Lecture 9: Dimensional Reduction

[AIX7021] Computer Vision

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

Tentative schedule

week	topic	date
1	Introduction and Basics	09.01
2	Image Process I	09.08
3	Image Process II	09.15
4	(휴강)	09.22
5	Feature Detection and Matching I	09.29
6	Feature Detection and Matching II	10.06
7	Clustering and Segmentation I & II	10.13 & 10.16
8	**Mid-Term Exam**	**10.20**
9	Mid-Term Solution	10.27
10	Dimensional Reduction	11.03
11	Robust Fitting and Matching	11.10
12	Object Recognition	11.17
13	Motion and Tracking	11.24
14	Image Classification	12.01
15	**Final exam**	**12.08**

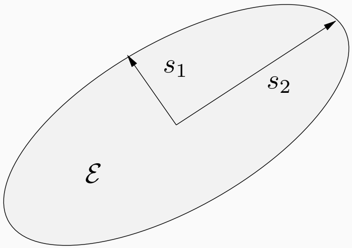
Singular Value Decomposition (SVD)

Ellipsoid

If $A = A^T > 0$, the set

$$\mathcal{E} = \{x \mid x^T A x \leq 1\} \tag{1}$$

is an *ellipsoid* in $\Re^n$, centered at 0



Singular value decomposition

Singular value decomposition (SVD) of a given matrix A

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{2}$$

$$= \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} — & \mathbf{v}_1^T & — \\ — & \mathbf{v}_2^T & — \\ & \vdots & \\ — & \mathbf{v}_r^T & — \end{bmatrix} \tag{3}$$

where

- $\mathbf{A} \in \Re^{m \times n},\ \mathrm{rank}(\mathbf{A}) = r$
- $\mathbf{U} \in \Re^{m \times r},\ \mathbf{U}^T\mathbf{U} = \mathbf{I}$
- $\mathbf{V} \in \Re^{n \times r},\ \mathbf{V}^T\mathbf{V} = \mathbf{I}$
- $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, ..., \sigma_r)$ where $\sigma_1 \geq \cdots \geq \sigma_r > 0$

Singular value decomposition

with $\mathbf{U} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \\ | & | & & | \end{bmatrix}$ and $\mathbf{V} = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_r \\ | & | & & | \end{bmatrix}$,

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^T \tag{4}$$

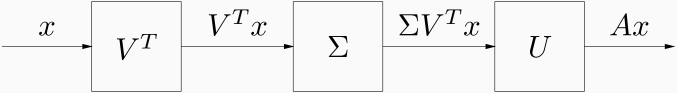
where

- σ_i are the nonzero *singular values* of $\mathbf{A}$
- $\mathbf{v}_i$ are the *right* or *input singular vectors* of $\mathbf{A}$
- $\mathbf{u}_i$ are the *left* or *output singular vectors* of $\mathbf{A}$

Interpretations

SVD:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^T \tag{5}$$

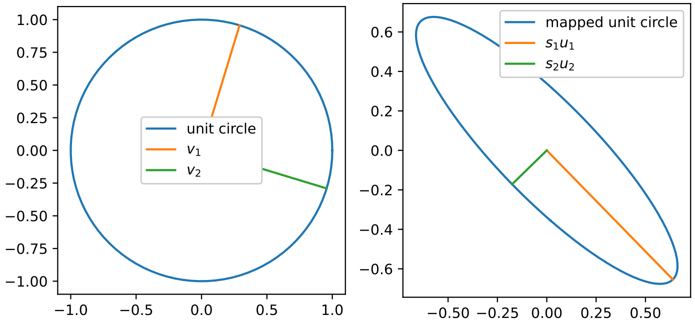


Linear mapping $\mathbf{y} = \mathbf{A}\mathbf{x}$ can be decomposed as

- compute coefficients of $\mathbf{x}$ along input directions $\mathbf{v}_1, ..., \mathbf{v}_r$
- scale coefficients by σ_i
- reconstitute along output directions $\mathbf{u}_1, ..., \mathbf{u}_r$

difference with eigenvalue decomposition for symmetric $\mathbf{A}$: input and output directions are *different*

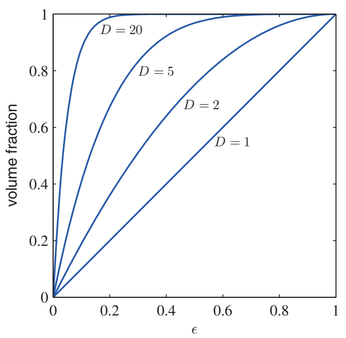
Geometric interpretation



Principal Component Analysis (PCA)

Curse of dimensionality

Figure 1.22 Plot of the fraction of the volume of a sphere lying in the range $r = 1-\epsilon$ to $r = 1$ for various values of the dimensionality D.



Dimensional reduction

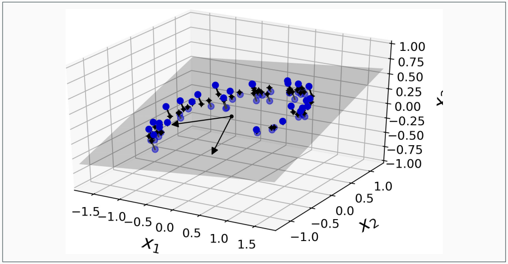


Figure 8-2. A 3D dataset lying close to a 2D subspace

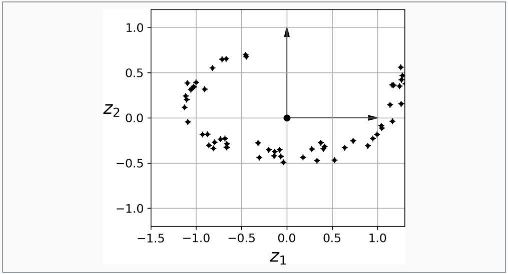


Figure 8-3. The new 2D dataset after projection

Covariance matrix

Covariance measures the strength of the linear relationship between two variables

$$\sigma_{xy} = \mathrm{E}[(x - \mu_x)(y - \mu_y)] \tag{6}$$

Covariance matrix C for multivariate random variable X

$$C_{ij} = \mathrm{E}[(x_i - \mu_i)(x_j - \mu_j)] \tag{7}$$

Preserving the variance

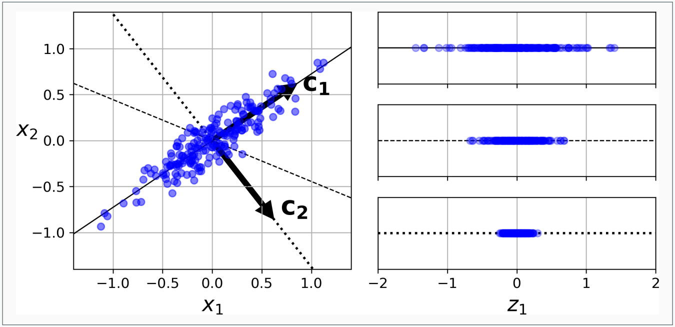


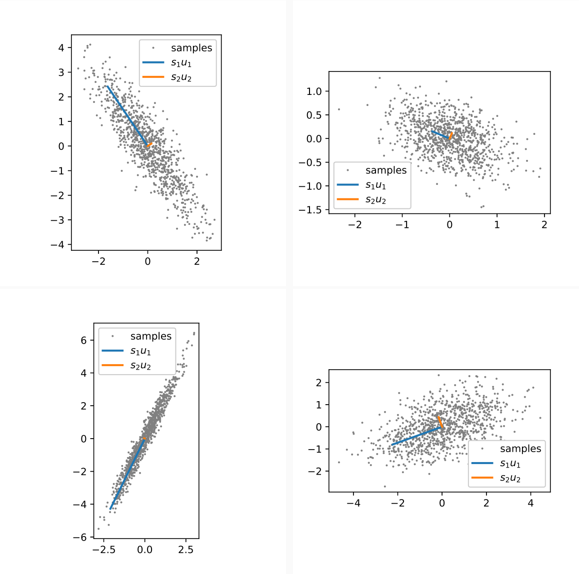
Figure 8-7. Selecting the subspace onto which to project

Principal component analysis (PCA)

For given data $x_1, x_2, ..., x_N \in \Re^D$

1. create a matrix $X \in \Re^{D \times N}$ with one column vector per each sample
2. covariance matrix $\Sigma = \mathrm{E}\left[(X - \mathrm{E}(X))(X - \mathrm{E}(X))^T\right] \in \Re^{D \times D}$
3. find singular vectors and singular values of Σ
4. principal components = largest singular values and vectors

Principal component analysis (PCA)



MNIST dataset

- 70,000 images of 28×28 handwritten digits from 0 to 9

Figure 3-1. A few digits from the MNIST dataset

PCA and dimensional reduction for MNIST dataset

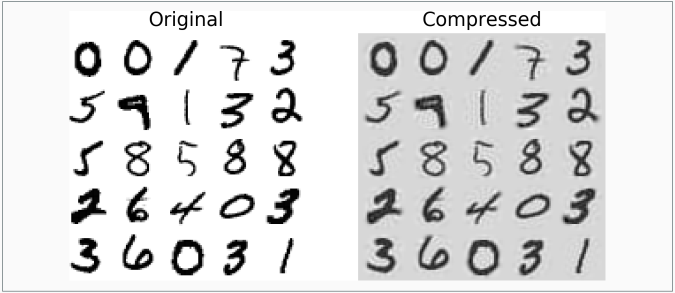


Figure 8-9. MNIST compression preserving 95% of the variance

Eigenface

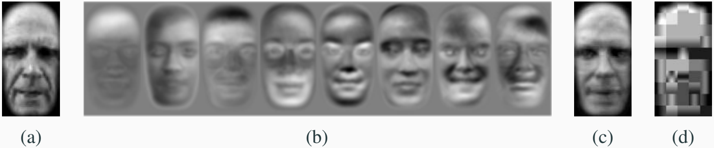


Figure 14.13 Face modeling and compression using eigenfaces (Moghaddam and Pentland 1997) © 1997 IEEE: (a) input image; (b) the first eight eigenfaces; (c) image reconstructed by projecting onto this basis and compressing the image to 85 bytes; (d) image reconstructed using JPEG (530 bytes).

- Project faces onto the subspace to maximize the appearance variations of faces by eigen–decomposition
- Compare two faces by projecting the images onto the subspace and measuring the Euclidean distance between them

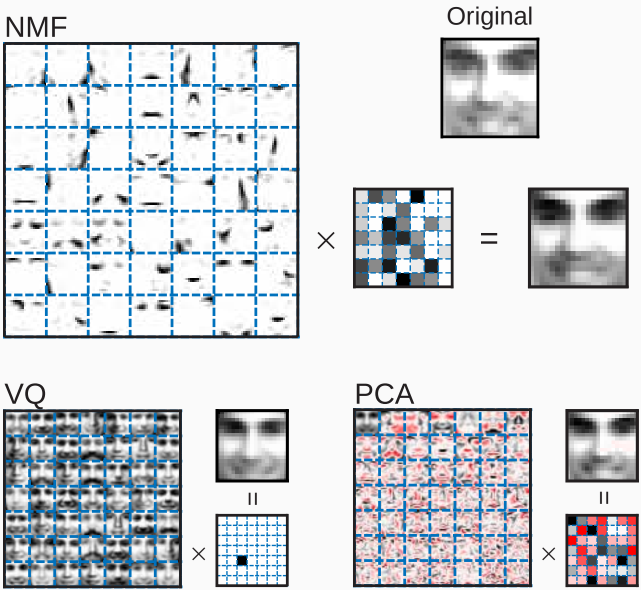
Matrix Factorization for Dimensional Reduction

Matrix factorization

- Principal component analysis (PCA): orthogonal property
- Vector quantization (VQ): unary property
- Non-negative matrix factorization (NMF): non-negativity

$$\begin{aligned} \text{PCA: } V &\approx WH & &\text{where } W^T W = I \\ \text{VQ: } V &\approx WH & &\text{where } H \text{ consists of unary vectors} \\ \text{NMF: } V &\approx WH & &\text{where } W_{ij} \geq 0, H_{ij} \geq 0 \end{aligned}$$

Matrix factorization and face image compression

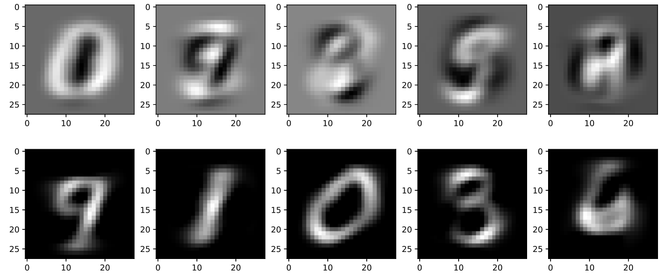


Nonnegative matrix factorization (NMF)

Nonnegative matrix factorization (NMF)

$$
\begin{aligned}
\text{given} \quad & A \in \Re^{n \times m} \\
\text{find} \quad & (W, H) = \underset{(W,H)}{\arg\min} \, \|A - WH\|_F^2 \\
\text{s.t.} \quad & W \in \Re_+^{n \times k} \text{ and } H \in \Re_+^{k \times m} \ (k \leq \operatorname{rank}(A))
\end{aligned}
$$

PCA vs. NMF for MNIST dataset



Appendix

Reference and further reading

- "Chap 8 | Dimensionality Reduction" of A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
- "Chap 4" of C. Bishop, Pattern Recognition and Machine Learning
- Stephen Boyd, "Lecture 15 | Symmetric matrices, quadratic forms, matrix norm, and SVD" and "Lecture 16 | SVD Applications" of *EE263: Introduction to Linear Dynamical Systems*, Stanford University (2008)
- D. D. Lee and H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature* **401**, 788-791 (1999)
- "Chap 14.2 | Face Recognition" of R. Szeliski, Computer Vision: Algorithms and Applications