Foundations of Machine Learning
AI2000 and AI5000

FoML-30
PCA - reconstruction interpretation

Dr. Konda Reddy Mopuri
Department of AI, IIT Hyderabad
July-Nov 2025

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

DiL
Data-driven Intelligence
& Learning Lab

So far in FoML

- Intro to ML and Probability refresher
- MLE, MAP, and fully Bayesian treatment
- Supervised learning
    a. Linear Regression with basis functions
    b. Bias-Variance Decomposition
    c. Decision Theory - three broad classification strategies
    d. Neural Networks
- Unsupervised learning
    a. K-Means, Hierarchical, and GMM for clustering

For today

- PCA - different interpretation based on reconstruction error
- Nonlinear PCA

PCA via minimizing reconstruction error

- Finding the transformation that minimizes $\frac{1}{N}\sum_{n=1}^{N}||\mathbf{x}_n - \tilde{\mathbf{x}}_n||_2$

PCA via minimizing reconstruction error

- Finding the transformation that minimizes $\frac{1}{N}\sum_{n=1}^{N}||\mathbf{x}_n - \tilde{\mathbf{x}}_n||_2$

$\tilde{\mathbf{x}}_n$ Is generated by the lower-dim latent variable $\mathbf{z}_n$

We restrict to linear models $\tilde{\mathbf{x}}_n = \mathbf{U}_M \mathbf{z}_n + \bar{\mathbf{x}}$

Basis

basis coeffs/latent variables

PCA via minimizing reconstruction error

- Represent the data in a new orthonormal basis (M-dimensional)

$$\{\mathbf{u}_i\}_{i=1}^{D}$$

In the new basis $\mathbf{x}_n =$

PCA via minimizing reconstruction error

- For the lower-dim reconstruction, use the first M elements from the basis
    - And a shared/common offset for the rest

$$\tilde{\mathbf{x}}_n =$$

PCA via minimizing reconstruction error

- The difference

$$\mathbf{x}_n - \tilde{\mathbf{x}}_n =$$

PCA via minimizing reconstruction error

- Find the optima for b_i and u_i

PCA via minimizing reconstruction error

- Find the optima for b_i and u_i

PCA via minimizing reconstruction error

$$\sum_{i=1}^{N} ||\mathbf{x}_n - \tilde{\mathbf{x}}_n||_2 = \sum_{i=M+1}^{D} \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i$$

- Solve for ui with constraint $\mathbf{u}_i^T \mathbf{u_i} = 1$
- Method of Lagrange multipliers → solving the eigen system of S
- D-M smallest eigenvalues and the corresponding eigenvectors are the solution

PCA via minimizing reconstruction error

$$\tilde{\mathbf{x}}_n = \mathbf{U}_M \mathbf{z}_n + \mathbf{U}_{M+1\rightarrow D}\mathbf{b}$$

$$\tilde{\mathbf{x}}_n = \mathbf{U}_M \mathbf{z}_n + \bar{\mathbf{x}}$$

PCA applications

- Compression, preprocessing, etc.
  - E.g. Eigenfaces

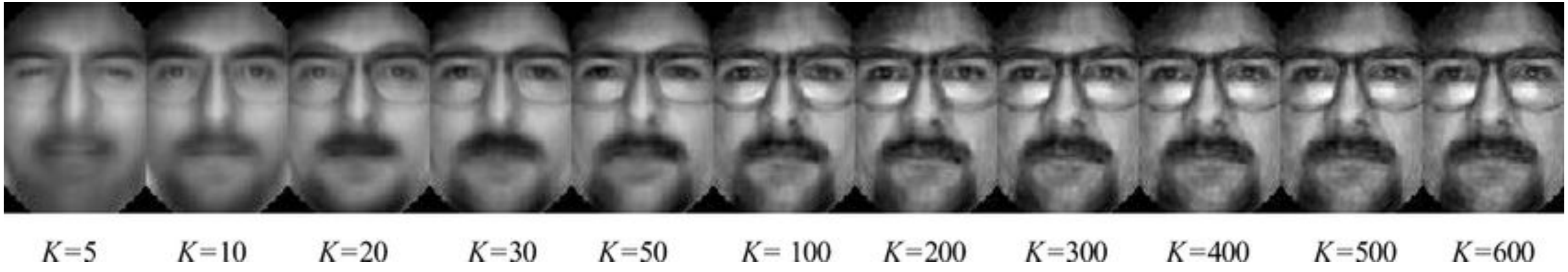
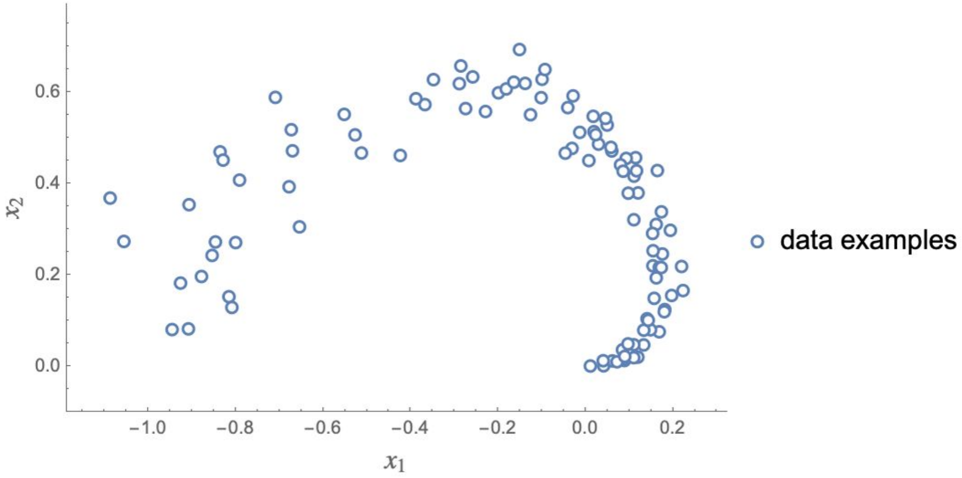


Figure: Xiaoou Tang et al.

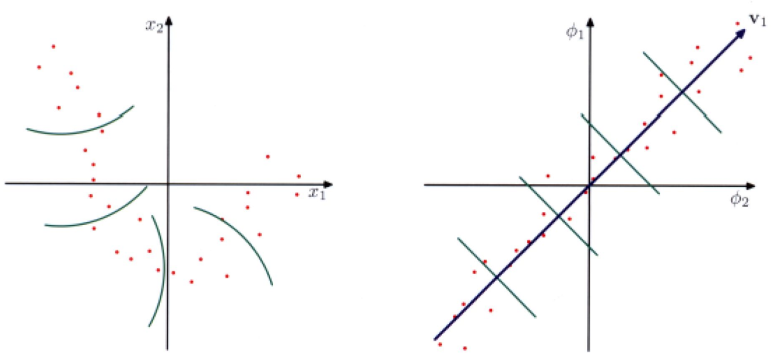
Nonlinear generalization to PCA
Kernel PCA

Manifold coordinates as Latent variables



$$\{x_1, x_2\} = \{t \cos(3\ t),\ t \sin(3\ t)\}$$

PCA via basis functions



- Apply nonlinear transformation on the D-dim data
- Perform standard PCA there
- → nonlinear PCA in the original D-dim space

Kernel PCA

$$\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}}$$

- We have to solve the eigen expansion of C
- But the goal is to avoid doing it in the feature space

Kernel PCA

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\left\{\phi(\mathbf{x}_n)^{\mathrm{T}}\mathbf{v}_i\right\} = \lambda_i\mathbf{v}_i$$

$$\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n).$$

- Eigenvectors can be represented as a linear combination of feature vectors

Kernel PCA

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\left\{\phi(\mathbf{x}_n)^{\mathrm{T}}\mathbf{v}_i\right\} = \lambda_i\mathbf{v}_i$$

$$\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n).$$

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}}\sum_{m=1}^{N} a_{im}\phi(\mathbf{x}_m) = \lambda_i\sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n).$$

Kernel PCA

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\left\{\phi(\mathbf{x}_n)^{\mathrm{T}}\mathbf{v}_i\right\} = \lambda_i\mathbf{v}_i$$

$$\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n).$$

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}}\sum_{m=1}^{N}a_{im}\phi(\mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}\phi(\mathbf{x}_n).$$

$$\frac{1}{N}\sum_{n=1}^{N}k(\mathbf{x}_l,\mathbf{x}_n)\sum_{m=1}^{m}a_{im}k(\mathbf{x}_n,\mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}k(\mathbf{x}_l,\mathbf{x}_n).$$

Kernel PCA

$$\frac{1}{N}\sum_{n=1}^{N} k(\mathbf{x}_l, \mathbf{x}_n) \sum_{m=1}^{m} a_{im} k(\mathbf{x}_n, \mathbf{x}_m) = \lambda_i \sum_{n=1}^{N} a_{in} k(\mathbf{x}_l, \mathbf{x}_n).$$

$$\mathbf{K}^2 \mathbf{a}_i = \lambda_i N \mathbf{K} \mathbf{a}_i \qquad \mathbf{K}\mathbf{a}_i = \lambda_i N \mathbf{a}_i$$

Kernel PCA

$$\tilde{\mathbf{x}} = \phi(\mathbf{x})^{\mathrm{T}}\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{x}_n) = \sum_{n=1}^{N} a_{in}k(\mathbf{x}, \mathbf{x}_n)$$

- M > D
  - No. of nonlinear PCs can exceed the original dimension D
  - However, it is <= N

Next

- Kernel Methods