Dimensionality Reduction

Ziping Zhao

School of Information Science and Technology
ShanghaiTech University, Shanghai, China

CS182: Introduction to Machine Learning (Fall 2021)
http://cs182.sist.shanghaitech.edu.cn

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Why Dimensionality/Dimension Reduction?

- Whether for classification or regression problem, observation data that we believe are informative are taken as inputs and fed to the system for decision making.
- The number of inputs (input dimensionality of the feature) often affects the time and space complexity of the learning algorithm (either classifier or regressor):
  - Having less computation reduces time complexity.
  - Having fewer parameters reduces space complexity.
- Eliminating an input deemed unnecessary saves the cost of extracting/observing it.
- Simpler models are often more robust on small data sets.
- Simpler models are more interpretable, leading to simpler explanation.
- Data visualization in 2 or 3 dimensions facilitates the detection of structure and outliers.

Feature Selection vs. Extraction

- Two main methods for reducing dimensionality: feature selection and feature extraction.
- Feature selection:
  - Choosing $k < d$ important features and discarding the remaining $d - k$.
  - Subset selection algorithms (supervised methods)
- Feature extraction:
  - Projecting the original d dimensions to k ($< d$) new dimensions.
  - Unsupervised methods (without using output information):
    - Principal component analysis (PCA)
    - Factor analysis (FA)
    - Multidimensional scaling (MDS)
    - Canonical correlation analysis (CCA)
  - Supervised methods (using output information):
    - Linear discriminant analysis (LDA)
  - The linear methods above also have nonlinear extensions.
- These k features may be interpreted as hidden or latent factors that in combination generate the observed d features.

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Subset Selection

- Goal: find the best subset of features.
- The best subset contains the least number of dimensions that most contribute to accuracy with respect to a certain task (e.g., classication, regression, visualization).
- There are $2^d - 1$ nonempty subsets of d features.
- Unless d is small, the search space is typically huge, making it impossible to conduct an exhaustive search for the best subset.
- Heuristic algorithms are often used to obtain reasonable (suboptimal) solutions in reasonable (polynomial) time.
- Two conventional approaches:
  - Forward search
  - Backward search
- More recent approach: using sparsity-inducing regularizers such as ℓ_1-norm.

Sequential Forward Search

- Start with no features and add them one by one, at each step adding the one that decreases the error measure the most, until the error cannot be further decreased.
- The error (e.g., misclassication error, mean squared error) should be measured on a validation set distinct from the training set.
- Algorithm skeleton:
  - Initialize feature set as empty set: $F = \emptyset$
  - At each iteration:
    - For each available feature x_i , train the model and calculate the error $E(F \cup \{x_i\})$ incurred on the validation set.
    - Find the best feature x_j: $j = \arg\min_i E(F \cup \{x_i\})$
    - If $E(F \cup \{x_j\}) < E(F)$ then add x_j to F and continue; else exit.
- To select k features from d, we need to train and test the model $d + (d-1) + (d-2) + \cdots + (d-k+1)$ times, which is of the order $O(d^2)$.
- No guarantee for optimal subset with greedy search.
- We can add multiple features at a time (requires more computation) or backtrack to check which previously added feature can be removed.

Sequential Backward Search

- Start with all features and do a similar process as forward search except by removing features one at a time.
- Algorithm skeleton:
  - Initialize feature set F with all features.
  - At each iteration:
    - For each feature $x_i \in F$, train the model and calculate the error $E(F \backslash \{x_i\})$ incurred on the validation set.
    - Find the best feature x_j: $j = \arg\min_i E(F \backslash \{x_i\})$
    - If $E(F \backslash \{x_i\}) < E(F)$ then remove x_j from F and continue; else exit.
- We can stop if removing a feature does not decrease the error.
- For model complexity reduction, we may decide to remove a feature if its removal causes only a slight increase in error.
- To select k features from d, we need to train and test the model $d + (d-1) + (d-2) + \cdots + (k+1)$ times.
- Backward search is more computationally demanding than forward search:
  - Usually $k \ll d$
  - Training a model with more features is more costly.

Remarks on Feature Selection

- In applications like face recognition, feature selection is not a good method for dimensionality reduction because individual pixels by themselves do not carry much discriminative information; it is the combination of values of several pixels together that carry information about the face identity.
- Dimensionality reduction in such cases is done by feature extraction methods that we will discuss next.

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Principal Component Analysis

- The projection methods aim to find a linear mapping from the d-dimensional input space ($\mathbf{x}$-space) to a k-dimensional space ($k < d$) ($\mathbf{z}$-space) with minimum information loss according to some criterion.
- Projection of $\mathbf{x}$ on the direction of $\mathbf{w}$:

$$z = \mathbf{w}^T \mathbf{x}$$

- PCA is one of the projection methods. The principal component is $\mathbf{w}_1$ such that the sample, after projection on to $\mathbf{w}_1$, is most spread out so that the difference between the sample points becomes most apparent and hence the criterion to be optimized is the variance.
- Finding the first principal component $\mathbf{w}_1$ s.t. the $\text{Var}(z_1)$ is maximized:

$$\begin{aligned}\text{Var}(z_1) =& \text{Var}(\mathbf{w}_1^T \mathbf{x}) = \mathbb{E}[(\mathbf{w}_1^T \mathbf{x} - \mathbb{E}(\mathbf{w}_1^T \mathbf{x}))^2] = \mathbb{E}[(\mathbf{w}_1^T \mathbf{x} - \mathbf{w}_1^T \boldsymbol{\mu})(\mathbf{w}_1^T \mathbf{x} - \mathbf{w}_1^T \boldsymbol{\mu})] \\ =& \mathbb{E}[\mathbf{w}_1^T (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{w}_1] = \mathbf{w}_1^T \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] \mathbf{w}_1 = \mathbf{w}_1^T \boldsymbol{\Sigma} \mathbf{w}_1\end{aligned}$$

where

$$\text{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] = \boldsymbol{\Sigma}$$

Optimization Problem for First Principal Component I

- The optimization problem is given by

$$\begin{aligned} \underset{\mathbf{w}_1}{\text{maximize}} \quad & \text{Var}(z_1) = \mathbf{w}_1^T \mathbf{\Sigma} \mathbf{w}_1 \\ \text{subject to} \quad & \|\mathbf{w}_1\| = 1 \end{aligned}$$

  which is a constrained optimization problem and the Lagrangian is given by

$$\mathcal{L}(\mathbf{w}_1, \alpha) = -\mathbf{w}_1^T \mathbf{\Sigma} \mathbf{w}_1 + \alpha(\mathbf{w}_1^T \mathbf{w}_1 - 1)$$

  where α is the Lagrange multiplier.

- Taking the derivative of the Lagrangian w.r.t. $\mathbf{w}_1$ and setting it to $\mathbf{0}$, we get an eigenvalue equation for the (first) principal component $\mathbf{w}_1$:

$$\mathbf{\Sigma} \mathbf{w}_1 = \alpha \mathbf{w}_1$$

- Because we have

$$\mathbf{w}_1^T \mathbf{\Sigma} \mathbf{w}_1 = \alpha \mathbf{w}_1^T \mathbf{w}_1 = \alpha$$

  we choose the eigenvector of $\mathbf{\Sigma}$ with the largest eigenvalue $\lambda_1 = \alpha$ for the variance $\text{Var}(z_1)$ to be maximum.

Optimization Problem for First Principal Component II

- The optimization problem for the first principal component can also be written as

$$\underset{\mathbf{w}_1}{\text{maximize}} \quad \frac{\mathbf{w}_1^T \boldsymbol{\Sigma} \mathbf{w}_1}{\mathbf{w}_1^T \mathbf{w}_1}$$

  which is to maximize the Rayleigh quotient.

Optimization Problem for Second Principal Component

- The second principal component $\mathbf{w}_2$ should also maximize the variance $\text{Var}(z_2)$ with the projection $z_2 = \mathbf{w}_2^T\mathbf{x}$ uncorrelated to z_1:

$$\text{Cov}(z_1, z_2) = \mathbb{E}[(\mathbf{w}_1^T\mathbf{x} - \mathbf{w}_1^T\boldsymbol{\mu})(\mathbf{w}_2^T\mathbf{x} - \mathbf{w}_2^T\boldsymbol{\mu})] = \mathbf{w}_1^T\boldsymbol{\Sigma}\mathbf{w}_2 = \lambda_1\mathbf{w}_2^T\mathbf{w}_1 = 0$$

- The optimization problem is

$$\begin{aligned} \underset{\mathbf{w}_2}{\text{maximize}} \quad & \text{Var}(z_2) = \mathbf{w}_2^T\boldsymbol{\Sigma}\mathbf{w}_2 \\ \text{subject to} \quad & \|\mathbf{w}_2\| = 1,\ \mathbf{w}_2^T\mathbf{w}_1 = 0 \end{aligned}$$

- The Lagrangian:

$$\mathcal{L}(\mathbf{w}_2, \alpha, \beta) = -\mathbf{w}_2^T\boldsymbol{\Sigma}\mathbf{w}_2 + \alpha(\mathbf{w}_2^T\mathbf{w}_2 - 1) + \beta(\mathbf{w}_2^T\mathbf{w}_1 - 0)$$

- Taking the derivative of the Lagrangian w.r.t. $\mathbf{w}_2$ and setting it to $\mathbf{0}$, we get

$$2\boldsymbol{\Sigma}\mathbf{w}_2 - 2\alpha\mathbf{w}_2 - \beta\mathbf{w}_1 = \mathbf{0}$$

- We can show that $\beta = 0$ and hence have this eigenvalue equation $\boldsymbol{\Sigma}\mathbf{w}_2 = \alpha\mathbf{w}_2$, implying that $\mathbf{w}_2$ is the eigenvector of $\boldsymbol{\Sigma}$ with the second largest eigenvalue.

Optimization Problem for Other Principal Components

- Similarly, we can show that the other dimensions are given by the eigenvectors of $\mathbf{\Sigma}$ with decreasing eigenvalues.
- The sample covariance $\mathbf{S} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$ is symmetric, so, for two different eigenvalues, the eigenvectors are orthogonal.
  - If $\mathbf{S}$ is positive definite, then all its eigenvalues are positive.
  - If $\mathbf{S}$ is singular, then its rank, the effective dimensionality, is k with $k < d$ and λ_i, $i = k+1, \ldots, d$ are 0 (λ_i are sorted in descending order).
  - The k eigenvectors with nonzero eigenvalues are the dimensions of the reduced space.
- The first eigenvector (the one with the largest eigenvalue), $\mathbf{w}_1$, namely, the principal component, explains the largest part of the variance; the second explains the second largest; and so on.
- The variance minimization method provides a statistical view for PCA.

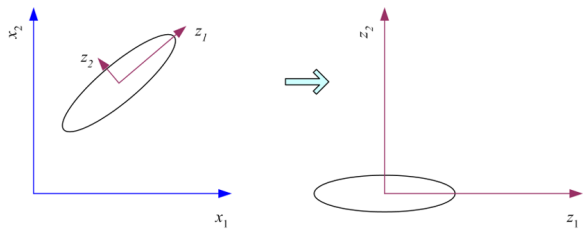
What PCA Does I

- Transformation of data:

$$\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \mathbf{m})$$

where the k columns of $\mathbf{W} \in \mathbb{R}^{d \times k}$ are the k leading eigenvectors of the sample covariance $\mathbf{S}$ and $\mathbf{m}$ is the sample mean.

- PCA intuition: centering the data at the origin and rotating the axes:



If $\text{Var}(z_2)$ is too small, it can be ignored to reduce the dimensionality from 2 to 1.

- After the linear transformation, we get a k-dimensional space whose dimensions are the eigenvectors, and the variances over them are equal to the eigenvalues.

What PCA Does II

- The eigenvalue decomposition or spectral decomposition of the sample covariance $\mathbf{S}$ is given by

$$\mathbf{S} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$

where $\mathbf{\Lambda} \in \mathbb{R}^{d\times d}$ and $\mathbf{Q} \in \mathbb{R}^{d\times d}$ with $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ are the eigenvalue matrix and eigenvector matrix, respectively, and hence

$$\mathbf{Q}^T\mathbf{S}\mathbf{Q} = \mathbf{\Lambda}$$

- We have

$$\text{Cov}(\mathbf{z}) = \mathbf{W}^T\mathbf{S}\mathbf{W} = \mathbf{\Lambda}_k$$

which is a diagonal matrix.

- PCA intuition: find a matrix $\mathbf{W}$ s.t. the linear transformed data $\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \mathbf{m})$ has diagonal covariance; that is, we would like to get uncorrelated z_i.
- PCA does not use output information and hence is a one-group procedure.

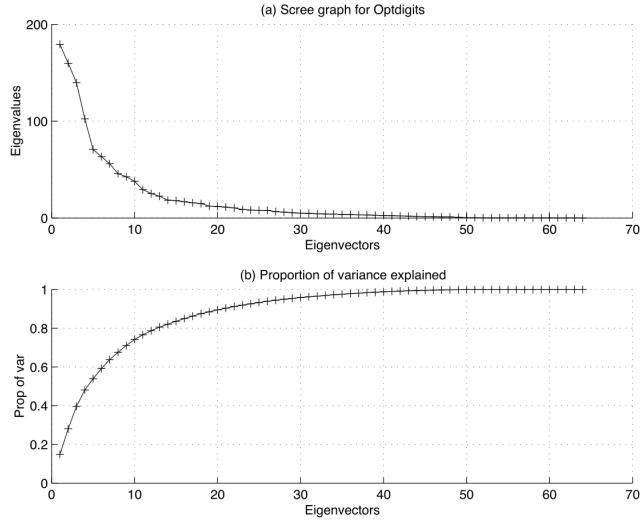
How to Choose k?

- Proportion of variance (PoV) explained:

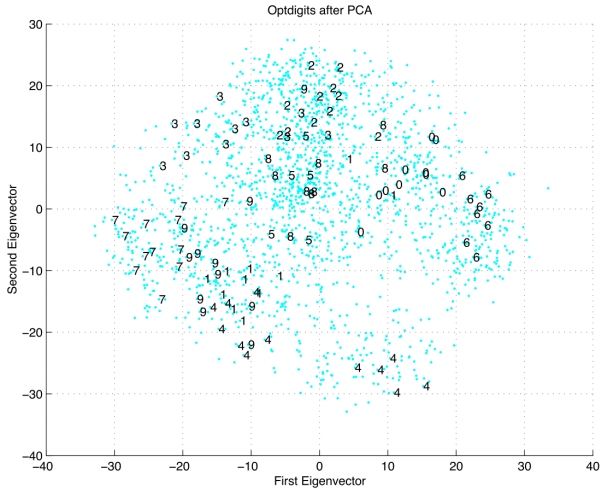
$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

  where λ_d are sorted in descending order.
- Typically, stop at PoV > 0.9.
- Scree graph plotting PoV against k; stop at "elbow".

Scree Graph


(a) Scree graph for Optdigits
Eigenvalues
Eigenvectors
0
100
200
0
10
20
30
40
50
60
70
(b) Proportion of variance explained
Prop of var
Eigenvectors
0
0.2
0.4
0.6
0.8
1
0
10
20
30
40
50
60
70

Scatterplot in Lower-Dimensional Space



An Alternative Equivalent Formulation for PCA

- Given the transformation of data:

$$\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \mathbf{m})$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ with $\mathbf{W}\mathbf{W}^T = \mathbf{I}$. We have

$$\mathbf{x} = \mathbf{m} + \mathbf{W}\mathbf{z}$$

- If $\mathbf{W} \in \mathbb{R}^{d \times k}$ with columns to be the principal components, the reconstruction of $\mathbf{x}^{(\ell)}$ from its representation in the lower-dimensional $\mathbf{z}$-space is

$$\hat{\mathbf{x}}^{(\ell)} = \mathbf{m} + \mathbf{W}\mathbf{z}^{(\ell)} \quad \text{or} \quad \mathbf{x}^{(\ell)} = \mathbf{m} + \mathbf{W}\mathbf{z}^{(\ell)} + \boldsymbol{\epsilon}^{(\ell)}$$

- It can be proved that among all orthogonal linear projections, PCA minimizes the reconstruction error (a geometric view of PCA), i.e.,

$$\begin{aligned} &\underset{\mathbf{m},\mathbf{W},\{\mathbf{z}^{(\ell)}\}}{\text{minimize}} \quad \frac{1}{2}\sum_{\ell} \|\mathbf{x}^{(\ell)} - (\mathbf{m} + \mathbf{W}\mathbf{z}^{(\ell)})\|_2^2 = \frac{1}{2}\|\mathbf{X} - \mathbf{m}\mathbf{1}^T - \mathbf{W}\mathbf{Z}\|_F^2 \\ &\text{subject to} \quad \mathbf{W}^T\mathbf{W} = \mathbf{I} \end{aligned}$$

- We can pre-subtract the sample mean from $\mathbf{x}^{(\ell)}$ or constrain $\sum_\ell \mathbf{z}^{(\ell)} = \mathbf{0}$ if we expect $\mathbf{m}$ to be the sample mean estimate of $\mathbf{x}^{(\ell)}$.

Probabilistic PCA

- PCA model is not a generative model, since the low-dimensional representation $\{\mathbf{z}^{(\ell)}\}$ and the error $\{\boldsymbol{\epsilon}^{(\ell)}\}$ are not treated as random variables. As a consequence, the PCA model cannot be used to generate new samples of the random variable $\mathbf{x}$.
- To address this issue, the probabilistic PCA (PPCA) assume that $\mathbf{z}$ and $\boldsymbol{\epsilon}$ are independent random variables with some pdfs, then it generates an $\mathbf{x}$ by

$$\mathbf{x} = \mathbf{m} + \mathbf{W}\mathbf{z} + \boldsymbol{\epsilon}$$

- Let the mean and covariance of $\mathbf{z}$ be denoted by $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ (commonly assuming $\boldsymbol{\Sigma}_z = \mathbf{I}_k$), respectively and the mean and covariance of $\mathbf{e}$ be denoted by $\mathbf{0}$ and $\boldsymbol{\Sigma}_\epsilon$ (commonly assuming $\boldsymbol{\Sigma}_\epsilon = \psi^2 \mathbf{I}_d$). Then we have

$$\boldsymbol{\mu} = \mathbf{m} + \mathbf{W}\boldsymbol{\mu}_z \quad \text{and} \quad \boldsymbol{\Sigma} = \mathbf{W}\boldsymbol{\Sigma}_z\mathbf{W}^T + \boldsymbol{\Sigma}_\epsilon$$

- Then we can estimate $\mathbf{m}$, $\mathbf{W}$, $\boldsymbol{\mu}_z$, $\boldsymbol{\Sigma}_z$, and $\boldsymbol{\Sigma}_\epsilon$ from the estimates of $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$ or directly from the sample $\{\mathbf{x}^{(\ell)}\}$ through, say, MLE.

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Factor Analysis

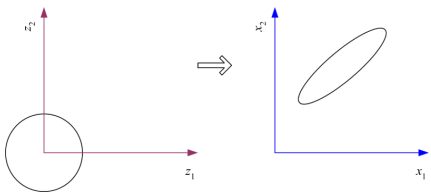
- Factor analysis (FA) or exploratory factor analysis (EFA) assumes that there is a set of latent factors z_j, $j = 1, \ldots, k$ which when acting in combination generate the observed variables $\mathbf{x}$.
- The goal of FA is to characterize the dependency among the observed variables by means of a smaller number of factors, i.e., in a smaller dimensional space without loss of information measured as the correlation between variables.
- Problem settings:
  - Sample $\mathcal{X} = \{\mathbf{x}^{(\ell)}\}$: drawn from some unknown probability density with $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}$ and $\text{Cov}(\mathbf{x}) = \boldsymbol{\Sigma}$.
  - Factors z_j are unit normals and uncorrelated: $\mathbb{E}[z_j] = 0$, $\text{Var}(z_j) = 1$, $\text{Cov}(z_i, z_j) = 0$, $i \neq j$.
  - Noise sources ϵ_i to explain what is not explained by the factors: $\mathbb{E}[\epsilon_i] = 0$, $\text{Var}(\epsilon_i) = \psi_{ii} = \psi_i^2$, $\text{Cov}(\epsilon_i, \epsilon_j) = \psi_{ij} = 0$, $i \neq j$, $\text{Cov}(\epsilon_i, z_j) = 0$, $\forall i, j$

Relationships Between Factors and Input Dimensions

- Each of the d input dimensions x_i, $i = 1, \ldots, d$, can be expressed as a weighted sum of the k $(< d)$ factors z_j, $j = 1, \ldots, k$, plus some residual error term:

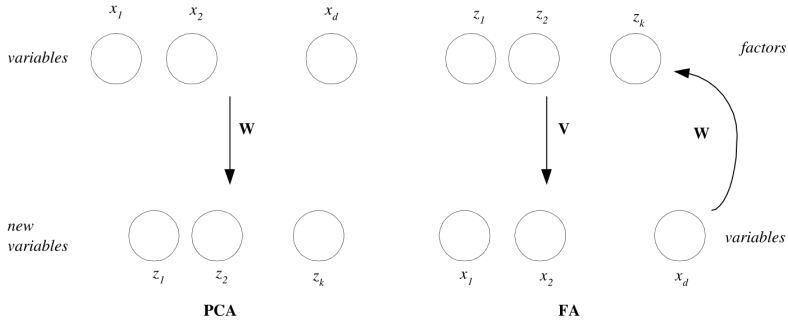
$$x_i - \mu_i = \sum_{j=1}^{k} v_{ij} z_j + \epsilon_i \quad \text{or} \quad \mathbf{x} - \boldsymbol{\mu} = \mathbf{V}\mathbf{z} + \boldsymbol{\epsilon}$$

  where $\mathbf{V} \in \mathbb{R}^{d \times k}$ is a matrix of weights, called factor loadings.
- Without loss of generality, we assume that $\boldsymbol{\mu} = \mathbf{0}$.
- The factors z_j are independent unit normals that are stretched, rotated and translated to generate the inputs $\mathbf{x}$.



PCA vs. FA

- The direction of FA is opposite to that of PCA:
  - PCA (from $\mathbf{x}$ to $\mathbf{z}$):
$$\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu})$$
  - FA (from $\mathbf{z}$ to $\mathbf{x}$ – generative model):
$$\mathbf{x} - \boldsymbol{\mu} = \mathbf{V}\mathbf{z} + \boldsymbol{\epsilon}$$



Covariance Matrix

- Given that $\text{Var}(z_j) = 1$ and $\text{Var}(\epsilon_j) = \psi_i^2$,

$$\text{Var}(x_i) = \sum_{j=1}^{k} v_{ij}^2 \text{Var}(z_j) + \text{Var}(\epsilon_i) = \sum_{j=1}^{k} v_{ij}^2 + \psi_i^2$$

  where the first part $\sum_{j=1}^{k} v_{ij}^2$ is the variance explained by the common factors and the second part (ψ_i^2) is the variance specific to x_i.
- Then, the covariance matrix:

$$\begin{aligned}\mathbf{\Sigma} = \text{Cov}(\mathbf{x}) &= \text{Cov}(\mathbf{Vz} + \boldsymbol{\epsilon}) \\ &= \text{Cov}(\mathbf{Vz}) + \text{Cov}(\boldsymbol{\epsilon}) \\ &= \mathbf{V}\text{Cov}(\mathbf{z})\mathbf{V}^T + \mathbf{\Psi} \\ &= \mathbf{V}\mathbf{V}^T + \mathbf{\Psi}\end{aligned}$$

  where $\mathbf{\Psi} = \text{diag}(\boldsymbol{\psi})$ with $\boldsymbol{\psi} = [\psi_1^2, \ldots, \psi_d^2]$.

2-Factor Example for Illustration

- Let

$$\mathbf{x} = (x_1, x_2)^T \qquad \mathbf{V} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}$$

- Since

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} = \mathbf{V}\mathbf{V}^T + \mathbf{\Psi} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} \\ v_{12} & v_{22} \end{bmatrix} + \begin{bmatrix} \Psi_1 & 0 \\ 0 & \Psi_2 \end{bmatrix}$$

  we have

$$\sigma_{12} = \mathrm{Cov}(x_1, x_2) = v_{11}v_{21} + v_{12}v_{22}$$

- If x_1 and x_2 have high covariance, then they are related through a factor:
  - If it is the first factor, then v_{11} and v_{21} will both be high.
  - If it is the second factor, then v_{12} and v_{22} will both be high.
- If x_1 and x_2 have low covariance, then they depend on different factors:
  - In each of the products $v_{11}v_{21}$ and $v_{12}v_{22}$, one term will be high and the other will be low.

Factor Loadings

- Because

$$\begin{aligned}\text{Cov}(x_1, z_1) =& \text{Cov}(v_{11}z_1 + v_{12}z_2 + \epsilon_1, z_1) \\ =& \text{Cov}(v_{11}z_1, z_1) = v_{11}\text{Var}(z_1) = v_{11}\end{aligned}$$

and similarly,

$$\begin{aligned}\text{Cov}(x_1, z_2) &= v_{12} \\ \text{Cov}(x_2, z_1) &= v_{21} \\ \text{Cov}(x_2, z_2) &= v_{22}\end{aligned}$$

so we have

$$\text{Cov}(\mathbf{x}, \mathbf{z}) = \mathbf{V}$$

i.e., the factor loadings $\mathbf{V}$ represent the covariances or correlations between the variables and the factors.

Factor Analysis

- Since
$$\mathbf{\Sigma} = \mathbf{V}\mathbf{V}^T + \mathbf{\Psi}$$
if there are only a few factors (i.e., $k \ll d$), then we can get a simplified structure for $\mathbf{S}$.
- The number of parameters is reduced from $d(d+1)/2$ (for $\mathbf{S}$) to $dk + d$ (for $\mathbf{V}\mathbf{V}^T + \mathbf{\Psi}$).
- Special cases:
  - Probabilistic PCA (PPCA): $\mathbf{\Psi} = \psi^2\mathbf{I}$ (i.e., all ψ_i^2 are equal)
  - Conventional PCA: $\mathbf{\Psi} = \mathbf{0}$, (i.e., $\psi_i^2 = 0$)
- The solution of factor loadings are not unique
$$\mathbf{V}\mathbf{V}^T = \mathbf{V}\mathbf{T}\mathbf{T}^T\mathbf{V}^T = (\mathbf{V}\mathbf{T})(\mathbf{V}\mathbf{T})^T = \tilde{\mathbf{V}}\tilde{\mathbf{V}}^T$$
for any orthogonal matrix $\mathbf{T} \in \mathbb{R}^{k\times k}$.
- The factors can be rotated to give maximum loading on as few factors as possible for each variable, to make the factors interpretable, for knowledge extraction.

Estimation of FA I

- Given $\mathbf{S}$ as the estimator of $\mathbf{\Sigma}$, we want to find $\mathbf{V}$ and $\mathbf{\Psi}$ such that

$$\mathbf{S} = \mathbf{V}\mathbf{V}^T + \mathbf{\Psi} \qquad (\text{or : } \mathbf{S} \approx \mathbf{V}\mathbf{V}^T + \mathbf{\Psi})$$

- A naive method is to obtain $\mathbf{V}$ firstly via PCA and then $\mathbf{\Psi}$ by taking directly the residual's sample variance.
- A joint estimation method over $\mathbf{V}$ and $\mathbf{\Psi}$ can also be chosen to minimize

$$\begin{aligned} &\underset{\mathbf{V},\mathbf{\Psi}}{\text{minimize}} && \|\mathbf{S} - (\mathbf{V}\mathbf{V}^T + \mathbf{\Psi})\|_F^2 \\ &\text{subject to} && \mathbf{\Psi} \succ \mathbf{0} \end{aligned}$$

Estimation of FA II

- The MLE for FA directly learns the parameters from raw data $\mathbf{x}^{(\ell)}$.
- It assumes that the data are generated from a certain statistical model, typically the multivariate Gaussian distribution.
- Then the parameters are estimated by maximizing the likelihood function

$$\begin{aligned} \underset{\boldsymbol{\mu},\mathbf{V},\boldsymbol{\Psi}}{\text{minimize}} \quad & \frac{N}{2}\log\det(\boldsymbol{\Sigma}) + \frac{1}{2}\sum_{\ell}(\mathbf{x}^{(\ell)}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}^{(\ell)}-\boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{\Sigma} = \mathbf{V}\mathbf{V}^T + \boldsymbol{\Psi} \\ & \boldsymbol{\Psi} \succ \mathbf{0} \end{aligned}$$

- Computationally this process is complex.
- In general, there is no closed-form solution to this optimization problem so iterative methods are applied.

Dimensionality Reduction

- FA can be used for dimensionality reduction when $k < d$.
- For dimensionality reduction, FA offers no advantage over PCA except the interpretability of factors allowing the identification of common causes, a simple explanation, and knowledge extraction.

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

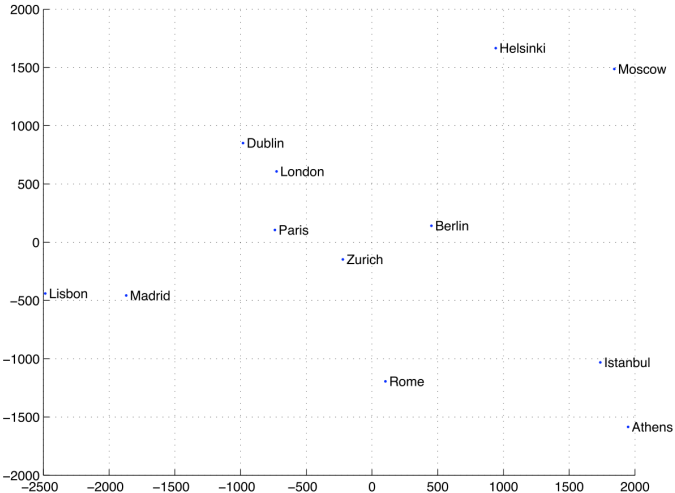
Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Multidimensional Scaling

- Problem formulation:
  - Given the pairwise distances between pairs of points in some space (but the exact coordinates of the points and their dimensionality are unknown).
  - We want to embed the points in a lower-dimensional space (e.g., two-dimensional space) such that the pairwise Euclidean distances in this space are as close as possible to those in the original space.
- The projection to the lower-dimensional space is not unique because the pairwise distances are invariant to such operations as translation, rotation, and reflection.
- MDS is closely related to the Euclidean distance matrix (EDM) problem.

MDS Embedding of Cities



Derivation I

- Sample $\mathcal{X} = \{\mathbf{x}^{(\ell)} \in \mathbb{R}^d\}_{\ell=1}^N$ which is not available as feature vectors
- Squared Euclidean distance between points r and s:

$$\begin{aligned} d_{rs}^2 = \|\mathbf{x}^{(r)} - \mathbf{x}^{(s)}\|^2 &= \sum_{j=1}^{d} (x_j^{(r)} - x_j^{(s)})^2 \\ &= \sum_{j=1}^{d} (x_j^{(r)})^2 + \sum_{j=1}^{d} (x_j^{(s)})^2 - 2\sum_{j=1}^{d} x_j^{(r)} x_j^{(s)} \\ &= b_{rr} + b_{ss} - 2b_{rs} \end{aligned} \tag{1}$$

where

$$b_{rs} = \sum_{j=1}^{d} x_j^{(r)} x_j^{(s)} = (\mathbf{x}^{(r)})^T \mathbf{x}^{(s)} \qquad \text{(dot product of } \mathbf{x}^{(r)} \text{ and } \mathbf{x}^{(s)}\text{)}$$

or in matrix form with $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}]$:

$$\mathbf{B} = \mathbf{X}^T \mathbf{X}$$

Derivation II

- Centering of data to constrain the solution:

$$\sum_{\ell=1}^{N} x_j^{(\ell)} = 0, \ \forall j = 1, \ldots, d$$

- Summing up equation (1) on r, s, and both r and s and defining

$$T = \sum_{\ell=1}^{N} b_{\ell\ell} = \sum_{\ell=1}^{N} \sum_{j=1}^{d} (x_j^{(\ell)})^2$$

we get

$$\sum_{r=1}^{N} d_{rs}^2 = T + N b_{ss}, \quad \sum_{s=1}^{N} d_{rs}^2 = N b_{rr} + T, \quad \sum_{r=1}^{N} \sum_{s=1}^{N} d_{rs}^2 = 2NT$$

Derivation III

- By defining

$$d_{*s}^2 = \frac{1}{N}\sum_r d_{rs}^2 \qquad d_{r*}^2 = \frac{1}{N}\sum_s d_{rs}^2 \qquad d_{**}^2 = \frac{1}{N^2}\sum_r\sum_s d_{rs}^2$$

and using equation (1), we get

$$b_{rs} = \frac{1}{2}(d_{r*}^2 + d_{*s}^2 - d_{**}^2 - d_{rs}^2)$$

- We have obtained the form of matrix $\mathbf{B}$.
- $\mathbf{B} = \mathbf{X}^T\mathbf{X}$ is p.s.d. with positive eigenvalues, so it can be expressed as its spectral decomposition:

$$\mathbf{B} = \mathbf{C}\mathbf{D}\mathbf{C}^T = \mathbf{C}\mathbf{D}^{1/2}\mathbf{D}^{1/2}\mathbf{C}^T = (\mathbf{C}\mathbf{D}^{1/2})(\mathbf{C}\mathbf{D}^{1/2})^T$$

where $\mathbf{C}$ is the matrix whose columns are the eigenvectors of $\mathbf{B}$ and $\mathbf{D}^{1/2}$ is the diagonal matrix whose diagonal elements are the square roots of the eigenvalues.

Projection to Lower-Dimensional Space

- If we ignore the eigenvectors of $\mathbf{B}$ with very small eigenvalues (the eigenvalues of $\mathbf{B} = \mathbf{X}^T\mathbf{X}$ are the same as the eigenvalues of $\mathbf{X}\mathbf{X}^T$) and keep the largest k ones, $\mathbf{C}\mathbf{D}^{1/2}$ will only be a low-rank approximation of $\mathbf{X}^T$.
- Let $\mathbf{c}_j \in \mathbb{R}^N$ be the k eigenvectors chosen with corresponding eigenvalues λ_j.
- New dimensions in k-dimensional embedding space:

$$z_j^{(\ell)} = \sqrt{\lambda_j} c_j^{(\ell)}, \quad j = 1, \ldots, k, \ \ell = 1, \ldots, N$$

So the new coordinates of instance ℓ are given by the ℓ-th elements of the eigenvectors after normalization, i.e.,

$$[\sqrt{\lambda_1} c_1^{(\ell)}, \ldots, \sqrt{\lambda_k} c_k^{(\ell)}]^T \in \mathbb{R}^k$$

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

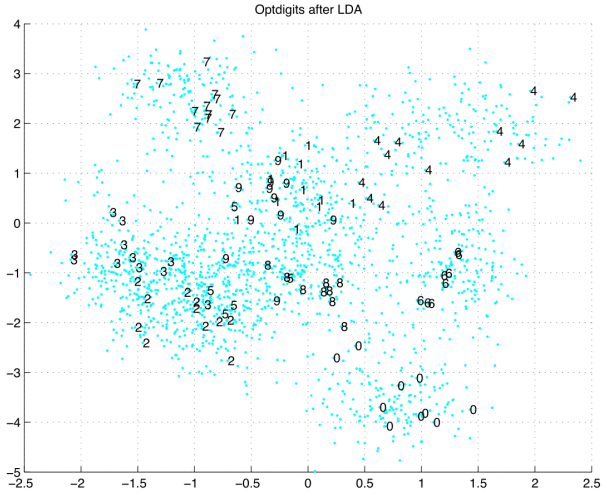
Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Linear Discriminant Analysis

- Unlike PCA, FA and MDS, LDA is a supervised dimensionality reduction method.
- LDA is typically used with a classifier for classification problems.
- Goal: the classes are well-separated after projecting to a low-dimensional space by utilizing the label information (output information).

Example



2-Class Case

- Given sample $\mathcal{X} = \{(\mathbf{x}^{(\ell)}, r^{(\ell)})\}$, where $r^{(\ell)} = 1$ if $\mathbf{x}^{(\ell)} \in C_1$ (class 1) and $r^{(\ell)} = 0$ if $\mathbf{x}^{(\ell)} \in C_2$ (class 2).
- Find vector $\mathbf{w}$ on which the data are projected such that the examples from C_1 and C_2 are as well separated as possible.
- Projection of $\mathbf{x}$ onto $\mathbf{w}$ (dimensionality reduced from d to 1):

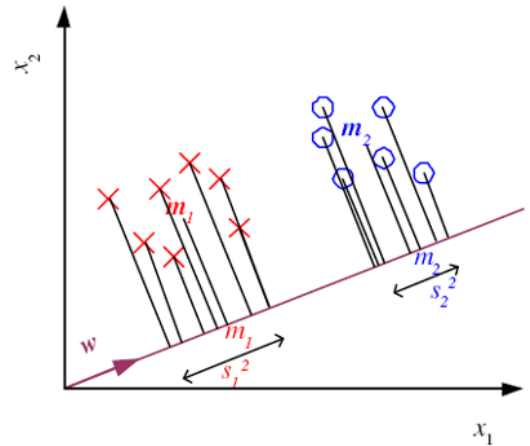
$$z = \mathbf{w}^T \mathbf{x}$$

- $\mathbf{m}_i \in \mathbb{R}^d$ and $m_i \in \mathbb{R}$ are sample means of C_i before and after projection:

$$m_1 = \frac{\sum_\ell \mathbf{w}^T \mathbf{x}^{(\ell)} r^{(\ell)}}{\sum_\ell r^{(\ell)}} = \mathbf{w}^T \mathbf{m}_1$$

$$m_2 = \frac{\sum_\ell \mathbf{w}^T \mathbf{x}^{(\ell)} (1 - r^{(\ell)})}{\sum_\ell (1 - r^{(\ell)})} = \mathbf{w}^T \mathbf{m}_2$$

Projection



Between-Class Scatter

- Between-class scatter (in a form of normalized sample variance / covariance matrix):

$$\begin{aligned}(m_1 - m_2)^2 =& (\mathbf{w}^T\mathbf{m}_1 - \mathbf{w}^T\mathbf{m}_2)^2 \\ =& \mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w} \\ =& \mathbf{w}^T\mathbf{S}_B\mathbf{w}\end{aligned}$$

where the between-class scatter matrix

$$\begin{aligned}\mathbf{S}_B =& (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \\ =& \frac{4}{N_1 + N_2}(N_1(\mathbf{m}_1 - \mathbf{m})(\mathbf{m}_1 - \mathbf{m})^T + N_2(\mathbf{m}_2 - \mathbf{m})(\mathbf{m}_2 - \mathbf{m})^T)\end{aligned}$$

where $\mathbf{m}$ is the mean of the class means, i.e., $\mathbf{m} = \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)$.

Within-Class Scatter

- Within-class scatter:

$$\begin{aligned} s_1^2 &= \sum_\ell (\mathbf{w}^T \mathbf{x}^{(\ell)} - m_1)^2 r^{(\ell)} \\ &= \sum_\ell \mathbf{w}^T (\mathbf{x}^{(\ell)} - \mathbf{m}_1)(\mathbf{x}^{(\ell)} - \mathbf{m}_1)^T \mathbf{w} r^{(\ell)} \\ &= \mathbf{w}^T \mathbf{S}_1 \mathbf{w} \end{aligned}$$

  where the within-class scatter matrix $\mathbf{S}_1 = \sum_\ell (\mathbf{x}^{(\ell)} - \mathbf{m}_1)(\mathbf{x}^{(\ell)} - \mathbf{m}_1)^T r^{(\ell)}$. Also,

$$s_2^2 = \mathbf{w}^T \mathbf{S}_2 \mathbf{w}$$

  with $\mathbf{S}_2 = \sum_\ell (\mathbf{x}^{(\ell)} - \mathbf{m}_2)(\mathbf{x}^{(\ell)} - \mathbf{m}_2)^T (1 - r^{(\ell)})$.

- So, the total within-class scatter

$$s_1^2 + s_2^2 = \mathbf{w}^T \mathbf{S}_W \mathbf{w}$$

  where

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

Fisher's Linear Discriminant

- Fisher's linear discriminant refers to the vector $\mathbf{w}$ that maximizes the Fisher criterion (a.k.a. generalized Rayleigh quotient):

$$\underset{\mathbf{w}}{\text{maximize}} \quad J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

  which is equivalent to solve

$$\begin{aligned} \underset{\mathbf{w}}{\text{maximize}} \quad & \mathbf{w}^T \mathbf{S}_B \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^T \mathbf{S}_W \mathbf{w} = 1 \end{aligned}$$

- We can prove the optimal solution satisfies the following generalized eigenvalue problem:

$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$$

  or, if $\mathbf{S}_W$ is nonsingular, an equivalent eigenvalue problem:

$$\mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w} = \lambda \mathbf{w}$$

2-Class Case

- For the 2-class case, we note that

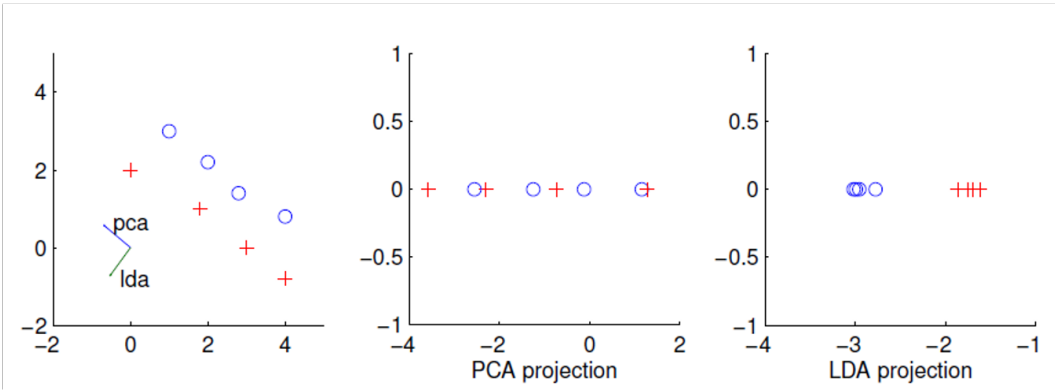
$$\mathbf{S}_B\mathbf{w} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w} = c(\mathbf{m}_1 - \mathbf{m}_2)$$

  for some constant c and hence $\mathbf{S}_B\mathbf{w}$ (also $\mathbf{S}_W\mathbf{w}$) is in the same direction of $\mathbf{m}_1 - \mathbf{m}_2$.
- So we get

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) = c(\mathbf{S}_1 + \mathbf{S}_2)^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

  where c is some irrelevant constant factor.
- We have projected the samples from d dimensions to 1, i.e., dimensionality reduction, and any classification method can be used afterward.
- Remember that when $p(\mathbf{x}|C_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$ (i.e., homoscedasticity), we have a linear discriminant where $\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$, and we see that Fisher's linear discriminant is optimal if the classes are normally distributed. But Fisher's linear discriminant can be used even when the classes are not normal.

PCA vs. LDA



$K > 2$ Classes I

- Find the matrix $\mathbf{W} \in \mathbb{R}^{d \times k}$ such that

$$\mathbf{z} = \mathbf{W}^T \mathbf{x} \in \mathbb{R}^k$$

- Within-class scatter matrix for C_i:

$$\mathbf{S}_i = \sum_{\ell} r_i^{(\ell)} (\mathbf{x}^{(\ell)} - \mathbf{m}_i)(\mathbf{x}^{(\ell)} - \mathbf{m}_i)^T$$

  where $r_i^{(\ell)} = 1$ if $\mathbf{x}^{(\ell)} \in C_i$ and 0 otherwise.
- Total within-class scatter matrix:

$$\mathbf{S}_W = \sum_{i=1}^{k} \mathbf{S}_i$$

$K > 2$ Classes II

- Between-class scatter matrix:

$$\mathbf{S}_B = \sum_{i=1}^{K} N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$$

  where $\mathbf{m}$ is the overall mean (i.e., mean of the class means) or sometime chosen as the mean weighted by sample numbers and $N_i = \sum_\ell r_i^{(\ell)}$.
- The optimal solution is the matrix $\mathbf{W}$ that maximizes

$$\underset{\mathbf{W}}{\text{maximize}} \quad J(\mathbf{W}) = \frac{\det(\mathbf{W}^T \mathbf{S}_B \mathbf{W})}{\det(\mathbf{W}^T \mathbf{S}_W \mathbf{W})}$$

  which corresponds to the eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$ with the largest eigenvalues.
- Take $k \leq K - 1$: since $\mathbf{S}_B$ is the sum of K rank-1 matrices and only $K - 1$ of them are independent, $\mathbf{S}_B$ has a maximum rank of $K - 1$.

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Canonical Correlation

- CCA (a.k.a. canonical variates analysis) is an unsupervised problem for two sets of variables $\mathcal{X} = \{\mathbf{x}^{(\ell)}, \mathbf{y}^{(\ell)}\}_{\ell=1}^{N}$ with $\mathbf{x}^{(\ell)} \in \mathbb{R}^d$ and $\mathbf{y}^{(\ell)} \in \mathbb{R}^e$.
- Define $\mathbf{S}_{xx} = \text{Cov}(\mathbf{x}) = \text{Var}(\mathbf{x})$, $\mathbf{S}_{yy} = \text{Cov}(\mathbf{y}) = \text{Var}(\mathbf{y})$, $\mathbf{S}_{xy} = \text{Cov}(\mathbf{x}, \mathbf{y})$, and $\mathbf{S}_{yx} = \text{Cov}(\mathbf{y}, \mathbf{x}) = \mathbf{S}_{xy}^T$.
- We want to find two projections $\mathbf{w}$ and $\mathbf{v}$ s.t. when $\mathbf{x}$ is projected along $\mathbf{w}$ (i.e., $a = \mathbf{w}^T\mathbf{x}$) and $\mathbf{y}$ is projected along $\mathbf{v}$ (i.e., $b = \mathbf{v}^T\mathbf{y}$), the correlation is maximized, i.e.,

$$\underset{\mathbf{w},\mathbf{v}}{\text{maximize}} \quad \rho_{ab} = \text{Corr}(\mathbf{w}^T\mathbf{x}, \mathbf{v}^T\mathbf{y})$$

with

$$\begin{aligned}\text{Corr}(\mathbf{w}^T\mathbf{x}, \mathbf{v}^T\mathbf{y}) &= \frac{\text{Cov}(\mathbf{w}^T\mathbf{x}, \mathbf{v}^T\mathbf{y})}{\sqrt{\text{Var}(\mathbf{w}^T\mathbf{x})}\sqrt{\text{Var}(\mathbf{v}^T\mathbf{y})}} \\ &= \frac{\mathbf{w}^T\text{Cov}(\mathbf{x}, \mathbf{y})\mathbf{v}}{\sqrt{\mathbf{w}^T\text{Var}(\mathbf{x})\mathbf{w}}\sqrt{\mathbf{v}^T\text{Var}(\mathbf{y})\mathbf{v}}} = \frac{\mathbf{w}^T\mathbf{S}_{xy}\mathbf{v}}{\sqrt{\mathbf{w}^T\mathbf{S}_{xx}\mathbf{w}}\sqrt{\mathbf{v}^T\mathbf{S}_{yy}\mathbf{v}}}\end{aligned}$$

Optimization

- The problem is equivalent to

$$\begin{aligned} \underset{\mathbf{w},\mathbf{v}}{\text{maximize}} \quad & \mathbf{w}^T \mathbf{S}_{xy} \mathbf{v} \\ \text{subject to} \quad & \mathbf{w}^T \mathbf{S}_{xx} \mathbf{w} = 1 \\ & \mathbf{v}^T \mathbf{S}_{yy} \mathbf{v} = 1 \end{aligned}$$

- The Lagrangian:

$$\mathcal{L}(\mathbf{w}, \mathbf{v}, \alpha, \beta) = -\mathbf{w}^T \mathbf{S}_{xy} \mathbf{v} + \alpha(\mathbf{w}^T \mathbf{S}_{xx} \mathbf{w} - 1) + \beta(\mathbf{v}^T \mathbf{S}_{yy} \mathbf{v} - 1)$$

- Taking the derivative of the Lagrangian w.r.t. $\mathbf{w}$ and $\mathbf{v}$, and setting it to $\mathbf{0}$, we get

$$\begin{aligned} \mathbf{S}_{xy}\mathbf{v} - 2\alpha \mathbf{S}_{xx}\mathbf{w} &= \mathbf{0} \\ \mathbf{S}_{yx}\mathbf{w} - 2\beta \mathbf{S}_{yy}\mathbf{v} &= \mathbf{0} \end{aligned} \implies \begin{aligned} \mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{w} &= 4\alpha\beta\mathbf{w} = \lambda\mathbf{w} \\ \mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{v} &= 4\alpha\beta\mathbf{v} = \lambda\mathbf{v} \end{aligned}$$

  indicating $\mathbf{w}$ is an eigenvector of $\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}$ corresponding to eigenvalue λ and similarly $\mathbf{v}$ is an eigenvector of $\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$ corresponding to eigenvalue λ.

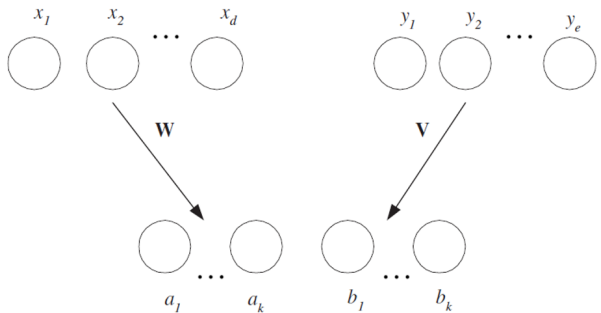
- To maximize ρ_{ab}, we choose the eigenvectors with the highest eigenvalue.

Canonical Correlation Analysis

- Like PCA, we can find $k \leq \min\{d, e\}$ vectors of $\mathbf{w}_i$ and $\mathbf{v}_i$ based on the PoV measure.
- We can obtain

$$\mathbf{a} = \mathbf{W}^T(\mathbf{x} - \mathbf{m}_x), \quad \mathbf{b} = \mathbf{V}^T(\mathbf{y} - \mathbf{m}_y)$$

which constitute the new, lower-dimensional representation with values of a_i uncorrelated and each a_i uncorrelated with all b_j, $j \neq i$.



Outline

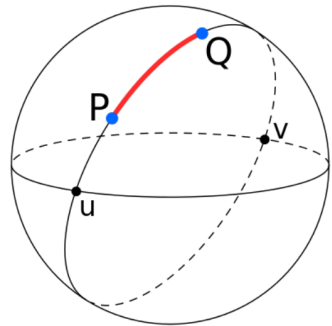
Introduction
Subset Selection
Principal Component Analysis
Factor Analysis
Multidimensional Scaling
Linear Discriminant Analysis
Canonical Correlation Analysis
Nonlinear Dimensionality Reduction
Kernel Dimensionality Reduction

Isometric Feature Mapping I

- PCA works when the data lies in a linear subspace.
- In many applications, the similarity between two features cannot be measured via the Euclidean distance.
- Isometric feature mapping (IsoMap) is MDS combined with a special metric, called geodesic distance , for reducing the dimensionality of data sampled from a smooth manifold.
- Instead of preserving the Euclidean distance, IsoMap preserves the geodesic distance.
- IsoMap is related to the manifold learning methods.

Isometric Feature Mapping II

- Given a sample $\mathcal{X}$, IsoMap uses the geodesic distances between all pairs of data points.
- The geodesic distance of two data points that live in a manifold is the shortest distance along the manifold.
- On a sphere, it is just the great-circle distance.
- In practice, where we are only given a sample $\mathcal{X}$ sampled from an unknown manifold, we can approximate the true geodesic distances by the shortest-path distances.

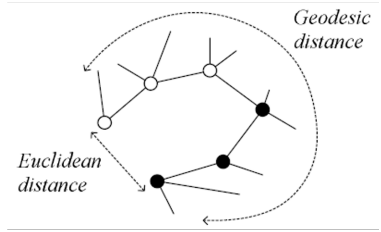


Isometric Feature Mapping III

- For neighboring points that are close in the input space, Euclidean distance can be used (i.e., geodesic distance is locally linear)

$$d_{rs} = \|\mathbf{x}^{(r)} - \mathbf{x}^{(s)}\|_2$$

  - ϵ-ball approach: for $\mathbf{x}^{(r)}$, $\mathbf{x}^{(s)}$ is close to $\mathbf{x}^{(r)}$ if $\|\mathbf{x}^{(r)} - \mathbf{x}^{(s)}\|_2 \leq \epsilon$, or
  - kNN approach: for $\mathbf{x}^{(r)}$, $\mathbf{x}^{(s)}$ is close to $\mathbf{x}^{(r)}$ if it is among the the k nearest neighbors of $\mathbf{x}^{(r)}$.



- For faraway points, geodesic distance is approximated by the sum of the distances between the points along the way over the manifold (shortest-path distance), say, via Dijkstra's algorithm.
- Points that are far apart in the manifold are also far apart in the new k-dim. space after MDS even if they are close in terms of Euclidean distance in the original d-dim. space.

Outline

Introduction

Subset Selection

Principal Component Analysis

Factor Analysis

Multidimensional Scaling

Linear Discriminant Analysis

Canonical Correlation Analysis

Nonlinear Dimensionality Reduction

Kernel Dimensionality Reduction

Kernel Methods for Dimensionality Reduction

- The kernel trick:
  - Choose a kernel $k(\cdot, \cdot)$.
  - Take any algorithm which can be computed purely using dot products $\mathbf{x}^{(\ell)T}\mathbf{x}^{(\ell')}$.
  - Replace each instance of $\mathbf{x}^{(\ell)T}\mathbf{x}^{(\ell')}$ with $k(\mathbf{x}^{(\ell)}, \mathbf{x}^{(\ell')})$.
- Since $k(\mathbf{x}^{(\ell)}, \mathbf{x}^{(\ell')}) = \boldsymbol{\phi}(\mathbf{x}^{(\ell)})^T \boldsymbol{\phi}(\mathbf{x}^{(\ell')})$, this procedure results in carrying out the original algorithm inside of $\mathbf{z} = \boldsymbol{\phi}(\mathbf{x})$ space.
- The result will be non-linear in the original data space.
- Similar idea to support vector machines.