DIMENSIONALITY REDUCTION TECHNIQUES

Shingchern D. You

Methods

- Principal components analysis (PCA)
- Factor analysis (FA)
- Independent components analysis (ICA)
- Linear discriminant analysis (LDA)

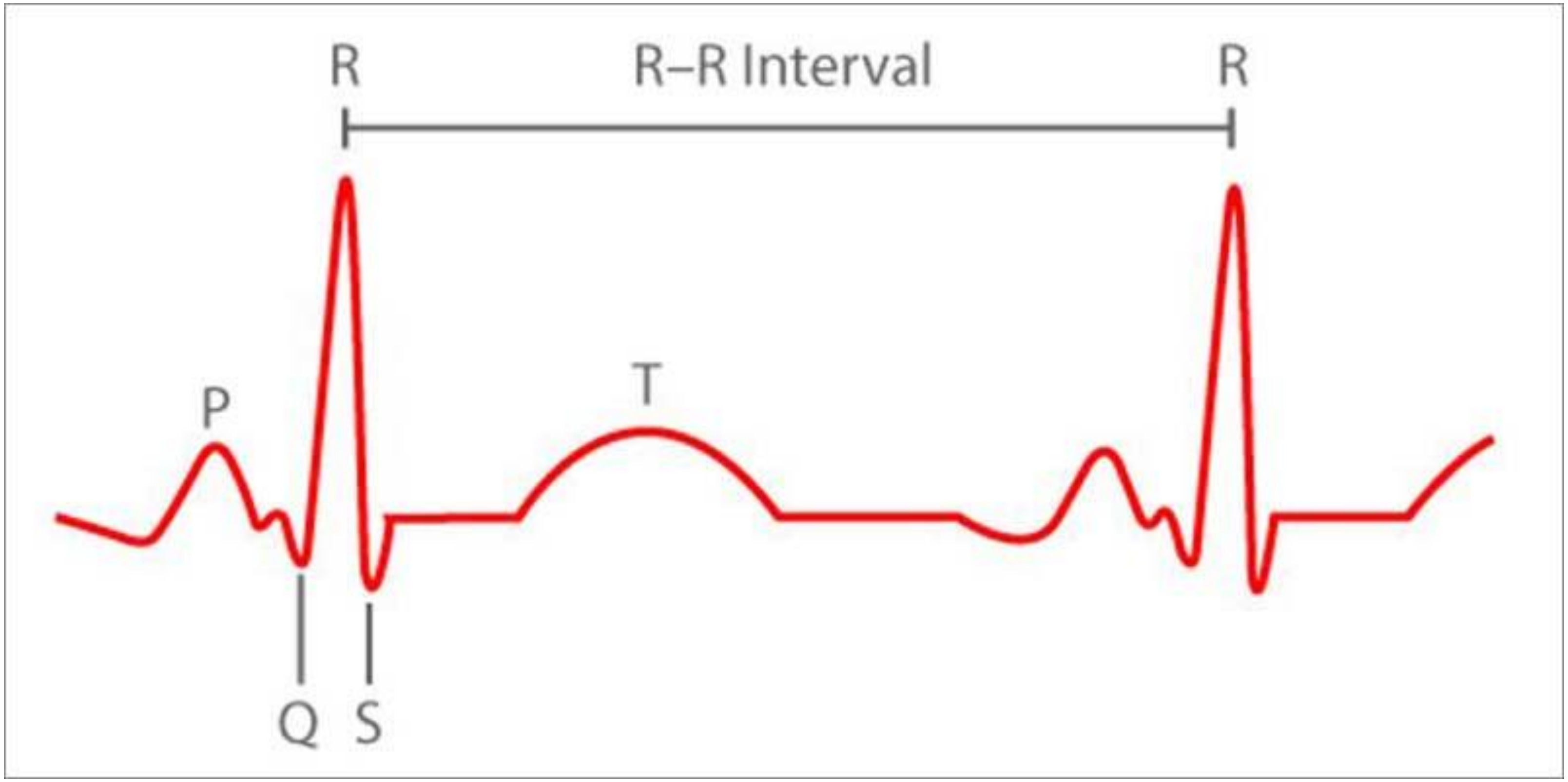
Motivation

- Why bother to reduce the dimensionality of the dataset
  - Higher classification speed (lower complexity)
  - Better visual inspection
  - Possibly better analysis
  - Lower influence due to noise or outliers

PCA Motivation

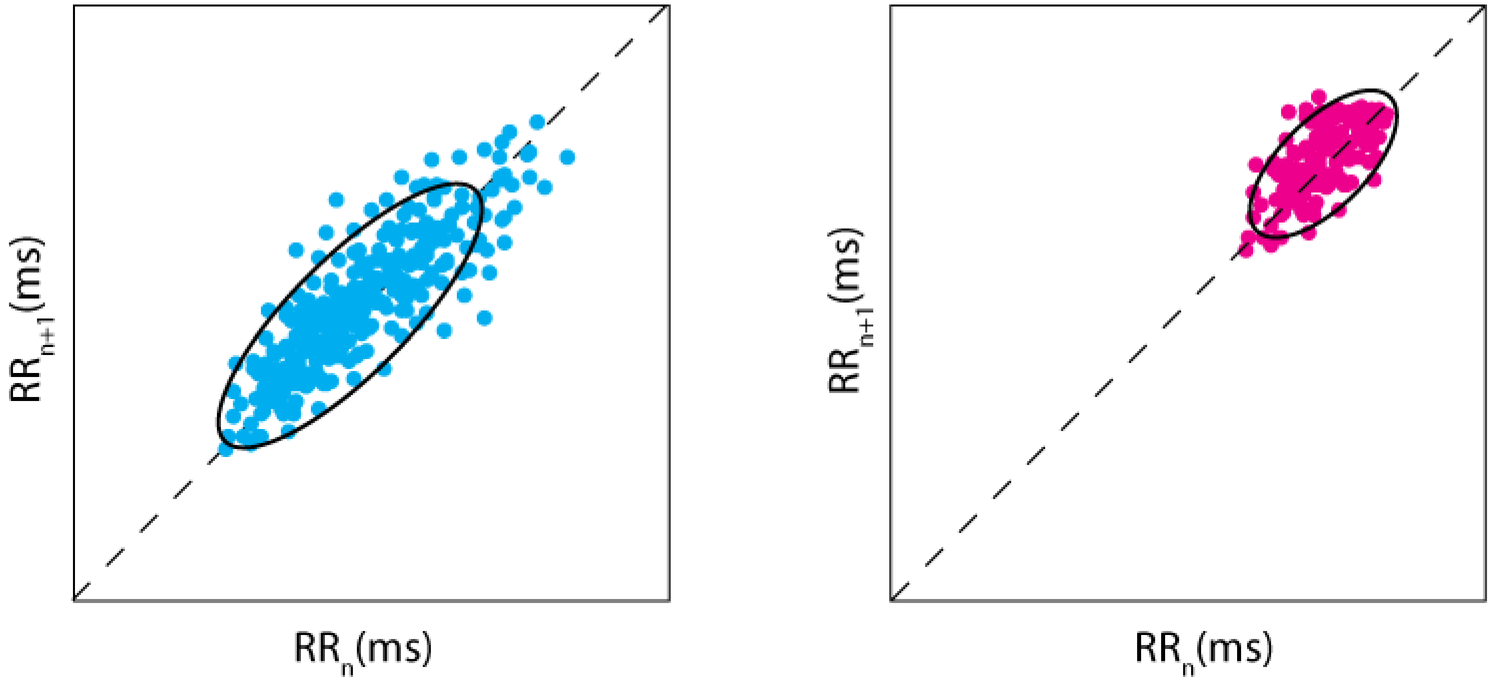
- Want to analyze heart problem via R-R interval

Source: https://simplifaster.com/articles/heart-rate-variability-training/



PCA Motivation

- Lorenz (Poincaré) plot of heartbeat R-R intervals
- Source: https://imotions.com/blog/heart-rate-variability/



PCA Motivation

- The data points are correlated
- If we want to compress the data, what would be a good approach to do it
- Encoding along the axes of ellipse

PCA Concept

- Source: https://en.wikipedia.org/wiki/Principal_component_analysis
- Let $\boldsymbol{x}_{(\boldsymbol{i})} \in R^p, i = 1 \ldots n$ be data points (zero mean)
- Let $\boldsymbol{w}_{(\boldsymbol{k})} \in R^p, k = 1 \ldots m$ be load vectors
- We want to find the first component

$$\boldsymbol{w}_{(\mathbf{1})} = \arg \max_{||w||=1} \sum_i (\boldsymbol{x}_{(\boldsymbol{i})} \cdot \boldsymbol{w})^2$$

where dot means inner product (projection)

- This optimization problem can be solved by methods of Lagrange multipliers or SVD

PCA Concept

- The idea is simple: We want to find an "axis" $\boldsymbol{w}_{(1)}$, when projected on it, the sum of $\boldsymbol{x}_{(i)}$ projections are maximum
- We can also do the second component by subtracting $\boldsymbol{x}_{(i)}$ from its projection
- We skip the math details here

PCA Concept

- The solution is $X^T X = W \Lambda W^T$

where Λ is a diagonal matrix (consists of eigenvalues)

$$X = \begin{bmatrix} \leftarrow & \boldsymbol{x}_{(\mathbf{1})} & \rightarrow \\ & \vdots & \\ \leftarrow & \boldsymbol{x}_{(\boldsymbol{n})} & \rightarrow \end{bmatrix} \text{ size (nxp)}$$

$$\mathrm{W} = \begin{bmatrix} \uparrow & & \uparrow \\ \boldsymbol{w}_{(\mathbf{1})} & \cdots & \boldsymbol{w}_{(\boldsymbol{p})} \\ \downarrow & & \downarrow \end{bmatrix} \text{ size (pxp)}$$

PCA Concept

- Some basic facts
- Covariance matrix is symmetric and positive semi-definite
- A symmetric matrix is **orthogonally** diagonalizable (also known as spectral decomposition)
- You may want to figure out how to prove above claims

PCA Concept

- It is recognized that $X^T X$ is proportional to sample covariance
- Fact: $X^T X$ is symmetric and semi-definite (Why?)
- Thus, it is diagonalizable and $\boldsymbol{w}_{(\boldsymbol{k})}$ is a normalized eigenvector of $X^T X$
- Note that $\boldsymbol{w}_{(\boldsymbol{j})}$ and $\boldsymbol{w}_{(\boldsymbol{k})}$ are orthonormal

PCA Concept

- If we arrange eigenvalues of $X^T X$ from large to small in $X^T X$ and corresponding eigenvalues in W, we can do dimensionality reduction
- Mathematically, we have (taking *k* components) dimensionality-reduced results in *T* with

$$T = \begin{bmatrix} \leftarrow & \boldsymbol{x}_{(1)} & \rightarrow \\ & \vdots & \\ \leftarrow & \boldsymbol{x}_{(n)} & \rightarrow \end{bmatrix} \begin{bmatrix} \uparrow & & \uparrow \\ \boldsymbol{w}_{(1)} & \cdots & \boldsymbol{w}_{(k)} \\ \downarrow & & \downarrow \end{bmatrix}$$

PCA Implementation

- Implementation reference: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf
- Check it out by yourself
- If you are interested in 2-D PCA steps, you can read my paper (Comparative Study of Methods for Reducing Dimensionality of MPEG-7 Audio Signature Descriptors)

PCA Implementation Tips

- Use the transformed results as new features for classification
- Compute $X^T X = W \Lambda W^T$ only with training set and keep the test dataset **untouched**
- Remember to perform transformation for test data before classification
- Whether to exclude mean or not before spectral decomposition is questionable, try both if you want

Factor Analysis

- Factor analysis is a statistical method used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors -- Wikepedia

Factor Analysis

- Basic formula (x_i, z_i and ε_i are **random variables**)

  $x_i - \mu_i = v_{i1}z_1 + v_{i2}z_2 + \ldots + v_{ik}z_k + \varepsilon_i$

  where z_j, $j = 1,\ldots,k$ are the latent factors with

  $E[\, z_j \,]=0$, $\mathrm{Var}(z_j)=1$, $\mathrm{Cov}(z_i, z_j)=0$, $i \neq j$,

  ε_i are the noise sources

  $E[\, \varepsilon_i \,]= 0$, $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$, $i \neq j$, $\mathrm{Cov}(\varepsilon_i, z_j) = \psi_i$,

  and v_{ij} are the factor loadings

Factor Analysis

- Let $\boldsymbol{x} = [x_1 \quad \cdots \quad x_p]^T$ and write the previous equation in matrix form
$$\boldsymbol{x} - \boldsymbol{\mu} = V\boldsymbol{z} + \boldsymbol{\varepsilon}$$
- Note in practical case, one outcome of $\boldsymbol{x}$ is one sample point (e.g., 4-D sample point in Iris case)
- It can be easily shown that
$$\mathrm{cov}[\boldsymbol{x}] = \Sigma = VV^T + \Psi$$
- Therefore, V is related to $\mathrm{cov}[\boldsymbol{x}]$

Factor Analysis

- The most widely used parameter estimation methods to find V based on observations is through principal components analysis
- However, other methods do exist, such as principal axis and Maximum likelihood methods (Ref: http://www.yorku.ca/ptryfos/f1400.pdf)
- A good source in Chinese is at https://ccjou.wordpress.com/2017/01/13/因素分析/

Factor Analysis

- If $\Psi = 0$, we have $\operatorname{cov}[\boldsymbol{x}] = \Sigma = VV^T$
- In practice, we use sample covariance S instead
- We know $S = W\Lambda W^T$ (diagonalizable, semi-positive)
- Thus, $VV^T = \left(W\Lambda^{\frac{1}{2}}\right)\left(\Lambda^{\frac{1}{2}}W^T\right) = \left(W\Lambda^{\frac{1}{2}}\right)\left(W\Lambda^{\frac{1}{2}}\right)^T$
- Therefore, $V = (W\Lambda^{\frac{1}{2}})$

Factor Analysis

- In practical situation, $\Psi \neq 0$, we can use other methods to estimate Ψ first, then decompose $(S - \Psi)$
- A simple method to find Ψ is by linear interpolation
- By considering Ψ, we need to solve Ψ and V iteratively

Factor Analysis

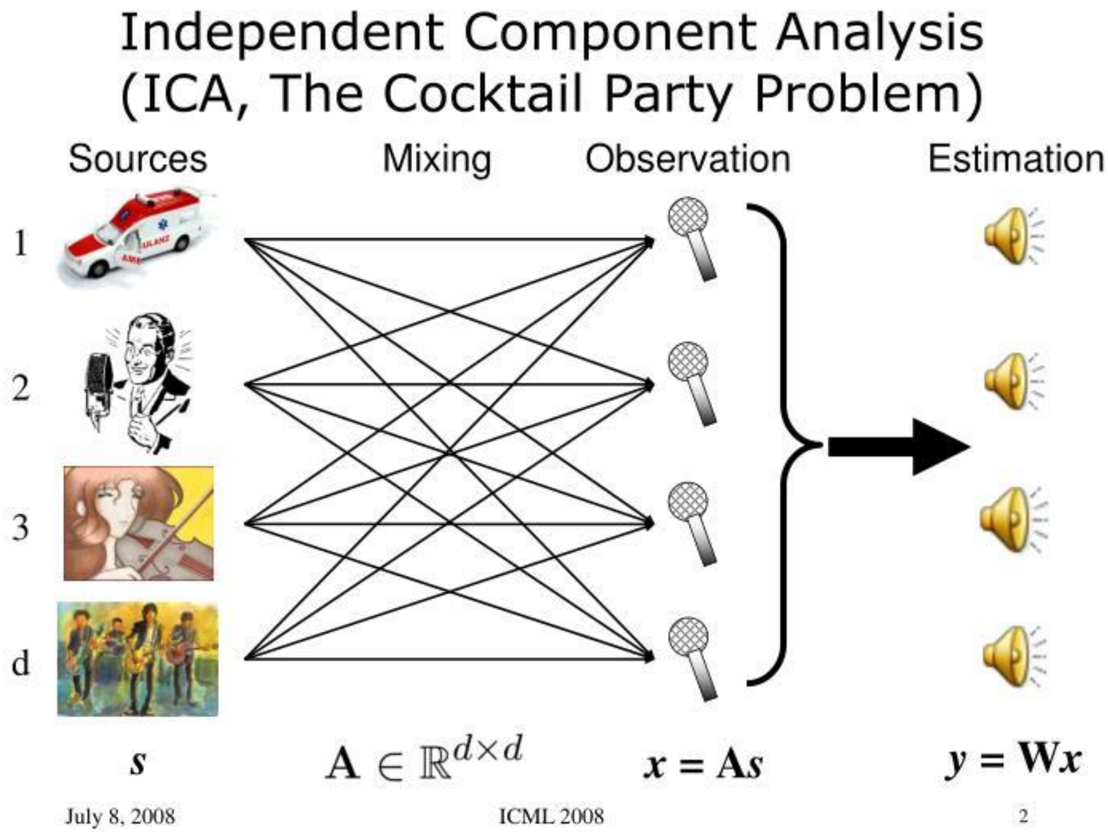
- Alternatively, we can use the following (source: Factor Analysis Based Anomaly Detection , Proceedings of the 2003 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY June 2003)
- $\tilde{V} = (V^T V)^{-1} V^T$ and $f = \tilde{V} x$ (LS solution)
- If we use only large eigenvalues and corresponding large eigenvectors, we can get the dimensionality-reduced $\boldsymbol{f}_{(\boldsymbol{k})}$ from $\boldsymbol{x}_{(\boldsymbol{k})}$ (observation, not RV)
- There is no unique solution to V

Factor Analysis

- We may also multiply the factors with a rotation matrix and the basic requirements (given previously) still hold
- Two different rotation methods
- Details omitted (cf. textbook or reference materials)

Independent Components Analysis

- (Blind Source Separation) problem source: https://www.slideserve.com/vladimir-kirkland/ica-and-isa-using-schweizer-wolff-measure-of-dependence



Independent Components Analysis

- Important assumptions (source: https://en.wikipedia.org/wiki/Independent_component_analysis)
  - The source signals are **independent** of each other
  - The values in each source signal have **non-Gaussian** distributions (at most one Gaussian source)
  - Number of observations must be **at least equal to** number of sources
- Assuming independent sources is easy to understand
- But, why non-Gaussian?

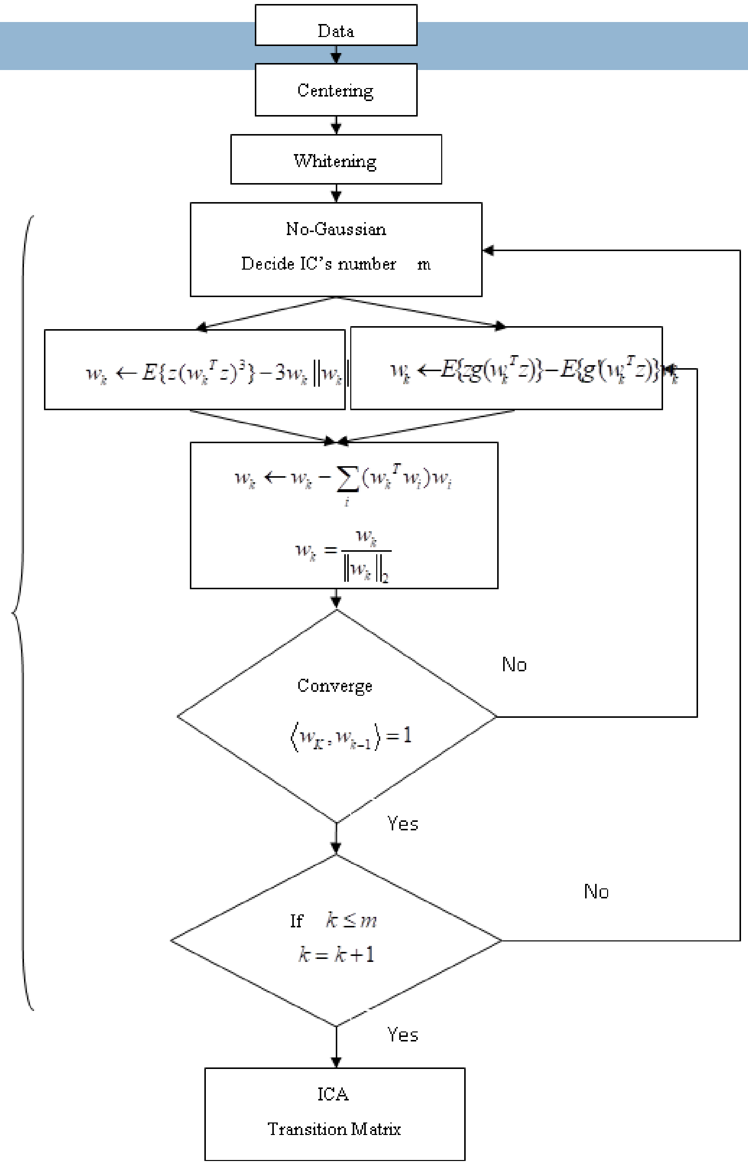
Independent Components Analysis

- There are several methods to implement ICA
- Because the math is lengthy, I will only briefly explain how to do it
- Three main steps of ICA
  - Centering (to make it zero mean)
  - Whitening (like PCA, to remove correlation)
  - Max non-Gaussian (Kurtosis, Negentropy, etc.)

Independent Components Analysis

- Flowchart

(source:洪名人 MS thesis)



Independent Components Analysis

- To use ICA for dimensionality reduction, a simple method is to use components with larger energy as features
- Previously, we tried Kurtosis and Negentropy in dimensionality reduction and found that Kurtosis was much better than Negentropy for MPEG-7 audio signature descriptors (洪名人 MS thesis)

Linear Discriminant Analysis

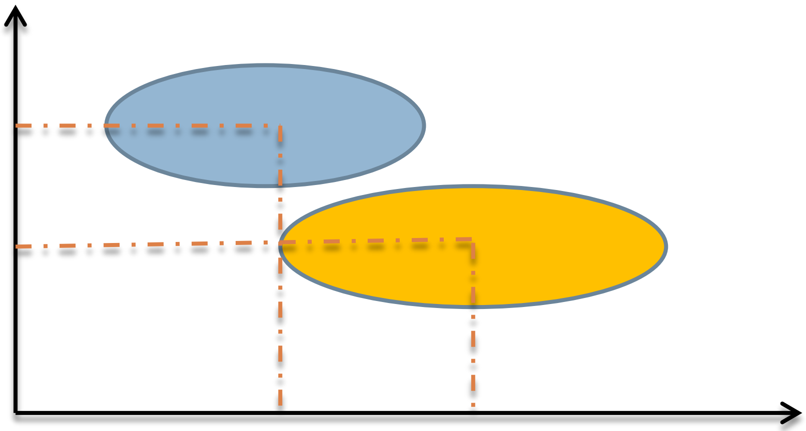
- PCA, FA, and LDA are "unsupervised" meaning that we do not need to know the classification results to apply these approaches
- LDA is a supervised approach
- Separately compute some parameters from each class of data

Linear Discriminant Analysis

- Reference: http://courses.cs.tamu.edu/rgutier/cs790_w02/l6.pdf
- Let $\boldsymbol{x}_{(\boldsymbol{i})} \in R^p, i = 1 \ldots n$ be data points (nonzero mean) in two classes
- Want to project data samples $\boldsymbol{x}_{(\boldsymbol{i})}$ onto a line $\boldsymbol{w}$ (or a vector) such that both classes can be separated as far as possible
- However, we also need to consider the "scattering" of classes

Linear Discriminant Analysis

- Example: Which line is better (X or Y axis)?
- X axis has higher distance, but Y is actually better



Linear Discriminant Analysis

- Therefore, we want to maximize the distance of means but keeping scatters as small as possible

$$J(\boldsymbol{w}) = \frac{(m_1 - m_2)^2}{s_1 + s_2} \text{ with } \|\boldsymbol{w}\| = 1$$

where $m_j = \frac{1}{n_j}\sum_{i \in C_j} \boldsymbol{x}_{(\boldsymbol{i})} \cdot \boldsymbol{w}$ and

$s_j = \sum_{i \in C_j}(\boldsymbol{x}_{(\boldsymbol{i})} \cdot \boldsymbol{w} - m_j)^2$ (within-class scatter)

C_j is the set of indices of samples belonging to class j

Dot again represents inner product

Linear Discriminant Analysis

- The solution is $\boldsymbol{w}_{opt} = S_w^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2)$

where $\boldsymbol{m}_j = \frac{1}{n_j}\sum_{i \in C_j} \boldsymbol{x}_{(\boldsymbol{i})}$ (n_j: # of samples in class j)

and $S_w = S_1 + S_2$

$$S_j = \sum_{i \in C_j} (\boldsymbol{x}_{(\boldsymbol{i})} - \boldsymbol{m}_j)(\boldsymbol{x}_{(\boldsymbol{i})} - \boldsymbol{m}_j)^T$$

- You can recognize $\boldsymbol{m}_j$ is in-class sample mean and S_j is similar to in-class sample covariance

Linear Discriminant Analysis

- We can extend the 2-class case into *k* classes
- Want to project data into at most (*k*-1) dimensional space (with (*k*-1) vectors as basis)
- Recall: project 2 classes to 1-D, so 3 classes to 2-D, and so on
- The goal is to maximize between-class scatter while minimize the in-class scatters

Linear Discriminant Analysis

- In short, we want to find $\arg\max_{W} J(W) = \frac{\det(\tilde{S}_B)}{\det(\tilde{S}_W)}$

where $\tilde{S}_B = W^T S_B W$ and $\tilde{S}_W = W^T S_W W$ and where

$S_W = \sum S_i$ (same as 2-class case)

and

$$S_B = \sum_{i \in C_j} n_j (\boldsymbol{m}_{(\boldsymbol{i})} - \boldsymbol{m})(\boldsymbol{m}_{(\boldsymbol{i})} - \boldsymbol{m})^T$$

where $\boldsymbol{m} = \frac{1}{n} \sum_i \boldsymbol{x}_{(\boldsymbol{i})}$ (mean for data in all classes)

Linear Discriminant Analysis

- The solution

$$W = \begin{bmatrix} \uparrow & & \uparrow \\ \boldsymbol{w}_{(\mathbf{1})} & \cdots & \boldsymbol{w}_{(\boldsymbol{k}-\mathbf{1})} \\ \downarrow & & \downarrow \end{bmatrix}$$ is composed of the eigenvectors of the matrix $S_W^{-1} S_B$

- To use LDA for dimensionality reduction, we use eigenvectors associated with largest eigenvalues of $S_W^{-1} S_B$

Linear Discriminant Analysis

- Then, use $\boldsymbol{z}_{(\boldsymbol{i})} = W\boldsymbol{x}_{(\boldsymbol{i})}$ as features
- Recall that we have at most (k-1) independent vector in W, LDA is not good for dimensionality reduction for 2-class problems (mapping to a line as we mentioned before)