Least Squares SVM Classifiers

Johan Suykens

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

Lecture 5

Contents

- Least squares SVM classifiers
- Primal-dual network interpretations
- Kernel Fisher discriminant analysis
- Multiclass problems
- UCI data sets benchmarking results
- Pruning and sparse approximation
- Application case studies

Least Squares SVM classifiers (1)

- Trying to further simplify the standard SVM formulations.
- **Least squares SVM (LS-SVM) classifiers** [Suykens & Vandewalle, 1999]: close to Vapnik's SVM formulation but solves linear system instead of QP problem.
- Classifier in primal space $y(x) = \text{sign}[w^T \varphi(x) + b]$ with given training data: $\{x_k, y_k\}_{k=1}^N$
- Optimization problem for training (**primal problem**):

$$\begin{aligned} \min_{w,b,e} \mathcal{J}(w,b,e) = \quad & \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2 \\ \text{subject to} \quad & y_k \left[w^T \varphi(x_k) + b \right] = 1 - e_k, \; k = 1, ..., N \end{aligned}$$

  Here we have **equality constraints** and 1 is a target value instead of a threshold value.

Least Squares SVM classifiers (2)

- Solve constrained optimization problem via the 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:

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

- Note that $\alpha_k = \gamma e_k$: at this point sparseness is lost.

Least Squares SVM classifiers (3)

- Define $y = [y_1; ...; y_N]$, $1_v = [1; ...; 1]$, $e = [e_1; ...; e_N]$, $\alpha = [\alpha_1; ...; \alpha_N]$.

- After elimination of w, e one obtains as **dual** problem

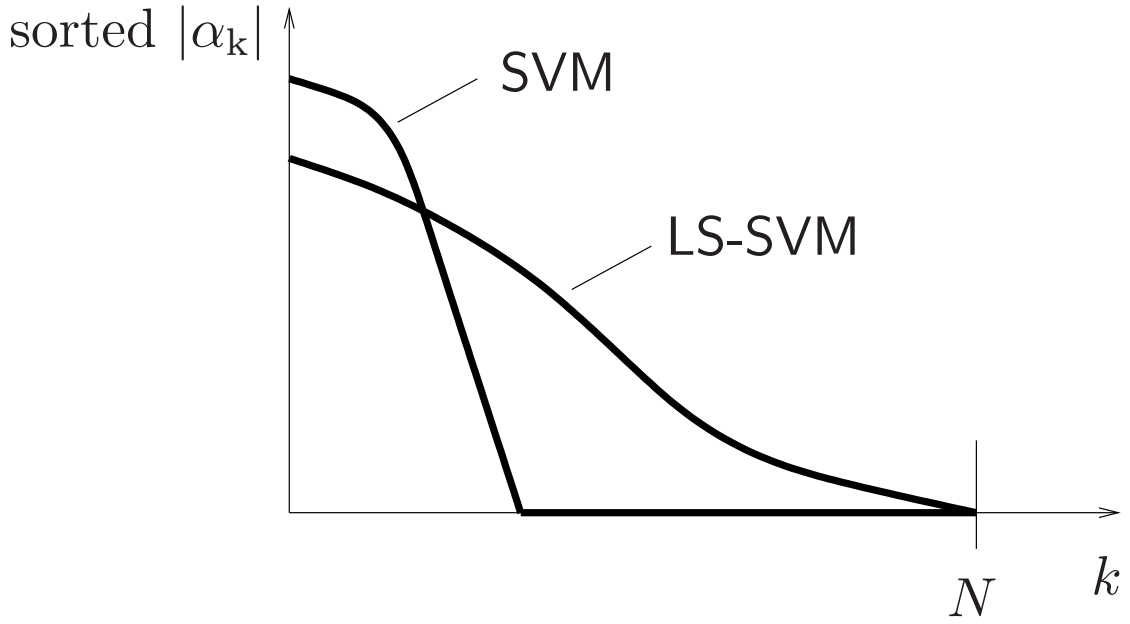
$$\left[\begin{array}{c|c} 0 & y^T \\ \hline y & \Omega + \gamma^{-1} I \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_v \end{array}\right]$$

  with application of the kernel trick

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

- Linear system (**KKT system**) obtained as dual problem because of quadratic cost function with linear equality constraints.

LS-SVM: loss of sparseness



Qualitative comparison of the sorted $|\alpha|$ solution vectors between SVMs (including the standard Vapnik SVM) and LS-SVMs. In the LS-SVM case each data point is contributing to the model and sparseness is lost.

LS-SVM classifier properties (1)

- *Choice of kernel function*: The chosen kernel function should be **positive definite** and satisfy the Mercer condition.

- *Global and unique solution*: The dual problem for linear and nonlinear LS-SVMs corresponds to solving a linear KKT system which is a square system with a **unique solution** when the matrix has full rank.

- *KKT system as a core problem:* Solving linear Karush-Kuhn-Tucker (KKT) systems is a **fundamental issue** in constrained nonlinear optimization problems in general.

- *Lack of sparseness and interpretation of support vectors*: A drawback of the simplified formulation is the lack of sparseness.

LS-SVM classifier properties (2)

- *Non-parametric/parametric issues*: LS-SVM classifiers have the same primal-dual neural network interpretations as shown for SVMs.

  In the **primal** weight space the problem is **parametric** with fixed size vector $w \in \mathbb{R}^{n_h}$ where n_h is the number of hidden units in the network interpretation for this space.

  In the **dual** space the problem is **non-parametric** as the size of the solution vector $\alpha \in \mathbb{R}^N$ grows with the number of training data N.

- *Tuning parameters*: If one takes for example an RBF kernel

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

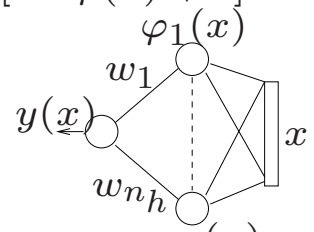
  then only α, b result from solving the linear KKT system. The tuning parameters (γ, σ) can be determined e.g. by cross-validation.

LS-SVMs: primal-dual interpretations

Primal problem P

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

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



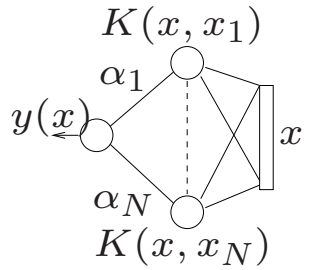
Kernel trick

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

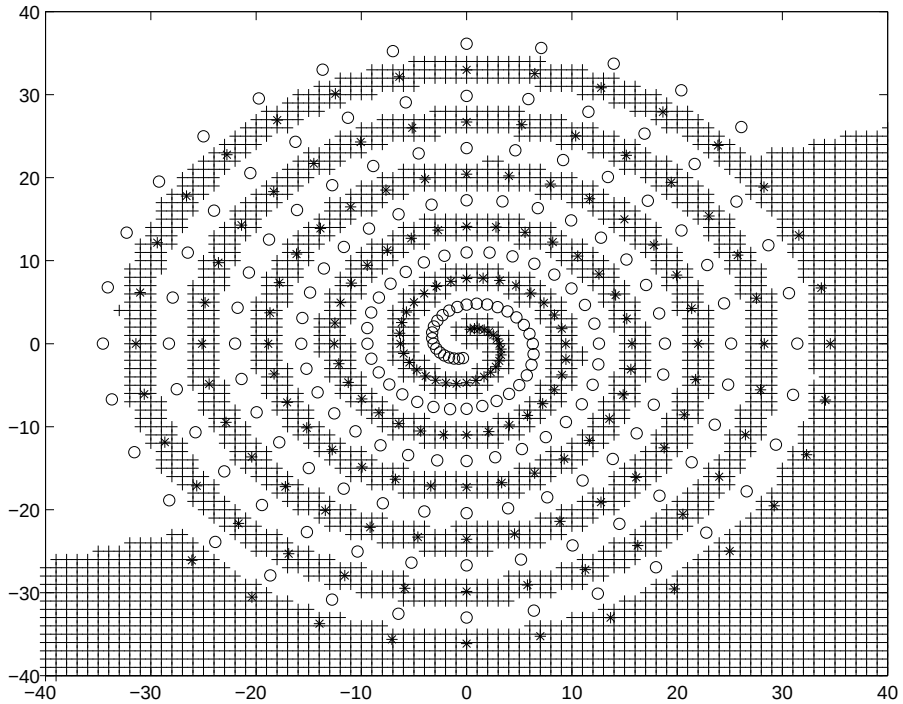
Dual problem D

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

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



Toy example: two-spiral problem



two-spiral classification problem with the two classes indicated by 'o' and '*' and 180 training data for each class. The figure shows the excellent generalization performance for an LS-SVM machine with RBF kernel.

LS-SVM classifier: Ripley data (1)

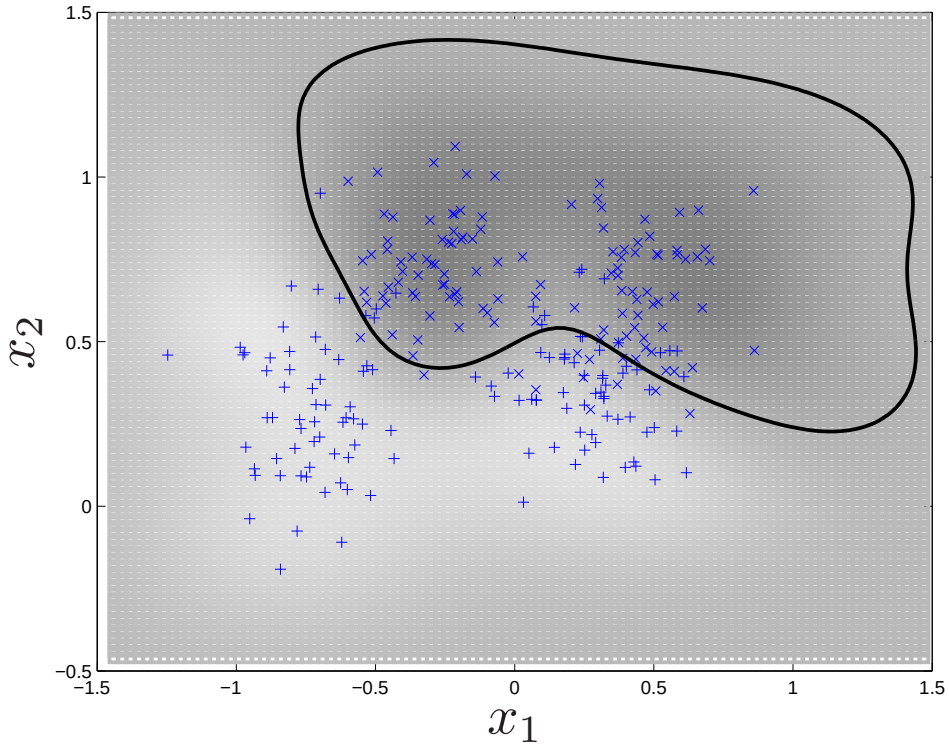


Illustration of the LS-SVM classifier on the Ripley binary classification data sets: decision boundary for LS-SVM with a well-tuned RBF kernel.

LS-SVM classifier: Ripley data (2)

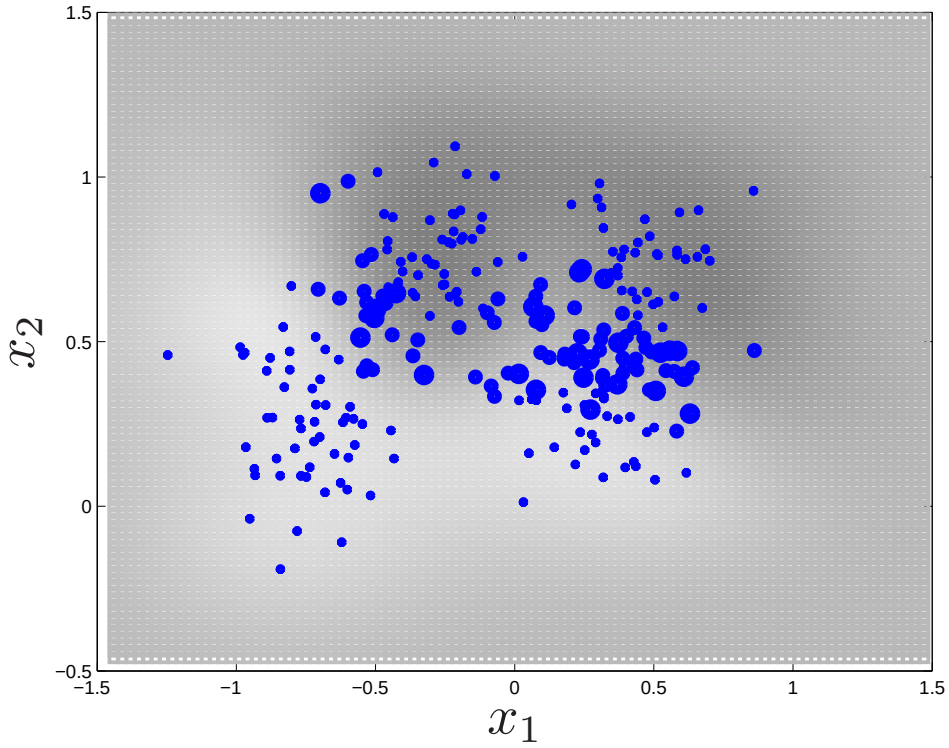
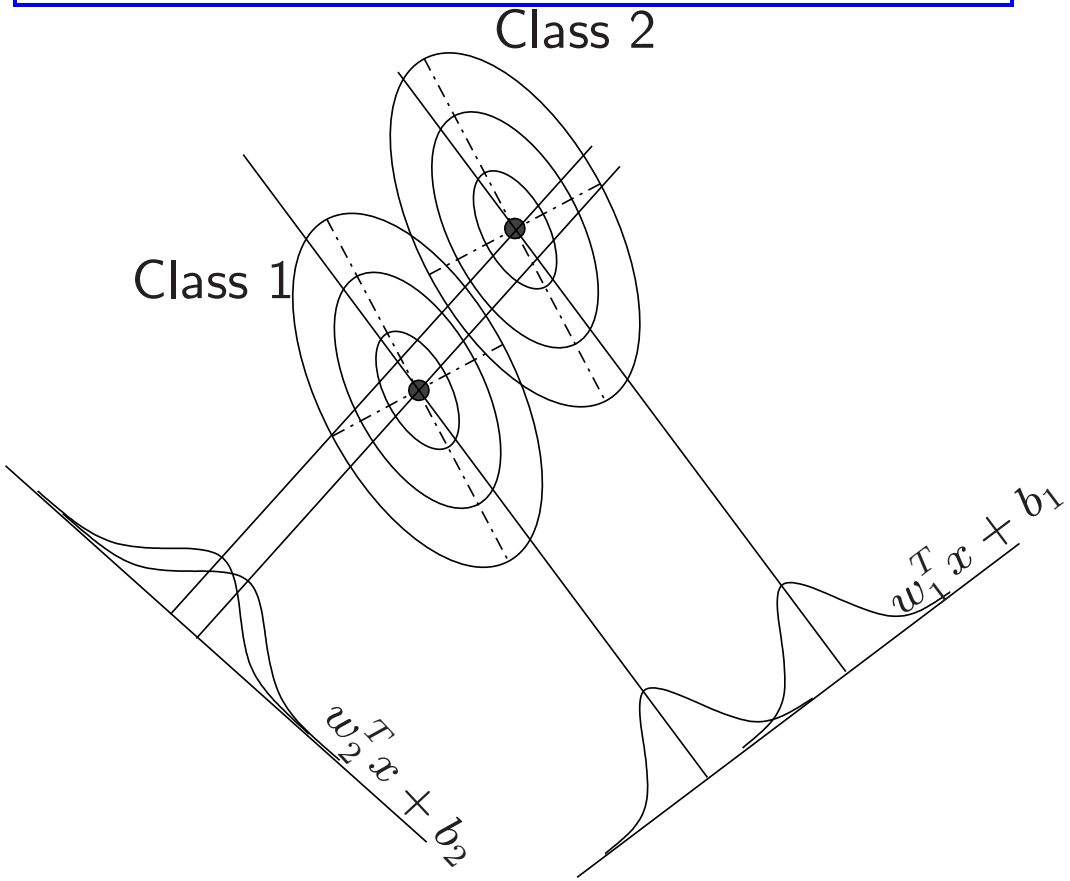


Illustration of the support values α_k with size of the black dots chosen proportional to the α_k values.

Fisher discriminant analysis (1)



In Fisher discriminant analysis (FDA) one aims at maximizing the between-class scatter and minimizing the within-class scatter. This figure shows that projection on the line $z = w_1^T x + b_1$ gives a much better discriminatory power than on $z = w_2^T x + b_2$.

Fisher discriminant analysis (2)

- **Project data** $x_k \in \mathbb{R}^n$ from the original input space to a one-dimensional variable $z_k \in \mathbb{R}$ (i.e. projecting multivariate data to univariate data):

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

with $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$.

- One maximizes the **between-class** variances and minimizes the **within-class** variances for the two classes via the **Rayleigh quotient**:

$$\begin{aligned} \max_{w,b} J_{\mathrm{FD}}(w,b) &= \frac{[\mathcal{E}[z^{(1)}] - \mathcal{E}[z^{(2)}]]^2}{\mathcal{E}\{[z^{(1)} - \mathcal{E}[z^{(1)}]]^2\} + \mathcal{E}\{[z^{(2)} - \mathcal{E}[z^{(2)}]]^2\}} \\ &= \frac{w^T \Sigma_{\mathcal{B}} w}{w^T \Sigma_{\mathcal{W}} w} \end{aligned}$$

with $\Sigma_{\mathcal{B}} = [\mu^{(1)} - \mu^{(2)}][\mu^{(1)} - \mu^{(2)}]^T$, $\Sigma_{\mathcal{W}} = \mathcal{E}\{[x - \mu^{(1)}][x - \mu^{(1)}]^T\} + \mathcal{E}\{[x - \mu^{(2)}][x - \mu^{(2)}]^T\}$ and $z_k^{(1)} = w^T x_k^{(1)} + b$, $z_k^{(2)} = w^T x_k^{(2)} + b$.

Fisher discriminant analysis (3)

- By taking $\partial J_{\mathrm{FD}}(w)/\partial w = 0$ we obtain the generalized eigenvalue problem

$$\Sigma_{\mathcal{W}} w = \Sigma_{\mathcal{B}} w \, (w^T \Sigma_{\mathcal{W}} w / w^T \Sigma_{\mathcal{B}} w).$$

  By using the expression for $\Sigma_{\mathcal{B}}$ one obtains then the following w_{FD} direction for the optimal

$$w_{\mathrm{FD}} \propto \Sigma_{\mathcal{W}}^{-1} [\mu^{(1)} - \mu^{(2)}].$$

- The projections of the means $\mu^{(1)}, \mu^{(2)}$ to the one-dimensional space give

$$\begin{array}{rcccl} f(\mu^{(1)}) & = & w_{\mathrm{FD}}^T \mu^{(1)} + b & \propto & [\mu^{(1)} - \mu^{(2)}]^T \Sigma_{\mathcal{W}}^{-1} \mu^{(1)} \\ f(\mu^{(2)}) & = & w_{\mathrm{FD}}^T \mu^{(2)} + b & \propto & [\mu^{(1)} - \mu^{(2)}]^T \Sigma_{\mathcal{W}}^{-1} \mu^{(2)}. \end{array}$$

Fisher discriminant analysis (4)

- In practice one works with the sample means

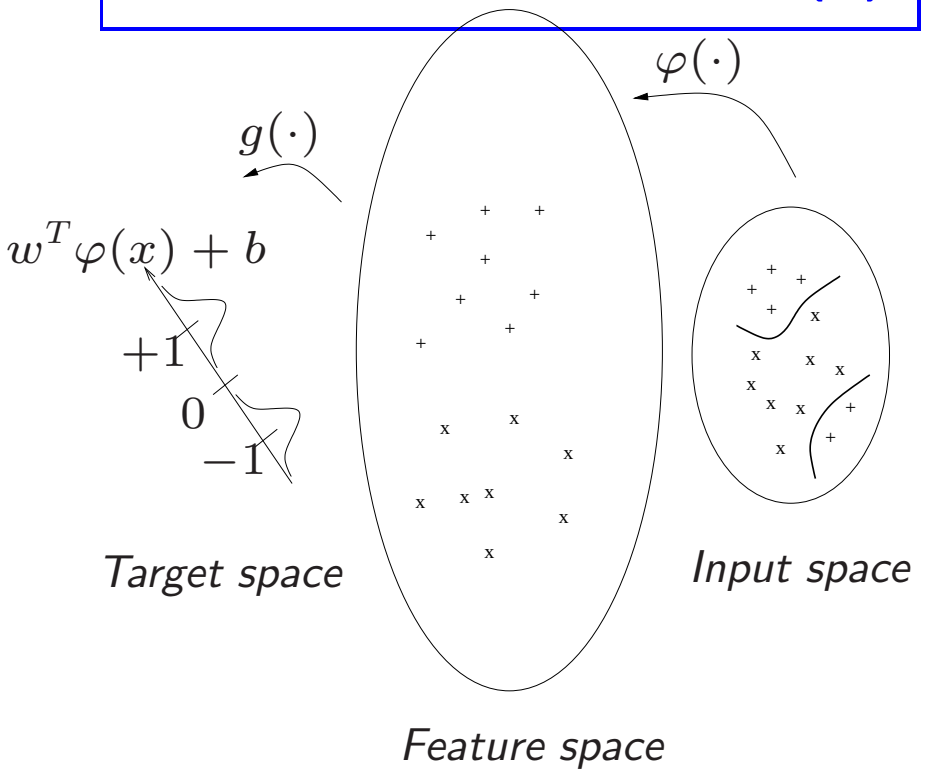
$$\hat{\mu}^{(1)} = \frac{1}{N_1}\sum_{k=1}^{N_1} x_k^{(1)}\,,\;\; \hat{\mu}^{(2)} = \frac{1}{N_2}\sum_{k=1}^{N_2} x_k^{(2)}$$

  for $\mu^{(1)}, \mu^{(2)}$ of class 1 and 2, respectively, and the sample covariance matrices

$$\begin{aligned} S_{\mathcal{W}_1} &= \frac{1}{N_1 - 1}\sum_{i=1}^{N_1}[x_k^{(1)} - \hat{\mu}^{(1)}][x_k^{(1)} - \hat{\mu}^{(1)}]^T \\ S_{\mathcal{W}_2} &= \frac{1}{N_2 - 1}\sum_{i=1}^{N_2}[x_k^{(2)} - \hat{\mu}^{(2)}][x_k^{(2)} - \hat{\mu}^{(2)}]^T \end{aligned}$$

  as estimates for the covariance matrices. N_1, N_2 denote the number of data points for class 1 and 2.

FDA in feature space (1)



Fisher discriminant analysis in the feature space is closely related to LS-SVM classification. In LS-SVM classification the constraints $y_k[w^T\varphi(x_k)+b] = 1 - e_k$ can be interpreted as having target values $+1$ and -1 on a line for which one aims at minimizing the within-class scattering, which is characterized by the term $\sum_k e_k^2$ in the objective function of the LS-SVM classifier.

FDA in feature space (2)

- The data are projected as follows

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

with $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ the mapping from the input space to the one-dimensional target space, $g(\cdot) : \mathbb{R}^{n_h} \rightarrow \mathbb{R}$ the mapping from the high dimensional feature space to the target space, $\varphi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{n_h}$ the mapping from the input space to a high dimensional **feature space**.

- **Rayleigh quotient**

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

where $\mu^{(1)} = \mathcal{E}[\varphi(x^{(1)})]$, $\mu^{(2)} = \mathcal{E}[\varphi(x^{(2)})]$ and

$$\begin{aligned} \Sigma_{\mathcal{B}} &= [\mu^{(1)} - \mu^{(2)}][\mu^{(1)} - \mu^{(2)}]^T \\ \Sigma_{\mathcal{W}} &= \mathcal{E}\{[\varphi(x) - \mu^{(1)}][\varphi(x) - \mu^{(1)}]^T\} + \mathcal{E}\{[\varphi(x) - \mu^{(2)}][\varphi(x) - \mu^{(2)}]^T\} \end{aligned}$$

FDA in feature space (3)

- Connection to the **LS-SVM classifier**:

$$\begin{aligned} \min_{w,b,e_k,e_l} \quad & \frac{1}{2}w^T w + \gamma \frac{1}{2}\left(\sum_{k=1}^{N_1} {e_k^{(1)}}^2 + \sum_{l=1}^{N_2} {e_l^{(2)}}^2\right) \\ \text{such that} \quad & y_k^{(1)}\left[w^T\varphi(x_k^{(1)}) + b\right] = t_1 - e_k^{(1)}, \;\; k = 1, .., N_1 \\ & y_l^{(2)}\left[w^T\varphi(x_l^{(2)}) + b\right] = t_2 - e_l^{(2)}, \;\; l = 1, ..., N_2 \end{aligned}$$

  where t_1, t_2 are positive target values for class 1 and 2, N_1, N_2 are the number of data points of class 1 and 2, respectively. The indices k, l run over the elements of class 1 and 2.

- Goal of the above LS-SVM classifiers:

  1. **Maximizing the soft margin** by minimizing $\|w\|_2$
  2. **Minimizing the within-class scatter** for fixed targets with a similar objective as FDA in a high dimensional feature space.

Multiclass problem

CAT ?

FISH ?

DOG ?

Multiclass LS-SVMs (1)

- There exist several ways for solving multiclass problems
- One can **decompose** the problem into a set of **binary** classification problems
- There exist **different coding/decoding schemes**, e.g.
  m outputs to encode 2^m classes;
  m outputs to encode m classes
- Multiclass LS-SVM by defining **additional outputs**
  (similar to MLP network approaches to multiclass problems)

Multiclass LS-SVMs (2)

- Classifier in **primal** space: classifier system with n_h outputs

$$\begin{cases} y^{(1)}(x) &= \text{sign}[{w^{(1)}}^T \varphi^{(1)}(x) + b^{(1)}] \\ y^{(2)}(x) &= \text{sign}[{w^{(2)}}^T \varphi^{(2)}(x) + b^{(2)}] \\ & \vdots \\ y^{(n_y)}(x) &= \text{sign}[{w^{(n_y)}}^T \varphi^{(n_y)}(x) + b^{(n_y)}] \end{cases}$$

- Optimization problem in **primal** space:

$$\min_{w^{(i)},b^{(i)},e_k^{(i)}} J_{\text{P}}(w^{(i)}, e_k^{(i)}) = \frac{1}{2}\sum_{i=1}^{n_y} {w^{(i)}}^T w^{(i)} + \frac{1}{2}\sum_{i=1}^{n_y} \gamma_i \sum_{k=1}^{N} \left(e_k^{(i)}\right)^2$$

$$\text{subject to} \quad \begin{array}{l} y_k^{(1)} \left[{w^{(1)}}^T \varphi^{(1)}(x_k) + b^{(1)}\right] = 1 - e_k^{(1)},\ k = 1, ..., N \\ y_k^{(2)} \left[{w^{(2)}}^T \varphi^{(2)}(x_k) + b^{(2)}\right] = 1 - e_k^{(2)},\ k = 1, ..., N \\ \vdots \\ y_k^{(n_y)} \left[{w^{(n_y)}}^T \varphi^{(n_y)}(x_k) + b^{(n_y)}\right] = 1 - e_k^{(n_y)},\ k = 1, ..., N. \end{array}$$

Multiclass LS-SVMs (3)

- Lagrangian

$$\mathcal{L}(w^{(i)}, b^{(i)}, e_k^{(i)}; \alpha_k^{(i)}) = J_{\mathrm{P}}(w^{(i)}, e_k^{(i)}) - \sum_{i=1}^{n_y} \sum_{k=1}^{N} \alpha_k^{(i)} \left(y_k^{(i)} [{w^{(i)}}^T \varphi^{(i)}(x_k) + b^{(i)}] - 1 + e_k^{(i)} \right)$$

- Conditions for optimality:

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

for $k = 1, ..., N$ and $i = 1, ..., n_y$.

Multiclass LS-SVMs (4)

- After elimination of the variables $w^{(i)}$ and $e_k^{(i)}$ one obtains as the **dual problem** the KKT system

$$\text{solve in } \alpha_k^{(i)}, b^{(i)}: \quad \left[\begin{array}{c|c} 0 & Y_M^T \\ \hline Y_M & \Omega_M + D_M \end{array}\right] \left[\begin{array}{c} b_M \\ \hline \alpha_M \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_v \end{array}\right]$$

where $b_M = [b^{(1)}; ...; b^{(n_y)}]$, $\alpha_M = [\alpha_1^{(1)}; ...; \alpha_N^{(1)}; ....; \alpha_1^{(n_y)}; ...; \alpha_N^{(n_y)}]$,

$$Y_M = \text{blockdiag}\{\left[\begin{array}{c} y_1^{(1)} \\ \vdots \\ y_N^{(1)} \end{array}\right], ..., \left[\begin{array}{c} y_1^{(n_y)} \\ \vdots \\ y_N^{(n_y)} \end{array}\right]\}$$

$$\Omega_M = \text{blockdiag}\{\Omega^{(1)}, ..., \Omega^{(n_y)}\}, \quad \Omega_{kl}^{(i)} = y_k^{(i)} y_l^{(i)} K^{(i)}(x_k, x_l)$$

$$D_M = \text{blockdiag}\{D^{(1)}, ..., D^{(n_y)}\}, \quad D_{kl}^{(i)} = \delta_{kl}/\gamma_i$$

for $k, l = 1, ..., N$, $i = 1, ..., n_y$; δ_{kl} denotes the Kronecker delta ($\delta_{kl} = 1$ if $k = l$ and 0 otherwise).

Multiclass LS-SVMs (5)

- The kernel trick is applied as follows, in the case of RBF kernels

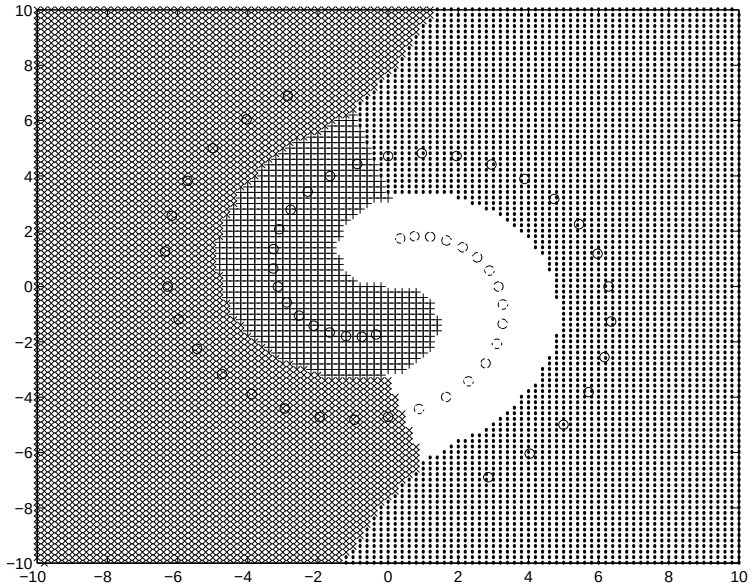
$$\begin{aligned} K^{(i)}(x_k, x_l) &= \varphi^{(i)}(x_k)^T \varphi^{(i)}(x_l) \\ &= \exp\left(-\|x_k - x_l\|_2^2/\sigma_i^2\right), \ i = 1, ..., n_y \end{aligned}$$

- Resulting classifier in the **dual space**

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

for $i = 1, ..., n_y$.

Multiclass LS-SVM: toy example



Illustrative example of a LS-SVM with RBF kernel on a four class classification problem. The classifier is obtained by solving a set of linear equations.

Benchmarking results (1)

UCI benchmark repository `http://kdd.ics.uci.edu/`

Binary class problems:
Statlog Australian credit (`acr`), Bupa liver disorders (`bld`), Statlog German credit (`gcr`), Statlog heart disease (`hea`), Johns Hopkins university ionosphere (`ion`), Pima Indians diabetes (`pid`), sonar (`snr`), tic-tac-toe endgame (`ttt`), Wisconsin breast cancer (`wbc`), adult dataset (`adu`)

Multiclass problems:
balance scale (`bal`), contraceptive method choice (`cmc`), image segmentation (`ims`), iris (`iri`), LED display (`led`), thyroid disease (`thy`), US postal service (`usp`), Statlog vehicle silhouette (`veh`), waveform (`wav`), wine recognition dataset (`win`)

Benchmarking results (2)

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

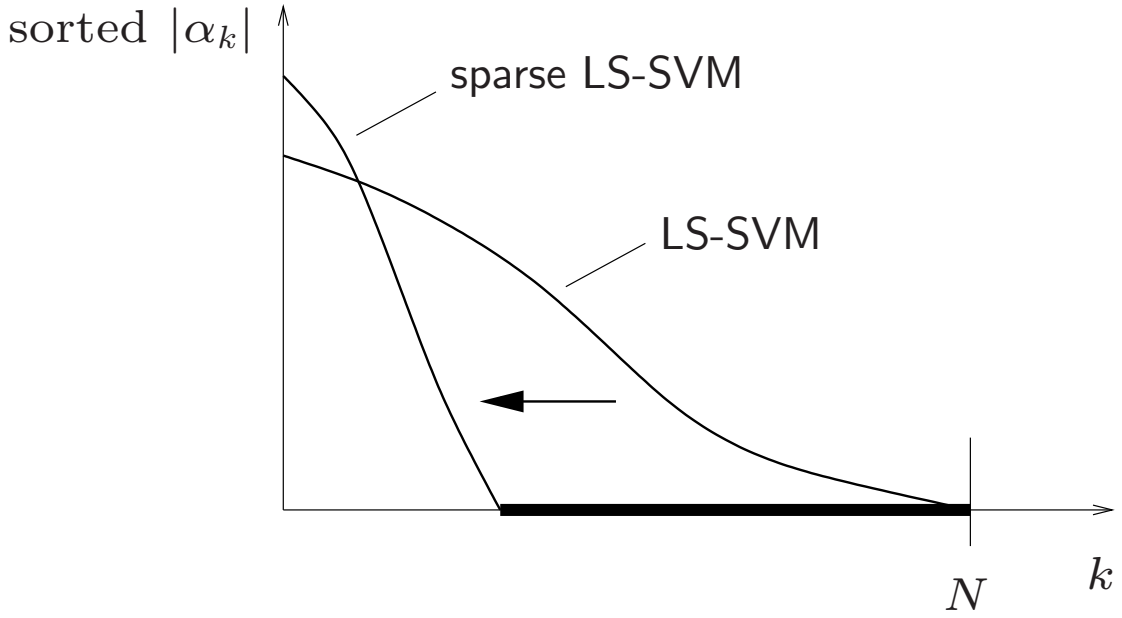
	acr	bld	gcr	hea	ion	pid	snr	ttt	wbc	adu	AA	AR	P_{ST}
N_{test}	230	115	334	90	117	256	70	320	228	12222			
n	14	6	20	13	33	8	60	9	9	14			
RBF LS-SVM	**87.0**(2.1)	**70.2**(4.1)	**76.3**(1.4)	**84.7**(4.8)	**96.0**(2.1)	**76.8**(1.7)	*73.1*(4.2)	*99.0*(0.3)	*96.4*(1.0)	*84.7*(0.3)	**84.4**	**3.5**	**0.727**
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*

Table 1: *Comparison of the 10 times randomized* **test set** *performance of LS-SVM and LS-SVM_F (linear, polynomial and Radial Basis Function kernel) with the performance of LDA, QDA, Logit, C4.5, oneR, IB1, IB10, NB_k, NB_n and the Majority Rule classifier on 10 binary domains. The Average Accuracy (AA), Average Rank (AR) and Probability of equal medians using the Sign Test (P_{ST}) taken over all domains are reported in the last three columns. Best performances are underlined and denoted in bold face, performances not significantly different at the 5% level are denoted in bold face, performances significantly different at the 1% level are emphasized. LS-SVMs with RBF kernels are performing very well in this comparative study.*

	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*

Table 1: *Comparison of the 10 times randomized* **test set** *performance of LS-SVM and LS-SVM_F (linear, polynomial and Radial Basis Function kernel) with the performance of LDA, QDA, Logit, C4.5, oneR, IB1, IB10, NB_k, NB_n and the Majority Rule classifier on 10 binary domains. The Average Accuracy (AA), Average Rank (AR) and Probability of equal medians using the Sign Test (P_{ST}) taken over all domains are reported in the last three columns. Best performances are underlined and denoted in bold face, performances not significantly different at the 5% level are denoted in bold face, performances significantly different at the 1% level are emphasized. Good results are obtained by LS-SVM 1vs1 with RBF kernel.*

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.)

Pruning

MLP: Computation of inverse Hessian is needed

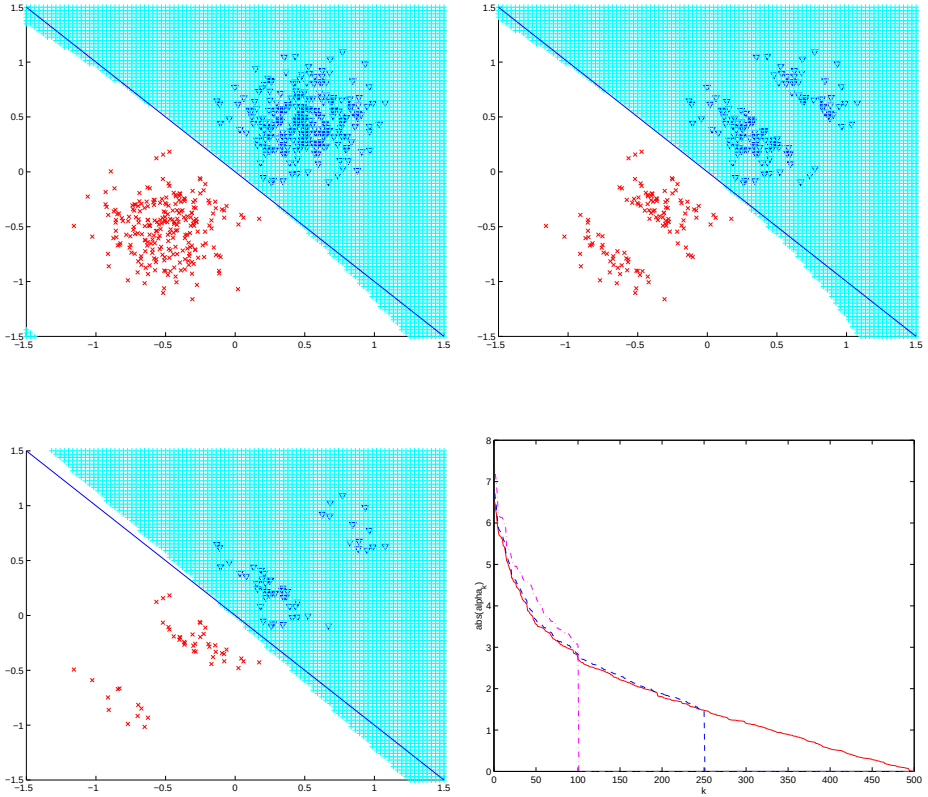
LS-SVM:

Support values α_k are available.
Pruning is immediately done based upon the solution vector α itself.

LS-SVM pruning

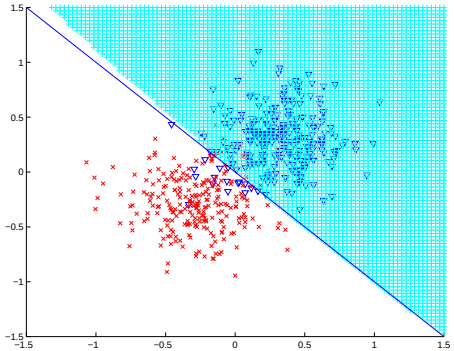
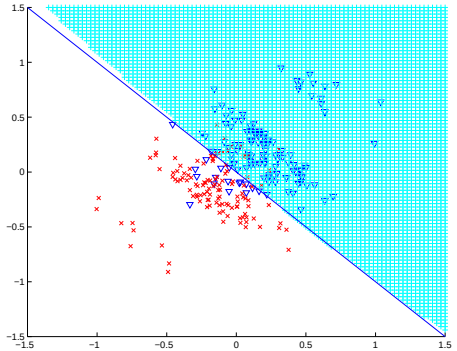
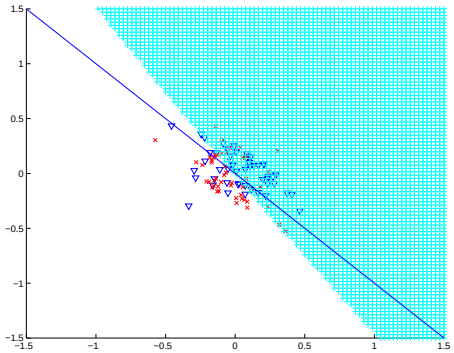
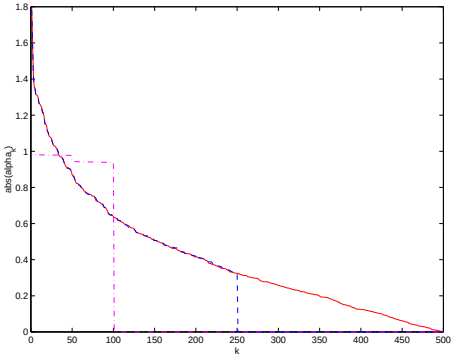
1. Compute LS-SVM for N training data
2. Reduce small amount of the training set (e.g. 5 %) based upon the sorted support value spectrum
3. Re-estimate the LS-SVM on the reduced training set
4. Go to 2, unless the user-defined performance index degrades

Pruning - toy example (1)



Classification problem with data generated from Gaussian distributions with the same covariance matrix (hence a straight line should be optimal). LS-SVMs with RBF kernel are trained and pruned in order to obtain a sparse approximation (500 SV $\rightarrow$ 250 SV $\rightarrow$ 100 SV).

Pruning - toy example (2)

(500 SV $\rightarrow$ 250 SV $\rightarrow$ 100 SV)

Selection of tuning parameters

For the selection of (γ, σ) in the case of an RBF kernel, one has several possibilities:

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

- *Cross-validation*
  (γ, σ) are optimized on the sum of the parts of the training set that were left out in the several runs. The advantage is that no additional validation set is needed and one can check the generalization performance on an independent test set.

- *Bayesian inference*

- *Generalization bounds from VC theory*

Examples LS-SVM applications (classification and regression)

- 20 UCI benchmark data sets (binary and multiclass problems)
- Ovarian cancer classification
- Classification of brain tumours from magnetic resonance spectroscopy
- 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
- Other

LS-SVM case study: Classification of brain tumours

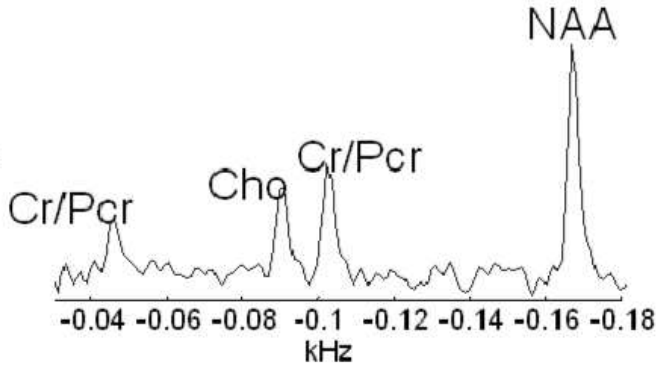
- Brain tumors are the 2nd leading cause of cancer death in children under age 15 and young adults up to age 34, and also the 2nd fastest growing cause of cancer death among humans over age 65.
  - Types : benign (e.g. meningiomas, low grade astrocytomas) and malignant (e.g. glioblastomas, metastases)
  - Each type of brain tumor needs a different treatment.

- Surgery and radiation therapy are effective but can damage the surrounding normal brain tissue. **Early detection** and **correct treatment** will minimize brain damage significantly.

- Medical techniques for **preoperative evaluation**
  - invasive : biopsy, make a small hole and take a sample from the brain
  - noninvasive : Magnetic Resonance Imaging (MRI), MR Spectroscopy (MRS)

Magnetic Resonance Spectroscopy

MR scanner



Feature Vector

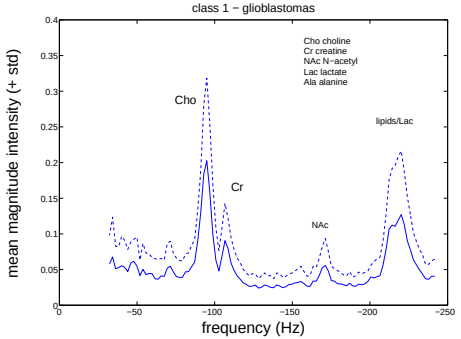
MRS based tumor classification

	1	2	3	total
CDP (STEAM, TE=135 ms)	15	5	4	24
IDI (PRESS, TE=136 ms)	23	24	14	61
SGHMS (PRESS, TE=136 ms)	12	8	8	28
total	50	37	26	113

Number of proton MRS data of glioblastomas (class 1), meningiomas (2) and metastases (3). The rows correspond to the acquisition centre, while the columns mention the type of brain tumor.

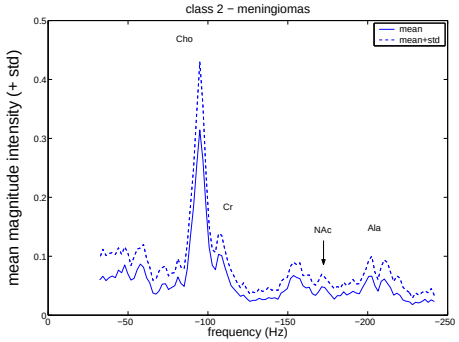
Acknowledgement : INTERPRET IST-1999-10310

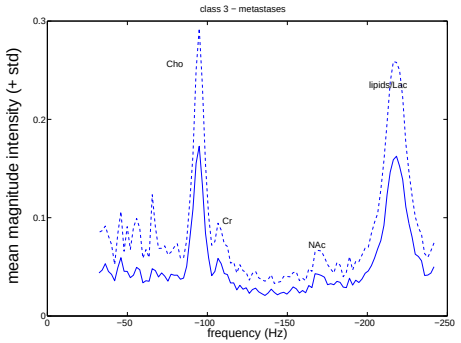
Data characteristics



← Class 1

Class 2 →





← Class 3

LS-SVM Classifiers (1)

Leave-one-out cross-validation to select the optimal hyperparameters of LS-SVM classifiers:

- LS-SVM classifier with RBF kernel for class 1 vs class 2:
  $\sigma = 0.7579, \gamma = 3.4822.$
- LS-SVM classifier with RBF kernel for class 1 vs class 3:
  $\sigma = 0.4353, \gamma = 1.3195.$
- LS-SVM classifier with RBF kernel for class 2 vs class 3:
  $\sigma = 0.6598, \gamma = 5.2780.$
- LS-SVM classifier with linear kernel

LS-SVM Classifiers (2)

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

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

References: International network for Pattern Recognition of Tumours Using Magnetic Resonance http://carbon.uab.es/INTERPRET/, Lukas et al., ESANN 2002, Bruges, Belgium, 2002, 131-136.

LS-SVM case study: Bankruptcy Prediction (1)

- **Introduction:** Bankruptcy prediction is important problem for the banking industry, which is to predict if a firm will suffer financial failure. If a firm fails, it causes substantial losses to the financial community and the society as a whole. From this point of view, good forecasts on the failure risks is a warning to managers, investors, who can take subsequent measures to reduce and avoid the loss.

- **Reference:** Van Gestel, T., Baesens, B., Suykens, J., Espinoza, M., Baestaens, D., Vanthienen, J., De Moor, B. "Bankruptcy Prediction with Least Squares Support Vector Machine Classifiers," (2003) International Conference in Computational Intelligence and Financial Engineering.

- **Problem and Data:** Data are financial indicators from middle-market capitalization firms in the Benelux. From a total of 422 firms, where 74 went bankrupt and 348 were solvent companies. The variables to be used in the model as explanatory inputs are 40 financial indicators, as liquidity, profitability and solvency measurements.

LS-SVM case study: Bankruptcy Prediction (2)

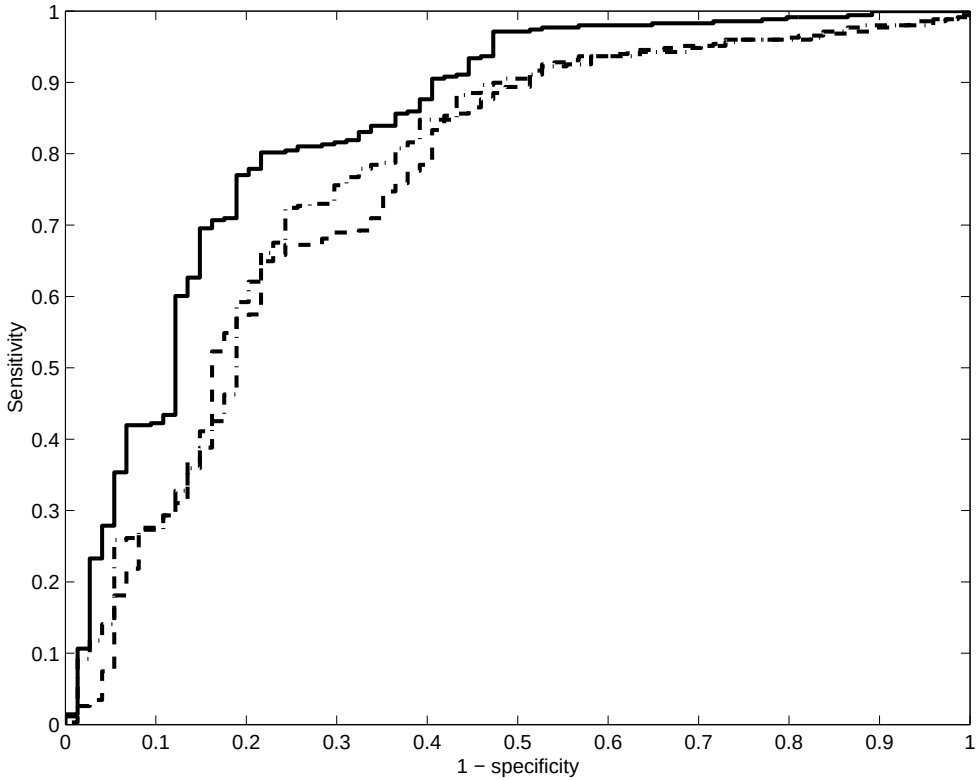
- **Goal:** Binary classification of firms (solvent or bankrupt).

- **Methodology and Implementation:** LS-SVM model is trained using leave-one-out cross-validation for selecting the hyperparameters. Additionally, input selection is performed, in order to characterize the inputs with more explanatory power among the original 40 variables included in the model.

  Results of the LS-SVM are compared with traditional techniqes such as Logistic Regression (Logit) and Linear Discriminant Analysis (LDA).

- **Results:** The LS-SVM model is able to obtain a larger percentage of correct classifications than the other methods, either using the full input set or using the refined input set, obtained after removing all the non-relevant inputs in a stepwise procedure.

LS-SVM case study: Bankruptcy Prediction (3)



Receiver Operating Characteristic curves obtained with LDA (dashed line), LOGIT (dash-dotted line) and LS-SVM (full line) using the full candidate input set.

LS-SVM case study: Bankruptcy Prediction (4)

Comparison LDA, LOGIT and LS-SVM:

	LDA	LOGIT	LS-SVM
PCC (F)	84.83 (0.0051)	84.12 (0.0027)	88.39
PCC (R)	86.97 (0.0147)	87.91 (0.0485)	91.00

Leave-one-out cross validation Percentage of Correct Classifications (PCC) LDA, LOGIT and LS-SVM using the full (F) or the reduced (R) set of inputs.