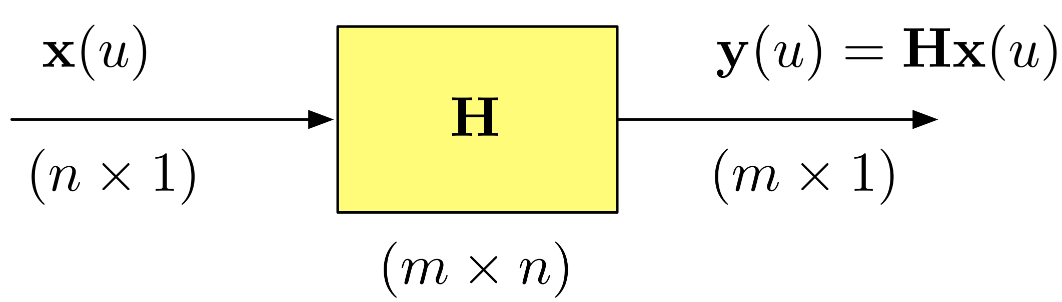
Principal Component Analysis

EE599 Deep Learning

Keith M. Chugg
Spring 2020

USC University of Southern California

Random Vectors



$$\mathbf{m}_{\mathbf{y}} = \mathbf{H}\mathbf{m}_{\mathbf{x}}$$

$$\mathbf{R}_{\mathbf{y}} = \mathbf{H}\mathbf{R}_{\mathbf{x}}\mathbf{H}^{\mathrm{t}}$$

$$\mathbf{K}_{\mathbf{y}} = \mathbf{H}\mathbf{K}_{\mathbf{x}}\mathbf{H}^{\mathrm{t}}$$

Special case

$$y(u) = \mathbf{b}^{\mathrm{t}}\mathbf{x}(u) \qquad (1 \times 1)$$

$$m_y = \mathbf{b}^{\mathrm{t}}\mathbf{m}_{\mathbf{x}}$$

$$\mathbb{E}\left\{y^2(u)\right\} = \mathbf{b}^{\mathrm{t}}\mathbf{R}_{\mathbf{x}}\mathbf{b}$$

$$\sigma_y^2 = \mathbf{b}^{\mathrm{t}}\mathbf{K}_{\mathbf{x}}\mathbf{b}$$

example math

$$\begin{aligned}
\mathbf{R}_{\mathbf{y}} &= \mathbb{E}\left\{\mathbf{y}(u)\mathbf{y}^{\mathrm{t}}(u)\right\} \\
&= \mathbb{E}\left\{(\mathbf{H}\mathbf{x}(u))(\mathbf{H}\mathbf{x}(u))^{\mathrm{t}}\right\} \\
&= \mathbb{E}\left\{\mathbf{H}\mathbf{x}(u)\mathbf{x}^{\mathrm{t}}(u)\mathbf{H}^{\mathrm{t}}\right\} \\
&= \mathbf{H}\mathbb{E}\left\{\mathbf{x}(u)\mathbf{x}^{\mathrm{t}}(u)\right\}\mathbf{H}^{\mathrm{t}} \\
&= \mathbf{H}\mathbf{R}_{\mathbf{x}}\mathbf{H}^{\mathrm{t}}
\end{aligned}$$

Note that covariance/correlation matrices are symmetric, non-negative definite

KL-Expansion

Can always find orthonormal set of e-vectors of **K**

These are an alternate coordinate systems (rotations, reflections)

in this eigen-coordinate system, the components are uncorrelated

(principle components)

The eigen-values are the variant (energy) in each of these principle directions

(can be used to reduce dimensions by throwing out components with low energy)

KL-Expansion

should change N to D

$$\mathbf{K_x}\mathbf{e}_k = \lambda_k \mathbf{e}_k \quad k = 0, 1, \ldots N-1 \qquad \text{(Eigen equation)}$$

$$\mathbf{e}_k^{\mathrm{t}}\mathbf{e}_l = \delta[k-l] \quad \lambda_k \geq 0 \qquad \text{(orthonormal e-vectors)}$$

$$\mathbf{x}(u) = \sum_{k=0}^{N-1} X_k(u)\mathbf{e}_k \qquad \text{(change of coordinates)}$$

$$X_k(u) = \mathbf{e}_k^{\mathrm{t}}\mathbf{x}(u)$$

$$\mathbb{E}\left\{X_k(u)X_l(u)\right\} = \mathbf{e}_k^{\mathrm{t}}\mathbf{K_x}\mathbf{e}_l = \lambda_k\delta[k-l] \qquad \text{(uncorrelated components)}$$

$$\mathbf{K_x} = \sum_{k=0}^{N-1} \lambda_k \mathbf{e}_k \mathbf{e}_k^{\mathrm{t}} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^{\mathrm{t}} \qquad \text{(Mercer's Theorem)}$$

$$\mathbb{E}\left\{\|\mathbf{x}(u)\|^2\right\} = \mathrm{tr}\left(\mathbf{K_x}\right) = \sum_{k=0}^{N-1} \lambda_k \qquad \text{(Total Energy)}$$

Always exists because **K** is nnd-symmetric

KL-Expansion

$$d_k(u) = \mathbf{e}_k^{\mathrm{t}} \mathbf{x}(u) \qquad k = 0, 1, \ldots D-1$$

$$\mathbf{d}(u) = \mathbf{E}^{\mathrm{t}} \mathbf{x}(u)$$

$$\begin{aligned} \mathbf{K_d} &= \mathbf{E}^{\mathrm{t}} \mathbf{K_x} \mathbf{E} \\ &= \mathbf{E}^{\mathrm{t}} \left(\mathbf{E} \boldsymbol{\Lambda} \mathbf{E}^{\mathrm{t}} \right) \mathbf{E} \\ &= \boldsymbol{\Lambda} = \mathbf{diag}(\lambda_k) \end{aligned}$$

Multiplying by **E**^t makes the components uncorrelated

$$\mathbf{E} = \left[\; \mathbf{e}_0 \; \middle| \; \mathbf{e}_1 \; \middle| \; \mathbf{e}_2 \; \middle| \; \cdots \; \middle| \; \mathbf{e}_{D-1} \; \right]$$

KL-Expansion - Relation to Whitening

$$w_k(u) = \frac{X_k(u)}{\sqrt{\lambda_k}} = \frac{\mathbf{e}_k^{\mathrm{t}}\mathbf{x}(u)}{\sqrt{\lambda_k}} \qquad k = 0, 1, \ldots D-1$$

$$\mathbf{w}(u) = \mathbf{\Lambda}^{-1/2}\mathbf{E}^{\mathrm{t}}\mathbf{x}(u)$$

$$\begin{aligned}
\mathbf{K}_{\mathbf{w}} &= \mathbf{\Lambda}^{-1/2}\mathbf{E}^{\mathrm{t}}\mathbf{K}_{\mathbf{x}}\mathbf{E}\mathbf{\Lambda}^{-1/2} \\
&= \mathbf{\Lambda}^{-1/2}\mathbf{\Lambda}\mathbf{\Lambda}^{-1/2} \\
&= \mathbf{I}
\end{aligned}$$

For any orthogonal matrix U, this whitening matrix also works:

$$\mathbf{G} = \mathbf{U}\mathbf{\Lambda}^{-1/2}\mathbf{E}^{\mathrm{t}}$$

KL-Expansion - Relation to PCA

$$\tilde{x}_k(u) = \mathbf{e}_k^{\mathrm{t}} \mathbf{x}(u) \qquad k = 0, 1, \ldots T-1$$

$$\tilde{\mathbf{x}}(u) = \mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{x}(u) \qquad \text{first } T \text{ components}$$

$$\mathbf{K}_{\tilde{\mathbf{x}}} = \mathbf{\Lambda}_{[:T]} \qquad \text{assumes ordered e-values: } \lambda_0 \geq \lambda_1 \geq \ldots \lambda_{D-1}$$

$$\mathbb{E}\left\{ \|\tilde{\mathbf{x}}(u)\|^2 \right\} = \sum_{k=0}^{T-1} \lambda_k$$

$$\mathbb{E}\left\{ \|\mathbf{x}(u) - \tilde{\mathbf{x}}(u)\|^2 \right\} = \sum_{k=T}^{D-1} \lambda_k \qquad \text{minimizes approximation error (lossy compression)}$$

PCA is simply taking only the T most important e-directions or principle components

$$\mathbf{E}_{[:T]} = \left[\begin{array}{c|c|c|c|c} \mathbf{e}_0 & \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_{T-1} \end{array} \right]$$

KL/PCA for Data

Everything is the same, except we use data-averaging instead of E{.}

$$\hat{\mathbf{R}}_{\mathbf{x}} = \left\langle \mathbf{x}\mathbf{x}^{\mathrm{t}} \right\rangle_{\mathcal{D}}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}_n \mathbf{x}_n^{\mathrm{t}}$$

$$= \frac{1}{N} \mathbf{X}^{\mathrm{t}} \mathbf{X}$$

Both KL/PCA can be applied to **R** or **K**. Center **x** if you want to use **K**

x <— **x** - **m**

(same if mean is zero)

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0^{\mathrm{t}} \\ \mathbf{x}_1^{\mathrm{t}} \\ \vdots \\ \mathbf{x}_{N-1}^{\mathrm{t}} \end{bmatrix} \qquad \mathbf{X}^{\mathrm{t}} = \begin{bmatrix} \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_{N-1} \end{bmatrix}$$

"stacked" data matrix

KL/PCA for Data

PCA for data

$$\tilde{\mathbf{x}}_n = \mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{x}_n \qquad \text{first } T \text{ components}$$

$$\mathbf{E}_{[:T]} = \left[\begin{array}{c|c|c|c|c} \mathbf{e}_0 & \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_{T-1} \end{array} \right]$$

apply to the "stacked" data matrix

$$\tilde{\mathbf{X}} = \begin{bmatrix} \left(\mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{x}_0\right)^{\mathrm{t}} \\ \left(\mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{x}_1\right)^{\mathrm{t}} \\ \vdots \\ \left(\mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{x}_{N-1}\right)^{\mathrm{t}} \end{bmatrix} = \mathbf{X}\mathbf{E}_{[:T]}$$

$$\underset{N \times T}{\tilde{\mathbf{X}}} = \underset{N \times D}{\mathbf{X}} \quad \underset{D \times T}{\mathbf{E}_{[:T]}}$$

$$\underset{T \times T}{\tilde{\mathbf{X}}^{\mathrm{t}} \tilde{\mathbf{X}}}$$

dimension reduced from D to T

$$\tilde{\mathbf{X}}^{\mathrm{t}} = \begin{bmatrix} \mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{x}_0 & \mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{x}_1 & \cdots & \mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{x}_{N-1} \end{bmatrix} = \mathbf{E}_{[:T]}^{\mathrm{t}} \mathbf{X}^{\mathrm{t}}$$

KL/PCA for Data — relation to SVD

SVD for an arbitrary matrix **A**

$$\underset{m\times n}{\mathbf{A}} = \underset{m\times m}{\mathbf{U}} \quad \underset{m\times n}{\mathbf{\Sigma}} \quad \underset{n\times n}{\mathbf{V}^{\mathrm{t}}}$$

U, **V** are orthogonal matrices, **sigma** is "diagonal" with singular values on diagonal

Use SVD to expand matrix **A^t A**

$$\mathbf{A}^{\mathrm{t}}\mathbf{A} = (\mathbf{U\Sigma V})^{\mathrm{t}}\,\mathbf{U\Sigma V}$$

$$= \mathbf{V\Sigma}^{\mathrm{t}}\mathbf{U}^{\mathrm{t}}\mathbf{U\Sigma V}$$

$$= \underset{n\times n}{\mathbf{V}} \quad \underset{n\times n}{\mathbf{\Sigma\Sigma}^{\mathrm{t}}} \quad \underset{n\times n}{\mathbf{V}^{\mathrm{t}}}$$

$$= \mathbf{E\Lambda E}^{\mathrm{t}}$$

The SVD for **A** provides the KL factorization for the non-negative definition, symmetric matrix **A^t A**

Note that this is also the SVD for **A^t A**

KL/PCA for Data — relation to SVD

SVD for stacked data matrix **X**

$$\underset{N\times D}{\mathbf{X}} = \underset{N\times N}{\mathbf{U}} \quad \underset{N\times D}{\mathbf{\Sigma}} \quad \underset{D\times D}{\mathbf{V}^{\mathrm{t}}}$$

$$\underset{D\times D}{\mathbf{X}^{\mathrm{t}}\mathbf{X}} = \underset{D\times D}{\mathbf{V}} \quad \underset{D\times D}{\mathbf{\Sigma}\mathbf{\Sigma}^{\mathrm{t}}} \quad \underset{D\times D}{\mathbf{V}^{\mathrm{t}}}$$

$$= \mathbf{E}\mathbf{\Lambda}\mathbf{E}^{\mathrm{t}}$$

Equivalent approaches:

1) Find SVD of **X**, take **V**

2) Find Eigen decomposition of **X**^t **X**, take **E** = **V**

3) Find SVD of **X**^t **X**, take **V** = **U** = **E**

$$\underset{N\times T}{\tilde{\mathbf{X}}} = \underset{N\times D}{\mathbf{X}} \quad \underset{D\times T}{\mathbf{V}_{[:T]}}$$

KL/PCA for Data — relation to SVD

Equivalent approaches:

1) Find SVD of **X**, take **V**

2) Find Eigen decomposition of **X**^t **X**, take **E** = **V**

3) Find SVD of **X**^t **X**, take **V** = **U** = **E**

$$\underset{N\times T}{\tilde{\mathbf{X}}} = \underset{N\times D}{\mathbf{X}} \quad \underset{D\times T}{\mathbf{V}_{[:T]}}$$

May want to use method 3, with numpy.linalg.svd, instead of method 2, with numpy.linalg.eig, since the SVD returns the e-vectors in sorted order and Eig does not