Large scale methods

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 8

Contents

- SVM large scale methods
- Nyström method (GP)
- Fixed size LS-SVM
- Random Fourier features
- Committee networks
- Multilayer approaches

SVM: large scale methods

- Chunking and decomposition methods
- Sequential minimal optimization (SMO)
- Distributed optimization
- Coordinate descent method
- Frank-Wolfe method
- On-line learning, stochastic learning
- Ensemble methods

SMO

- Sequential minimal optimization (SMO) [Platt, 1998]

- Consider dual problem of SVM:

You get analytical solution by considering 2 datapoints each time

$$\begin{array}{ll} \max\limits_{\alpha} & \sum\limits_{i} \alpha_i - \frac{1}{2} \sum\limits_{i,j} y_i y_j K(x_i, x_j) \alpha_i \alpha_j \\ \text{subject to} & 0 \leq \alpha_i \leq C, \forall i \\ & \sum_i y_i \alpha_i = 0 \end{array}$$

SMO method:
Consider subproblems of very small size (size 2);
This QP problem can be solved analytically;
Find a Lagrange multiplier α_1 that violates the KKT conditions
Pick a second α_2 and optimize the pair (α_1, α_2)
Repeat the procedure until convergence

Pegasos (1)

Subgradient is used when the loss function is non-differentiable like Huber loss. You generalise the concept of a gradient.

- Pegasos (Primal estimated subgradient solver for SVM) [Shalev-Shwartz et al., ICML 2007]:

  Objective function for SVM with hinge loss

$$\frac{\lambda}{2} w^T w + \frac{1}{m} \sum_{(x,y) \in \mathcal{A}_t} L(w, (x, y))$$

  hinge loss $L(w, (x, y)) = \max\{0, 1 - y \langle w, x \rangle\}$
  $\mathcal{A}_t$ random subsample at iteration t
  decision function $\hat{y} = \mathrm{sign}[\langle w, x \rangle]$

Pegasos (2)

Algorithm 1: Pegasos with hinge loss

```
Data: $\mathcal{S}, \lambda, T, k, \epsilon$
Select $w_1$ randomly s.t. $\|w^{(1)}\| \leq 1/\sqrt{\lambda}$
for $t = 1 \to T$ do
    Set $\eta_t = \frac{1}{\lambda t}$
    Select $\mathcal{A}_t \subseteq \mathcal{S}$, where $|\mathcal{A}_t| = k$
    $\rho = \frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{A}_t} (y - \langle w_t, x \rangle), \forall i$
    $\mathcal{A}_t^+ = \{(x, y) \in \mathcal{A}_t : y(\langle w_t, x \rangle + \rho) < 1\}, \forall i$
    $w_{t+\frac{1}{2}} = w_t - \eta_t(\lambda w_t - \frac{1}{k} \sum_{(x,y) \in \mathcal{A}_t^+} yx)$
    $w_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|w_{t+\frac{1}{2}}\|}\right\} w_{t+\frac{1}{2}}$
    if $\|w_{t+1} - w_t\| \leq \epsilon$ then
        return $(w_{t+1}, \frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} (y - \langle w_t, x \rangle))$
    end
end
return $(w_{T+1}, \frac{1}{|\mathcal{S}|} \sum_{(x,y) \in \mathcal{S}} (y - \langle w_t, x \rangle))$
```

Nyström method (1)

- Nyström method in Gaussian processes [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 $\Omega_{(M,M)}$:

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

Imagine that N is very large, like a million. You work with a smaller subset M eg. size 1000. You relate kernel matrices to both sets of data points.
Is there some relation between the eigenvalue decompositions of the two?

Nyström method (2)

- Relation to eigenvalues and eigenfunctions of the integral equation

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

Phi = eigenfunction, p = probability, lambda = corresponding eigenvalues.
You do the integral over an infinite amount of data points.

is given by

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

Out of sample extension ... you can do this for any x' eg. from test set.

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 $\overline{U}$.

Nyström method (3)

- For the big matrix:

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

Furthermore, one has

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

One can show then that

You only have to take the inverse for the smaller matrix. Note that it's an approximation, not an equality.

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

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

Nyström method (4)

- The approximate solution to the big linear system

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

can be written as

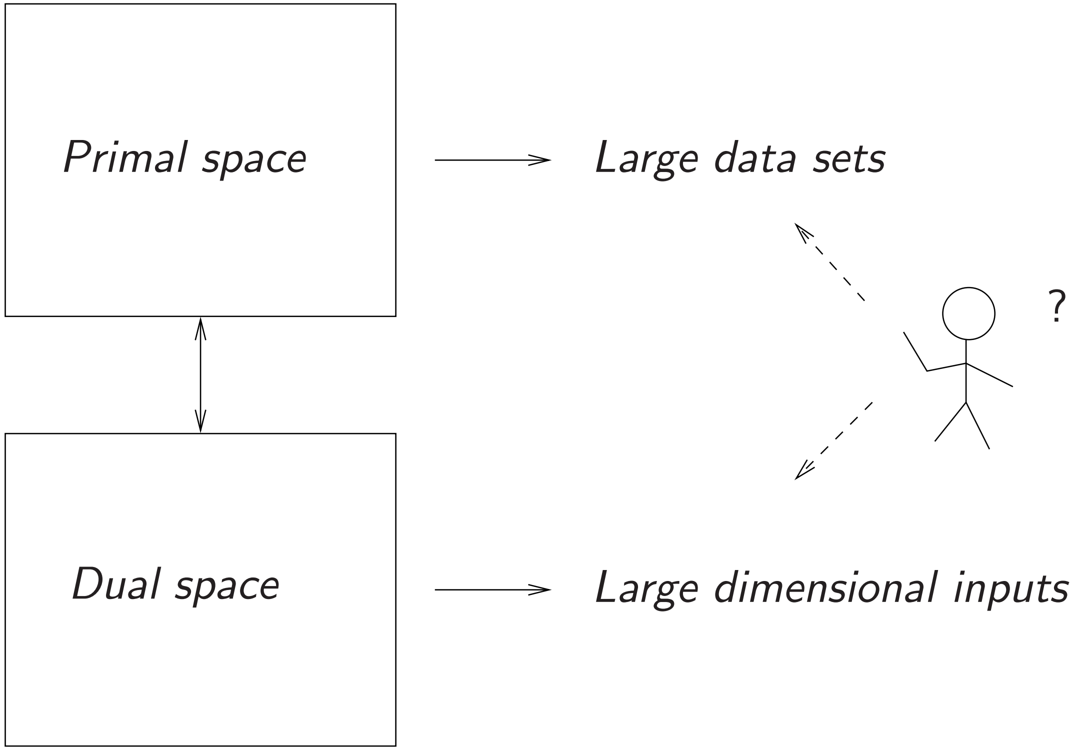
$$\alpha = \gamma \left(y - \tilde{U}(\frac{1}{\gamma}I + \tilde{\Lambda}\tilde{U}^T\tilde{U})^{-1}\tilde{\Lambda}\tilde{U}^T y \right)$$

by applying Sherman-Morrison-Woodbury formula

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

You don't have a sparsity property, also.
alpha will be N-dimensional and not sparse

Computation in primal or dual space?



Fixed Size LS-SVM (1)

- Model in primal space:

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

For a linear model one can solve the primal problem (one knows the feature map: $\varphi(x_k) = x_k$)

- Can we do this for the nonlinear case too?

- Employ the Nyström method to get an approximation to the feature map

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

assuming a fixed size M.

Uses out of sample extension (slide 7)
phi-tilde is an approximation of the feature map
and its dimension is M-dimensional
So w will be M-dimensional
The subset plays the role of the support vectors
so in that sense you get sparsity.

Fixed Size LS-SVM (2)

- The model becomes then

In essence we defined an explicit approximation of a feature map so we applied the Kernel trick in the opposite way.

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

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

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

when ones represent the model as $y(x) = \sum_{k=1}^{M} \alpha_k K(x_k, x) + b.$

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

You can define a Kernel function by taking the dot product of the feature maps ...

Selection of subset

- random
- quadratic Renyi entropy

Random samples could be successful if you're lucky but you might be unlucky,
so in that sense it may not be preferable depending on what you want.

Fixed Size LS-SVM: Selection of SV

- Link between Nyström method, kernel PCA, density estimation and entropy criteria [Girolami, 2002]. The quadratic Renyi entropy

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

Note that in big data the method shouldn't just be good, it should be FAST

has been related to kernel PCA and density estimation with

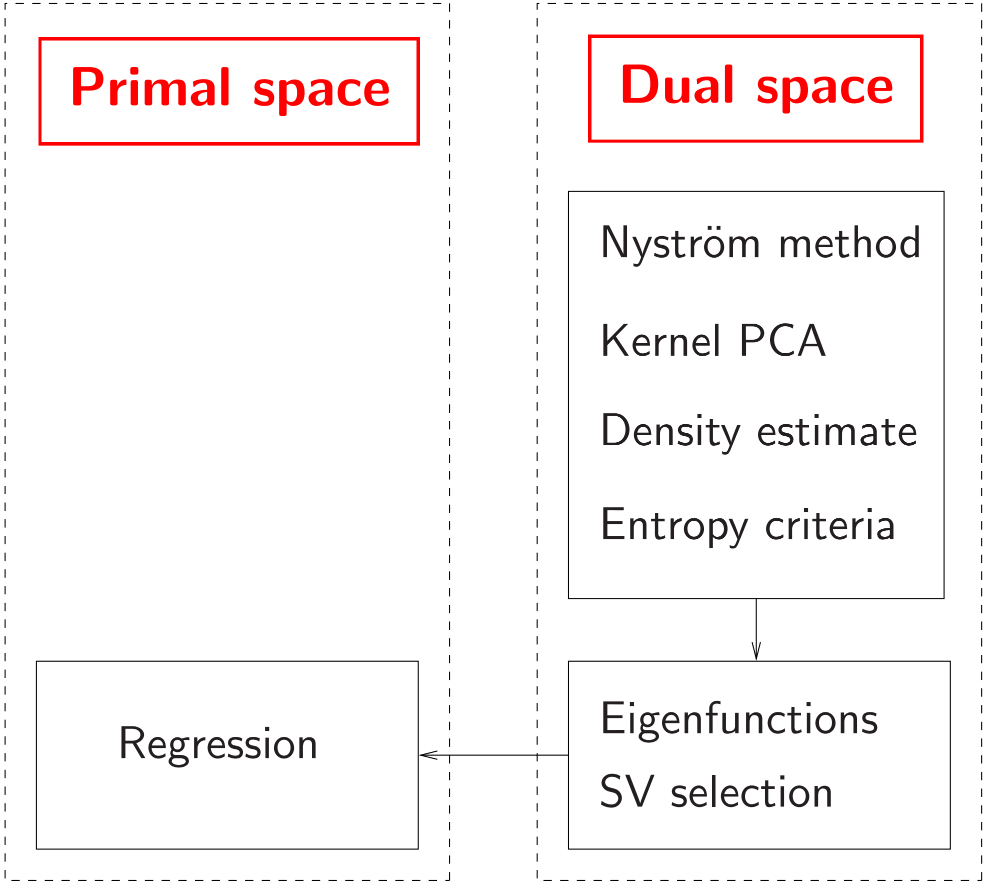
Here you estimate the Renyi entropy by summing over all the elements of the Kernel matrix.

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

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 kernel method



Primal space

Dual space

Nyström method

Kernel PCA

Density estimate

Entropy criteria

Eigenfunctions

SV selection

Regression

Modelling in view of primal-dual representations
Link Nyström approximation (GP) - kernel PCA - density estimation

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

Fixed-size method: using quadratic Renyi entropy (1)

Algorithm:

1. Given N training data

2. Choose a working set with fixed size M (i.e. M support vectors) (typically $M \ll N$).

3. Randomly select a SV $\boxed{x^*}$ from the working set of M support vectors.

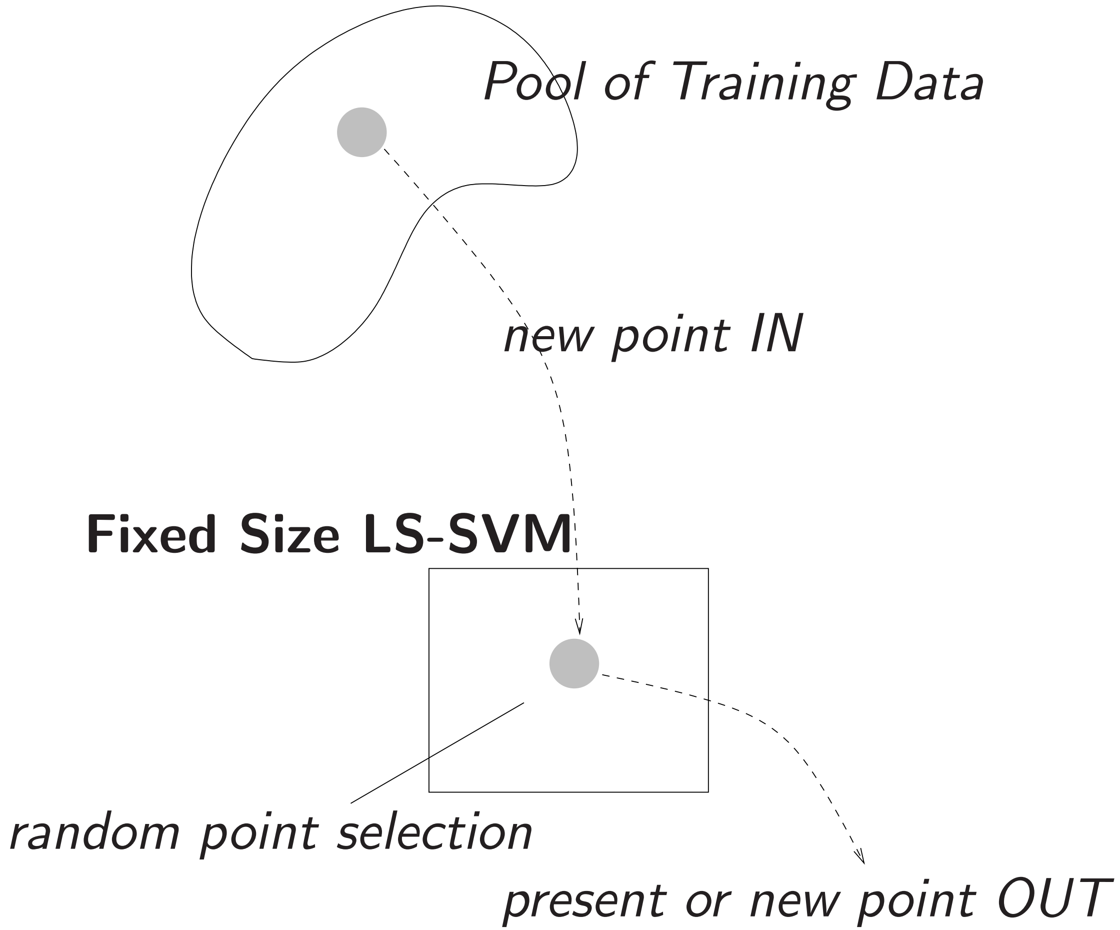
4. Randomly select a point $\boxed{x^{t*}}$ from the training data and replace x^* by x^{t*}.
   If the entropy increases by taking the point x^{t*} instead of x^* then this point x^{t*} is accepted for the working set of M SVs, otherwise the point x^{t*} is rejected (and returned to the training data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

5. Calculate the entropy value for the present working set. The quadratic Renyi entropy equals $H_R = -\log \frac{1}{M^2} \sum_{ij} \Omega_{(M,M)_{ij}}$.

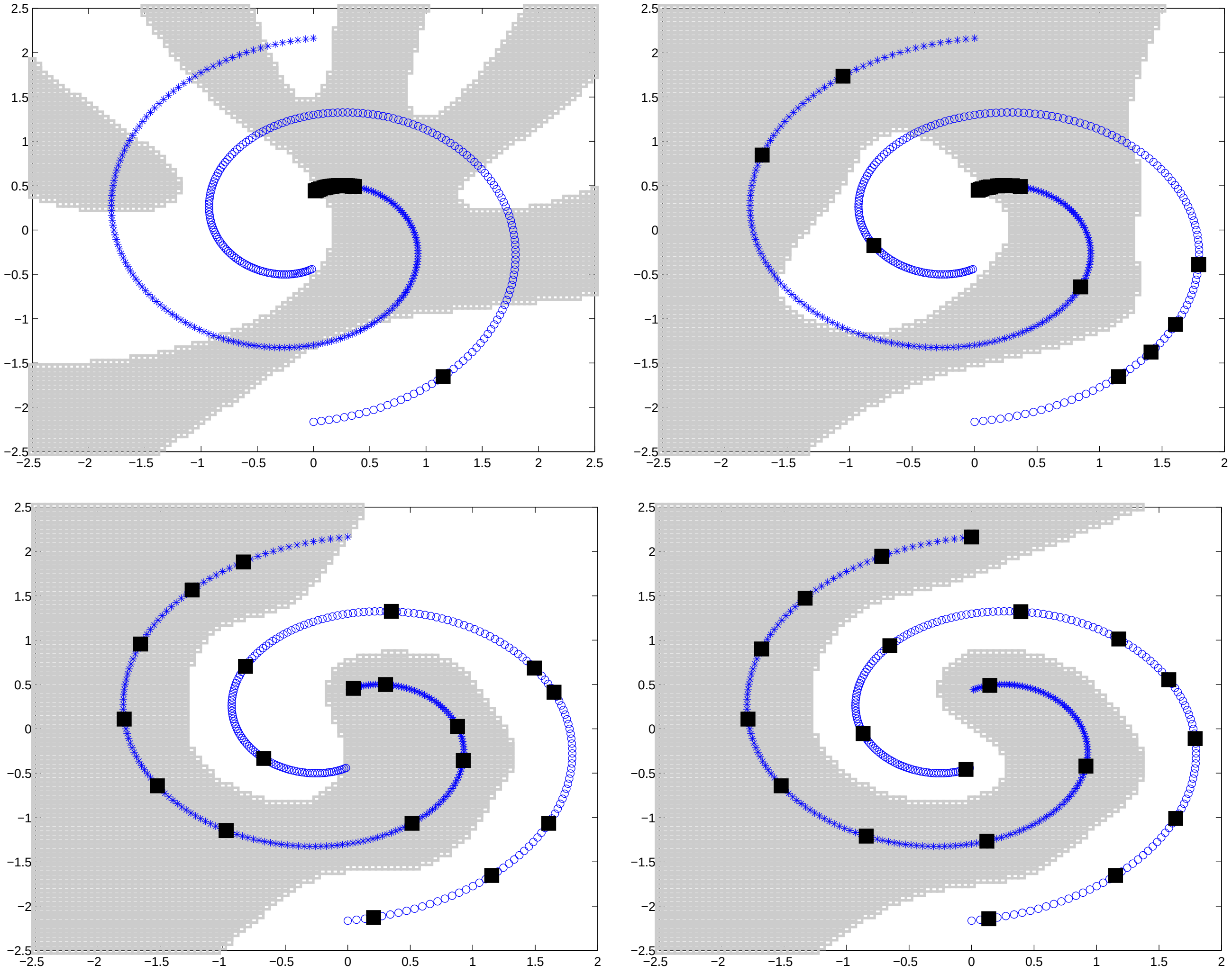
[Suykens et al., 2002]

Fixed-size method: using quadratic Renyi entropy (2)

You pick the data point providing it improves the entropy criterion.

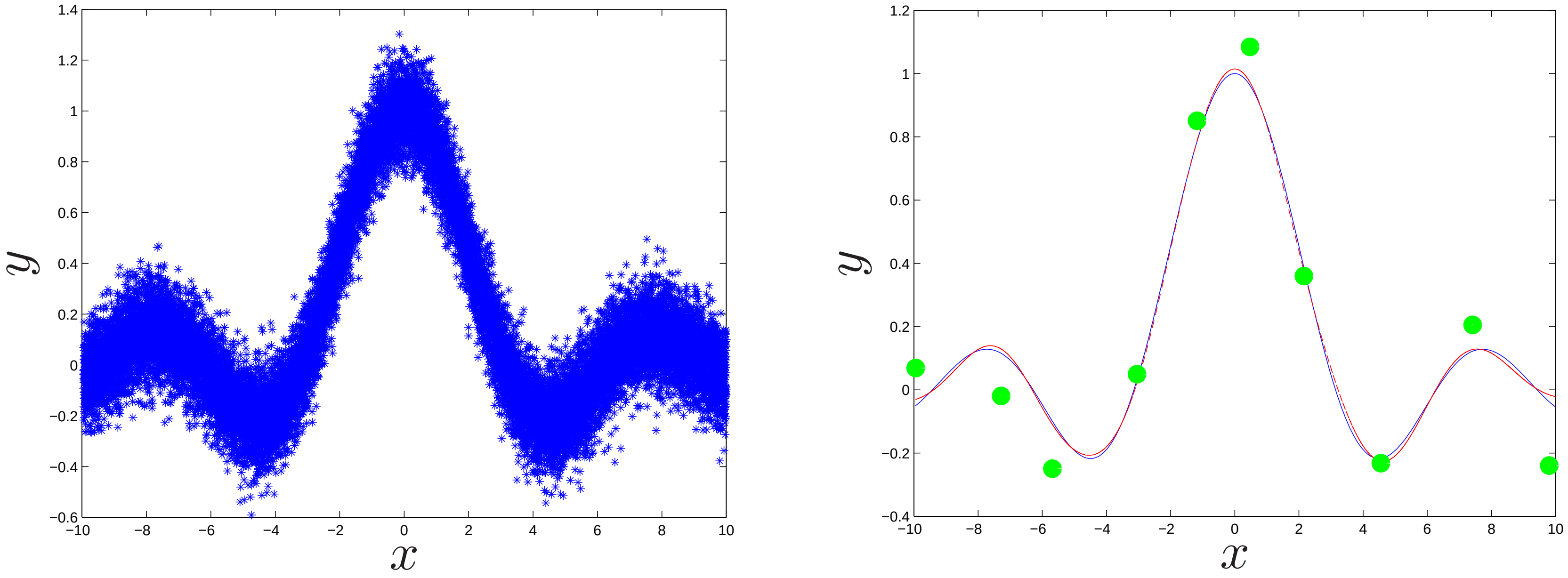


Subset selection using quadratic Renyi entropy: example



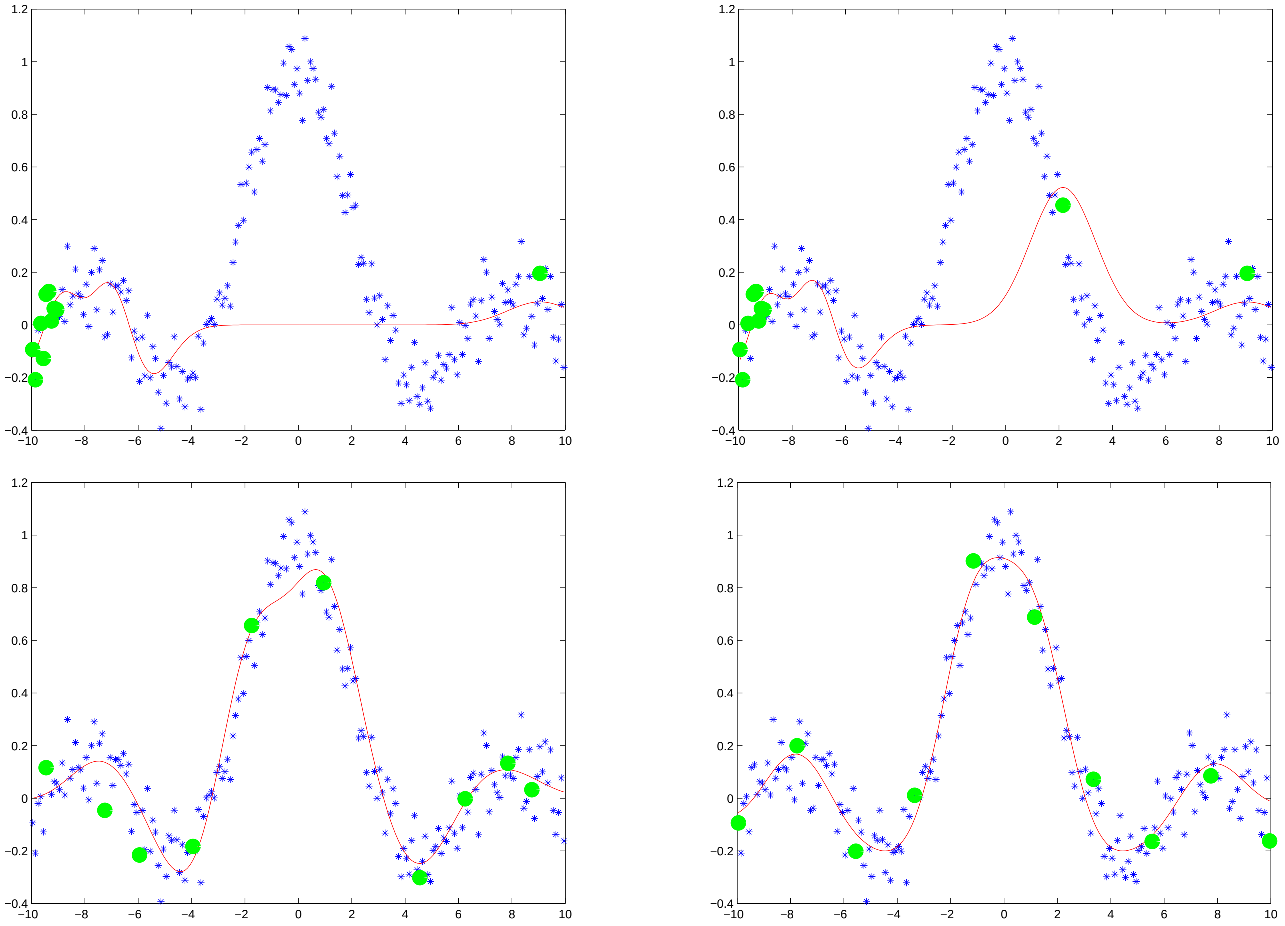
Fixed-size LS-SVM: regression example (1)

Sinc function (20.000 data, 10 SV)



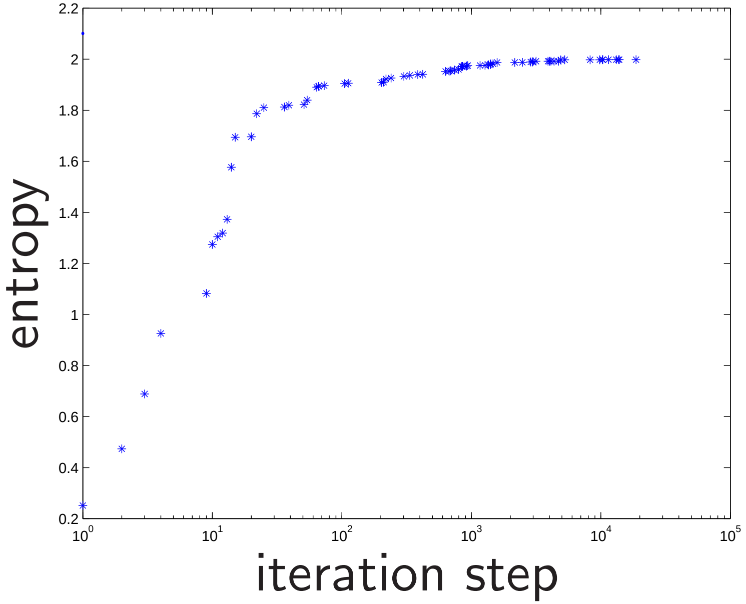
Green dots = support vectors
They're equidistant

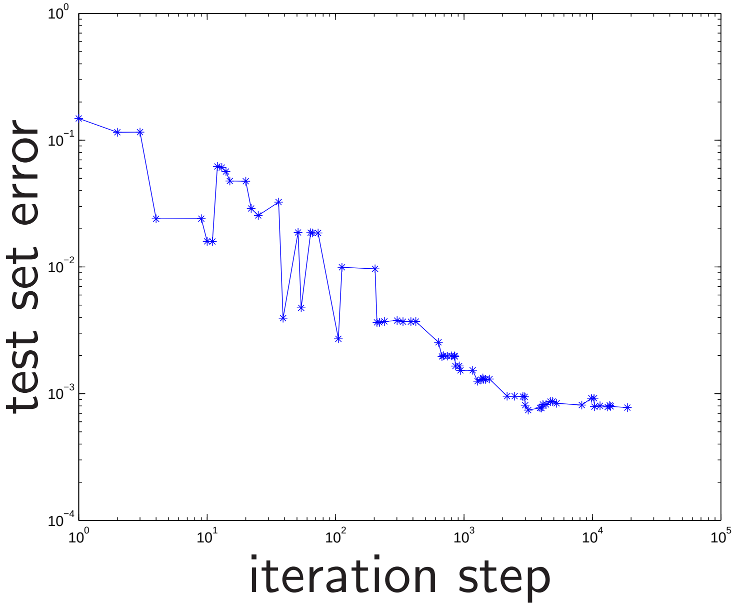
Fixed-size LS-SVM: regression example (2)



Example of intentionally choosing bad data points and then Renyi entropy

Fixed-size LS-SVM: regression example (3)





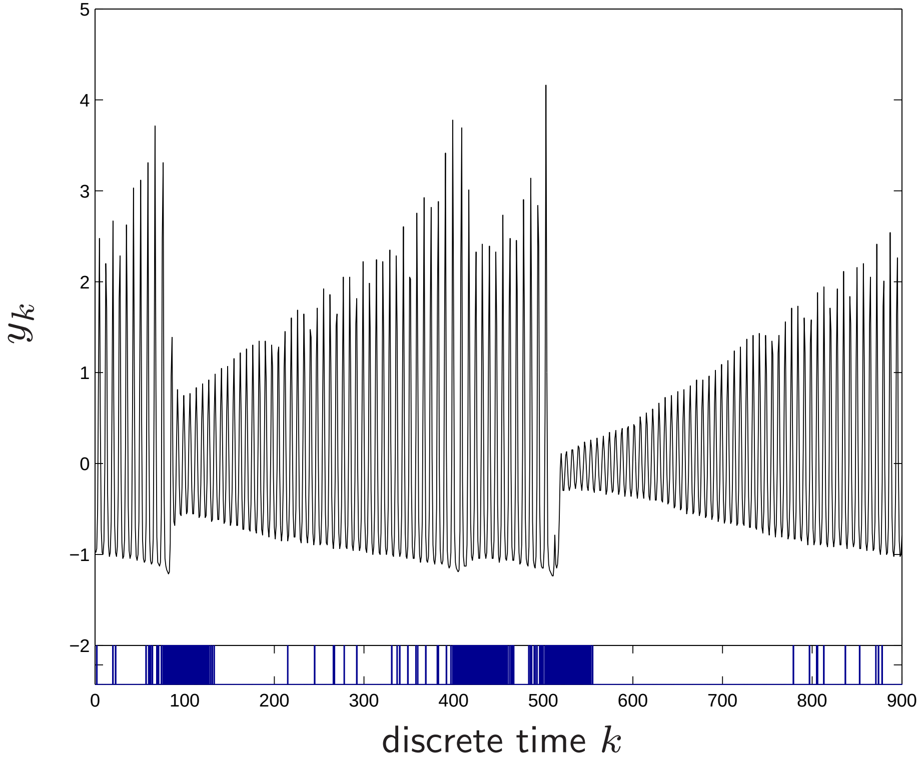
By construction the entropy can only improve.
You can stop earlier since a random sample is usually 'not too bad'.

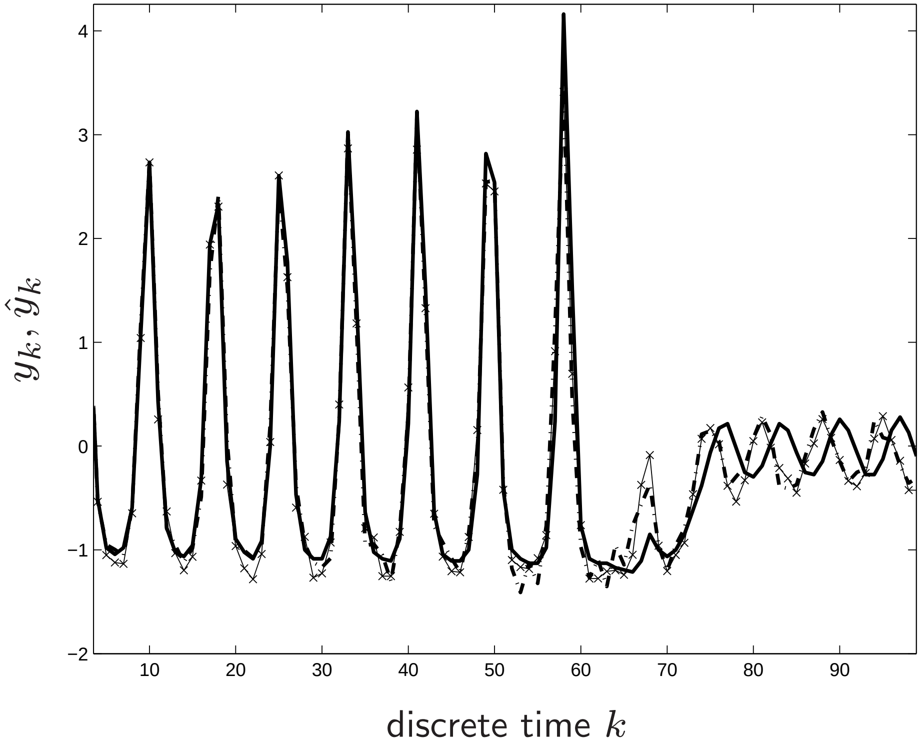
Fixed-size method: example time-series prediction

Bar chart at the bottom indicates where the support vectors are

More support vectors in regions where there are spikes;

Santa Fe laser data



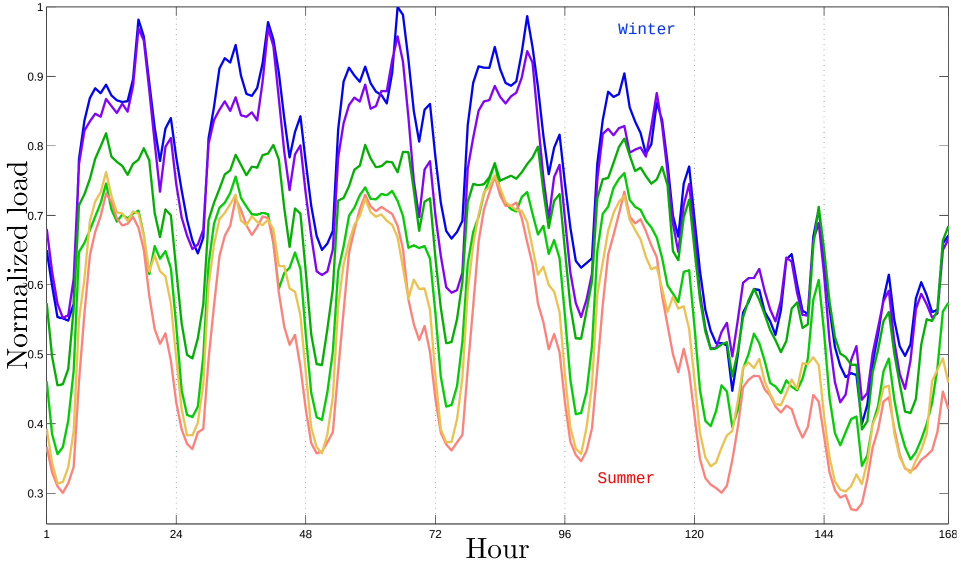


Training: $\hat{y}_{k+1} = f(y_k, y_{k-1}, ..., y_{k-p})$

Iterative prediction: $\hat{y}_{k+1} = f(\hat{y}_k, \hat{y}_{k-1}, ..., \hat{y}_{k-p})$

[Espinoza et al., 2003]

Electricity load forecasting



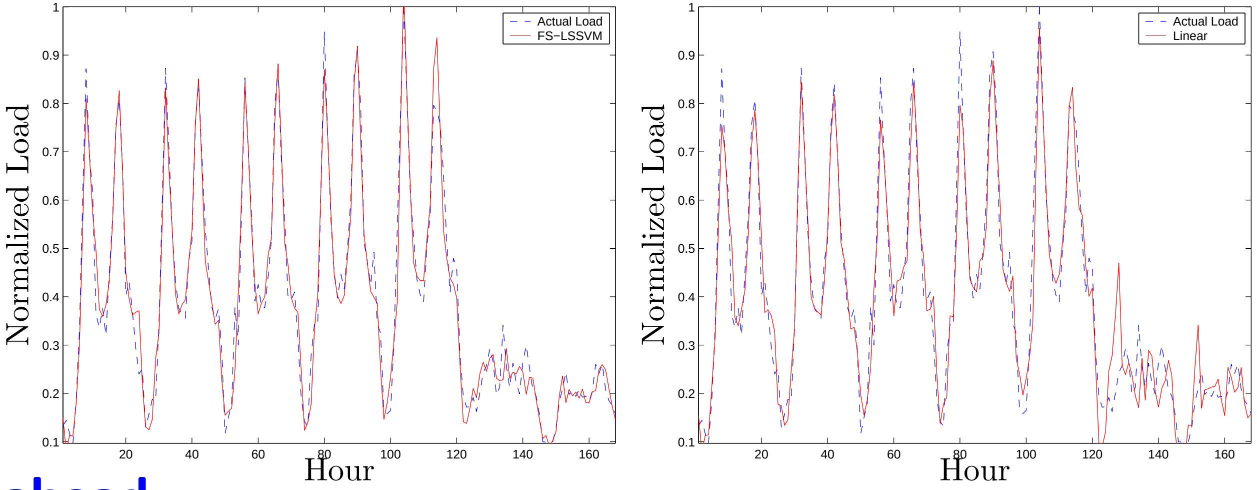
Short-term load forecasting, important for power generation decisions
Hourly load from substations in Belgian grid (ELIA transmission operator)
Seasonal/weekly/intra-daily patterns [Espinoza et al., IEEE CSM 2007]

NARX and AR-NARX model structures: 98 explanatory variables:

- *lagged load values previous two days (48)*
- *effect of temperature on cooling and heating requirements (3)*
- *calendar information: month, day, hour indications (43)*

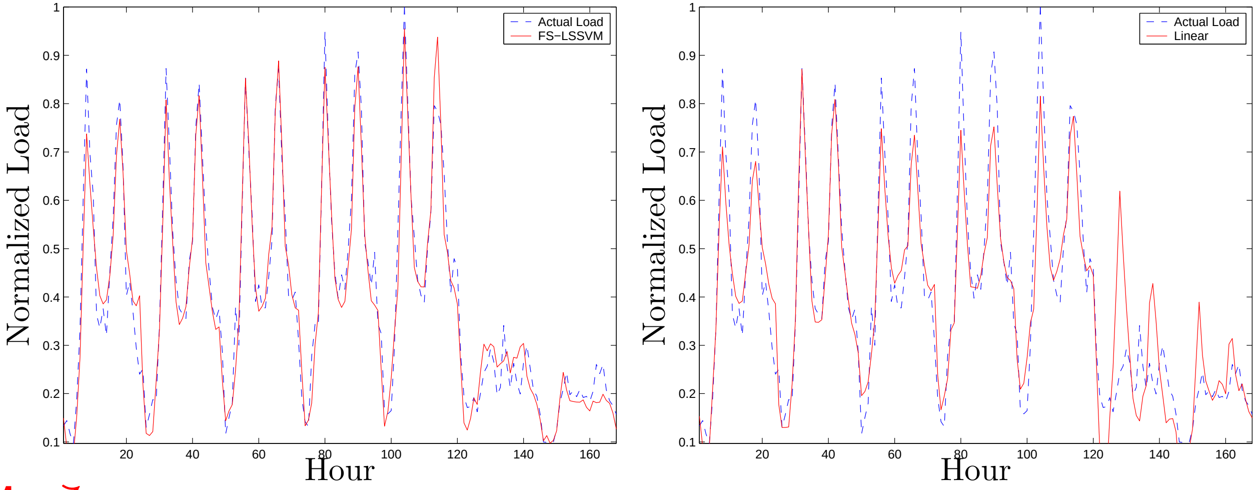
Electricity load forecasting (2)

1-hour ahead



(a) (b)

24-hours ahead



(c) (d)

Fixed-size LS-SVM ↗

↖ Linear ARX model

[Espinoza, Suykens, Belmans, De Moor, IEEE CSM 2007]

Peaks are predicted quite nicely by fixed-size LSSVM (it is nonlinear and that seems to be important).

Fixed-size method: performance in classification

	pid	spa	mgt	adu	ftc
N	768	4601	19020	45222	581012
N_{cv}	512	3068	13000	33000	531012
N_{test}	256	1533	6020	12222	50000
d	8	57	11	14	54
FS-LSSVM (# SV)	150	200	1000	500	500
C-SVM (# SV)	290	800	7000	11085	185000
ν-SVM (# SV)	331	1525	7252	12205	165205
RBF FS-LSSVM	76.7(3.43)	92.5(0.67)	86.6(0.51)	85.21(0.21)	81.8(0.52)
Lin FS-LSSVM	77.6(0.78)	90.9(0.75)	77.8(0.23)	83.9(0.17)	75.61(0.35)
RBF C-SVM	75.1(3.31)	92.6(0.76)	85.6(1.46)	84.81(0.20)	81.5(no cv)
Lin C-SVM	76.1(1.76)	91.9(0.82)	77.3(0.53)	83.5(0.28)	75.24(no cv)
RBF ν-SVM	75.8(3.34)	88.7(0.73)	84.2(1.42)	83.9(0.23)	81.6(no cv)
Maj. Rule	64.8(1.46)	60.6(0.58)	65.8(0.28)	83.4(0.1)	51.23(0.20)

- Fixed-size (FS) LSSVM: good performance and sparsity wrt C-SVM and ν-SVM
- Challenging to achieve high performance by very sparse models

[De Brabanter et al., CSDA 2010]

Two stages of sparsity

primal	
dual	subset selection Nyström approximation

You can further improve sparsity of the model.

Two stages of sparsity

	stage 1
primal	FS model estimation
	↑
dual	subset selection Nyström approximation

Two stages of sparsity

	stage 1		stage 2
primal	FS model estimation	→	reweighted ℓ_1
	↑		
dual	subset selection Nyström approximation		

Synergy between parametric & kernel-based models

[Mall & Suykens, IEEE-TNNLS, 2015], reweighted ℓ_1 [Candes et al., 2008]

This is related to regularization techniques.

Random Fourier Features

- Proposed by [Rahimi & Recht, 2007].

- It requires a positive definite shift-invariant kernel $K(x, y) = K(x - y)$. One obtains a randomized feature map $z(x) : \mathbb{R}^d \to \mathbb{R}^{2D}$ so that

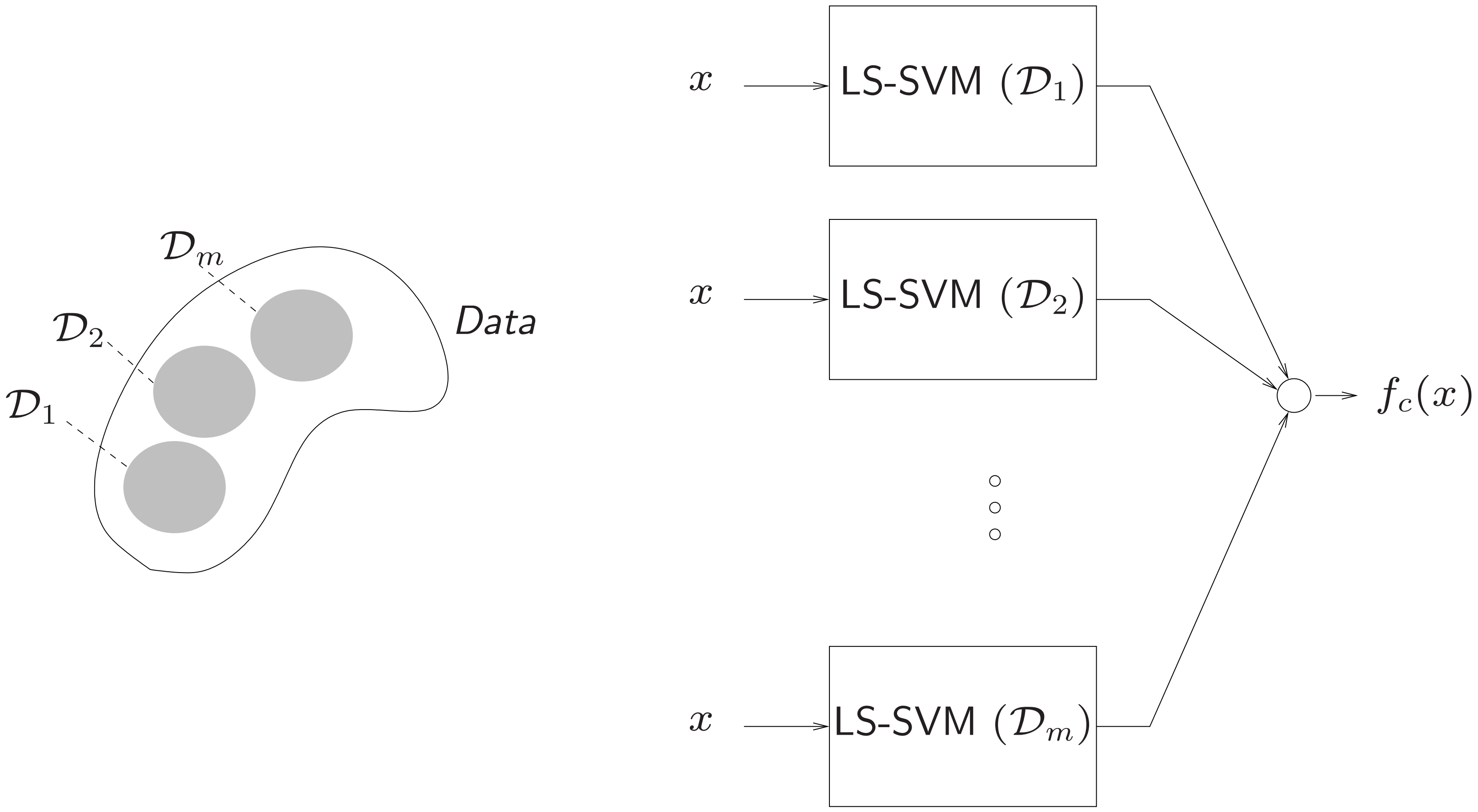
$$z(x)^T z(y) \simeq K(x - y).$$

- Compute the Fourier transform p of the kernel K:

$$p(\omega) = \frac{1}{2\pi} \int \exp(-j\omega^T \Delta) K(\Delta) d\Delta$$

Draw D iid samples $\omega_1, ..., \omega_D \in \mathbb{R}^d$ from p.
Obtain $z(x) = \sqrt{\frac{1}{D}}[\cos(\omega_1^T x) ... \cos(\omega_D^T x) \sin(\omega_1^T x) ... \sin(\omega_D^T x)]^T$.

Committee network of LS-SVMs (1)



Committee network of LS-SVMs (2)

- The committee network that consists of the m submodels takes the form

$$\begin{aligned} f_c(x) &= \sum_{i=1}^{m} \beta_i f_i(x) \\ &= h(x) + \sum_{i=1}^{m} \beta_i \epsilon_i(x) \end{aligned}$$

What betas to use?

where $\sum_{i=1}^{m} \beta_i = 1$, $h(x)$ is the true function to be estimated and $\epsilon_i(x) = f_i(x) - h(x)$ where

$$f_i(x) = \sum_{k=1}^{N_i} \alpha_k^{(i)} K^{(i)}(x, x_k) + b^{(i)}$$

is the i-th LS-SVM model trained on the data $\{x_k, y_k\}_{k=1}^{N_i}$ with resulting support values $\alpha_k^{(i)}$, bias term $b^{(i)}$ and kernel $K^{(i)}(\cdot, \cdot)$ for the i-th submodel and $i = 1, ..., m$ with m the number of LS-SVM submodels.

Committee network of LS-SVMs (3)

- One considers the covariance matrix

$$C_{ij} = \mathcal{E}[\epsilon_i(x)\epsilon_j(x)]$$

where in practice one works with a finite-sample approximation

$$C_{ij} = \frac{1}{N}\sum_{k=1}^{N}[f_i(x_k) - y_k][f_j(x_k) - y_k]$$

and the N data are a representative subset of the overall training data set (or the whole training data set itself).

Committee network of LS-SVMs (4)

- The committee error equals

$$\begin{aligned} J_c &= \mathcal{E}[\{f_c(x) - h(x)\}^2] = \mathcal{E}[\,(\sum_{i=1}^{m} \beta_i \epsilon_i)\,(\sum_{j=1}^{m} \beta_j \epsilon_j)\,] \\ &\simeq \sum_{i=1}^{m} \sum_{j=1}^{m} \beta_i \beta_j C_{ij} = \beta^T C \beta. \end{aligned}$$

An optimal choice of β follows then from

C has to be full rank (so that it can be inverted), if it weren't it could be that two models are exactly the same.

$$\min_{\beta} \frac{1}{2} \beta^T C \beta \quad \text{such that} \quad \sum_{i=1}^{m} \beta_i = 1.$$

The sum of the betas should be equal to 1 to avoid the trivial solution where betas are zero.

with optimal solution

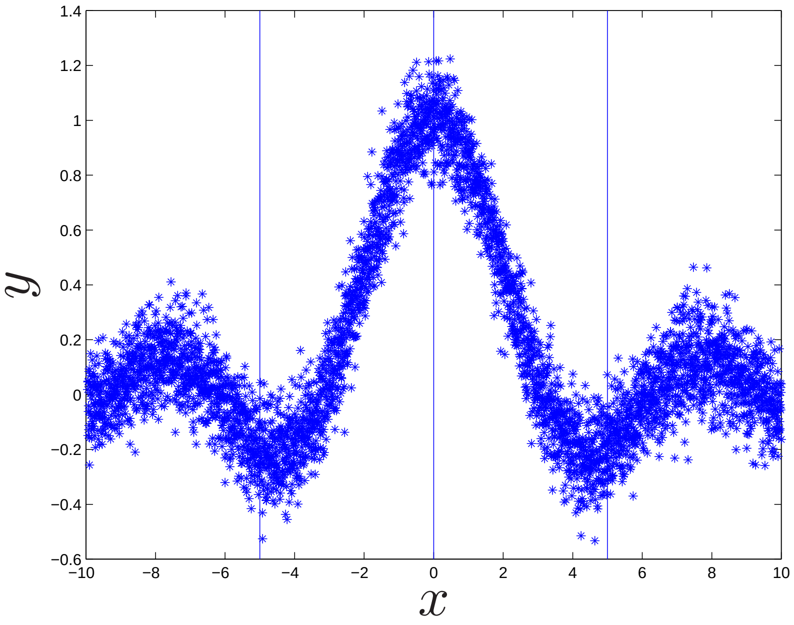
$$\beta = \frac{C^{-1} 1_v}{1_v^T C^{-1} 1_v}$$

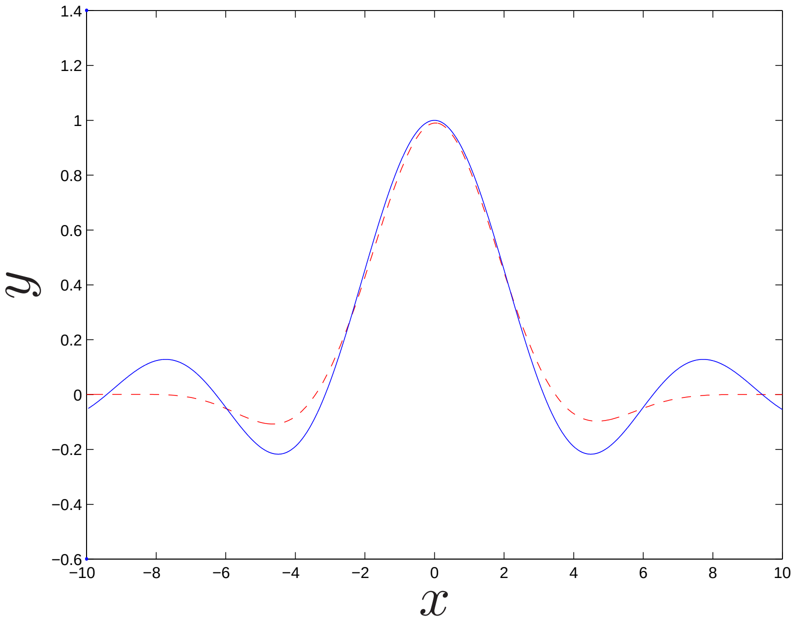
with $1_v = [1; 1; ...; 1]$.

You get another optimization problem solved through Lagrange multipliers.

Example (1)

sinc function with 4000 training data

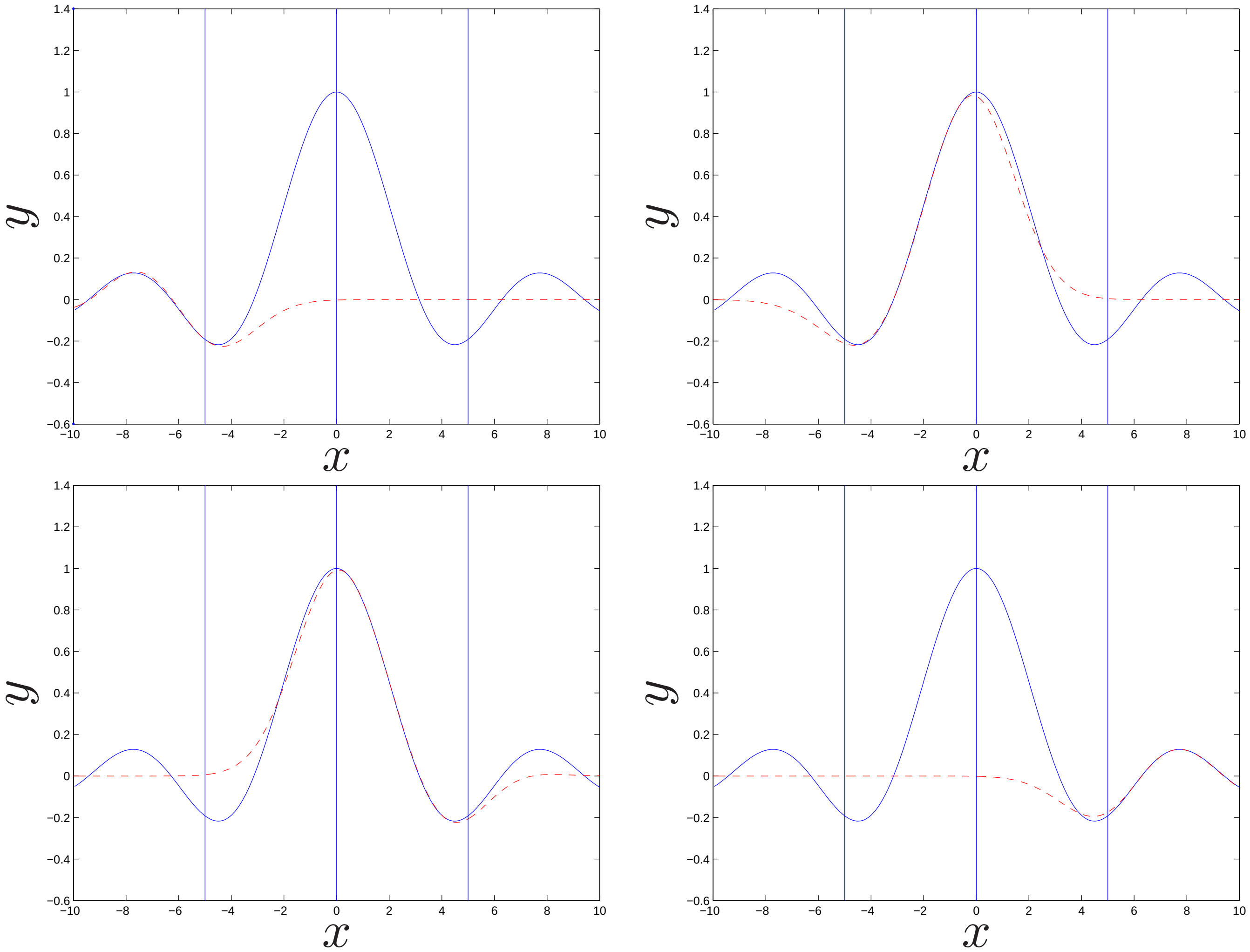




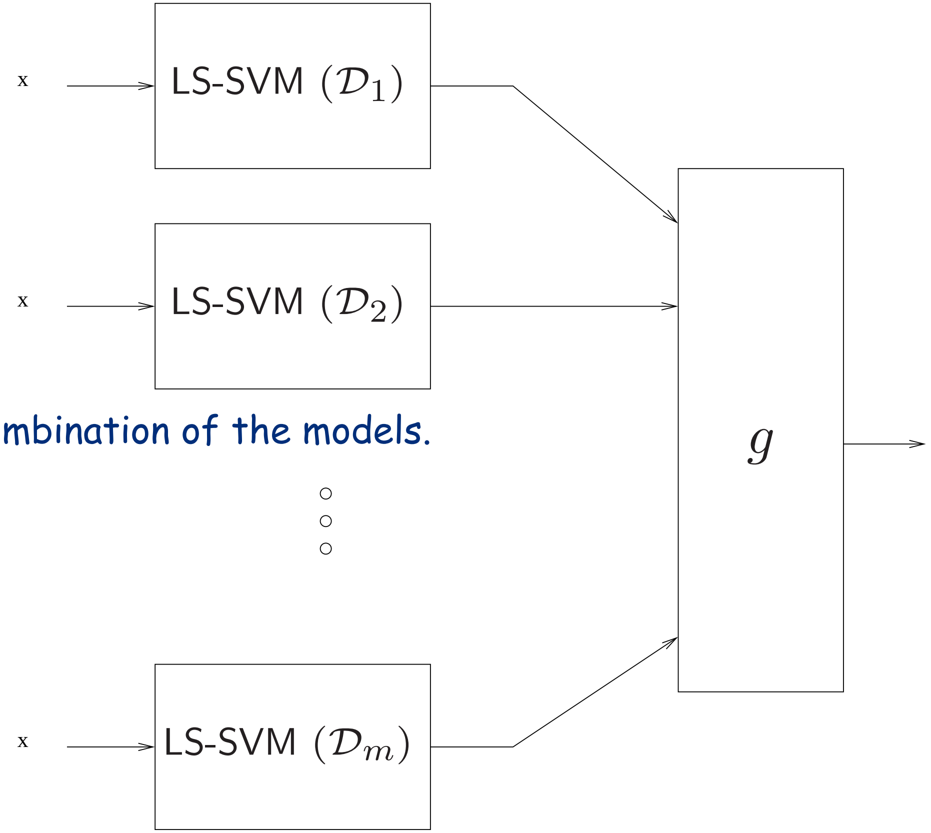
We divide the input into 4 regions and use a model for each region, then we find a combination.

Example (2)

Results of the individual LS-SVM models



Nonlinear combination of LS-SVMs (1)



You can also take a non-linear combination of the models.

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

Nonlinear combination of LS-SVMs (2)

- When taking an MLP in the second layer, the model is described by

$$g(z) = w^T \tanh(Vz + d)$$

  with

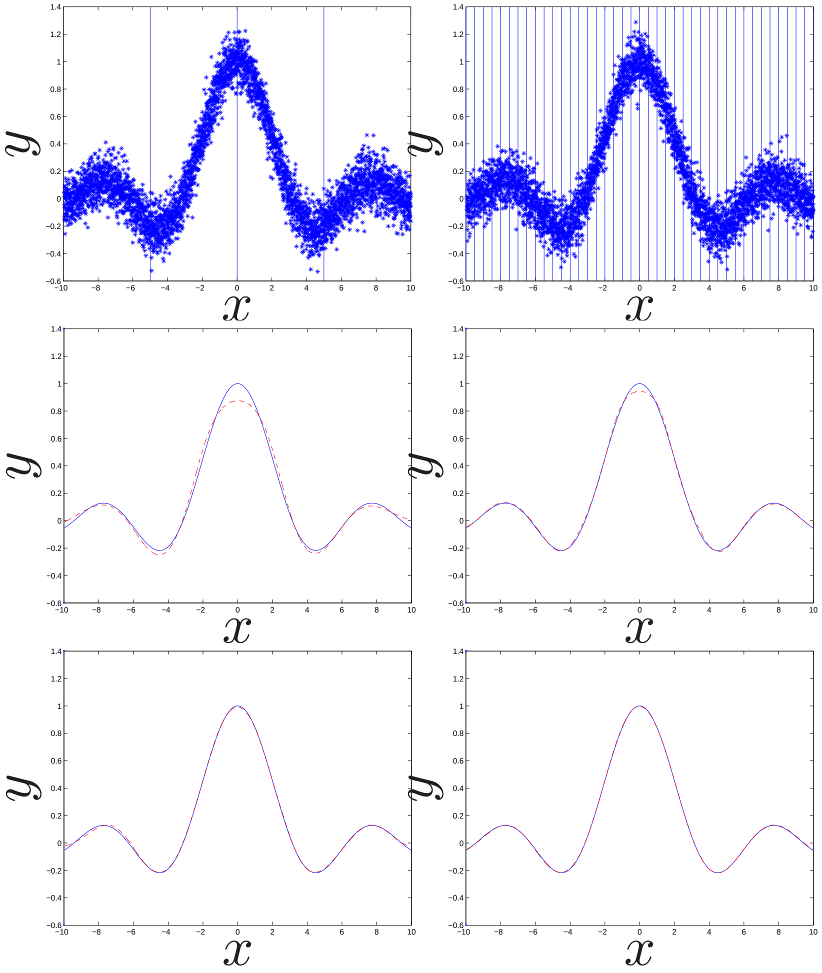
$$z_i(x) = \sum_{k=1}^{N_i} \alpha_k^{(i)} K^{(i)}(x, x_k) + b^{(i)} \ , \ i = 1, ..., m$$

  where m denotes the number of individual LS-SVM models whose outputs z_i are the input to a MLP with output weight vector $w \in \mathbb{R}^{n_h}$, hidden layer matrix $V \in \mathbb{R}^{n_h \times m}$ and bias vector $d \in \mathbb{R}^{n_h}$ where n_h denotes the number of hidden units (alternative: linear output layer $g(z) = w^T z + d$).

- The coefficients $\alpha_k^{(i)}, b^{(i)}$ for $i = 1, ..., m$ are the solutions to a number of m linear systems for each of the individual LS-SVMs trained on data sets $\mathcal{D}_i$.

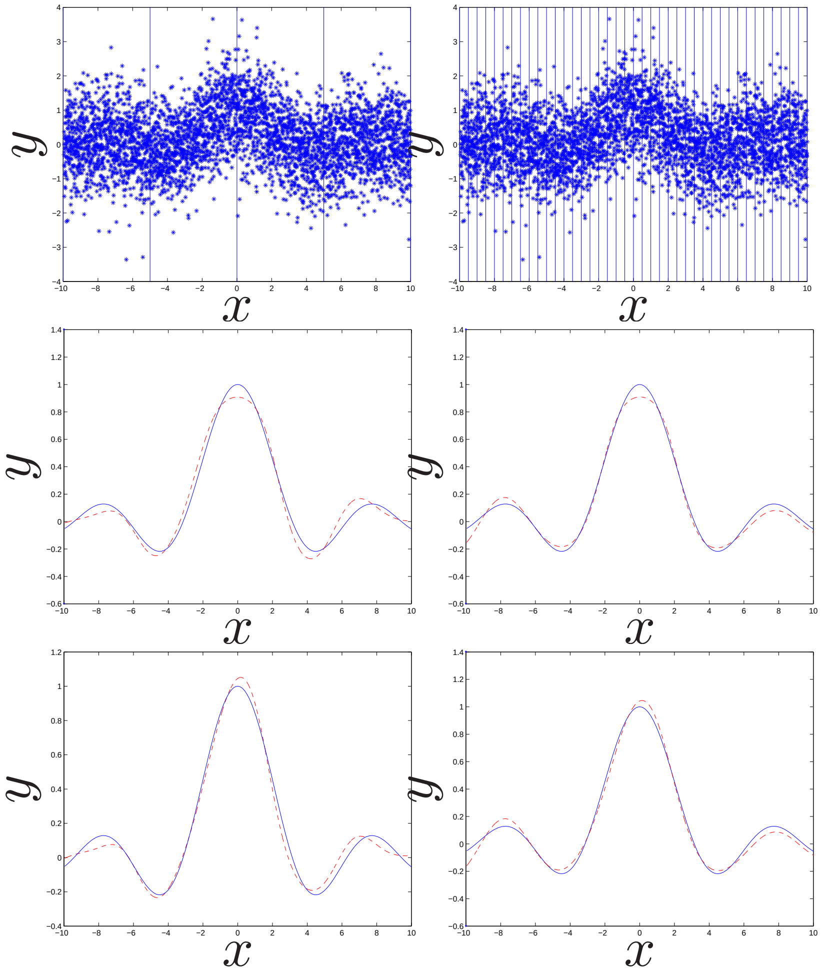
Example (1)

Linear versus nonlinear combinations of trained LS-SVM submodels



Example (2)

Linear versus nonlinear combinations under heavy noise


y
x
−10
−8
−6
−4
−2
0
2
4
6
8
10
−4
−3
−2
−1
0
1
2
3
4
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4