Autoencoding any Data through Kernel Autoencoders (AIstats 2019)

Pierre Laforgue, Télécom ParisTech, Paris, France

Joint work with Stephan Clémençon and Florence d'Alché-Buc

Representation Learning

Autoencoders

Kernel Methods

Kernel Autoencoders

Experiments

Conclusion & Future Work

Outline

Example: Type II diabetes occurrence prediction

A representation: a collection of features that characterizes the observation

Example: Type II diabetes occurrence prediction

A representation: a collection of features that characterizes the observation

- **Representation 1:** (PL, 42, dark brown, **green**, 175 cm, ...)

  $\rightarrow$ Diabetes occurrence prediction complex (impossible)

Example: Type II diabetes occurrence prediction

A representation: a collection of features that characterizes the observation

- **Representation 1:** (PL, 42, dark brown, **green**, 175 cm, ...)
  $\rightarrow$ Diabetes occurrence prediction complex (impossible)

- **Representation 2:** (175 cm, 62 kg, 26 years old, ♂, ...)
  $\rightarrow$ Diabetes occurrence prediction possible

Example: Type II diabetes occurrence prediction

A representation: a collection of features that characterizes the observation

- **Representation 1:** (PL, 42, dark brown, **green**, 175 cm, ...)
  $\rightarrow$ Diabetes occurrence prediction complex (impossible)
- **Representation 2:** (175 cm, 62 kg, 26 years old, ♂, ...)
  $\rightarrow$ Diabetes occurrence prediction possible
- **Representation 3:** (BMI=20.24, family background, ...)
  $\rightarrow$ Diabetes occurrence prediction facilitated

Representation Learning

- **Raw data:** redundant, non relevant, massive
- **Ideal data:** independent, discriminative

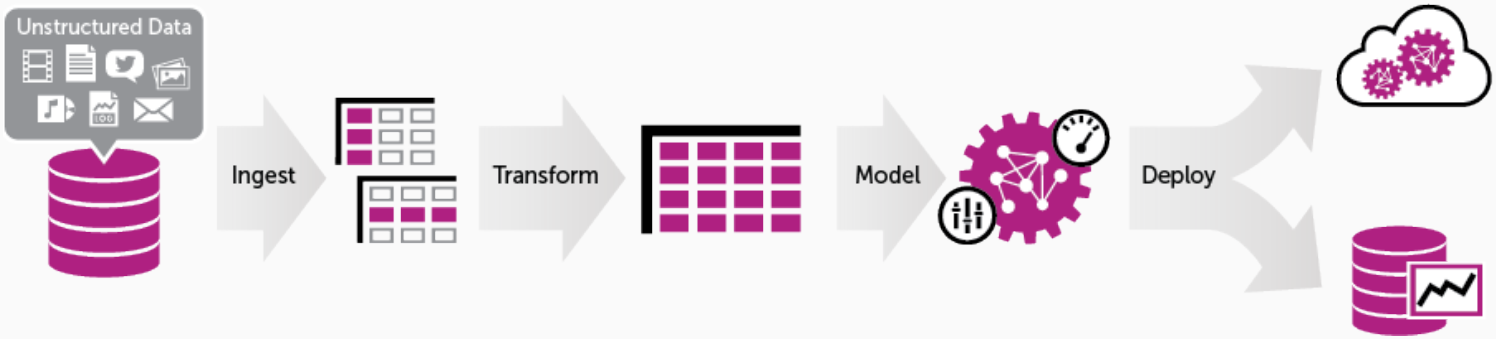
Representation Learning

- **Raw data:** redundant, non relevant, massive
- **Ideal data:** independent, discriminative
- **Feature engineering:** implies domain experts

Representation Learning

- **Raw data:** redundant, non relevant, massive
- **Ideal data:** independent, discriminative

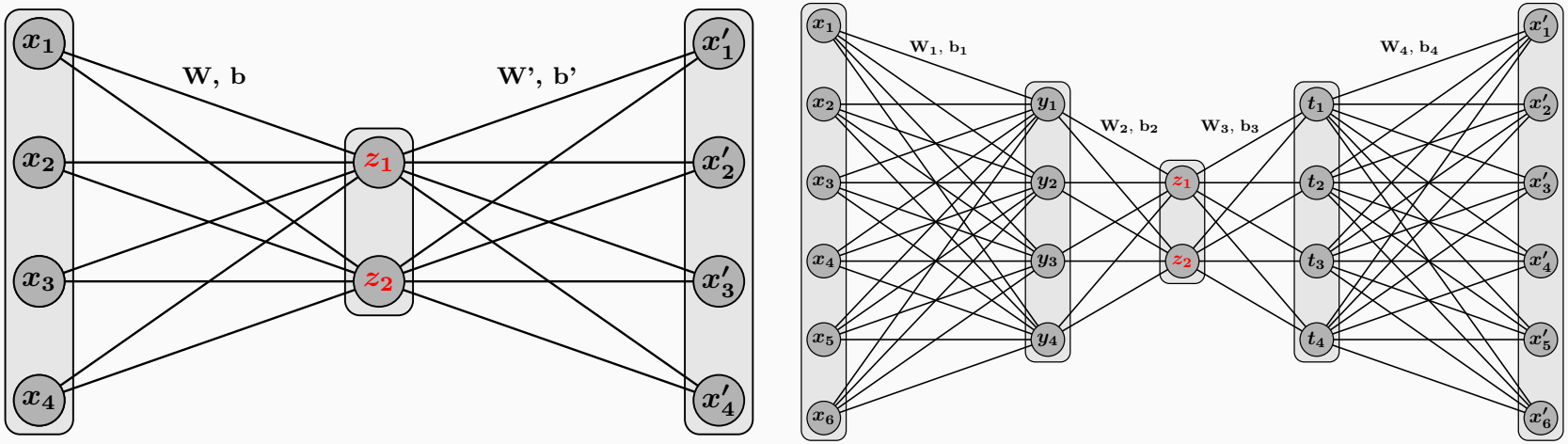
- **Feature engineering:** implies domain experts
- **Feature/Representation Learning:** automate the process



Outline

Autoencoders (AEs): Principle

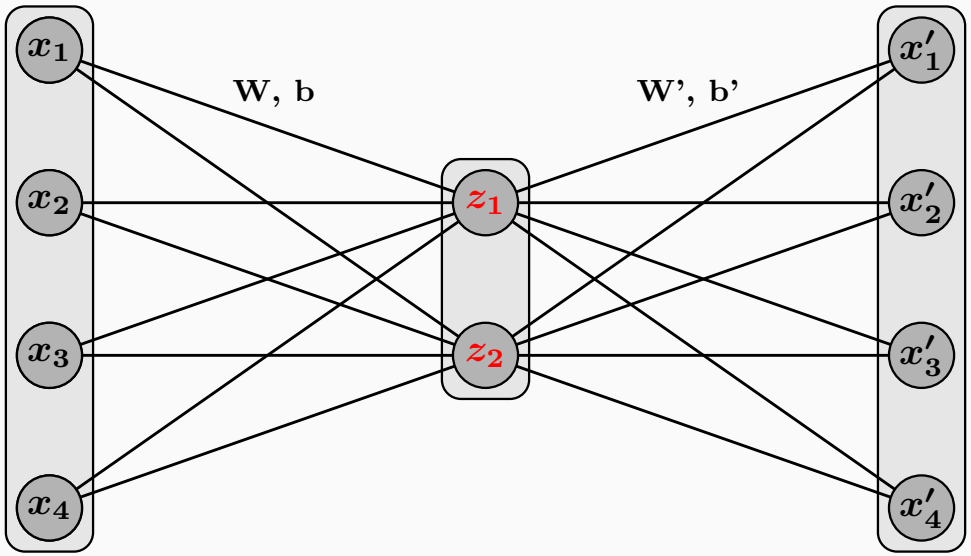
- **Idea:** compress and reconstruct inputs by a Neural Net (NN)
- Elementary mapping: $f : [0,1]^d \rightarrow [0,1]^p$ such that
$$f(x) = \sigma(Wx + b), \; W \in \mathbb{R}^{p \times d}, b \in \mathbb{R}^p$$
- Neural network: symmetric, hour-glass shaped
- **AE:** output x' must match input x (self-supervised)



(a) 1 hidden layer AE (b) 3 hidden layers AE

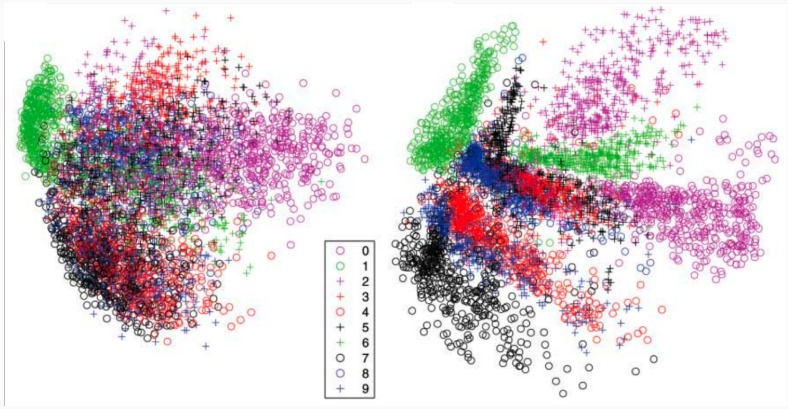
Autoencoders: Training

- $z = f_{\boldsymbol{W},\boldsymbol{b}}(x) = \sigma(\boldsymbol{W}x + \boldsymbol{b}) \quad x' = f_{\boldsymbol{W'},\boldsymbol{b'}}(z) = \sigma(\boldsymbol{W'}z + \boldsymbol{b'})$
- $\boldsymbol{\theta}^* = argmin_\theta \|x - x'\|^2 = argmin_\theta \left\|x - f_{\boldsymbol{W'},\boldsymbol{b'}} \circ f_{\boldsymbol{W},\boldsymbol{b}}(x)\right\|^2$
- Optimal encoding $z^* = \sigma(\boldsymbol{W}^*x + \boldsymbol{b}^*)$

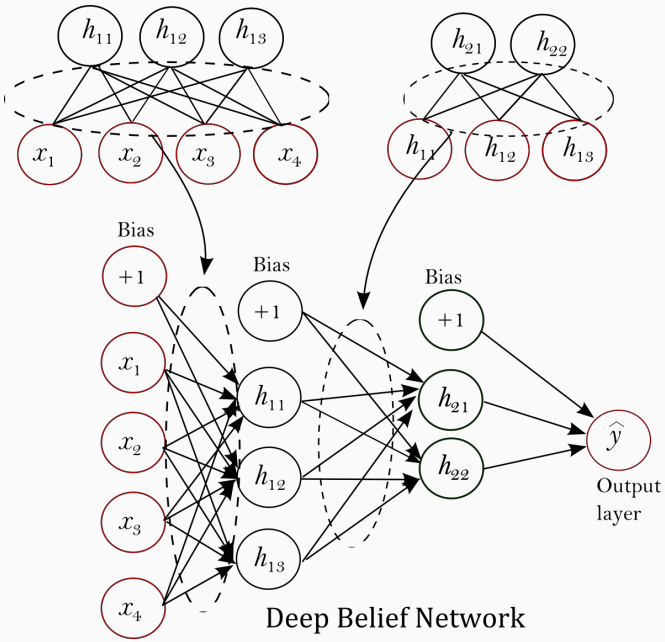


Autoencoders: Uses

- Data compression (PCA) [*Bourlard 1988, Hinton 2006*]
- Pre-training of neural networks [*Bengio & al. 2007*]
- Denoising [*Vincent, Larochelle & al. 2010*]



(c) PCA / AE

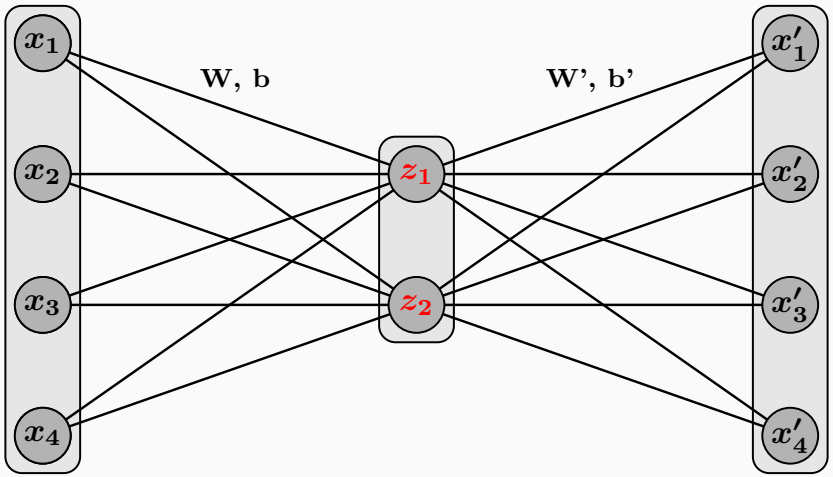


(d) Pre-training by AE

Autoencoders: Summary

❶ $$\min_{f_l \in \mathsf{NN}_{\mathsf{em}}} \frac{1}{n} \sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|^2$$

❷ $x_i \in [0,1]^d$ or $x_i \in \mathbb{R}^d$



Outline

Kernel Methods: Definitions

- $\mathcal{X}$: space where observations live. Example: $\mathbb{R}^d$, graphs, ...
- Let $\mathcal{H}_k$ be a Hilbert space, and $\varphi : \mathcal{X} \to \mathcal{H}_k$, such that $\forall (x, x') \in \mathcal{X} \times \mathcal{X}$, $\left\langle \varphi(x), \varphi(x') \right\rangle_{\mathcal{H}_k}$ is easy to compute
- Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $k(x, x') = \left\langle \varphi(x), \varphi(x') \right\rangle_{\mathcal{H}_k}$
- $\mathcal{H}_k = \overline{Span\left\{\varphi(x) : x \in \mathcal{X}\right\}}$ (RKHS)

Kernel Methods: Examples

Linear kernel

- $\mathcal{X} = \mathbb{R}^d$
- $\varphi(x) = x \in \mathbb{R}^d$
- $k(x, x') = \langle \varphi(x), \varphi(x') \rangle = \langle x, x' \rangle$

Kernel Methods: Examples

Linear kernel

- $\mathcal{X} = \mathbb{R}^d$
- $\varphi(x) = x \in \mathbb{R}^d$
- $k(x, x') = \langle \varphi(x), \varphi(x') \rangle = \langle x, x' \rangle$

Polynomial kernel

- $\mathcal{X} = \mathbb{R}^d$
- $\varphi(x) = (x_1^2, x_1 x_2, \ldots, x_d x_{d-1}, x_d^2) \in \mathbb{R}^{d^2}$
- $k(x, x') = \sum_{i,j=1}^{d} x_i x_j x_i' x_j' = \left(\sum_{i=1}^{d} x_i x_i' \right)^2 = \langle x, x' \rangle^2$

Kernel Methods: Examples

Linear kernel

- $\mathcal{X} = \mathbb{R}^d$
- $\varphi(x) = x \in \mathbb{R}^d$
- $k(x, x') = \langle \varphi(x), \varphi(x') \rangle = \langle x, x' \rangle$

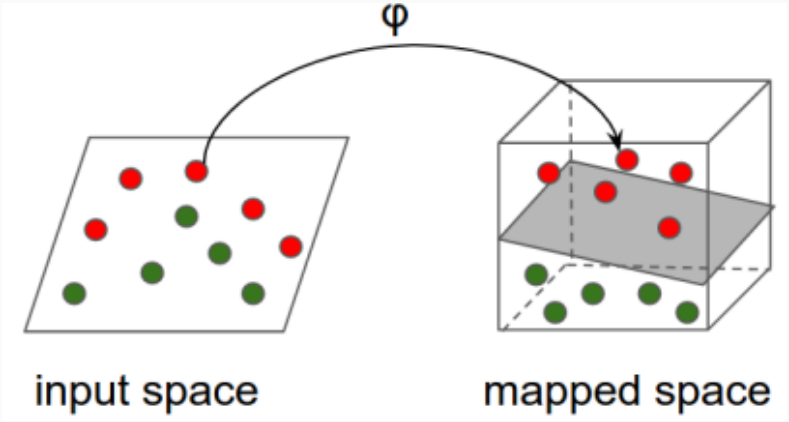
Polynomial kernel

- $\mathcal{X} = \mathbb{R}^d$
- $\varphi(x) = (x_1^2, x_1 x_2, \ldots, x_d x_{d-1}, x_d^2) \in \mathbb{R}^{d^2}$
- $k(x, x') = \sum_{i,j=1}^{d} x_i x_j x_i' x_j' = \left(\sum_{i=1}^{d} x_i x_i' \right)^2 = \langle x, x' \rangle^2$

Gaussian kernel

- $\mathcal{X} = \mathbb{R}^d$
- $\varphi(x) = t \mapsto e^{-\gamma \|t - x\|^2} \in \mathcal{F}(\mathbb{R}^d, \mathbb{R})$
- $k(x, x') = e^{-\gamma \|x - x'\|^2}$

Kernel Methods: Motivation 1



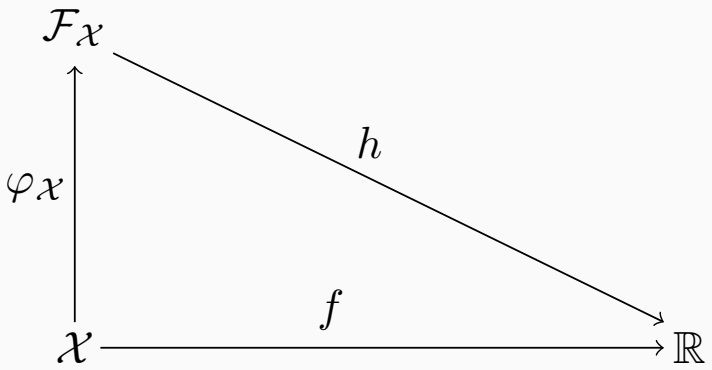
$X \in \mathbb{R}^{n \times p}, Y \in \mathbb{R}^n$

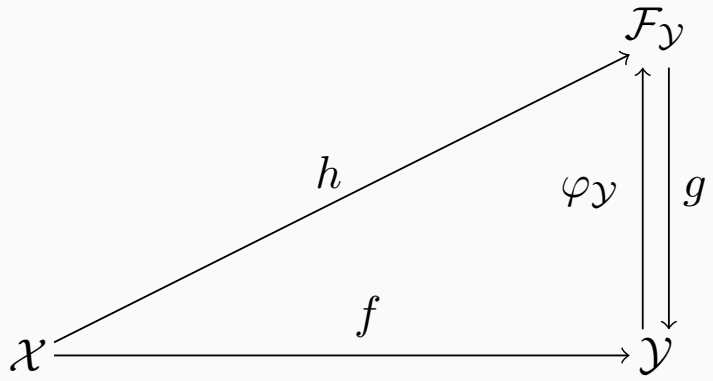
- $\min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|^2 + 2n\lambda\|\beta\|^2$
- $\min_{\beta \in \mathbb{R}^p} \sum_i (y_i - \langle x_i, \beta \rangle_{\mathbb{R}^p})^2 + 2n\lambda\|\beta\|^2_{\mathbb{R}^p}$

- $\min_{\omega \in \mathcal{H}_k} \sum_i (y_i - \langle \varphi(x_i), \omega \rangle_{\mathcal{H}_k})^2 + 2n\lambda\|\omega\|^2_{\mathcal{H}_k}$
- $\min_{\alpha \in \mathbb{R}^n} \|Y - K\alpha\|^2 + 2n\lambda\alpha^T K\alpha$

Kernel Methods: Motivation 2

$$\min_h \sum_i \ell(y_i, h(x_i))$$





$$\min_h \sum_i (y_i - h(\varphi_\mathcal{X}(x_i)))^2$$

$$\min_h \sum_i \|\varphi_\mathcal{Y}(y_i) - h(x_i)\|^2_{\mathcal{F}_\mathcal{Y}}$$

$$f(x) = \arg\min_y \|\varphi_\mathcal{Y}(y) - h(x)\|^2_{\mathcal{F}_\mathcal{Y}}$$

Kernel Methods: Summary

Autoencoders

❶ $\min_{f_l \in \mathrm{NN_{em}}} \frac{1}{n} \sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathbb{R}^d}^2$

❷ $x_i \in [0,1]^d$ or $x_i \in \mathbb{R}^d$

Kernelization

❸ allows to deal with non-vectorial data

❹ $x \longleftrightarrow \varphi(x)$

❺ computable as long as only dot products (or norms) are involved

❻ can be done on input or output

Kernel Methods: Summary

Autoencoders

❶ $$\min_{f_l \in \mathrm{NN}_{\mathrm{em}}} \frac{1}{n} \sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathbb{R}^d}^2$$

❷ $x_i \in [0,1]^d$ or $x_i \in \mathbb{R}^d$

Kernelization

❸ allows to deal with non-vectorial data

❹ $x \longleftrightarrow \varphi(x)$

❺ computable as long as only dot products (or norms) are involved

❻ can be done on input or output

$$\min_{f_l \in \mathrm{NN}_{\mathrm{em}}} \frac{1}{n} \sum_{i=1}^{n} \left\| \varphi(x_i) - f_L \circ \ldots \circ f_1(\varphi(x_i)) \right\|_{\mathcal{H}_k}^2$$

Kernel Methods: Summary

Autoencoders

❶ $$\min_{f_l \in \mathrm{NN_{em}}} \frac{1}{n} \sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathbb{R}^d}^2$$

❷ $x_i \in [0,1]^d$ or $x_i \in \mathbb{R}^d$

Kernelization

❸ allows to deal with non-vectorial data

❹ $x \longleftrightarrow \varphi(x)$

❺ computable as long as only dot products (or norms) are involved

❻ can be done on input or output

$$\min_{f_l \in \mathrm{NN_{em}}} \frac{1}{n} \sum_{i=1}^{n} \left\| \varphi(x_i) - f_L \circ \ldots \circ f_1(\varphi(x_i)) \right\|_{\mathcal{H}_k}^2$$

$\rightarrow$ Need for OVKs and vv-RKHSs

Outline

Schema

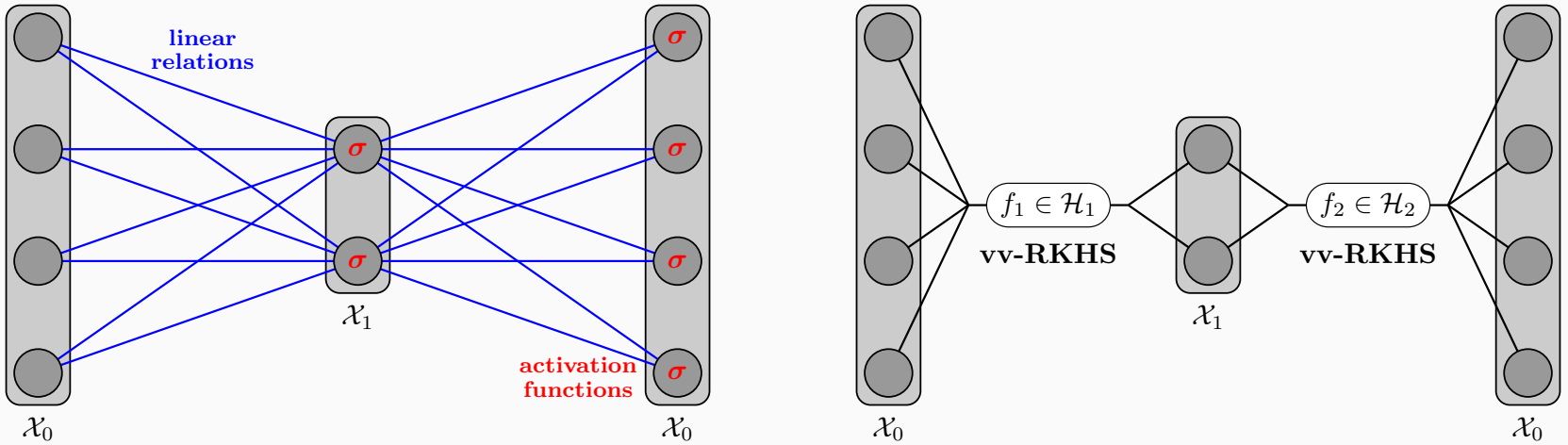


Figure 1: Standard and Kernel 2-layer Autoencoders

Formally

$$\textbf{AE}: \quad \min_{f_l \in \mathrm{NN}_{\mathrm{em}}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2$$

$$\textbf{KAE}: \quad \min_{f_l \in \text{vv-RKHS}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2 + \sum_{l=1}^{L} \lambda_l \| f_l \|_{\mathcal{H}_l}^2$$

Formally

$$\textbf{AE}: \quad \min_{f_l \in \mathrm{NN}_{\mathrm{em}}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2$$

$$\textbf{KAE}: \quad \min_{f_l \in \text{vv-RKHS}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2 + \sum_{l=1}^{L} \lambda_l \|f_l\|_{\mathcal{H}_l}^2$$

1. Novel algorithm of Representation Learning

Formally

$$\textbf{AE}: \quad \min_{f_l \in \mathsf{NN_{em}}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2$$

$$\textbf{KAE}: \quad \min_{f_l \in \mathsf{vv\text{-}RKHS}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2 + \sum_{l=1}^{L} \lambda_l \|f_l\|_{\mathcal{H}_l}^2$$

1. Novel algorithm of Representation Learning
2. $\mathcal{X}_0$ Hilbert non necessarily Euclidean (not only $\mathbb{R}^d$)

Formally

$$\textbf{AE}: \quad \min_{f_l \in \mathsf{NN_{em}}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2$$

$$\textbf{KAE}: \quad \min_{f_l \in \mathsf{vv\text{-}RKHS}} \quad \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - f_L \circ \ldots \circ f_1(x_i) \right\|_{\mathcal{X}_0}^2 + \sum_{l=1}^{L} \lambda_l \| f_l \|_{\mathcal{H}_l}^2$$

1. Novel algorithm of Representation Learning
2. $\mathcal{X}_0$ Hilbert non necessarily Euclidean (not only $\mathbb{R}^d$)
3. Interesting Hilbert: (kernel) feature space

Autoencoding any data

$$\textbf{K}^2\textbf{AE:} \min_{f_l \in \text{vv-RKHS}} \frac{1}{n} \sum_{i=1}^{n} \left\| \varphi(x_i) - f_L \circ \ldots \circ f_1(\varphi(x_i)) \right\|_{\mathcal{X}_0}^2 + \sum_{l=1}^{L} \lambda_l \|f_l\|_{\mathcal{H}_l}^2$$

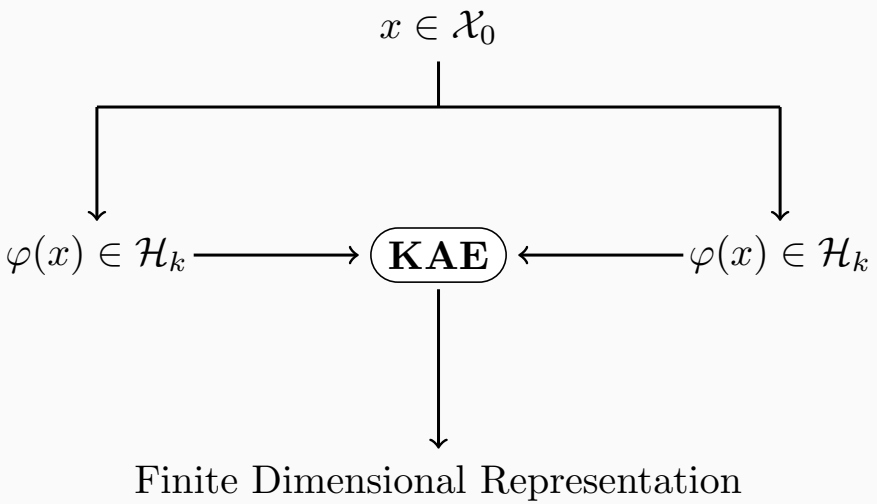


Figure 2: Autoencoding on any $\mathcal{X}_0$

Optimization

(RT) $\exists \left(\varphi^*_{1,1}, \ldots, \varphi^*_{1,n}, \ldots, \varphi^*_{L_0,n}\right) \in \mathcal{X}_1^n \times \ldots \times \mathcal{X}_{L_0}^n$:

$$\forall\ l \leq L, \quad f_l^*(\cdot) = \sum_{i=1}^n \mathcal{K}_l\left(\ \cdot\ , x_i^{*(l-1)}\right) \varphi^*_{l,i}.$$

Algorithm 1 General Hilbert KAE and K^2AE

input : Gram matrix K_{in}

init : $\Phi_1 = \Phi_1^{init}, \ldots, \Phi_{L-1} = \Phi_{L-1}^{init}$,
$N_L = N_{\mathrm{KRR}}\left(\Phi_1, \ldots, \Phi_{L-1}, K_{in}, \lambda_L\right)$

for *epoch* t *from* 1 *to* T **do**

  // inner coefficients updates at fixed N_L

  for *layer* l *from* 1 *to* $L-1$ **do**

    $\Phi_l = \Phi_l - \gamma_t\ \nabla_{\Phi_l}\left(\hat{\epsilon}_n + \Omega \mid N_L\right)$

  // N_L update

  $N_L = N_{\mathrm{KRR}}\left(\Phi_1, \ldots, \Phi_{L-1}, K_{in}, \lambda_L\right)$

return $\Phi_1, \ldots, \Phi_{L-1}$

Connection to Kernel PCA

2-layer K^{2}AE with internal layer of size p, only linear kernels, and without penalization. $K_\phi \in \mathbb{R}^{n \times n}$ denotes the input Gram matrix, $((\sigma_1, u_1) \ldots, (\sigma_p, u_p))$ its p largest eigen values/vectors. Then:

K^{2}AE output: $\left(\sqrt{\sigma_1} u_1, \ldots, \sqrt{\sigma_p} u_p\right) \in \mathbb{R}^{n \times p}$

KPCA output: $(\sigma_1 u_1, \ldots, \sigma_p u_p) \in \mathbb{R}^{n \times p}$

Outline

Concentric Circles

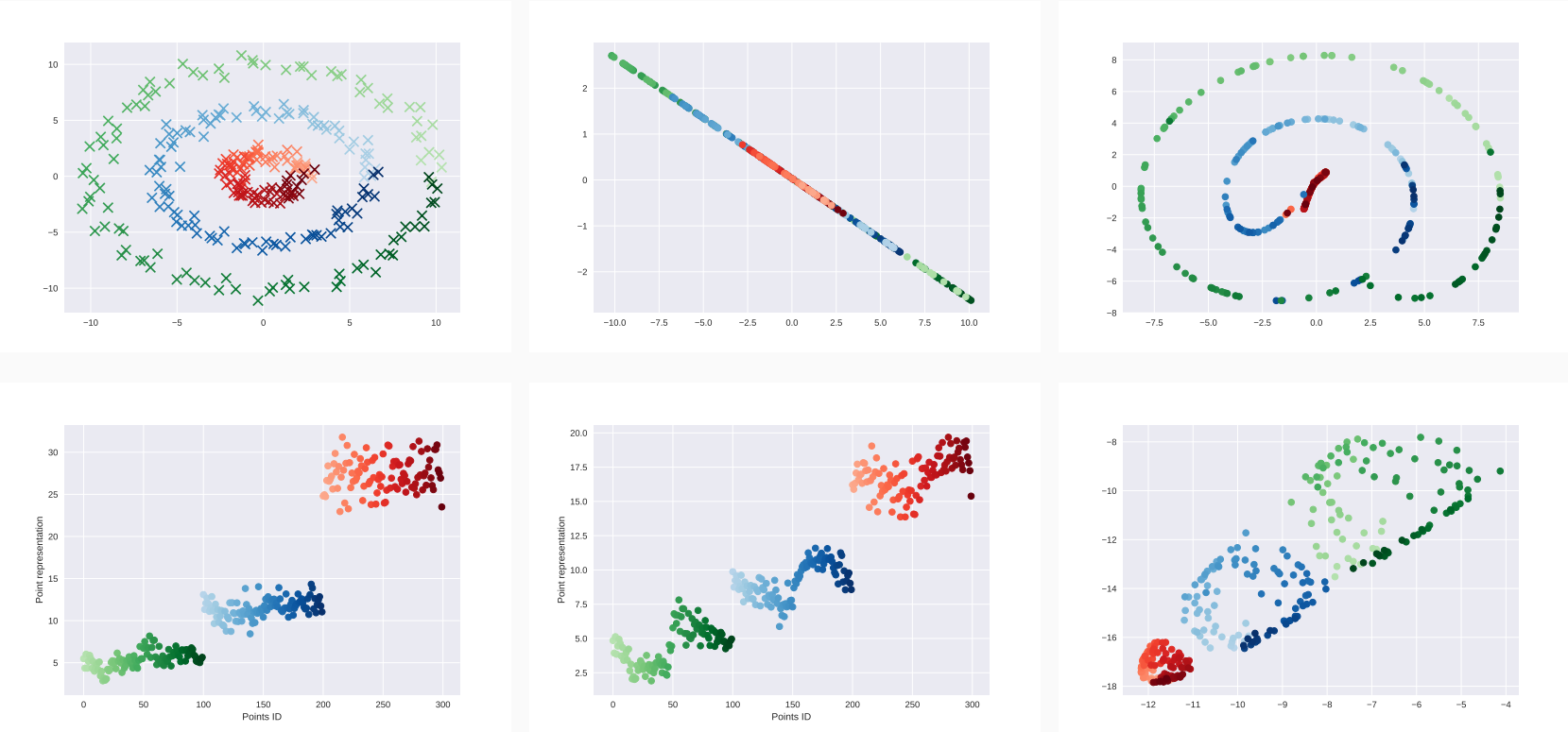


Figure 3: KAE performance on concentric circles

Molecular Data (Graphs): Unsupervised Task

Table 1: MSREs on Test Metabolites

Dimension	AE (sigmoid)	AE (relu)	KAE
5	99.81	96.62	**76.38**
10	87.36	84.02	**65.76**
25	72.31	68.77	**51.63**
50	63.00	58.29	**40.72**
100	55.43	48.63	**36.27**

Molecular Data (Graphs): Supervised Task

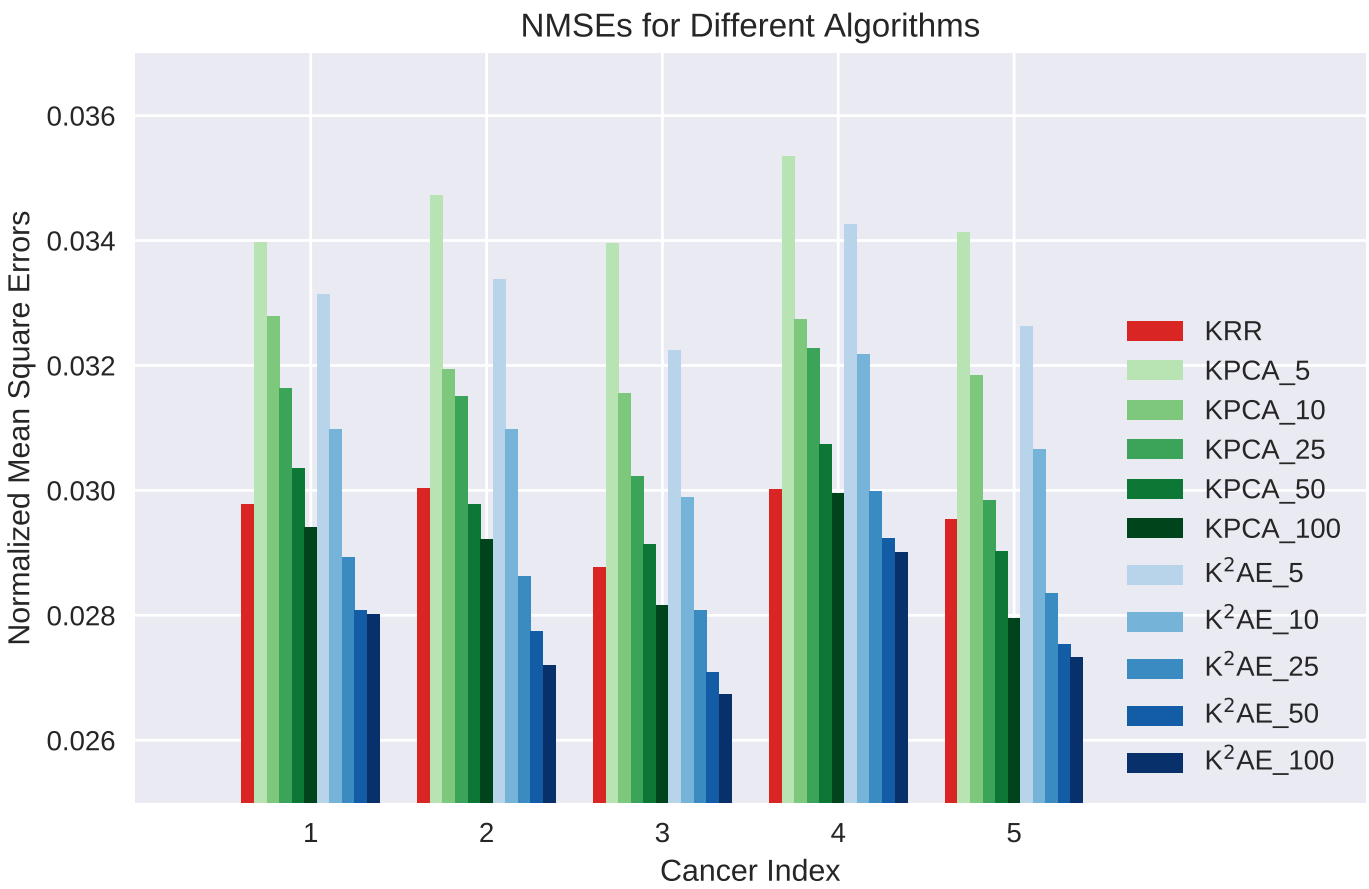


Figure 4: Performance of the different strategies on 5 cancers

Outline

Conclusion & Future Work

- Flexible tool for Representation Learning
- Advantages from AEs and Kernel Methods
- Extension of standard AEs to any type of data
- Connection with Kernel PCA

- Combine with a supervised criterion
- Consider another loss / optimization strategy

Generalization Bound

2-layer KAE on data bounded in norm by M, with:

- internal layer of size p
- encoder $f \in \mathcal{H}_1$ such that $\|f\| \leq s$
- decoder $g \in \mathcal{H}_2$ such that $\|g\| \leq t$, with Lipschitz constant L

Then it holds:

$$\epsilon(\hat{g}_n \circ \hat{f}_n) - \epsilon^* \leq C_0 LMst\sqrt{\frac{Kp}{n}} + 24M^2\sqrt{\frac{\log(2)/\delta}{2n}}.$$

with $\epsilon(g \circ f) = \mathbb{E}_X \|X - g \circ f(X)\|^2_{\mathcal{X}_0}$

Connection with KPCA (Proof)

- $X \in \mathbb{R}^{n \times d}$
- $Y = f(X) = XX^T A \in \mathbb{R}^{n \times p}, \quad A \in \mathbb{R}^{n \times p}$
- $Z = g(Y) = YY^T B \in \mathbb{R}^{n \times d}, \quad B \in \mathbb{R}^{n \times d}$
- Goal: $\min_{A,B} \quad \|X - Z\|_{Fr}^2 = \sum_{i=1}^{n} \|x_i - z_i\|_2^2$

SVD (thin with $d < n$):

- $X = U_d \, \overline{\Sigma}_d \, {V_d}^T$
- $Y = U_d \, {\overline{\Sigma}_d}^2 \, {U_d}^T \, A$
- $Z = U_d \, {\overline{\Sigma}_d}^2 \, {U_d}^T \, A \, A^T \, U_d \, {\overline{\Sigma}_d}^2 \, {U_d}^T \, B$

Eckart-Young:

$$Z^* = U_d \, \overline{\Sigma}_p \, {V_d}^T$$

Sufficient:

$$A = U_p \, {\overline{\Sigma}_p}^{-\frac{3}{2}} \in \mathbb{R}^{n \times p} \qquad B = U_d \, {V_d}^T \in \mathbb{R}^{n \times d}$$