

Deep Learning and Kernel Machines

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/>

VISUM 2020, Porto, July 2020
Computer Vision and Machine Learning Summer School



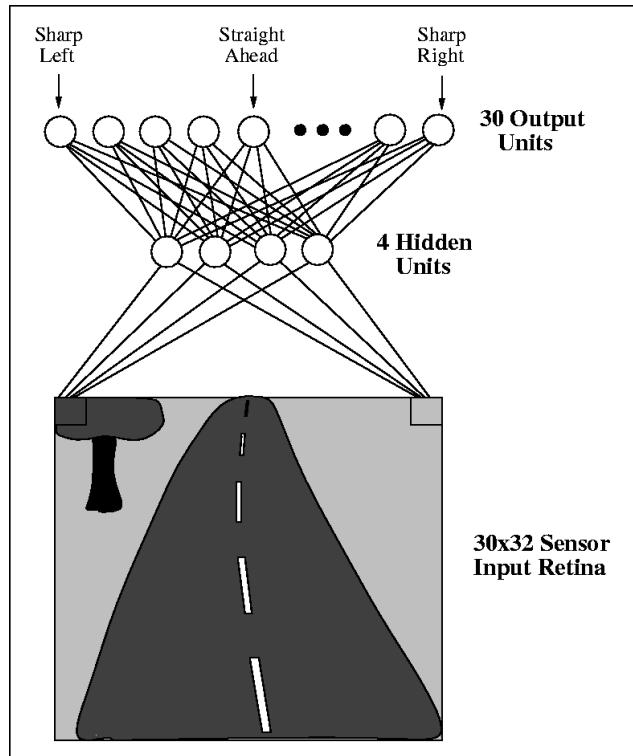
Overview

- Introduction
- Function estimation and model representations
- Least squares support vector machines (LS-SVM) as core models
- Kernel spectral clustering (KSC)
- Restricted kernel machines (RKM), generative models
- Deep learning and kernel machines: new synergies
- Conclusions

Introduction

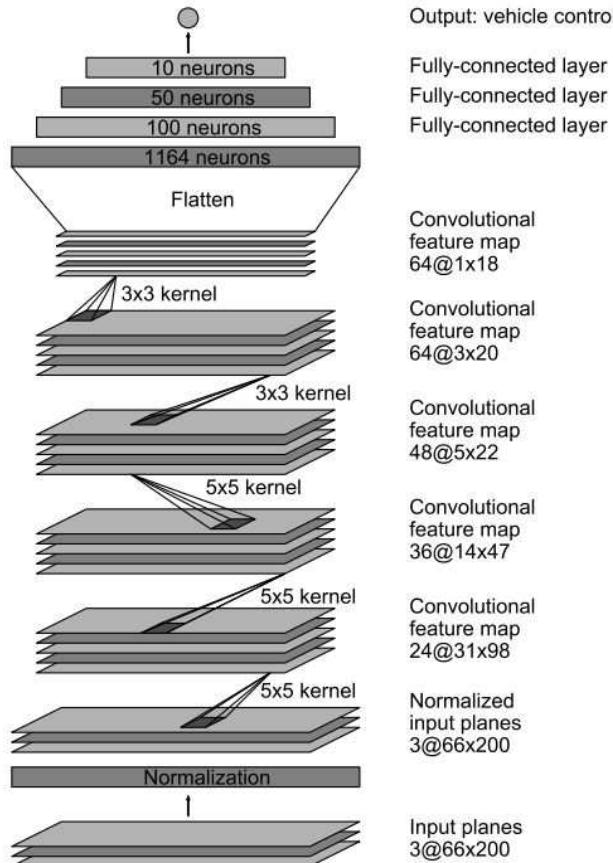
Self-driving cars and neural networks

in the early days of neural networks:



ALVINN (Autonomous Land Vehicle In a Neural Network)
[Pomerleau, Neural Computation 1991]

Self-driving cars and deep learning



Waymo / Google Self-Driving Car



Tesla Autopilot



Uber



nuTonomy

from: [selfdrivingcars.mit.edu (Lex Fridman et al.), 2017]

Convolutional neural networks

PROC. OF THE IEEE, NOVEMBER 1998

7

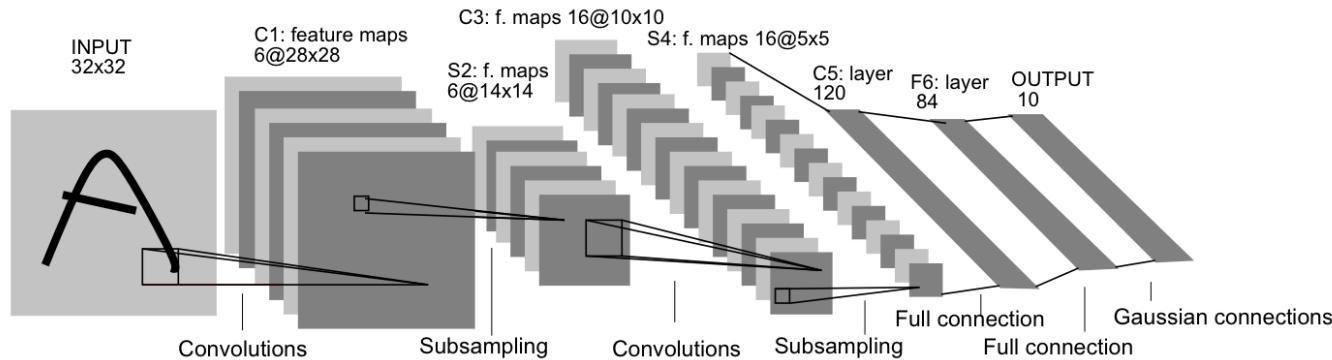


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

[LeCun et al., Proc. IEEE 1998]

Further advanced architectures:

Alexnet (2012): 5 convolutional layers, 3 fully connected

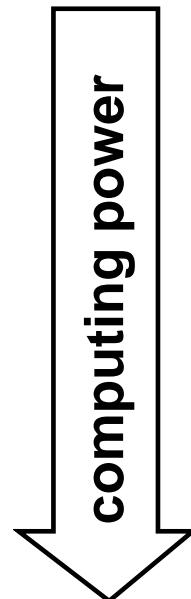
VGGnet (2014): 19 layers

GoogLeNet (2014): 22 layers

ResNet (2015): 152 layers

Historical context

- 1942 McCulloch & Pitts: mathematical model for neuron
- 1958 Rosenblatt: perceptron learning
- 1960 Widrow & Hoff: adaline and lms learning rule
- 1969 Minsky & Papert: limitations of perceptron
- 1986 Rumelhart et al.: error backpropagating neural networks
→ *booming of neural network universal approximators*
- 1992 Vapnik et al.: support vector machine classifiers
→ *convex optimization, kernel machines*
- 1998 LeCun et al.: convolutional neural networks
- 2006 Hinton et al.: deep belief networks
- 2010 Bengio et al.: stacked autoencoders
→ *booming of deep neural networks*



Different paradigms

Deep
Learning

Neural
Networks

SVM, LS-SVM &
Kernel methods

Different paradigms

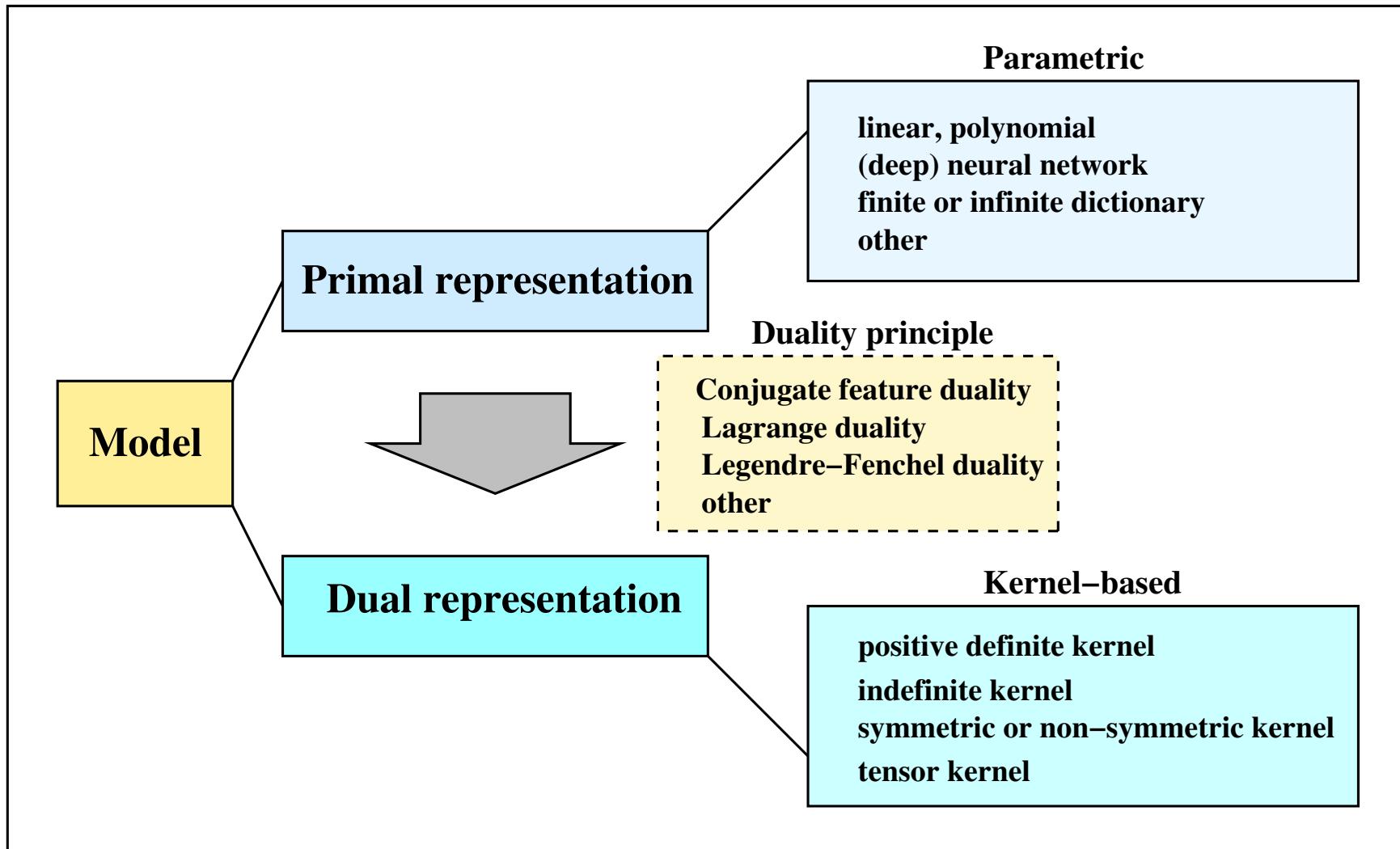
Deep
Learning

Neural
Networks

new synergies?

SVM, LS-SVM &
Kernel methods

Towards a unifying picture



[Suykens 2017]

Function estimation and model representations

Linear function estimation (1)

- Given $\{(x_i, y_i)\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, consider $\hat{y} = f(x)$ where f is parametrized as

$$\hat{y} = w^T x + b$$

with \hat{y} the estimated output of the linear model.

- Consider estimating w, b by

$$\min_{w,b} \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^N (y_i - w^T x_i - b)^2$$

→ one can directly solve in w, b

Linear function estimation (2)

- ... or write as a constrained optimization problem:

$$\begin{aligned} \min_{w, b, e} \quad & \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_i e_i^2 \\ \text{subject to} \quad & e_i = y_i - w^T x_i - b, \quad i = 1, \dots, N \end{aligned}$$

Lagrangian: $\mathcal{L}(w, b, e_i, \alpha_i) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_i e_i^2 - \sum_i \alpha_i (e_i - y_i + w^T x_i + b)$

- From optimality conditions:

$$\hat{y} = \sum_i \alpha_i x_i^T x + b$$

where α, b follows from solving a linear system

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

with $\Omega_{ij} = x_i^T x_j$ for $i, j = 1, \dots, N$ and $y = [y_1; \dots; y_N]$.

Linear model: solving in primal or dual?

inputs $x \in \mathbb{R}^d$, output $y \in \mathbb{R}$

training set $\{(x_i, y_i)\}_{i=1}^N$

$$(P) : \hat{y} = \mathbf{w}^T x + b, \quad w \in \mathbb{R}^d$$

↗
Model

Linear model: solving in primal or dual?

inputs $x \in \mathbb{R}^d$, output $y \in \mathbb{R}$

training set $\{(x_i, y_i)\}_{i=1}^N$

Model

$$(P) : \hat{y} = \mathbf{w}^T x + b, \quad w \in \mathbb{R}^d$$
$$(D) : \hat{y} = \sum_i \alpha_i x_i^T x + b, \quad \alpha \in \mathbb{R}^N$$

Linear model: solving in primal or dual?

few inputs, many data points: $d \ll N$

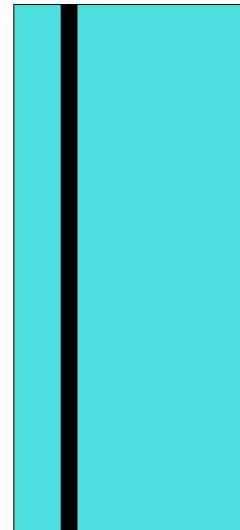


primal: $w \in \mathbb{R}^d$

dual: $\alpha \in \mathbb{R}^N$ (large kernel matrix: $N \times N$)

Linear model: solving in primal or dual?

many inputs, few data points: $d \gg N$



primal: $w \in \mathbb{R}^d$

dual: $\alpha \in \mathbb{R}^N$ (small kernel matrix: $N \times N$)

Feature map and kernel

From linear to nonlinear model:

Model

$$(P) : \hat{y} = w^T \varphi(x) + b$$
$$(D) : \hat{y} = \sum_i \alpha_i K(x_i, x) + b$$

Mercer theorem (one can **either** choose φ **or** positive definite K):

$$K(x, z) = \varphi(x)^T \varphi(z)$$

Feature map φ , Kernel function $K(x, z)$ (e.g. linear, polynomial, RBF, ...)

- SVMs: feature map and positive definite kernel [Cortes & Vapnik, 1995]
- Neural networks: consider hidden layer as feature map [Suykens & Vandewalle, 1999]
- Least squares support vector machines [Suykens et al., 2002]: L_2 loss and regularization

Least Squares Support Vector Machines: “core models”

- Regression

$$\min_{w,b,e} w^T w + \gamma \sum_i e_i^2 \quad \text{s.t.} \quad y_i = w^T \varphi(x_i) + b + e_i, \quad \forall i$$

- Classification

$$\min_{w,b,e} w^T w + \gamma \sum_i e_i^2 \quad \text{s.t.} \quad y_i(w^T \varphi(x_i) + b) = 1 - e_i, \quad \forall i$$

- Kernel pca ($V = I$), Kernel spectral clustering ($V = D^{-1}$)

$$\min_{w,b,e} -w^T w + \gamma \sum_i v_i e_i^2 \quad \text{s.t.} \quad e_i = w^T \varphi(x_i) + b, \quad \forall i$$

- Kernel canonical correlation analysis/partial least squares

$$\min_{w,v,b,d,e,r} w^T w + v^T v + \nu \sum_i (e_i - r_i)^2 \quad \text{s.t.} \quad \begin{cases} e_i &= w^T \varphi^{(1)}(x_i) + b \\ r_i &= v^T \varphi^{(2)}(y_i) + d \end{cases}$$

[Suykens & Vandewalle, 1999; Suykens et al., 2002; Alzate & Suykens, 2010]

Sparsity: through regularization or loss function

- through regularization: model $\hat{y} = w^T x + b$

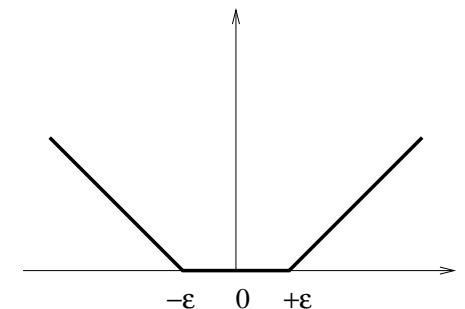
$$\min \sum_j |w_j| + \gamma \sum_i e_i^2$$

⇒ sparse w (e.g. Lasso)

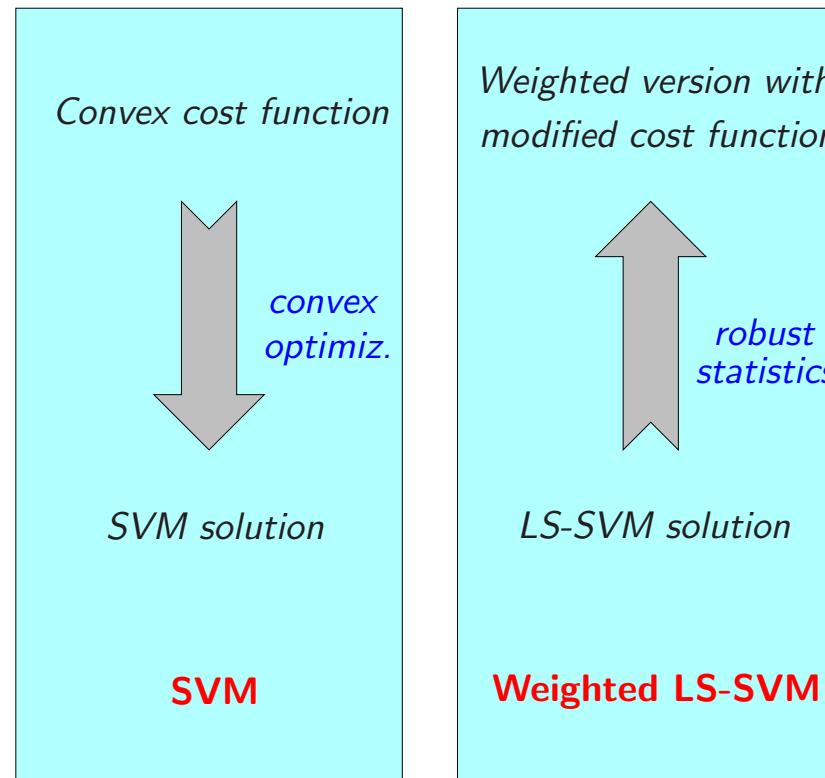
- through loss function: model $\hat{y} = \sum_i \alpha_i K(x, x_i) + b$

$$\min w^T w + \gamma \sum_i L(e_i)$$

⇒ sparse α (e.g. SVM)

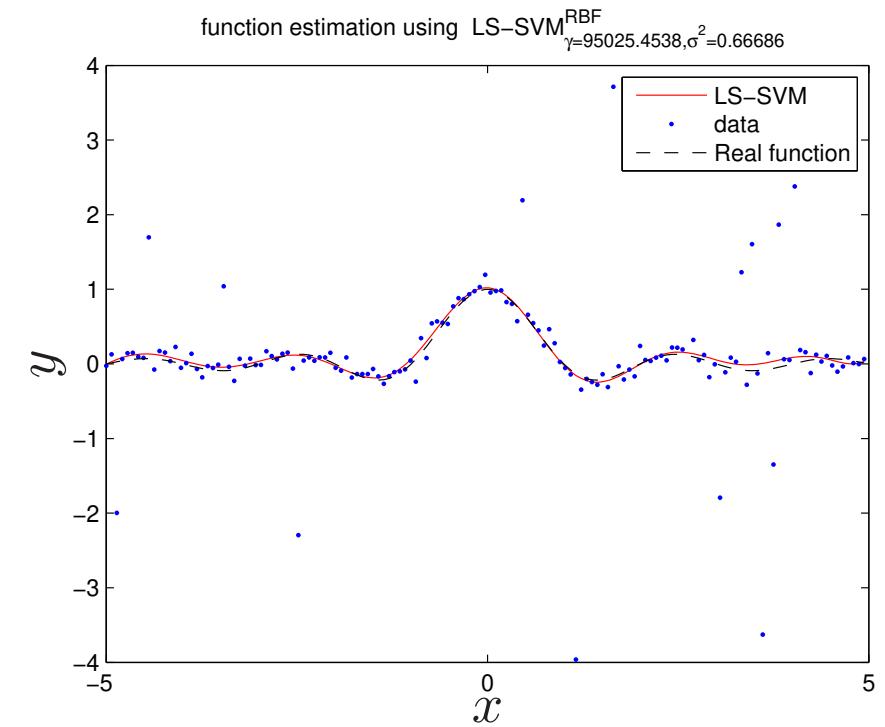
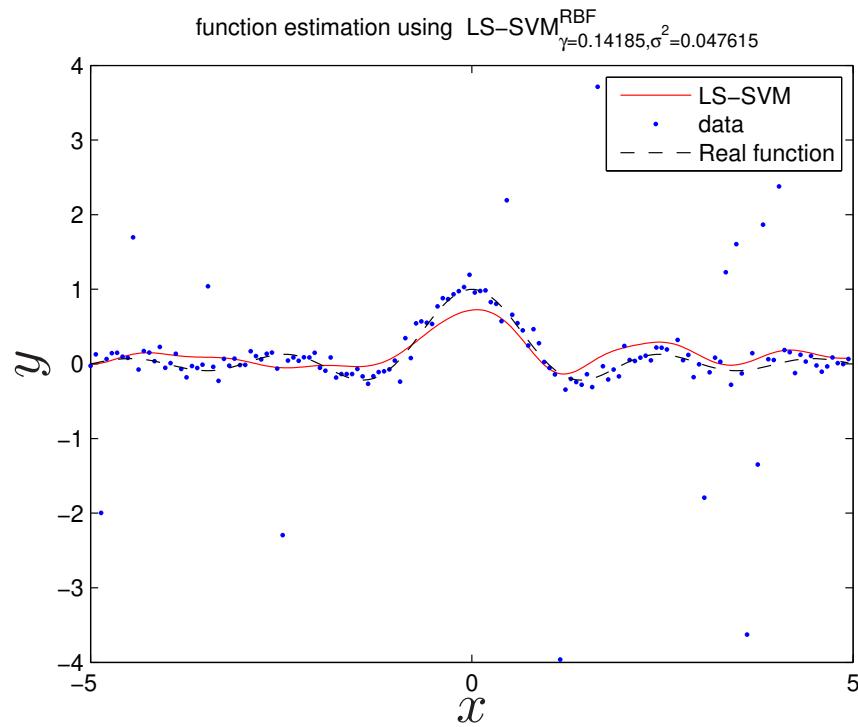


Weighted versions and robustness



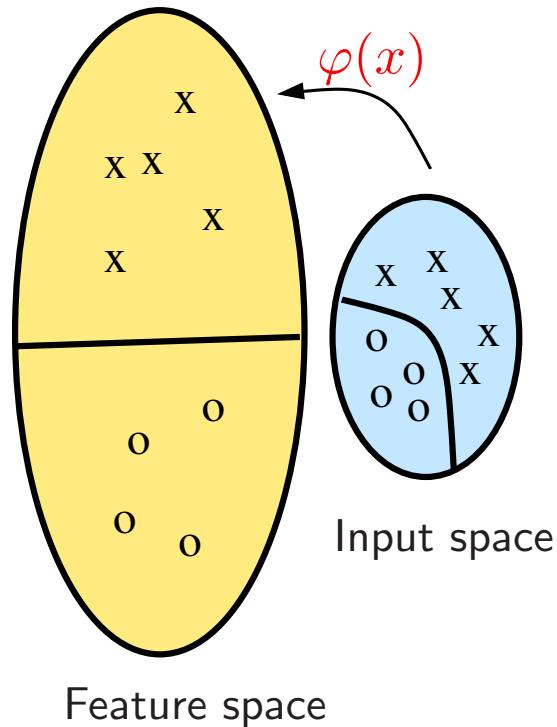
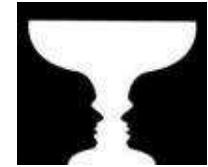
- **Weighted LS-SVM:** $\min_{w,b,e} \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^N v_i e_i^2$ s.t. $y_i = w^T \varphi(x_i) + b + e_i, \forall i$
with v_i determined from $\{e_i\}_{i=1}^N$ of unweighted LS-SVM [Suykens et al., 2002].
Robustness and stability [Debruyne et al., JMLR 2008, 2010].
- SVM solution by applying **iteratively weighted LS** [Perez-Cruz et al., 2005]

Example: robust regression using weighted LS-SVM



using LS-SVMLab v1.8 <http://www.esat.kuleuven.be/sista/lssvmlab/>

SVMs and neural networks

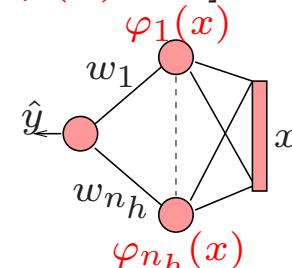


[Suykens et al., 2002]

Primal space

Parametric

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



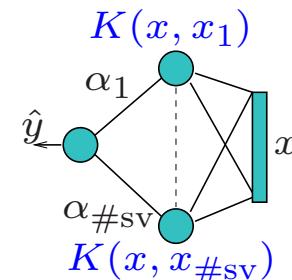
$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \text{ ("Kernel trick")}$$

Parametric

Dual space

Nonparametric

$$\hat{y} = \text{sign}\left[\sum_{i=1}^{\#\text{sv}} \alpha_i y_i K(x, x_i) + b\right]$$



Non-parametric

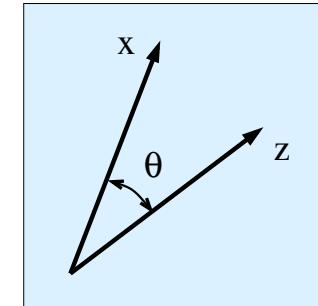
Wider use of the “kernel trick”

- Angle between vectors: (e.g. correlation analysis)

Input space:

$$\cos \theta_{xz} = \frac{x^T z}{\|x\|_2 \|z\|_2}$$

Feature space:



$$\cos \theta_{\varphi(x), \varphi(z)} = \frac{\varphi(x)^T \varphi(z)}{\|\varphi(x)\|_2 \|\varphi(z)\|_2} = \frac{K(x, z)}{\sqrt{K(x, x)} \sqrt{K(z, z)}}$$

- Distance between vectors: (e.g. for “kernelized” clustering methods)

Input space:

$$\|x - z\|_2^2 = (x - z)^T (x - z) = x^T x + z^T z - 2x^T z$$

Feature space:

$$\|\varphi(x) - \varphi(z)\|_2^2 = K(x, x) + K(z, z) - 2K(x, z)$$

Interpretation of kernel-based models

Decision making: classification problem (e.g. apples versus tomatoes)

Input data $x_i \in \mathbb{R}^d$ and class labels $y_i \in \{-1, +1\}$. N training data.

SVM or LS-SVM classifier: given a new x (e.g. ) , obtain

$$\hat{y} = \text{sign}\left[\sum_i \alpha_i y_i K(x, x_i) + b\right]$$

with x_i for $i = 1, \dots, N$:



Here $K(x, x_i)$ characterizes the similarity between x and x_i .

The bias term b can be related to prior class probabilities (Ethics & AI !).

Kernels

Wide range of positive definite kernel functions possible:

- linear $K(x, z) = x^T z$
- polynomial $K(x, z) = (\eta + x^T z)^d$
- radial basis function $K(x, z) = \exp(-\|x - z\|_2^2/\sigma^2)$
- χ^2 kernel (on images)
- Wasserstein exponential kernels (optimal transport)
- splines, wavelets
- string kernel
- Fisher kernels, kernels from graphical models
- kernels for dynamical systems
- graph kernels
- data fusion kernels
- additive kernels (good for explainability)
- other

[Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004; Jebara et al., 2004; other]

Function estimation in RKHS

- Find function f such that [Wahba, 1990; Evgeniou et al., 2000]

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

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

- Sparse representation by hinge and ϵ -insensitive loss [Vapnik, 1998]

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_-

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

- also: KPCA with indefinite kernel [X. Huang et al. 2017], KSC and semi-supervised learning [Mehrkanon et al., 2018]

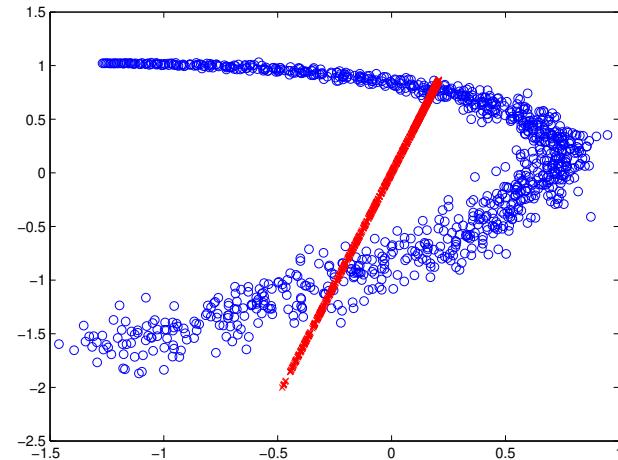
[X. Huang, Maier, Hornegger, Suykens, ACHA 2017]

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

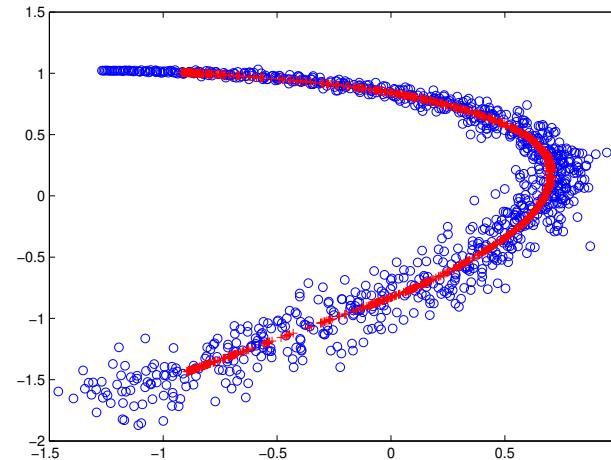
Related work of RKKS: [Ong et al 2004; Haasdonk 2005; Luss 2008; Loosli et al. 2015]

Kernel spectral clustering

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

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

- Dual problem corresponds to kernel PCA

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

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

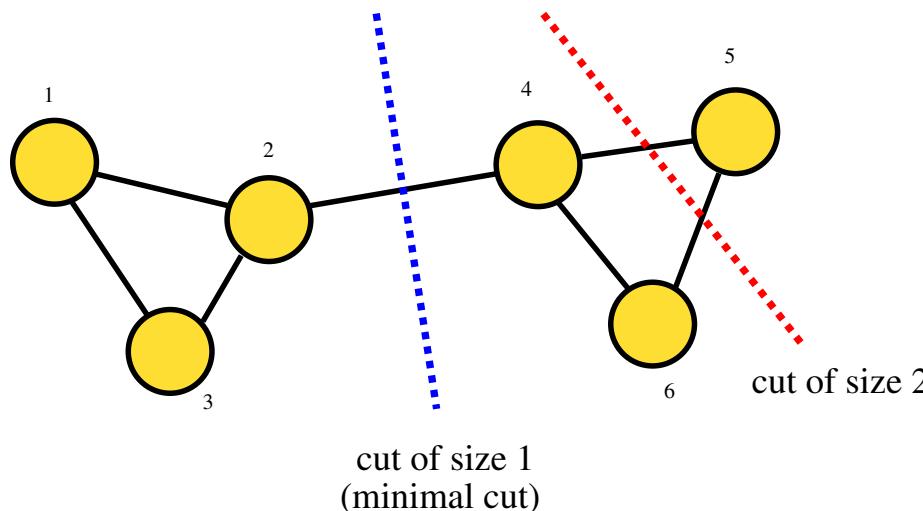
- Interpretation:
 1. pool of candidate components (objective function equals zero)
 2. select relevant components
- Robust and sparse versions [Alzate & Suykens, 2008]

Spectral graph clustering (1)

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.



Spectral graph clustering (2)

- **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 $D = \text{diag}(d_1, \dots, d_N)$, $d_i = \sum_j w_{ij}$, giving

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

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

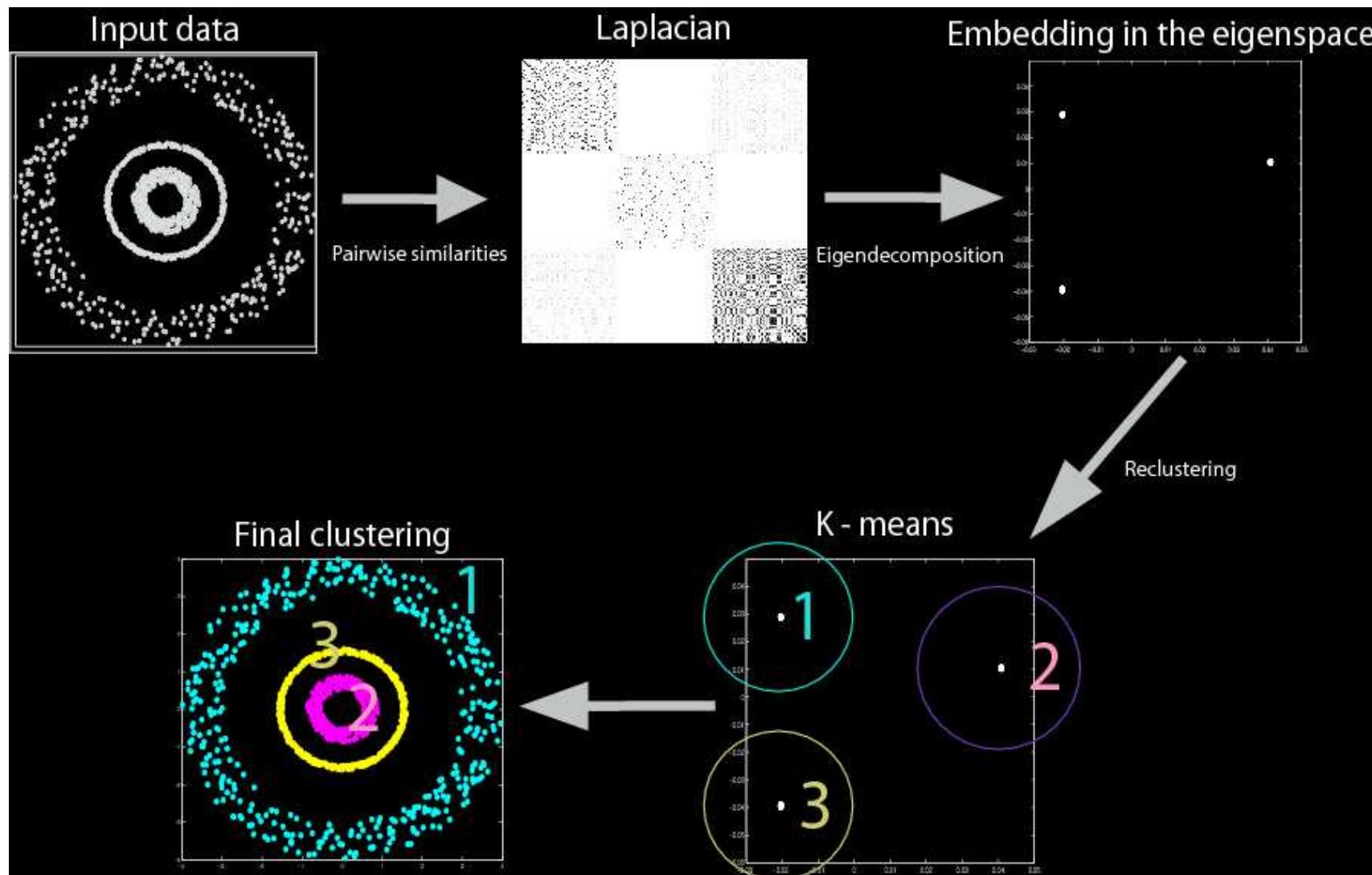
- **Normalized cut**

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

[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 (KSC): case of two clusters

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

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

with weighting matrix V and $\varphi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^h$ the feature map.

- **Dual:**

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

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 $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$

-

[Alzate & Suykens, IEEE-PAMI, 2010]

Kernel Spectral Clustering (KSC): case of two clusters

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

$$\begin{array}{ll}\min_{w,b,e} & \frac{1}{2}w^T w - \gamma \frac{1}{2} e^T \textcolor{red}{V} e \\ \text{subject to} & e_i = w^T \varphi(x_i) + b, \quad i = 1, \dots, N\end{array}$$

with weighting matrix V and $\varphi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^h$ the feature map.

- **Dual:**

$$VM_V\Omega\alpha = \lambda\alpha$$

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 $\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$

- Taking $\textcolor{red}{V} = D^{-1}$ with degree matrix $D = \text{diag}\{d_1, \dots, d_N\}$ and $d_i = \sum_{j=1}^N \Omega_{ij}$ relates to random walks algorithm.

[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:

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_i \alpha_i \varphi(x_i) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_i \alpha_i = 0 \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 \Rightarrow \alpha_i = \gamma v_i e_i, \quad i = 1, \dots, N \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \Rightarrow e_i = w^T \varphi(x_i) + b, \quad i = 1, \dots, N \end{array} \right.$$

- Eliminate w, b, e , write solution in Lagrange multipliers α_i .

Kernel spectral clustering: more clusters

- Case of k clusters: additional sets of constraints

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

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

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

[Alzate & Suykens, IEEE-PAMI, 2010]

Primal and dual model representations

k clusters

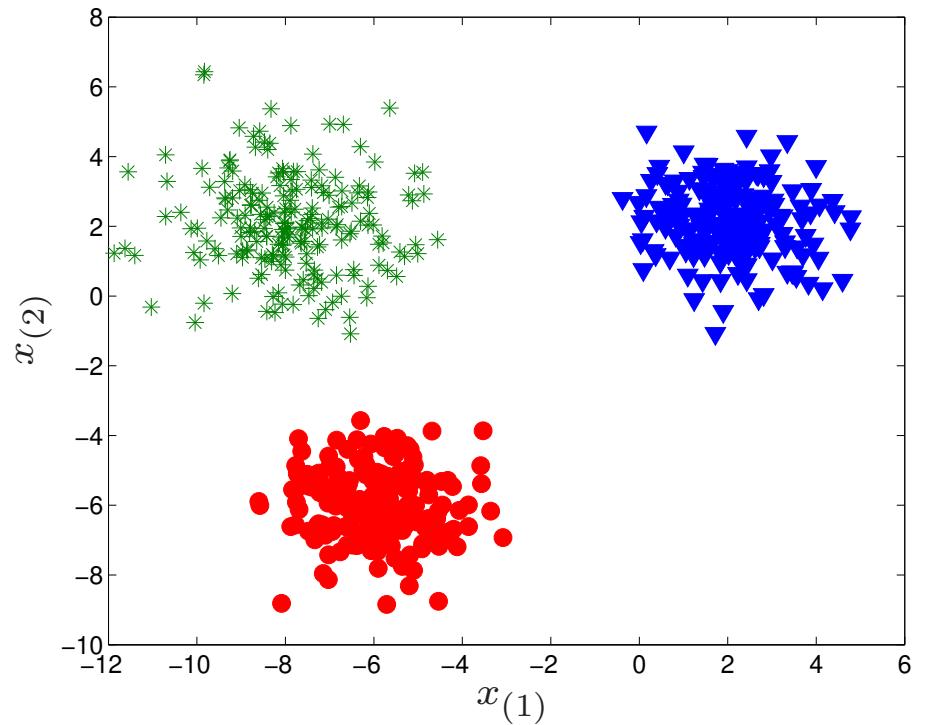
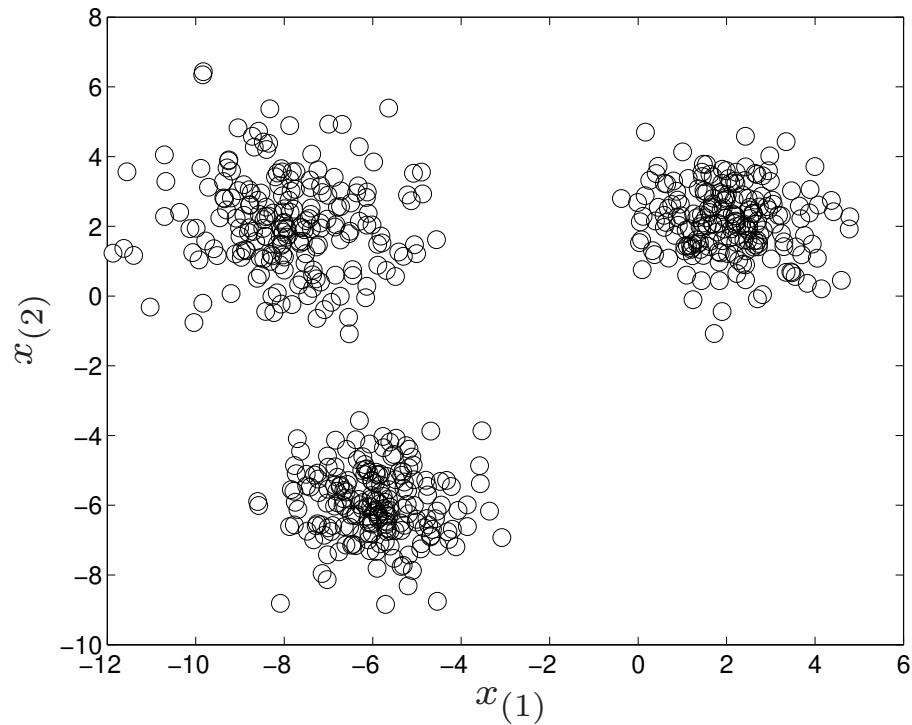
$k - 1$ sets of constraints (index $l = 1, \dots, k - 1$)

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

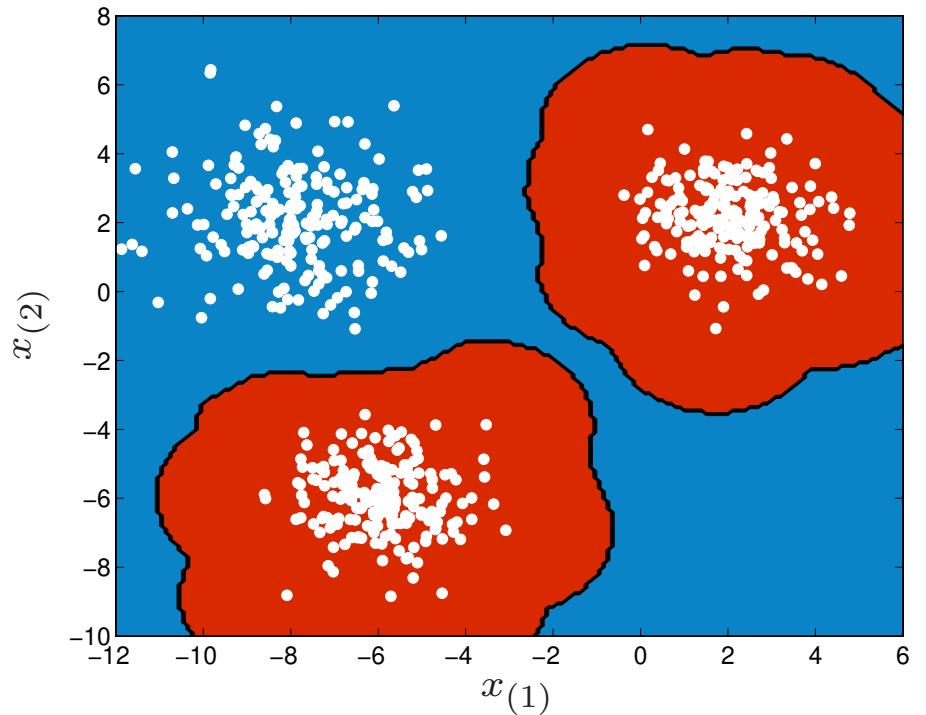
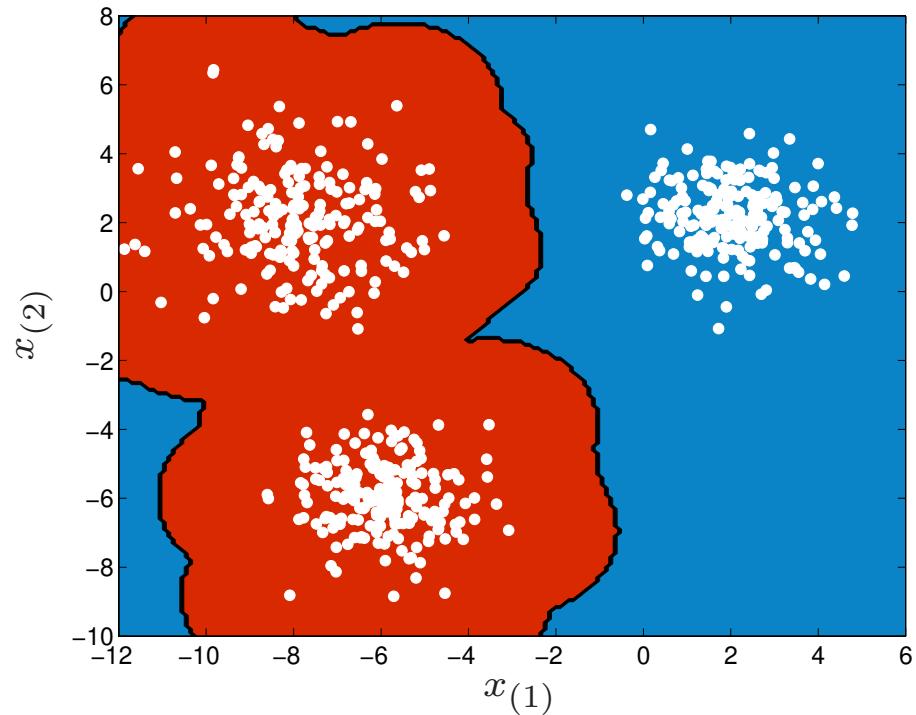
Advantages of kernel-based setting

- **model-based** approach
- **out-of-sample extensions**, applying model to new data
- consider **training, validation and test data**
(training problem corresponds to eigenvalue decomposition problem)
- model selection procedures
- **sparse representations and large scale methods**

Out-of-sample extension and coding



Out-of-sample extension and coding



Piecewise constant eigenvectors and extension (1)

- **Definition.** [Meila & Shi, 2001] Vector α is called *piecewise constant* relative to a partition $(\mathcal{A}_1, \dots, \mathcal{A}_k)$ iff $\alpha_i = \alpha_j \forall x_i, x_j \in \mathcal{A}_p, p = 1, \dots, k$.
- **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 $\{\mathcal{A}_1, \dots, \mathcal{A}_k\}$ 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 $l = 1, \dots, k - 1$ 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 $E_{ml}^v = e_m^{(l)} = \sum_{i=1}^N \alpha_i^{(l)} K(x_i, x_m^v) + b_l$.

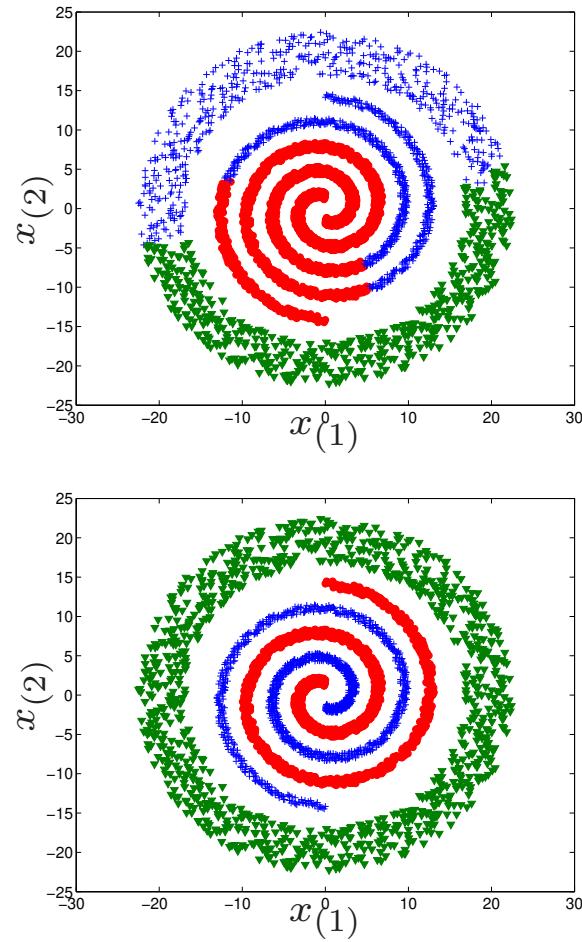
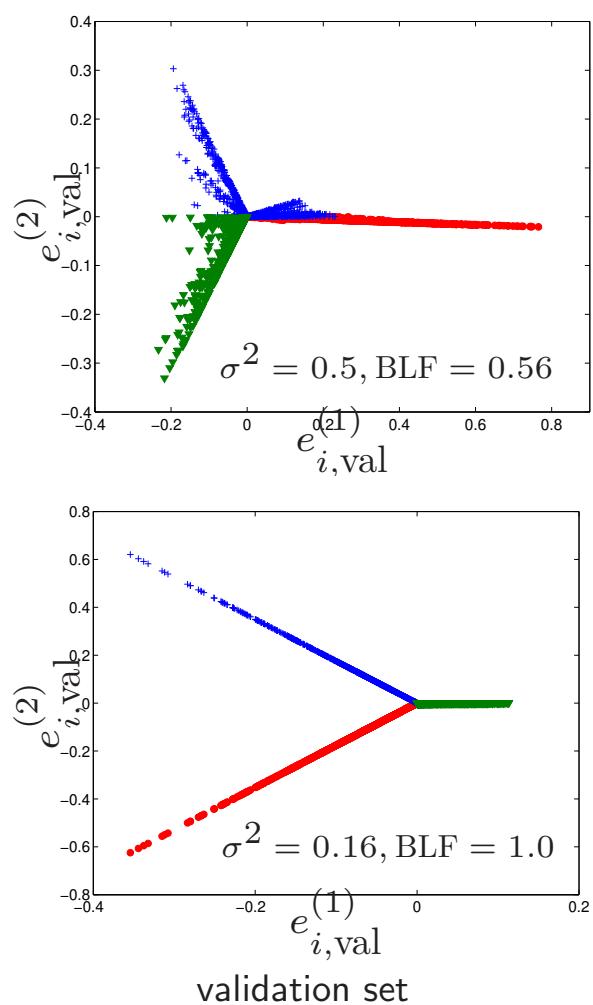
Piecewise constant eigenvectors and extension (2)

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

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

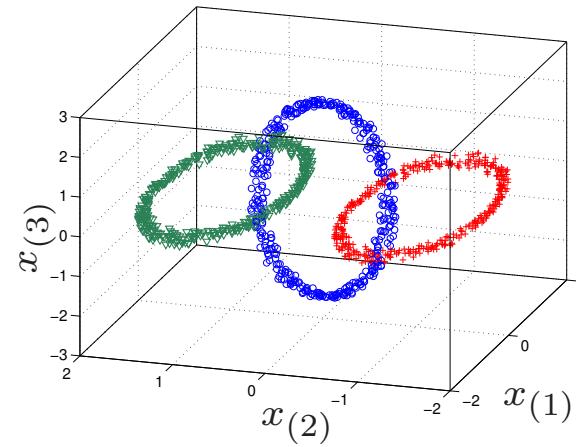
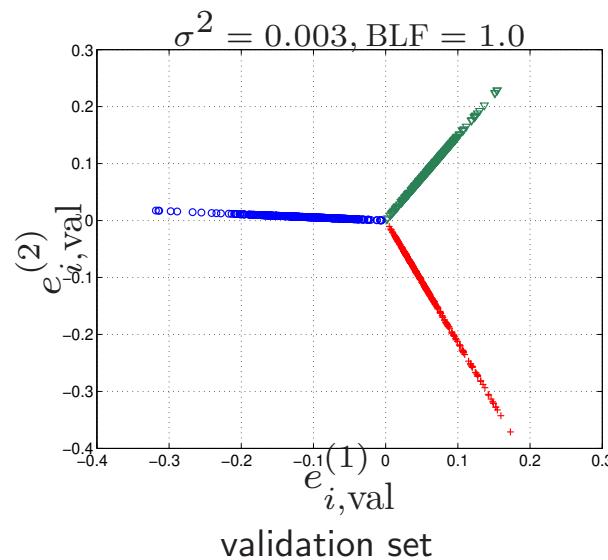
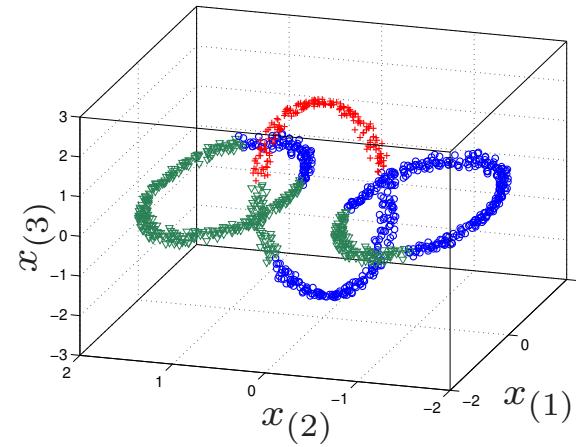
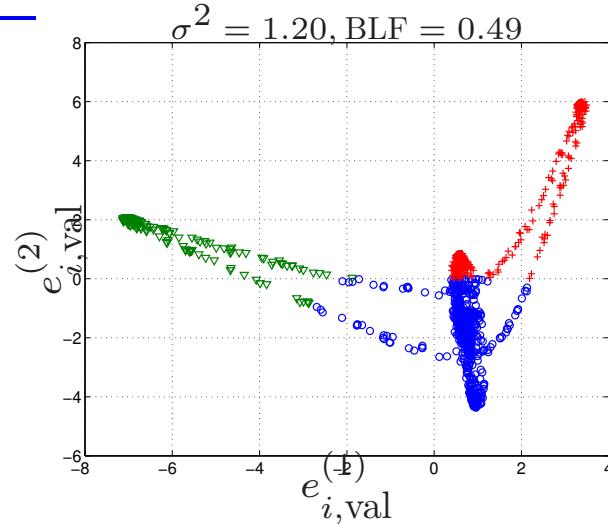
- **Model selection** to determine kernel parameters and k :
Looking for **line structures** in the space $(e_i^{(1)}, e_i^{(2)}, \dots, e_i^{(k-1)})$,
evaluated on **validation data** (aiming for good generalization)
- **Choice kernel:**
Gaussian RBF kernel
 χ^2 -kernel for images

Model selection (looking for lines): toy problem 1



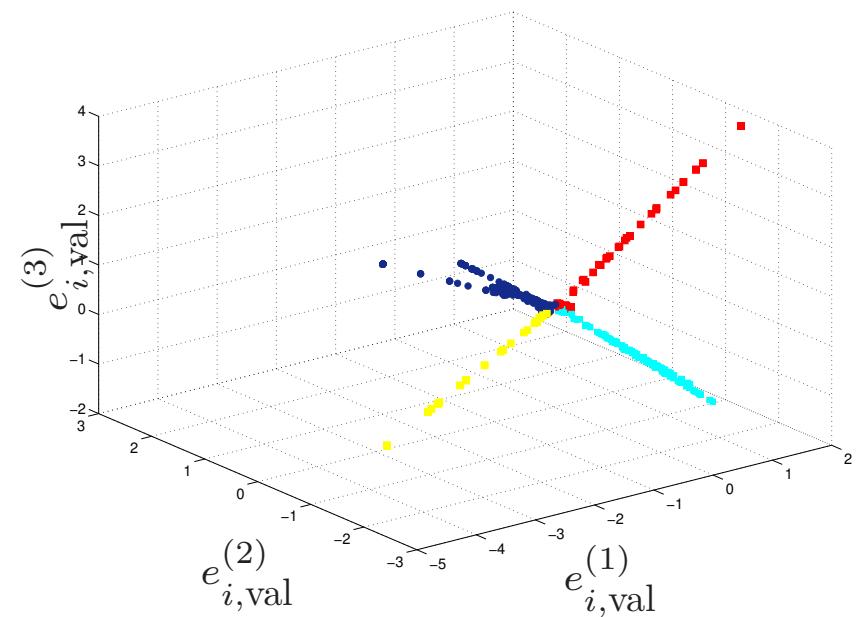
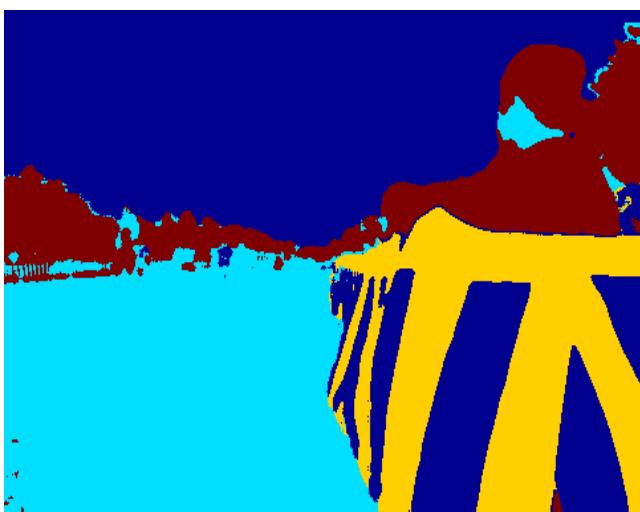
train + validation + test data

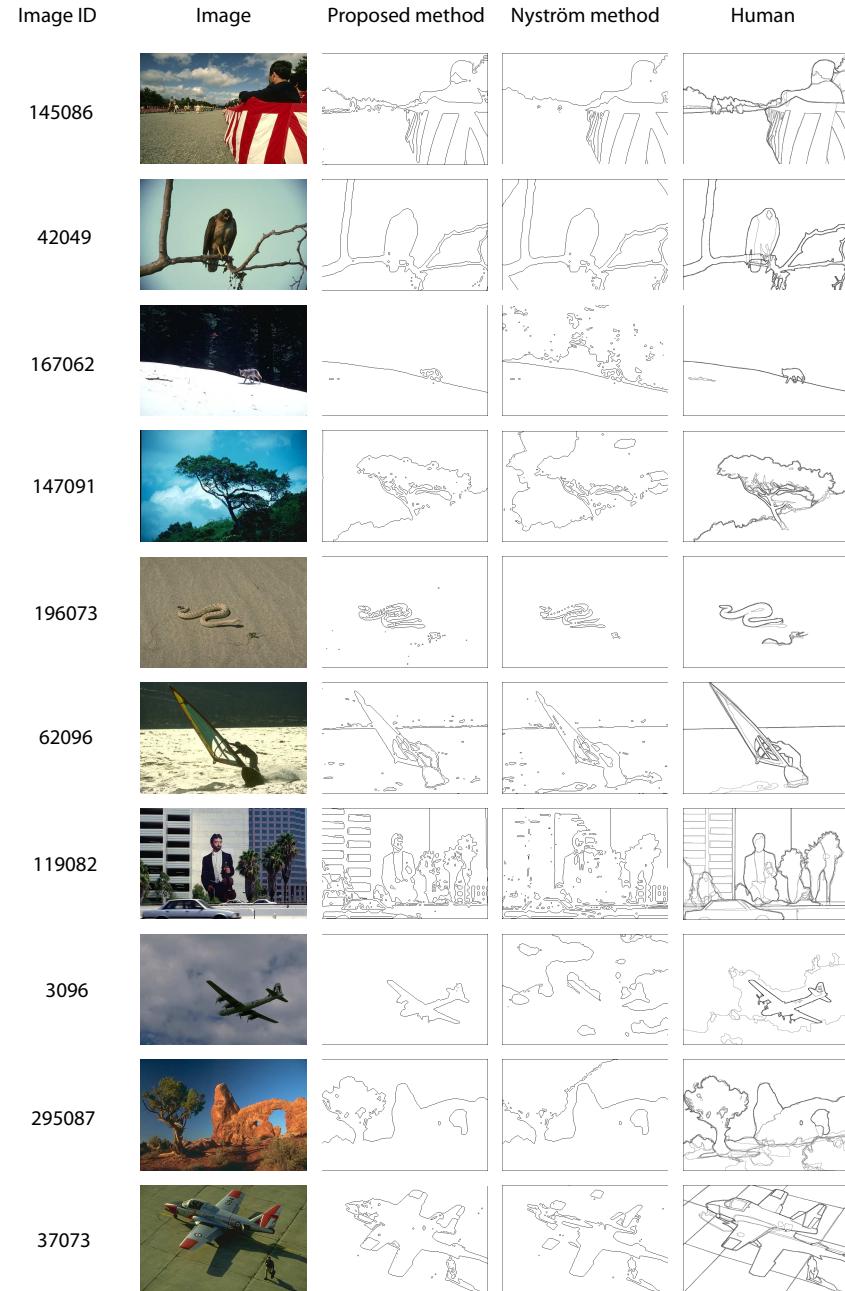
Model selection (looking for lines): toy problem 2



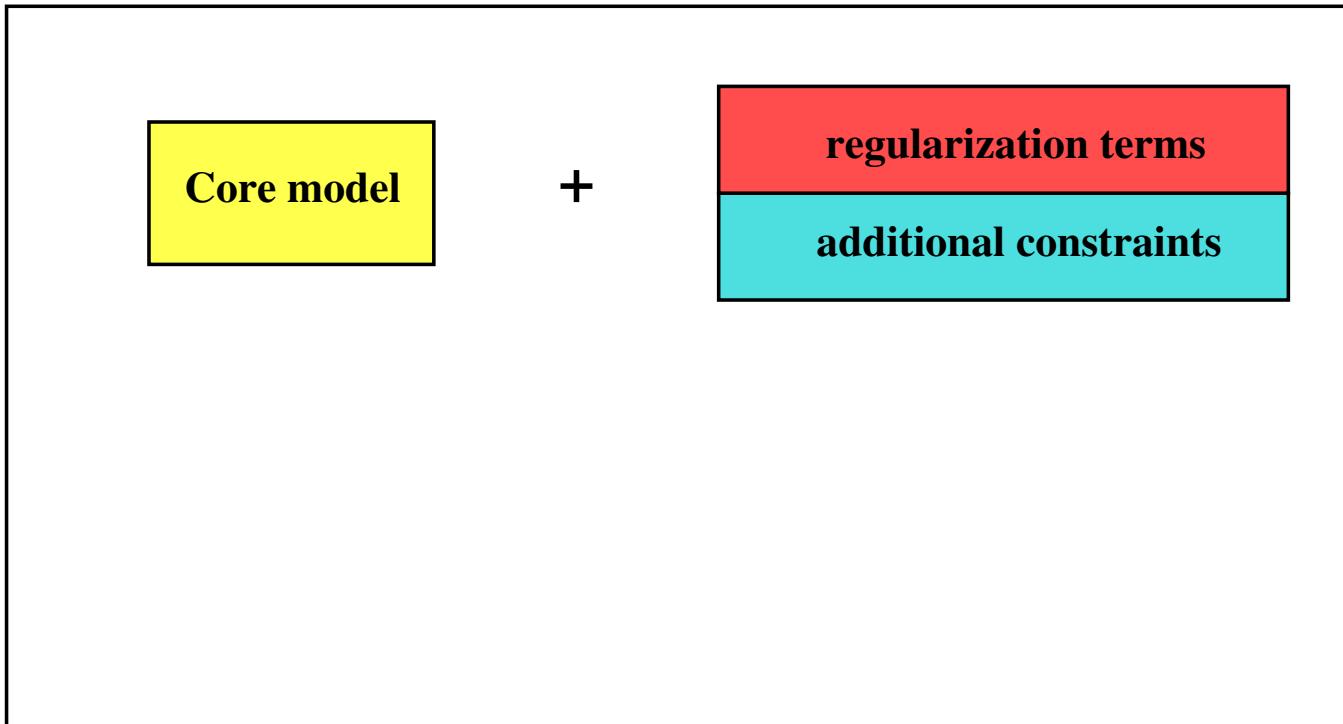
train + validation + test data

Example: image segmentation (looking for lines)

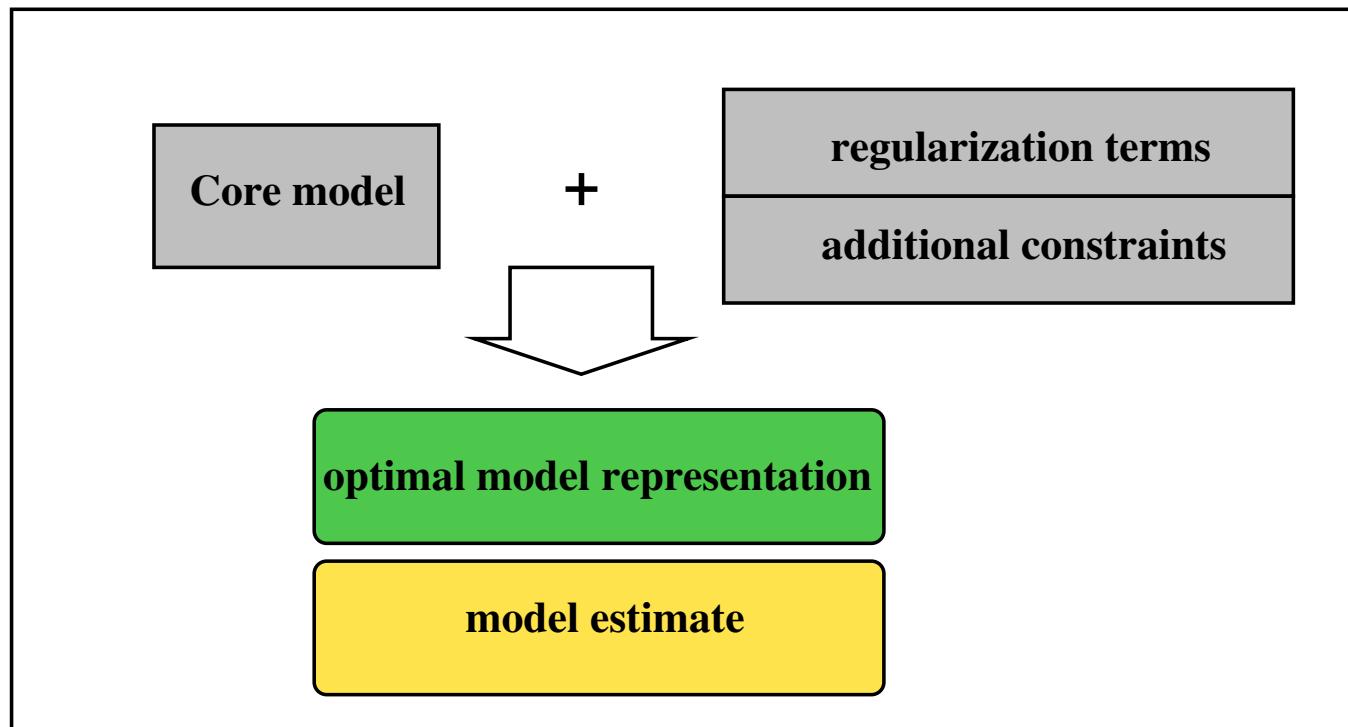




Core models + constraints



Core models + constraints



Kernel spectral clustering: adding prior knowledge

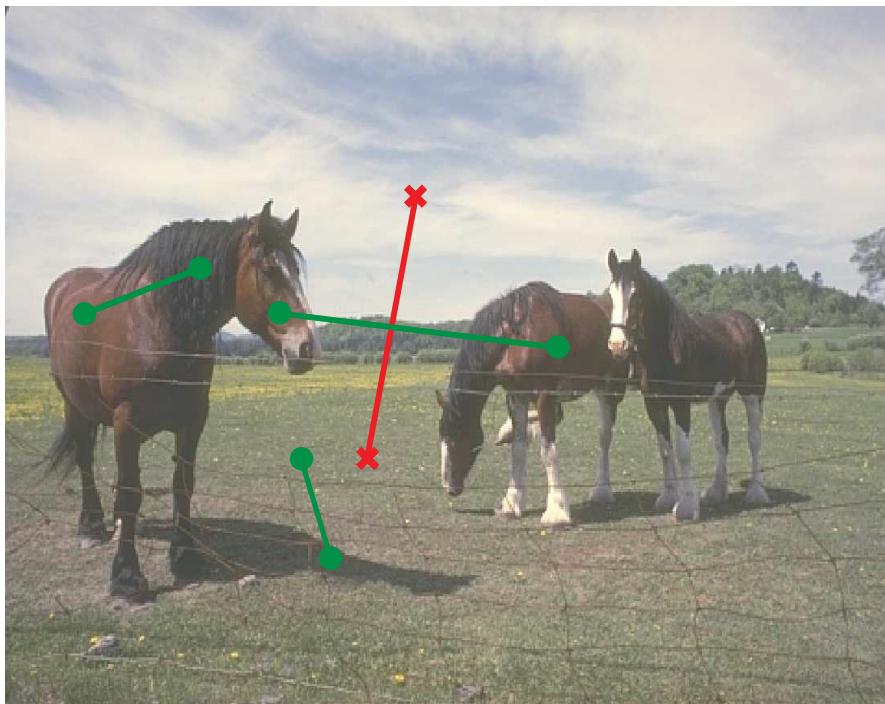
- Pair of points x_\dagger, x_\ddagger : $c = 1$ must-link, $c = -1$ cannot-link
- Primal problem [Alzate & Suykens, IJCNN 2009]

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

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

Adding prior knowledge

original image

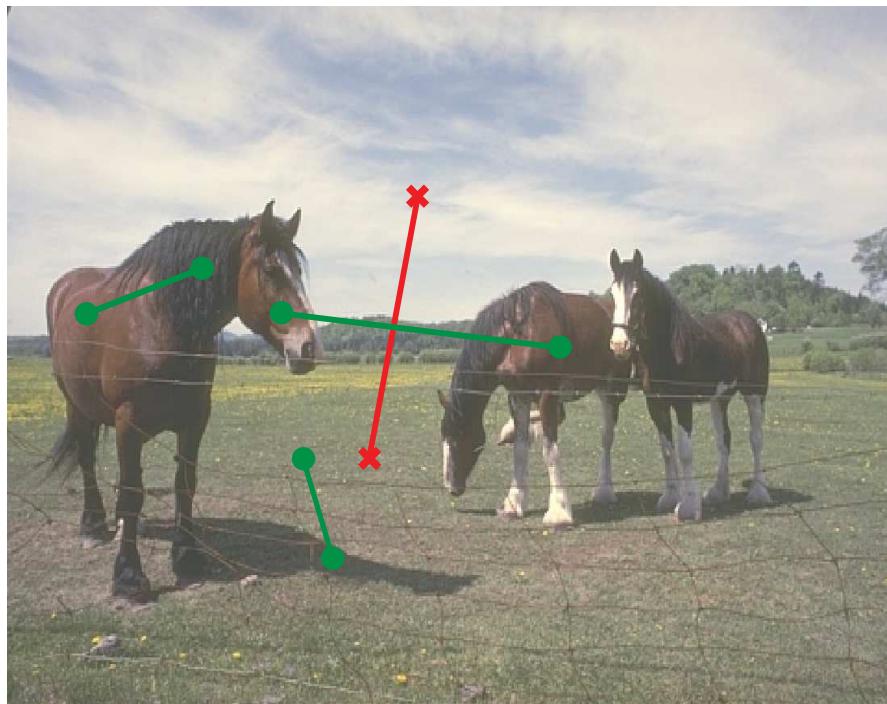


without constraints

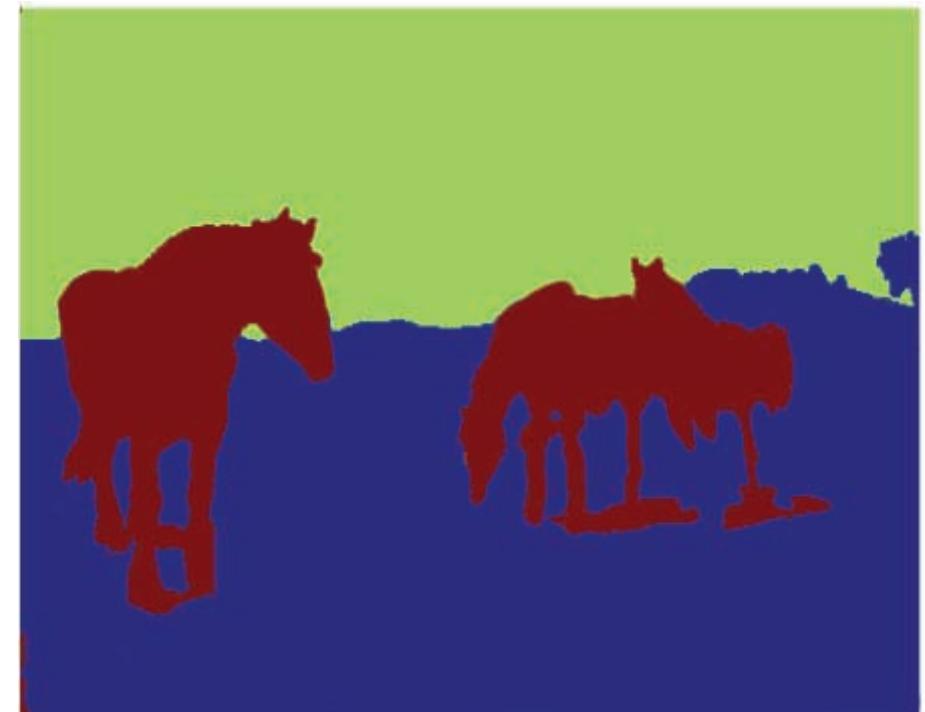


Adding prior knowledge

original image



with constraints



Semi-supervised learning using KSC (1)

- N unlabeled data, but **additional labels** on $M - N$ data
 $\mathcal{X} = \{x_1, \dots, x_N, \textcolor{red}{x_{N+1}}, \dots, x_M\}$
- Kernel spectral clustering as core model (binary case [Alzate & Suykens, WCCI 2012], multi-way/multi-class [Mehrkanon et al., TNNLS 2015])

$$\min_{w,e,b} \quad \frac{1}{2} w^T w - \gamma \frac{1}{2} e^T D^{-1} e + \rho \frac{1}{2} \sum_{m=N+1}^M (e_m - y_m)^2$$

subject to $e_i = w^T \varphi(x_i) + b, \quad i = 1, \dots, M$

Dual solution is characterized by a linear system. Suitable for clustering as well as classification.

- Other approaches in semi-supervised learning and manifold learning, e.g. [Belkin et al., 2006]

Semi-supervised learning using KSC (2)

Dataset	size	n_L/n_U	test (%)	FS semi-KSC	RD semi-KSC	Lap-SVMp
Spambase	4597	368/736	919 (20%)	0.885 ± 0.01	0.883 ± 0.01	0.880 ± 0.03
Satimage	6435	1030/1030	1287 (20%)	0.864 ± 0.006	0.831 ± 0.009	0.834 ± 0.007
Ring	7400	592/592	1480 (20%)	0.975 ± 0.005	0.974 ± 0.005	0.972 ± 0.006
Magic	19020	761/1522	3804 (20%)	0.836 ± 0.006	0.829 ± 0.006	0.827 ± 0.005
Cod-rna	331152	1325/1325	66230 (20%)	0.957 ± 0.006	0.947 ± 0.008	0.951 ± 0.001
Covertype	581012	2760/2760	29050 (5%)	0.715 ± 0.005	0.684 ± 0.008	0.697 ± 0.001
		2760/27600		0.729 ± 0.04	0.709 ± 0.05	—
		2760/82800		0.739 ± 0.04	0.716 ± 0.03	—
		2760/138000		0.742 ± 0.05	0.723 ± 0.06	—

FS semi-KSC: Fixed-size semi-supervised KSC

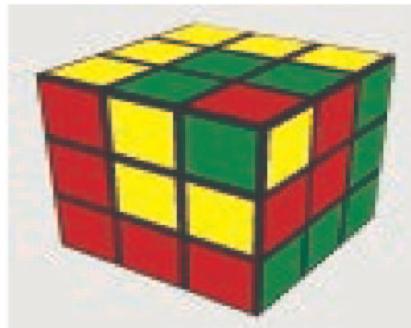
RD semi-KSC: other subset selection related to [Lee & Mangasarian, 2001]

Lap-SVM: Laplacian support vector machine [Belkin et al., 2006]

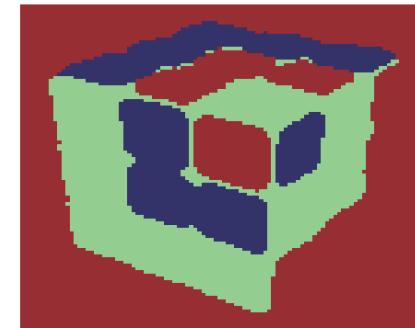
[Mehrkanoon & Suykens, 2014]

Semi-supervised learning using KSC (3)

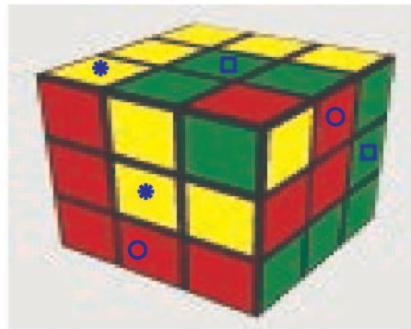
original image



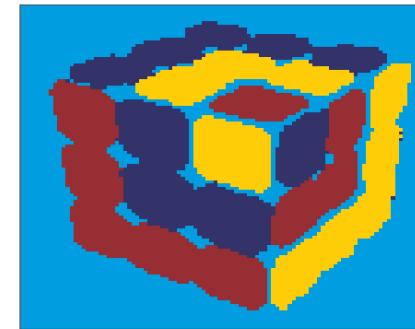
KSC



given a few labels

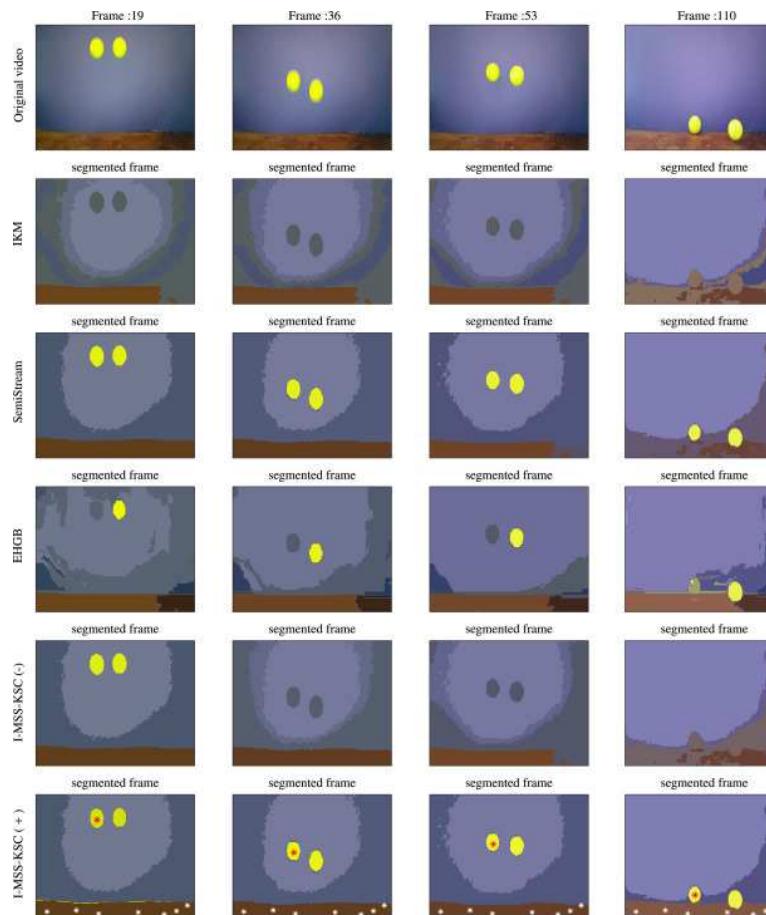


semi-supervised KSC



[Mehrkanon, Alzate, Mall, Langone, Suykens, IEEE-TNNLS, 2015]

Semi-supervised learning using KSC (4)



KSC regularized by Kalman filtering for on-line video segmentation:

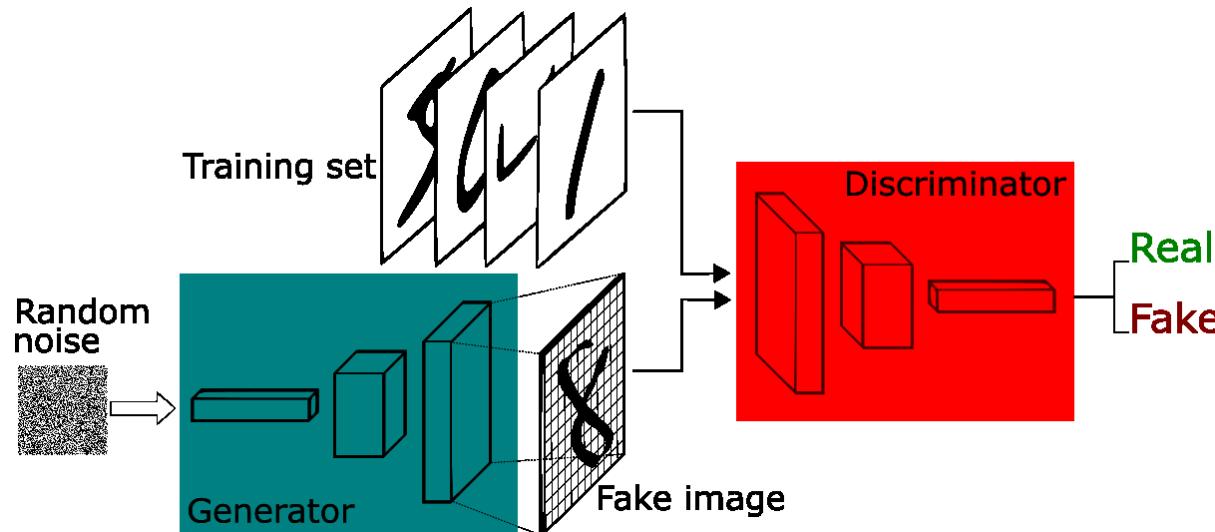
Mehrkanon S., Agudelo M., Suykens J.A.K, "Incremental multi-class semi-supervised clustering regularized by Kalman filtering", *Neural Networks*, vol. 71, Nov. 2015, pp. 88-104.

Generative models

Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN) [Goodfellow et al., 2014]
Training of two competing models in a zero-sum game:

- (Generator) generate fake output examples from random noise
- (Discriminator) discriminate between fake examples and real examples.



source: <https://deeplearning4j.org/generative-adversarial-network>

GAN: example on MNIST

MNIST training data:

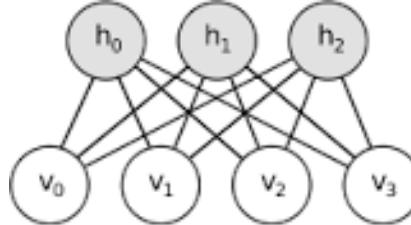
3 2 6 4
4 1 4 3
2 0 3 0
5 9 4 5

GAN generated examples:

3 9 5 3
9 6 4 2
0 5 7 8
8 7 3 2

source: <https://www.kdnuggets.com/2016/07/mnist-generative-adversarial-model-keras.html>

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:

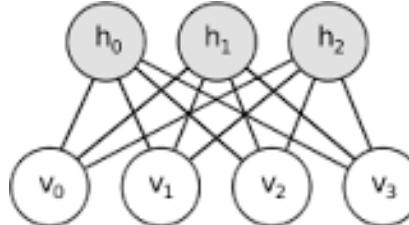
$$E(v, h; \theta) = -v^T Wh - c^T v - a^T h \quad \text{with } \theta = \{W, c, a\}$$

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]

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:

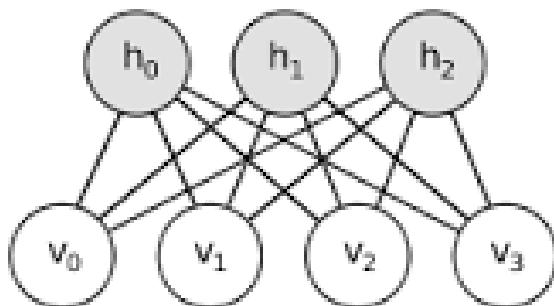
$$E(v, h; \theta) = -v^T Wh - c^T v - a^T h \quad \text{with } \theta = \{W, c, a\}$$

Joint distribution:

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

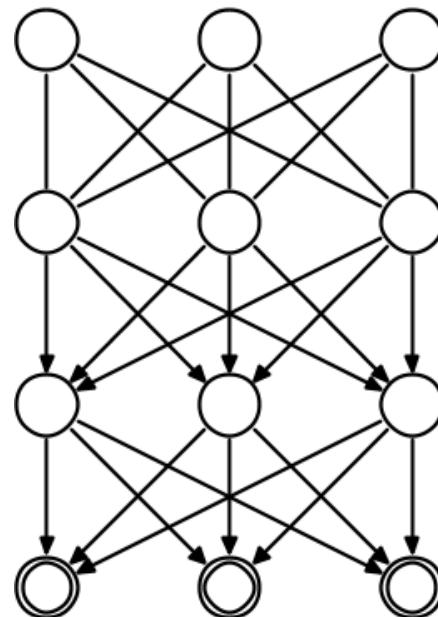
with partition function $Z(\theta) = \sum_v \sum_h \exp(-E(v, h; \theta))$
[Hinton, Osindero, Teh, Neural Computation 2006]

RBM and deep learning

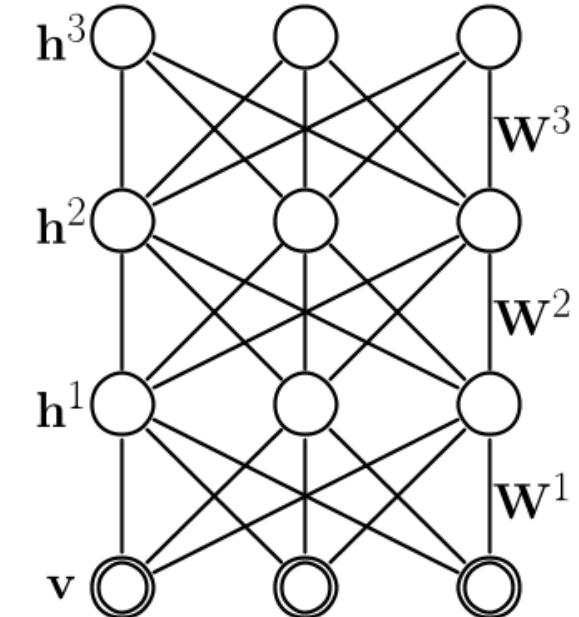


$$p(v, h)$$

Deep Belief Network



Deep Boltzmann Machine

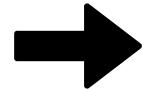
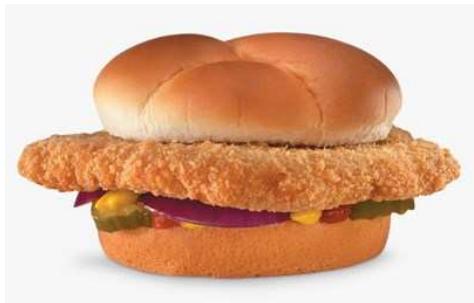


$$p(v, h^1, h^2, h^3, \dots)$$

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

in other words ...

"sandwich"



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

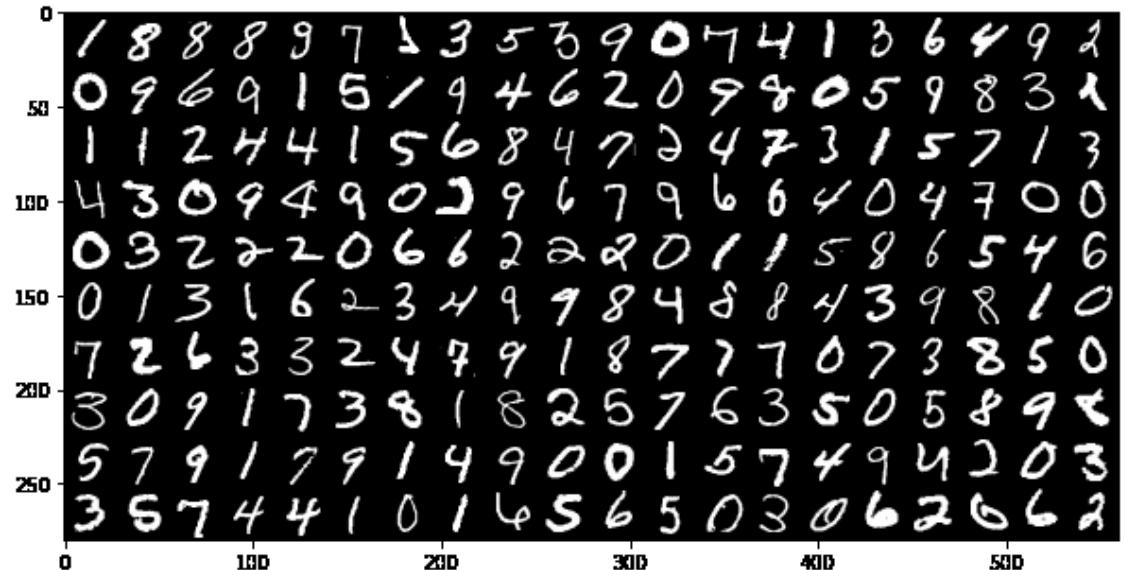
"deep sandwich"



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

RBM: example on MNIST

MNIST training data:

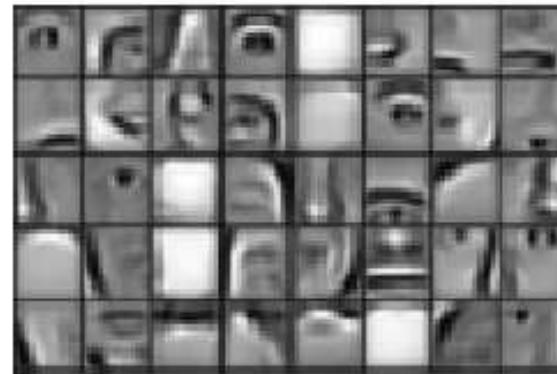
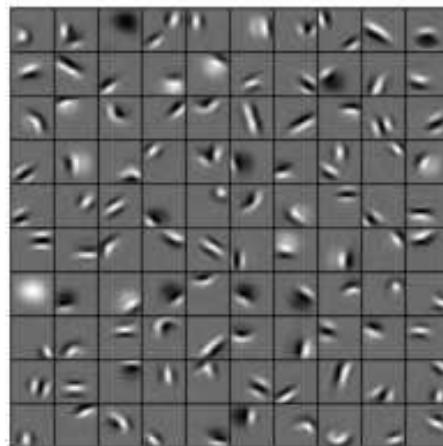


Generating new images:



source: <https://www.kaggle.com/nicw102168/restricted-boltzmann-machine-rbm-on-mnist>

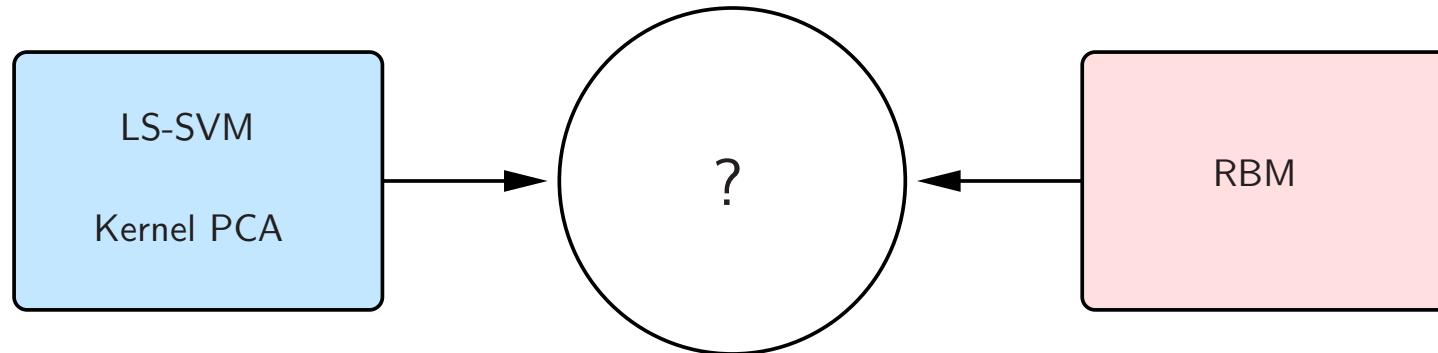
Convolutional Deep Belief Networks



Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks [Lee et al. 2011]

Restricted kernel machines

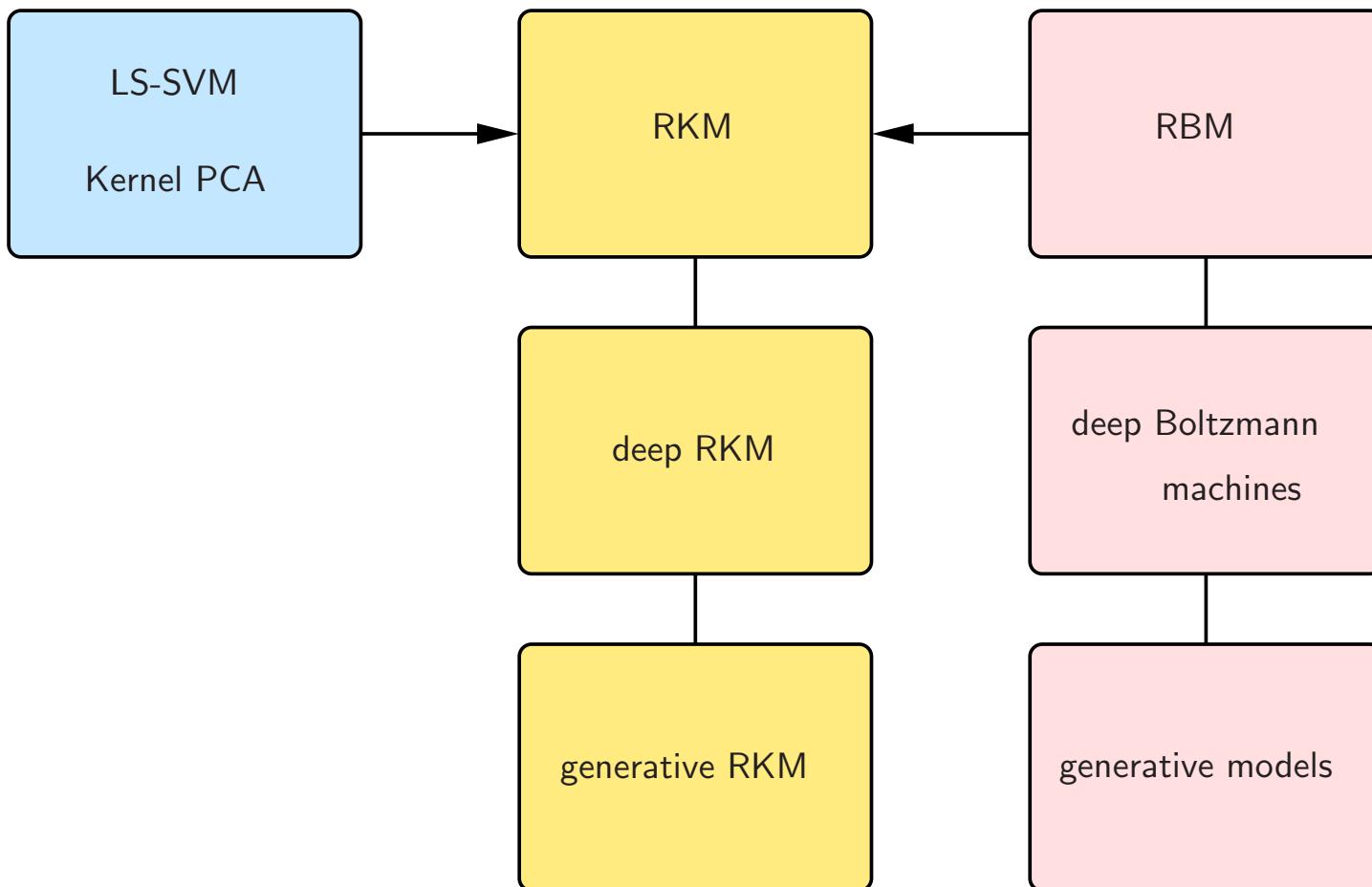
New connections



New connections



New connections

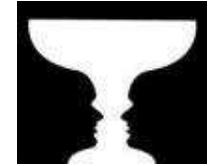


Restricted Kernel Machines (RKM)

- 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]

From KPCA to RKM representation (1)



Model:

$$e = W^T \varphi(x)$$

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

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

RKM representation:

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

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

From KPCA to RKM representation (2)

- Objective

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

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

$$\left\{
 \begin{array}{l}
 \frac{\partial \bar{J}}{\partial h_i} = 0 \Rightarrow W^T \varphi(x_i) = \lambda h_i, \quad \forall i \\
 \frac{\partial \bar{J}}{\partial W} = 0 \Rightarrow W = \frac{1}{\eta} \sum_i \varphi(x_i) h_i^T
 \end{array}
 \right.$$

From KPCA to RKM representation (3)

- Elimination of W gives the eigenvalue decomposition:

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

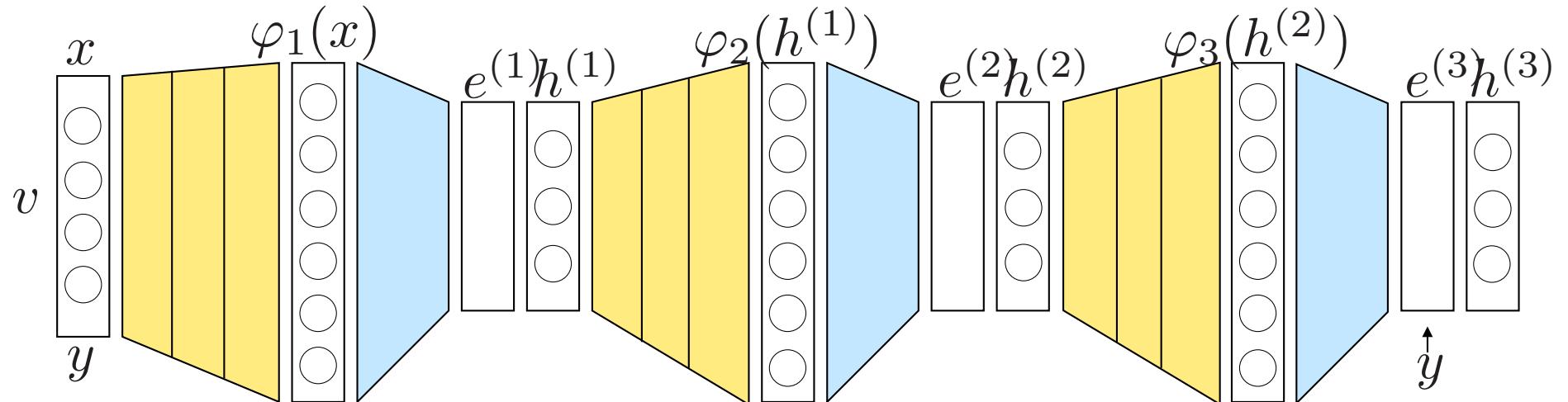
where $H = [h_1 \dots h_N] \in \mathbb{R}^{s \times N}$ and $\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_s\}$ with $s \leq N$

- Primal and dual model representations

$$\begin{array}{ccc} & (P)_{\text{RKM}} : \hat{e} = W^T \varphi(x) & \\ \nearrow & & \\ \mathcal{M} & & \\ \searrow & & \\ & (D)_{\text{RKM}} : \hat{e} = \frac{1}{\eta} \sum_j h_j K(x_j, x). & \end{array}$$

Deep Restricted Kernel Machines

Deep RKM: example



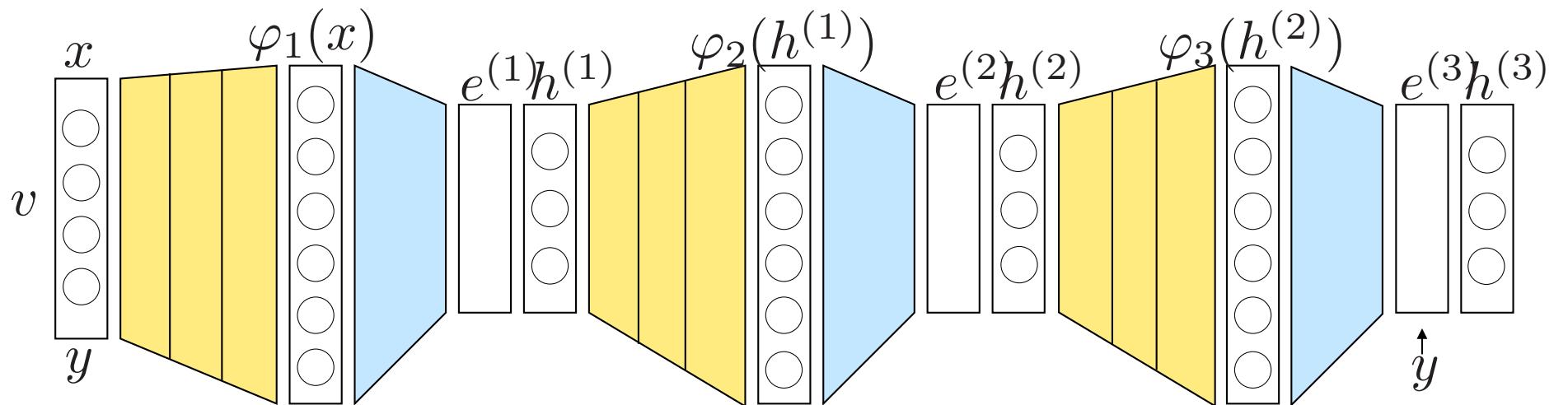
Deep RKM: KPCA + KPCA + LSSVM [Suykens, 2017]

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 ...



$$\begin{aligned}
 J_{\text{deep}} = & - \sum_{i=1}^N \varphi_1(x_i)^T \mathbf{W}_1 h_i^{(1)} + \frac{\lambda_1}{2} \sum_{i=1}^N h_i^{(1)T} h_i^{(1)} + \frac{\eta_1}{2} \text{Tr}(W_1^T W_1) \\
 & - \sum_{i=1}^N \varphi_2(h_i^{(1)})^T \mathbf{W}_2 h_i^{(2)} + \frac{\lambda_2}{2} \sum_{i=1}^N h_i^{(2)T} h_i^{(2)} + \frac{\eta_2}{2} \text{Tr}(W_2^T W_2) \\
 & + \sum_{i=1}^N (y_i^T - \varphi_3(h_i^{(2)})^T \mathbf{W}_3 - b^T) h_i^{(3)} - \frac{\lambda_3}{2} \sum_{i=1}^N h_i^{(3)T} h_i^{(3)} + \frac{\eta_3}{2} \text{Tr}(W_3^T W_3)
 \end{aligned}$$

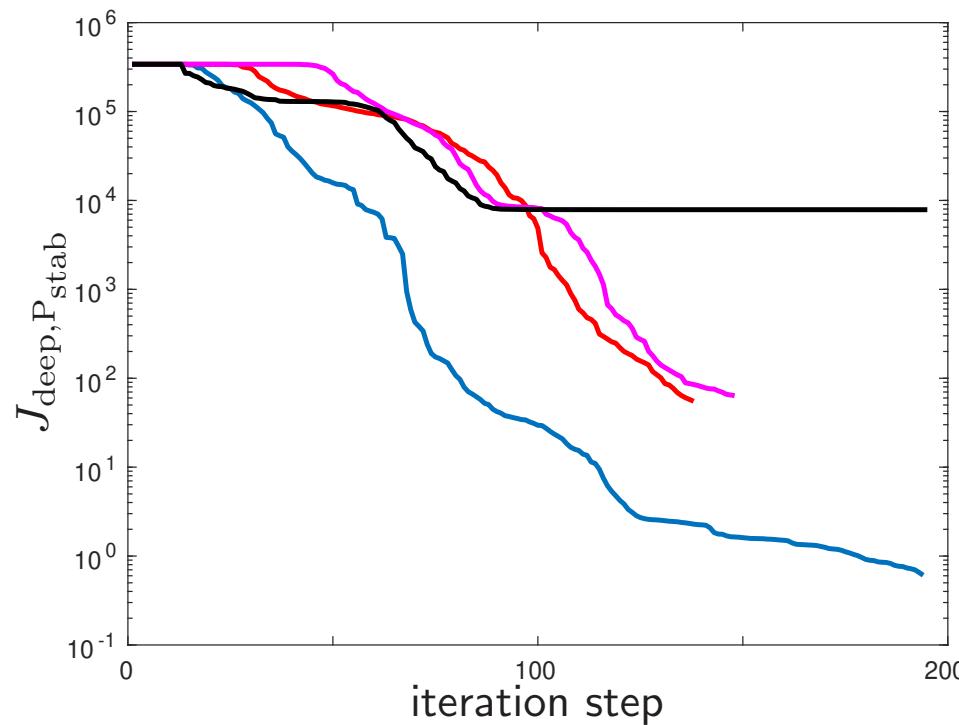
Primal and dual model representations

$$\begin{array}{ccc} & \hat{e}^{(1)} = W_1^T \varphi_1(x) \\ (P)_{\text{DeepRKM}} : & \hat{e}^{(2)} = W_2^T \varphi_2(\Lambda_1^{-1} \hat{e}^{(1)}) \\ \nearrow & \hat{y} = W_3^T \varphi_3(\Lambda_2^{-1} \hat{e}^{(2)}) + b \\ \mathcal{M} & \\ \searrow & \hat{e}^{(1)} = \frac{1}{\eta_1} \sum_j h_j^{(1)} K_1(x_j, x) \\ (D)_{\text{DeepRKM}} : & \hat{e}^{(2)} = \frac{1}{\eta_2} \sum_j h_j^{(2)} K_2(h_j^{(1)}, \Lambda_1^{-1} \hat{e}^{(1)}) \\ & \hat{y} = \frac{1}{\eta_3} \sum_j h_j^{(3)} K_3(h_j^{(2)}, \Lambda_2^{-1} \hat{e}^{(2)}) + b \end{array}$$

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

Generative RKM

RKM objective for training and generating

- RBM energy function

$$E(v, h; \theta) = -v^T Wh - c^T v - a^T h$$

with model parameters $\theta = \{W, c, a\}$

- RKM "super-objective" function (for training and for generating)

$$\bar{J}(v, h, W) = -v^T Wh + \frac{\lambda}{2} h^T h + \frac{1}{2} v^T v + \frac{\eta}{2} \text{Tr}(W^T W)$$

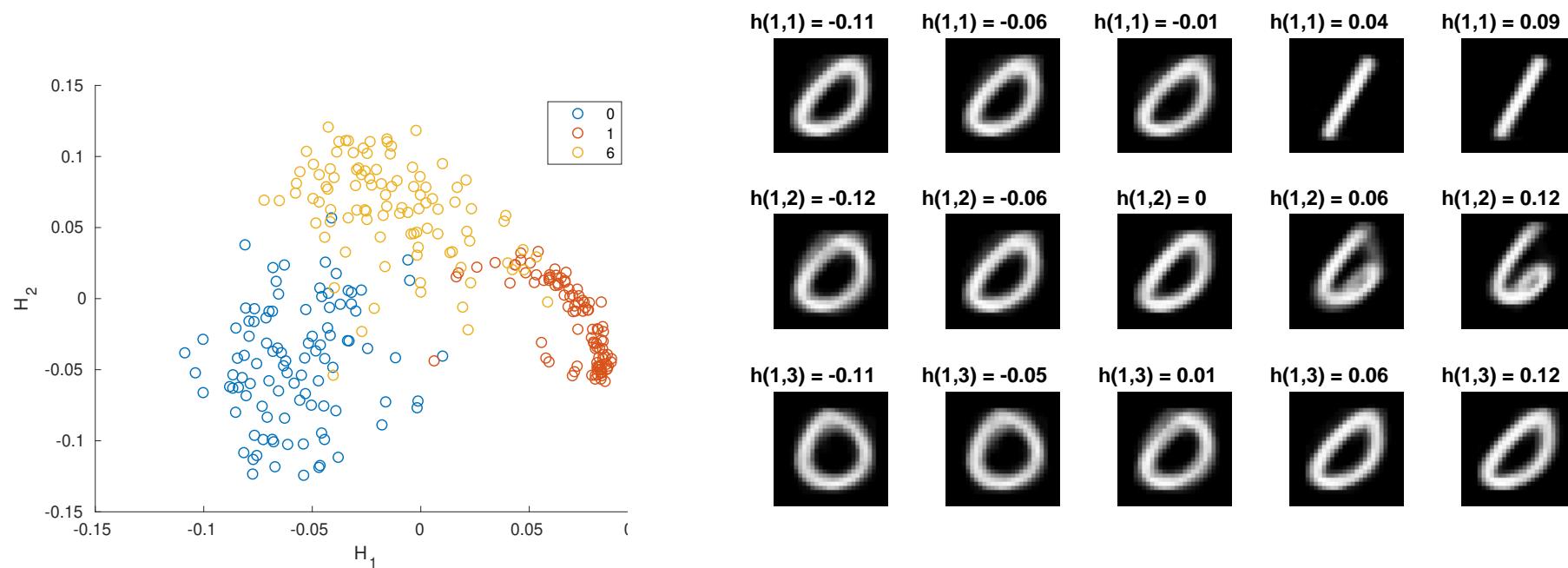
Training: clamp $v \rightarrow \bar{J}_{\text{train}}(h, W)$

Generating: clamp $h, W \rightarrow \bar{J}_{\text{gen}}(v)$

[Schreurs & Suykens, ESANN 2018]

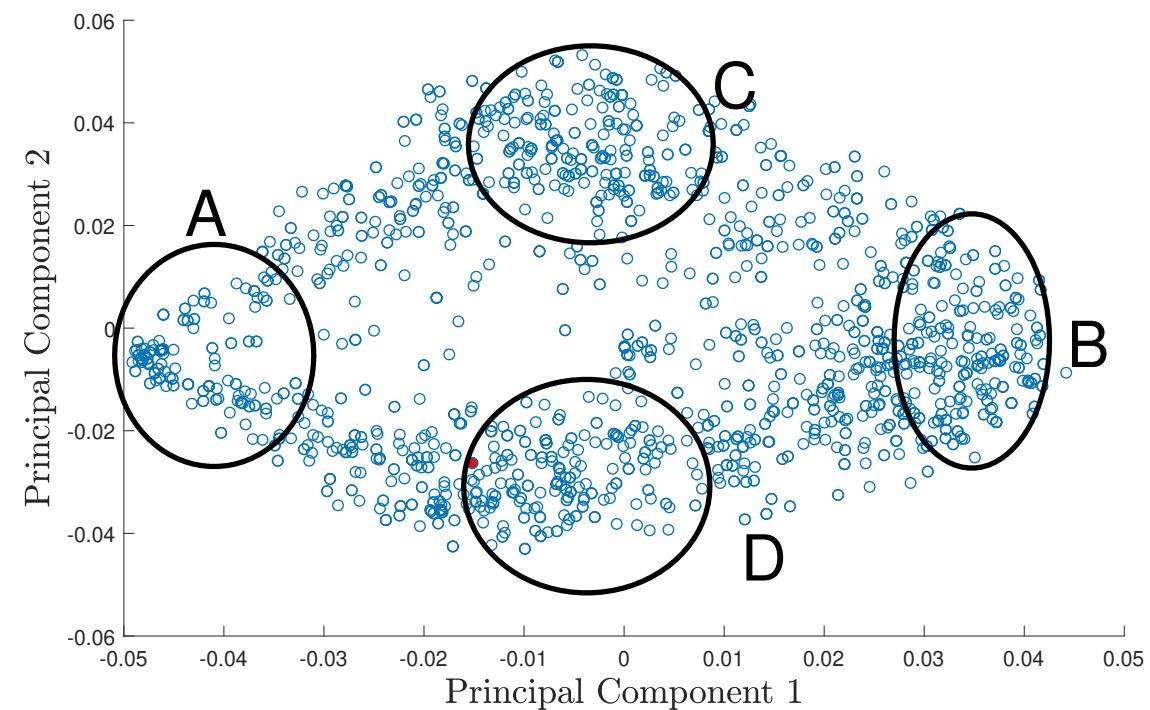
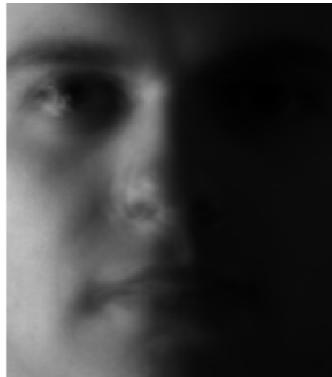
Explainable AI: latent space exploration (1)

hidden units: exploring the **whole continuum**:



[figures by Joachim Schreurs]

Explainable AI: latent space exploration (2)

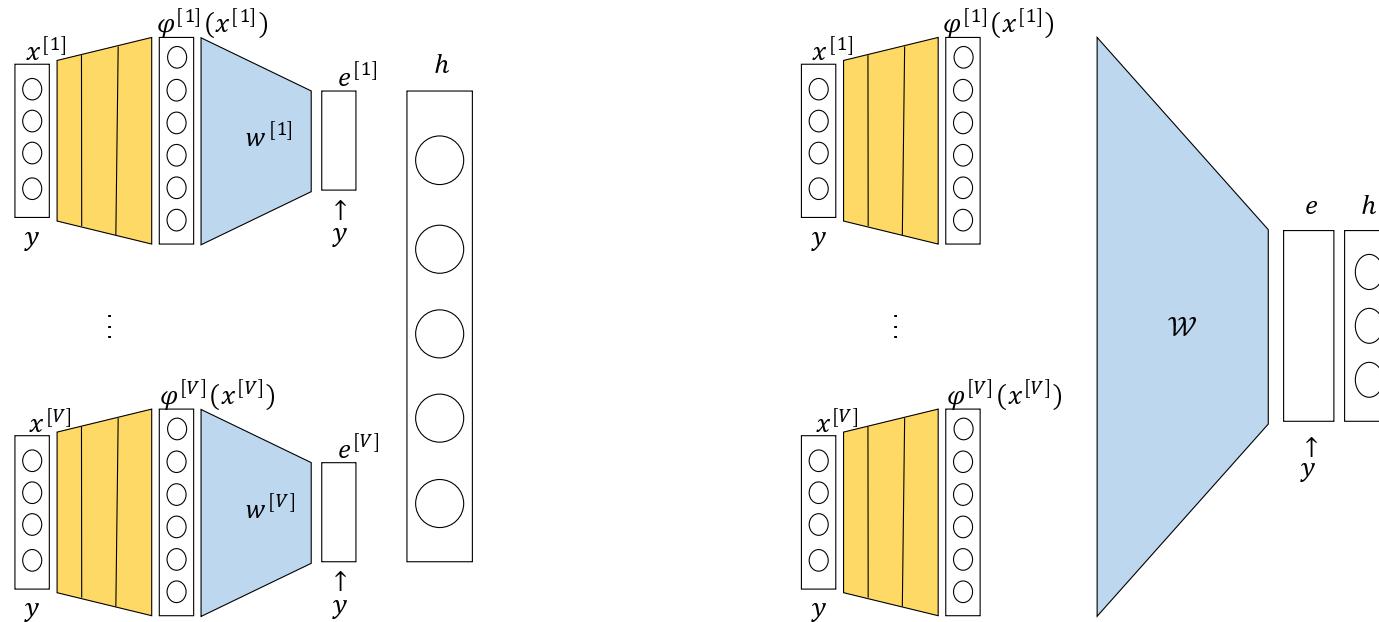


Yale Face database - generated faces from different regions A,B,C,D

[Winant, Schreurs, Suykens, BNAIC 2019]

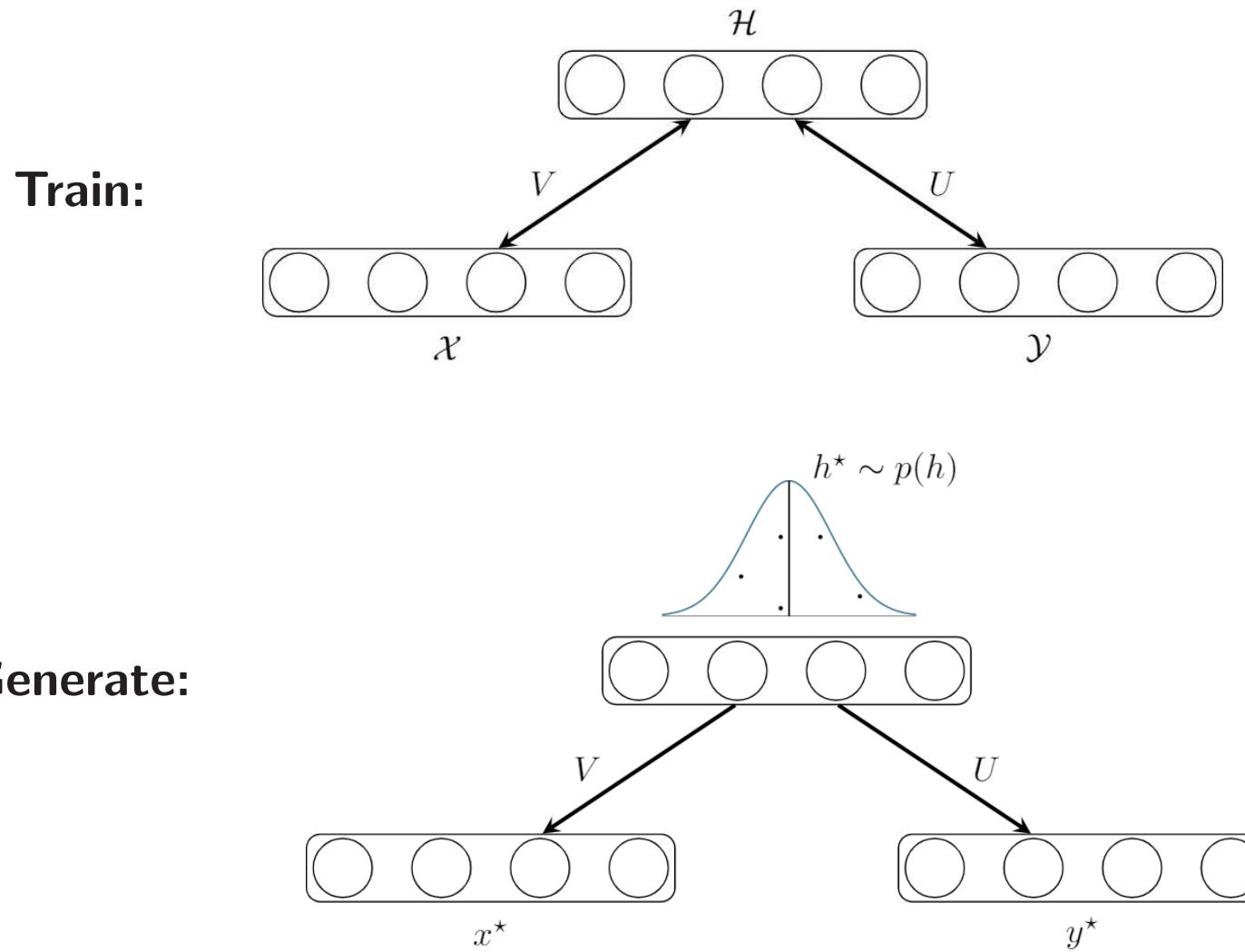
Tensor-based RKM for Multi-view KPCA

$$\min \langle \mathcal{W}, \mathcal{W} \rangle - \sum_{i=1}^N \langle \Phi_{(i)}, \mathcal{W} \rangle h_i + \lambda \sum_{i=1}^N h_i^2 \quad \text{with} \quad \Phi_{(i)} = \varphi^{[1]}(x_i^{[1]}) \otimes \dots \otimes \varphi^{[V]}(x_i^{[V]})$$



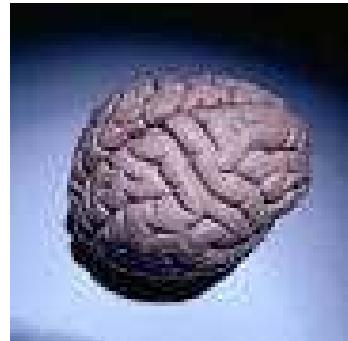
[Houthuys & Suykens, ICANN 2018]

Generative RKM (Gen-RKM) (1)

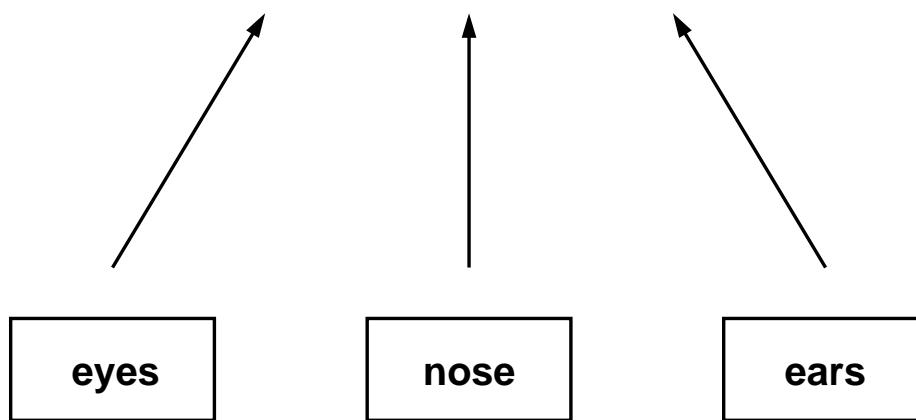


[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Analogy with the human brain



shared feature representation



multimodal data

Gen-RKM (2)

The objective

$$\begin{aligned} J_{\text{train}}(\mathbf{h}_i, \mathbf{U}, \mathbf{V}) = & \sum_{i=1}^N (-\phi_1(\mathbf{x}_i)^T \mathbf{U} \mathbf{h}_i - \phi_2(\mathbf{y}_i)^T \mathbf{V} \mathbf{h}_i + \frac{\lambda}{2} \mathbf{h}_i^T \mathbf{h}_i) \\ & + \frac{\eta_1}{2} \text{Tr}(\mathbf{U}^T \mathbf{U}) + \frac{\eta_2}{2} \text{Tr}(\mathbf{V}^T \mathbf{V}) \end{aligned}$$

results for **training** into the eigenvalue problem

$$(\frac{1}{\eta_1} \mathbf{K}_1 + \frac{1}{\eta_2} \mathbf{K}_2) \mathbf{H}^T = \mathbf{H}^T \boldsymbol{\Lambda}$$

with $\mathbf{H} = [\mathbf{h}_1 \dots \mathbf{h}_N]$ and kernel matrices $\mathbf{K}_1, \mathbf{K}_2$ related to ϕ_1, ϕ_2 .

[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Gen-RKM (3)

Generating data is based on a newly generated \mathbf{h}^* and the objective

$$J_{\text{gen}}(\phi_1(\mathbf{x}^*), \varphi_2(\mathbf{y}^*)) = -\phi_1(\mathbf{x}^*)^T \mathbf{V} \mathbf{h}^* - \phi_2(\mathbf{y}^*)^T \mathbf{U} \mathbf{h}^* + \frac{1}{2} \phi_1(\mathbf{x}^*)^T \phi_1(\mathbf{x}^*) + \frac{1}{2} \phi_2(\mathbf{y}^*)^T \phi_2(\mathbf{y}^*)$$

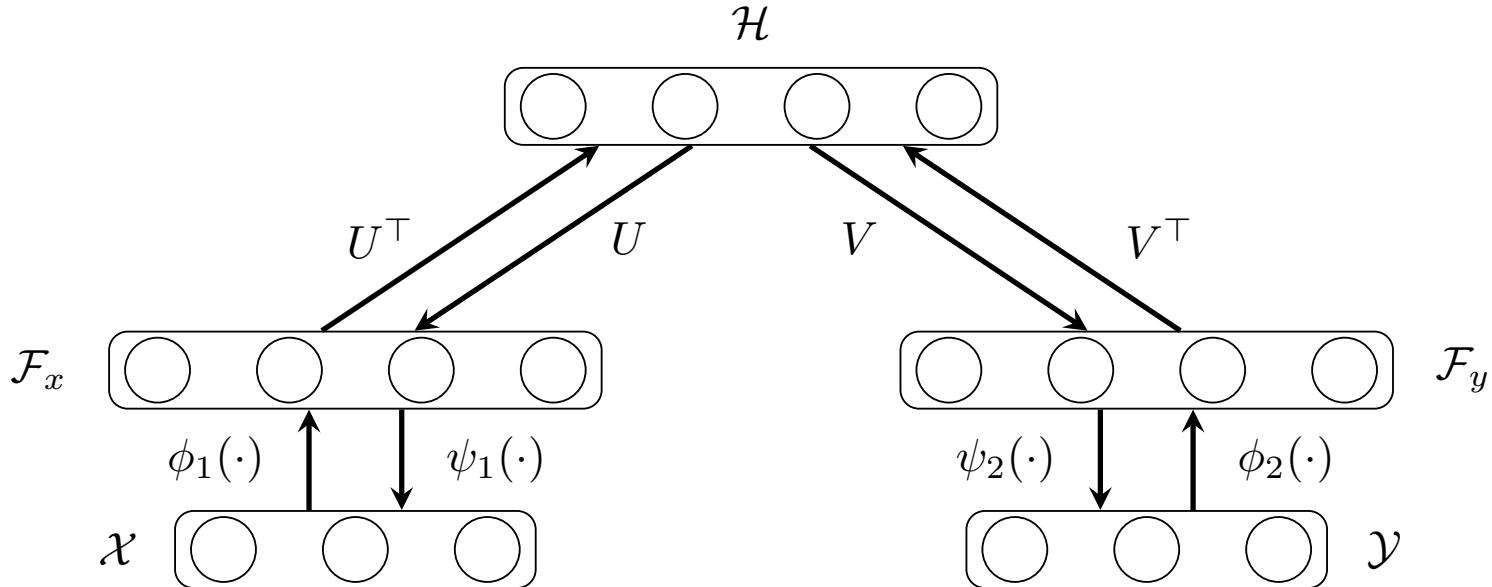
giving

$$\phi_1(\mathbf{x}^*) = \frac{1}{\eta_1} \sum_{i=1}^N \phi_1(\mathbf{x}_i) \mathbf{h}_i^T \mathbf{h}^*, \quad \phi_2(\mathbf{y}^*) = \frac{1}{\eta_2} \sum_{i=1}^N \phi_2(\mathbf{y}_i) \mathbf{h}_i^T \mathbf{h}^*.$$

For generating $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ one can either work with the kernel smoother or work with an explicit feature map using a (deep) neural network or CNN.

[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Gen-RKM (4)



Gen-RKM schematic representation modeling a common subspace \mathcal{H} between two data sources \mathcal{X} and \mathcal{Y} . The ϕ_1, ϕ_2 are the feature maps (\mathcal{F}_x and \mathcal{F}_y represent the feature-spaces) corresponding to the two data sources. While ψ_1, ψ_2 represent the pre-image maps. The interconnection matrices U, V model dependencies between latent variables and the mapped data sources.

[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Gen-RKM: implicit feature map (choose kernel)

Obtain

$$k_{\mathbf{x}^*} = \frac{1}{\eta_1} \mathbf{K}_1 \mathbf{H}^\top \mathbf{h}^*, \quad k_{\mathbf{y}^*} = \frac{1}{\eta_2} \mathbf{K}_2 \mathbf{H}^\top \mathbf{h}^*,$$

with $\mathbf{k}_{\mathbf{x}^*} = [k(\mathbf{x}_1, \mathbf{x}^*), \dots, k(\mathbf{x}_N, \mathbf{x}^*)]^\top$.

Using the kernel-smoother:

$$\hat{\mathbf{x}} = \psi_1(\phi_1(\mathbf{x}^*)) = \frac{\sum_{j=1}^{n_r} \tilde{k}_1(\mathbf{x}_j, \mathbf{x}^*) \mathbf{x}_j}{\sum_{j=1}^{n_r} \tilde{k}_1(\mathbf{x}_j, \mathbf{x}^*)}, \quad \hat{\mathbf{y}} = \psi_2(\phi_2(\mathbf{y}^*)) = \frac{\sum_{j=1}^{n_r} \tilde{k}_2(\mathbf{y}_j, \mathbf{y}^*) \mathbf{y}_j}{\sum_{j=1}^{n_r} \tilde{k}_2(\mathbf{y}_j, \mathbf{y}^*)},$$

with $\tilde{k}_1(\mathbf{x}_i, \mathbf{x}^*)$ and $\tilde{k}_2(\mathbf{y}_i, \mathbf{y}^*)$ the scaled similarities between 0 and 1.

n_r is the number of closest points based on the similarity defined by kernels \tilde{k}_1 and \tilde{k}_2 .

Gen-RKM: choose explicit (CNN) feature map

Parametrized feature maps: $\phi_{\boldsymbol{\theta}}(\cdot)$, $\psi_{\boldsymbol{\zeta}}(\cdot)$ (e.g. CNN and transposed CNN).

Overall objective function, using a stabilization mechanism [Suykens, 2017]:

$$\begin{aligned} \min_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2} \mathcal{J}_c &= \mathcal{J}_{\text{train}} + \frac{c_{\text{stab}}}{2} \mathcal{J}_{\text{train}}^2 \\ &\quad + \frac{c_{\text{acc}}}{2N} \left(\sum_{i=1}^N \left[\mathcal{L}_1(\mathbf{x}_i^*, \psi_{1\boldsymbol{\zeta}_1}(\phi_{1\boldsymbol{\theta}_1}(\mathbf{x}_i^*))) + \mathcal{L}_2(\mathbf{y}_i^*, \psi_{2\boldsymbol{\zeta}_2}(\phi_{2\boldsymbol{\theta}_2}(\mathbf{y}_i^*))) \right] \right) \end{aligned}$$

with reconstruction errors

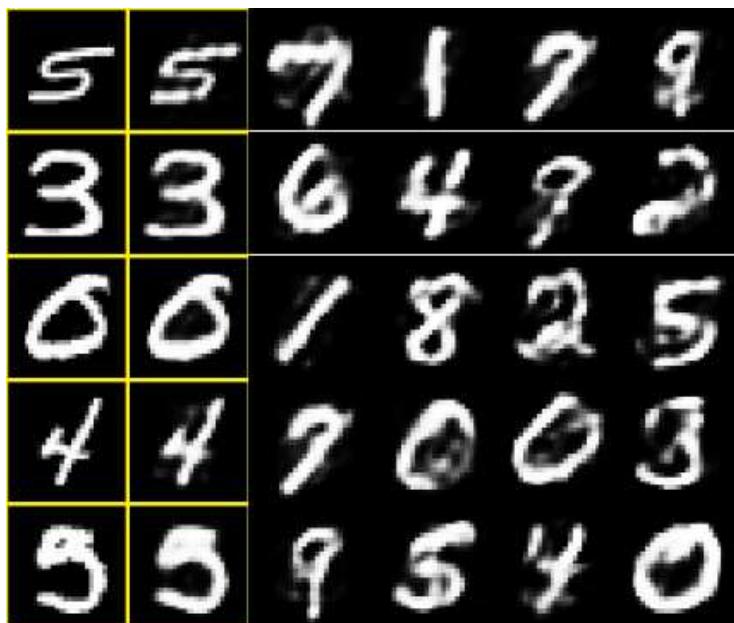
$$\begin{aligned} \mathcal{L}_1(\mathbf{x}_i^*, \psi_{1\boldsymbol{\zeta}_1}(\phi_{1\boldsymbol{\theta}_1}(\mathbf{x}_i^*))) &= \frac{1}{N} \|\mathbf{x}_i^* - \psi_{1\boldsymbol{\zeta}_1}(\phi_{1\boldsymbol{\theta}_1}(\mathbf{x}_i^*))\|_2^2, \\ \mathcal{L}_2(\mathbf{y}_i^*, \psi_{2\boldsymbol{\zeta}_2}(\phi_{2\boldsymbol{\theta}_2}(\mathbf{y}_i^*))) &= \frac{1}{N} \|\mathbf{y}_i^* - \psi_{2\boldsymbol{\zeta}_2}(\phi_{2\boldsymbol{\theta}_2}(\mathbf{y}_i^*))\|_2^2 \end{aligned}$$

and with $\Phi_{\mathbf{x}} = [\phi_1(\mathbf{x}_1), \dots, \phi_1(\mathbf{x}_N)]$, $\Phi_{\mathbf{y}} = [\phi_2(\mathbf{y}_1), \dots, \phi_2(\mathbf{y}_N)]$, U, V from

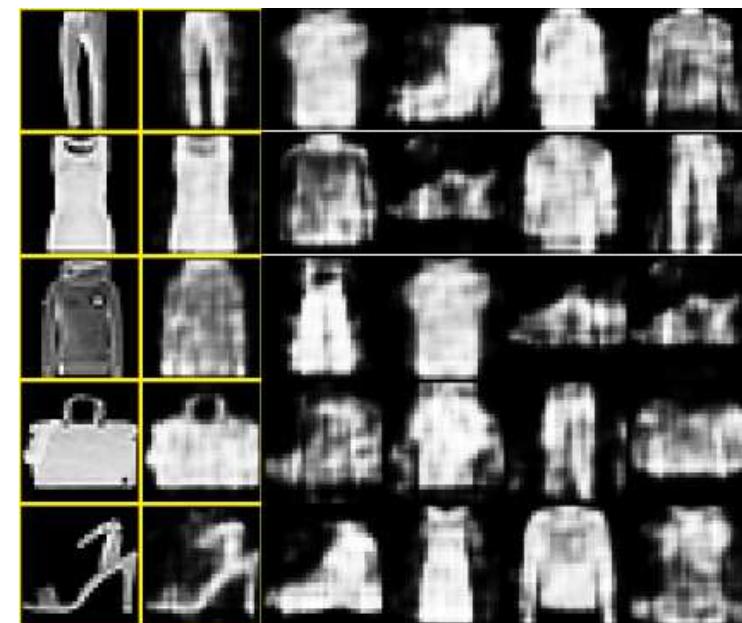
$$\begin{bmatrix} \frac{1}{\eta_1} \Phi_{\mathbf{x}} \Phi_{\mathbf{x}}^\top & \frac{1}{\eta_1} \Phi_{\mathbf{x}} \Phi_{\mathbf{y}}^\top \\ \frac{1}{\eta_2} \Phi_{\mathbf{y}} \Phi_{\mathbf{x}}^\top & \frac{1}{\eta_2} \Phi_{\mathbf{y}} \Phi_{\mathbf{y}}^\top \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \Lambda.$$

Hence, joint feature learning and subspace learning.

Gen-RKM: examples (1)



MNIST

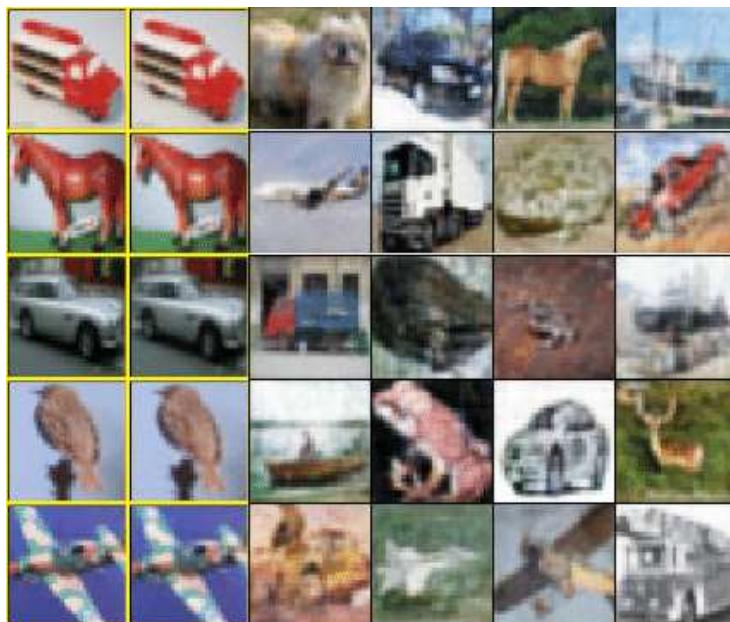


Fashion-MNIST

Generated samples from the model using CNN as explicit feature map in the kernel function. The yellow boxes show training examples and the adjacent boxes show the reconstructed samples. The other images (columns 3-6) are generated by random sampling from the fitted distribution over the learned latent variables.

[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Gen-RKM: examples (2)



CIFAR-10



CelebA

Generated samples from the model using CNN as explicit feature map in the kernel function. The yellow boxes show training examples and the adjacent boxes show the reconstructed samples. The other images (columns 3-6) are generated by random sampling from the fitted distribution over the learned latent variables.

[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Gen-RKM: multi-view generation (1)

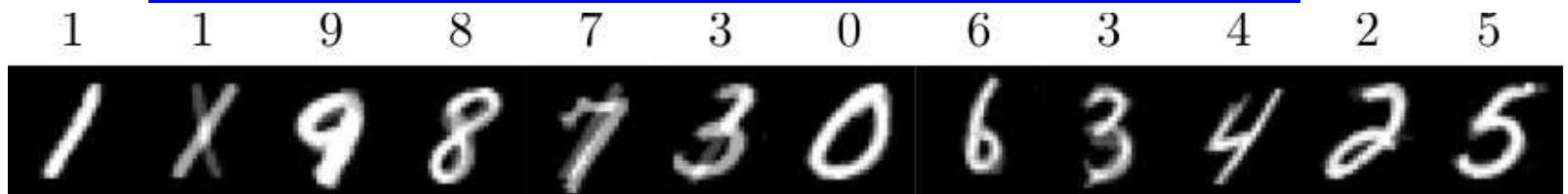


CelebA

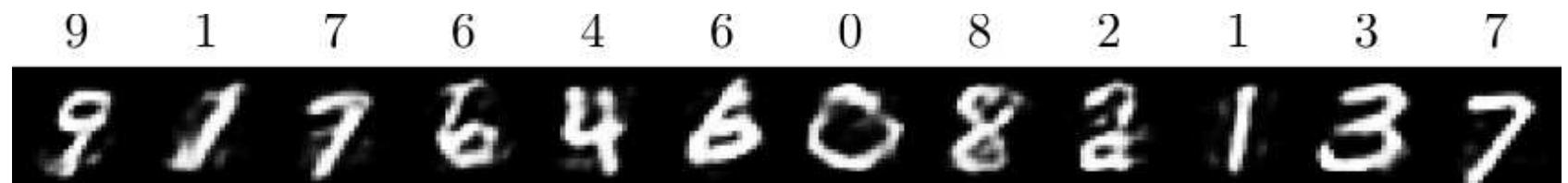
Multi-view generation on CelebA dataset showing images and attributes

[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Gen-RKM: multi-view generation (2)



MNIST: Implicit feature maps with Gaussian kernel + generation by kernel-smoother



MNIST: Explicit feature maps using Convolutional Neural Networks

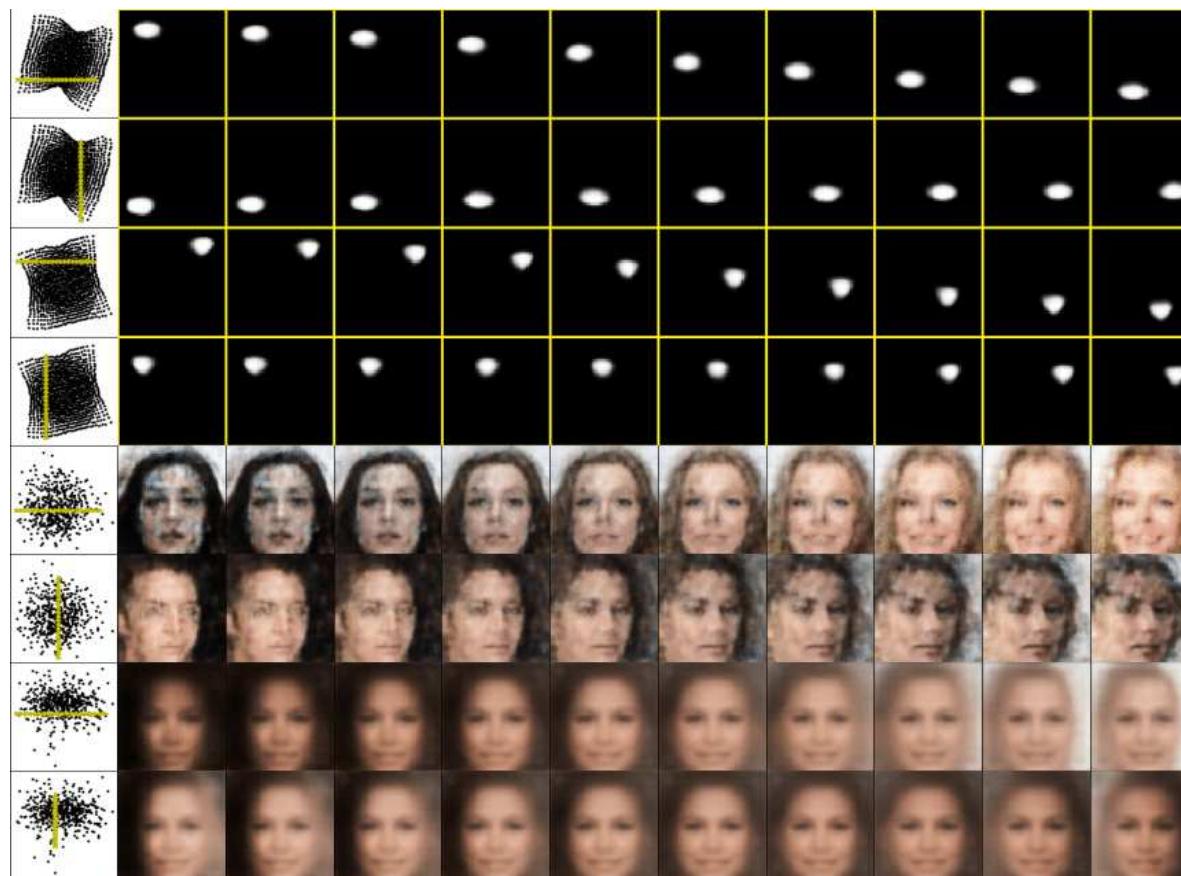


CIFAR-10: Explicit feature maps using CNNs + Transposed CNNs

Multi-view Generation (images and labels) using implicit and explicit feature maps

[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

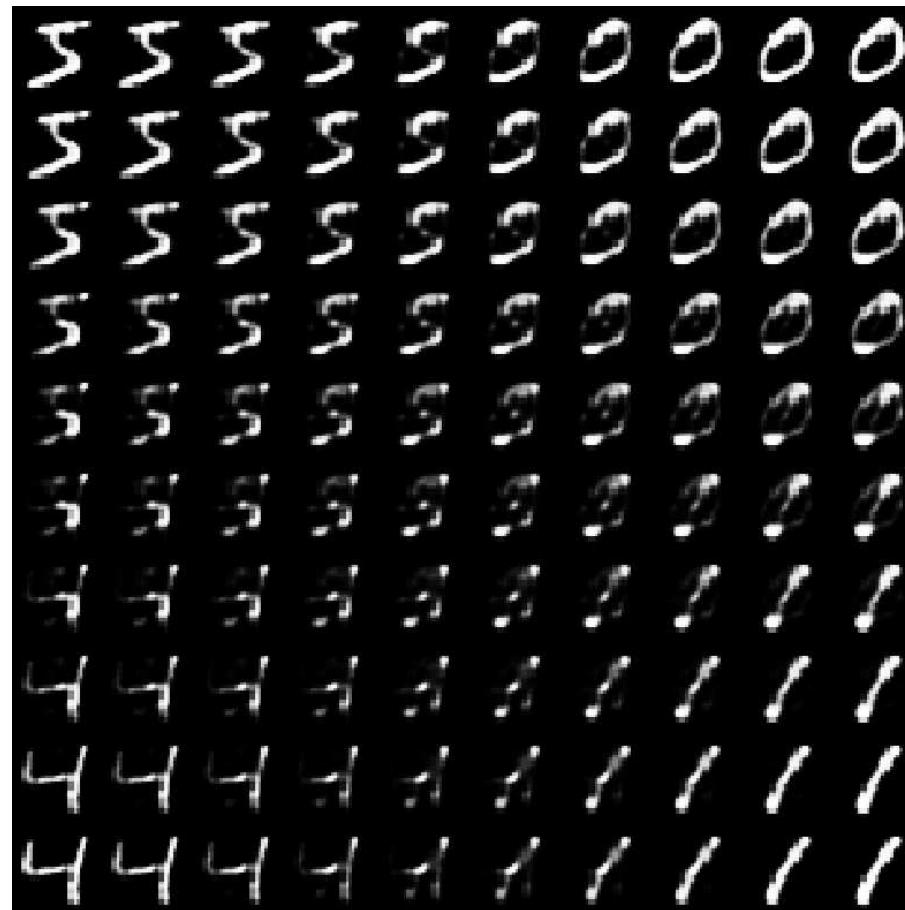
Gen-RKM: latent space exploration (1)



Exploring the learned **uncorrelated-features** by traversing along the eigenvectors

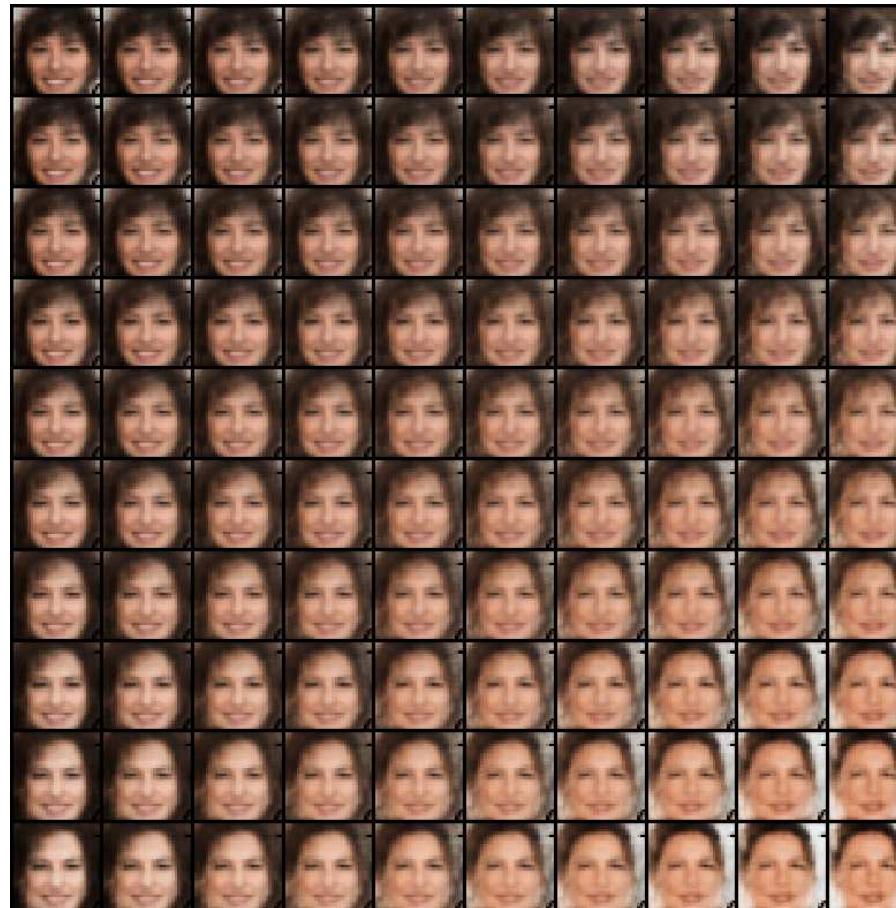
Explainability: changing one single neuron's hidden feature changes the hair color while preserving face structure! [Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Gen-RKM: latent space exploration (2)



MNIST reconstructed images by bilinear-interpolation in latent space
[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

Gen-RKM: latent space exploration (3)



CelebA reconstructed images by bilinear-interpolation in latent space
[Pandey, Schreurs & Suykens, 2019, arXiv:1906.08144]

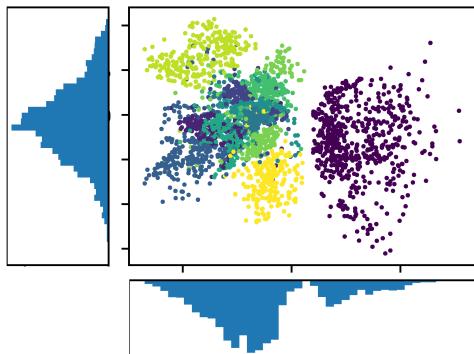
Gen-RKM - traversals along principal components



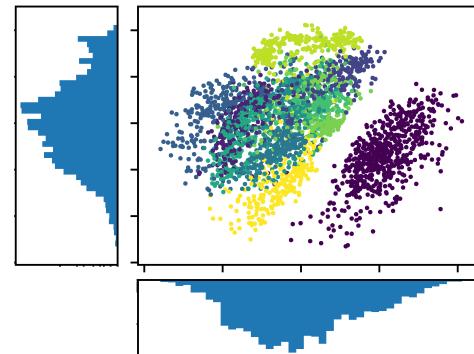
Disentangled Representation Learning and Generation with Manifold Optimization

[Pandey, Schreurs & Suykens, 2019, arXiv:2006.07046]

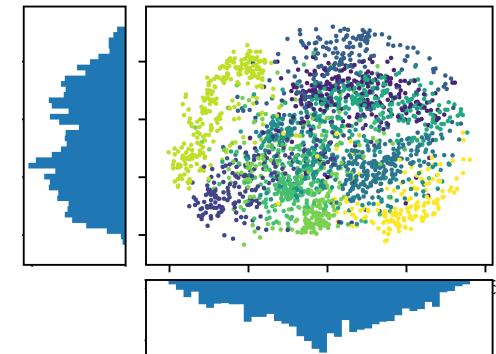
Robust Gen-RKM



VAE



Gen-RKM



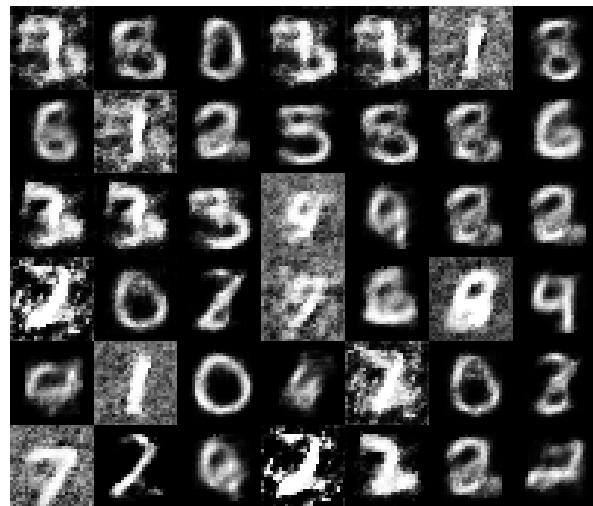
Robust Gen-RKM

- VAE and Gen-RKM: presence of outliers distorts the distribution of the latent variables
- Robust Gen-RKM: down-weighting of the outliers makes close to Gaussian distribution

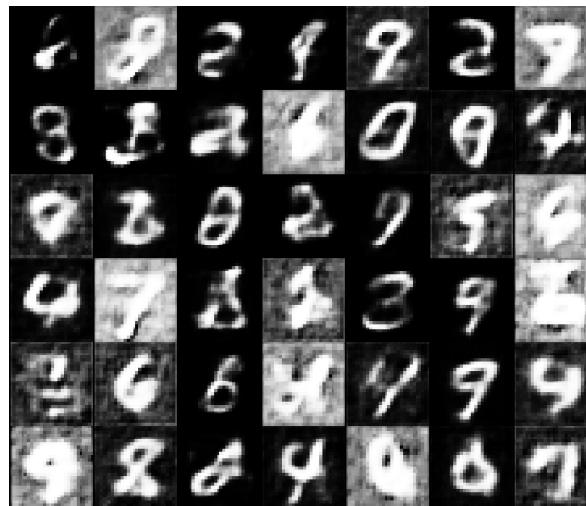
$$\text{Weighted conjugate feature duality: } \frac{1}{2\lambda} e^T D e + \frac{\lambda}{2} h^T D^{-1} h \geq e^T h$$

[Pandey, Schreurs & Suykens, 2019, arXiv:2002.01180, LOD 2020]

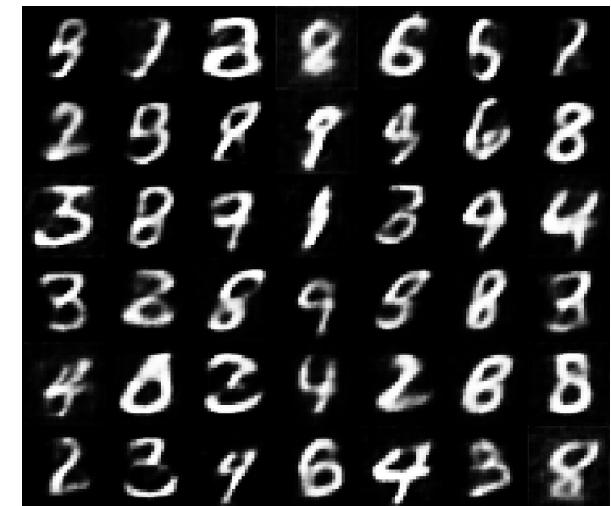
Robust Gen-RKM - Robust generation



VAE



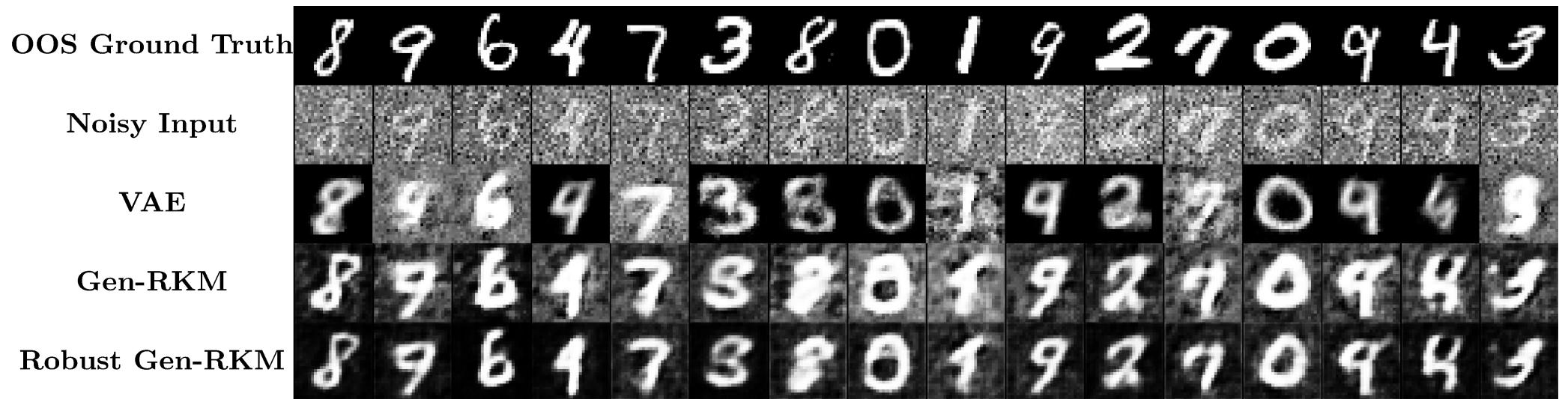
Gen-RKM



Robust Gen-RKM

[Pandey, Schreurs & Suykens, 2019, arXiv:2002.01180, LOD 2020]

Robust Gen-RKM - Robust denoising



[Pandey, Schreurs & Suykens, 2019, arXiv:2002.01180, LOD 2020]

RKM references and software

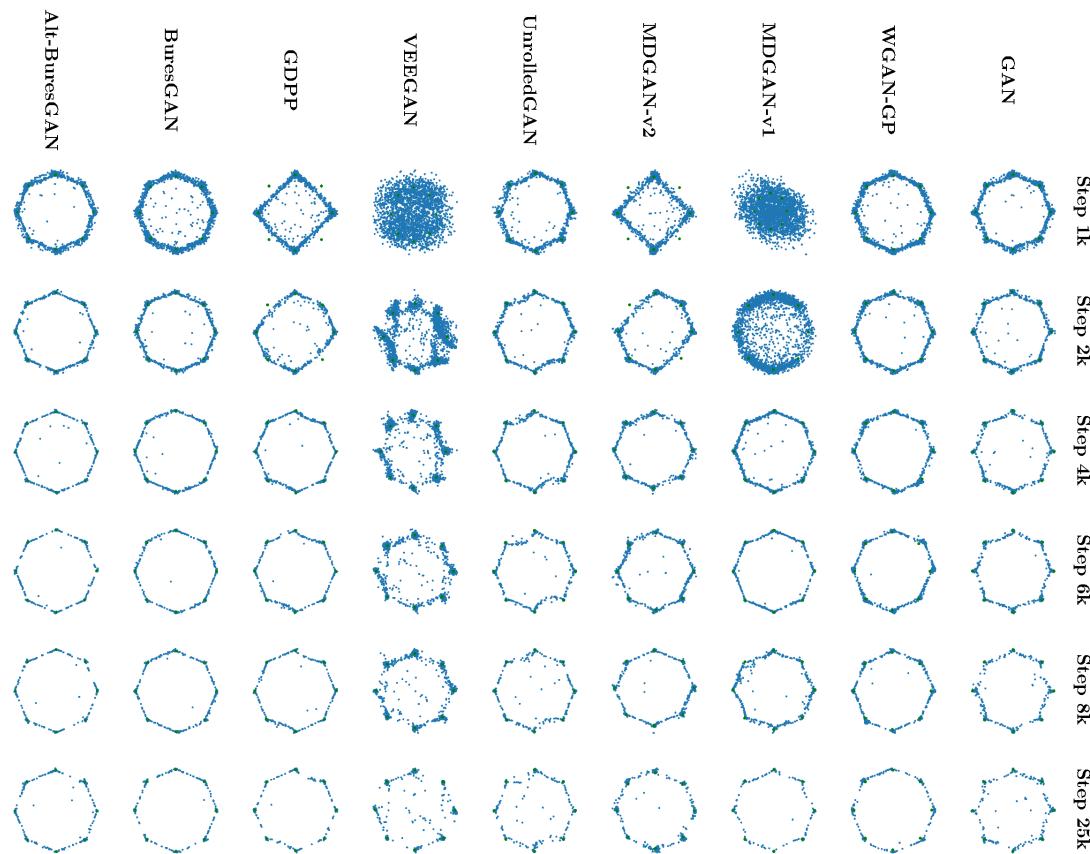
- Suykens J.A.K., "Deep Restricted Kernel Machines using Conjugate Feature Duality", *Neural Computation*, vol. 29, no. 8, pp. 2123-2163, Aug. 2017.
https://www.mitpressjournals.org/doi/pdf/10.1162/neco_a_00984
- Houthuys L., Suykens J.A.K., "Tensor Learning in Multi-View Kernel PCA", *International Conference on Artificial Neural Networks 2018 (ICANN 2018)*, Rhodes, Greece, Oct. 2018, pp. 205-215.
- Schreurs J., Suykens J.A.K., "Generative Kernel PCA", *European Symposium on Artificial Neural Networks (ESANN 2018)*, Bruges, Belgium, Apr. 2018, pp. 129-134.
- Pandey A., Schreurs J., Suykens J.A.K., Generative Restricted Kernel Machines, arXiv:1906.08144
- Pandey A., Fanuel M., Schreurs J., Suykens J.A.K., Disentangled Representation Learning and Generation with Manifold Optimization, arXiv:2006.07046
- Pandey A., Schreurs J., Suykens J.A.K., Robust Generative Restricted Kernel Machines using Weighted Conjugate Feature Duality, arXiv:2002.01180, *LOD 2020*.

Software: see <https://www.esat.kuleuven.be/stadius/E/software.php>

Future challenges

- efficient algorithms and implementations for large data
- extension to other loss functions and regularization schemes
- multimodal data, tensor models, coupling schemes
- models for deep clustering and semi-supervised learning
- choice kernel functions, invariances and symmetry properties
- deep generative models
- optimal transport
- synergies between neural networks, deep learning and kernel machines

BuresGAN (1)

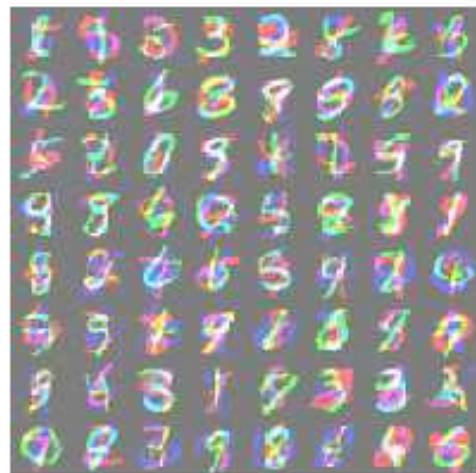


Bures Metric for Taming Mode Collapse in Generative Adversarial Networks
Computation of Bures distance in either *feature space or kernel space*

[De Meulemeester H., Schreurs J., Fanuel M., De Moor B., Suykens J.A.K., arXiv:2006.09096]

BuresGAN (2)

Stacked MNIST



CIFAR-10



CIFAR-100



Generated samples from a trained BuresGAN: high-quality images and preserving diversity.

[De Meulemeester H., Schreurs J., Fanuel M., De Moor B., Suykens J.A.K., arXiv:2006.09096]

Conclusions

- function estimation: **parametric versus kernel-based**
- **primal and dual** model representations
- **neural network interpretations** in primal and dual
- RKM: **new connections** between RBM, kernel PCA and LS-SVM
- **deep kernel machines**
- **generative models**
- **explainability**: latent space exploration, understanding the role of each individual neuron

Acknowledgements (1)

- Current and former co-workers at ESAT-STADIUS:
C. Alzate, Y. Chen, J. De Brabanter, K. De Brabanter, B. De Cooman, L. De Lathauwer, H. De Meulemeester, B. De Moor, H. De Plaen, Ph. Dreesen, M. Espinoza, T. Falck, M. Fanuel, Y. Feng, B. Gauthier, X. Huang, L. Houthuys, V. Jumutc, Z. Karevan, R. Langone, F. Liu, R. Mall, S. Mehrkanoon, G. Nisol, M. Orchel, A. Pandey, P. Patrinos, K. Pelckmans, S. RoyChowdhury, S. Salzo, J. Schreurs, M. Signoretto, Q. Tao, F. Tonin, J. Vandewalle, T. Van Gestel, S. Van Huffel, C. Varon, Y. Yang, and others
- Many other people for joint work, discussions, invitations, organizations
- Support from ERC AdG E-DUALITY, ERC AdG A-DATADRIVE-B, KU Leuven, OPTEC, IUAP DYSCO, FWO projects, IWT, Flanders AI

Acknowledgements (2)



Acknowledgements (3)



NEW: ERC Advanced Grant E-DUALITY
Exploring duality for future data-driven modelling

Thank you