Kernel PCA and Relevant Dimension Estimation

Lecture 11 – Machine Learning 2 (SS 2024), TU Berlin

Pat Chormai
(2024.07.04)

Recap: Kernel Methods

Linear model on nonlinear feature space (through $\phi : \mathcal{X} \to \mathcal{F}$)

$$f(\boldsymbol{x}) = \langle \boldsymbol{\beta}, \phi(\boldsymbol{x}) \rangle_{\mathcal{F}},$$

Choose the model class from the span of the data (in the feature space $\mathcal{F}$)

$$\boldsymbol{\beta} = \sum_{i=1}^{N} \alpha_i \phi(\boldsymbol{x}_i).$$

Then,

$$f(\boldsymbol{x}) = \sum_i \alpha_i \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) \rangle_{\mathcal{F}}$$

Recap: Kernel Trick

Challenge: finding $\phi(\cdot)$ and computing $\phi(\boldsymbol{x})$ might be difficult.

Positive Definite Kernels (p.d.)[1]

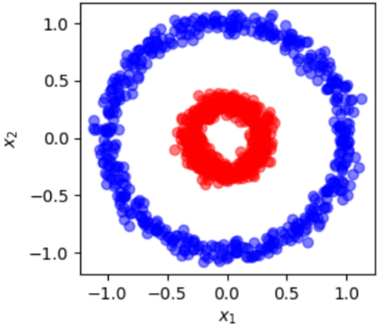
If $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a p.d. kernel function, then

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$$

Consequently, if our algorithm only relies on the dot product between data points, then

- no need to compute $\phi(\boldsymbol{x})$ explicitly

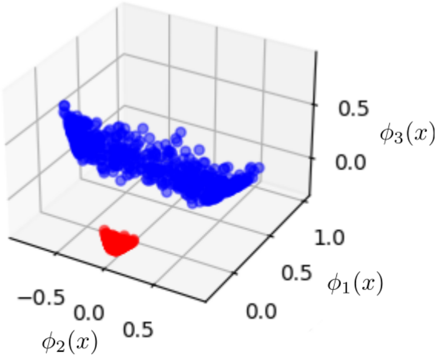
[1] Associated kernel matrix K (with $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$) is positive semi-definite.



Input Space

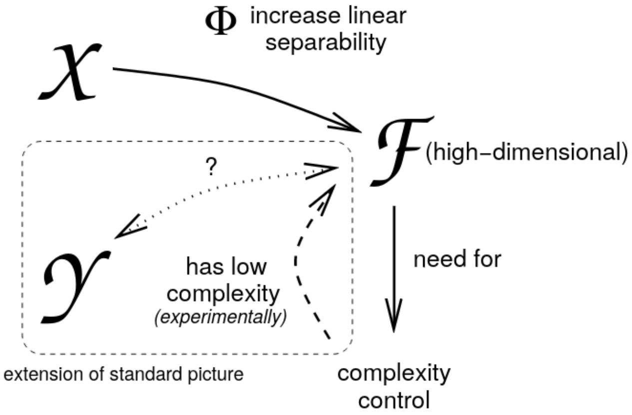
$$\phi(\boldsymbol{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^\top$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^\top \boldsymbol{x}')^2$$



Feature Space

How does the transformation relate to 'label' information?



[Braun et al., 2008]

Outline

- Kernel PCA (Schölkopf et al. 1998)
- Relevant Dimension Estimation (Braun et al. 2008)

Recap: Principal Component Analysis

Let $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d\}$ and assume that $\sum_i \boldsymbol{x}_i = 0$

Formulation

$$\min_{\boldsymbol{u} \in \mathbb{R}^d} \sum_{i=1}^{N} \|\boldsymbol{x}_n - \boldsymbol{u}\boldsymbol{u}^\top \boldsymbol{x}_n\|_2^2 \qquad \text{subject to} \qquad \boldsymbol{u}^\top \boldsymbol{u} = 1$$

Solution: 1st eigenvector of $\Sigma = \frac{1}{N} \sum_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$

Kernel PCA

Let $\phi : \mathcal{X} \to \mathcal{F}$ and assume that $\sum_i \phi(\boldsymbol{x}_i) = 0$

Formulation

Perform PCA on

$$\hat{\Sigma} = \frac{1}{N} \sum_i \phi(\boldsymbol{x}_i) \phi(\boldsymbol{x}_i)^\top$$

Kernel PCA: Solution[1]

Let $K \in \mathbb{R}^{N \times N}$ be the kernel matrix associated to a p.d. kernel function

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

Properties of Kernel Matrix K

- $K = K^\top \rightsquigarrow$ orthogonal eigenvectors (Spectral Theorem)
- K is positive semi-definite $\rightsquigarrow$ all eigenvalues $\lambda_n(K) \geq 0$

From Eigenvalue Decomposition,

$$K = U \Lambda U^\top,$$

where the columns of $U \in \mathbb{R}^{N \times N}$ are its eigenvectors and $\Lambda_{\tau\tau} = \lambda_\tau(K)$.

[1] We only state results and refer to the original paper (Schölkopf et al. 1998) or (Bishop 2007, Chapter 12.3) for the derivation.

Kernel PCA: Solution (cont.)

Let $\boldsymbol{u}_\tau \in \mathbb{R}^N$ be the τth eigenvector of K and define $\boldsymbol{\alpha}^{(\tau)} = \frac{1}{\sqrt{\lambda_\tau(K)}}\boldsymbol{u}_\tau$

τth Eigenvector of Kernel PCA $\hat{\Sigma}$

$$\boldsymbol{v}_\tau = \sum_i \alpha_i^{(\tau)} \phi(\boldsymbol{x}_i)$$

Computing Principal Components

Consider a datapoint $\boldsymbol{x}$ and the τth eigenvector of $\hat{\Sigma}$

τth Principal Component of Datapoint $\boldsymbol{x}$

$$
\begin{aligned}
\langle \boldsymbol{v}_\tau, \phi(\boldsymbol{x}) \rangle &= \left\langle \sum_i \alpha_i^{(\tau)} \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) \right\rangle \\
&= \sum_i \alpha_i^{(\tau)} \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}) \rangle \\
&= \sum_i \alpha_i^{(\tau)} k(\boldsymbol{x}_i, \boldsymbol{x})
\end{aligned}
$$

Relaxing Centering Assumption

Consider a kernel function k with its associated feature map ϕ and kernel matrix K
Let $\tilde{\phi}(\boldsymbol{x}) = \phi(\boldsymbol{x}) - \frac{1}{N}\sum_i \phi(\boldsymbol{x}_i)$ be the **centered feature map**

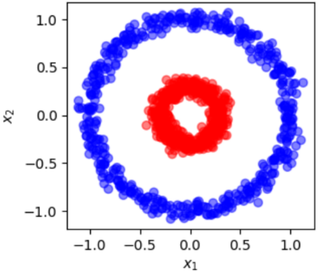
Kernel Matrix $\tilde{K}$ Cerresponding to Centered Feature Map

$$\tilde{K} = K - \mathbf{1}_N K - K\mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N,$$

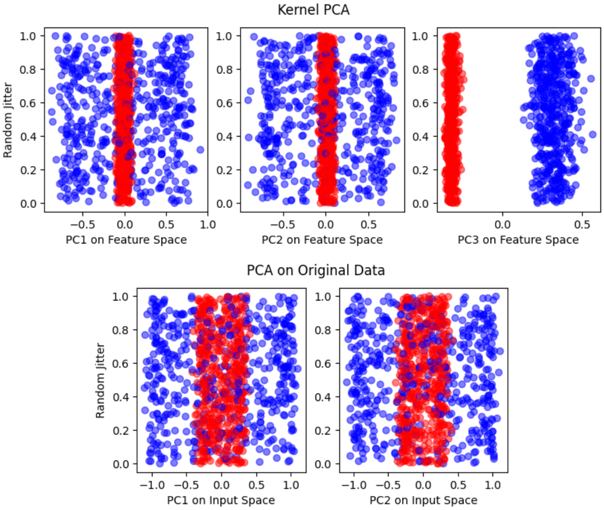
where $\mathbf{1}_N \in \{\frac{1}{N}\}^{N\times N}$.

Derivation Sketch[1]: Expand $\langle \tilde{\phi}(\boldsymbol{x}), \tilde{\phi}(\boldsymbol{x}') \rangle$ and write each term with $k(\cdot,\cdot)$

[1] See Bishop 2007, Chapter 12.3 for the complete derivation


x_2
x_1

Input Space

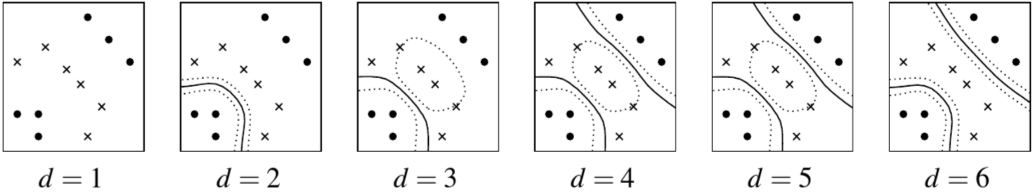

Kernel PCA
Random jitter
PC1 on Feature Space
PC2 on Feature Space
PC3 on Feature Space
PCA on Original Data
Random jitter
PC1 on Input Space
PC2 on Input Space

Remarks on Aspects of PCA and Kernel PCA

	PCA	Kernel PCA
Eigenvectors of	$\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$	$\sum_i \tilde{\phi}(\boldsymbol{x}_i)\tilde{\phi}(\boldsymbol{x}_i)^\top$
τth Principal Component	$\boldsymbol{u}_\tau^\top \boldsymbol{x}$	$\sum_i \alpha_i^{(\tau)} k(\boldsymbol{x}_i, \boldsymbol{x})$
Reconstruction	$\sum_\tau \boldsymbol{u}_\tau \boldsymbol{u}_\tau^\top \boldsymbol{x}$	*not straightforward*[1]

[1] See Mika et al. 1998

Observation: Only Few Components Needed to Preserve Decision Boundary



(Montavon et al. 2011)

Projecting Label Information on Kernel PCA Components

Consider a vector $\boldsymbol{y} \in \mathbb{R}^N$ (each entry corresponding to $\boldsymbol{x}_i$)
Recall $K = U\Lambda U^\top \in \mathbb{R}^{N\times N}$ whose τth column is denoted as $\boldsymbol{u}_\tau$

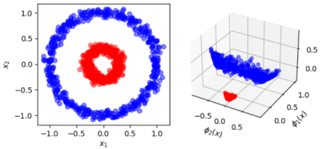
Lemma 1[1] Projection of Y onto Leading d Kernel PCA Components

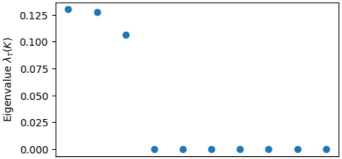
$$\sum_{\tau=1}^{d} \boldsymbol{u}_\tau \boldsymbol{u}_\tau^\top \boldsymbol{y}$$

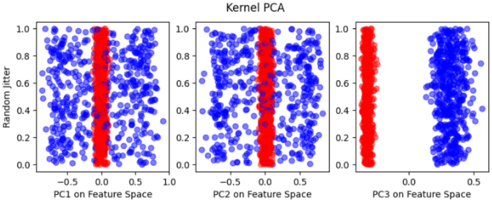
Definition τth Kernel PCA Coefficient

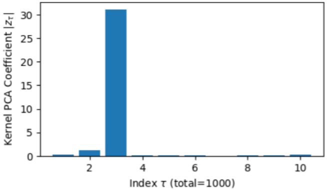
$$z_\tau = \boldsymbol{u}_\tau^\top \boldsymbol{y}$$

[1] Braun et al. 2008


x_1
x_2
$\phi_2(x)$
$\phi_1(x)$


Eigenvalue $\lambda_\tau(K)$


Kernel PCA
Random Jitter
PC1 on Feature Space
PC2 on Feature Space
PC3 on Feature Space


Kernel PCA Coefficient $|z_\tau|$
Index τ (total=1000)

Relevant Information

Assumption Label Corrupted with Additive Noise N_i

$$Y_i = g(X_i) + N_i$$

Definition Relevant Information g

$$g(X) = \mathbb{E}[Y|X] \qquad \text{(Population)}$$

$$G = (g(X_1), \ldots g(X_N)) \qquad \text{(Sample)}$$

Remarks

- (regression): $\mathbb{E}[Y|X]$ is the true function.
- (binary classification): $\mathbb{E}[Y|X] = P(Y = 1|X) - P(Y = -1|X)$

Theoretical Result

Theorem 1 (Braun et al. 2008) Upperbound on Contribution of Eigenvector on Relevant Information

If the learning problem can be represented by the kernel asymptotically, then

$$\frac{1}{N}|\boldsymbol{u}_\tau^\top G| \leq \lambda_\tau C(N) + E(N),$$

where $C(N), E(N)$ are some constants and $E(N) \to 0$ as $N \to 0$.

Interpretation: the relevant information about Y is contained in the leading kernel PCA directions up to a small error.

Relevant Dimension Estimation (RDE)

Formulation

Estimate number of dimensions $d \ll N$ such that

$$\forall i > d : \quad |\boldsymbol{u}_i^\top G| \approx 0$$

Implication: $\forall i > d$, $\boldsymbol{u}_i^\top Y$ captures the noise.

RDE by Fitting a Two-Component Model

Assumption[1]**:** Modeling Kernel Coefficients with Two Zero-Mean Gaussians

Let d be a cut-off point that splits kernel coefficients z_τ's into two parts (*relevant information and noise*)

$$z_\tau \sim \begin{cases} \mathcal{N}(0, \sigma_1^2) & 1 \leq \tau \leq d, \\ \mathcal{N}(0, \sigma_2^2) & d < \tau \leq N \end{cases}$$

where $\sigma_1^2 = \frac{1}{N} \sum_{i=1}^{d} z_i^2$ and $\sigma_2^2 = \frac{1}{N-d} \sum_{i=d+1}^{N} z_i^2$.

[1] See Braun et al. 2008 for a more general approach via Leave-One-Out Cross-Validation

RDE by Fitting a Two-Component Model (cont.)

The negative log-likehood $\ell(d)$ is proportional to

$$-\log \ell(d) \propto \frac{d}{n} \log \sigma_1^2 + \frac{n-d}{d} \log \sigma_2^2$$

Estimated Relevant Dimension via Maximum Likelihood

$$\hat{d} = \operatorname*{arg\,min}_{1 \leq d < N} -\log \ell(d)$$

Estimating Noise Level

Project labels $\boldsymbol{y}$ to onto the first $\hat{d}$ kernel PCA components

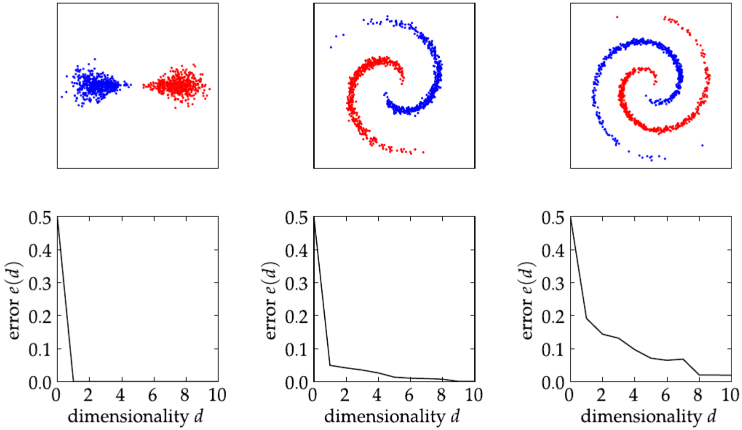
$$\hat{\boldsymbol{y}} = \sum_{\tau=1}^{\hat{d}} \boldsymbol{u}_\tau \boldsymbol{u}_\tau^\top \boldsymbol{y}$$

Estimate Noise Level

$$e(\hat{d}) = \sum_{i=1}^{N} L(\hat{y}_i, y_i),$$

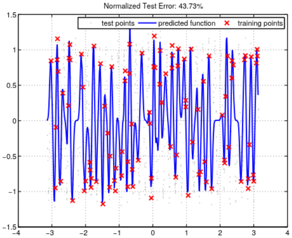
where L is a loss function

Relationship between Data Complexity and Noise Level

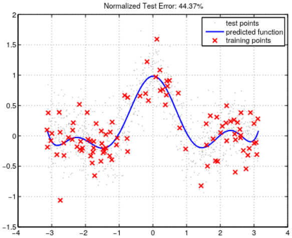


(Montavon et al. 2011)

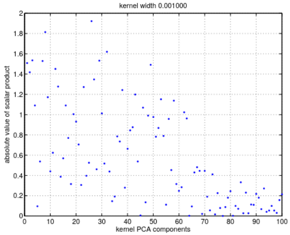
Applications of RDE: Dataset Assessment



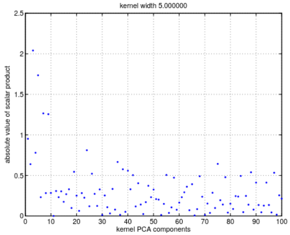
(a) A complex data set.



(b) A noisy data set.



(c) Kernel PCA coefficients for the complex data set.



(d) Kernel PCA coefficients for the noisy data set.

Applications of RDE: Dataset Assessment (cont.)

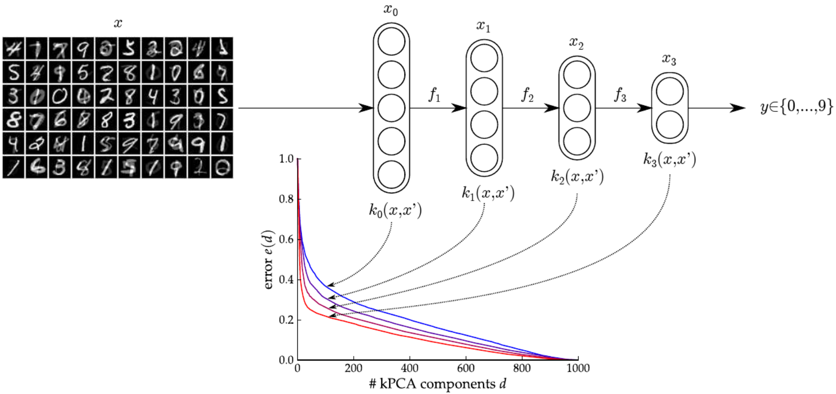
Dataset	RDE Method	Estimated Dimension $\hat{d}$	Noise Level $e(\hat{d})$
Complex Dataset	TCM	50	16.07%
	LOO-CV	25	40.59 %
Noisy Dataset	TCM	9	40.71%
	LOO-CV	9	40.71 %

Applications of RDE: Dataset Assessment (cont.)

Data Scenario	$\hat{d}$	$e(\hat{d})$	Possible Mitigation Strategy
Noisy	small	large	find ways to reduce label noise
Complex	large	small	model selection[1] and/or acquire more data

[1] e.g., incorporate domain knowledge in order to determine a more appropriate kernel function and its parameters

Application: Layer-wise Analysis of Neural Network Representation



layer-wise reduction of both noise and dimensionality
(Montavon 2013)

Summary

- Kernel PCA
- Relevant Dimension Estimation and its applications on
  - Data Assessment
  - Analysis of Neural Network Representation

References

Bishop, Christopher M. (2007). *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer. Chap. 12.3.

Braun, Mikio L. et al. (2008). "On Relevant Dimensions in Kernel Feature Spaces". In: *J. Mach. Learn. Res.* 9, pp. 1875–1908.

Mika, Sebastian et al. (1998). "Kernel PCA and de-noising in feature spaces". In: *Advances in neural information processing systems* 11.

Montavon, Grégoire (2013). "On layer-wise representations in deep neural networks". PhD thesis. Technische Universität Berlin.

Montavon, Grégoire et al. (2011). "Kernel Analysis of Deep Networks.". In: *Journal of Machine Learning Research* 12.9.

Schölkopf, Bernhard et al. (1998). "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". In: *Neural Computation* 10.5, pp. 1299–1319. DOI: 10.1162/089976698300017467.