More on LS-SVM, GP and RKHS

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 6

Contents

- LS-SVM for function estimation
- Solving the linear system
- Reproducing kernel Hilbert spaces (RKHS), representer theorem
- Gaussian processes (GP)
- Automatic relevance determination
- Other spaces
- Choice of loss function, robustness
- Weighted LS-SVM
- Sparseness by pruning

LS-SVMs for function estimation (1)

- LS-SVM model in **primal** space:

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

with $x \in \mathbb{R}^n, y \in \mathbb{R}$. Given training set $\{x_k, y_k\}_{k=1}^N$.

- Optimization problem (primal)

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

subject to equality constraints

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

For $b = 0$ it relates to ridge regression in the feature space.

LS-SVMs for function estimation (2)

- 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

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

LS-SVMs for function estimation (3)

- Solution

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

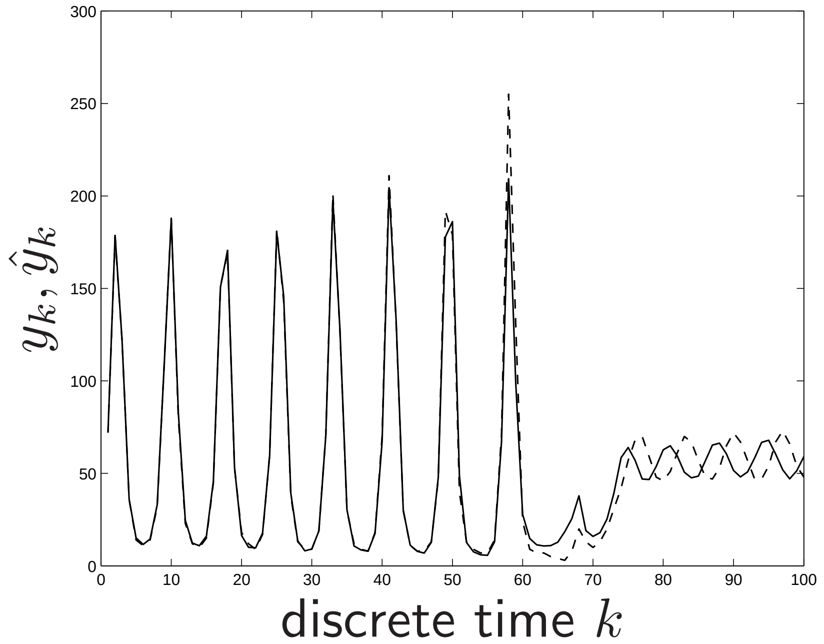
with $y = [y_1; ...; y_N]$, $1_v = [1; ...; 1]$, $\alpha = [\alpha_1; ...; \alpha_N]$ and by applying the **kernel trick**

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

- Resulting LS-SVM model in **dual** space

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

LS-SVM: Santa Fe laser data



Time-series prediction by an LS-SVM with RBF kernel for the Santa Fe chaotic laser data. The figure shows the iterative prediction of an NARX LS-SVM model over a time horizon of 100 points, that was trained on the previous 1000 given data points; true data y_k (solid line), iterative prediction $\hat{y}_k$ (dashed line).

Computing LS-SVM solutions (1)

- Problem to be solved is of the form

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

- Takes only one line command in Matlab:

```
>> x = A\B
```

  Problem: matrix $\mathcal{A}$ needs to be stored. Hence only applicable to smaller problems, depending on the computer memory

- For larger data sets (e.g. range 3000-10000 training data) **iterative** methods can be used to solve the linear system

- Examples of iterative methods: conjugate gradient (CG), successive over-relaxation (SOR), generalized minimal residual (GMRES)

Computing LS-SVM solutions (2)

- **Conjugate gradient** (CG) method: iterative method applicable to

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

  with $\mathcal{A}$ symmetric and positive definite. The KKT system of size $(N+1) \times (N+1)$ is not positive definite. It should be transformed before CG can be applied to it.

- Represent the original problem, which is 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 $s = Y^T H^{-1} Y > 0$ $(H = H^T > 0)$.

Computing LS-SVM solutions (3)

- **Hestenes-Stiefel conjugate gradient algorithm** [Golub & Van Loan]

$$
\begin{array}{l}
i = 0; x_0 = 0; r_0 = \mathcal{B}; \\
\text{while } r_i \neq 0 \\
\quad i = i + 1 \\
\quad \text{if } i = 1 \\
\quad\quad p_1 = r_0 \\
\quad \text{else} \\
\quad\quad \beta_i = r_{i-1}^T r_{i-1} / r_{i-2}^T r_{i-2} \\
\quad\quad p_i = r_{i-1} + \beta_i p_{i-1} \\
\quad \text{end} \\
\quad \lambda_i = r_{i-1}^T r_{i-1} / p_i^T \mathcal{A} p_i \\
\quad x_i = x_{i-1} + \lambda_i p_i \\
\quad r_i = r_{i-1} - \lambda_i \mathcal{A} p_i \\
\text{end} \\
x = x_i
\end{array}
$$

Computing LS-SVM solutions (4)

- **LS-SVM - Large Scale Algorithm**

  1. Solve η, ν from $H\eta = Y$ and $H\nu = d_2$.
  2. Compute $s = Y^T\eta$.
  3. Find solution: $b = \xi_1 = \eta^T d_2 / s$ and $\alpha = \xi_2 = \nu - \eta\xi_1$.

  Note: $\mathcal{A}$ is not stored.

- **Speed of convergence** depends on the condition number

$$\|x_i - x_*\|_{\mathcal{A}} \leq \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^i \|x_0 - x_*\|_{\mathcal{A}}$$

  where x_* is the solution to be reached, $\|v\|_{\mathcal{A}} = (v^T \mathcal{A} v)^{1/2}$, $\kappa = \|\mathcal{A}\| \, \|\mathcal{A}^{-1}\|$ (**condition number**).

Kernel-based models: different perspectives

SVM

LS−SVM

Kriging

RKHS

Gaussian Processes

Some early history on RKHS:

1910-1920: Moore

1940: Aronszajn

1951: Krige

1970: Parzen

1971: Kimeldorf & Wahba

Complementary insights from different perspectives:
kernels are used in different methodologies

Support vector machines (SVM):	optimization approach (primal/dual)
Reproducing kernel Hilbert spaces (RKHS):	functional analysis
Gaussian processes (GP):	probabilistic/Bayesian approach

Estimation in Reproducing Kernel Hilbert Spaces (RKHS)

- Variational problem: [Wahba, 1990; Cucker & Zhou, 2007]
  for given input-output data $\{(x_i, y_i)\}_{i=1}^N$, find function f such that

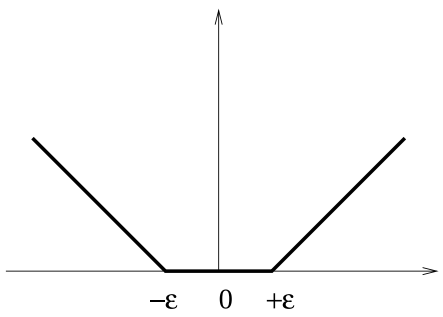
$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda \|f\|_K^2$$

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

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

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

  Reproducing property $f(x) = \langle f, K_x \rangle_K$ with $K_x(\cdot) = K(x, \cdot)$



- Some special cases:
  $L(y, f(x)) = (y - f(x))^2$: least squares
  $L(y, f(x)) = |y - f(x)|_\epsilon$: ϵ-insensitive loss function

Gaussian processes (1)

Given N data points $\{x_k, y_k\}_{k=1}^N$, denote $y_{1\_N} = [y_1; ...; y_N] \in \mathbb{R}^N$
Covariance matrix C with $C_{kl} = C(x_k, x_l)$ and covariance function $C(\cdot, \cdot)$

Consider

$$P(y_{N+1}|y_{1\_N}) = \frac{P(y_{N+1}, y_{1\_N})}{P(y_{1\_N})}$$

with joint density and conditional density assumed to be Gaussian with C_{N+1} the $(N+1) \times (N+1)$ covariance matrix

$$C_{N+1} = \begin{bmatrix} C_N & \theta \\ \theta^T & \nu \end{bmatrix}.$$

This gives

$$P(y_{N+1}, y_{1\_N}) \propto \exp\left(-\frac{1}{2} \begin{bmatrix} y_{1\_N} & y_{N+1} \end{bmatrix} C_{N+1}^{-1} \begin{bmatrix} y_{1\_N} \\ y_{N+1} \end{bmatrix}\right).$$

Gaussian processes (2)

One obtains the posterior distribution

$$P(y_{N+1}|y_{1\_N}) \propto \exp\left(-\frac{1}{2}\frac{(y_{N+1}-\hat{y}_{N+1})^2}{\sigma^2_{\hat{y}_{N+1}}}\right)$$

where

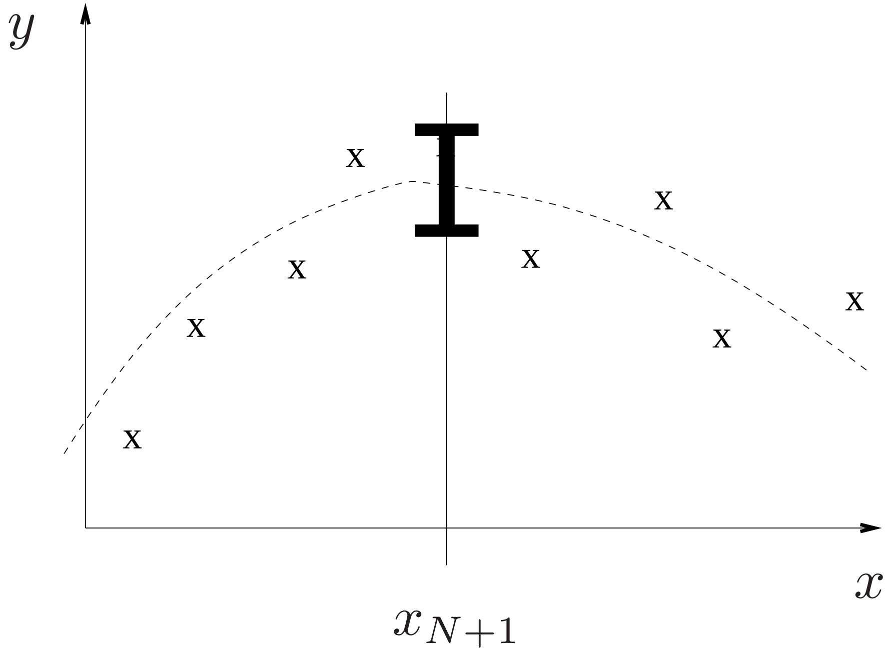
$$\begin{cases} \hat{y}_{N+1} & = & \theta^T C_N^{-1} y_{1\_N} \\ \sigma^2_{\hat{y}_{N+1}} & = & \nu - \theta^T C_N^{-1} \theta \end{cases}$$

with predictive mean $\hat{y}_{N+1}$ and $\sigma_{\hat{y}_{N+1}}$ the error bar.

[MacKay, 1998; Rasmussen & Williams, 2006]

The predictive mean $\hat{y}_{N+1}$ relates to KRR/LS-SVM (zero bias term) for the choice $C(x_k, x_l) = K(x_k, x_l) + \delta_{kl}/\gamma$ [Suykens et al. 2002]

Illustration of error bar on $\hat{y}_{N+1}$



$$\begin{cases} \hat{y}_{N+1} = \theta^T C_N^{-1} y_{1\_N} \\ \sigma^2_{\hat{y}_{N+1}} = \nu - \theta^T C_N^{-1} \theta \end{cases}$$

The error bar characterizes the uncertainty for the prediction.

Automatic relevance determination

- **Input selection:** Which are the most relevant inputs in order to explain the data with respect to the considered model ?

- **Different models can lead to different conclusions about relevance of inputs:** Note that the obtained relevance is not valid in an absolute sense, but only relative with respect to the considered model.

- A choice of C for automatic relevance determination:

$$C(x,z) = \theta_1 \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\frac{(x_i - z_i)^2}{\sigma_i^2}\right) + \theta_2$$

  where $x, z \in \mathbb{R}^n$ and x_i, z_i denote the i-th component of these vectors, θ_1, θ_2 are constants.

- **Interpretation**: σ_i large: irrelevant to the model; σ_i small: i-th input is relevant for the considered model.

LS-SVM and the representer theorem (1)

- In a support vector machines primal-dual optimization formulation context:

$$\min_{w,b,e} J_{\mathrm{P}}(w,e) = \frac{1}{2}w^T w + \gamma \sum_{k=1}^{N} L(e_k)$$

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

where $L(e)$ is a general and differentiable cost function.

- Lagrangian

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

with Lagrange multipliers α_k.

LS-SVM and the representer theorem (2)

- Conditions for optimality:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow \quad w = \sum_{k=1}^{N} \alpha_k \varphi(x_k) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow \quad \sum_{k=1}^{N} \alpha_k = 0 \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 & \rightarrow \quad \alpha_k = \gamma L^{'}(e_k), \qquad\qquad\qquad k = 1, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 & \rightarrow \quad w^T \varphi(x_k) + b + e_k - y_k = 0, \quad k = 1, ..., N. \end{cases}$$

- Note that $\alpha_k = \gamma L^{'}(e_k)$ gives an interpretation about the sparsity property related to the choice of the loss function

LS-SVM and the representer theorem (3)

- After elimination of the variables w and e and application of the kernel trick $K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l)$ one gets

$$\begin{array}{l} \text{solve in } \alpha, b: \\ \alpha_k = \gamma L^{'}(y_k - \sum_{l=1}^{N} \alpha_l K(x_l, x_k) + b) \;, \quad k = 1, ..., N \end{array}$$

  which is a set of nonlinear equations to be solved in α, b. Alternatively, one may also eliminate α instead of e and solve the nonlinear equations in e.

- The resulting dual representation of the model:

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

Krein spaces: indefinite kernels

LS-SVM for indefinite kernel case:

$$\min_{w_+,w_-,b,e} \frac{1}{2}(w_+^T w_+ - w_-^T w_-) + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \text{ s.t. } y_i = w_+^T \varphi_+(x_i) + w_-^T \varphi_-(x_i) + b + e_i, \forall i$$

and **indefinite kernel**

$$K(x_i, x_j) = K_+(x_i, x_j) - K_-(x_i, x_j)$$

with positive definite kernels K_+, K_-

$$K_+(x_i, x_j) = \varphi_+(x_i)^T \varphi_+(x_j) \text{ and } K_-(x_i, x_j) = \varphi_-(x_i)^T \varphi_-(x_j)$$

[X. Huang, Maier, Hornegger, Suykens, ACHA 2017]

[Mehrkanoon, X. Huang, Suykens, Pattern Recognition, 2018]

Related work of RKKS: [Ong et al 2004; Haasdonk 2005; Luss 2008; Loosli et al. 2015]

Banach spaces: tensor kernels

- Regression problem:

$$\begin{array}{cc} \min\limits_{(w,b,e)\in\ell^r(\mathbb{K})\times\mathbb{R}\times\mathbb{R}^N} & \rho(\|w\|_r) + \frac{\gamma}{N}\sum_{i=1}^N L(e_i) \\ \text{subject to} & y_i = \langle w, \varphi(x_i)\rangle + b + e_i\,, \forall i = 1, ..., N \end{array}$$

with $r = \frac{m}{m-1}$ for even $m \geq 2$, ρ convex and even.

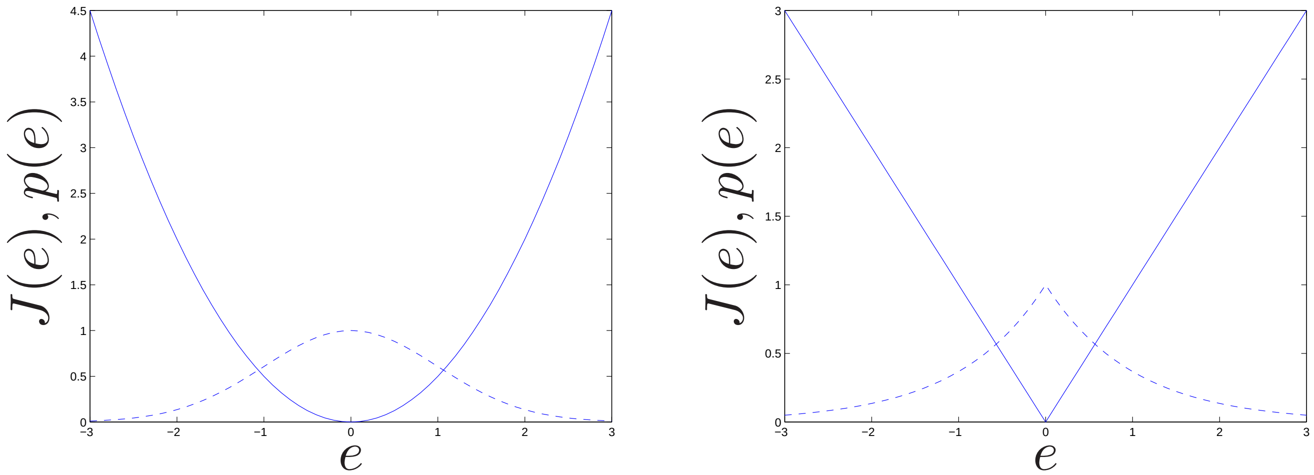
For m large this approaches ℓ^1 regularization.

- **Tensor-kernel representation**

$$\hat{y} = \langle w, \varphi(x)\rangle_{r,r^*} + b = \frac{1}{N^{m-1}} \sum_{i_1,\ldots,i_{m-1}=1}^{N} u_{i_1} \ldots u_{i_{m-1}} K(x_{i_1}, \ldots, x_{i_{m-1}}, x) + b$$

[Salzo & Suykens, arXiv 1603.05876; Salzo, Suykens, Rosasco, AISTATS 2018]
related: RKBS [Zhang 2013; Fasshauer et al. 2015]

Choice of loss function (1)



(Left) In a maximum likelihood setting the least squares cost function (full line) is optimal in case of a Gaussian noise distribution (dashed line); (Right) for a Laplacian noise distribution the L_1 estimator is optimal.

Choice of loss function (2)

- For a model
$$y = f(x) + e$$
the best choice of the cost function depends on the given noise model (in a maximum likelihood setting)

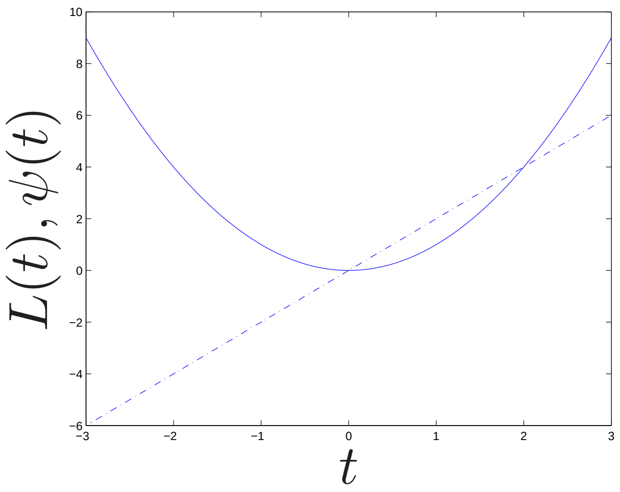
- Best choice of the cost function
$$J(e) = -\log p(e)$$

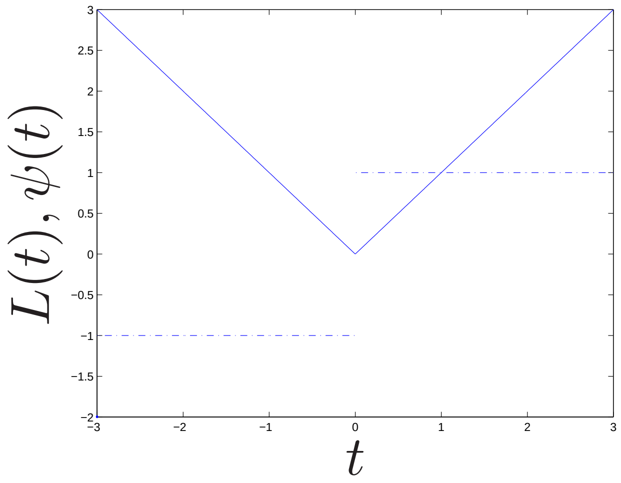
- The Gaussian noise model $p(e) = \exp(-\frac{1}{2}e^2)$ corresponds to cost function (L_2 estimator)
$$J(e) = \frac{1}{2}e^2$$
The Laplacian noise model $p(e) = \exp(-|e|)$ corresponds to the L_1-estimator
$$J(e) = |e|$$

Robustness of a loss function

- A drawback of least squares (L_2 estimator) is that it is less robust against outliers and non-Gaussian noise.

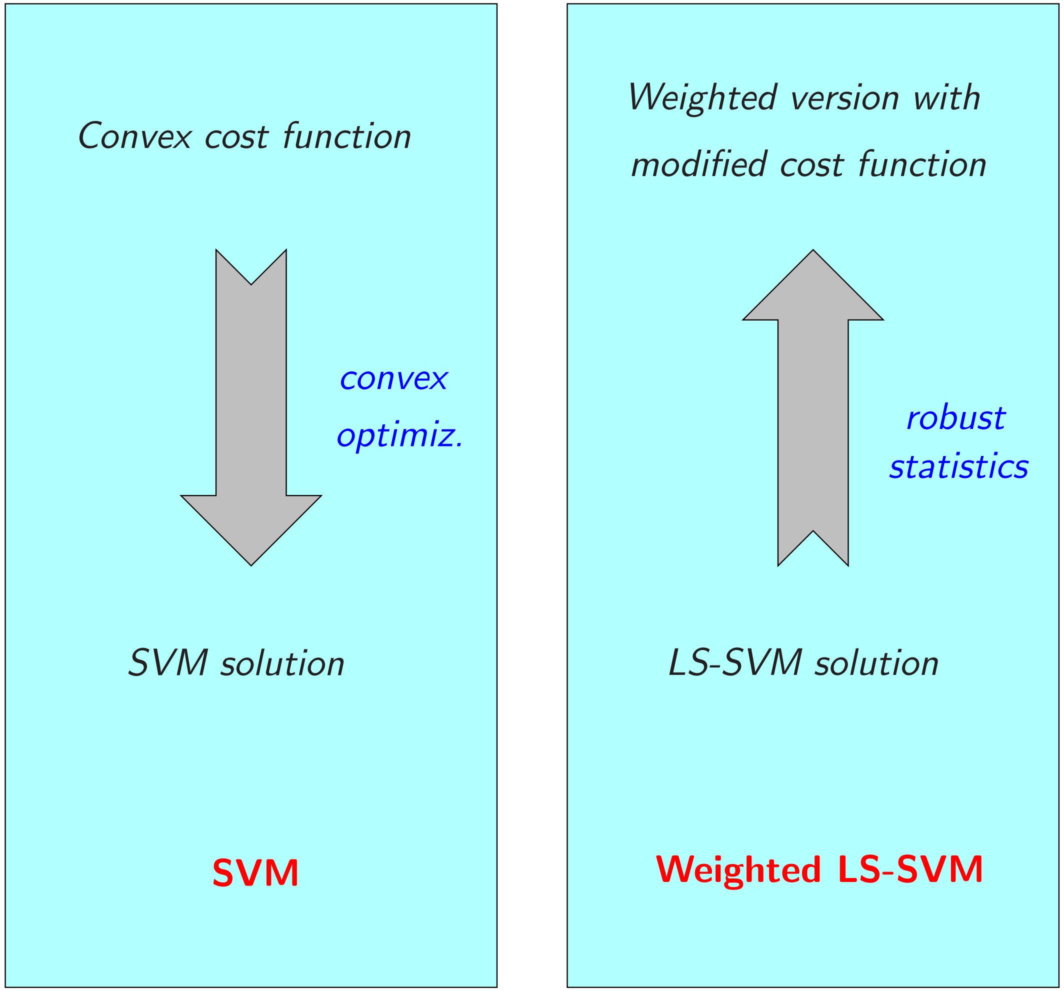
- The gradient of the cost function is important at this point





The L_1-estimator reduces the influence of outliers, while outliers are too much emphasized in the cost function of the L_2 estimator.

Robustness


Convex cost function
convex
optimiz.
SVM solution
SVM
Weighted version with
modified cost function
robust
statistics
LS-SVM solution
Weighted LS-SVM

Weighted LS-SVM for robustness (1)

- Optimization problem for weighted LS-SVM:

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

$$\text{subject to} \quad y_k = {w^\star}^T \varphi(x_k) + b^\star + e_k^\star, \; k = 1, ..., N.$$

with $\mathcal{L}(w^\star, b^\star, e^\star; \alpha^\star) = J_{\mathrm{P}}(w^\star, e^\star) - \sum_{k=1}^{N} \alpha_k^\star \{ {w^\star}^T \varphi(x_k) + b^\star + e_k^\star - y_k \}$.

- KKT system: solve in $\alpha^\star, b^\star$

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

with diagonal matrix $V_\gamma = \mathrm{diag}([\frac{1}{\gamma v_1}; ...; \frac{1}{\gamma v_N}])$.

Weighted LS-SVM for robustness (2)

- The choice of the weights v_k is determined based upon the error variables $e_k = \alpha_k/\gamma$ resulting from the unweighted LS-SVM case.

- Robust estimates are obtained e.g. by (based on robust statistics)

$$v_k = \begin{cases} 1 & \text{if} \quad |e_k/\hat{s}| \leq c_1 \\ \dfrac{c_2 - |e_k/\hat{s}|}{c_2 - c_1} & \text{if} \quad c_1 \leq |e_k/\hat{s}| \leq c_2 \\ 10^{-4} & \quad\quad \text{otherwise} \end{cases}$$

where $\hat{s} = \frac{\text{IQR}}{2 \times 0.6745}$ is a robust scale estimator (a robust estimate of the standard deviation of the LS-SVM error variables e_k). The interquartile range IQR is the difference between the 75th and 25th percentile. The constants c_1, c_2 are typically chosen as $c_1 = 2.5$ and $c_2 = 3$.

Weighted LS-SVM for robustness (3)

Weighted LS-SVM algorithm:

1. Given training data $\{x_k, y_k\}_{k=1}^N$, estimate an unweighted LS-SVM. Compute $e_k = \alpha_k/\gamma$ from the solution vector.

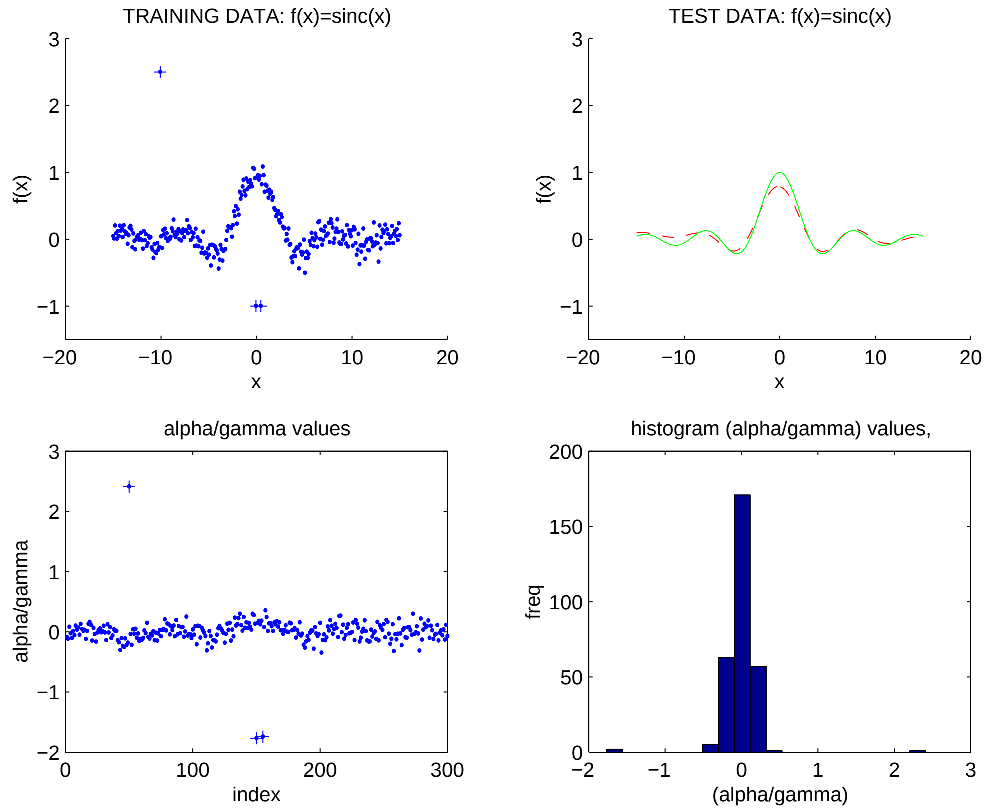
2. Compute a robust estimate of the standard deviation $\hat{s}$ based on the empirical e_k distribution.

3. Determine the weights v_k based upon e_k and $\hat{s}$.

4. Solve the Weighted LS-SVM system, giving the model

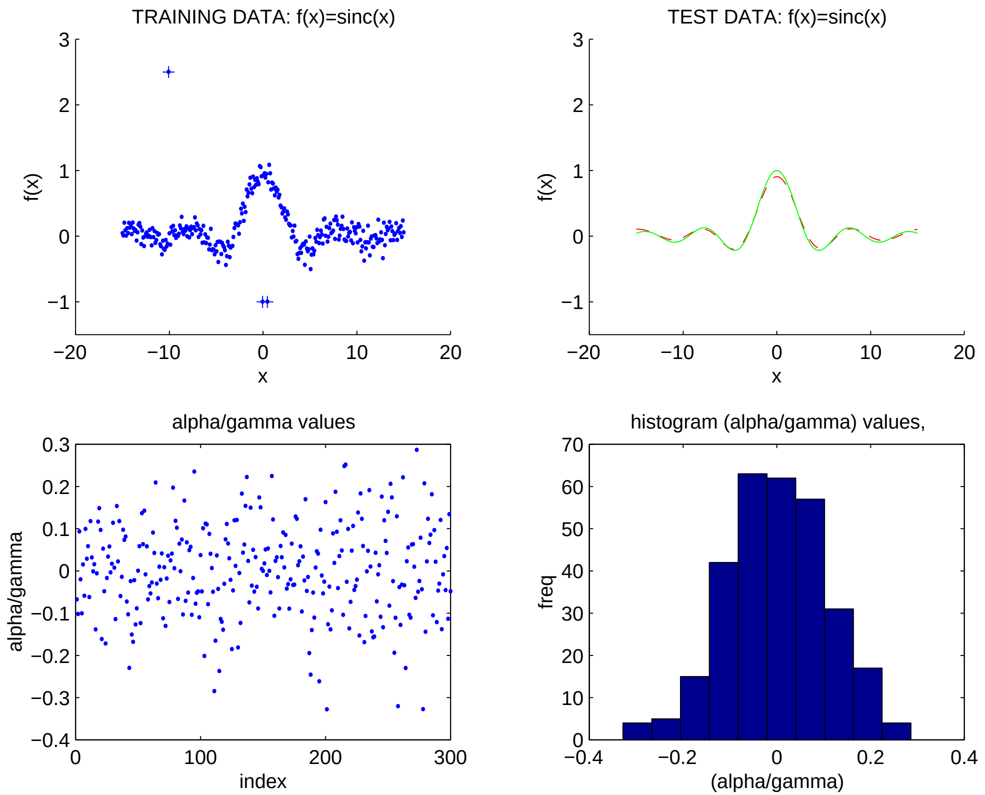
$$y(x) = \sum_{k=1}^{N} \alpha_k^\star K(x, x_k) + b^\star.$$

Weighted LS-SVM: example (1)



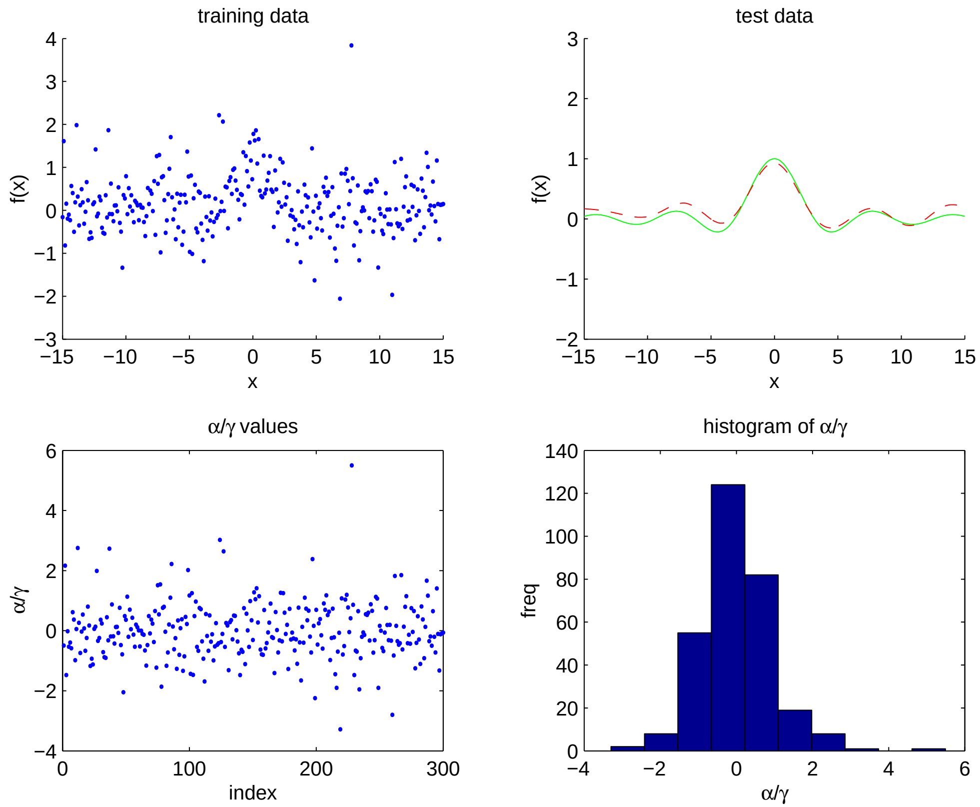
Estimation of sinc function by LS-SVM with RBF kernel, given 300 training data, corrupted by zero mean Gaussian noise and 3 outliers (denoted by '+'). (Top-Left) Training data; (Top-Right) resulting LS-SVM model on independent test set: (solid line) true function, (dashed line) estimate; (Bottom-Left) $e_k = \alpha_k/\gamma$; (Bottom-Right) histogram of e_k.

Weighted LS-SVM: example (2)



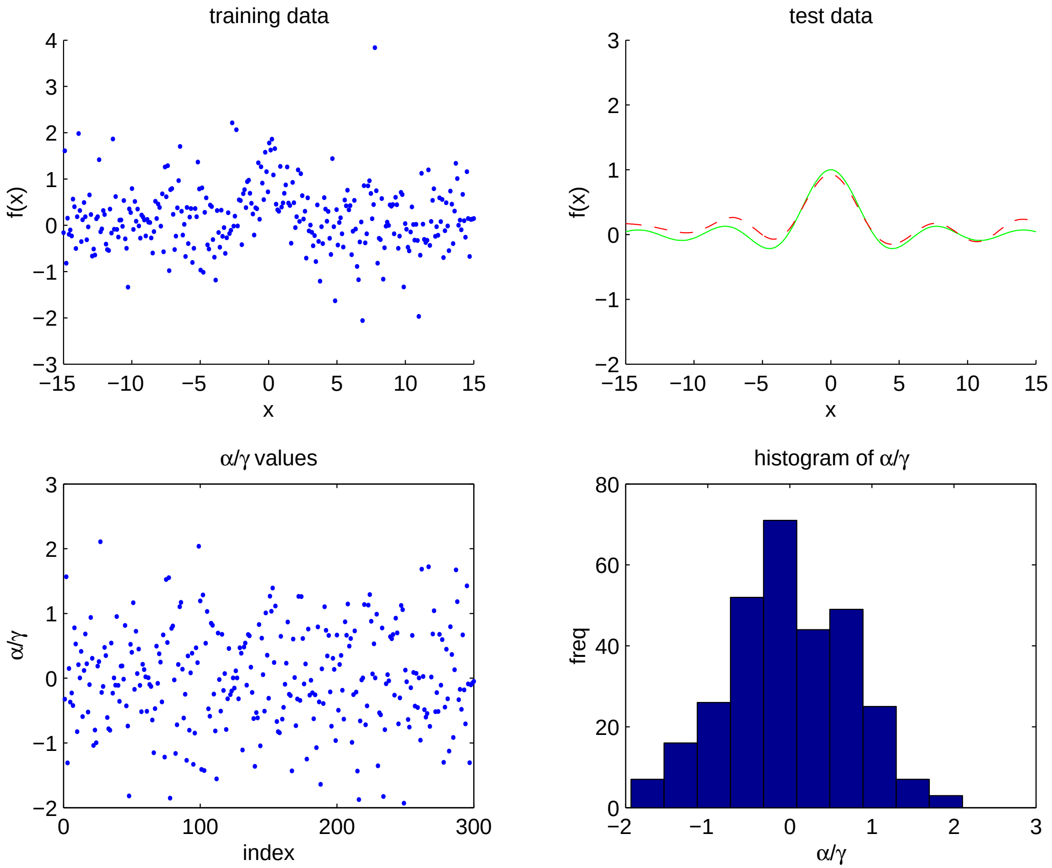
(Continued) **Weighted LS-SVM** applied to the results of the previous figure. The e_k distribution becomes Gaussian and the generalization performance on the test data improves.

Weighted LS-SVM: example (3)



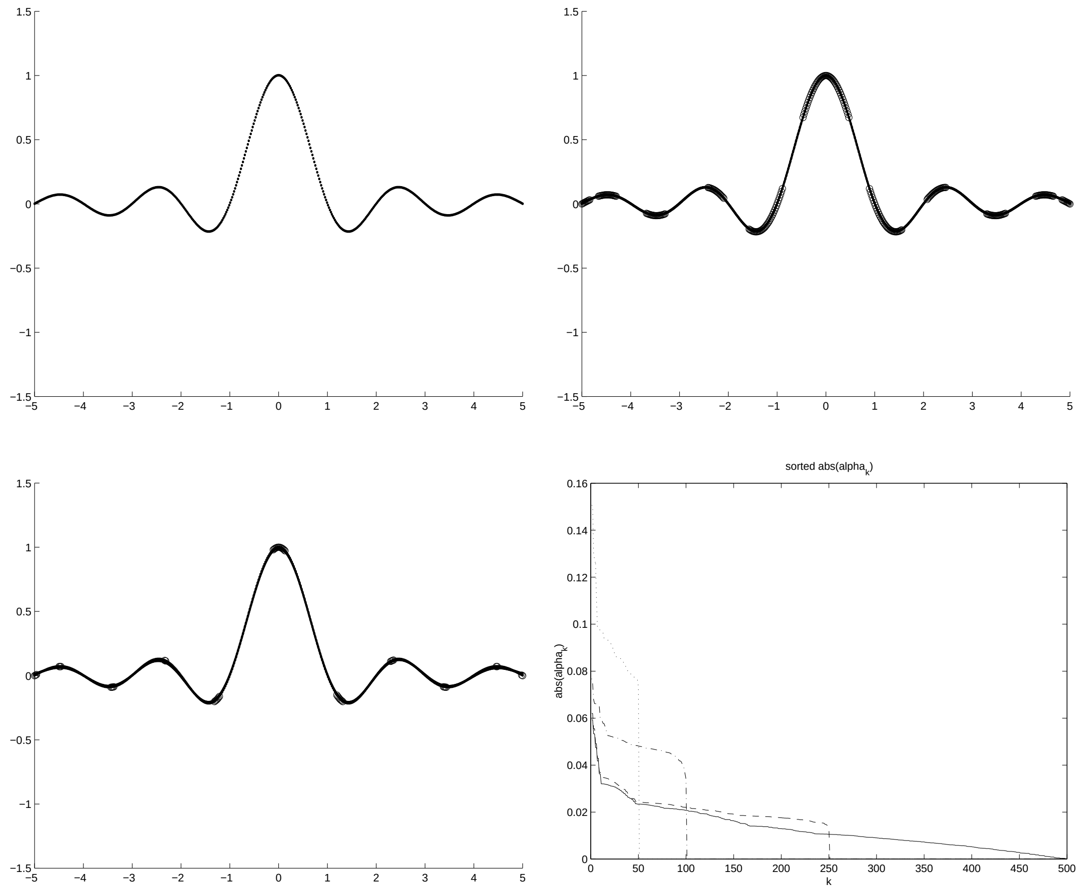
Estimation of a sinc function by LS-SVM with RBF kernel, given 300 training data, corrupted by a central t-distribution with heavy tails. (Top-Left) Training data; (Top-Right) resulting LS-SVM model on independent test set: (solid line) true function, (dashed line) estimate; (Bottom-Left) $e_k = \alpha_k/\gamma$; (Bottom-Right) histogram of e_k.

Weighted LS-SVM: example (4)



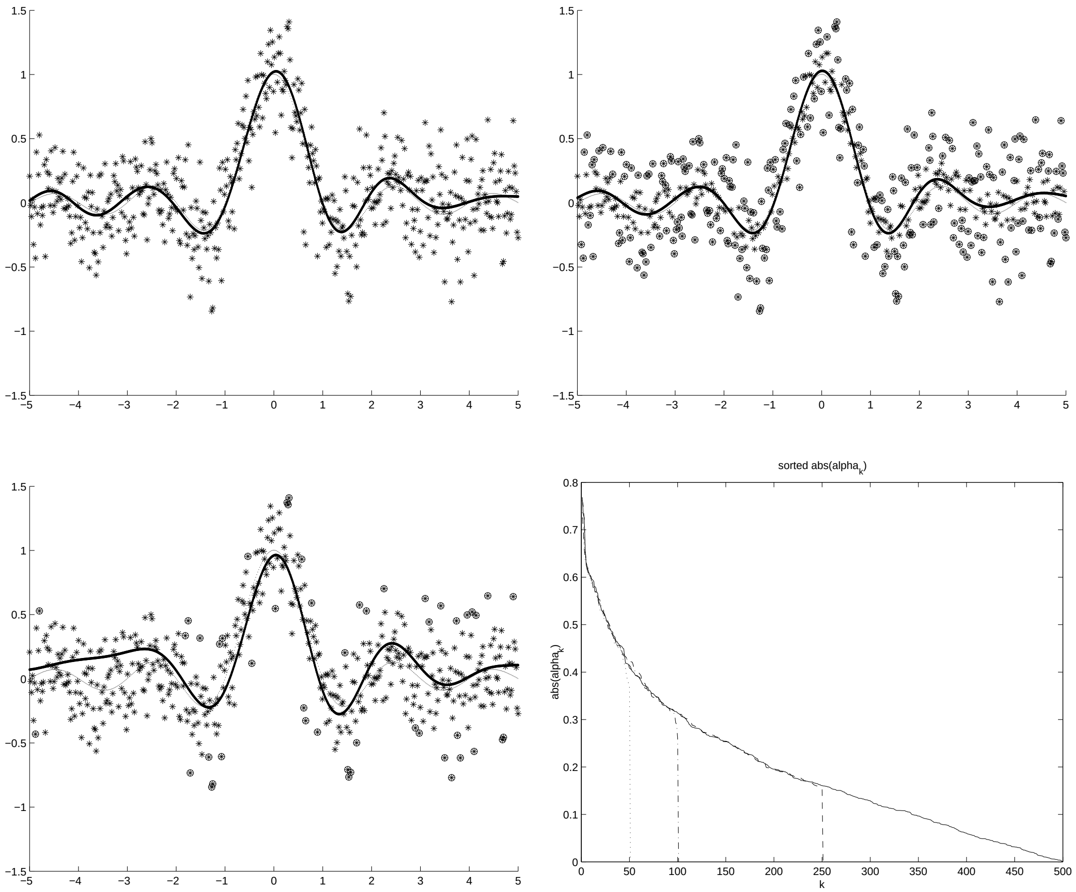
(Continued) **Weighted LS-SVM** applied to the results of the previous figure. The e_k distribution becomes Gaussian and the generalization performance on the test data improves.

Sparseness by pruning (1)



(500 SV $\rightarrow$ 250 SV $\rightarrow$ 50 SV)

Sparseness by pruning (2)



(500 SV $\rightarrow$ 250 SV $\rightarrow$ 50 SV)