Additional topics

Johan Suykens

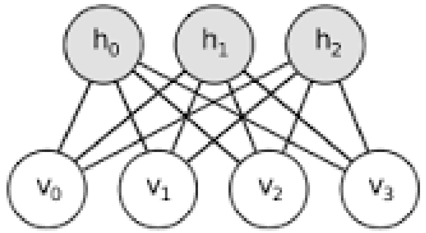
KU Leuven, ESAT-STADIUS
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium
Email: johan.suykens@esat.kuleuven.be
http://www.esat.kuleuven.be/stadius

Lecture 10

Contents - additional topics

Restricted kernel machines

Restricted Boltzmann Machines (RBM)



- Markov random field, bipartite graph, stochastic binary units
  Layer of visible units v and layer of hidden units h
  No hidden-to-hidden connections
- Energy:

Two correction term (last two), you want to maximise the energy (first term)
Two ways to read it : correlation of vtW with h OR vt with Wh
You can relate the energy function to a probability distribution

$$E(v,h;\theta) = -v^T W h - c^T v - a^T h \quad \text{with} \quad \theta = \{W, c, a\}$$

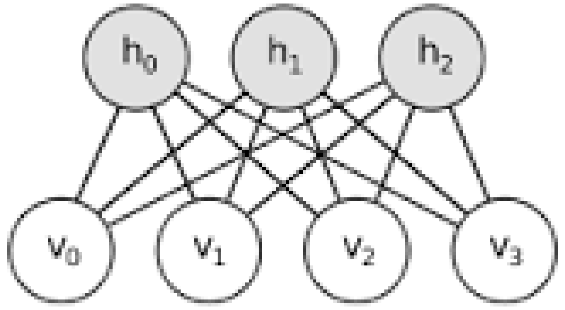
Joint distribution:

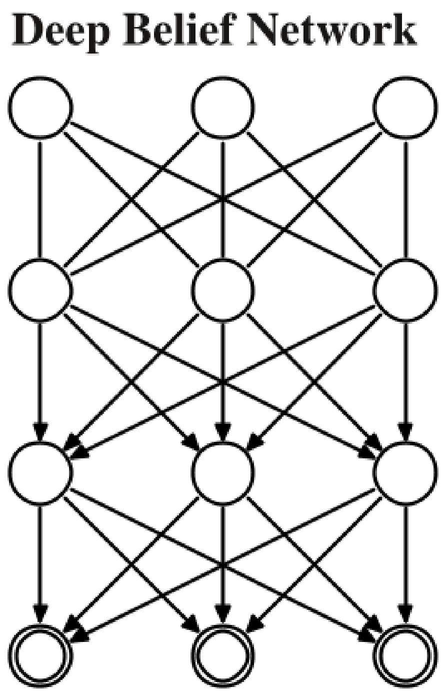
$$P(v,h;\theta) = \frac{1}{Z(\theta)} \exp(-E(v,h;\theta))$$

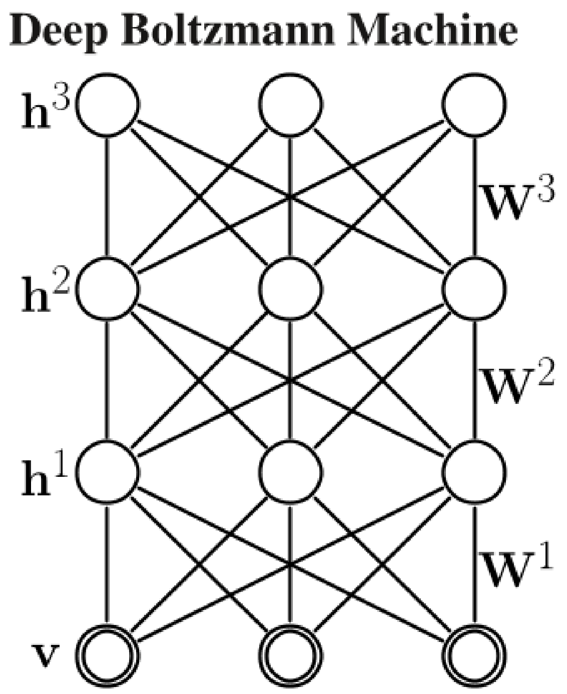
with partition function $Z(\theta) = \sum_v \sum_h \exp(-E(v,h;\theta))$

[Hinton, Osindero, Teh, Neural Computation 2006]

RBM and deep learning







$p(v, h)$ You can generate new data with these models. $p(v, h^1, h^2, h^3, ...)$

[Hinton et al., 2006; Salakhutdinov, 2015]

Energy function

- RBM:

$$E = -v^T W h$$

- Deep Boltzmann machine (two layers):

$$E = -v^T W^1 h^1 - {h^1}^T W^2 h^2$$

- Deep Boltzmann machine (three layers):

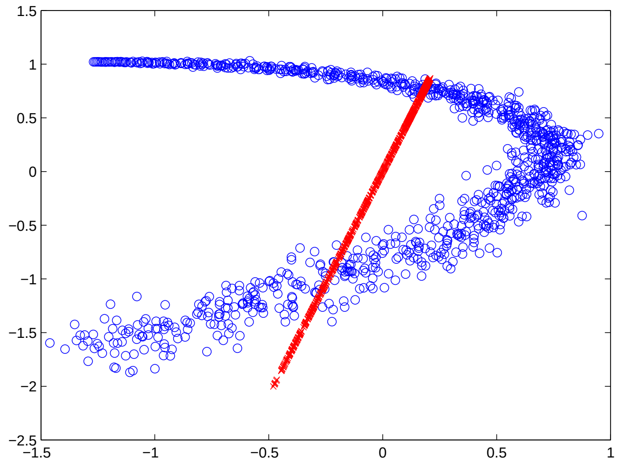
$$E = -v^T W^1 h^1 - {h^1}^T W^2 h^2 - {h^2}^T W^3 h^3$$

Restricted Kernel Machines (RKM)

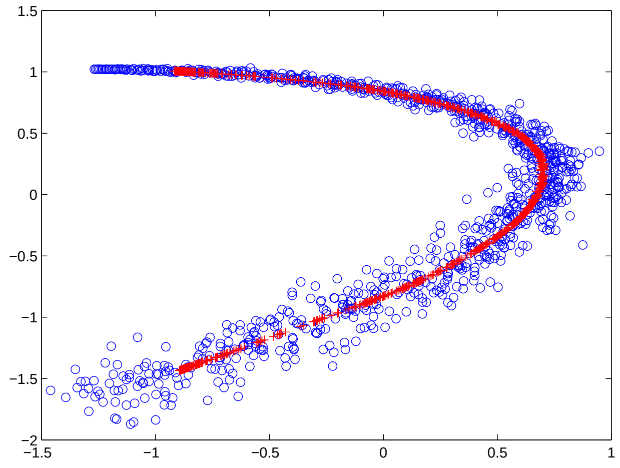
- Kernel machine interpretations in terms of **visible and hidden units** (similar to Restricted Boltzmann Machines (**RBM**))
- Restricted Kernel Machine (**RKM**) representations for
  - LS-SVM regression/classification
  - Kernel PCA
  - Matrix SVD
  - Parzen-type models
  - other
- Based on principle of **conjugate feature duality** (with hidden features corresponding to dual variables)
- Deep Restricted Kernel Machines (Deep RKM)

[Suykens, Neural Computation, 2017]

Kernel principal component analysis (KPCA)



linear PCA



kernel PCA (RBF kernel)

Kernel PCA [Schölkopf et al., 1998]:
take eigenvalue decomposition of the kernel matrix

$$\begin{bmatrix} K(x_1,x_1) & \dots & K(x_1,x_N) \\ \vdots & & \vdots \\ K(x_N,x_1) & \dots & K(x_N,x_N) \end{bmatrix}$$

(applications in dimensionality reduction and denoising)

Kernel PCA: classical LS-SVM approach

- Primal problem: [Suykens et al., 2002]: **model-based** approach

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

A bias term automatically centers the data (see previous lecture)

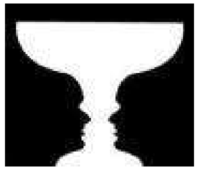
- Dual problem corresponds to kernel PCA

$$\Omega^{(c)} \alpha = \lambda \alpha \;\; \text{with} \;\; \lambda = 1/\gamma$$

with $\Omega^{(c)}_{ij} = (\varphi(x_i) - \hat{\mu}_\varphi)^T (\varphi(x_j) - \hat{\mu}_\varphi)$ the *centered kernel matrix* and $\hat{\mu}_\varphi = (1/N) \sum_{i=1}^{N} \varphi(x_i)$.

- Interpretation:
  1. pool of candidate components (objective function equals zero)
  2. select relevant components

From KPCA to RKM representation (1)



Model:

$$\boxed{e = W^T \varphi(x)}$$

objective J
$=$ regularization term $\mathrm{Tr}(W^T W)$
- $(\frac{1}{\lambda})$ variance term $\sum_i e_i^T e_i$

It's similar to the transition from primal to dual but instead of Lagrange multipliers you use the inequality here.

$\downarrow$ use property $e^T h \leq \frac{1}{2\lambda} e^T e + \frac{\lambda}{2} h^T h$

Quadratic form in vectors e and h
The property holds for any e and h

RKM representation:

$$\boxed{e = \sum_j h_j K(x_j, x)}$$

obtain $J \leq \overline{J}(h_i, W)$
solution from stationary points of $\overline{J}$:
$\frac{\partial \overline{J}}{\partial h_i} = 0$, $\frac{\partial \overline{J}}{\partial W} = 0$

From KPCA to RKM representation (2)

- Objective

$$\begin{aligned} J &= \frac{\eta}{2}\mathrm{Tr}(W^TW) - \frac{1}{2\lambda}\sum_{i=1}^{N} e_i^T e_i \text{ s.t. } e_i = W^T\varphi(x_i), \forall i \\ &\leq -\sum_{i=1}^{N} e_i^T h_i + \frac{\lambda}{2}\sum_{i=1}^{N} h_i^T h_i + \frac{\eta}{2}\mathrm{Tr}(W^TW) \text{ s.t. } e_i = W^T\varphi(x_i), \forall i \\ &= -\sum_{i=1}^{N} \varphi(x_i)^T W h_i + \frac{\lambda}{2}\sum_{i=1}^{N} h_i^T h_i + \frac{\eta}{2}\mathrm{Tr}(W^TW) \triangleq \overline{J} \end{aligned}$$

Similar to RBM but instead of probability distribution assumptions you don't make such assumptions.

- Stationary points of $\overline{J}(h_i, W)$:

$$\begin{cases} \dfrac{\partial \overline{J}}{\partial h_i} = 0 \quad \Rightarrow \quad W^T\varphi(x_i) = \lambda h_i, \ \forall i \\ \dfrac{\partial \overline{J}}{\partial W} = 0 \quad \Rightarrow \quad W = \dfrac{1}{\eta}\sum_i \varphi(x_i) h_i^T \end{cases}$$

From KPCA to RKM representation (3)

- Elimination of W gives the eigenvalue decomposition:

$$\frac{1}{\eta} K H^T = H^T \Lambda$$

where $H = [h_1 ... h_N] \in \mathbb{R}^{s \times N}$ and $\Lambda = \mathrm{diag}\{\lambda_1, ..., \lambda_s\}$ with $s \leq N$

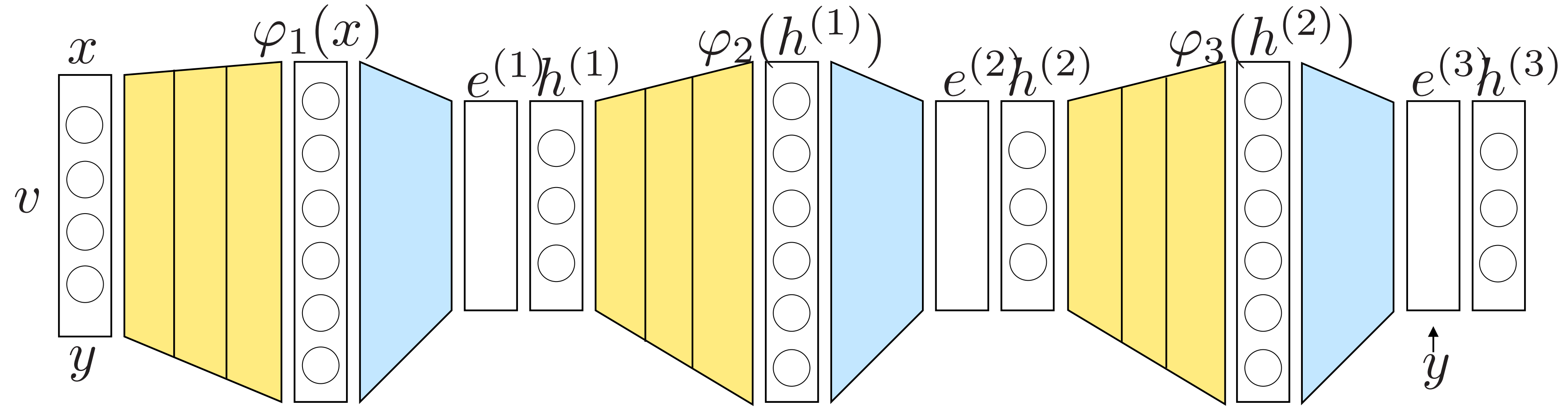
- Primal and dual model representations

$$\begin{tikzcd} & (P)_{\mathrm{RKM}}: \quad \hat{e} = W^T \varphi(x) \\ \mathcal{M} \arrow[ur] \arrow[dr] & \\ & (D)_{\mathrm{RKM}}: \quad \hat{e} = \frac{1}{\eta} \sum_j h_j K(x_j, x). \end{tikzcd}$$

The new duality is called the conjugate feature duality.
The hidden units (features you extract) relate to the support vectors

Deep Restricted Kernel Machines

Deep RKM: example



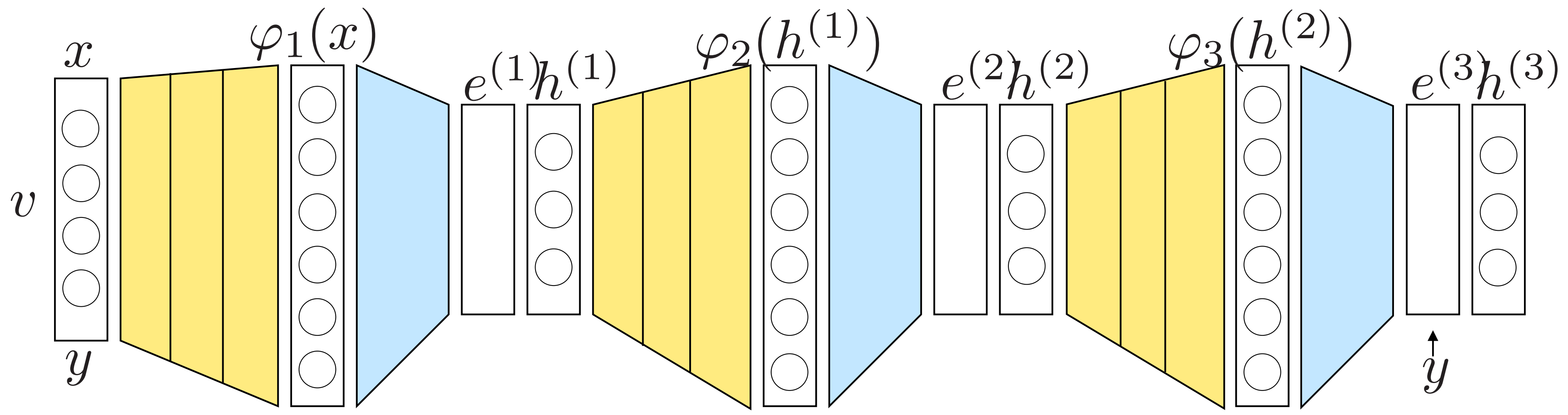
Deep RKM: KPCA + KPCA + LSSVM [Suykens, 2017]

Coupling of RKMs by taking sum of the objectives

$$J_{\text{deep}} = \overline{J}_1 + \overline{J}_2 + \underline{J}_3$$

Multiple *levels* and multiple *layers* per level.

in more detail ...



The frist terms (in red) are directly related to the energy function.

$$\begin{aligned} J_{\text{deep}} \;=\; & -\sum_{i=1}^{N} \varphi_1(x_i)^T W_1 h_i^{(1)} + \frac{\lambda_1}{2}\sum_{i=1}^{N} h_i^{(1)^T} h_i^{(1)} + \frac{\eta_1}{2}\mathrm{Tr}(W_1^T W_1) \\ & -\sum_{i=1}^{N} \varphi_2(h_i^{(1)})^T W_2 h_i^{(2)} + \frac{\lambda_2}{2}\sum_{i=1}^{N} h_i^{(2)^T} h_i^{(2)} + \frac{\eta_2}{2}\mathrm{Tr}(W_2^T W_2) \\ & +\sum_{i=1}^{N} (y_i^T - \varphi_3(h_i^{(2)})^T W_3 - b^T) h_i^{(3)} - \frac{\lambda_3}{2}\sum_{i=1}^{N} h_i^{(3)^T} h_i^{(3)} + \frac{\eta_3}{2}\mathrm{Tr}(W_3^T W_3) \end{aligned}$$

Primal and dual model representations

$$\begin{tikzcd} & (P)_{\mathrm{DeepRKM}} : \\ \mathcal{M} \arrow[ur] \arrow[dr] & \\ & (D)_{\mathrm{DeepRKM}} : \end{tikzcd}$$

$$(P)_{\mathrm{DeepRKM}} : \quad \begin{aligned} \hat{e}^{(1)} &= W_1^T \varphi_1(x) \\ \hat{e}^{(2)} &= W_2^T \varphi_2(\Lambda_1^{-1} \hat{e}^{(1)}) \\ \hat{y} &= W_3^T \varphi_3(\Lambda_2^{-1} \hat{e}^{(2)}) + b \end{aligned}$$

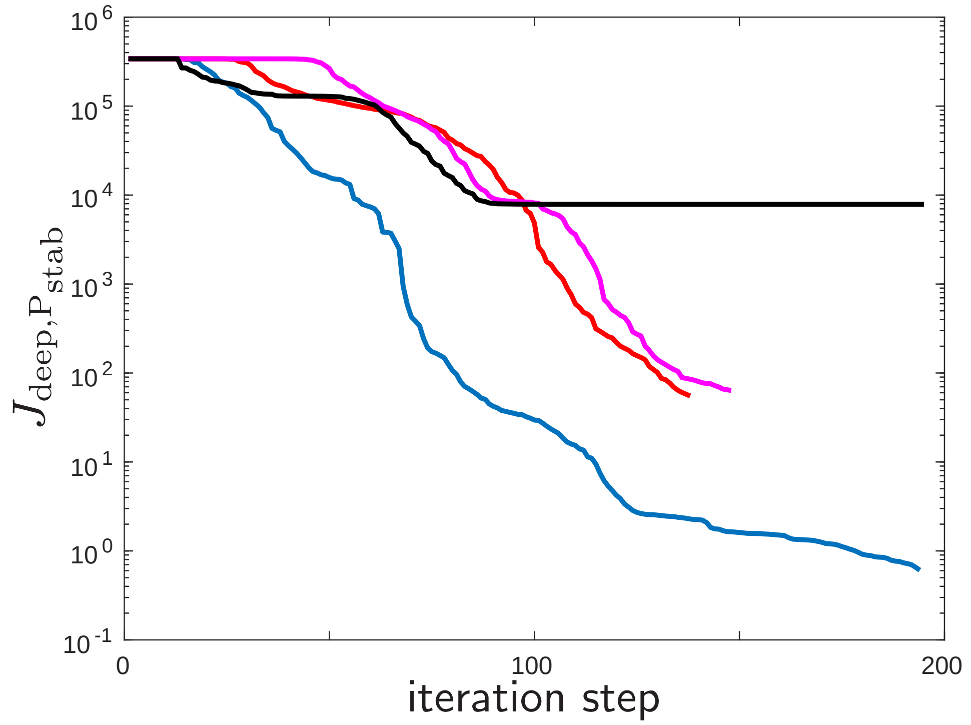
$$(D)_{\mathrm{DeepRKM}} : \quad \begin{aligned} \hat{e}^{(1)} &= \tfrac{1}{\eta_1} \textstyle\sum_j h_j^{(1)} K_1(x_j, x) \\ \hat{e}^{(2)} &= \tfrac{1}{\eta_2} \textstyle\sum_j h_j^{(2)} K_2(h_j^{(1)}, \Lambda_1^{-1} \hat{e}^{(1)}) \\ \hat{y} &= \tfrac{1}{\eta_3} \textstyle\sum_j h_j^{(3)} K_3(h_j^{(2)}, \Lambda_2^{-1} \hat{e}^{(2)}) + b \end{aligned}$$

You can chose kernel functions that are invariant with respect to eg. rotation (RBF kernels), translation, ...

The framework can be used for training deep feedforward neural networks and deep kernel machines [Suykens, 2017].

(Other approaches: e.g. kernels for deep learning [Cho & Saul, 2009], mathematics of the neural response [Smale et al., 2010], deep gaussian processes [Damianou & Lawrence, 2013], convolutional kernel networks [Mairal et al., 2014], multi-layer support vector machines [Wiering & Schomaker, 2014])

Training process



Objective function (logarithmic scale) during training on the ion data set:

- black color: level 3 objective only
- J_{deep} for $c_{\text{stab}} = 1$, 10, 100 (blue, red, magenta color) in stabilization term

Kernel spectral clustering

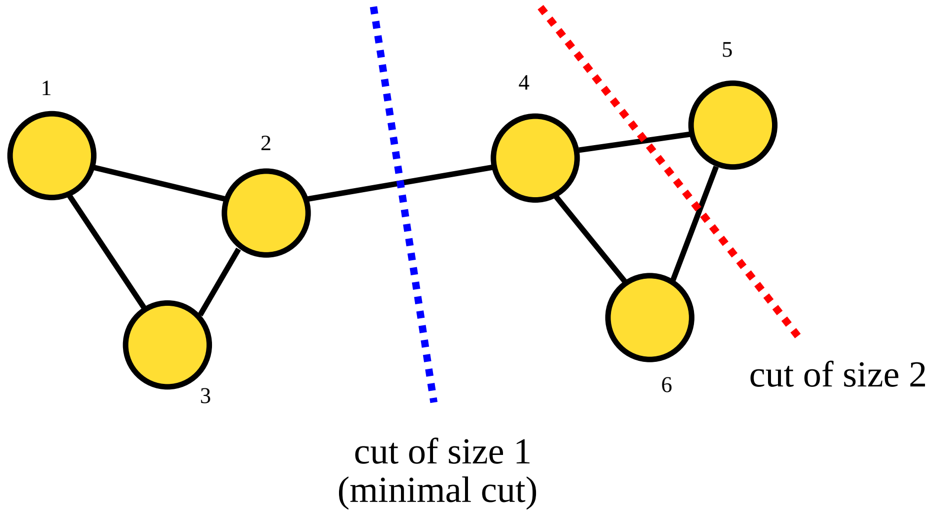
Spectral graph clustering

Minimal cut: given the graph $\mathcal{G} = (V, E)$, find clusters $\mathcal{A}_1, \mathcal{A}_2$

$$\min_{q_i \in \{-1,+1\}} \frac{1}{2} \sum_{i,j} w_{ij} (q_i - q_j)^2$$

with cluster membership indicator q_i ($q_i = 1$ if $i \in \mathcal{A}_1$, $q_i = -1$ if $i \in \mathcal{A}_2$) and $W = [w_{ij}]$ the weighted adjacency matrix.

This is a combinatorial optimization problem.
(indicator is 1 or -1)



Spectral graph clustering

- Relaxation to **Min-cut** spectral clustering problem

$$\min_{\tilde{q}^T\tilde{q}=1} \; \tilde{q}^T L \tilde{q}$$

with $L = D - W$ the unnormalized graph Laplacian, degree matrix $D = \mathrm{diag}(d_1, ..., d_N)$, $d_i = \sum_j w_{ij}$, giving

$$L\tilde{q} = \lambda \tilde{q}.$$

Cluster member indicators: $\hat{q}_i = \mathrm{sign}(\tilde{q}_i - \theta)$ with threshold θ.

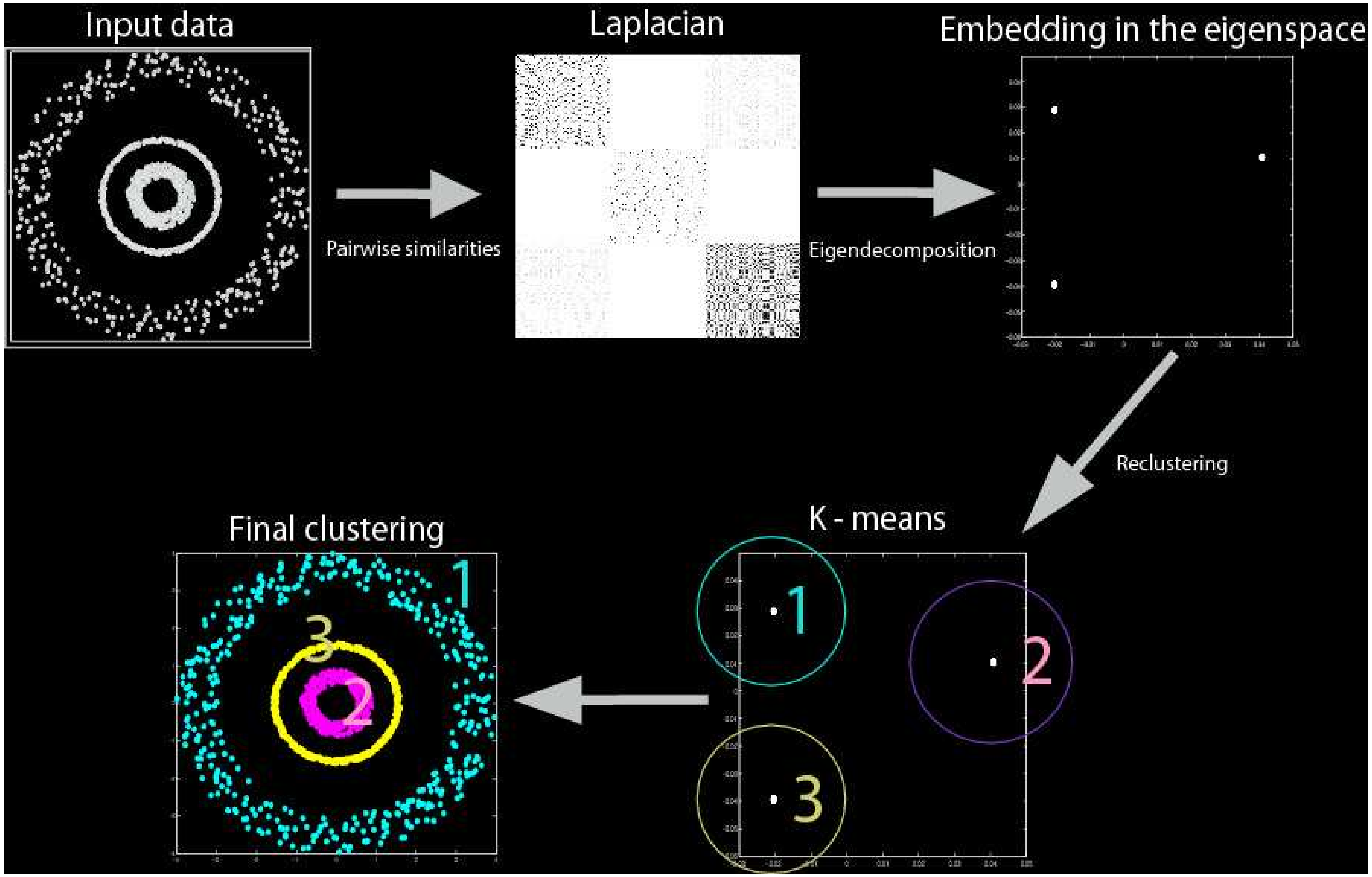
- **Normalized cut**

$$L\tilde{q} = \lambda D \tilde{q}$$

[Fiedler, 1973; Shi & Malik, 2000; Ng et al. 2002; Chung, 1997; von Luxburg, 2007]

- **Discrete version** to continuous problem (Laplace operator)
[Belkin & Niyogi, 2003; von Luxburg et al., 2008; Smale & Zhou, 2007]

Spectral clustering + K-means



Kernel spectral clustering: case of two clusters

- **Underlying model** (primal representation):

$$\hat{e}_* = w^T \varphi(x_*) + b$$

with $\hat{q}_* = \mathrm{sign}[\hat{e}_*]$ the estimated cluster indicator at any $x_* \in \mathbb{R}^d$.

Weighted version of kernel PCA if you want

- **Primal problem:** training on given data $\{x_i\}_{i=1}^N$

$$\begin{array}{ll} \min\limits_{w,b,e} & -\frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^N v_i \, e_i^2 \\ \text{subject to} & e_i = w^T \varphi(x_i) + b, \quad i = 1, ..., N \end{array}$$

with **positive weights** v_i (will be related to inverse degree matrix).

[Alzate & Suykens, IEEE-PAMI, 2010]

Lagrangian and conditions for optimality

- Lagrangian:

$$\mathcal{L}(w,b,e;\alpha) = -\frac{1}{2}w^T w + \gamma\,\frac{1}{2}\sum_{i=1}^{N} v_i e_i^2 - \sum_{i=1}^{N} \alpha_i (e_i - w^T \varphi(x_i) - b)$$

- Conditions for optimality:

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w} = 0 & \Rightarrow \quad w = \sum_i \alpha_i \varphi(x_i) \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 & \Rightarrow \quad \sum_i \alpha_i = 0 \\ \dfrac{\partial \mathcal{L}}{\partial e_i} = 0 & \Rightarrow \quad \alpha_i = \gamma v_i e_i,\ \ i = 1, ..., N \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_i} = 0 & \Rightarrow \quad e_i = w^T \varphi(x_i) + b,\ \ i = 1, ..., N \end{cases}$$

- Eliminate w, b, e, write solution in α.

Kernel-based model representation

- **Dual problem:**

$$V M_V \Omega \alpha = \lambda \alpha$$

with

$\lambda = 1/\gamma$
$M_V = I_N - \frac{1}{1_N^T V 1_N} 1_N 1_N^T V$: weighted centering matrix
$\Omega = [\Omega_{ij}]$: kernel matrix with $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$

- **Dual model representation:**

$$\hat{e}_* = \sum_{i=1}^{N} \alpha_i K(x_i, x_*) + b$$

with $K(x_i, x_*) = \varphi(x_i)^T \varphi(x_*)$.

Choice of weights v_i

- Take $V = D^{-1}$ where $D = \text{diag}\{d_1, ..., d_N\}$ and $d_i = \sum_{j=1}^N \Omega_{ij}$
- This gives the **generalized eigenvalue problem**:

$$M_D \Omega \alpha = \lambda D \alpha$$

  with $M_D = I_N - \frac{1}{1_N^T D^{-1} 1_N} 1_N 1_N^T D^{-1}$

  This is a modified version of random walks spectral clustering.

- Note that $\text{sign}[e_i] = \text{sign}[\alpha_i]$ if $\gamma v_i > 0$ (on training data)

  ... but $\text{sign}[e_*]$ applies beyond training data

Kernel spectral clustering: more clusters

- Case of k clusters: additional sets of constraints

$$\begin{array}{ll} \min\limits_{w^{(l)},e^{(l)},b_l} & -\frac{1}{2}\sum_{l=1}^{k-1} w^{(l)^T} w^{(l)} + \frac{1}{2}\sum_{l=1}^{k-1} \gamma_l e^{(l)^T} D^{-1} e^{(l)} \\ \text{subject to} & e^{(1)} = \Phi_{N\times n_h} w^{(1)} + b_1 1_N \\ & e^{(2)} = \Phi_{N\times n_h} w^{(2)} + b_2 1_N \\ & \vdots \\ & e^{(k-1)} = \Phi_{N\times n_h} w^{(k-1)} + b_{k-1} 1_N \end{array}$$

where $e^{(l)} = [e_1^{(l)}; \ldots; e_N^{(l)}]$ and $\Phi_{N\times n_h} = [\varphi(x_1)^T; \ldots; \varphi(x_N)^T] \in \mathbb{R}^{N\times n_h}$.

- **Dual problem**: $M_D \Omega \alpha^{(l)} = \lambda D \alpha^{(l)}$, $l = 1, \ldots, k-1$.

[Alzate & Suykens, IEEE-PAMI, 2010]

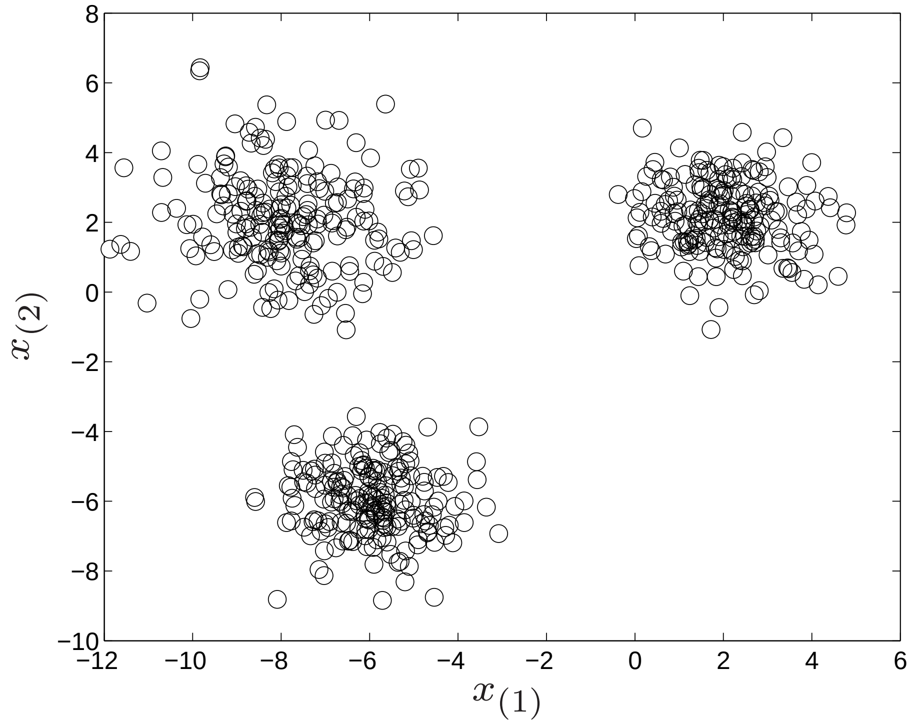
Primal and dual model representations

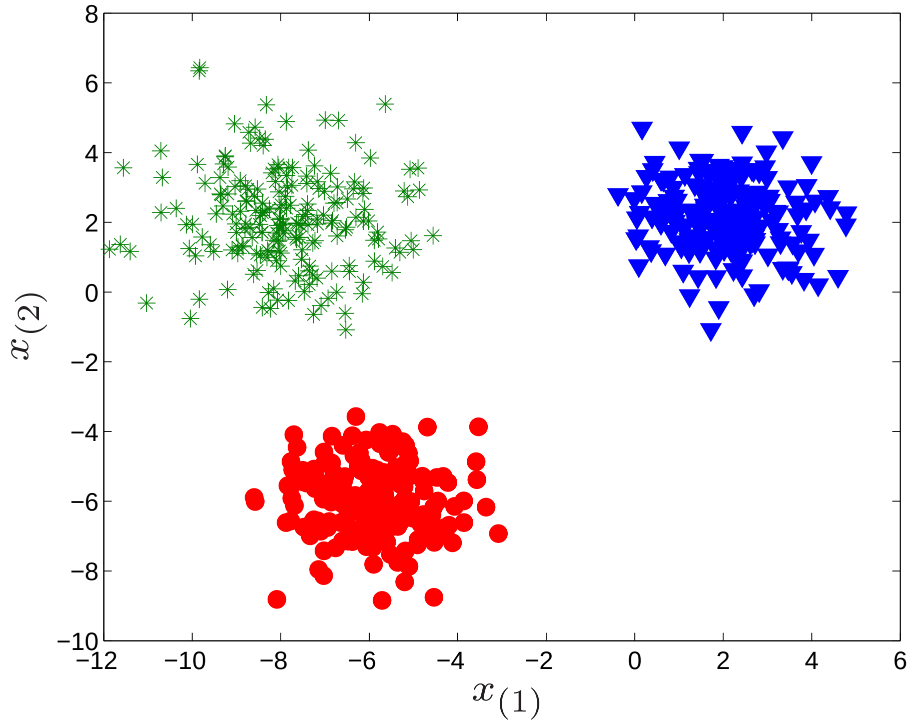
k clusters
$k-1$ sets of constraints (index $l = 1, ..., k-1$)

$$\begin{tikzcd}
 & (P): \quad \text{sign}[\hat{e}_*^{(l)}] = \text{sign}[{w^{(l)}}^T \varphi(x_*) + b_l] \\
\mathcal{M} \arrow[ru] \arrow[rd] & \\
 & (D): \quad \text{sign}[\hat{e}_*^{(l)}] = \text{sign}[\sum_j \alpha_j^{(l)} K(x_*, x_j) + b_l]
\end{tikzcd}$$

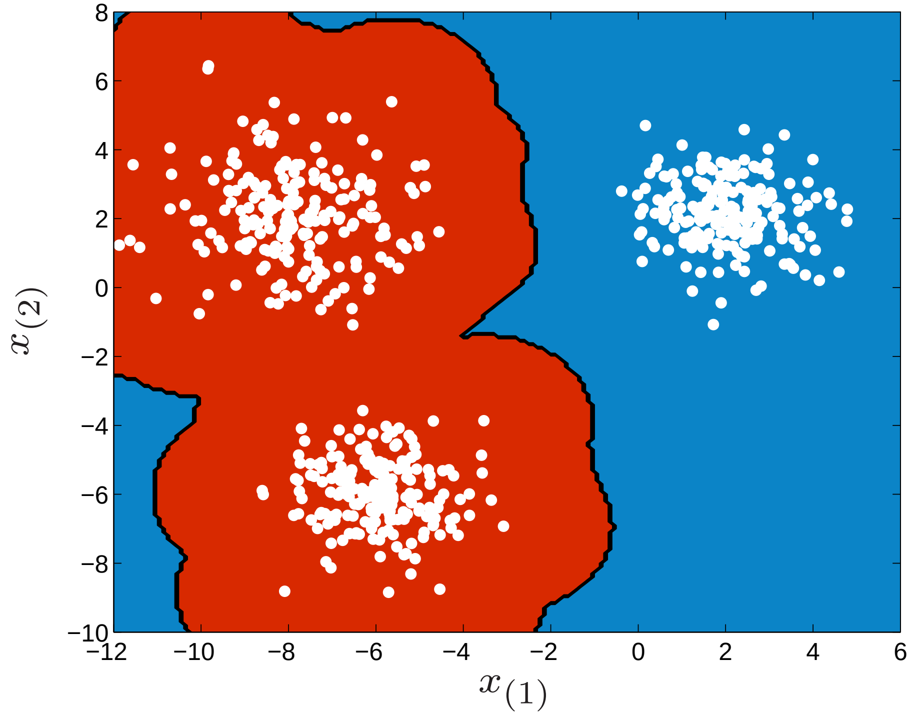
- *Note:* additional sets of constraints also in multi-class and vector-valued output LS-SVMs [Suykens et al., 1999]
- *Advantages:* out-of-sample extensions, model selection procedures, large scale methods

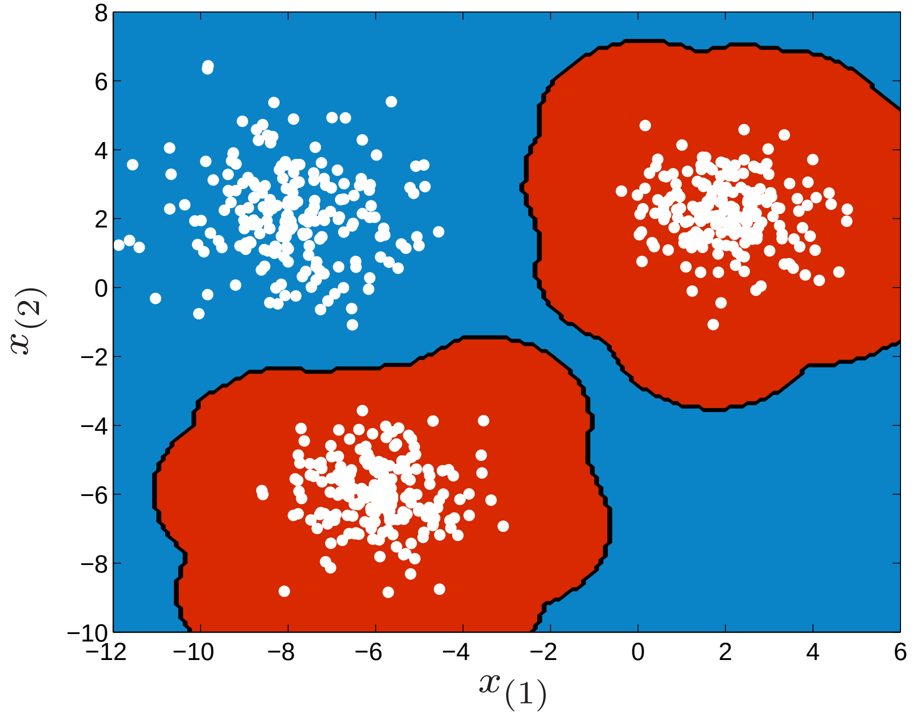
Out-of-sample extension and coding





Out-of-sample extension and coding





Similar to coding scheme (one versus all), you can do out of sample extension because you have models.

Piecewise constant eigenvectors and extension (1)

- **Definition.** [Meila & Shi, 2001] Vector α is called *piecewise constant* relative to a partition $(\mathcal{A}_1, ..., \mathcal{A}_k)$ iff $\alpha_i = \alpha_j \ \forall x_i, x_j \in \mathcal{A}_p, p = 1, ..., k$.

- **Proposition.** [Alzate & Suykens, 2010] Assume

  (i) a training set $\mathcal{D} = \{x_i\}_{i=1}^N$ and validation set $\mathcal{D}^v = \{x_m^v\}_{m=1}^{N_v}$ i.i.d. sampled from the same underlying distribution;
  (ii) a set of k clusters $\{\mathcal{A}_1, ..., \mathcal{A}_k\}$ with $k > 2$;
  (iii) an isotropic kernel function such that $K(x, z) = 0$ when x and z belong to different clusters;
  (iv) the eigenvectors $\alpha^{(l)}$ for $l = 1, ..., k-1$ are piecewise constant.

  Then validation set points belonging to the same cluster are *collinear* in the $k-1$ dimensional subspace spanned by the columns of $E^v \in \mathbb{R}^{N_v \times (k-1)}$ where $E_{ml}^v = e_m^{(l)} = \sum_{i=1}^N \alpha_i^{(l)} K(x_i, x_m^v) + b_l$.

Piecewise constant eigenvectors and extension (2)

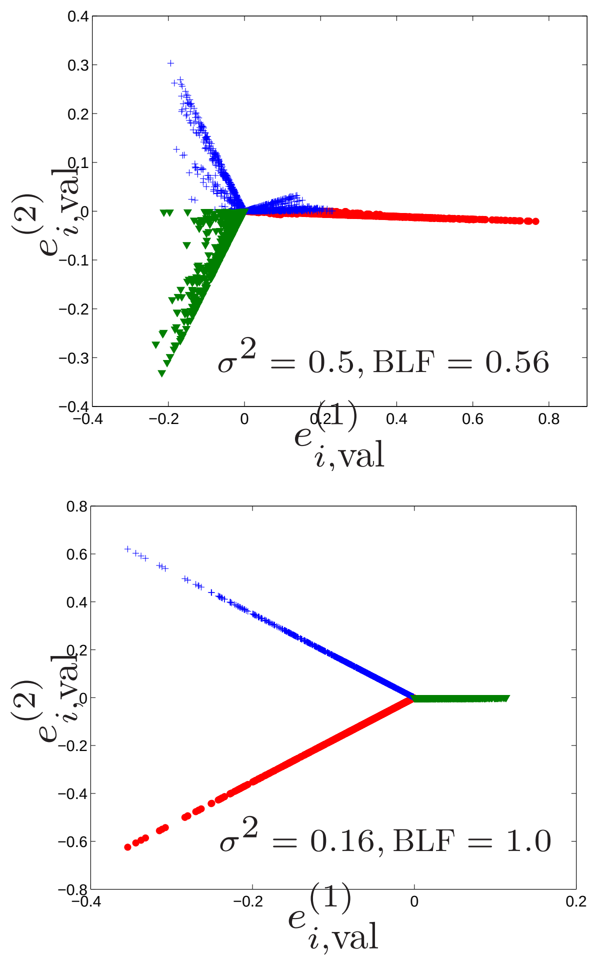
- **Key aspect of the proof:** for $x_* \in \mathcal{A}_p$ one has

$$\begin{aligned} e_*^{(l)} &= \sum_{i=1}^N \alpha_i^{(l)} K(x_i, x_*) + b^{(l)} \\ &= c_p^{(l)} \sum_{i \in \mathcal{A}_p} K(x_i, x_*) + \sum_{i \notin \mathcal{A}_p}^N \alpha_i^{(l)} K(x_i, x_*) + b^{(l)} \\ &= c_p^{(l)} \sum_{i \in \mathcal{A}_p} K(x_i, x_*) + b^{(l)} \end{aligned}$$

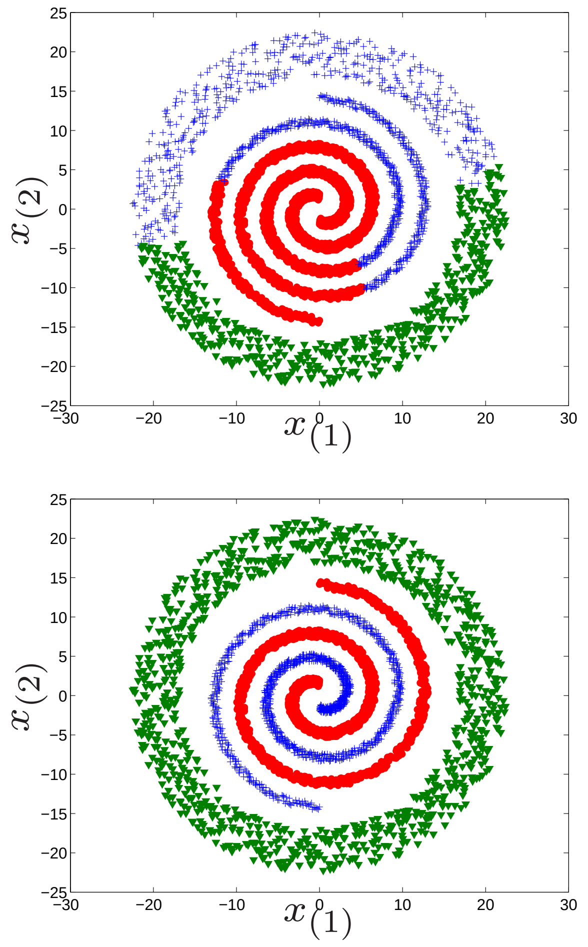
- **Model selection** to determine kernel parameters and k:
  Looking for line structures in the space $(e_i^{(1)}, e_i^{(2)}, ..., e_i^{(k-1)})$,
  evaluated on validation data (aiming for good generalization)

- **Choice kernel:**
  Gaussian RBF kernel
  χ^2-kernel for images

Model selection (looking for lines): toy problem 1

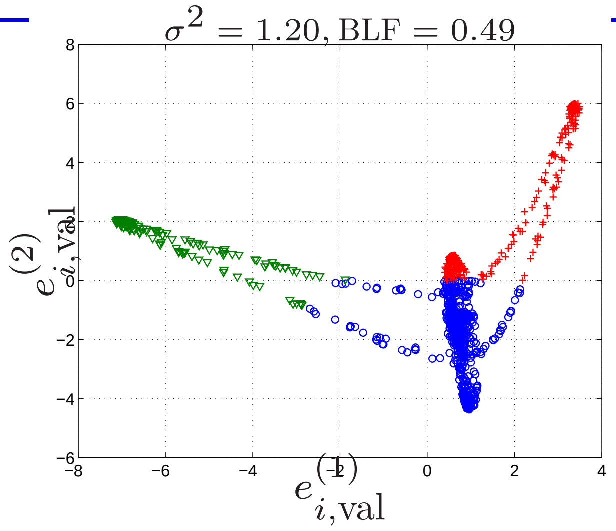


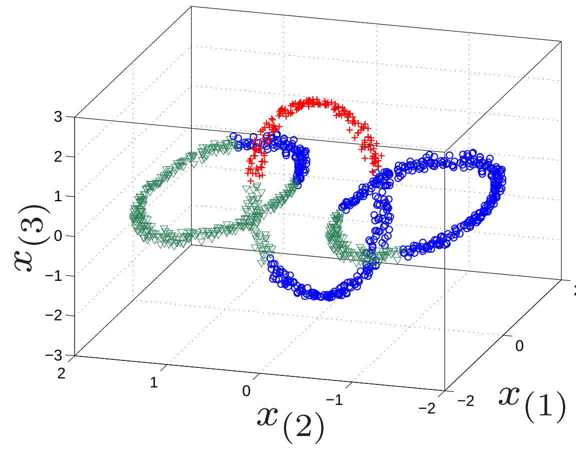
validation set

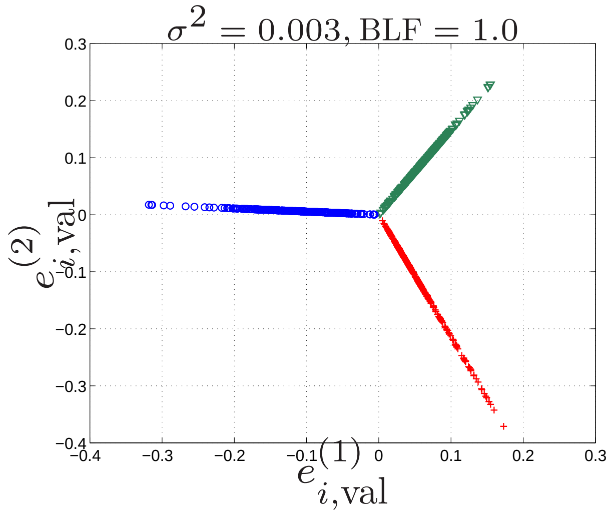


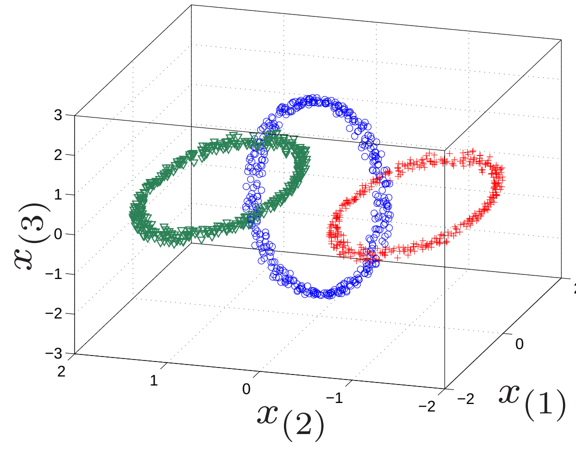
train + validation + test data

Model selection (looking for lines): toy problem 2





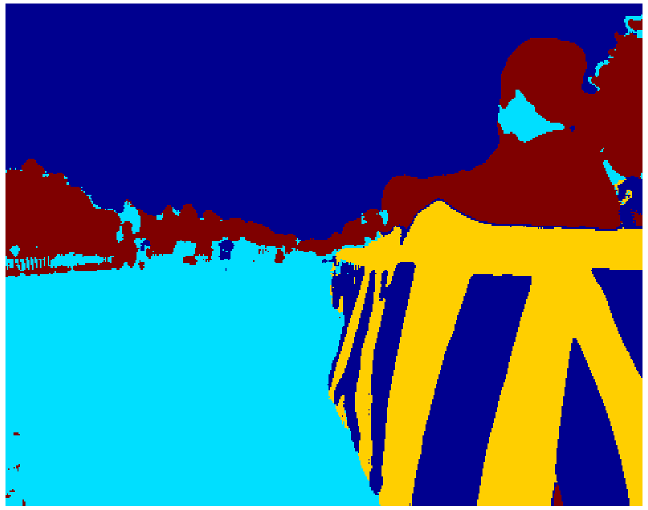


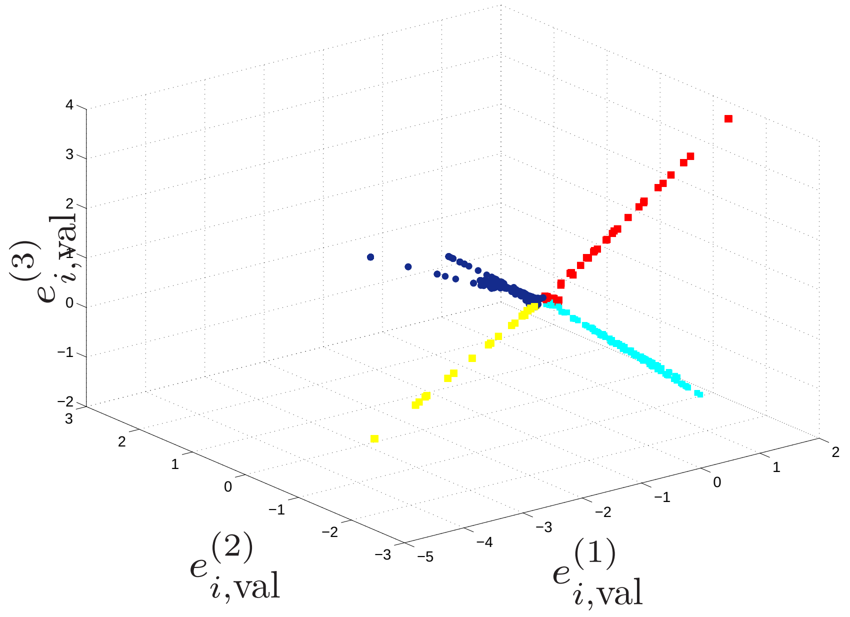


validation set

train + validation + test data

Example: image segmentation (looking for lines)





$e^{(3)}_{i,\mathrm{val}}$
$e^{(2)}_{i,\mathrm{val}}$
$e^{(1)}_{i,\mathrm{val}}$

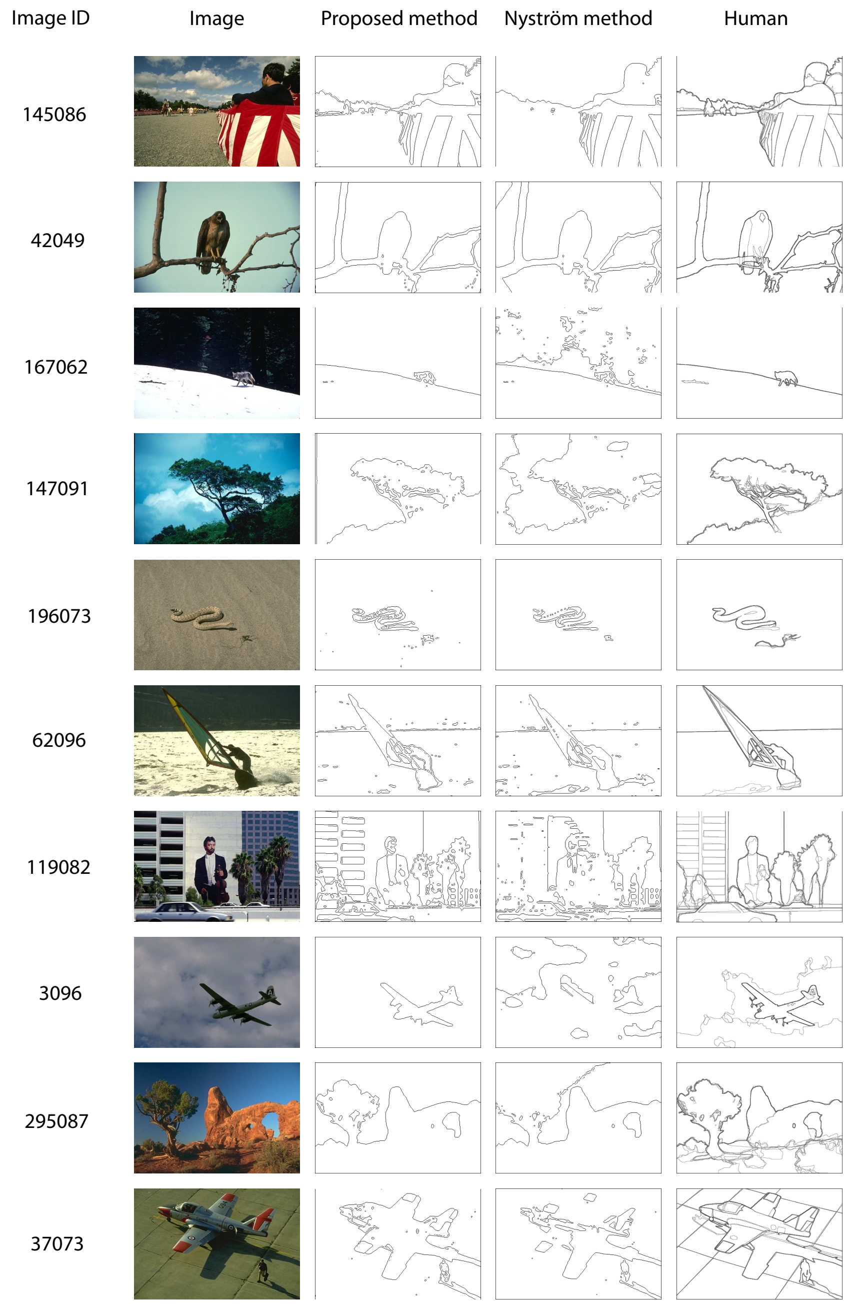
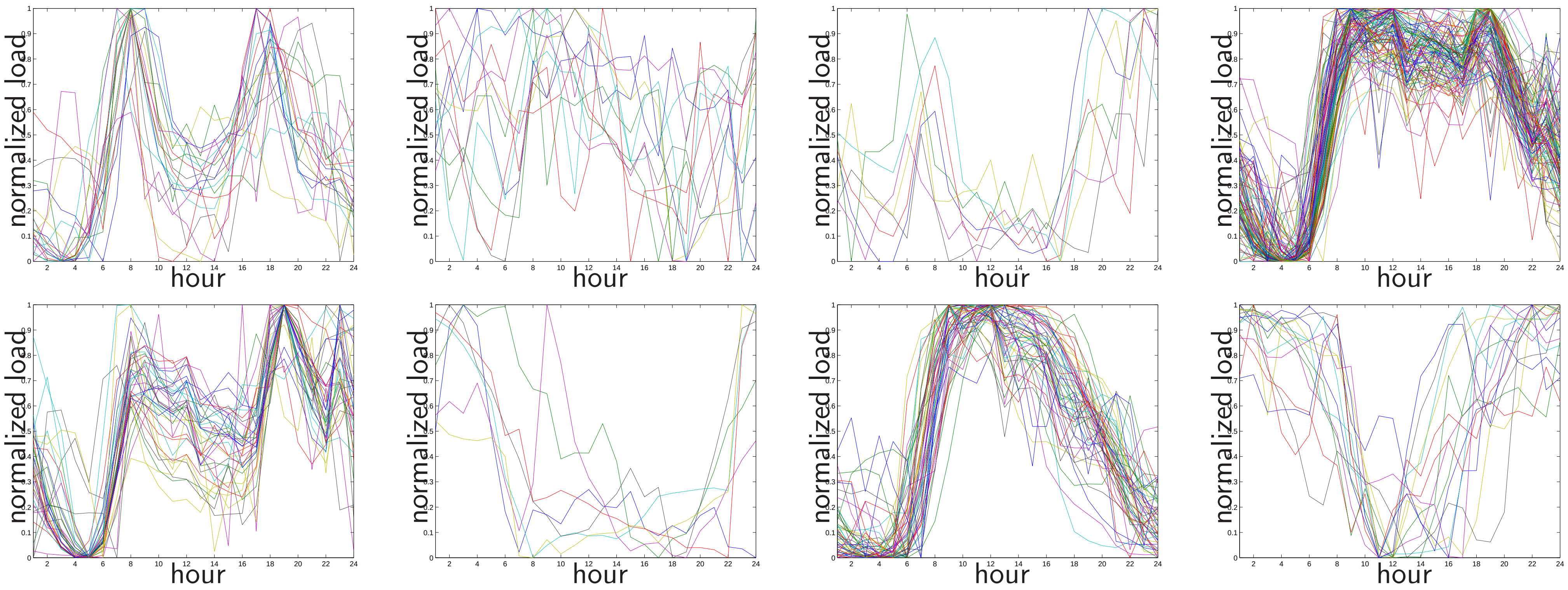


Image ID
Image
Proposed method
Nyström method
Human
145086
42049
167062
147091
196073
62096
119082
3096
295087
37073

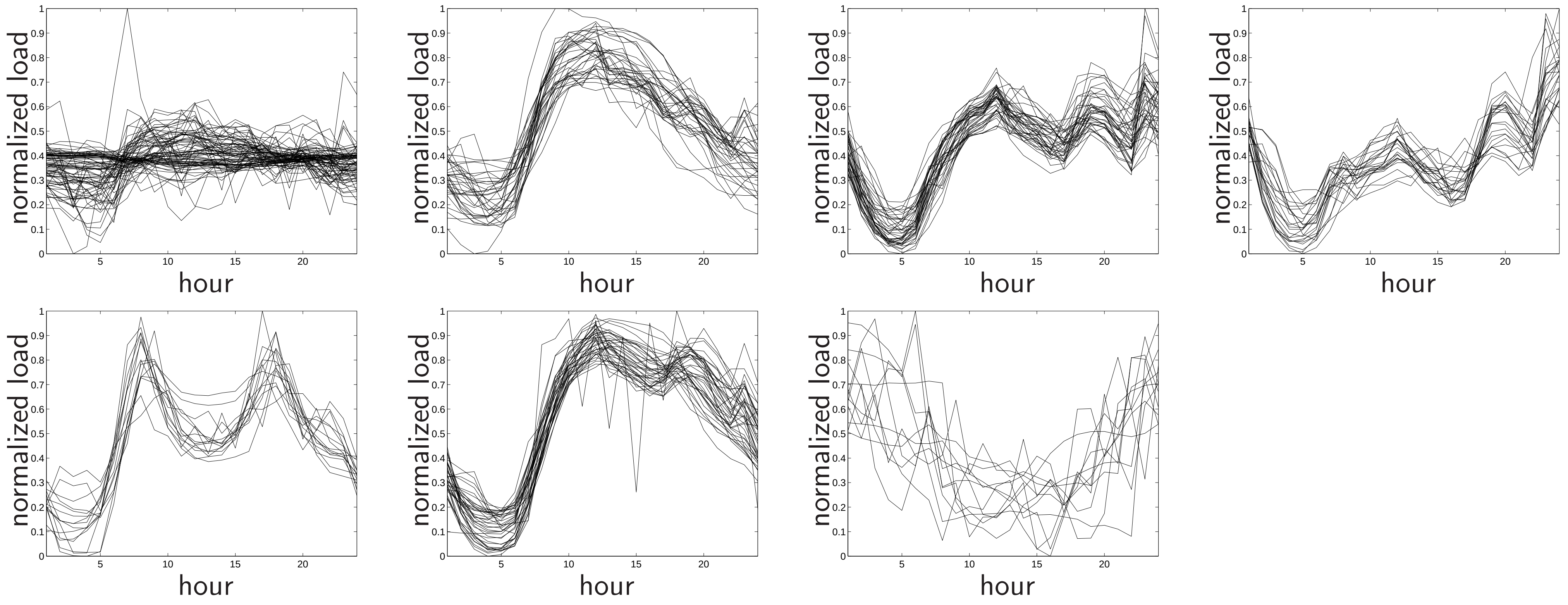
Example: power grid - identifying customer profiles (1)

Power load: 245 substations, hourly data (5 years), $d = 43.824$
Periodic AR modelling: dimensionality reduction $43.824 \rightarrow 24$
k-means clustering applied after dimensionality reduction

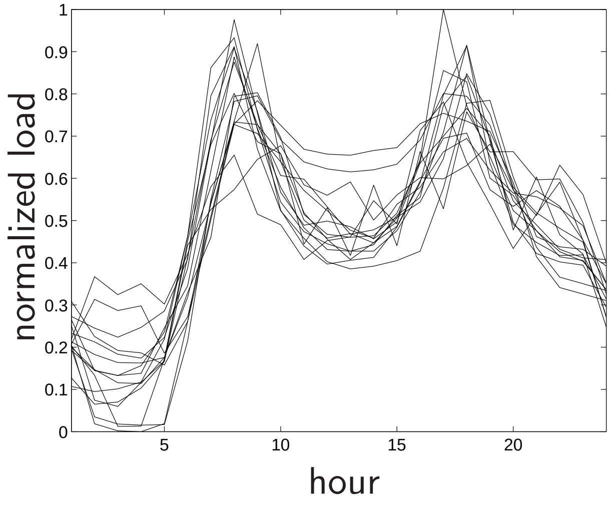


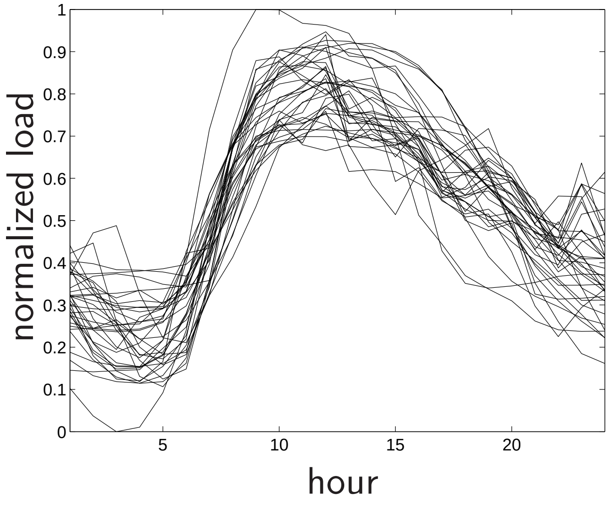
Example: power grid - identifying customer profiles (2)

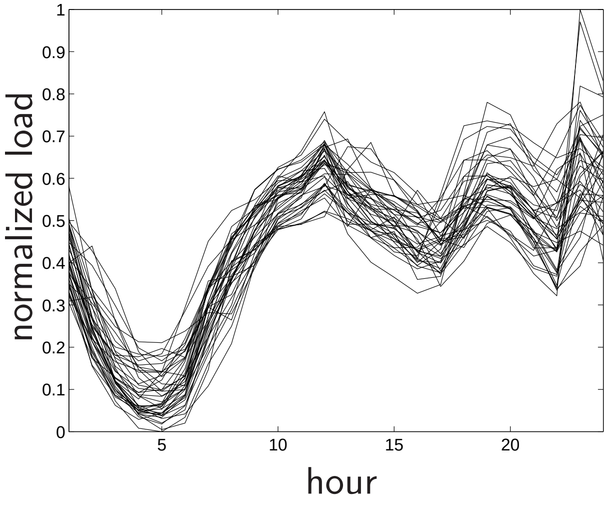
Application of kernel spectral clustering, directly on $d = 43.824$
Model selection on kernel parameter and number of clusters
[Alzate, Espinoza, De Moor, Suykens, 2009]



Example: power grid - identifying customer profiles (3)







Electricity load: 245 substations in Belgian grid (1/2 train, 1/2 validation)
$x_i \in \mathbb{R}^{43.824}$: spectral clustering on **high dimensional data** (5 years)

3 of 7 detected clusters:
- *1: Residential profile: morning and evening peaks*
- *2: Business profile: peaked around noon*
- *3: Industrial profile: increasing morning, oscillating afternoon and evening*

Kernel spectral clustering: adding prior knowledge

- Pair of points $x_\dagger, x_\ddagger$: $c = 1$ must-link, $c = -1$ cannot-link
- Primal problem [Alzate & Suykens, IJCNN 2009]

$$\begin{array}{ll}\displaystyle\min_{w^{(l)},e^{(l)},b_l} & \displaystyle -\frac{1}{2}\sum_{l=1}^{k-1} {w^{(l)}}^T w^{(l)} + \frac{1}{2}\sum_{l=1}^{k-1} \gamma_l {e^{(l)}}^T D^{-1} e^{(l)} \\ \text{subject to} & e^{(1)} = \Phi_{N\times n_h} w^{(1)} + b_1 1_N \\ & \vdots \\ & e^{(k-1)} = \Phi_{N\times n_h} w^{(k-1)} + b_{k-1} 1_N \end{array}$$

-

Kernel spectral clustering: adding prior knowledge

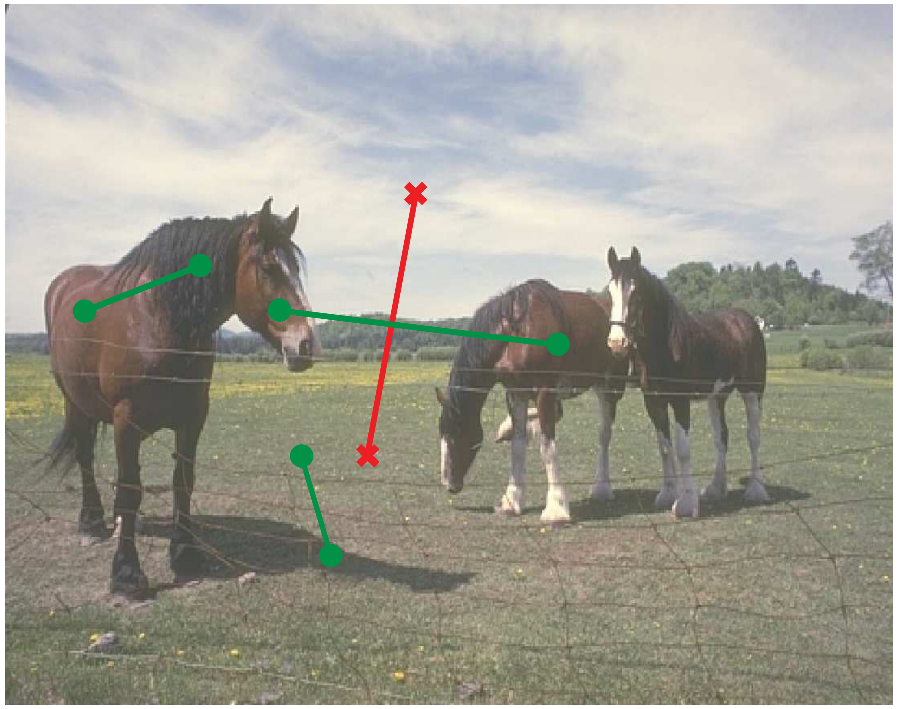
- Pair of points $x_\dagger, x_\ddagger$: $c = 1$ must-link, $c = -1$ cannot-link
- Primal problem [Alzate & Suykens, IJCNN 2009]

$$\begin{array}{ll}\displaystyle\min_{w^{(l)},e^{(l)},b_l} & \displaystyle -\frac{1}{2}\sum_{l=1}^{k-1} {w^{(l)}}^T w^{(l)} + \frac{1}{2}\sum_{l=1}^{k-1} \gamma_l {e^{(l)}}^T D^{-1} e^{(l)} \\ \text{subject to} & e^{(1)} = \Phi_{N\times n_h} w^{(1)} + b_1 1_N \\ & \vdots \\ & e^{(k-1)} = \Phi_{N\times n_h} w^{(k-1)} + b_{k-1} 1_N \\ & {w^{(1)}}^T \varphi(x_\dagger) = c {w^{(1)}}^T \varphi(x_\ddagger) \\ & \vdots \\ & {w^{(k-1)}}^T \varphi(x_\dagger) = c {w^{(k-1)}}^T \varphi(x_\ddagger)\end{array}$$

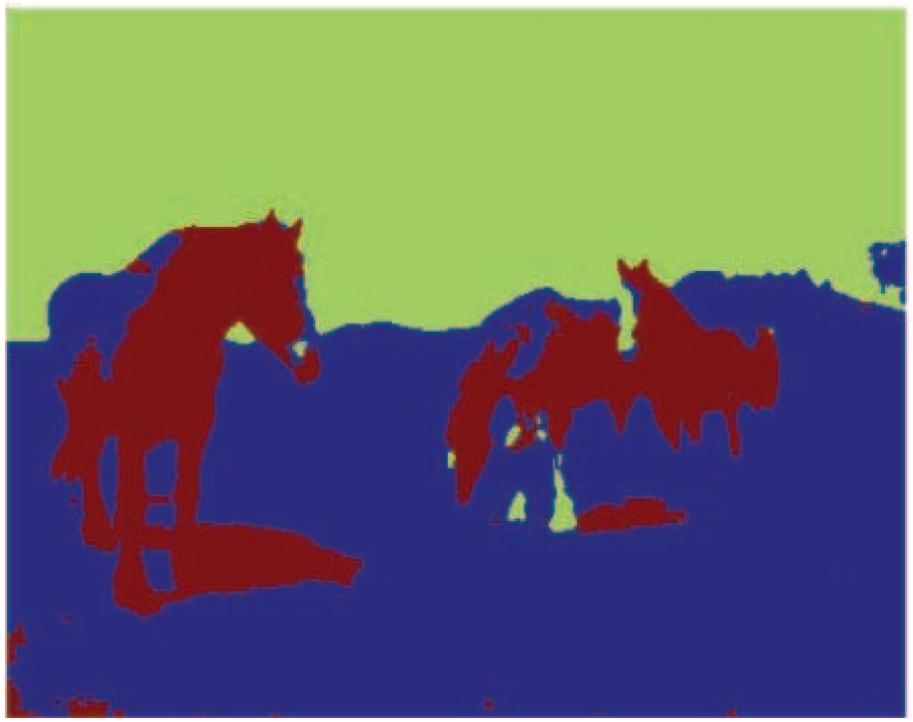
- Dual problem: yields rank-one downdate of the kernel matrix

Adding prior knowledge

original image

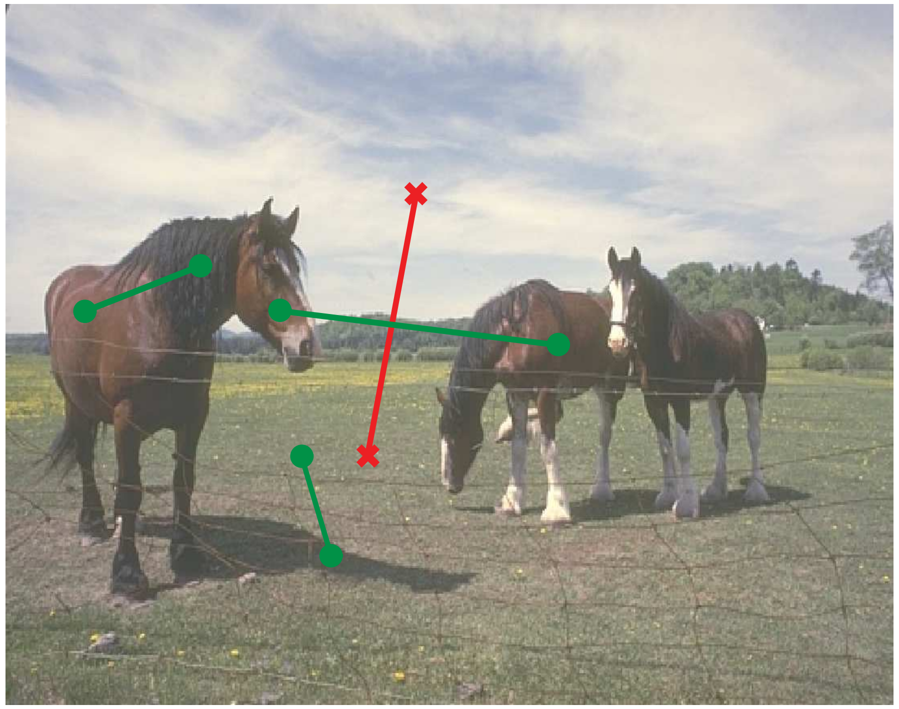


without constraints

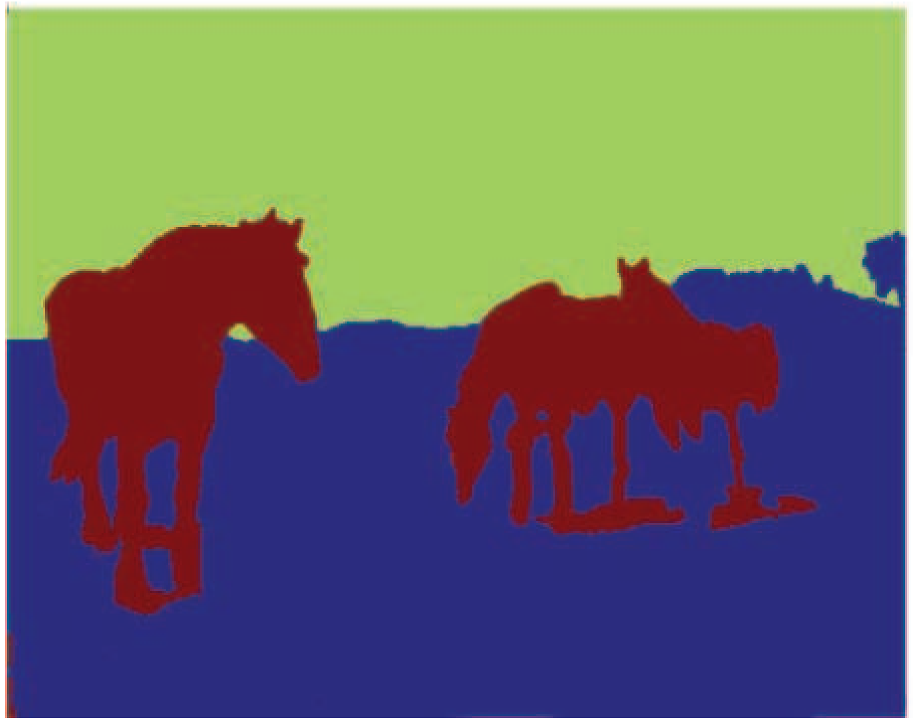


Adding prior knowledge

original image



with constraints



You give the model a bit of side information.

LS-SVM in recurrent networks and optimal control

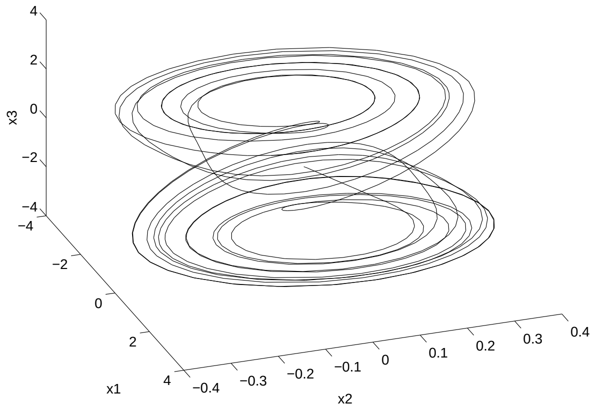
Recurrent least squares support vector machines

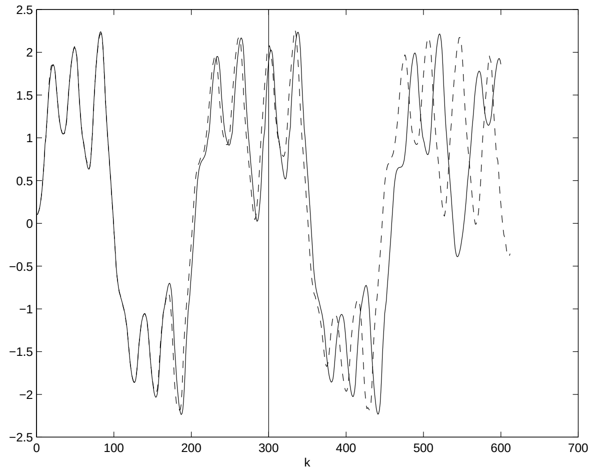
- Nonlinear AR model structure (feedforward):
  $\hat{y}_k = f(y_{k-1}, y_{k-2}, ..., y_{k-p})$
  Nonlinear OE model structure (recurrent):
  $\hat{y}_k = f(\hat{y}_{k-1}, \hat{y}_{k-2}, ..., \hat{y}_{k-p})$

- Recurrent LS-SVM (leads to set of nonlinear equations with kernel trick)

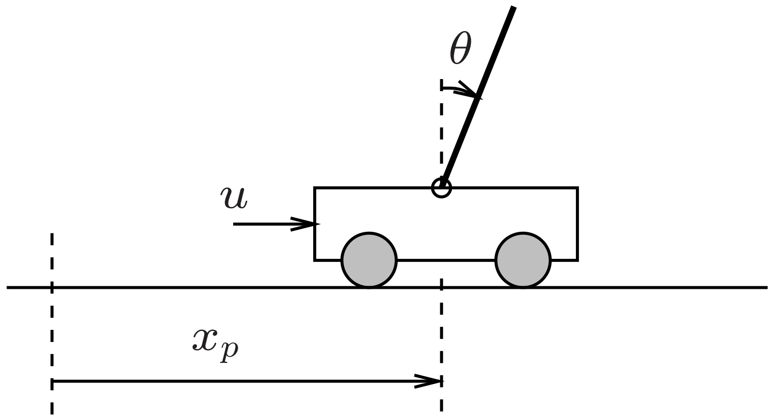
$$\min_{w,b,e} \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_k e_k^2 \text{ s.t. } y_k - e_k = w^T \varphi(y_{k-1|k-p} - e_{k-1|k-p}) + b, \ \forall k$$

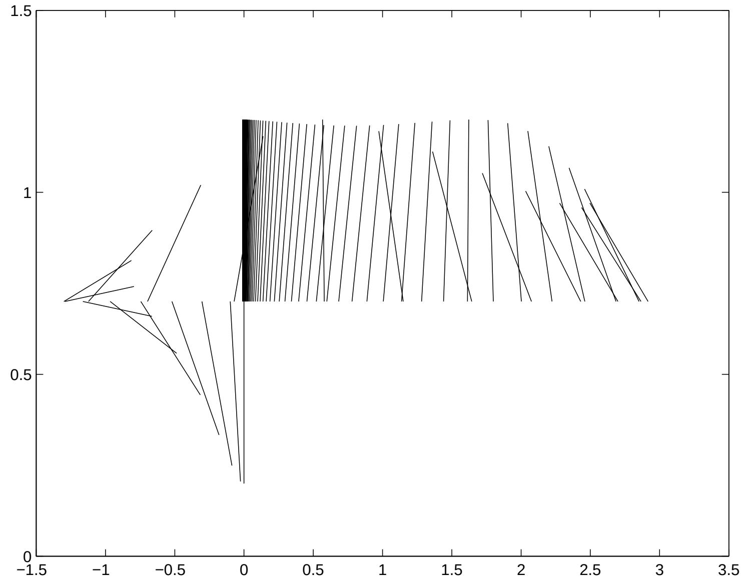
- Example: chaotic time-series prediction





Nonlinear control





min Control objective + LS−SVM objective

subject to $\begin{cases}\text{System dynamics (time k = 1, 2, ... , N)} \\ \text{LS−SVM controller (time k = 1, 2, ... , N)}\end{cases}$

Merging optimal control and support vector machine objectives:

Approximate solutions to optimal control problems [Suykens et al., NN 2001]

N-stage optimal control problem (1)

- 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

N-stage optimal control problem (2)

- 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:

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

Optimal control using LS-SVM (1)

- **Optimal control problem:**

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

subject to the **system dynamics**

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

and the **LS-SVM based control law**

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

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

Optimal control using LS-SVM (2)

- Lagrangian: $\mathcal{L}(x_k, u_k, w, e_k; \lambda_k, \alpha_k) = \mathcal{J}_N(x_k, u_k) + \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2 + \sum_{k=1}^N \lambda_k^T [x_{k+1} - f(x_k, u_k)] + \sum_{k=1}^N \alpha_k [u_k - w^T \varphi(x_k) - e_k]$

- Conditions for optimality:

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

Eliminating w, e gives $u_k = \sum_{l=1}^N \alpha_l K(x_l, x_k)$.