DIMENSIONALITY REDUCTION

Dimensionality of input

- Number of Observables (e.g. age and income)
- If number of observables is increased
  - More time to compute
  - More memory to store inputs and intermediate results
  - More complicated explanations (knowledge from learning)
    - Regression from 100 vs. 2 parameters
  - No simple visualization
    - 2D vs. 10D graph
  - **Need much more data (curse of dimensionality)**
    - 1M of 1-d inputs is not equal to 1 input of dimension 1M

Dimensionality reduction

- Some features (dimensions) bear little or nor useful information (e.g. color of hair for a car selection)
  - Can drop some features
  - Have to estimate which features can be dropped from data

- Several features can be combined together without loss or even with gain of information (e.g. income of all family members for loan application)
  - Some features can be combined together
  - Have to estimate which features to combine from data

Feature Selection vs Extraction

- Feature selection: Choosing $k<d$ important features, ignoring the remaining $d - k$
  - Subset selection algorithms
- Feature extraction: Project the original x_i , i =1,...,d dimensions to new $k<d$ dimensions, z_j , j =1,...,k
  - Principal Components Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Factor Analysis (FA)

Usage

- Have data of dimension d
- Reduce dimensionality to k<d
  - Discard unimportant features
  - Combine several features in one
- Use resulting k-dimensional data set for
  - Learning for classification problem (e.g. parameters of probabilities P(x|C)
  - Learning for regression problem (e.g. parameters for model y=g(x|Thetha)

Subset selection

- Have initial set of features of size d
- There are 2^d possible subsets
- Need a criteria to decide which subset is the best
- A way to search over the possible subsets
- Can’t go over all 2^d possibilities
- Need some heuristics

“Goodness” of feature set

- Supervised
  - Train using selected subset
  - Estimate error on validation data set
- Unsupervised
  - Look at input only(e.g. age, income and savings)
  - Select subset of 2 that bear most of the information about the person

Mutual Information

- Have a 3 random variables(features) X,Y,Z and have to select 2 which gives most information
- If X and Y are "correlated" then much of the information about of Y is already in X
- Make sense to select features which are "uncorrelated"
- Mutual Information (Kullback–Leibler Divergence) is more general measure of "mutual information"
- Can be extended to n variables (information variables x_1,.. x_n have about variable x_{n+1})

Subset-selection

- Forward search
  - Start from empty set of features
  - Try each of remaining features
  - Estimate classification/regression error for adding specific feature
  - Select feature that gives maximum improvement in validation error
  - Stop when no significant improvement

- Backward search
  - Start with original set of size d
  - Drop features with smallest impact on error

Floating Search

- Forward and backward search are "greedy" algorithms
  - Select best options at single step
  - Do not always achieve optimum value

- Floating search
  - Two types of steps: Add *k*, remove *l*
  - *More computations*

Feature Extraction

- Face recognition problem
  - Training data input: pairs of Image + Label(name)
  - Classifier input: Image
  - Classifier output: Label(Name)
- Image: Matrix of 256X256=65536 values in range 0..256
- Each pixels bear little information so can’t select 100 best ones
- Average of pixels around specific positions may give an indication about an eye color.

Projection

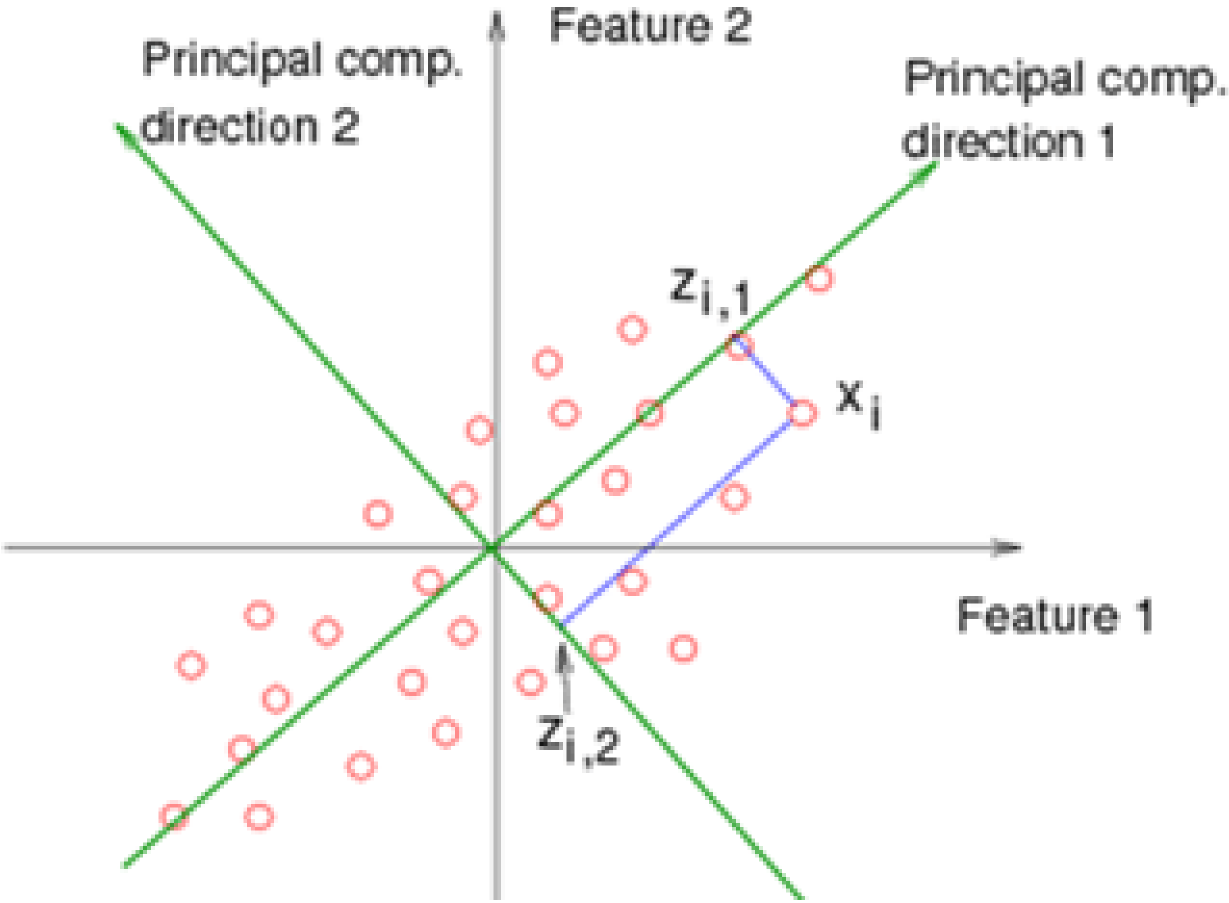
- Find a projection matrix w from d-dimensional to k-dimensional vectors that keeps error low

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

PCA: Motivation

- Assume that d observables are linear combination of k<d vectors
- $z_i = w_{i1} x_{i1} + \ldots + w_{ik} x_{id}$
- We would like to work with basis as it has lesser dimension and have all(almost) required information
- What we expect from such basis
  - Uncorrelated or otherwise can be reduced further
  - Have large variance (e.g. w_{i1} have large variation) or otherwise bear no information

PCA: Motivation



Based on E Alpaydın 2004 Introduction to Machine Learning © The MIT Press (V1.1)

PCA: Motivation

- Choose directions such that a total variance of data will be maximum
  - Maximize Total Variance
- Choose directions that are orthogonal
  - Minimize correlation
- Choose k<d orthogonal directions which maximize total variance

PCA

- Choosing only directions: $\|\boldsymbol{w}_1\| = 1$
- $z_1 = \boldsymbol{w}_1^T \boldsymbol{x}$ $\mathrm{Cov}(\boldsymbol{x}) = \Sigma$, $\mathrm{Var}(z_1) = \boldsymbol{w}_1^T \Sigma \boldsymbol{w}_1$
- Maximize variance subject to a constrain using Lagrange Multipliers

$$\max_{\boldsymbol{w}_1} \boldsymbol{w}_1^T \Sigma \boldsymbol{w}_1 - \alpha(\boldsymbol{w}_1^T \boldsymbol{w}_1 - 1)$$

- Taking Derivatives

$$2\Sigma \boldsymbol{w}_1 - 2\alpha \boldsymbol{w}_1 = 0 \qquad \Sigma \boldsymbol{w}_1 = \alpha \boldsymbol{w}_1$$

- Eigenvector. Since want to maximize $w_1^T \Sigma w_1 = \alpha w_1^T w_1 = \alpha$ we should choose an eigenvector with largest eigenvalue

PCA

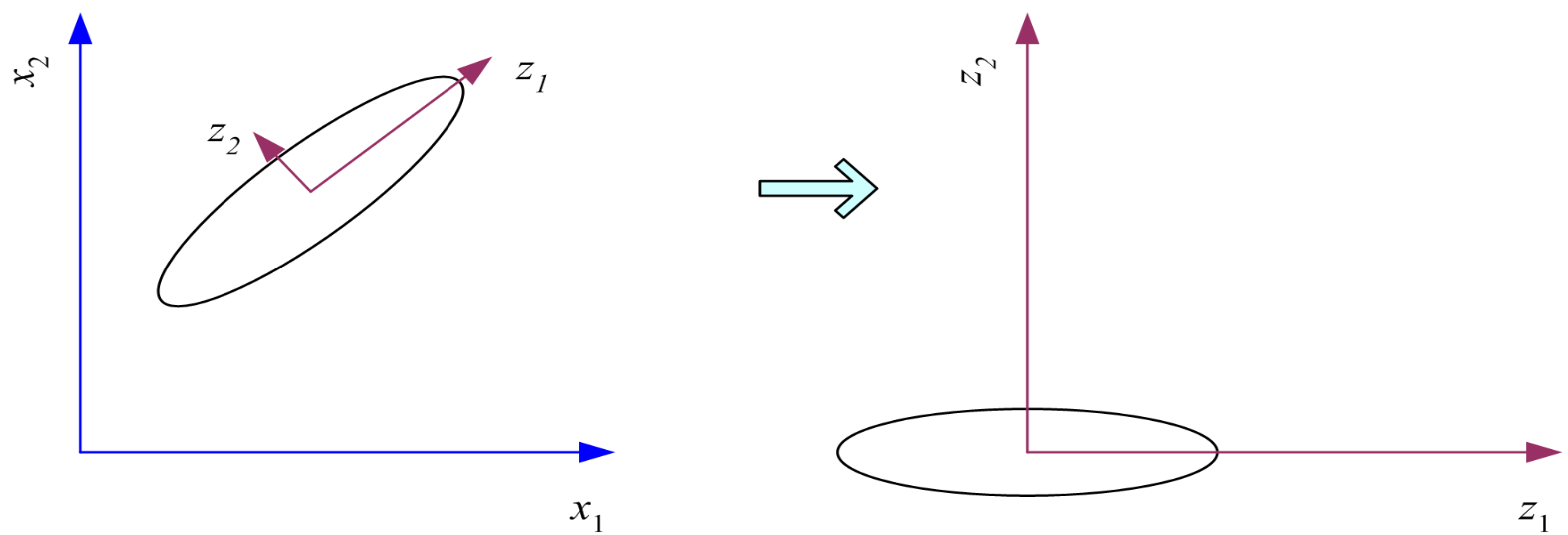
- d-dimensional feature space
- d by d symmetric covariance matrix estimated from samples $\mathrm{Cov}(\boldsymbol{x}) = \Sigma,$
- Select k largest eigenvalue of the covariance matrix and associated k eigenvectors
- The first eigenvector will be a direction with largest variance

What PCA does

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

where the columns of **W** are the eigenvectors of $\sum$, and $\boldsymbol{m}$ is sample mean

Centers the data at the origin and rotates the axes

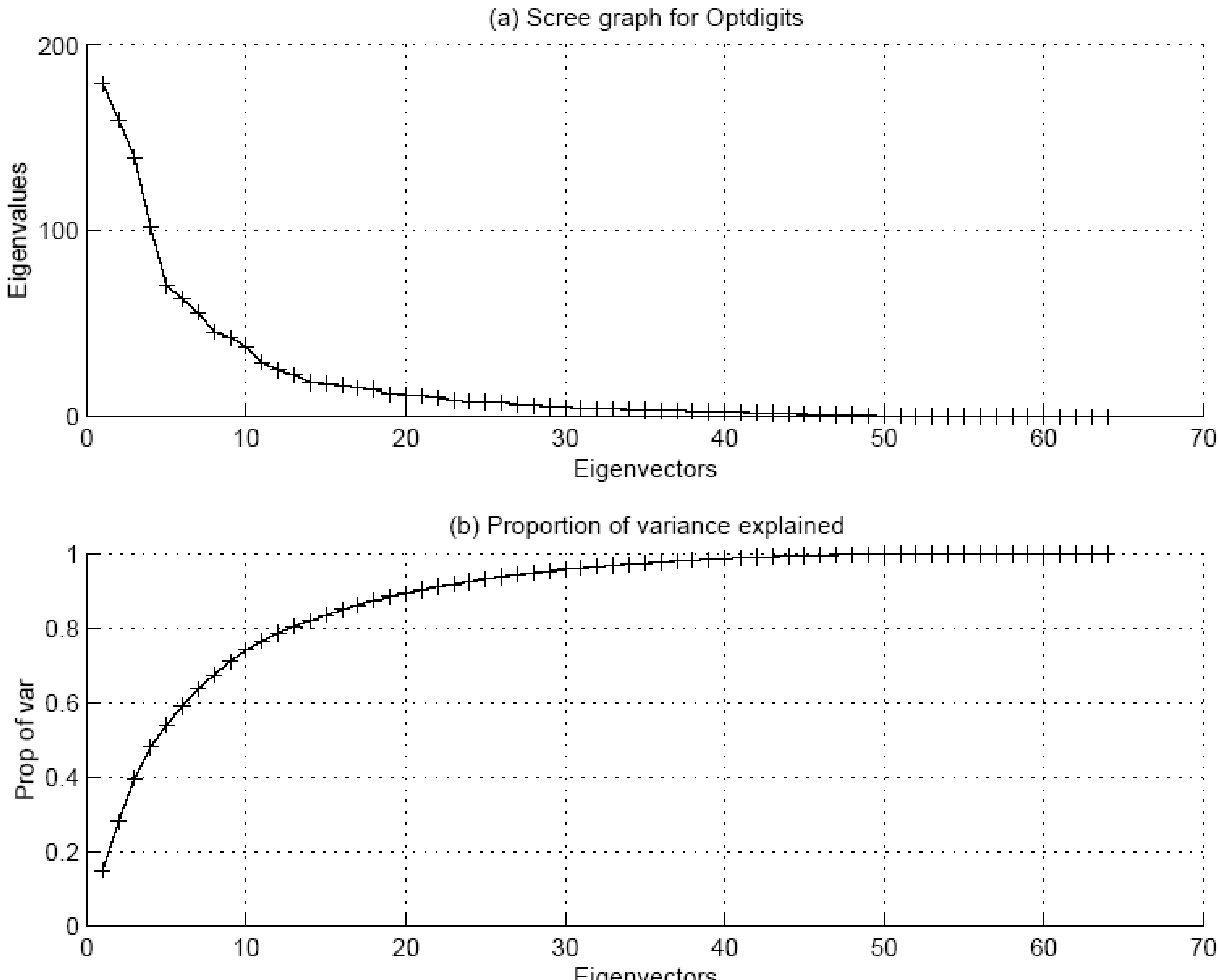


How to choose k ?

- Proportion of Variance (PoV) explained

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

  when λ_i are sorted in descending order
- Typically, stop at PoV>0.9
- Scree graph plots of PoV vs *k*, stop at "elbow"



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

PCA

- PCA is unsupervised (does not take into account class information)
- Can take into account classes : Karhuned-Loeve Expansion
  - Estimate Covariance Per Class
  - Take average weighted by prior
- Common Principle Components
  - Assume all classes have same eigenvectors (directions) but different variances

PCA

- Does not try to explain noise
  - Large noise can become new dimension/largest PC
- Interested in resulting uncorrelated variables which explain large portion of **total** sample variance
- Sometimes interested in explained shared variance (common factors) that affect data

Factor Analysis

- Assume set of unobservable ("latent") variables
- Goal: Characterize dependency among observables using latent variables
- Suppose group of variables having large correlation among themselves and small correlation with other variables
- Single factor?

Factor Analysis

- Assume k input factors (latent unobservable) variables generating d observables

- Assume all variations in observable variables are due to latent or noise (with unknown variance)

- Find transformation from unobservable to observables which explain the data

Factor Analysis

- Find a small number of factors **z**, which when combined generate **x** :

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

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

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

and v_{ij} are the factor loadings

$$\mathbf{x} - \boldsymbol{\mu} = \mathbf{V}\mathbf{z} + \boldsymbol{\epsilon}$$

Factor Analysis

- Find V such that $\mathbf{S} = \mathbf{V}\mathbf{V}^T + \mathbf{\Psi}$ where S is estimation of covariance matrix and V loading (explanation by latent variables)

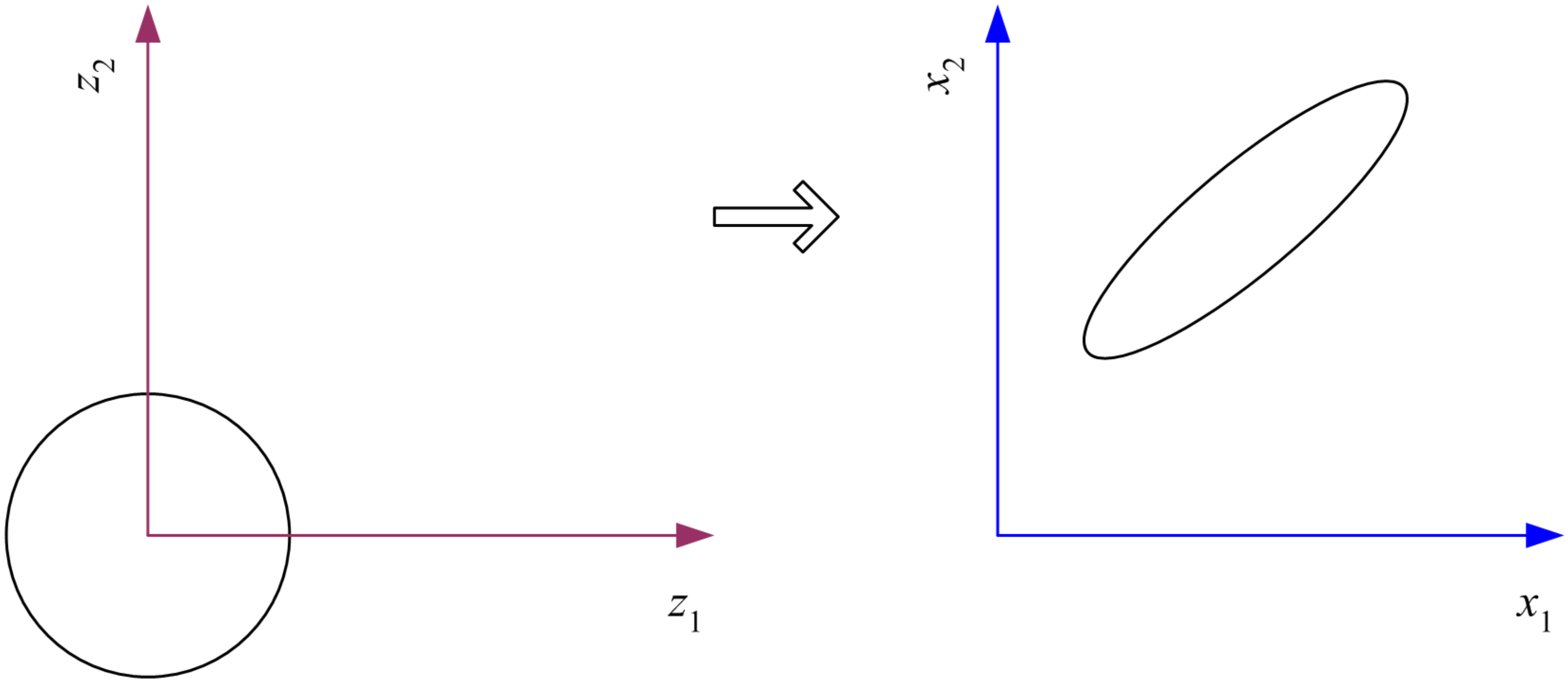
- V is d x k matrix (k<d)

- Solution using eigenvalue and eigenvectors

$$\mathbf{Z} = \mathbf{X}\mathbf{W} = \mathbf{X}\mathbf{S}^{-1}\mathbf{V}$$

Factor Analysis

- In FA, factors z_i are stretched, rotated and translated to generate **x**

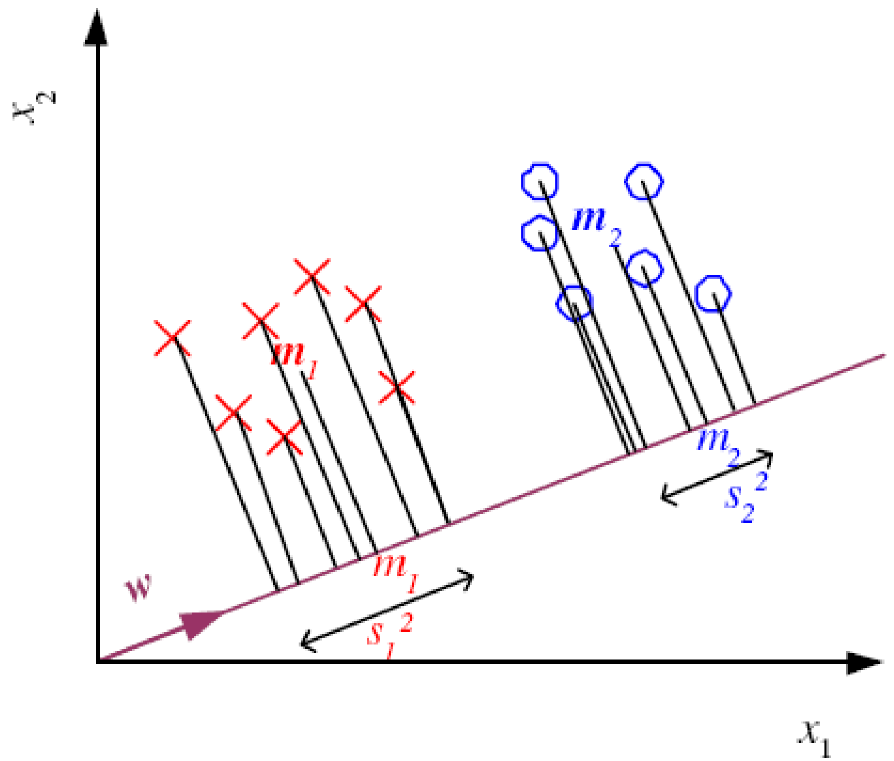


FA Usage

- Speech is a function of position of small number of articulators (lungs, lips, tongue)
- Factor analysis: go from signal space (4000 points for 500ms) to articulation space (20 points)
- Classify speech (assign text label) by 20 points
- Speech Compression: send 20 values

Linear Discriminant Analysis

- Find a low-dimensional space such that when **x** is projected, classes are well-separated



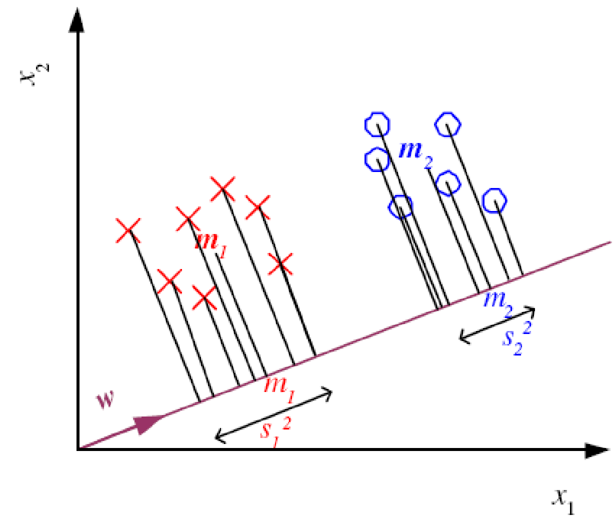
Means and Scatter after projection

$$
\begin{aligned}
m_1 &= \frac{\sum_t \mathbf{w}^T \mathbf{x}^t r^t}{\sum_t r^t} = \mathbf{w}^T \boldsymbol{m}_1 \\
m_2 &= \frac{\sum_t \mathbf{w}^T \mathbf{x}^t (1 - r^t)}{\sum_t (1 - r^t)} = \mathbf{w}^T \boldsymbol{m}_2
\end{aligned}
$$

$$
\begin{aligned}
s_1^2 &= \sum_t (\mathbf{w}^T \mathbf{x}^t - m_1)^2 r^t \\
s_2^2 &= \sum_t (\mathbf{w}^T \mathbf{x}^t - m_2)^2 (1 - r^t)
\end{aligned}
$$

Good Projection

- Means are far away as possible
- Scatter is small as possible
- Fisher Linear Discriminant



$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

Summary

- Feature selection
  - Supervised: drop features which don’t introduce large errors (validation set)
  - Unsupervised: keep only uncorrelated features (drop features that don’t add much information)
- Feature extraction
  - Linearly combine feature into smaller set of features
  - Supervised
    - PCA: explain most of the total variability
    - FA: explain most of the common variability
  - Unsupervised
    - LDA: best separate class instances