SVD $A = \underline{U}\,\underline{\Sigma}\,\underline{V}^T$

Inclass 16: Dimensional Reduction

[SCS4049] Machine Learning and Data Science

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

$$X \in \mathbb{R}^{D \times N} \qquad \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_N \\ | & | & & | \end{bmatrix}$$

$$u \in \mathbb{R}^{D}$$

D차원 벡터들

$$\underbrace{u^T}_{1 \times D} \underbrace{X}_{D \times N} = u^T \begin{bmatrix} \boxed{\begin{matrix} | \\ x_1 \\ | \end{matrix}} & \boxed{\begin{matrix} | \\ x_2 \\ | \end{matrix}} & \cdots & \begin{matrix} | \\ x_N \\ | \end{matrix} \end{bmatrix}$$

$$= \begin{bmatrix} u^T x_1 & u^T x_2 & \cdots & u^T x_N \end{bmatrix} \in \mathbb{R}^{1 \times N}$$

1차원 reduction.

$x \in \mathbb{R}^D$ 벡터.

SVD = $U \Sigma V^T$

$$\begin{bmatrix} | & | & | & & | \\ u_1 & u_2 & u_3 & \cdots & u_N \\ | & | & | & & | \end{bmatrix}$$

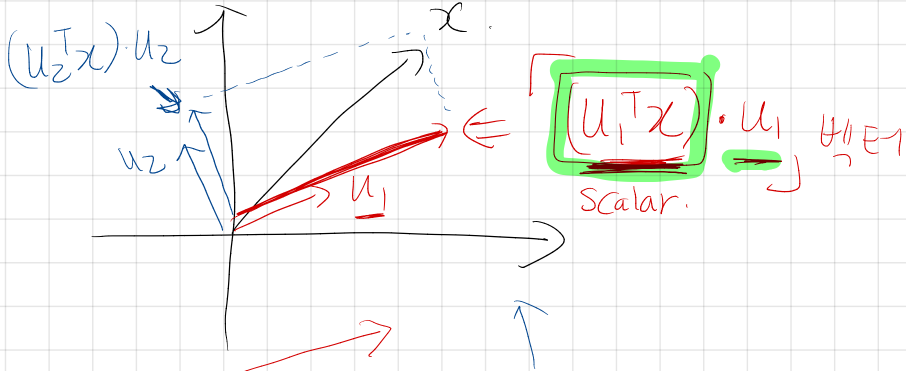
$u_1^T x$. → 1차원

$u_2^T x$. → 1차원

$u_3^T x$.

$\vdots$

$u_N^T x$.



$$x = (u_1^T x) u_1 + (u_2^T x) u_2$$

D차원.

$$\underbrace{\begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_D \\ | & | & & | \end{bmatrix}}_{} \quad U$$

$x \in \mathbb{R}^D$.

D개의 벡터 u를 이용해서
x를 표현하는 것.

$$x = (u_1^T x)u_1 \pm (u_2^T x)u_2 \pm (u_3^T x)u_3 \cdots \pm (u_D^T x)u_D$$

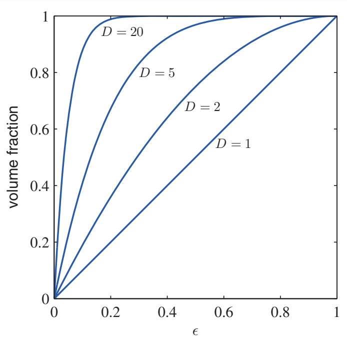
주성분 분석

Principal Component Analysis (PCA).

주. 성분

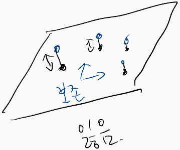
Curse of dimensionality

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

→ 데이터의 분포를 유지하는 방향.



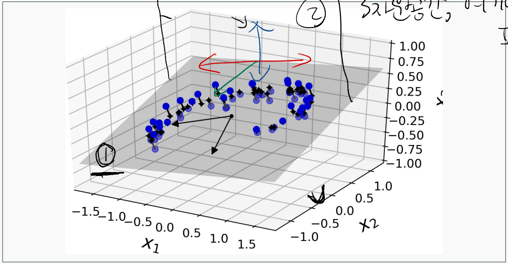


Figure 8-2. A 3D dataset lying close to a 2D subspace

3차원공간, 여기에 존재하는

파란색 sample들.

↓

2차원 축소.

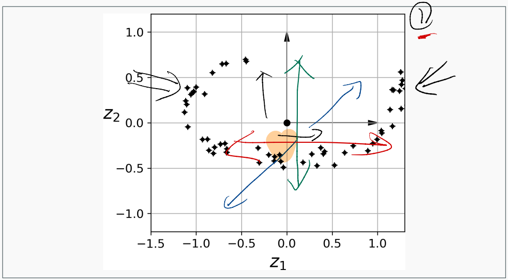
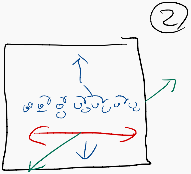


Figure 8-3. The new 2D dataset after projection



Covariance matrix

Covariance measures the strength of the linear relationship between two variables

$$\sigma_{xy} = \mathrm{E}[(x - \mu_x)(y - \mu_y)] \tag{1}$$

Covariance matrix C for multivariate random variable X

$$C_{ij} = \mathrm{E}[(x_i - \mu_i)(x_j - \mu_j)] \tag{2}$$

Principal component analysis (PCA)

Preserving the variance

$C_1^T \cdot [\, x_1 \ x_2 \ \cdots \ x_N \,] \in \mathbb{R}^{2\times N}$

$C_2^T \cdot$ $\qquad \mathbb{R}^{1\times N}$.

$C_3^T \cdot$

2차원데이터.

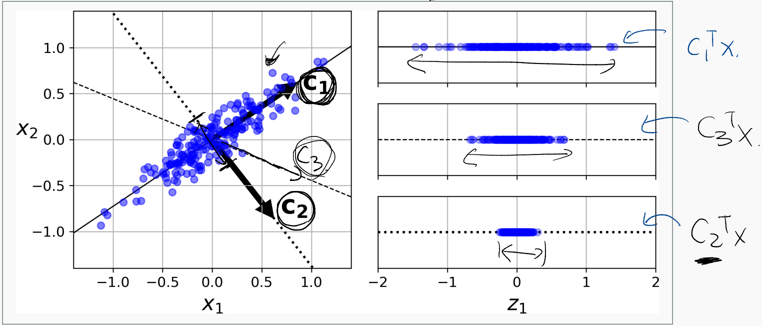


Figure 8-7. Selecting the subspace onto which to project

2차원 → 1차원.

C_1 vs. C_2 vs C_3,

최악. 애매.

Principal component analysis (PCA)

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

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

$$C = U \Sigma V^T$$

$$\Sigma = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

$$U = [u_1 \; u_2 \; u_3]$$

3차원 → 1차원 u_1

→ 2차원 u_1, u_2

$$X \in \mathbb{R}^{4 \times 1000}.$$

구할수있음.

$$C \in \mathbb{R}^{4 \times 4}$$

Covariance.

$$C = U \Sigma V^T$$

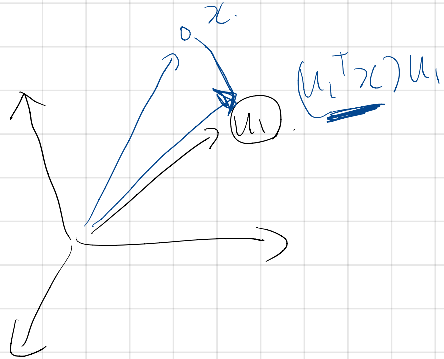
$$\Sigma = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

4차원 → 1차원 u_1.

내적을한값.

$$(\underline{u_1^T X}) \in \mathbb{R}^{1 \times 1000}.$$

스칼라들을 모은값.



$$\underbrace{[u_1 \cdot (u_1^T X)]}_{\text{reconstruction}} \in \mathbb{R}^{4 \times 1000}.$$

4차원 → 2차원. $\begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \in \mathbb{R}^{4\times2}$.

내적 $\underline{\begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix}^T X} \in \mathbb{R}^{2\times1000}$

reconstruction $\underbrace{\begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix}}_{4\times2} \underbrace{\begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix}^T X}_{2\times1000} \in \mathbb{R}^{4\times1000}$

$$\Sigma = \begin{bmatrix} \underline{1000} & & & \\ & \underline{10} & & \\ & & \underline{1} & \\ & & & \underline{0.1} \end{bmatrix}$$

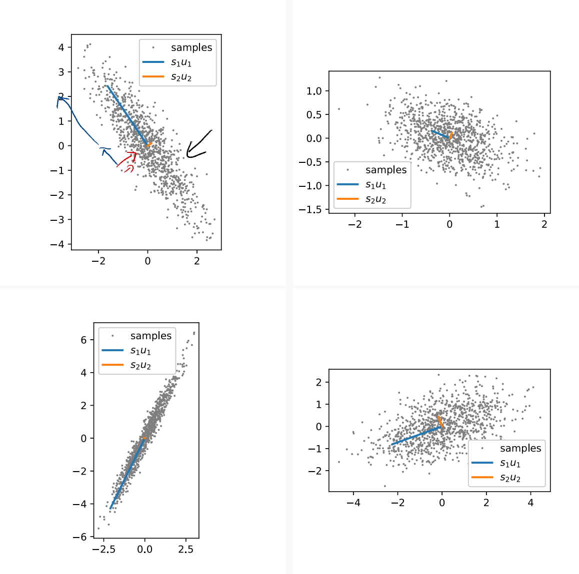
1차원 u_1 선택. $\frac{1000}{1011.1}$ 만큼 보존.

$$\begin{bmatrix} \underline{1000} & & \\ & 0.1 & \\ & & 0.001 \end{bmatrix} \quad \frac{1000}{1000.11} \approx 1.$$

2차원 u_1, u_2 $\frac{1000+10}{1011.1}$

$$\begin{bmatrix} \underline{10} & & \\ & \underline{5} & \\ & & 1 \end{bmatrix} \quad \frac{10}{16} \approx \frac{2}{3}$$

3차원 u_1, u_2, u_3. $\frac{1011}{1011.1}$


samples
$s_1 u_1$
$s_2 u_2$
samples
$s_1 u_1$
$s_2 u_2$
samples
$s_1 u_1$
$s_2 u_2$
samples
$s_1 u_1$
$s_2 u_2$

Principal component analysis (PCA)

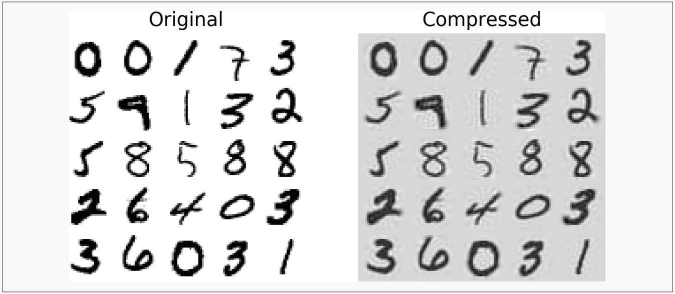


Figure 8-9. MNIST compression preserving 95% of the variance

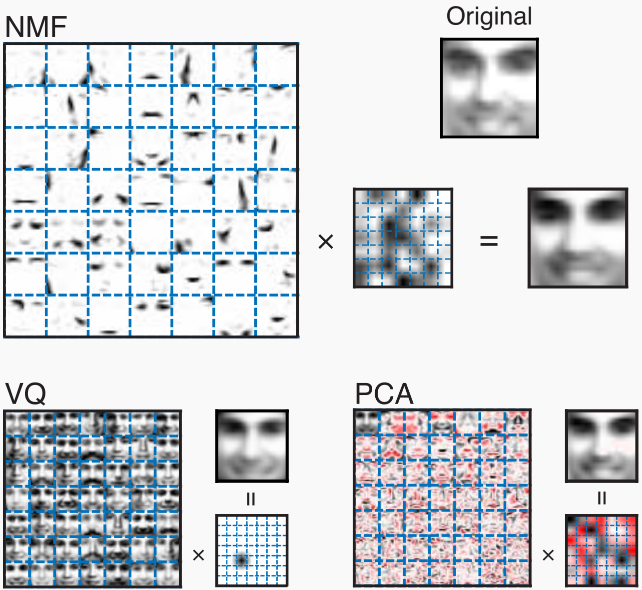
Matrix Factorization and Dimensional Reduction

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)

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

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)