14. Dimensionality Reduction

Dongwoo Kim

dongwoo.kim@postech.ac.kr

CSED515 - 2023 Spring

Examples of Unsupervised Learning

In fact, we've focused on supervised learning tasks, in particular, regression and classification. In next several lectures, we will study about unsupervised learning:

- Clustering, e.g., image segmentation
- **Feature selection or dimensionality reduction, e.g., PCA**
- Generative model, e.g., GAN

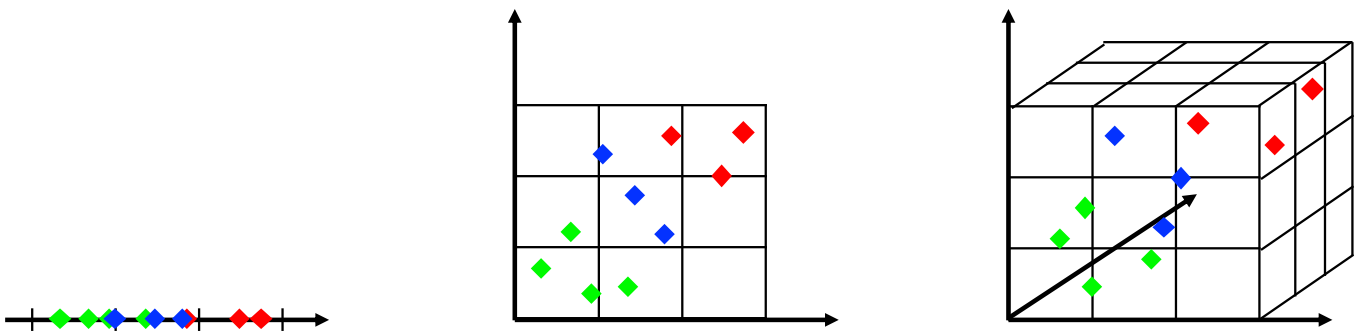
Curse of Dimensionality

Datasets are typically high-dimensional

- Vision: 1920x1080 pixels
- Text: 170k+ English words

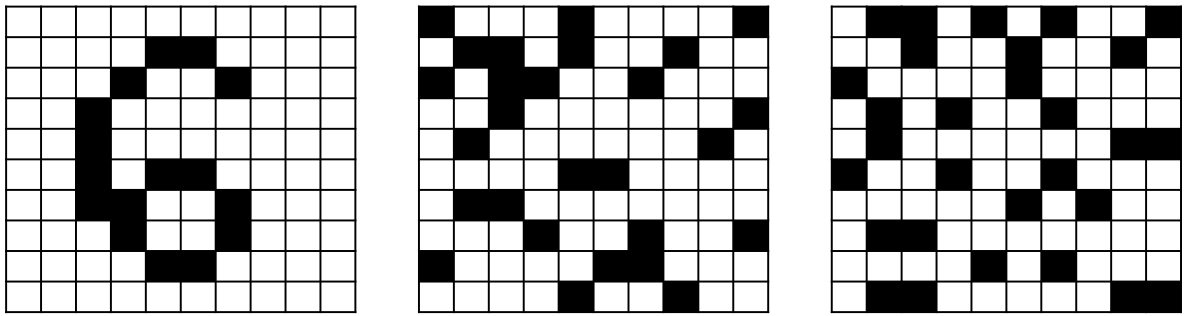
Machine learning method is essentially statistical approach that requires some repetitions of similar events, while as dimensionality grows, fewer observations per region but also higher computation cost, i.e., curse of dimensionality

- 1D: 3 regions, 2D: 3^2 regions, ... , 100D: hopelessly many regions



True vs. Observed Dimensionality

10x10 bitmap images of handwriting in $\{0,1\}^{100}$:



Handwriting of 6 | Random incidence 1 | Random incidence 2

However, the true dimensionality may be low as we do not observe every possible image.

Principal Component Analysis (PCA) is a technique primarily used for dimensionality reduction in machine learning.

- Other usages: data compression, feature extraction, data visualization ...

Table of Contents

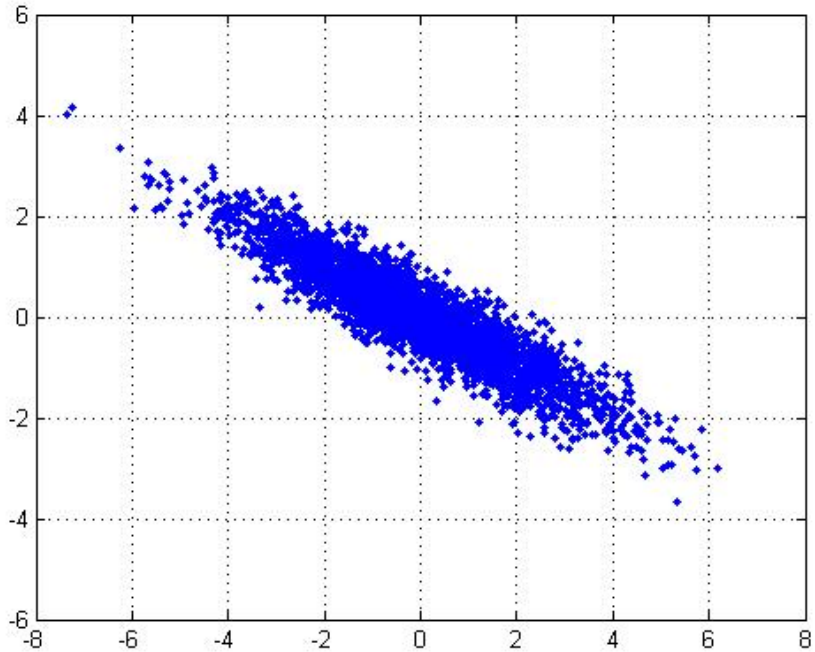
1 Principal Component Analysis (PCA)

- Maximum variance formulation
- Minimum-error formation
- Applications of PCA
- PCA for high-dimensional data

2 Factor Analysis

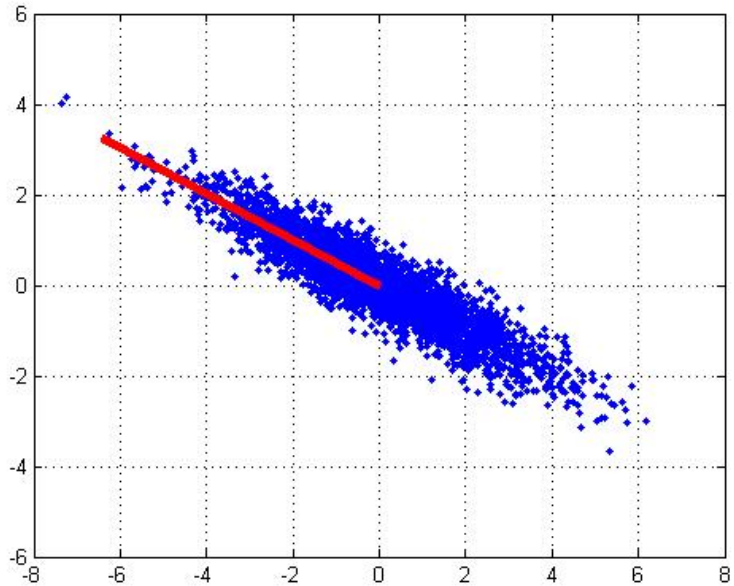
- Maximum Likelihood Factor Analysis

Let's Project Data into Subspace



- How can we project the data points in two dimensional space into one dimensional space?

New Basis



- We can find a new basis of 2D space that can carry most of the information.
- By projecting each point onto this basis, we can represent a point as a single number.

How to choose basis?

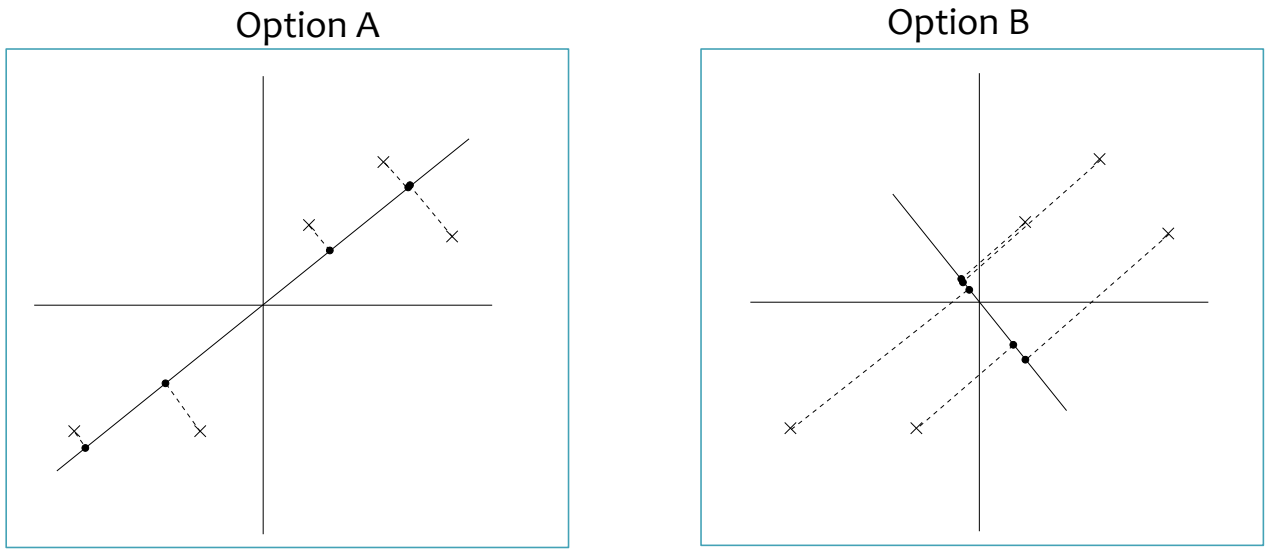


Figure: Given 5 data points, we come up with two different bases. What would be a better one?

How to choose basis?

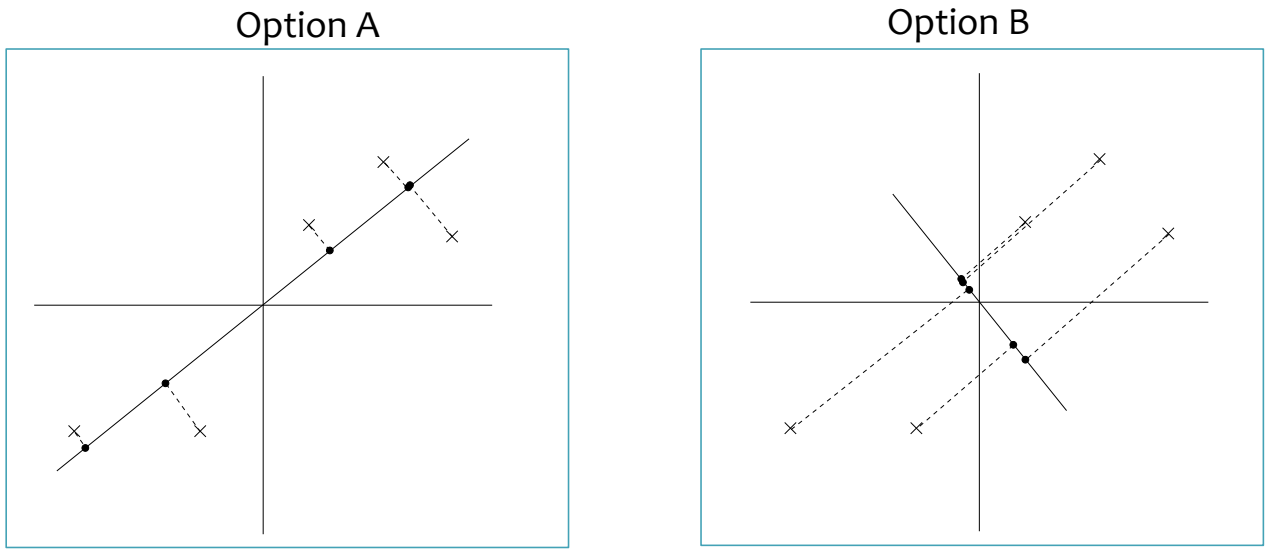
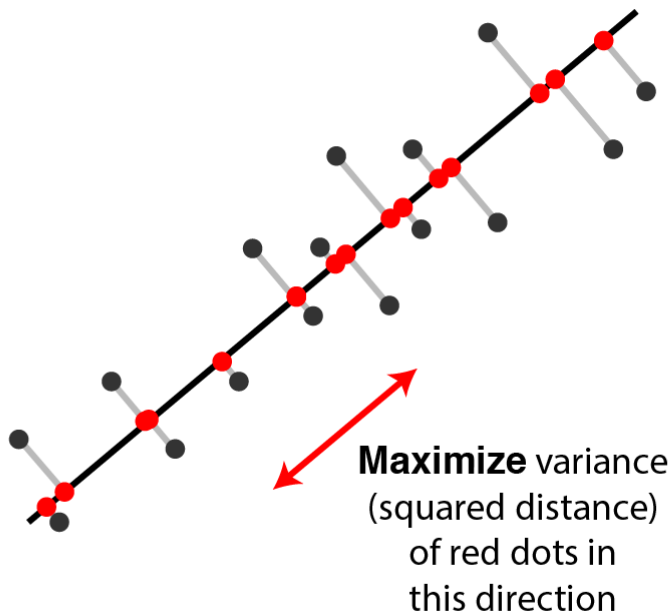


Figure: Given 5 data points, we come up with two different bases. What would be a better one?

Answer: Option A because it maximally spread the data points after projection.

Maximum Variance Formulation

Consider a dataset $\{x_n\}_{n\in[N]}$ where $x_n \in \mathbb{R}^D$. Our goal is to project the data onto a space having dimensionality $M < D$ while maximizing the variance of the projected data.



Maximum Variance Formulation with $M = 1$

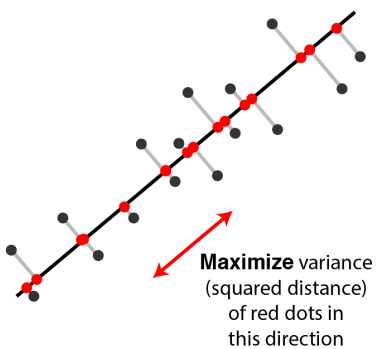
A projection onto 1-dimensional can be described by a unit vector $u_1 \in \mathbb{R}^D$, i.e., $u_1^\top u_1 = 1$, where each x_n is projected onto $u_1^\top x_n \in \mathbb{R}^1$.

The variance of the projected data is given by

$$\frac{1}{N} \sum_{n \in [N]} (u_1^\top x_n - u_1^\top \bar{x})^2 = u_1^\top S u_1 \ ,$$

where we define the mean vector $\bar{x}$ and covariance matrix S as:

$$\bar{x} \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n \in [N]} x_n \quad \text{and} \quad S \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n \in [N]} (x_n - \bar{x})(x_n - \bar{x})^\top \ .$$



$$\iff \quad \underset{u_1 \in \mathbb{R}^D}{\text{maximize}} \quad u_1^\top S u_1 \quad \text{subject to} \quad u_1^\top u_1 = 1$$

1-dimensional Principal Subspace

PCA can be formulated as:

$$\underset{u_1 \in \mathbb{R}^D}{\text{maximize}} \quad u_1^\top S u_1 \quad \text{subject to} \quad u_1^\top u_1 = 1 \, .$$

Writing Lagrange function $u_1^\top S u_1 + \lambda_1 (1 - u_1^\top u_1)$, it follows that the solution must verify[1]:

$$S u_1 = \lambda_1 u_1 \, ,$$

i.e., u_1 must be an eigenvector of S having eigenvalue λ_1.

Noting the variance of the projected data is $u_1^\top S u_1 = \lambda_1 u_1^\top u_1 = \lambda_1$, we can conclude PCA = calculating the eigenvector of the data covariance matrix corresponding to the largest eigenvalue.

[1] A matrix calculus cookbook is available at here: https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

M-dimensional Principal Subspace (1)

A projection onto M-dimensional can be described by a matrix $U = [u_1, u_2, ..., u_M] \in \mathbb{R}^{D \times M}$, so that $U^\top x_n$ is the projection of x_n, consisting of orthonormal column vector u_i, i.e.,

$$u_i^\top u_j = \delta_{ij} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} .$$

PCA can be given as the maximization of total variance, i.e., the summation of variances in each dimension:

$$\begin{aligned} \underset{U \in \mathbb{R}^{D \times M}}{\text{maximize}} \quad & \sum_{i \in [M]} u_i^\top S u_i \\ \text{subject to} \quad & u_i^\top u_j = \delta_{ij} \quad \forall i, j \in [M] . \end{aligned}$$

M-dimensional Principal Subspace (2)

PCA can be given as:

$$\begin{aligned} \underset{U \in \mathbb{R}^{D \times M}}{\text{maximize}} \quad & \sum_{i \in [M]} u_i^\top S u_i \\ \text{subject to} \quad & u_i^\top u_j = \delta_{ij} \quad \forall i, j \in [M] \, . \end{aligned}$$

Again, from writing Lagrange function, we can conclude that PCA = calculating the eigenvectors of the data covariance matrix corresponding to the M-largest eigenvalues.

Remark that the covariance matrix S is positive semidefinite, i.e., $v^\top S v \geq 0$ for all $v \in \mathbb{R}^D$, hence S has non-negative eigenvalues $\{\lambda_i\}_{i \in D}$, where $\sum_{i \in [M]} \lambda_i$ is the maximum total variance of the projected data.

Table of Contents

1 Principal Component Analysis (PCA)

- Maximum variance formulation
- Minimum-error formation
- Applications of PCA
- PCA for high-dimensional data

2 Factor Analysis

- Maximum Likelihood Factor Analysis

An Alternative Understanding of PCA

Consider a complete orthonormal set of D-dimensional basis vectors $\{u_i\}_{i\in[D]}$ such that $u_i^\top u_j = \delta_{ij}$ for all $i, j \in [D]$. Hence, each x_n can be written as:

$$x_n = \sum_{i\in[D]} \alpha_{ni} u_i \quad \text{where } \alpha_{ni} \stackrel{\text{def}}{=} x_n^\top u_i \ .$$

When we approximate each x_n by $\tilde{x}_n$ on M-dimensional linear subspace of the first M basis vectors, i.e., $\{u_i\}_{i\in[M]}$,

$$\tilde{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i \ ,$$

we want to find coefficients $\{z_{ni}\}_{n\in[N],i=1,\ldots,M}$, $\{b_i\}_{i\in M+1,\ldots,D}$ and basis vectors $\{u_i\}$ minimizing the projection error J:

$$J \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} \|x_n - \tilde{x}_n\|^2 \ .$$

Minimum Error Formulation (1)

For a given complete orthonormal set $\{u_i\}$, setting the derivative of J w.r.t. z_{ni} or b_i to be zero, the optimal coefficients $\{z_{ni}\}_{n\in[N],i=1,\ldots,M}$ and $\{b_i\}_{i\in M+1,\ldots,D}$ are obtained as:

$$z_{ni} = x_n^\top u_i \stackrel{\text{def}}{=} \alpha_{ni} \quad \forall n \in [N], i = 1, ..., M \ ,$$
$$b_i = \bar{x}^\top u_i \quad \forall i \in M+1, ..., D \ ,$$

where $\bar{x} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n\in[N]} x_n$ is again defined as the mean of data points. Hence, we have:

$$\begin{aligned} x_n - \tilde{x}_n &= \left(\sum_{i=1}^{D}(x_n^\top u_i)u_i\right) - \left(\sum_{i=1}^{M}(x_n^\top u_i)u_i + \sum_{i=M+1}^{D}(\bar{x}^\top u_i)u_i\right) \\ &= \sum_{i=M+1}^{D}(x_n^\top u_i - \bar{x}^\top u_i)u_i \ . \end{aligned}$$

Minimum Error Formulation (2)

From $x_n - \tilde{x}_n = \sum_{i=M+1}^{D} (x_n^\top u_i - \bar{x}^\top u_i) u_i$, it follows that

$$
\begin{aligned}
J &= \frac{1}{N} \sum_{n \in [N]} \left(\sum_{i=M+1}^{D} (x_n^\top u_i - \bar{x}^\top u_i) u_i \right)^\top \left(\sum_{i=M+1}^{D} (x_n^\top u_i - \bar{x}^\top u_i) u_i \right) \\
&= \frac{1}{N} \sum_{n \in [N]} \sum_{i=M+1}^{D} \sum_{j=M+1}^{D} (x_n^\top u_i - \bar{x}^\top u_i)(x_n^\top u_j - \bar{x}^\top u_j) u_i^\top u_j \\
&= \frac{1}{N} \sum_{n \in [N]} \sum_{i=M+1}^{D} (x_n^\top u_i - \bar{x}^\top u_i)^2 = \sum_{i=M+1}^{D} u_i^\top S u_i \ ,
\end{aligned}
$$

where the third equality is from the orthonormality of $\{u_i\}_{i \in [D]}$. Hence, PCA wants to solve:

$$
\begin{aligned}
\underset{\{u_i\}_{i \in [D]}}{\text{minimize}} \quad & \sum_{i=M+1}^{D} u_i^\top S u_i \\
\text{subject to} \quad & u_i^\top u_j = \delta_{ij} \quad \forall i, j \in [M] \ .
\end{aligned}
$$

Minimum Error Formulation (3)

$$\underset{\{u_i\}_{i\in[D]}}{\text{minimize}} \quad \sum_{i=M+1}^{D} u_i^\top S u_i$$

$$\text{subject to} \quad u_i^\top u_j = \delta_{ij} \quad \forall i, j \in [M] \, .$$

Using Lagrange method again, we can conclude that the PCA solution is choosing $\{u_i\}_{i=M+1,\ldots,D}$ to be the eigenvectors corresponding to $(D-M)$-smallest eigenvalues, so that $\{u_i\}_{i=1,\ldots,M}$ to be the eigenvectors corresponding to M-largest eigenvalues.

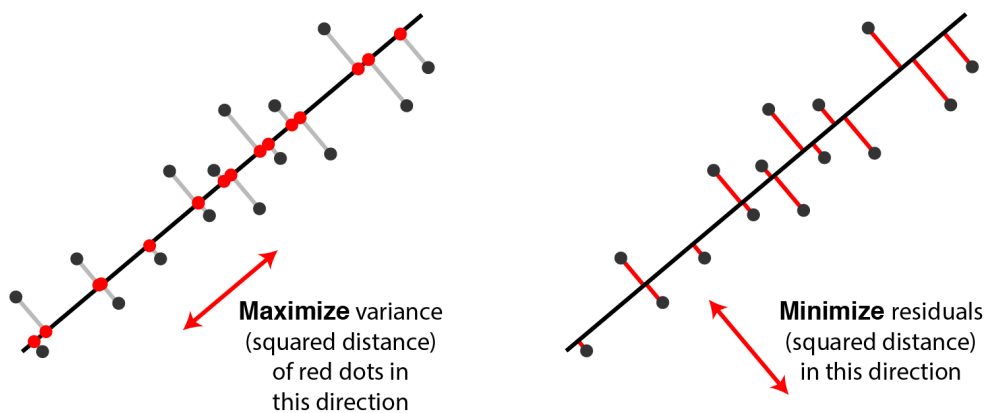


Table of Contents

1 Principal Component Analysis (PCA)

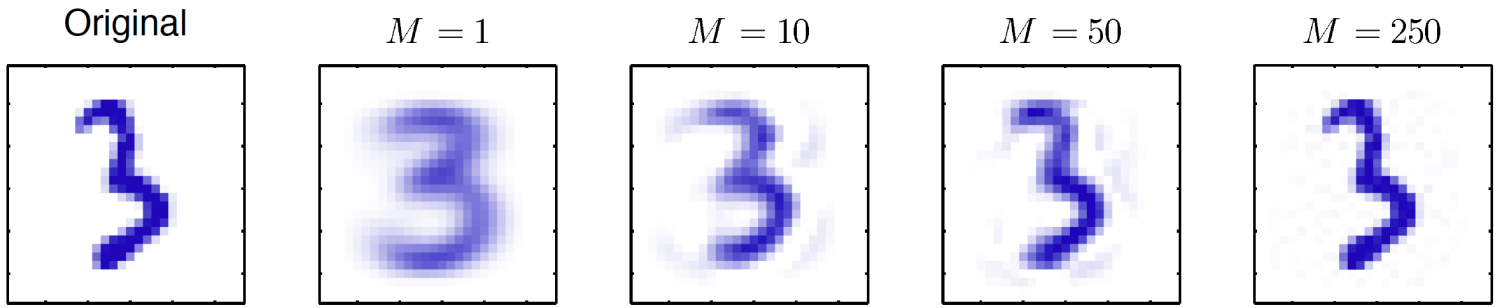
- Maximum variance formulation
- Minimum-error formation
- Applications of PCA
- PCA for high-dimensional data

2 Factor Analysis

- Maximum Likelihood Factor Analysis

An Example of PCA

28×28 grayscale images of handwriting of numeral 3, i.e., $D = 28 \times 28 = 784$, represented by $M = 1, 10, 50, 250$ principal components:



We may reduce computational cost by using $M = 250$ components only.

Table of Contents

1 Principal Component Analysis (PCA)

- Maximum variance formulation
- Minimum-error formation
- Applications of PCA
- PCA for high-dimensional data

2 Factor Analysis

- Maximum Likelihood Factor Analysis

PCA for High-dimensional Data

We typically say that a dataset is high-dimensional if the number of data points N is smaller than the dimensionality D.

- Finding eigenvectors of $D \times D$ matrix have a computational cost $O(D^3)$, which is often computationally intractable in practice.
- However, note that at least $D - N + 1$ eigenvalues are zero.

Define X to be the $(N \times D)$ dimensional centered data matrix, of which n-th row is given by $(x_n - \bar{x})^\top$ so that the covariance matrix is given as $S = \frac{1}{N} X^\top X$, and thus the corresponding eigenvector equation becomes

$$\frac{1}{N} X^\top X u_i = \lambda_i u_i \ .$$

Multiplying X, we have

$$\frac{1}{N} X X^\top (X u_i) = \lambda_i (X u_i) \ .$$

PCA for High-dimensional Data

Replacing $v_i = Xu_i$

$$\frac{1}{N} XX^\top v_i = \lambda_i v_i \ ,$$

which is an eigenvector equation for $N \times N$ matrix $\frac{1}{N} XX^\top$, and can be computed using $O(N^3)$ complexity.

Multiplying $X^\top$,

$$\frac{1}{N} X^\top X(X^\top v_i) = \lambda_i (X^\top v_i) \ ,$$

which provides a method to compute u_i from v_i:

- Obtain the eigenvectors v_i's of $XX^\top$
- Compute the normalized eigenvectors u_i for S as:

$$u_i = \frac{1}{(N\lambda_i)^{1/2}} X^\top v_i \ .$$

Table of Contents

1 Principal Component Analysis (PCA)

- Maximum variance formulation
- Minimum-error formation
- Applications of PCA
- PCA for high-dimensional data

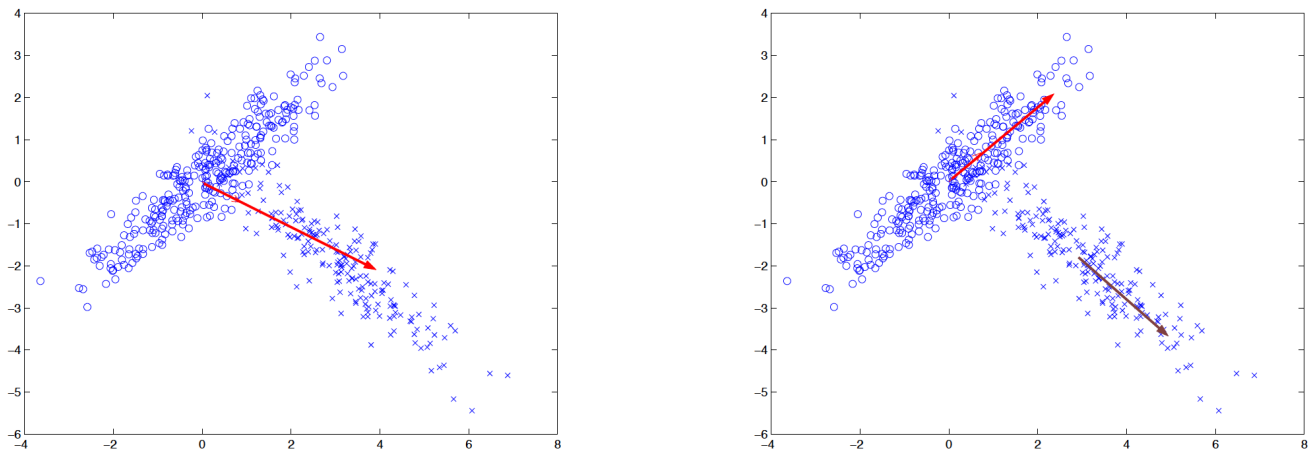
2 Factor Analysis

- Maximum Likelihood Factor Analysis

Factor Analysis

The formulation of PCA discussed previously was based on a linear projection of the data onto a subspace of lower dimensionality than the original data space, i.e., no generative model but just a data compression.

Factor analysis (FA) and probabilistic PCA (PPCA) adopt generative models so that we can address concern on unseen data.



(a) (b)

Figure: (a) PCA; (b) Mixture of factor analyzers.

Generative Model in Factor Analysis

A linear generative model relates a D-dimensional observed data $x \in \mathbb{R}^D$ to M-dimensional latent factor $y \in \mathbb{R}^M$:

$$x = Ay + \mu + \varepsilon \ ,$$

where

- $D > M$, i.e., dimensionality reduction
- $A \in \mathbb{R}^{D \times M}$ is known as the factor loading matrix
- y is assumed to be an M-dimensional Gaussian latent variable s.t. $p(y_n) = \mathcal{N}(y_n|0, I)$
- ε is an D-dimensional Gaussian noise s.t. $p(\varepsilon_n) = \mathcal{N}(\varepsilon_n|0, \Sigma)$ with a $D \times D$ diagonal matrix Σ
- μ is a parameter vector allowing non-zero mean
- x is an observation distributed as[2]:

$$p(x_n \mid y_n) = \mathcal{N}(Ay_n + \mu, \Sigma) \quad \text{and thus} \quad p(x_n) = \mathcal{N}(\mu, AA^\top + \Sigma) \ .$$

[2] See the PRML, p.91 for the details.

Learning in Factor Analysis

A linear generative model relates a D-dimensional observed data $x \in \mathbb{R}^D$ to M-dimensional latent factor $y \in \mathbb{R}^M$:

$$x = Ay + \mu + \varepsilon \,,$$

where

- The goal of factor analysis is to find the maximum likelihood estimates (MLE) of parameters:

$$\mu \,, \quad A \quad \text{and} \quad \Sigma \,.$$

- There is an indeterminacy in factor analysis since a factor loading matrix $A' = AR$ with any orthogonal matrix R produces the same covariance of x using hidden factors given by $y' = R^\top y$, but we do not care much on rotation in the subspace.

Remarks on Factor Analysis

- The M-dimensional factor plays the same role as the principal components in PCA, where the columns of A span the space of the first M principal components
- For the case of isotropic Gaussian noise, i.e., $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, the ML-FA boils down to probabilistic PCA by Tipping & Bishop (1999)
- To find the MLE solution, factor analysis performs expectation-maximization (EM) algorithm!

MLE of μ

Given a data set $X = \{x_n\}_{n\in[N]}$, noting the marginal distribution $p(x_n) = \mathcal{N}(\mu, AA^\top + \Sigma)$, the log-likelihood is given as:

$$\log p(X) = -\frac{ND}{2}\log 2\pi - \frac{N}{2}\log|C| - \frac{1}{2}\sum_{n=1}^{N}(x_n-\mu)^\top C^{-1}(x_n-\mu)$$

where $C \stackrel{\text{def}}{=} AA^\top + \Sigma$ and $|C| \stackrel{\text{def}}{=} \det(C)$.

Setting the derivative of the log-likelihood w.r.t. μ to zero, i.e., $\sum_{n=1}^{N} C^{-1}(x_n - \mu) = 0$, gives

$$\mu = \bar{x} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N} x_n \ .$$

For simplicity and without loss of generality, we assume zero mean of x, i.e., $\mu = 0$, and obtain EM algorithm for factor analysis.

EM Algorithm for Maximum Likelihood Factor Analysis

EM algorithm for data set $X = \{x_n\}_{n\in[N]}$, latent variables $Y = \{y_n\}_{n\in[N]}$, parameters $\{A, \Sigma\}$:

- E-step calculates the expected complete-data log-likelihood $\mathbb{E}_Y[\mathcal{L}_c]$, where the expectation is taken w.r.t. the posterior $p(Y \mid X, A, \Sigma)$ for fixed parameters A, Σ:

$$\begin{aligned}\mathbb{E}_Y[\mathcal{L}_c] &= \mathbb{E}_Y[\log p(X, Y \mid A, \Sigma)] \\ &= \sum_{n=1}^{N} \mathbb{E}_{y_n}\left[\log p(x_n, y_n \mid A, \Sigma)\right] \ .\end{aligned}$$

- M-step re-estimates parameters A and Σ maximizing the expected complete-data log-likelihood:

$$\begin{aligned}A^{\text{new}} &\leftarrow \arg\max_A \mathbb{E}[\mathcal{L}_c] \\ \Sigma^{\text{new}} &\leftarrow \arg\max_\Sigma \mathbb{E}[\mathcal{L}_c]\end{aligned}$$

Conditional Distribution $p(y \mid x, A, \Sigma)$

Lemma (Gaussian identities)

Consider an augmented random vector $z = [x^\top, y^\top]^\top$ of which joint distribution is

$$z = \begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix} \right) \quad \text{with cross-covariance } C = \mathbb{E}[xy^\top] \,,$$

so that $x \sim \mathcal{N}(a, A)$ and $y \sim \mathcal{N}(b, B)$. Then, the conditional distributions are given as:

$$p(x \mid y) = \mathcal{N}(x \mid a + CB^{-1}(y - b), A - CB^{-1}C^\top) \,,$$
$$p(y \mid x) = \mathcal{N}(y \mid b + C^\top A^{-1}(x - a), B - C^\top A^{-1}C) \,.$$

From the Gaussian identities[3] with $\mathbb{E}[xy^\top] = \mathbb{E}[(Ay + \varepsilon)y^\top] = A$, it follows that

$$p(y \mid x, A, \Sigma) = \mathcal{N}(y \mid \Phi x, I - \Phi A) \,,$$

where $\Phi \stackrel{\text{def}}{=} A^\top (AA^\top + \Sigma)^{-1}$.

[3] Sec 2.3.1 of the PRML.

E-Step for ML-FA

Using the product-rule,

$$\begin{aligned}
\mathbb{E}_Y[\mathcal{L}_c] &= \sum_{n=1}^{N} \mathbb{E}_{y_n}\left[\log p(x_n, y_n \mid A, \Sigma)\right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{y_n}\left[\log p(x_n \mid y_n, A, \Sigma)\right] + \mathbb{E}_{y_n}\left[\log p(y_n \mid A, \Sigma)\right] \\
&\propto -\frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{n=1}^{N}\left(x_n^\top \Sigma^{-1} x_n\right) \\
&\quad - \frac{1}{2}\sum_{n=1}^{N}\left(-2\phi_n^\top A^\top \Sigma^{-1} x_n + \mathrm{tr}\left(A^\top \Sigma^{-1} A \Phi_n\right)\right) \ ,
\end{aligned}$$

where using $p(y \mid x, A, \Sigma) = \mathcal{N}(y \mid \Phi x, I - \Phi A)$, we define

$$\begin{aligned}
\phi_n &\stackrel{\text{def}}{=} \mathbb{E}_{y_n}[y_n \mid x_n] = \Phi x_n \ , \\
\Phi_n &\stackrel{\text{def}}{=} \mathbb{E}_{y_n}[y_n y_n^\top \mid x_n] = I - \Phi A + \Phi x_n x_n^\top \Phi^\top \ .
\end{aligned}$$

(See Rubin & Thayer (1982) for the details..., Eq 2.62 in the PRML)

M-Step for ML-FA

M-step re-estimates parameters A and Σ maximizing the expected complete-data log-likelihood:

$$\begin{aligned} A^{\text{new}} &\leftarrow \arg\max_{A} \mathbb{E}[\mathcal{L}_c] \\ \Sigma^{\text{new}} &\leftarrow \arg\max_{\Sigma} \mathbb{E}[\mathcal{L}_c] \ . \end{aligned}$$

Solving $\frac{\partial \mathbb{E}_Y[\mathcal{L}_c]}{\partial A} = 0$ and $\frac{\partial \mathbb{E}_Y[\mathcal{L}_c]}{\partial \Sigma^{-1}} = 0$ with the fact that Σ is a diagonal matrix,

$$A^{\text{new}} = \left(\sum_{n=1}^{N} x_n \phi_n^\top \right) \left(\sum_{n=1}^{N} \Phi_n \right)^{-1} ,$$

$$\Sigma^{\text{new}} = \frac{1}{N} \left(\sum_{n=1}^{N} x_n x_n^\top - 2 A^{\text{new}} \sum_{n=1}^{N} \phi_n x_n^\top + A^{\text{new}} \Phi_n A^{\text{new}\top} \right) \odot I \ ,$$

where $\odot$ is the element-wise product.

EM for ML-FA

Starting from an arbitrary initialization of $A \in \mathbb{R}^{D\times M}, \Sigma \in \mathbb{R}^{D\times D}$,

- E-step:

$$\phi_n = \mathbb{E}_{y_n}[y_n \mid x_n] = \Phi x_n \ ,$$
$$\Phi_n = \mathbb{E}_{y_n}[y_n y_n^\top \mid x_n] = I - \Phi A + \Phi x_n x_n^\top \Phi^\top \ ,$$

  where $\Phi = A^\top (AA^\top + \Sigma)^{-1}$.

- M-step:

$$A^{\text{new}} = \left(\sum_{n=1}^{N} x_n \phi_n^\top\right) \left(\sum_{n=1}^{N} \Phi_n\right)^{-1} \ ,$$
$$\Sigma^{\text{new}} = \frac{1}{N}\left(\sum_{n=1}^{N} x_n x_n^\top - 2A^{\text{new}} \sum_{n=1}^{N} \phi_n x_n^\top + A^{\text{new}} \Phi_n A^{\text{new}\top}\right) \odot I \ .$$

Summary

Feature selection is necessary due to the curse of dimensionality

- Principal Component Analysis (PCA)
  - linear data compression
- Factor Analysis
  - Compression based on generative model
  - MFA
- Next lecture: non-linear transformation
  - Kernel PCA, VAE, t-SNE, GAN