ASSIGNMENT 3: PCA AND KERNEL PCA

Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

Institute for
Machine Learning

Contact

Heads:
Thomas Adler,
Philipp Renz,
Andreas Radler

Institute for Machine Learning
Johannes Kepler University
Altenberger Str. 69
A-4040 Linz

E-Mail: `{adler,renz,radler}@ml.jku.at`
Institute Homepage

Copyright statement:

Agenda

- Recap of PCA
- Intuition of Kernel PCA
- Example of Kernel PCA
- Derivation of Kernel PCA

Lecture notes

Mathematics for Machine Learning, [2018, Marc Peter Deisenroth at. el.]

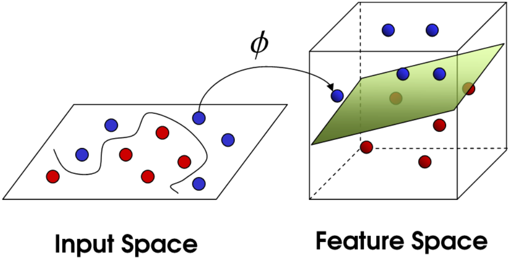
Notation

Notation as in the lecture, i.e. we are given a data matrix $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^\top$, where each $\boldsymbol{x}_i \in \mathbb{R}^m$ for $1 \leq i \leq n$. Thus, $\boldsymbol{X}$ is an $n \times m$ matrix.

Recap of Principal Component Analysis

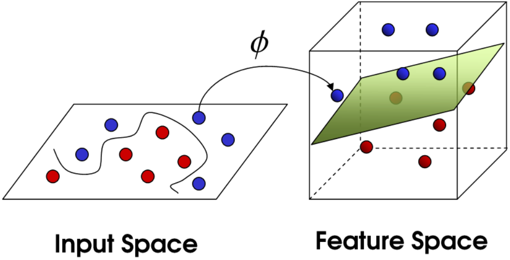
- Intuition: find directions of **largest variances** in $\boldsymbol{X}$
- Assume $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ is centered, i.e. $\sum_{i=1}^{n} \boldsymbol{x}_i = \boldsymbol{0}$
- Compute the **covariance matrix** $\boldsymbol{C} = \frac{1}{n} \boldsymbol{X}^\top \boldsymbol{X}$
- Compute its **eigendecomposition** $\boldsymbol{C} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^\top$
  - $\boldsymbol{\Lambda}$ is the diagonal matrix of **eigenvalues** $\mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ in descending order, i.e. $\lambda_1 \geq \cdots \geq \lambda_m$
  - $\boldsymbol{U}$ is the matrix of respective **eigenvectors** $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)$
  - $\boldsymbol{U}$ is **orthogonal** (i.e. $\boldsymbol{U}^{-1} = \boldsymbol{U}^\top$) because $\boldsymbol{C}$ is **symmetric**
- The eigenvector $\boldsymbol{u}_i$ is the i-th **principal component** (PC)
- The eigenvalue λ_i is the **variance** along the direction $\boldsymbol{u}_i$
  - $\boldsymbol{U}$ uncorrelates $\boldsymbol{X}$ because $\frac{1}{n}(\boldsymbol{X}\boldsymbol{U})^\top \boldsymbol{X}\boldsymbol{U} = \boldsymbol{\Lambda}$
  - $\boldsymbol{\Lambda}$ is the (diagonal) covariance matrix of $\boldsymbol{X}\boldsymbol{U}$

Intuition of Kernel PCA (1/2)



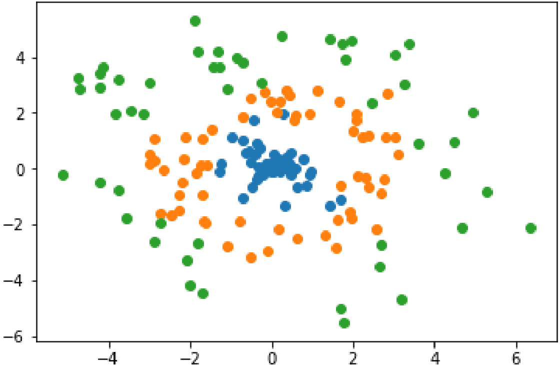
- PCA is a linear method
- However, data are often not linearly separable
- **Idea**: map data into high-dimensional space where it becomes linearly separable and apply PCA there

Intuition of Kernel PCA (2/2)



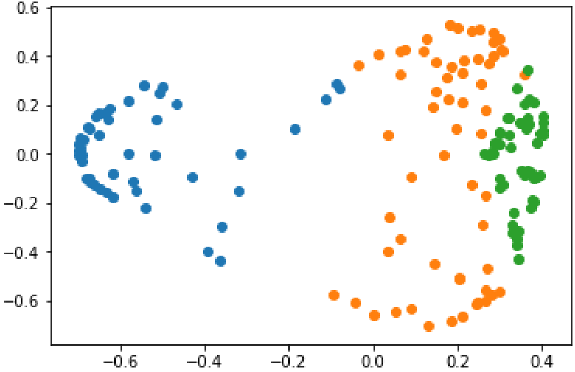
- This space is called the **feature space**
- The mapping Φ is called the **feature map**
- Use a **kernel function** instead of explicitly computing Φ
- This approach is called the **kernel trick**

Example of Kernel PCA (1/2)



- Data cannot be separated linearly

Example of Kernel PCA (2/2)



- Data can be well separated linearly using only the first PC

Definitions and Assumptions

- Let $\mathcal{V}$ denote the **feature space**
- Let $\boldsymbol{\Phi} : \mathbb{R}^m \to \mathcal{V}$ denote the **feature map**
- Let $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ denote the **kernel**

$$k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\Phi}^\top(\boldsymbol{x})\boldsymbol{\Phi}(\boldsymbol{y})$$

- Assume the data is **centered** in $\mathcal{V}$, i.e. $\sum_{i=1}^n \boldsymbol{\Phi}(\boldsymbol{x}_i) = \boldsymbol{0}$
- The covariance matrix $\boldsymbol{C} \in \mathbb{R}^{\dim\mathcal{V}\times\dim\mathcal{V}}$ is

$$\boldsymbol{C} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\Phi}(\boldsymbol{x}_i)\boldsymbol{\Phi}^\top(\boldsymbol{x}_i)$$

- The **Gram matrix** $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is defined as

$$K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\Phi}^\top(\boldsymbol{x}_i)\boldsymbol{\Phi}(\boldsymbol{x}_j)$$

Eigenvector in Feature Space

- Note that possibly $\dim \mathcal{V} > n$
- In fact, even $\dim \mathcal{V} = \infty$ is possible
- Therefore, consider only eigenvectors $\boldsymbol{u} \in \mathcal{V}$ which are in the span of the mapped data $\boldsymbol{\Phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\Phi}(\boldsymbol{x}_n)$
- Thus, the considered eigenvectors can be written as a linear combination of the mapped samples

$$\boldsymbol{u} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\Phi}(\boldsymbol{x}_i)$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$

Eigenvalue Equation in Feature Space

$$\boldsymbol{C}\boldsymbol{u} = \lambda \boldsymbol{u}$$

$$\boldsymbol{C}\sum_{i=1}^{n} \alpha_i \boldsymbol{\Phi}(\boldsymbol{x}_i) = \lambda \sum_{i=1}^{n} \alpha_i \boldsymbol{\Phi}(\boldsymbol{x}_i)$$

$$\boldsymbol{\Phi}^\top(\boldsymbol{x}_l)\boldsymbol{C}\sum_{i=1}^{n} \alpha_i \boldsymbol{\Phi}(\boldsymbol{x}_i) = \lambda \sum_{i=1}^{n} \alpha_i \boldsymbol{\Phi}^\top(\boldsymbol{x}_l)\boldsymbol{\Phi}(\boldsymbol{x}_i)$$

$$\frac{1}{n}\boldsymbol{\Phi}^\top(\boldsymbol{x}_l)\left(\sum_{j=1}^{n} \boldsymbol{\Phi}(\boldsymbol{x}_j)\boldsymbol{\Phi}^\top(\boldsymbol{x}_j)\right)\sum_{i=1}^{n} \alpha_i \boldsymbol{\Phi}(\boldsymbol{x}_i) = \lambda \left[\boldsymbol{K}\boldsymbol{\alpha}\right]_l$$

$$\boldsymbol{K}^2\boldsymbol{\alpha} = n\lambda \boldsymbol{K}\boldsymbol{\alpha}$$

Eigenvalue Equation of the Gram Matrix

- We can solve the eigenvalue equation in feature space by solving the eigenvalue equation of the Gram matrix

$$\boldsymbol{K}\boldsymbol{\alpha} = n\lambda\boldsymbol{\alpha}$$

- We have to normalize $\boldsymbol{\alpha} \leftarrow \left(\|\boldsymbol{\alpha}\|\sqrt{n\lambda}\right)^{-1} \boldsymbol{\alpha}$ to account for an orthonormal basis in feature space because

$$1 = \boldsymbol{u}^\top \boldsymbol{u} = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \boldsymbol{\Phi}^\top(\boldsymbol{x}_i)\boldsymbol{\Phi}(\boldsymbol{x}_j) = \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} = n\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}$$

- Project $\boldsymbol{x}$ onto a principal component $\boldsymbol{u}$ by

$$\boldsymbol{u}^\top \boldsymbol{\Phi}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \boldsymbol{\Phi}^\top(\boldsymbol{x}_i)\boldsymbol{\Phi}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x})$$

The Kernel Trick

- The **feature map** $\Phi(\cdot)$ need never be computed explicitly
- Instead, we reformulated everything which requires explicit computation using only the **kernel** $k(\cdot, \cdot)$
- This is known as the **kernel trick**
- It can be applied not only to PCA but also to various other methods (e.g. support vector machines)
- It is a popular device to make a linear method non-linear
- The linear method is applied in a high-dimensional space that has a non-linear relation to the input space

Centering the Gram Matrix

- We assumed that the data are centered in feature space
- We can center K_{ij} by

$$\left(\boldsymbol{\Phi}(\boldsymbol{x}_i) - \frac{1}{n}\sum_{p=1}^{n}\boldsymbol{\Phi}(\boldsymbol{x}_p)\right)^{\top}\left(\boldsymbol{\Phi}(\boldsymbol{x}_j) - \frac{1}{n}\sum_{q=1}^{n}\boldsymbol{\Phi}(\boldsymbol{x}_q)\right)$$

$$= \boldsymbol{\Phi}^{\top}(\boldsymbol{x}_i)\boldsymbol{\Phi}(\boldsymbol{x}_j) - \frac{1}{n}\sum_{q=1}^{n}\boldsymbol{\Phi}^{\top}(\boldsymbol{x}_i)\boldsymbol{\Phi}(\boldsymbol{x}_q)$$

$$-\frac{1}{n}\sum_{p=1}^{n}\boldsymbol{\Phi}^{\top}(\boldsymbol{x}_p)\boldsymbol{\Phi}(\boldsymbol{x}_j) + \frac{1}{n^2}\sum_{p=1}^{n}\sum_{q=1}^{n}\boldsymbol{\Phi}^{\top}(\boldsymbol{x}_p)\boldsymbol{\Phi}(\boldsymbol{x}_q)$$

- In matrix form, this is

$$\boldsymbol{K} - \frac{1}{n}\boldsymbol{K}\mathbf{1}\mathbf{1}^{\top} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\top}\boldsymbol{K} + \frac{\mathbf{1}^{\top}\boldsymbol{K}\mathbf{1}}{n^2}$$

Summary

- Choose a kernel $k(\cdot, \cdot)$
- Compute the Gram matrix $\boldsymbol{K}$ by $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Center the Gram matrix in feature space by

$$\boldsymbol{K} \leftarrow \boldsymbol{K} - \frac{1}{n}\boldsymbol{K}\mathbf{1}\mathbf{1}^\top - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\boldsymbol{K} + \frac{\mathbf{1}^\top\boldsymbol{K}\mathbf{1}}{n^2}$$

- Compute eigenvectors $\boldsymbol{\alpha}$ and eigenvalues $n\lambda$
- Normalize eigenvectors $\boldsymbol{\alpha} \leftarrow \left(\|\boldsymbol{\alpha}\|\sqrt{n\lambda}\right)^{-1}\boldsymbol{\alpha}$
- Given a new point $\boldsymbol{x}$, define $\boldsymbol{k} = (k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n))^\top$
- Center it by $\boldsymbol{k} \leftarrow \boldsymbol{k} - \frac{1}{n}\boldsymbol{K}\mathbf{1} - \frac{1}{n}\mathbf{1}^\top\boldsymbol{k} + \frac{1}{n^2}\mathbf{1}^\top\boldsymbol{K}\mathbf{1}$
- Project $\boldsymbol{k}$ onto principal component by $\boldsymbol{\alpha}^\top\boldsymbol{k}$