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 $\frac{1}{N}\sum_{n=1}^{N} ||\mathbf{x}_n - \tilde{\mathbf{x}}_n||_2$

$$z_n = U_M^T x_n$$

PCA via minimizing reconstruction error

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

(M) $z_n \longrightarrow x_n$ (D)

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

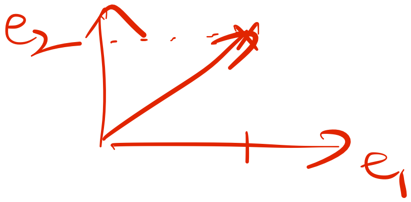
- Represent the data in a new orthonormal basis (M-dimensional)

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

$u_i^T u_j = 1$ $u_i^T u_j = 0$ $i \neq j$

In the new basis $\mathbf{x}_n = \sum_{i=1}^{D} z_{ni} u_i$

$z_{ni} = x_n^T u_i$



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

$u_1 \cdots u_D$

$$\tilde{\mathbf{x}}_n = \underbrace{\sum_{i=1}^{M} z_{ni} \underline{u_i}}_{\text{specific to } x_n} + \sum_{i=M+1}^{D} b_i \underline{u_i}$$

is not

PCA via minimizing reconstruction error

- The difference

$$\mathbf{x}_n - \tilde{\mathbf{x}}_n = \underbrace{\sum_{i=1}^{D} z_{ni} u_i}_{x_n} - \underbrace{\sum_{i=1}^{M} z_{ni} u_i - \sum_{i=M+1}^{D} b_i u_i}_{\tilde{x}_n}$$

$$=$$

$$z_{ni} = x_n^T u_i$$

PCA via minimizing reconstruction error

- Find the optima for b_i and u_i

$$J = \sum_{n=1}^{N} \left\| \sum_{i=M+1}^{D} (z_{ni}\underline{u_i} - b_i \underline{u_i}) \right\|^2$$

$$= \sum_{n=1}^{N} \left\| (z_{ni} - b_i)\underline{u_i} \right\|^2$$

$$= \sum_{n=1}^{N} \sum_{i=M+1}^{D} (z_{ni} - b_i)^2$$

$$\underset{b_i}{\operatorname{argmin}} \; J$$

$$\frac{\partial J}{\partial b_i} = \sum_{n=1}^{N} 2(z_{ni} - b_i) = 0$$

$$\sum_{n=1}^{N} z_{ni} = \sum_{n=1}^{N} b_i = N b_i$$

$$b_i = \frac{1}{N} \sum_{n=1}^{N} \underline{z_{ni}} = \frac{1}{N} \sum_{n=1}^{N} (\underline{x_n}^T \cdot \underline{u_i})$$

$$b_i = (\bar{x})^T \underline{u_i}$$

PCA via minimizing reconstruction error

- Find the optima for b_i and u_i

$$J = \sum_{n=1}^{N} \left\| \sum_{i=M+1}^{D} (x_{ni} - b_i)\underline{u_i} \right\|^2$$

$$J = \sum_{n=1}^{N} \left(\sum_{i=m+1}^{D} (x_{ni} - b_i)\underline{u_i} \right)^T \left[\sum_{j=m+1}^{D} (x_{nj} - b_j)\underline{u_j} \right]$$

$$= \sum_{n=1}^{N} \sum_{i=m+1}^{D} (x_{ni} - b_i)\underline{u_i}^T \sum_{j=m+1}^{D} (x_{nj} - b_j)\underline{u_j} = \sum_{n=1}^{N} \sum_{i=m+1}^{D} \sum_{j=M+1}^{D} (x_{ni} - b_i)(x_{nj} - b_j)\underline{u_i}^T \underline{u_j}$$

$$\begin{cases} u_i^T u_j = 1 & i = j \\ \quad = 0 & \text{else} \end{cases}$$

orthonormal

$$= \sum_{n=1}^{N} \sum_{i=M+1}^{D} (x_{ni} - b_i)^2 = \sum_{n=1}^{N} \sum_{i=m+1}^{D} \left\| \underline{x_n}^T \underline{u_i} - \underline{\bar{x}}^T \underline{u_i} \right\|_2^2$$

$$= \sum_{n=1}^{N} \sum_{i=m+1}^{D} \underline{u_i}^T (\underline{x_n}^T - \underline{\bar{x}})^T (\underline{x_n} - \underline{\bar{x}}) \underline{u_i} = \sum_{i=m+1}^{D} \underline{u_i}^T S \underline{u_i}$$

S - covariance of the data

$\bar{x}$ - mean data

PCA via minimizing reconstruction error

$$\underset{u_i}{\operatorname{argmin}} \sum_{i=1}^{N} ||\mathbf{x}_n - \tilde{\mathbf{x}}_n||_2 = \boxed{\sum_{i=M+1}^{D} \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i} + \lambda\left[1 - u_i^T u_i\right]$$

- 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

$Su_i = \lambda u_i$

1 to M eigenvectors

M+1 to D eigenvectors

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

DiL
Data-driven Intelligence & Learning Lab

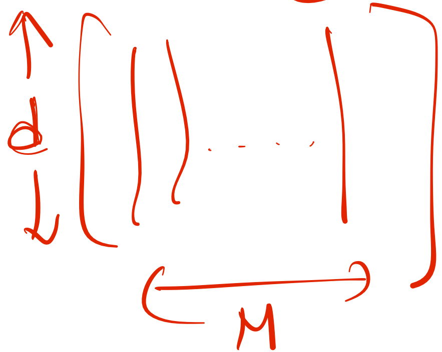
PCA via minimizing reconstruction error

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

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

$$\left[\frac{1}{N}\sum_{i=1}^{N} x_n\right] \forall i$$

$i = M+1$ to D



$M < d$

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

DiL
Data-driven Intelligence & Learning Lab

PCA applications

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

x_1, x_2, ..., x_n

$z_1 \rightarrow \lambda_1$, $z_2 \rightarrow \lambda_2$, ..., $z_N \rightarrow \lambda_N$

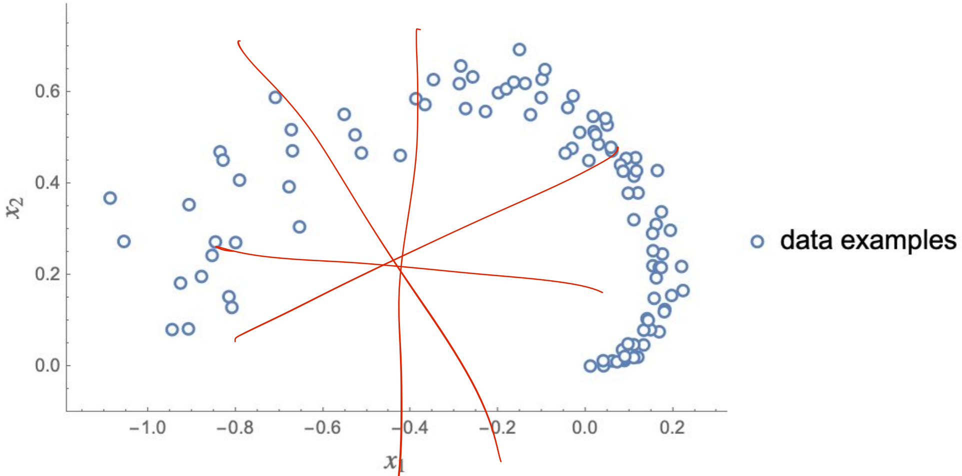
$M < D$

K=5 K=10 K=20 K=30 K=50 K= 100 K=200 K=300 K=400 K=500 K=600

Figure: Xiaoou Tang et al.

Nonlinear generalization to PCA
Kernel PCA

Manifold coordinates as Latent variables



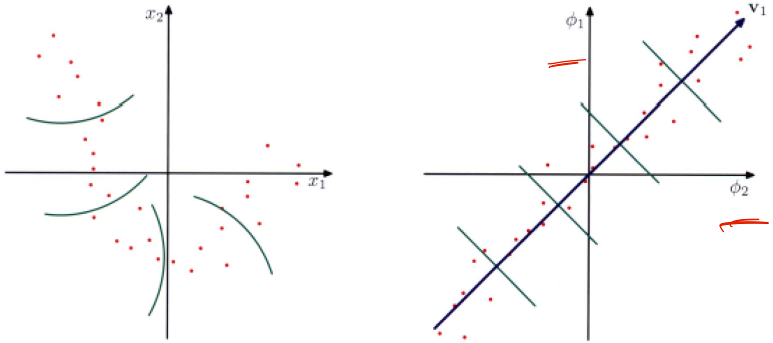
$$\{x_1, x_2\} = \{t \cos(3\ t),\ t \sin(3\ t)\}$$

Figure credits

PCA via basis functions

D

M



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

M > D

Kernel PCA

$$\phi(\underline{x}_1) = \begin{bmatrix} \phi_1(\underline{x}_1) \\ \phi_2(\underline{x}_1) \end{bmatrix}$$

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

$$C v_i = \lambda v_i$$

- 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

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

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

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

$\phi(x_l)^T$

$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$

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

Kernel PCA

$K_{ij} = k(\underline{x}_i, \underline{x}_j)$

$N \times N$

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

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

$\frac{\bar{a}_i}{N\lambda_i}$

Kernel PCA

v_i eigvec C

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

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

$N > D$

Next

- Kernel Methods