Lecture 15: Dimensional Reduction

[SCS4049-02] Machine Learning and Data Science

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

Tentative schedule

week	topic	date (수 / 월)
1	Machine Learning Introduction & Basic Mathematics	09.02 / 09.07
2	Python Practice I & Regression	09.09 / 09.14
3	AI Department Seminar I & Clustering I	09.16 / 09.21
4	Clustering II & Classification I	09.23 / 09.28
5	Classification II	(추석) / 10.05
6	Python Practice II & Support Vector Machine I	10.07 / 10.12
7	Support Vector Machine II & Decision Tree and Ensemble Learning	10.14 / 10.19
8	Mid-Term Practice & **Mid-Term Exam**	10.21 / **10.26**
9	휴강 & Dimensional Reduction	10.28 / 11.02
10	Neural networks & Backpropagation	11.04 / 11.09
11	Convolutional Neural Network	11.11 / 11.16
12	Model Optimization	11.18 / 11.23
13	Recurrent Neural network	11.25 / 11.30
14	Autoencoders	12.02 / 12.07
15	**Final exam**	(휴강) / **12.14**

Singular Value Decomposition (SVD)

HW #5

Due: 10월 20일 화요일, 23시 59분까지

- 컴퓨터로 작성(latex, word, ppt, 한글 등)해서 pdf로 업로드해주세요.
- 또는 손으로 작성한 파일을 스캔앱(Adobe scan, Office lens 등)을 써서 pdf로 업로드해주세요.

1. 행렬의 singular value decomposition (SVD)에 대해 공부해보고 다음에 대해 설명하세요.
   1.1 Linear transformation 관점에서 행렬 A를 $A = U\Sigma V^T$로 SVD 했을 때, 주어진 벡터 x를 어떤 단계를 거쳐 $y = Ax$로 변환하는지 (10점)
   1.2 $A = U\Sigma V^T$에서 각 행렬 U, V이 행렬의 column space 및 row space와 어떤 물리적인 관계를 갖는지 (5점)
   1.3 행렬의 singular value와 Σ 행렬의 관계 (5점)
   1.4 행렬의 positive definiteness와 singular value 사이의 관계 (10점)

Positive (semi-)definite matrix

Suppose $A = A^T \in \Re^{n\times n}$

We say A is *positive semi-definite* if $x^T A x \geq 0$ for all x

- denoted $A \geq 0$
- $A \geq 0$ if and only if $\lambda_{\min}(A) \geq 0$, i.e., all eigenvalues are nonnegative
- **not** the same as $A_{ij} \geq 0$ for all i, j

We say A is *positive definite* if $x^T A x > 0$ for all $x \neq 0$

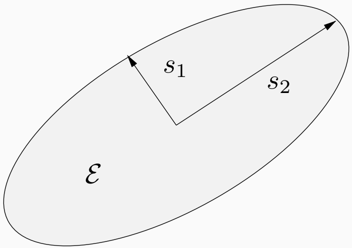
- denoted $A > 0$
- $A > 0$ if and only if $\lambda_{\min}(A) > 0$, i.e., all eigenvalues are positive

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

$$
\begin{aligned}
\mathbf{A} &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T && (2) \\
&= \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_r \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} — & \mathbf{v}_1^T & — \\ — & \mathbf{v}_2^T & — \\ & \vdots & \\ — & \mathbf{v}_r^T & — \end{bmatrix} && (3)
\end{aligned}
$$

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

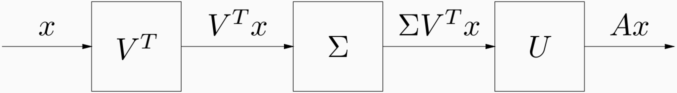
$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^T \tag{4}$$

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 $\mathbf{A}$

SVD:

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

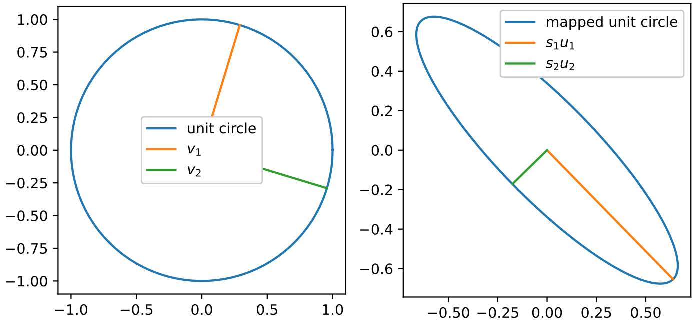


Linear mapping $\mathbf{y} = \mathbf{A}\mathbf{x}$ can be decomposed as

- 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*

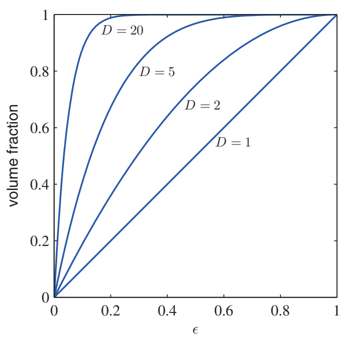
Geometric interpretation



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

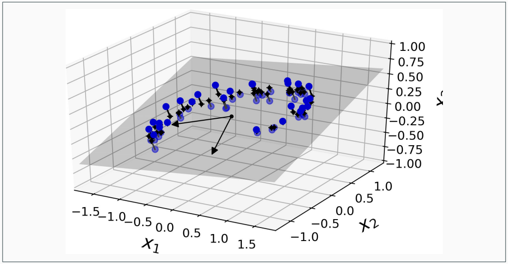


Figure 8-2. A 3D dataset lying close to a 2D subspace

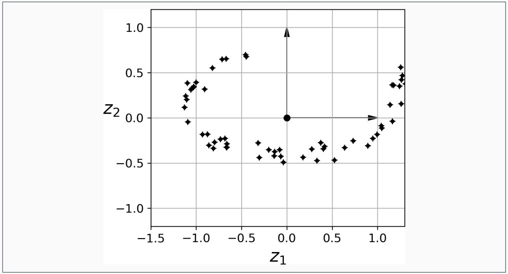


Figure 8-3. The new 2D dataset after projection

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

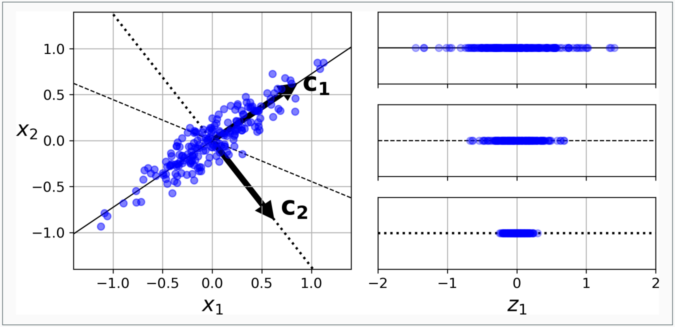


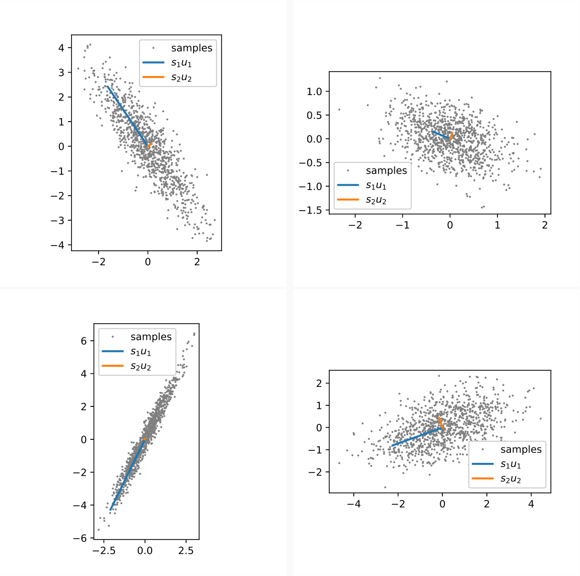
Figure 8-7. Selecting the subspace onto which to project

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 $\Sigma = \mathrm{E}\big[(X - \mathrm{E}(X))(X - \mathrm{E}(X))^T\big] \in \Re^{D \times D}$
3. find singular vectors and singular values of Σ
4. principal components = largest singular values and vectors

Principal component analysis (PCA)



Principal component analysis (PCA)

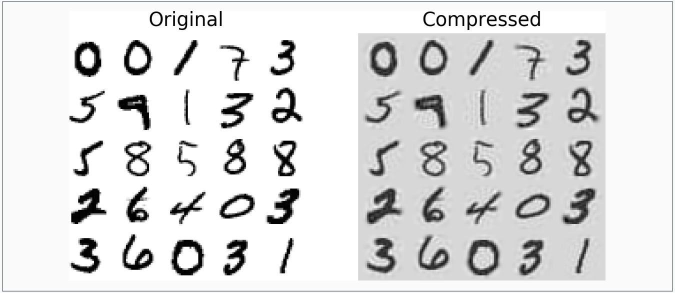


Figure 8-9. MNIST compression preserving 95% of the variance

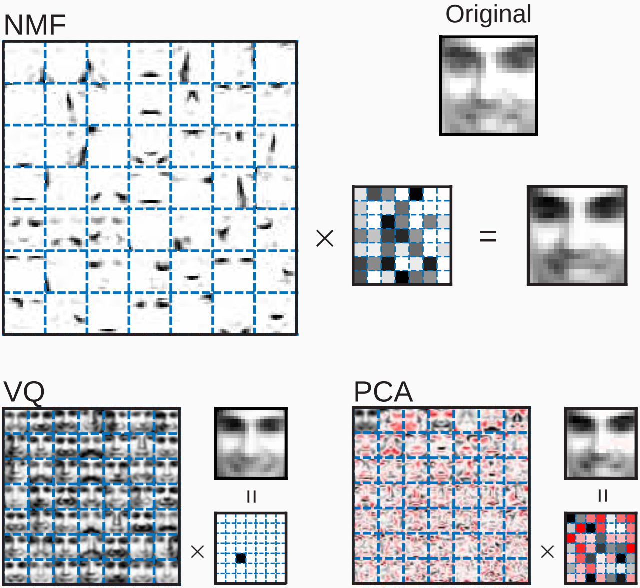
Matrix Factorization and Dimensional Reduction

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)

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

Appendix

Reference and further reading

- “Chap 8 | Dimensionality Reduction” of A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
- 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)