

DIMENSIONALITY REDUCTION



Dimensionality of input

2

- 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

3

- 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

4

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

Usage

5

- 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 $y = g(x | \theta)$)

Subset selection

6

- 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

7

- 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

8

- 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 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, \dots, x_n have about variable x_{n+1})

Subset-selection

9

- 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

10

- 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

11

- Face recognition problem
 - ▣ Training data input: pairs of Image + Label(name)
 - ▣ Classifier input: Image
 - ▣ Classifier output: Label(Name)
- Image: Matrix of $256 \times 256 = 65536$ 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

12

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

$$Z = W^T X$$

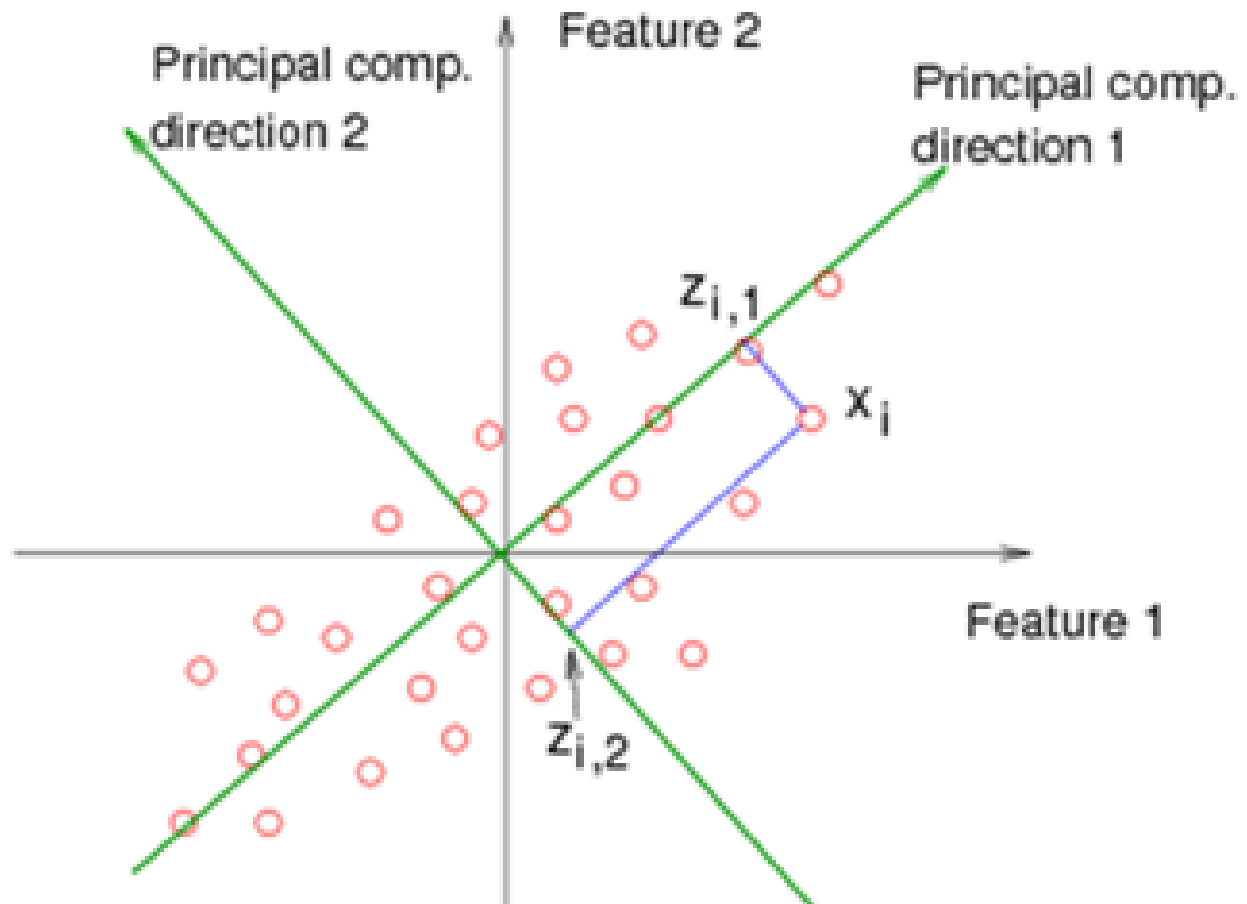
PCA: Motivation

13

- Assume that d observables are linear combination of $k < d$ vectors
- $z_i = w_{i1}x_{i1} + \dots + w_{ik}x_{id}$
- 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

14



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

PCA: Motivation

15

- 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

16

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

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

- Taking Derivatives

$$2\Sigma \mathbf{w}_1 - 2\alpha \mathbf{w}_1 = 0 \quad \Sigma \mathbf{w}_1 = \alpha \mathbf{w}_1$$

- Eigenvector. Since want to maximize $\mathbf{w}_1^T \Sigma \mathbf{w}_1 = \alpha \mathbf{w}_1^T \mathbf{w}_1 = \alpha$ we should choose an eigenvector with largest eigenvalue

PCA

17

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

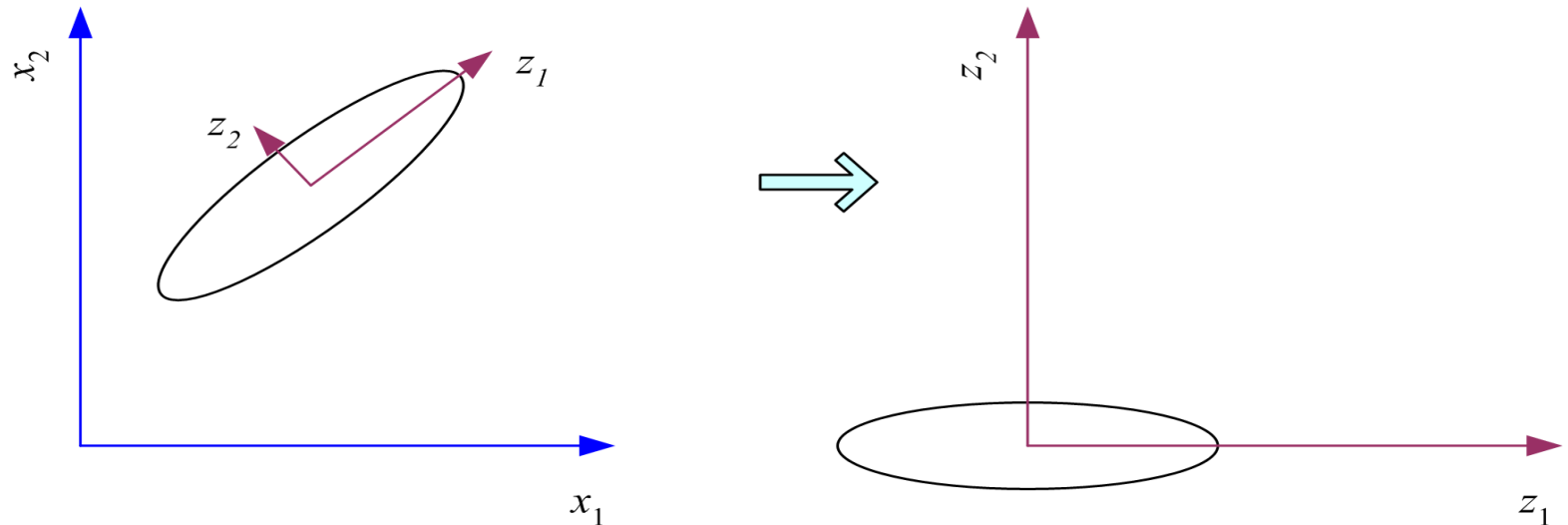
What PCA does

18

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

where the columns of \mathbf{W} are the eigenvectors of Σ ,
and \mathbf{m} is sample mean

Centers the data at the origin and rotates the axes



How to choose k ?

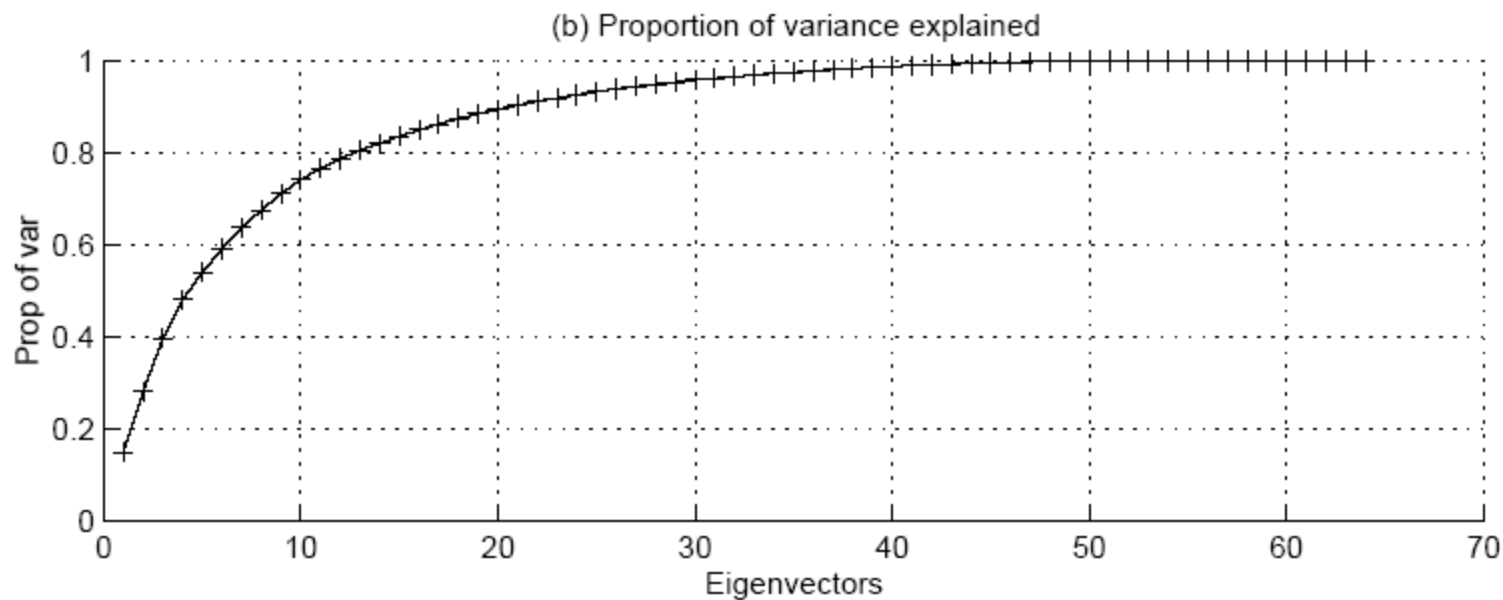
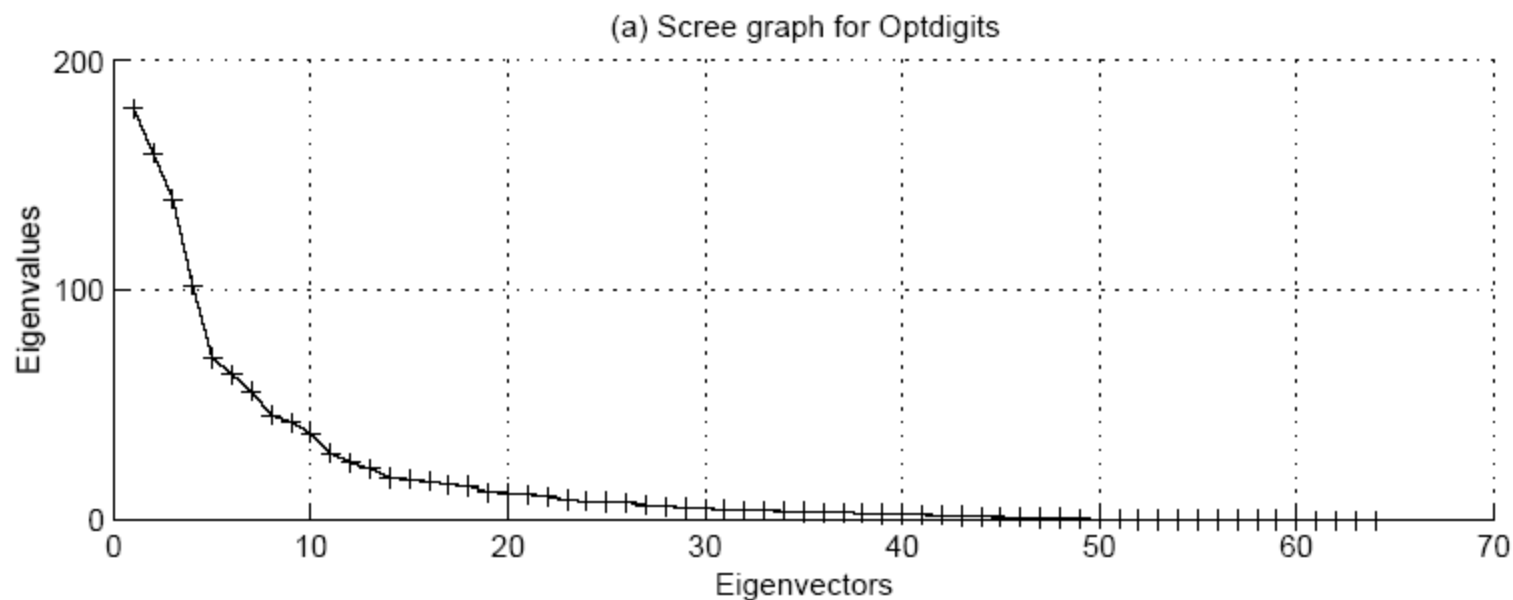
19

- 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 $\text{PoV} > 0.9$
- Scree graph plots of PoV vs k , stop at “elbow”



PCA

21

- PCA is unsupervised (does not take into account class information)
- Can take into account classes : Karhunen-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

22

- 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

23

- 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

24

- 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

25

- Find a small number of factors \mathbf{z} , which when combined generate \mathbf{x} :

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

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

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

ε_i are the noise sources

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

and v_{ij} are the factor loadings

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

Factor Analysis

26

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

- V is $d \times k$ matrix ($k < d$)

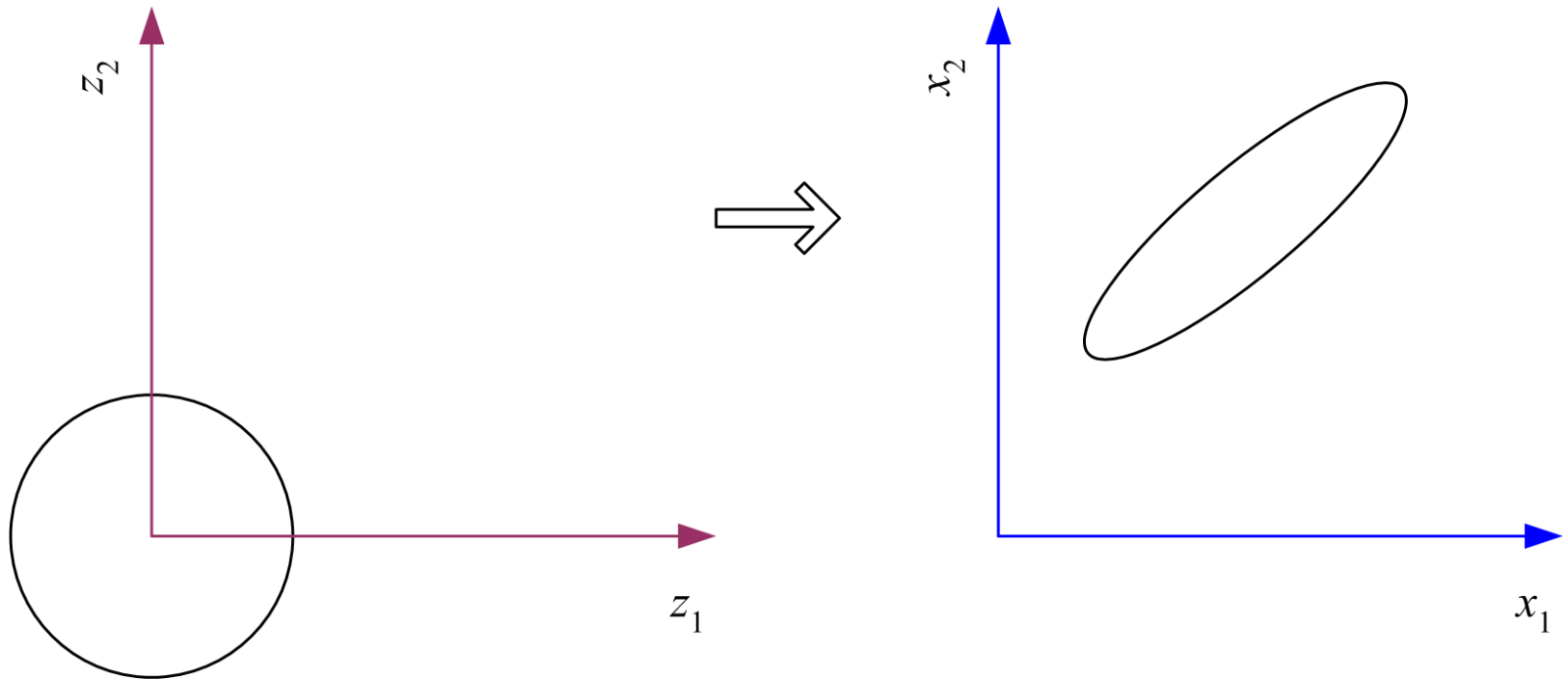
- Solution using eigenvalue and eigenvectors

$$Z = XW = XS^{-1}V$$

Factor Analysis

27

- In FA, factors z_i are stretched, rotated and translated to generate \mathbf{x}



FA Usage

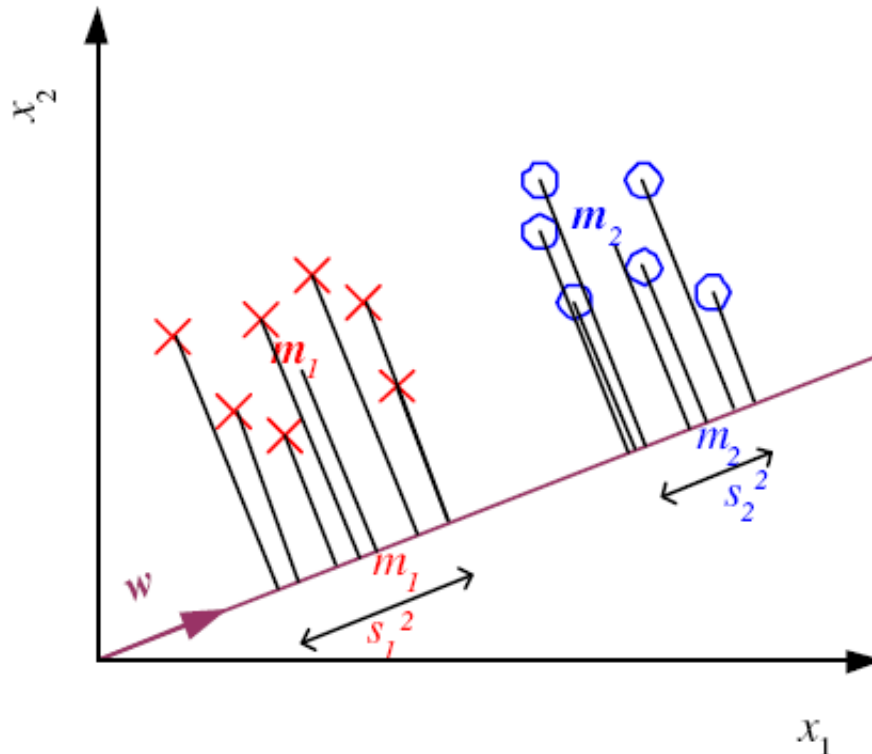
28

- 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

29

- Find a low-dimensional space such that when \mathbf{x} is projected, classes are well-separated



Means and Scatter after projection

30

$$m_1 = \frac{\sum_t \mathbf{w}^T \mathbf{x}^t r^t}{\sum_t r^t} = \mathbf{w}^T \mathbf{m}_1$$

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

$$s_1^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_1)^2 r^t$$

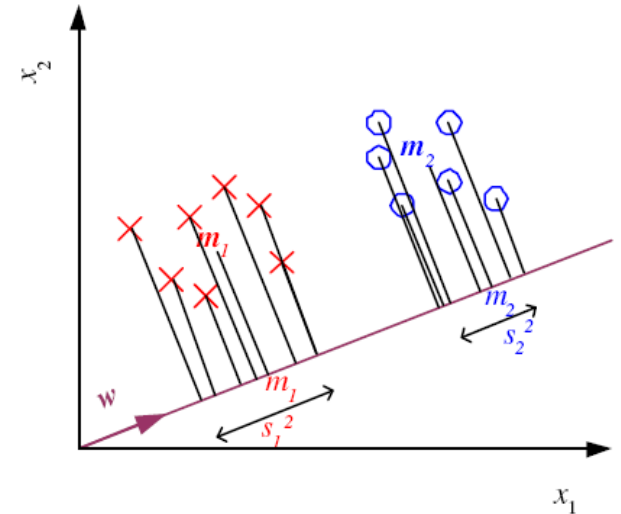
$$s_2^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_2)^2 (1 - r^t)$$

Good Projection

31

- 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

32

- 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