fine & Scheinberg (2001).

Computation in primal or dual space ?



Fixed Size LS-SVM

- Related work: basis construction in feature space
  Cawley (2002), Csato & Opper (2002), Smola & Schölkopf (2002), Baudat & Anouar (2001).

- Model in primal space:

$$\min_{w\in\mathbb{R}^{n_h},b\in\mathbb{R}} \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{k=1}^{N}\left(y_k - (w^T\varphi(x_k)+b)\right)^2.$$

  Observation: for linear model it is computationally better to solve the primal problem (one knows that $\varphi(x_k) = x_k$)

- Can we do this for the nonlinear case too ?

- Employ the Nyström method:

$$\varphi_i(x') = \sqrt{\tilde{\lambda}_i}\,\hat{\phi}_i(x') = \frac{\sqrt{M}}{\sqrt{\tilde{\lambda}_i}}\sum_{k=1}^{M} u_{ki}K(x_k,x'),$$

  assuming a fixed size M.

- The model becomes then

$$\begin{aligned} y(x) &= w^T\varphi(x)+b \\ &= \sum_{i=1}^{M} w_i \frac{\sqrt{M}}{\sqrt{\tilde{\lambda}_i}}\sum_{k=1}^{M} u_{ki}K(x_k,x). \end{aligned}$$

The support values corresponding to the number of M support vectors equal

$$\alpha_k = \sum_{i=1}^{M} w_i \frac{\sqrt{M}}{\sqrt{\tilde{\lambda}_i}} u_{ki}$$

when ones represent the model as

$$y(x) = \sum_{k=1}^{M} \alpha_k K(x_k, x)$$

- How to select a working set of M support vectors ?

  Is taking a random subsample the only option ?

Fixed Size LS-SVM: Selection of SV

- Girolami (2002): link between Nyström method, kernel PCA, density estimation and entropy criteria

  The quadratic Renyi entropy

  $$H_R = -\log \int p(x)^2 dx$$

  has been related to kernel PCA and density estimation with

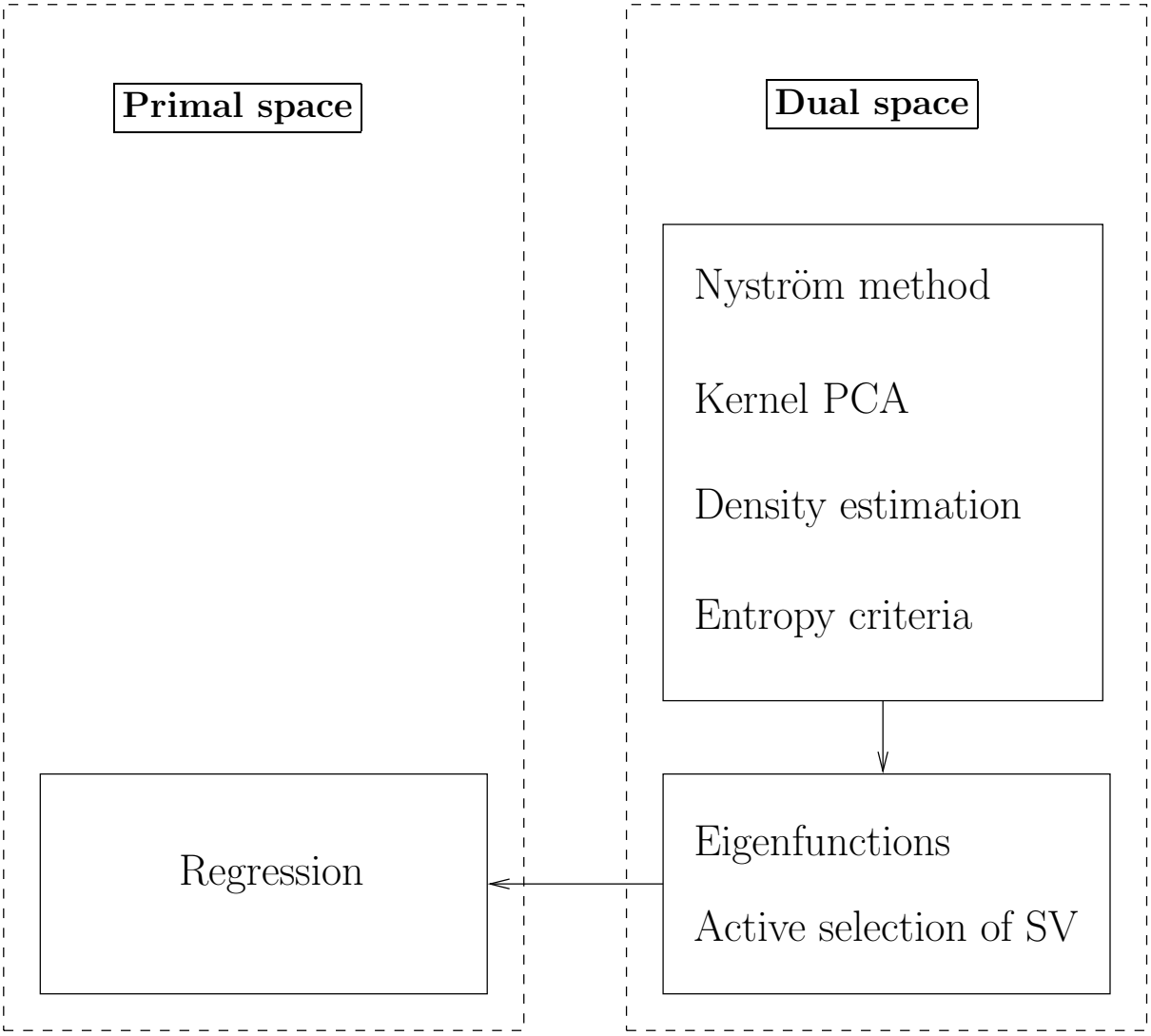
  $$\int \hat{p}(x)^2 dx = \frac{1}{N^2} 1_v^T \Omega 1_v$$

  where $1_v = [1; 1; ...; 1]$ and a normalized kernel is assumed with respect to density estimation.

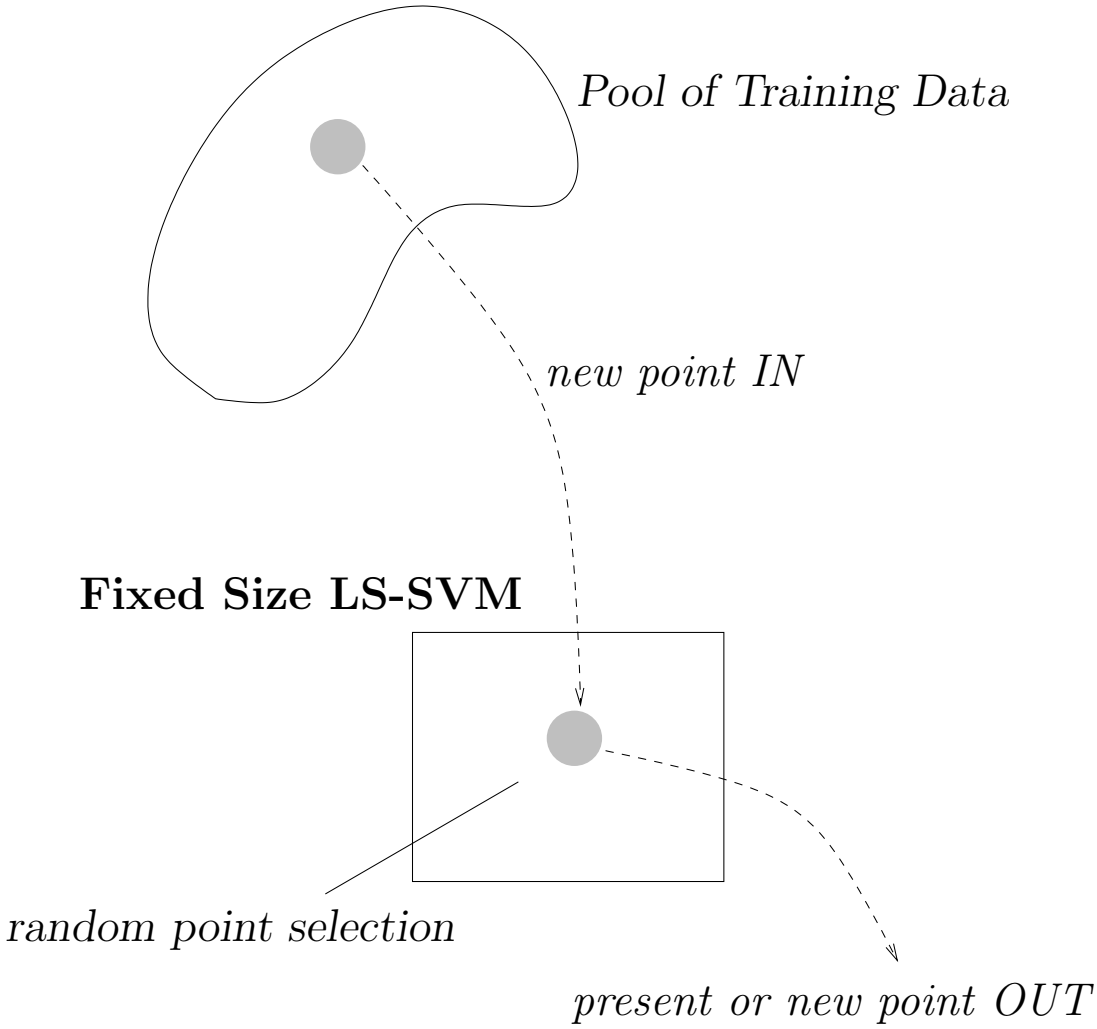
- Fixed Size LS-SVM:

  Take a working set of M support vectors and select vectors according to the entropy criterion (instead of a random subsample as in the Nyström method)

Fixed Size LS-SVM



Fixed Size LS-SVM



- **Fixed size LS-SVM algorithm:**

1. Given normalized and standardized training data $\{x_k, y_k\}_{k=1}^N$ with inputs $x_k \in \mathbb{R}^n$, outputs $y_k \in \mathbb{R}$ and N training data.

2. Choose a working set with size M and impose in this way a number of M support vectors (typically $M \ll N$).

3. Randomly select a support vector x^* from the working set of M support vectors.

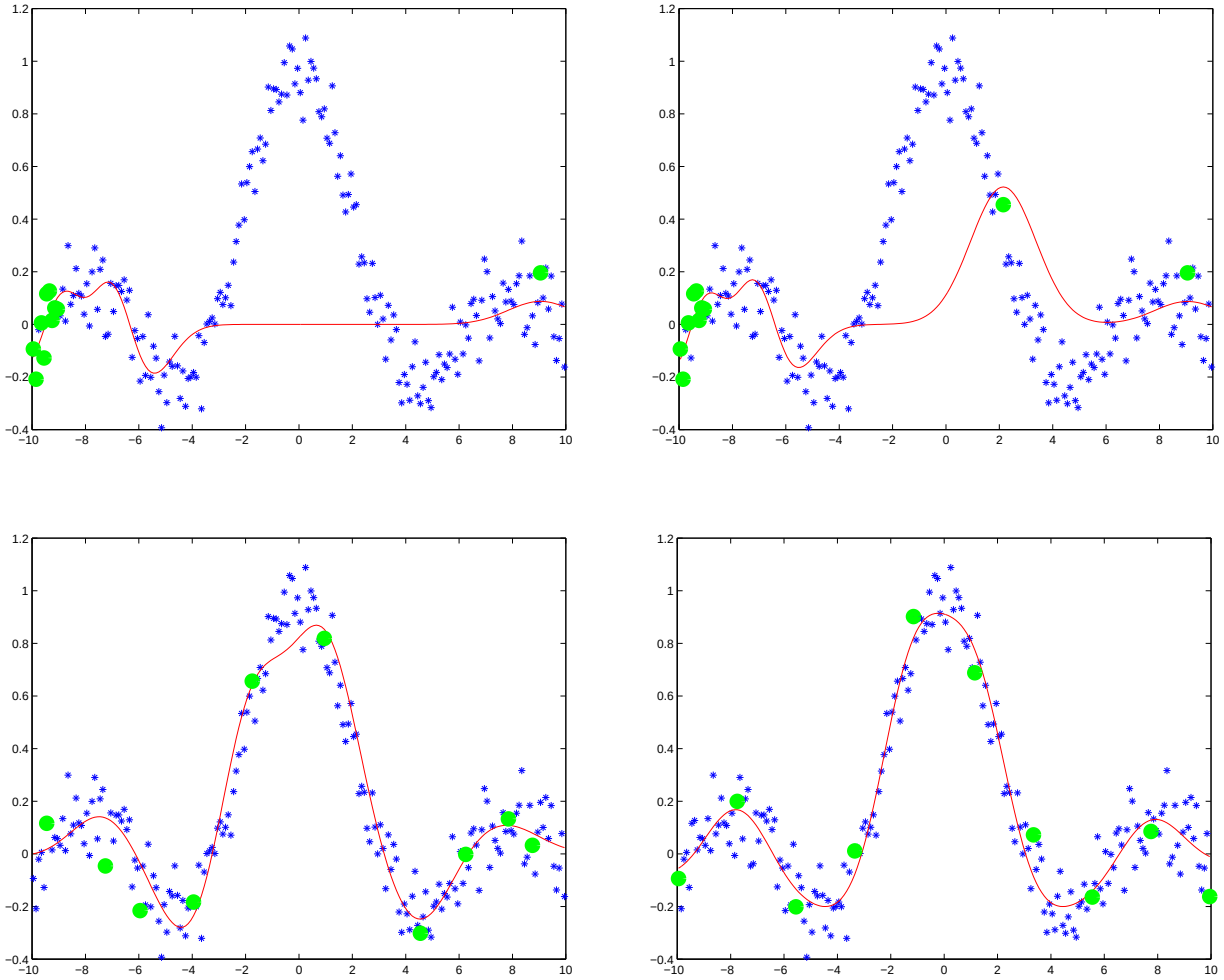
4. Randomly select a point x^{t*} from the N training data and replace x^* by x^{t*} in the working set. If the entropy increases by taking the point x^{t*} instead of x^* then this point x^{t*} is accepted for the working set of M support vectors, otherwise the point x^{t*} is rejected (and returned to the training data pool) and the support vector x^* stays in the working set.

5. Calculate the entropy value for the present working set.

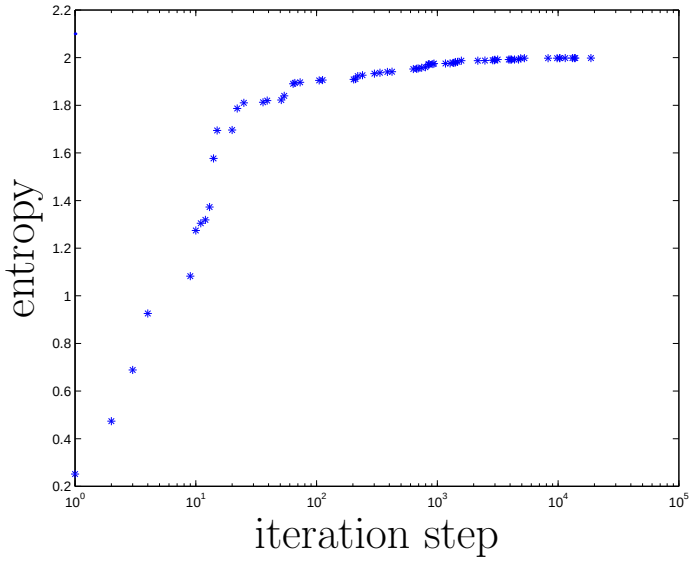
6. Stop if the change in entropy value is small or the number of iterations is exceeded, otherwise go to (3).

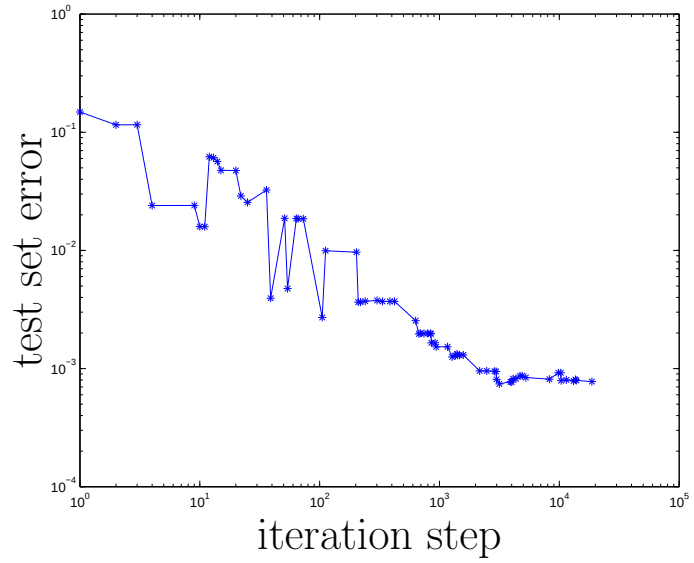
7. Estimate w, b in the primal space after estimating the eigenfunctions from the Nyström approximation.

Fixed Size LS-SVM: sinc example

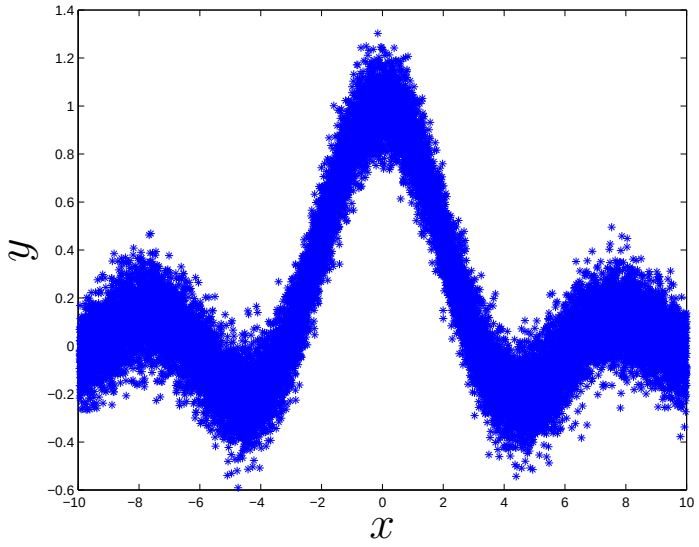


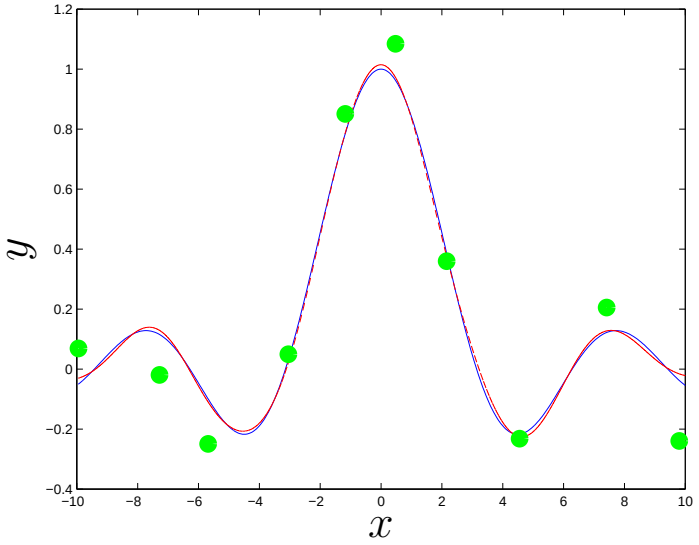
demo in LS-SVMlab


entropy
iteration step


test set error
iteration step

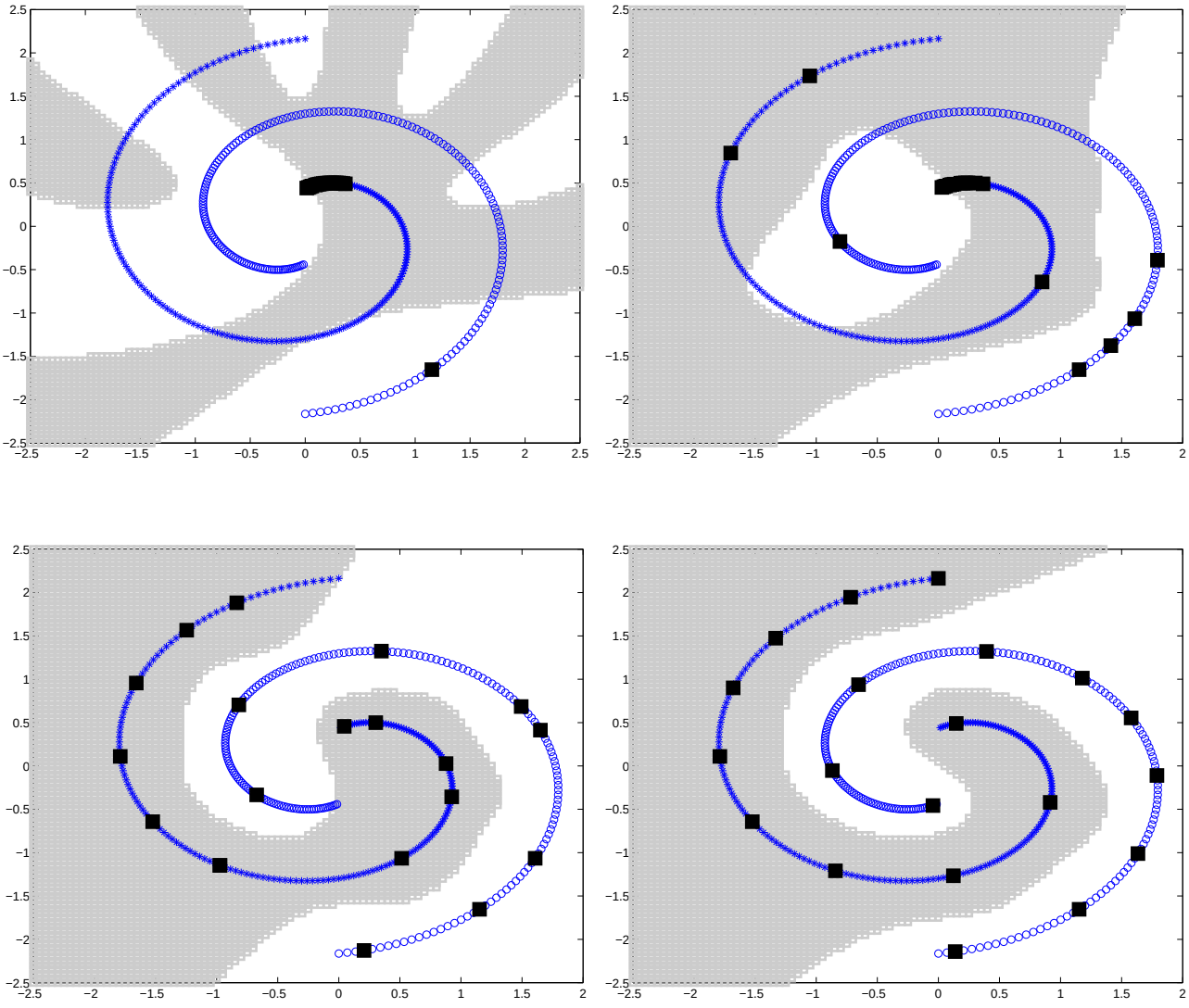
Fixed Size LS-SVM: sinc example



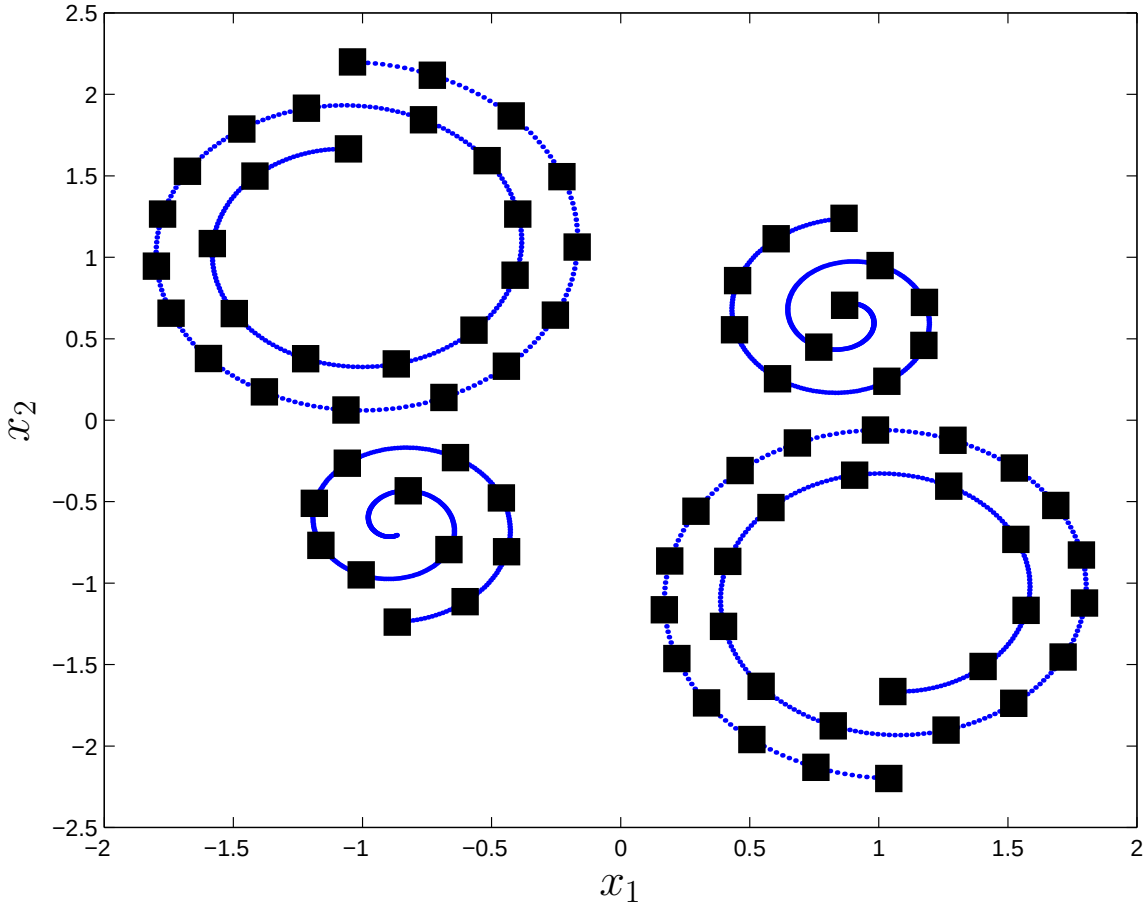


Sinc function with 20000 data points

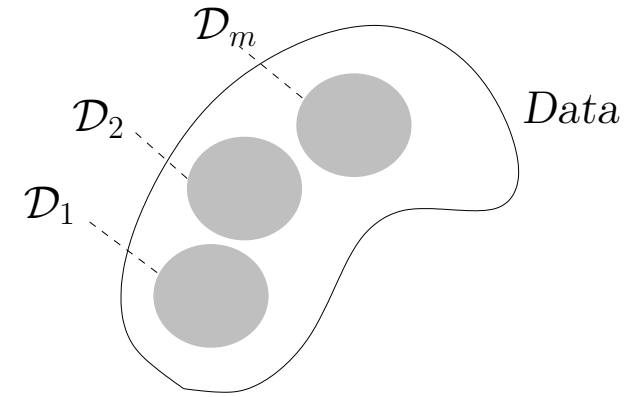
Fixed Size LS-SVM: spiral example

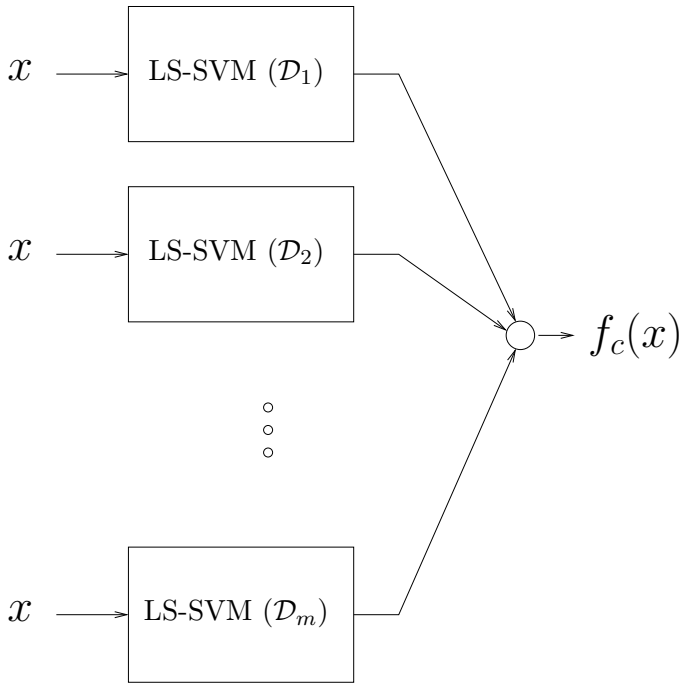


Fixed Size LS-SVM: spiral example



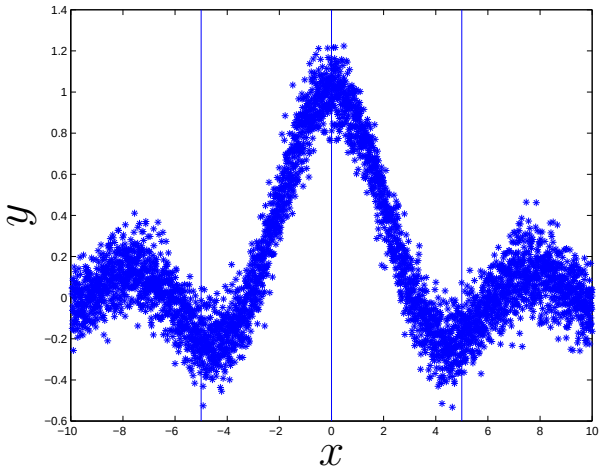
Committee network of LS-SVMs

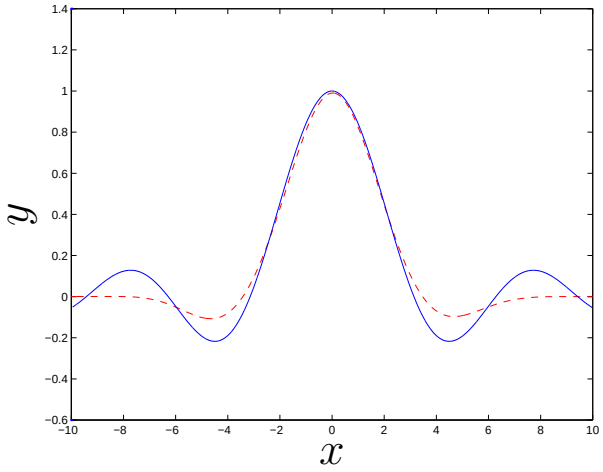




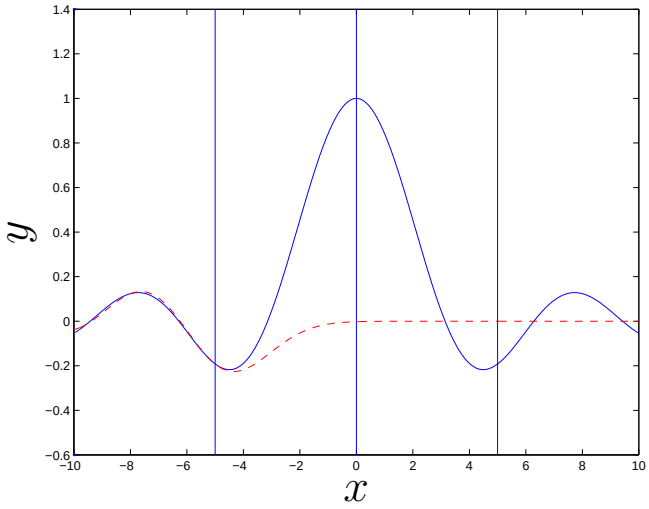
Committee network of LS-SVMs

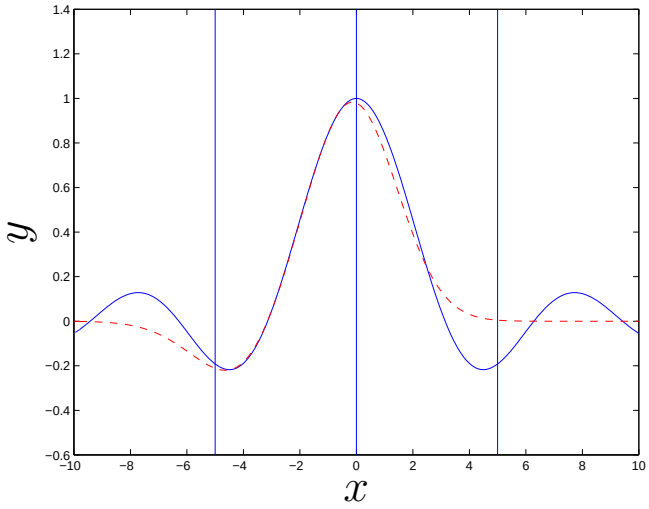
sinc function with 4000 training data

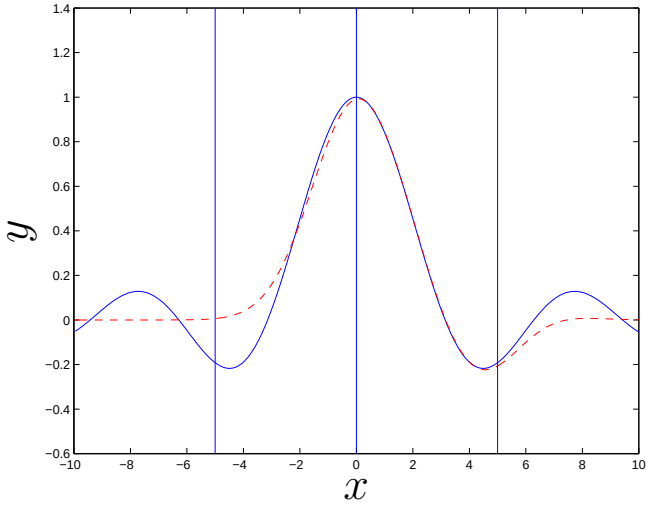


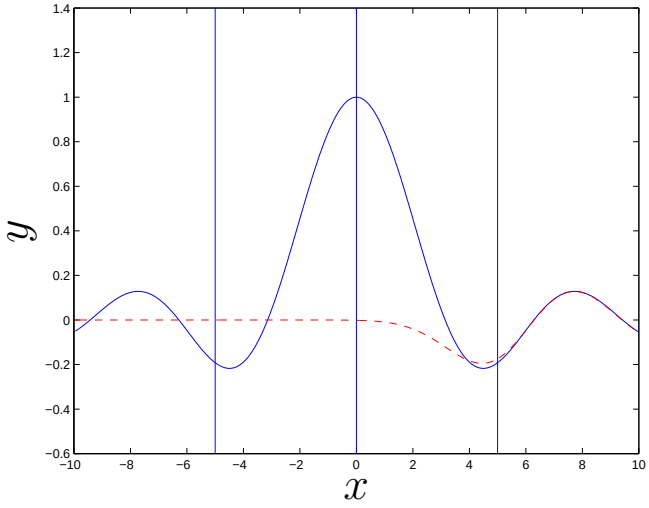


Results of the individual LS-SVM models

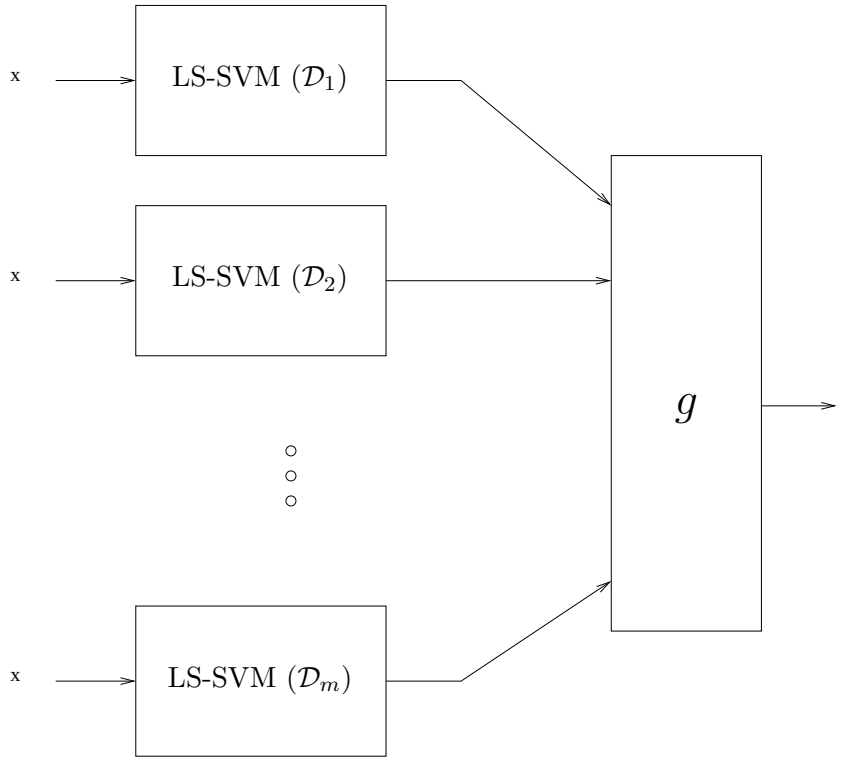








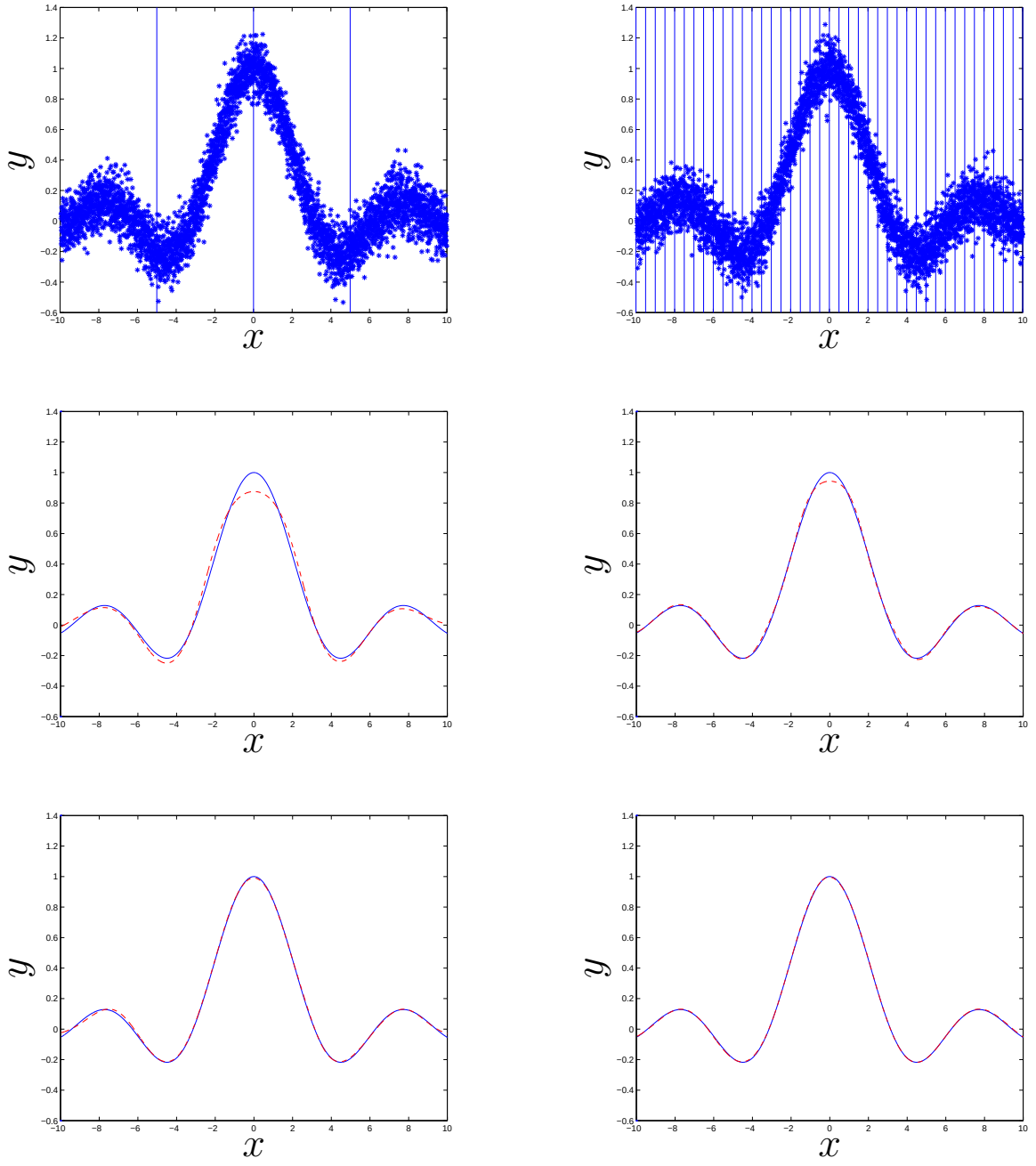
Nonlinear combination of LS-SVMs



This results into a multilayer network
(layers of (LS)-SVMs or e.g. MLP + LS-SVM combination)

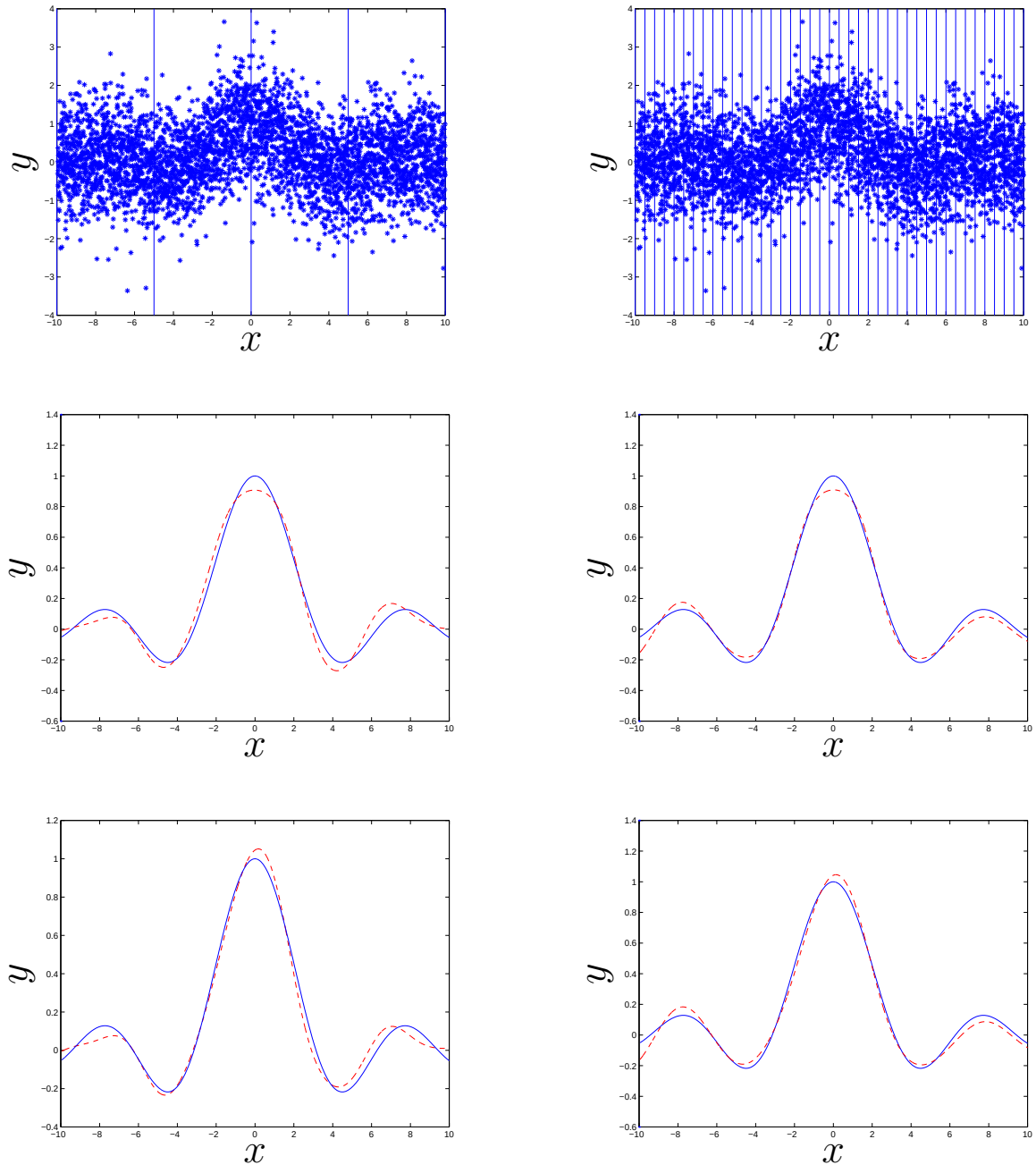
Committee network of LS-SVMs

Linear versus nonlinear combinations of trained LS-SVM submodels



Committee network of LS-SVMs

Linear versus nonlinear combinations of trained LS-SVM submodels under heavy Gaussian noise



Classical PCA formulation

- Given data $\{x_k\}_{k=1}^N$ with $x_k \in \mathbb{R}^n$
- Find projected variables $w^T x_k$ with maximal variance

$$\begin{aligned} \max_w \mathrm{Var}(w^T x) &= \mathrm{Cov}(w^T x, w^T x) \simeq \frac{1}{N}\sum_{k=1}^{N}(w^T x_k)^2 \\ &= w^T C\, w \end{aligned}$$

  where $C = (1/N)\sum_{k=1}^{N} x_k x_k^T$.

  Consider constraint $w^T w = 1$.

- Constrained optimization:

$$\mathcal{L}(w;\lambda) = \frac{1}{2} w^T C w - \lambda(w^T w - 1)$$

  with Lagrange multiplier λ.

- Eigenvalue problem

$$Cw = \lambda w$$

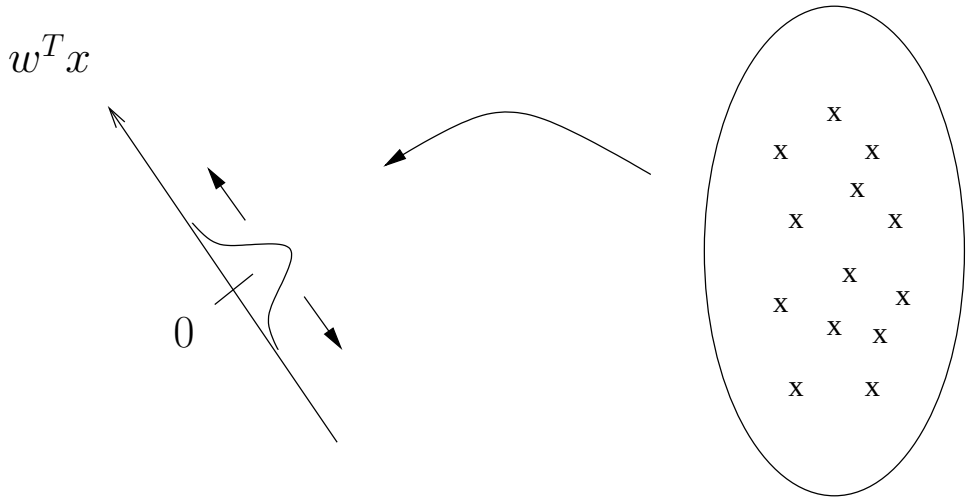
  with $C = C^T \geq 0$, obtained from $\partial\mathcal{L}/\partial w = 0$, $\partial\mathcal{L}/\partial\lambda = 0$.

PCA analysis as a one-class modelling problem

- One-class with target value zero:

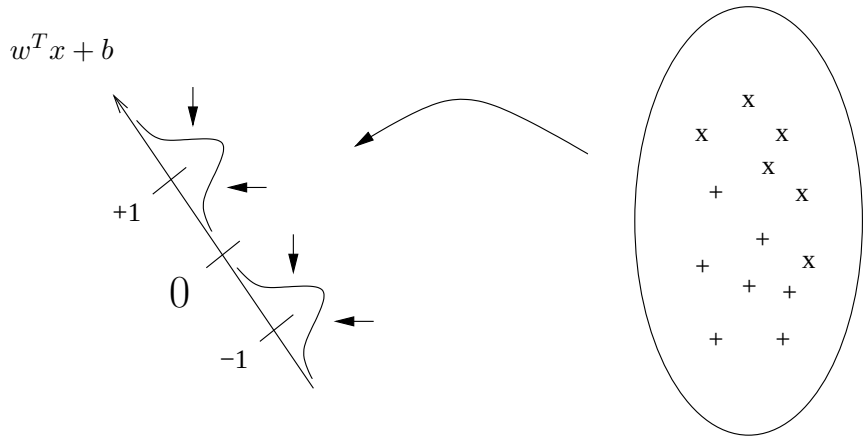
$$\max_{w} \sum_{k=1}^{N} (0 - w^T x_k)^2$$

- Score variables: $z = w^T x$
- Illustration:



Target space *Input space*

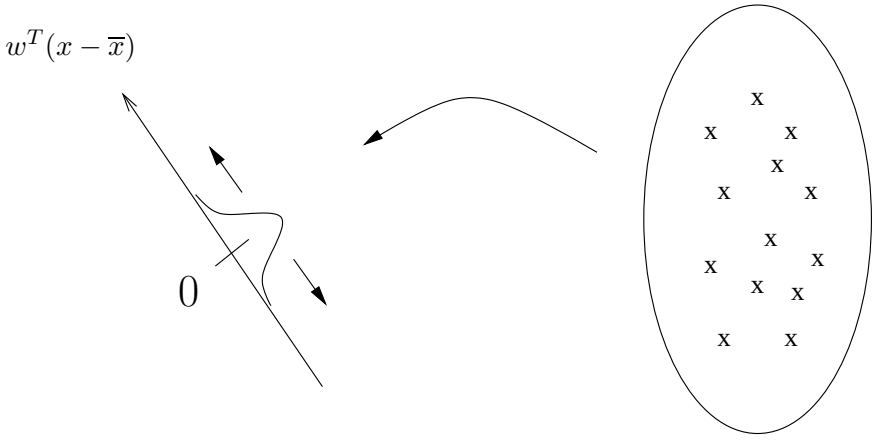
LS-SVM interpretation to FDA



Target space *Input space*

Minimize within class scatter

LS-SVM interpretation to PCA



Target space *Input space*

Find direction with maximal variance

SVM formulation to linear PCA

- Primal problem:

$$\boxed{\text{P}}: \max_{w,e} J_{\text{P}}(w,e) = \gamma \frac{1}{2}\sum_{k=1}^{N} e_k^2 - \frac{1}{2} w^T w$$
$$\text{such that} \qquad e_k = w^T x_k, \;\; k = 1, ..., N$$

- Lagrangian

$$\mathcal{L}(w, e; \alpha) = \gamma \frac{1}{2}\sum_{k=1}^{N} e_k^2 - \frac{1}{2} w^T w - \sum_{k=1}^{N} \alpha_k \left(e_k - w^T x_k \right)$$

- Conditions for optimality

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \;\rightarrow\; w = \sum_{k=1}^{N} \alpha_k x_k \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\rightarrow\; \alpha_k = \gamma e_k, \qquad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\rightarrow\; e_k - w^T x_k = 0, \;\; k = 1, ..., N \end{cases}$$

- Elimination of variables e, w gives

$$\frac{1}{\gamma}\alpha_k - \sum_{l=1}^{N} \alpha_l x_l^T x_k = 0 \ , \quad k = 1, ..., N$$

and after defining $\lambda = 1/\gamma$ one obtains the eigenvalue problem

$\boxed{\text{D}}$: solve in α :

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

as the dual problem (quantization in terms of $\lambda = 1/\gamma$).

- Score variables become

$$z(x) = w^T x = \sum_{l=1}^{N} \alpha_l x_l^T \, x$$

- Optimal solution corresponding to largest eigenvalue

$$\sum_{k=1}^{N} (w^T x_k)^2 = \sum_{k=1}^{N} e_k^2 = \sum_{k=1}^{N} \frac{1}{\gamma^2} \alpha_k^2 = \lambda_{max}^2$$

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

- Many data : better solve primal problem
  Many inputs: better solve dual problem

SVM formulation to PCA analysis with a bias term

- Usually: apply PCA analysis to *centered data* and consider

$$\max_{w} \sum_{k=1}^{N} [w^T (x_k - \hat{\mu}_x)]^2$$

where $\hat{\mu}_x = \dfrac{1}{N} \sum_{k=1}^{N} x_k$.

- Bias term formulation:

score variables

$$z(x) = w^T x + b$$

and objective

$$\max_{w,b} \sum_{k=1}^{N} [0 - (w^T x + b)]^2$$

- Primal optimization problem

$$\boxed{\text{P}}: \max_{w,e} J_{\text{P}}(w,e) = \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \frac{1}{2}w^T w$$
$$\text{such that} \qquad e_k = w^T x_k + b, \quad k = 1, ..., N$$

- Lagrangian

$$\mathcal{L}(w, e; \alpha) = \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \frac{1}{2}w^T w - \sum_{k=1}^{N} \alpha_k \left(e_k - w^T x_k - b\right)$$

- Conditions for optimality

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \;\rightarrow\; w = \sum_{k=1}^{N} \alpha_k x_k \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\rightarrow\; \alpha_k = \gamma e_k, \qquad\qquad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\rightarrow\; \sum_{k=1}^{N} \alpha_k = 0 \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\rightarrow\; e_k - w^T x_k - b = 0, \;\; k = 1, ..., N \end{cases}$$

- Applying $\sum_{k=1}^{N} \alpha_k = 0$ yields

$$b = -\frac{1}{N}\sum_{k=1}^{N}\sum_{l=1}^{N} \alpha_l x_l^T x_k.$$

- By defining $\lambda = 1/\gamma$ one obtains the dual problem

D : solve in α :

$$\begin{bmatrix} (x_1 - \hat{\mu}_x)^T(x_1 - \hat{\mu}_x) & \dots & (x_1 - \hat{\mu}_x)^T(x_N - \hat{\mu}_x) \\ \vdots & & \vdots \\ (x_N - \hat{\mu}_x)^T(x_1 - \hat{\mu}_x) & \dots & (x_N - \hat{\mu}_x)^T(x_N - \hat{\mu}_x) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} = \lambda \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}$$

which is an eigenvalue decomposition of the centered Gram matrix

$$\Omega_c \alpha = \lambda \alpha$$

with $\Omega_c = M_c \Omega M_c$ where $M_c = I - 1_v 1_v^T / N$, $1_v = [1; 1; ...; 1]$ and $\Omega_{kl} = x_k^T x_l$ for $k, l = 1, ..., N$.

- Score variables:

$$z(x) = w^T x + b = \sum_{l=1}^{N} \alpha_l x_l^T x + b$$

where α is the eigenvector corresponding to the largest eigenvalue and

$$\sum_{k=1}^{N} (w^T x_k + b)^2 = \sum_{k=1}^{N} e_k^2 = \sum_{k=1}^{N} \frac{1}{\gamma^2} \alpha_k^2 = \lambda_{max}^2$$

Reconstruction problem for linear PCA

- Reconstruction error:

$$\min \sum_{k=1}^{N} \|x_k - \tilde{x}_k\|_2^2$$

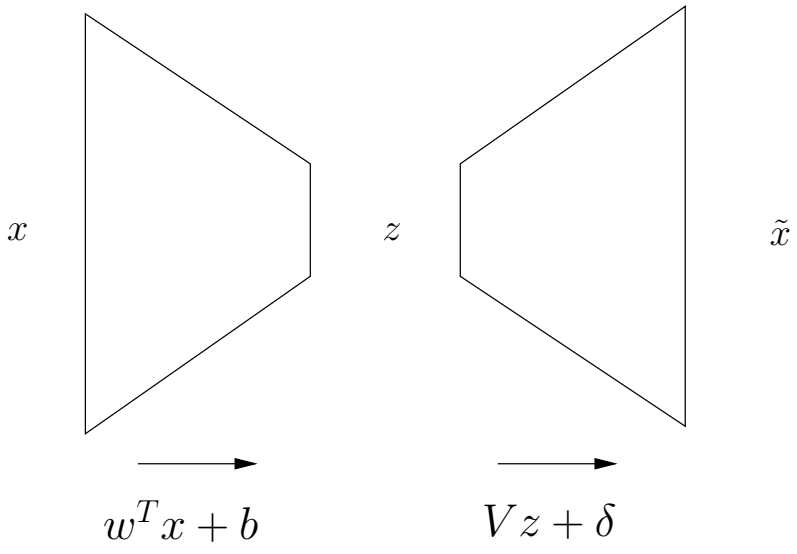
where $\tilde{x}_k$ are variables reconstructed from the score variables, with

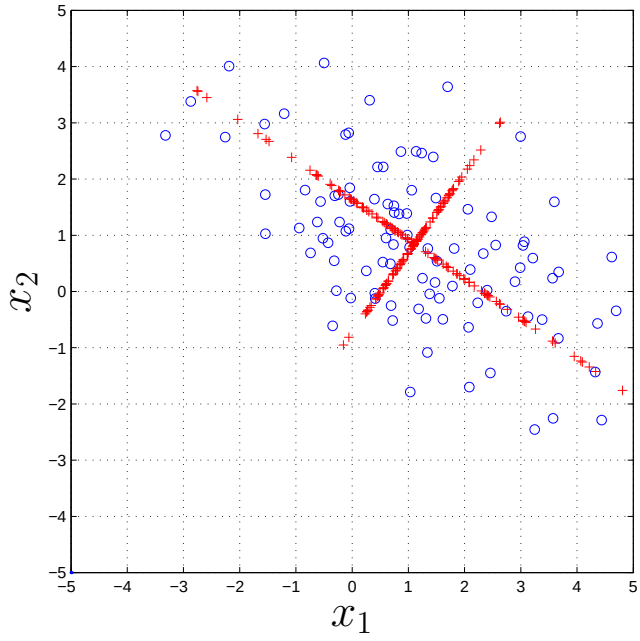
$$\tilde{x} = Vz + \delta$$

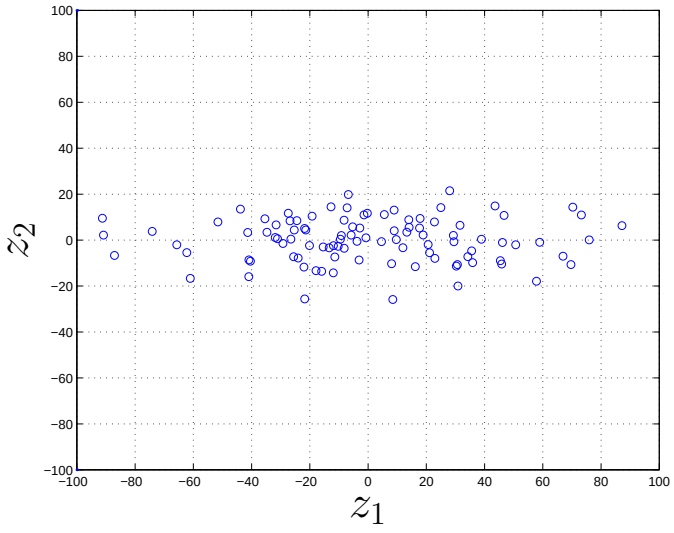
Hence

$$\min_{V,\delta} \sum_{k=1}^{N} \|x_k - (Vz_k + \delta)\|_2^2$$

- Information bottleneck:




5
4
3
2
1
0
−1
−2
−3
−4
−5
−5
−4
−3
−2
−1
0
1
2
3
4
5
x_2
x_1


100
80
60
40
20
0
−20
−40
−60
−80
−100
−100
−80
−60
−40
−20
0
20
40
60
80
100
z_2
z_1

LS-SVM approach to kernel PCA

- Create nonlinear version of the method by

  Mapping input space to a high dimensional feature space
  Applying the kernel trick

  (kernel PCA - Schölkopf *et al.*; SVM approach - Suykens *et al.*, 2002)

- Primal optimization problem:

$$\boxed{\text{P}}: \max_{w,e} J_{\text{P}}(w,e) = \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \frac{1}{2}w^T w$$
$$\text{such that} \qquad e_k = w^T(\varphi(x_k) - \hat{\mu}_\varphi),\ k = 1,...,N.$$

- Lagrangian

$$\mathcal{L}(w,e;\alpha) = \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 - \frac{1}{2}w^T w - \sum_{k=1}^{N}\alpha_k\left(e_k - w^T(\varphi(x_k) - \hat{\mu}_\varphi)\right)$$

- Conditions for optimality

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial w} = 0 \ \rightarrow\ w = \sum_{k=1}^{N}\alpha_k(\varphi(x_k) - \hat{\mu}_\varphi) & \\ \frac{\partial\mathcal{L}}{\partial e_k} = 0 \ \rightarrow\ \alpha_k = \gamma e_k, & k = 1,...,N \\ \frac{\partial\mathcal{L}}{\partial \alpha_k} = 0 \ \rightarrow\ e_k - w^T(\varphi(x_k) - \hat{\mu}_\varphi) = 0, & k = 1,...,N. \end{cases}$$

- By elimination of the variables e, w and defining $\lambda = 1/\gamma$ one obtains

$$\boxed{\text{D}}: \text{ solve in } \alpha:$$

$$\Omega_c \alpha = \lambda \alpha$$

with

$$\Omega_c = \begin{bmatrix} (\varphi(x_1) - \hat{\mu}_\varphi)^T(\varphi(x_1) - \hat{\mu}_\varphi) & \dots & (\varphi(x_1) - \hat{\mu}_\varphi)^T(\varphi(x_N) - \hat{\mu}_\varphi) \\ \vdots & & \vdots \\ (\varphi(x_N) - \hat{\mu}_\varphi)^T(\varphi(x_1) - \hat{\mu}_\varphi) & \dots & (\varphi(x_N) - \hat{\mu}_\varphi)^T(\varphi(x_N) - \hat{\mu}_\varphi) \end{bmatrix}$$

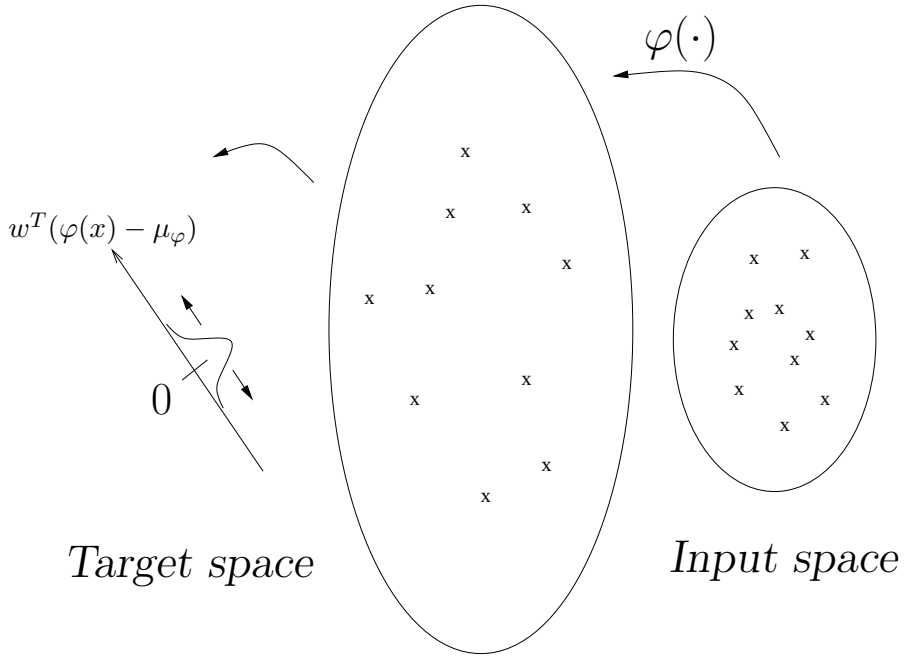
Elements of the centered kernel matrix

$$\Omega_{c,kl} = (\varphi(x_k) - \hat{\mu}_\varphi)^T(\varphi(x_l) - \hat{\mu}_\varphi), \quad k, l = 1, ..., N$$

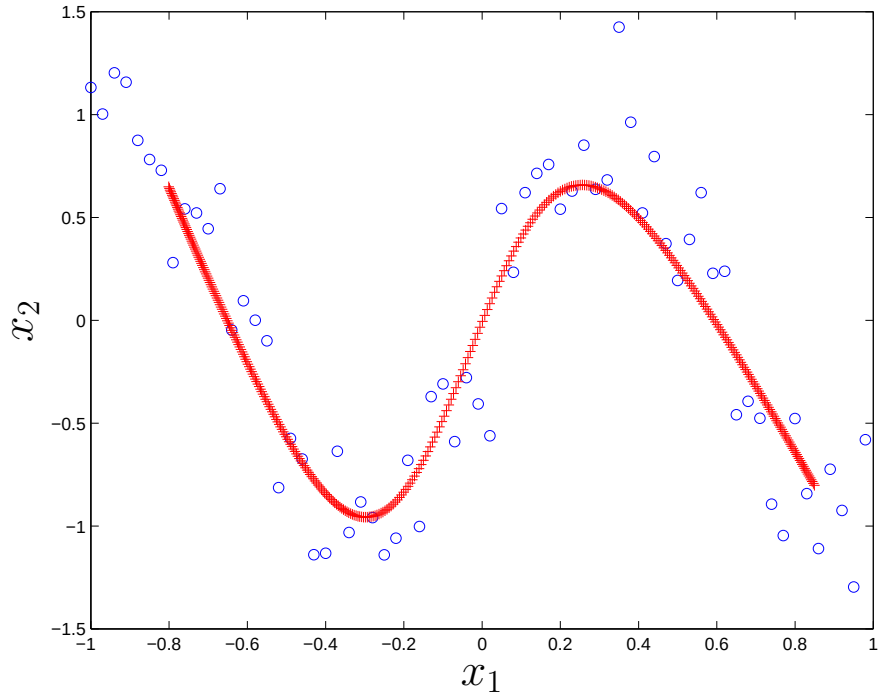
- Score variables

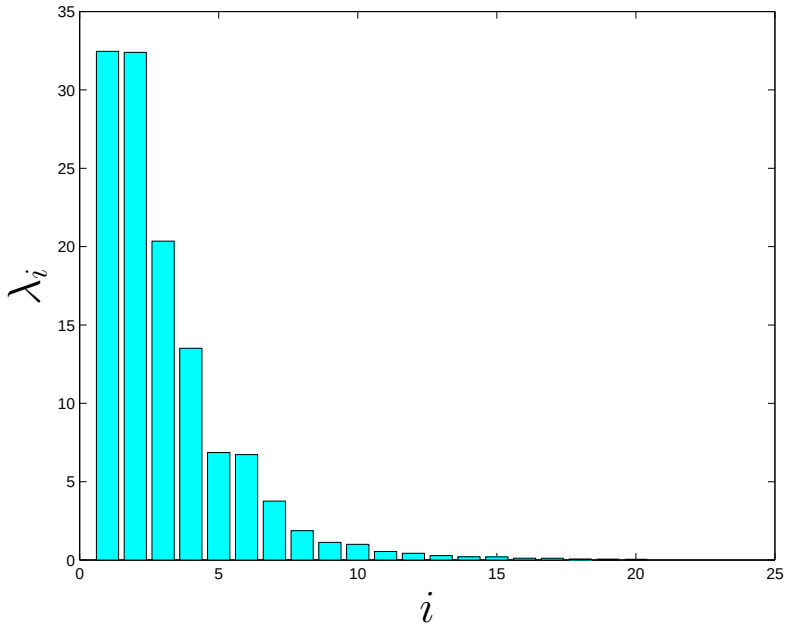
$$\begin{aligned} z(x) &= w^T(\varphi(x) - \hat{\mu}_\varphi) \\ &= \sum_{l=1}^{N} \alpha_l (\varphi(x_l) - \hat{\mu}_\varphi)^T (\varphi(x) - \hat{\mu}_\varphi) \\ &= \sum_{l=1}^{N} \alpha_l \left(K(x_l, x) - \tfrac{1}{N} \sum_{r=1}^{N} K(x_r, x) - \tfrac{1}{N} \sum_{r=1}^{N} K(x_r, x_l) + \right. \\ &\quad \left. \frac{1}{N^2} \sum_{r=1}^{N} \sum_{s=1}^{N} K(x_r, x_s) \right). \end{aligned}$$

LS-SVM interpretation to Kernel PCA



Find direction with maximal variance


1.5
1
0.5
0
−0.5
−1
−1.5
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
x_2
x_1


35
30
25
20
15
10
5
0
0
5
10
15
20
25
λ_i
i

Canonical Correlation Analysis

- Hotelling (1936): linear CCA analysis

  Applications e.g. in system identification, subspace algorithms, signal processing and information theory.
  New applications to bioinformatics (correlations between genes and between experiments in both directions for microarray data matrix)

- Finding maximal correlation between projected variables

$$z_x = w^T x \text{ and } z_y = v^T y$$

  where $x \in \mathbb{R}^{n_x}, y \in \mathbb{R}^{n_y}$ denote given random vectors with zero mean.

- Objective: achieve maximal correlation coefficient

$$\begin{aligned} \max_{w,v} \rho &= \frac{\mathcal{E}[z_x z_y]}{\sqrt{\mathcal{E}[z_x z_x]}\sqrt{\mathcal{E}[z_y z_y]}} \\ &= \frac{w^T C_{\text{xy}} v}{\sqrt{w^T C_{\text{xx}} w}\sqrt{v^T C_{\text{yy}} v}} \end{aligned}$$

  with $C_{\text{xx}} = \mathcal{E}[xx^T]$, $C_{\text{yy}} = \mathcal{E}[yy^T]$, $C_{\text{xy}} = \mathcal{E}[xy^T]$.

- This is formulated as the constrained optimization problem

$$\begin{aligned} \max_{w,v} \quad & w^T C_{\mathrm{xy}} v \\ \text{such that} \quad & w^T C_{\mathrm{xx}} w = 1 \\ & v^T C_{\mathrm{yy}} v = 1 \end{aligned}$$

  which leads to a generalized eigenvalue problem.

- The solution follows from the Lagrangian

$$\mathcal{L}(w, v; \eta, \nu) = w^T C_{\mathrm{xy}} v - \eta \frac{1}{2}(w^T C_{\mathrm{xx}} w - 1) - \nu \frac{1}{2}(v^T C_{\mathrm{yy}} v - 1)$$

  with Lagrange multipliers η, ν, which gives

$$\begin{cases} C_{\mathrm{xy}} v &= \eta\, C_{\mathrm{xx}} w \\ C_{\mathrm{yx}} w &= \nu\, C_{\mathrm{yy}} v. \end{cases}$$

SVM formulation to CCA

- In primal weight space:

$$\boxed{\text{P}}: \quad \max_{w,v,e,r} \quad \gamma \sum_{k=1}^{N} e_k r_k - \nu_1 \frac{1}{2} \sum_{k=1}^{N} e_k^2 - \nu_2 \frac{1}{2} \sum_{k=1}^{N} r_k^2 - \frac{1}{2} w^T w - \frac{1}{2} v^T v$$

$$\text{such that } e_k = w^T x_k, \qquad k = 1, ..., N$$

$$r_k = v^T y_k \qquad k = 1, ..., N$$

- Lagrangian:

$$\mathcal{L}(w, v, e, r; \alpha, \beta) = \gamma \sum_{k=1}^{N} e_k r_k - \nu_1 \frac{1}{2} \sum_{k=1}^{N} e_k^2 - \nu_2 \frac{1}{2} \sum_{k=1}^{N} r_k^2$$

$$- \frac{1}{2} w^T w - \frac{1}{2} v^T v - \sum_{k=1}^{N} \alpha_k [e_k - w^T x_k] - \sum_{k=1}^{N} \beta_k [r_k - v^T y_k]$$

  where α_k, β_k are Lagrange multipliers.

- Related objective in CCA literature (Gittins, 1985):

$$\min_{w,v} \sum_{k} \| w^T x_k - v^T y_k \|_2^2$$

- Conditions for optimality

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \;\rightarrow\; w = \sum_{k=1}^{N} \alpha_k x_k \\ \frac{\partial \mathcal{L}}{\partial v} = 0 \;\rightarrow\; v = \sum_{k=1}^{N} \beta_k y_k \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\rightarrow\; \gamma v^T y_k = \nu_1 w^T x_k + \alpha_k, \quad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial r_k} = 0 \;\rightarrow\; \gamma w^T x_k = \nu_2 v^T y_k + \beta_k, \quad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\rightarrow\; e_k = w^T x_k, \quad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \beta_k} = 0 \;\rightarrow\; r_k = v^T y_k, \quad k = 1, ..., N \end{cases}$$

resulting into the following dual problem after defining $\lambda = 1/\gamma$

$\boxed{\text{D}}$: solve in α, β :

$$\left[\begin{array}{ccc|ccc} & & & y_1^T y_1 & \dots & y_1^T y_N \\ & 0 & & \vdots & & \vdots \\ & & & y_N^T y_1 & \dots & y_N^T y_N \\ \hline x_1^T x_1 & \dots & x_1^T x_N & & & \\ \vdots & & \vdots & & 0 & \\ x_N^T x_1 & \dots & x_N^T x_N & & & \end{array}\right] \left[\begin{array}{c} \alpha_1 \\ \vdots \\ \alpha_N \\ \hline \beta_1 \\ \vdots \\ \beta_N \end{array}\right]$$

$$= \lambda \left[\begin{array}{ccc|ccc} \nu_1 x_1^T x_1 + 1 & \dots & \nu_1 x_1^T x_N & & & \\ \vdots & & \vdots & & 0 & \\ \nu_1 x_N^T x_1 & \dots & \nu_1 x_N^T x_N + 1 & & & \\ \hline & & & \nu_2 y_1^T y_1 + 1 & \dots & \nu_2 y_1^T y_N \\ & 0 & & \vdots & & \vdots \\ & & & \nu_2 y_N^T y_1 & \dots & \nu_2 y_N^T y_N + 1 \end{array}\right] \left[\begin{array}{c} \alpha_1 \\ \vdots \\ \alpha_N \\ \hline \beta_1 \\ \vdots \\ \beta_N \end{array}\right]$$

- Resulting score variables

$$z_{x_k} = e_k = \sum_{l=1}^{N} \alpha_l x_l^T x_k$$

$$z_{y_k} = r_k = \sum_{l=1}^{N} \beta_l y_l^T y_k.$$

The eigenvalues λ will be both positive and negative for the CCA problem. Also note that one has $\rho \in [-1, 1]$.

Extension to kernel CCA

- Score variables

$$\begin{aligned} z_x &= w^T(\varphi_1(x) - \hat{\mu}_{\varphi_1}) \\ z_y &= v^T(\varphi_2(y) - \hat{\mu}_{\varphi_2}) \end{aligned}$$

where $\varphi_1(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{h_x}}$ and $\varphi_2(\cdot) : \mathbb{R}^{n_y} \to \mathbb{R}^{n_{h_y}}$ are mappings (which can be chosen to be different) to high dimensional feature spaces and $\hat{\mu}_{\varphi_1} = (1/N)\sum_{k=1}^{N} \varphi_1(x_k)$, $\hat{\mu}_{\varphi_2} = (1/N)\sum_{k=1}^{N} \varphi_2(y_k)$.

- Primal problem

$$\left[\begin{array}{lll} \boxed{\text{P}}: & \max\limits_{w,v,e,r} & \gamma \sum\limits_{k=1}^{N} e_k r_k - \nu_1 \frac{1}{2} \sum\limits_{k=1}^{N} e_k^2 - \nu_2 \frac{1}{2} \sum\limits_{k=1}^{N} r_k^2 \\ & & -\frac{1}{2} w^T w - \frac{1}{2} v^T v \\ & \text{such that} & e_k = w^T(\varphi_1(x_k) - \hat{\mu}_{\varphi_1}), \qquad k = 1, ..., N \\ & & r_k = v^T(\varphi_2(y_k) - \hat{\mu}_{\varphi_2}), \qquad k = 1, ..., N \end{array}\right]$$

with Lagrangian

$$\mathcal{L}(w, v, e, r; \alpha, \beta) = \gamma \sum_{k=1}^{N} e_k r_k - \nu_1 \frac{1}{2} \sum_{k=1}^{N} e_k^2 - \nu_2 \frac{1}{2} \sum_{k=1}^{N} r_k^2 - \frac{1}{2} w^T w -$$

$$\frac{1}{2} v^T v - \sum_{k=1}^{N} \alpha_k [e_k - w^T(\varphi_1(x_k) - \hat{\mu}_{\varphi_1})] - \sum_{k=1}^{N} \beta_k [r_k - v^T(\varphi_2(y_k) - \hat{\mu}_{\varphi_2})]$$

where α_k, β_k are Lagrange multipliers. Note that w and v might be infinite dimensional now.

- Conditions for optimality

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \;\rightarrow\; w = \sum_{k=1}^{N} \alpha_k (\varphi_1(x_k) - \hat{\mu}_{\varphi_1}) \\ \frac{\partial \mathcal{L}}{\partial v} = 0 \;\rightarrow\; v = \sum_{k=1}^{N} \beta_k (\varphi_2(y_k) - \hat{\mu}_{\varphi_2}) \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\rightarrow\; \gamma v^T (\varphi_2(y_k) - \hat{\mu}_{\varphi_2}) = \nu_1 w^T (\varphi_1(x_k) - \hat{\mu}_{\varphi_1}) + \alpha_k \quad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial r_k} = 0 \;\rightarrow\; \gamma w^T (\varphi_1(x_k) - \hat{\mu}_{\varphi_1}) = \nu_2 v^T (\varphi_2(y_k) - \hat{\mu}_{\varphi_2}) + \beta_k \quad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \;\rightarrow\; e_k = w^T (\varphi_1(x_k) - \hat{\mu}_{\varphi_1}) \quad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \beta_k} = 0 \;\rightarrow\; r_k = v^T (\varphi_2(y_k) - \hat{\mu}_{\varphi_2}) \quad k = 1, ..., N \end{cases}$$

resulting into the following dual problem after defining $\lambda = 1/\gamma$

$$\left[\boxed{\text{D}} : \text{ solve in } \alpha, \beta : \begin{bmatrix} 0 & \Omega_{c,2} \\ \Omega_{c,1} & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \lambda \begin{bmatrix} \nu_1 \Omega_{c,1} + I & 0 \\ 0 & \nu_2 \Omega_{c,2} + I \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right]$$

where

$$\begin{aligned} \Omega_{c,1_{kl}} &= (\varphi_1(x_k) - \hat{\mu}_{\varphi_1})^T (\varphi_1(x_l) - \hat{\mu}_{\varphi_1}) \\ \Omega_{c,2_{kl}} &= (\varphi_2(y_k) - \hat{\mu}_{\varphi_2})^T (\varphi_2(y_l) - \hat{\mu}_{\varphi_2}) \end{aligned}$$

are the elements of the centered Gram matrices for $k, l = 1, ..., N$. In practice these matrices can be computed by

$$\begin{aligned} \Omega_{c,1} &= M_c \Omega_1 M_c \\ \Omega_{c,2} &= M_c \Omega_2 M_c \end{aligned}$$

with centering matrix $M_c = I - (1/N) 1_v 1_v^T$.

- The resulting score variables can be computed by applying the kernel trick with kernels

$$\begin{aligned} K_1(x_k, x_l) &= \varphi_1(x_k)^T \varphi_1(x_l) \\ K_2(y_k, y_l) &= \varphi_2(y_k)^T \varphi_2(y_l) \end{aligned}$$

- Related work: Bach & Jordan (2002), Rosipal & Trejo (2001)

Recurrent Least Squares SVMs

- Standard SVM only for static estimation problem.
  Dynamic case: Recurrent LS-SVM (Suykens, 1999)
- Feedforward/NARX/series-parallel:

$$\hat{y}_k = f(y_{k-1}, y_{k-2}, ..., y_{k-p}, u_{k-1}, u_{k-2}, ..., u_{k-p})$$

  with input $u_k \in \mathbb{R}$ and output $y_k \in \mathbb{R}$.
  Training by classical SVM method.
- Recurrent/NOE/parallel:

$$\hat{y}_k = f(\hat{y}_{k-1}, \hat{y}_{k-2}, ..., \hat{y}_{k-p}, u_{k-1}, u_{k-2}, ..., u_{k-p})$$

  Classically trained by Narendra's Dynamic Backpropagation or Werbos' backpropagation through time.
- Parametrization for autonomous case:

$$\hat{y}_k = w^T \varphi(\begin{bmatrix} \hat{y}_{k-1} \\ \hat{y}_{k-2} \\ \vdots \\ \hat{y}_{k-p} \end{bmatrix}) + b$$

  which is equivalent to

$$y_k - e_k = w^T \varphi(x_{k-1|k-p} - \xi_{k-1|k-p}) + b$$

  where

$$e_k = y_k - \hat{y}_k,\ \xi_{k-1|k-p} = [e_{k-1}; e_{k-2}; ...;\ e_{k-p}]$$
$$x_{k-1|k-p} = [y_{k-1}; y_{k-2}; ...; y_{k-p}]$$

- Optimization problem:

$$\min_{w,e} \mathcal{J}(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=p+1}^{N+p} e_k^2$$

subject to

$$y_k - e_k = w^T \varphi(x_{k-1|k-p} - \xi_{k-1|k-p}) + b, \quad (k = p+1, ..., N+p).$$

- Lagrangian

$$\mathcal{L}(w,b,e;\alpha) = \mathcal{J}(w,b,e) + \sum_{k=p+1}^{N+p} \alpha_{k-p} \left[y_k - e_k - w^T \varphi(x_{k-1|k-p} - \xi_{k-1|k-p}) - b \right]$$

- Conditions for optimality

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = w - \sum_{k=p+1}^{N+p} \alpha_{k-p} \varphi(x_{k-1|k-p} - \xi_{k-1|k-p}) = 0 \\ \frac{\partial \mathcal{L}}{\partial b} = \sum_{k=p+1}^{N+p} \alpha_{k-p} = 0 \\ \frac{\partial \mathcal{L}}{\partial e_k} = \gamma e_k - \alpha_{k-p} - \sum_{i=1}^{p} \alpha_{k-p+i} \frac{\partial}{\partial e_{k-i}} [w^T \varphi(x_{k-1|k-p} - \xi_{k-1|k-p})] = 0, \\ \quad (k = p+1, ..., N) \\ \frac{\partial \mathcal{L}}{\partial \alpha_{k-p}} = y_k - e_k - w^T \varphi(x_{k-1|k-p} - \xi_{k-1|k-p}) - b = 0, \\ \quad (k = p+1, ..., N+p) \end{cases}$$

- Mercer condition

$$K(z_{k-1|k-p}, z_{l-1|l-p}) = \varphi(z_{k-1|k-p})^T \varphi(z_{l-1|l-p})$$

gives

$$\begin{cases} \sum_{k=p+1}^{N+p} \alpha_{k-p} = 0 \\ \gamma e_k - \alpha_{k-p} - \sum_{i=1}^{p} \alpha_{k-p+i} \frac{\partial}{\partial e_{k-i}} [\sum_{l=p+1}^{N+p} \alpha_{l-p} K(z_{k-1|k-p}, z_{l-1|l-p})] = 0, \\ (k = p+1, ..., N) \\ y_k - e_k - \sum_{l=p+1}^{N+p} \alpha_{l-p} K(z_{k-1|k-p}, z_{l-1|l-p}) - b = 0, \\ (k = p+1, ..., N+p) \end{cases}$$

- Solving the set of nonlinear equations in the unknowns e, α, b is computationally very expensive.

Recurrent LS-SVMs: example

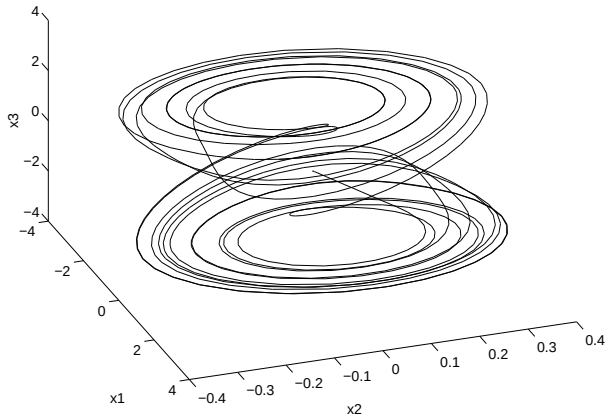
- Chua's circuit:

$$\begin{cases} \dot{x} &= a\,[y - h(x)] \\ \dot{y} &= x - y + z \\ \dot{z} &= -by \end{cases}$$

with piecewise linear characteristic

$$h(x) = m_1 x + \frac{1}{2}(m_0 - m_1)\,(|x+1| - |x-1|).$$

Double scroll attractor for
$a = 9$, $b = 14.286$, $m_0 = -1/7$, $m_1 = 2/7$.



- Recurrent LS-SVM:

$N = 300$ training data
Model: $p = 12$
RBF kernel: $\nu = 1$
SQP with early stopping

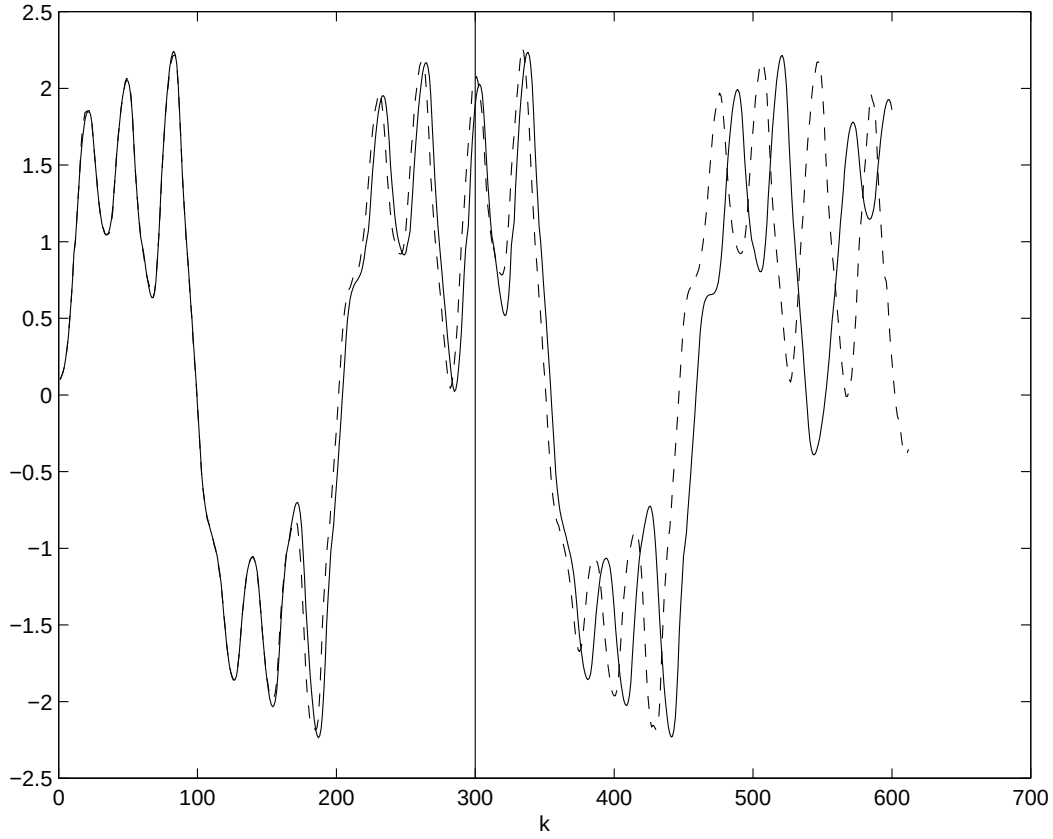


Figure: *Trajectory learning of the double scroll (full line) by a recurrent least squares SVM with RBF kernel. The simulation result after training (dashed line) on* $N = 300$ *data points is shown with as initial condition data points* $k = 1$ *to 12. For the model structure* $p = 12$ *and* $\nu = 1$ *is taken. Early stopping is done in order to avoid overfitting.*

LS-SVM for control

The N-stage optimal control problem

- Optimal control problem:

$$\min \mathcal{J}_N(x_k, u_k) = \rho(x_{N+1}) + \sum_{k=1}^{N} h(x_k, u_k)$$

subject to the system dynamics

$$x_{k+1} = f(x_k, u_k), \quad k = 1, ..., N \ (x_1 \text{ given})$$

where

$x_k \in \mathbb{R}^n$	state vector
$u_k \in \mathbb{R}$	input
$\rho(\cdot)$, $h(\cdot, \cdot)$	positive definite functions

- Lagrangian

$$\mathcal{L}_N(x_k, u_k; \lambda_k) = \mathcal{J}_N(x_k, u_k) + \sum_{k=1}^{N} \lambda_k^T [x_{k+1} - f(x_k, u_k)]$$

with Lagrange multipliers $\lambda_k \in \mathbb{R}^n$.

- Conditions for optimality:

$$\begin{cases} \frac{\partial \mathcal{L}_N}{\partial x_k} = \frac{\partial h}{\partial x_k} + \lambda_{k-1} - (\frac{\partial f}{\partial x_k})^T \lambda_k = 0, \quad k = 2, ..., N & \text{(adjoint equation)} \\ \frac{\partial \mathcal{L}_N}{\partial x_{N+1}} = \frac{\partial \rho}{\partial x_{N+1}} + \lambda_N = 0 & \text{(adjoint final condition)} \\ \frac{\partial \mathcal{L}_N}{\partial u_k} = \frac{\partial h}{\partial u_k} - \lambda_k^T \frac{\partial f}{\partial u_k} = 0, \quad k = 1, ..., N & \text{(variational condition)} \\ \frac{\partial \mathcal{L}_N}{\partial \lambda_k} = x_{k+1} - f(x_k, u_k) = 0, \quad k = 1, ..., N & \text{(system dynamics)} \end{cases}$$

Optimal control with LS-SVMs

- Consider LS-SVM from state space to action space
- Combine the optimization problem formulations of the optimal control problem and of the LS-SVM into one formulation
- Optimal control problem with LS-SVM:

$$\min \mathcal{J}(x_k, u_k, w, e_k) = \mathcal{J}_N(x_k, u_k) + \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2$$

subject to the system dynamics

$$x_{k+1} = f(x_k, u_k), \quad k = 1, ..., N \ (x_1 \text{ given})$$

and the control law

$$u_k = w^T \varphi(x_k) + e_k, \quad k = 1, ..., N$$

- Actual control signal applied to the plant: $w^T \varphi(x_k)$
- Lagrangian

$$\mathcal{L}(x_k, u_k, w, e_k; \lambda_k, \alpha_k) = \mathcal{J}_N(x_k, u_k) + \tfrac{1}{2} w^T w + \gamma \tfrac{1}{2} \textstyle\sum_{k=1}^{N} e_k^2 +$$

$$\textstyle\sum_{k=1}^{N} \lambda_k^T [x_{k+1} - f(x_k, u_k)] + \sum_{k=1}^{N} \alpha_k [u_k - w^T \varphi(x_k) - e_k]$$

- Conditions for optimality

$$
\begin{cases}
\frac{\partial \mathcal{L}}{\partial x_k} = \frac{\partial h}{\partial x_k} + \lambda_{k-1} - (\frac{\partial f}{\partial x_k})^T \lambda_k - \\
\qquad \alpha_k \frac{\partial}{\partial x_k}[w^T \varphi(x_k)] = 0, \qquad k = 2, ..., N \qquad \text{(adjoint equation)} \\
\frac{\partial \mathcal{L}}{\partial x_{N+1}} = \frac{\partial \rho}{\partial x_{N+1}} + \lambda_N = 0 \qquad \text{(adjoint final condition)} \\
\frac{\partial \mathcal{L}}{\partial u_k} = \frac{\partial h}{\partial u_k} - \lambda_k^T \frac{\partial f}{\partial u_k} + \alpha_k = 0, \quad k = 1, ..., N \qquad \text{(variational condition)} \\
\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{k=1}^{N} \alpha_k \varphi(x_k) = 0 \qquad \text{(support vectors)} \\
\frac{\partial \mathcal{L}}{\partial e_k} = \gamma e_k - \alpha_k = 0 \qquad k = 1, ..., N \qquad \text{(support values)} \\
\frac{\partial \mathcal{L}}{\partial \lambda_k} = x_{k+1} - f(x_k, u_k) = 0, \qquad k = 1, ..., N \qquad \text{(system dynamics)} \\
\frac{\partial \mathcal{L}}{\partial \alpha_k} = u_k - w^T \varphi(x_k) - e_k = 0, \quad k = 1, ..., N \qquad \text{(SVM control)}
\end{cases}
$$

- One obtains set of nonlinear equations of the form

$$F_1(x_k, x_{N+1}, u_k, w, e_k, \lambda_k, \alpha_k) = 0$$

for $k = 1, ..., N$ with x_1 given.

- Apply Mercer condition trick:

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

For RBF kernels one takes

$$K(x_k, x_l) = \exp(-\eta \left\| x_k - x_l \right\|_2^2)$$

- By replacing $w = \sum_k \alpha_k \varphi(x_k)$ in the equations, one obtains

$$\begin{cases} \frac{\partial h}{\partial x_k} + \lambda_{k-1} - (\frac{\partial f}{\partial x_k})^T \lambda_k - \alpha_k \sum_{l=1}^N \alpha_l \frac{\partial K(x_k, x_l)}{\partial x_k} = 0, & k = 2, ..., N \\ \frac{\partial \rho}{\partial x_{N+1}} + \lambda_N = 0 \\ \frac{\partial h}{\partial u_k} - \lambda_k^T \frac{\partial f}{\partial u_k} + \alpha_k = 0, & k = 1, ..., N \\ x_{k+1} - f(x_k, u_k) = 0, & k = 1, ..., N \\ u_k - \sum_{l=1}^N \alpha_l K(x_l, x_k) - \alpha_k / \gamma = 0, & k = 1, ..., N \end{cases}$$

  which is of the form

$$F_2(x_k, x_{N+1}, u_k, \lambda_k, \alpha_k) = 0$$

  for $k = 1, ..., N$ with x_1 given.

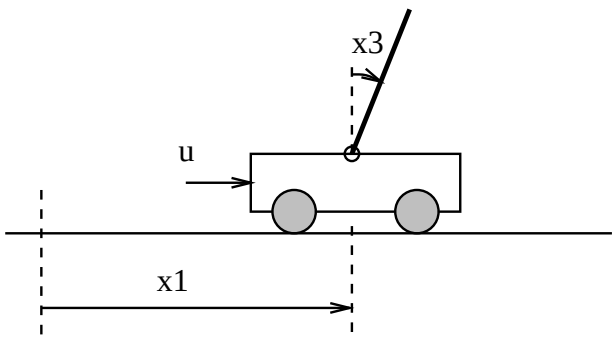
- For RBF kernels one has

$$\frac{\partial K(x_k, x_l)}{\partial x_k} = -2\eta \, (x_k - x_l) \, \exp(-\eta \, \|x_k - x_l\|_2^2)$$

  The actual control signal applied to the plant becomes

$$u_k = \sum_{l=1}^{N} \alpha_l K(x_l, x_k)$$

  where $\{x_l\}_{l=1}^N, \{\alpha_l\}_{l=1}^N$ are the solution to $F_2 = 0$.



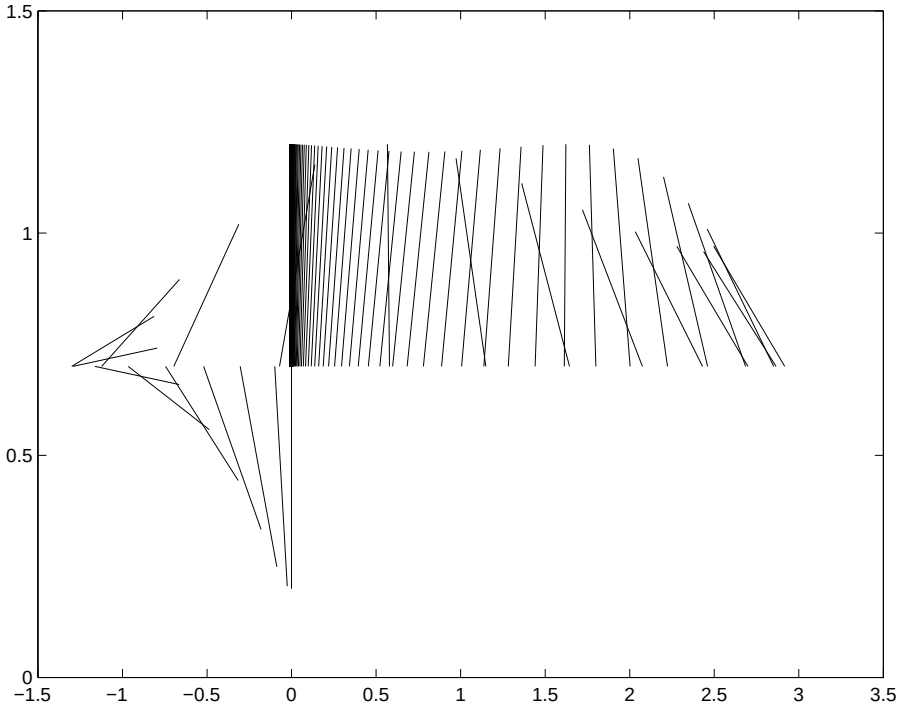


Figure: *Swinging up an inverted pendulum by a LS-SVM controller with local stabilization in its upright position. Around the target point the controller is behaving as a LQR controller. (Top) inverted pendulum system; (Bottom) simulation result which visualizes the several pole positions in time.*