Learning from data: model representations and duality

Johan Suykens

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

> ELO-X Leuven, March 2022



Overview

• Introduction

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

Introduction

Data world













Neural networks: powerful mappings (1)

in the early days of neural networks:



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

Neural networks: powerful mappings (2)



Output: vehicle control

Fully-connected layer Fully-connected layer Fully-connected layer



Convolutional feature map 64@3x20

Convolutional feature map 48@5x22

Convolutional feature map 36@14x47

Convolutional feature map 24@31x98

Normalized input planes 3@66x200



(27 million connections)



Waymo / Google Self-Driving Car



Uber



Tesla Autopilot



nuTonomy

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

Different paradigms

Deep	
Learning	

Neural

Networks

SVM, LS-SVM &

Kernel methods

Different paradigms



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

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

 \rightarrow one can directly solve in w, b

Linear function estimation (2)

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

$$\min_{\substack{w,b,e \\ \text{subject to}}} \frac{\frac{1}{2}w^T w + \gamma \frac{1}{2} \sum_i e_i^2}{\sum_i e_i^2}$$

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

$$\begin{bmatrix} 0 & 1_N^T \\ \hline 1_N & \Omega + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \hline \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \hline y \end{bmatrix}$$

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

inputs $x \in \mathbb{R}^d$, output $y \in \mathbb{R}$ training set $\{(x_i, y_i)\}_{i=1}^N$

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

 Model

inputs $x \in \mathbb{R}^d$, output $y \in \mathbb{R}$ training set $\{(x_i, y_i)\}_{i=1}^N$

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

Model
$$\searrow \qquad (D): \quad \hat{y} = \sum_i \alpha_i x_i^T x + b, \quad \alpha \in \mathbb{R}^N$$

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

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

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

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:



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: hidden layer as feature map [Suykens & Vandewalle, IEEE-TNN 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

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

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

Core models + constraints



Core models + constraints



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

Sparsity: through regularization or loss function

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

$$\min \sum_{j} |w_{j}| + \gamma \sum_{i} e_{i}^{2}$$

 \Rightarrow sparse w (e.g. Lasso)

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

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

 \Rightarrow sparse α (e.g. SVM)





• 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/



[Suykens et al., 2002]

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]

Kernels

Wide range of positive definite kernel functions possible:

- $K(x,z) = x^T z$
- 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]

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.



$$\hat{y} = \operatorname{sign}\left[\sum_{i} \alpha_{i} y_{i} K(x, x_{i}) + b\right]$$

with x_i for i = 1, ..., 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 !).

Krein spaces: indefinite kernels

• LS-SVM for indefinite kernel case:

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

and indefinite kernel $K(x_i, x_j) = K_+(x_i, x_j) - K_-(x_i, x_j)$ with positive definite kernels K_+, K_-

$$K_+(x_i, x_j) = \varphi_+(x_i)^T \varphi_+(x_j)$$
 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 [Mehrkanoon et al., 2018]

[X. Huang, Maier, Hornegger, Suykens, ACHA 2017] [Mehrkanoon, X. Huang, Suykens, Pattern Recognition, 2018] Related work of RKKS: [Ong et al 2004; Haasdonk 2005; Luss 2008; Loosli et al. 2015]

Banach spaces: tensor kernels

• Regression problem:

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

with $r = \frac{m}{m-1}$ for even $m \ge 2$, ρ convex and even. For m large this approaches ℓ^1 regularization.

• Tensor-kernel representation

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

[Salzo & Suykens, arXiv 1603.05876 AA2020; Salzo, Suykens, Rosasco, AISTATS 2018] related: RKBS [Zhang 2013; Fasshauer et al. 2015]

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.



 $source:\ https://deeplearning4j.org/generative-adversarial-network$



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:

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

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 <u>visible units</u> v and layer of <u>hidden units</u> h
 No hidden-to-hidden connections
- Energy:

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

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 <u>visible units</u> v and layer of <u>hidden units</u> h
 No hidden-to-hidden connections
- Energy:

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

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 <u>visible units</u> v and layer of <u>hidden units</u> h
 No hidden-to-hidden connections
- Energy:

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

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)

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

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

in other words ...

"deep sandwich"



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

"sandwich"



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

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 (\mathbf{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)



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

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

(applications in dimensionality reduction and denoising)

Kernel PCA: classical LS-SVM approach

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

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

• Dual problem (Lagrange duality) corresponds to kernel PCA

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

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]

From KPCA to RKM representation (1)

Model:

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

$$= regularization term Tr(W^T W)$$

$$- (\frac{1}{\lambda}) \text{ variance term } \sum_i e_i^T e_i$$

$$\downarrow$$
 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 $J \leq \overline{J}(h_i, W)$ 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

$$J = \frac{\eta}{2} \operatorname{Tr}(W^T W) - \frac{1}{2\lambda} \sum_{i=1}^{N} e_i^T e_i \text{ s.t. } e_i = W^T \varphi(x_i), \forall i$$

$$\leq -\sum_{i=1}^{N} e_i^T h_i + \frac{\lambda}{2} \sum_{i=1}^{N} h_i^T h_i + \frac{\eta}{2} \operatorname{Tr}(W^T W) \text{ s.t. } e_i = W^T \varphi(x_i), \forall i$$

$$= -\sum_{i=1}^{N} \varphi(x_i)^T W h_i + \frac{\lambda}{2} \sum_{i=1}^{N} h_i^T h_i + \frac{\eta}{2} \operatorname{Tr}(W^T W) \triangleq \overline{J}$$

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

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

From KPCA to RKM representation (3)

• Elimination of W gives the eigenvalue decomposition:

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

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

• Primal and dual model representations

$$(P)_{\rm RKM}: \quad \hat{e} = W^T \varphi(x)$$

$$\mathcal{M}$$

$$(D)_{\rm RKM}: \quad \hat{e} = \frac{1}{\eta} \sum_j h_j K(x_j, x)$$

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.



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

Primal and dual model representations



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])



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

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

Generative RKM

RKM objective for training and generating

• RBM energy function

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

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

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

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

Training: clamp $v \rightarrow \overline{J}_{train}(h, W)$ **Generating:** clamp $h, W \rightarrow \overline{J}_{gen}(v)$

[Schreurs & Suykens, ESANN 2018]

Explainable AI: latent space exploration

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



[figures by Joachim Schreurs]

Tensor-based RKM for Multi-view KPCA



 \bigcirc

v

Õ

[Houthuys & Suykens, ICANN 2018]

Ō

 \cap

v

 $\uparrow y$

Generative RKM (Gen-RKM) (1) \mathcal{H} Train: VU \mathcal{X} Y $h^\star \sim p(h)$ **Generate:** U y^{\star} x^{\star}

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

Analogy with the human brain



Gen-RKM (2)

The objective

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

results for training into the eigenvalue problem

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

with $H = [h_1...h_N]$ and kernel matrices K_1, 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 h^{\star} and the objective

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

giving

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

For generating \hat{x}, \hat{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 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 featurespaces) 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

$$\boldsymbol{k}_{\boldsymbol{x}^{\star}} = \frac{1}{\eta_1} \boldsymbol{K}_1 \boldsymbol{H}^{\top} \boldsymbol{h}^{\star}, \quad \boldsymbol{k}_{\boldsymbol{y}^{\star}} = \frac{1}{\eta_2} \boldsymbol{K}_2 \boldsymbol{H}^{\top} \boldsymbol{h}^{\star},$$

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

Using the kernel-smoother:

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

with $\tilde{k}_1(\boldsymbol{x}_i, \boldsymbol{x}^*)$ and $\tilde{k}_2(\boldsymbol{y}_i, \boldsymbol{y}^*)$ 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: $\phi_{\theta}(\cdot)$, $\psi_{\zeta}(\cdot)$ (e.g. CNN and transposed CNN).

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

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

with reconstruction errors

$$\mathcal{L}_{1}(\boldsymbol{x}_{i}^{\star}, \psi_{1_{\boldsymbol{\zeta}_{1}}}(\phi_{1_{\boldsymbol{\theta}_{1}}}(\boldsymbol{x}_{i}^{\star}))) = \frac{1}{N} \|\boldsymbol{x}_{i}^{\star} - \psi_{1_{\boldsymbol{\zeta}_{1}}}(\phi_{1_{\boldsymbol{\theta}_{1}}}(\boldsymbol{x}_{i}^{\star}))\|_{2}^{2}, \\ \mathcal{L}_{2}(\boldsymbol{y}_{i}^{\star}, \psi_{2_{\boldsymbol{\zeta}_{2}}}(\phi_{2_{\boldsymbol{\theta}_{2}}}(\boldsymbol{y}_{i}^{\star}))) = \frac{1}{N} \|\boldsymbol{y}_{i}^{\star} - \psi_{2_{\boldsymbol{\zeta}_{2}}}(\phi_{2_{\boldsymbol{\theta}_{2}}}(\boldsymbol{y}_{i}^{\star}))\|_{2}^{2}$$

and with $\Phi_{\boldsymbol{x}} = [\phi_1(\boldsymbol{x}_1), \dots, \phi_1(\boldsymbol{x}_N)], \Phi_{\boldsymbol{y}} = [\phi_2(\boldsymbol{y}_1), \dots, \phi_2(\boldsymbol{y}_N)], U, V$ from $\begin{bmatrix} \frac{1}{\eta_1} \Phi_{\boldsymbol{x}} \Phi_{\boldsymbol{x}}^\top & \frac{1}{\eta_1} \Phi_{\boldsymbol{x}} \Phi_{\boldsymbol{y}}^\top \\ \frac{1}{-\Phi_{\boldsymbol{x}}} \Phi_{\boldsymbol{x}}^\top & \frac{1}{-\Phi_{\boldsymbol{x}}} \Phi_{\boldsymbol{x}}^\top \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \Lambda.$

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

Hence, joint feature learning and subspace learning.

Gen-RKM: examples



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, Neural Networks 2021]

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, Neural Networks 2021] **Gen-RKM:** latent space exploration (2)



MNIST reconstructed images by bilinear-interpolation in latent space [Pandey, Schreurs & Suykens, Neural Networks 2021]
Gen-RKM: latent space exploration (3)



CelebA reconstructed images by bilinear-interpolation in latent space [Pandey, Schreurs & Suykens, Neural Networks 2021]

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

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

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

Robust Gen-RKM - Robust generation



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

Robust Gen-RKM - Robust denoising



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

Deep unsupervised learning

Unsupervised learning of disentangled representations in deep restricted kernel machines with **orthogonality constraints**

$$\min_{W_1, W_2, H^{(1)}, H^{(2)}} -\sum_i \varphi_1(x_i)^T W_1 h_i^{(1)} + \frac{\lambda_1}{2} \sum_i h_i^{(1)^T} h_i^{(1)} + \frac{1}{2} \operatorname{Tr}(W_1^T W_1) -\sum_i \varphi_2(h_i^{(1)})^T W_2 h_i^{(2)} + \frac{\lambda_2}{2} \sum_i h_i^{(2)^T} h_i^{(2)} + \frac{1}{2} \operatorname{Tr}(W_2^T W_2)$$

subject to

$$\left[\begin{array}{c} H^{(1)} \\ H^{(2)} \end{array}\right] \left[\begin{array}{c} H^{(1)} & H^{(2)} \end{array}\right] = I$$
 with $H^{(1)} = [h_1^{(1)}...h_N^{(1)}]$ and $H^{(2)} = [h_1^{(2)}...h_N^{(2)}].$

[[]F. Tonin, P. Patrinos, J.A.K. Suykens, Neural Networks 2021]

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

Dynamical Systems



Nonlinear system identification

- white/grey/black box
- aims: prediction, models for control, monitoring
- model structures: input/output, state space, block oriented deterministic, stochastic
- parametrizations: linear, polynomial, piecewise linear, neural nets, ...
- time domain, frequency domain, discrete time, continuous time
- different aspects: complexity, optimization, statistics, scalability, ...

[Ljung, 1999; Schoukens et al., 2012; Sjoberg et al., 1995; Billings, 2013; Giri & Bai, 2010; Pillonetto et al., 2014; Suykens et al., 1995; ...]

Benchmarks: chaotic time-series prediction



Training: $\hat{y}_{k+1} = f(y_k, y_{k-1}, \dots, y_{k-p})$, prediction: $\hat{y}_{k+1} = f(\hat{y}_k, \hat{y}_{k-1}, \dots, \hat{y}_{k-p})$ Fixed-size kernel methods [Suykens et al., 2002; Espinoza et al., 2003]

Fixed-size kernel method

- Find finite dimensional approximation to feature map \$\tilde{\varphi}(\cdot): \mathbb{R}^p → \mathbb{R}^M\$ based on the eigenvalue decomposition of the kernel matrix (on a subset of size \$M ≪ N\$).
- Based on [Williams & Seeger, 2001]: relates KPCA to a Nyström approximation of the integral equation

$$\int K(z,x)\phi_i(x)dP_X = \lambda_i\phi_i(z)$$

- Fixed-size method [Suykens et al., 2002; De Brabanter et al., 2009]:
 - selects subset such that it represents the data distribution P_X
 - optimizes quadratic Renyi entropy citerion (instead of random subset)
 - LS-SVM: estimate in **primal** (**sparse** representation):

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



Block oriented models: Hammerstein LS-SVM

• Linear part with ARX structure:

$$\hat{y}_k = \sum_{i=1}^{n_y} a_i y_{k-i} + \sum_{j=1}^{n_u} \beta_j (w^T \varphi(u_{k-j}) + b_0)$$

Estimation of a_i, β_j, w, b_0 : non-convex problem.

- Overparametrization approach [Bai & Fu, 1998; Chang & Luus, 1971]
- LS-SVM formulation with constraints [Goethals et al., 2005; Falck et al., 2009]:

$$\min_{\substack{w_{j}, a, b, e_{k} \\ \text{subject to}}} \frac{\frac{1}{2} \sum_{j=1}^{n_{u}} w_{j}^{T} w_{j} + \gamma \frac{1}{2} \sum_{k=d+1}^{d+N} e_{k}^{2} \\
\frac{y_{k}}{\sum_{i=1}^{n_{y}} a_{i} y_{k-i}} + \sum_{j=1}^{n_{u}} w_{j}^{T} \varphi(u_{k-j}) + b + e_{k}, \quad \forall k \\
\sum_{k=1}^{N} w_{j}^{T} \varphi(u_{k}) = 0, \quad \forall j = 1, ..., n_{u}$$

- Hammerstein/Wiener system identification using Best Linear Approximation (BLA) within LS-SVM framework [Castro-Garcia et al., 2015, 2017]
- Impulse Response Constrained LS-SVM modeling for SISO/MIMO Hammerstein system identification [Castro-Garcia et al., 2017]

Subspace algorithms: nonlinear state space models

• Given I/O data, estimate parameter vector θ of the nonlinear state space model:

$$\begin{cases} x_{k+1} = f(x_k, u_k; \theta) \\ \hat{y}_k = g(x_k, u_k; \theta) \end{cases}$$

- Conceptually 2 steps:
 - **Step 1**: estimate state vector sequence $\{\hat{x}_k\}$ from the I/O data
 - **Step 2**: given I/O data and $\{\hat{x}_k\}$, solve a set of nonlinear equations to estimate θ .
- State vector sequence from Kernel CCA [Verdult et al., MTNS 2004]
- Hammerstein systems [Goethals et al., IEEE-TAC 2005]
- Linear systems [Larimore 1992; Van Overschee & De Moor 1996]

Kernel CCA

• Kernel CCA: primal formulation [Suykens et al., 2002]

 $\min_{w,v,b,d,e,r} w^T w + v^T v + \nu \sum_i (e_i - r_i)^2 \text{ s.t. } \begin{cases} e_i = w^T \varphi_1(x_i) + b, \forall i \\ r_i = v^T \varphi_2(z_i) + d, \forall i \end{cases}$

- Data $\{x_i\}$: **past** of time-series
- Data $\{z_i\}$: future of time-series
- State vector sequence from kernel CCA
- System order estimate from kernel CCA
- Dual problem: generalized eigenvalue problem

Stability issues

• Example: $x_{k+1} = ax_k(1 - x_k)$ (logistic map) [Strogatz, 1994] \rightarrow simple looking systems may possess complex behaviour



If knowledge of single fixed point, then impose $a \in [0, 1]$ in system identification

- Examples of **imposing stability constraints** in system identification:
 - linear systems [Chui & Maciejowski 1996; Van Gestel et al. 2001]
 - neural state space models, (deep) recurrent networks (NLq) [Suykens et al., 1996]

Black-box weather forecasting (1)



Weather data 350 stations located in US

Features:

Tmax, Tmin, precipitation, wind speed, wind direction ,...

Black-box forecasting multiple weather stations simultaneously

[Signoretto, Frandi, Karevan, Suykens, IEEE-SCCI, 2014]

Black-box weather forecasting (2)

- multi-task learning with kernel-based models and nuclear norm regularization [Signoretto et al. 2014]
- multi-view LS-SVM regression [Houthuys et al., 2017]
- feature selection: spatio-temporal, clustering-based, transductive [Karevan & Suykens, 2016, 2017]
- **optimal prediction** schemes: moving least squares (transductive learning) [Karevan et al., 2017]

Multi-view learning: kernel-based (1)

• Primal problem:

$$\min_{\substack{w^{[v]}, e^{[v]} \\ \text{subject to}}} \frac{\frac{1}{2} \sum_{v=1}^{V} w^{[v]^{T}} w^{[v]} + \frac{1}{2} \sum_{v=1}^{V} \gamma^{[v]} e^{[v]^{T}} e^{[v]} + \rho \sum_{v,u=1; v \neq u}^{V} e^{[v]^{T}} e^{[u]} }{y = \Phi^{[v]} w^{[v]} + b^{[v]} 1_{N} + e^{[v]}, \ v = 1, ..., V }$$

• Dual:

$$\begin{bmatrix} 0_{V \times V} & 1_M^T \\ \hline \Gamma_M 1_M + \rho \ \mathcal{I}_M 1_M & \Gamma_M \Omega_M + \mathbb{I}_{NV} + \rho \ \mathcal{I}_M \Omega_M \end{bmatrix} \begin{bmatrix} b_M \\ \hline \alpha_M \end{bmatrix} = \begin{bmatrix} 0_V \\ \hline \Gamma_M y_M + (V-1)\rho \ y_M \end{bmatrix}$$

• Prediction:

$$\hat{y}(\mathbf{x}) = \sum_{v=1}^{V} \beta_v \sum_{k=1}^{N} \alpha_k^{[v]} K^{[v]}(\mathbf{x}^{[v]}, \mathbf{x}_k^{[v]}) + b^{[v]}$$

[Houthuys et al., 2017]

Multi-view learning: kernel-based (2)

- Data set:
 - Real measurements for weather elements such as temperature, humidity, etc.
 - From 2007 until mid 2014
 - Two test sets:
 - mid-November 2013 until mid-December 2013
 - from mid-April 2014 to mid-May 2014
- **Goal:** Forecasting minimum and maximum temperature for one to six days ahead in Brussels Belgium
- Views: Brussels together with 9 neighboring cities
- Tuning parameters:
 - kernel parameters for each view
 - regularization parameters $\gamma^{[v]}$
 - coupling parameter ho

Multi-view learning: kernel-based (3)

Apr/May



Nov/Dec



Kernel Spectral Clustering (KSC)

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

$$\min_{\substack{w,b,e\\\text{subject to}}} \frac{1}{2}w^T w - \gamma \frac{1}{2}e^T V e$$

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

with weighting matrix V and $\varphi(\cdot):\mathbb{R}^d\to\mathbb{R}^h$ 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 $V = D^{-1}$ with degree matrix $D = \text{diag}\{d_i\}$, $d_i = \sum_{j=1}^N \Omega_{ij}$ relates to random walks algorithm [Chung, 1997; Shi & Malik, 2000; Ng 2002]

[Alzate & Suykens, IEEE-PAMI, 2010]

Kernel spectral clustering: more clusters

• Case of k clusters: additional sets of constraints

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

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

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

[Alzate & Suykens, IEEE-PAMI, 2010]

Primal and dual model representations

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

$$(P): \operatorname{sign}[\hat{e}_{*}^{(l)}] = \operatorname{sign}[w^{(l)}{}^{T}\varphi(x_{*}) + b_{l}]$$

$$\mathcal{M}$$

$$(D): \operatorname{sign}[\hat{e}_{*}^{(l)}] = \operatorname{sign}[\sum_{j} \alpha_{j}^{(l)} K(x_{*}, x_{j}) + b_{l}]$$





[Alzate & Suykens, 2012]

Example: power grid - identifying customer profiles (1)

Power load: 245 substations, hourly data (5 years), d = 43.824Periodic 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 = 43.824Model 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

KSC for automated structural health monitoring



[Langone, Reynders, Mehrkanoon, Suykens, MSSP 2017]

Dynamic clustering of PM10 concentrations (1)

PM10 time-series: PM10 data (Particulate Matter) registered during a heavy pollution episode (Jan 20 2010 - Feb 1 2010) in Europe.



Kernel spectral clustering [Langone, Agudelo, De Moor, Suykens, Neurocomputing, 2014]

Example: dynamic clustering of PM10 concentrations (2)



video - [Langone, Agudelo, De Moor, Suykens, Neurocomputing, 2014]

Evolving networks - temporal smoothness

• Binary clustering case: adding a memory effect

$$\min_{\substack{w,e,b\\\text{subject to}}} \frac{1}{2}w^T w - \gamma \frac{1}{2}e^T D^{-1}e - \nu w^T w_{\text{old}}$$

with $w_{\rm old}$ the previous result in time.

- Aims at including **temporal smoothness**
- Smoothed modularity criterion

[Langone, Alzate, Suykens, Physica A, 2013]



- 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, Leuven.AI

Acknowledgements (2)






Acknowledgements (3)



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

Thank you