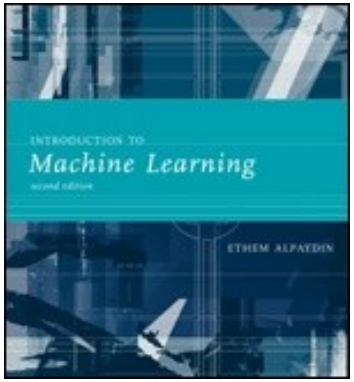
Lecture Slides for

INTRODUCTION TO

Machine Learning

2nd Edition



ETHEM ALPAYDIN, modified by Leonardo Bobadilla and some parts from http://www.cs.tau.ac.il/~apartzin/MachineLearning/

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2m

Outline

Previous class

Multivariate Data

Parameter Estimation

Estimation of Missing Values

Multivariate Classification

.

This class: Ch 5: Multivariate Methods

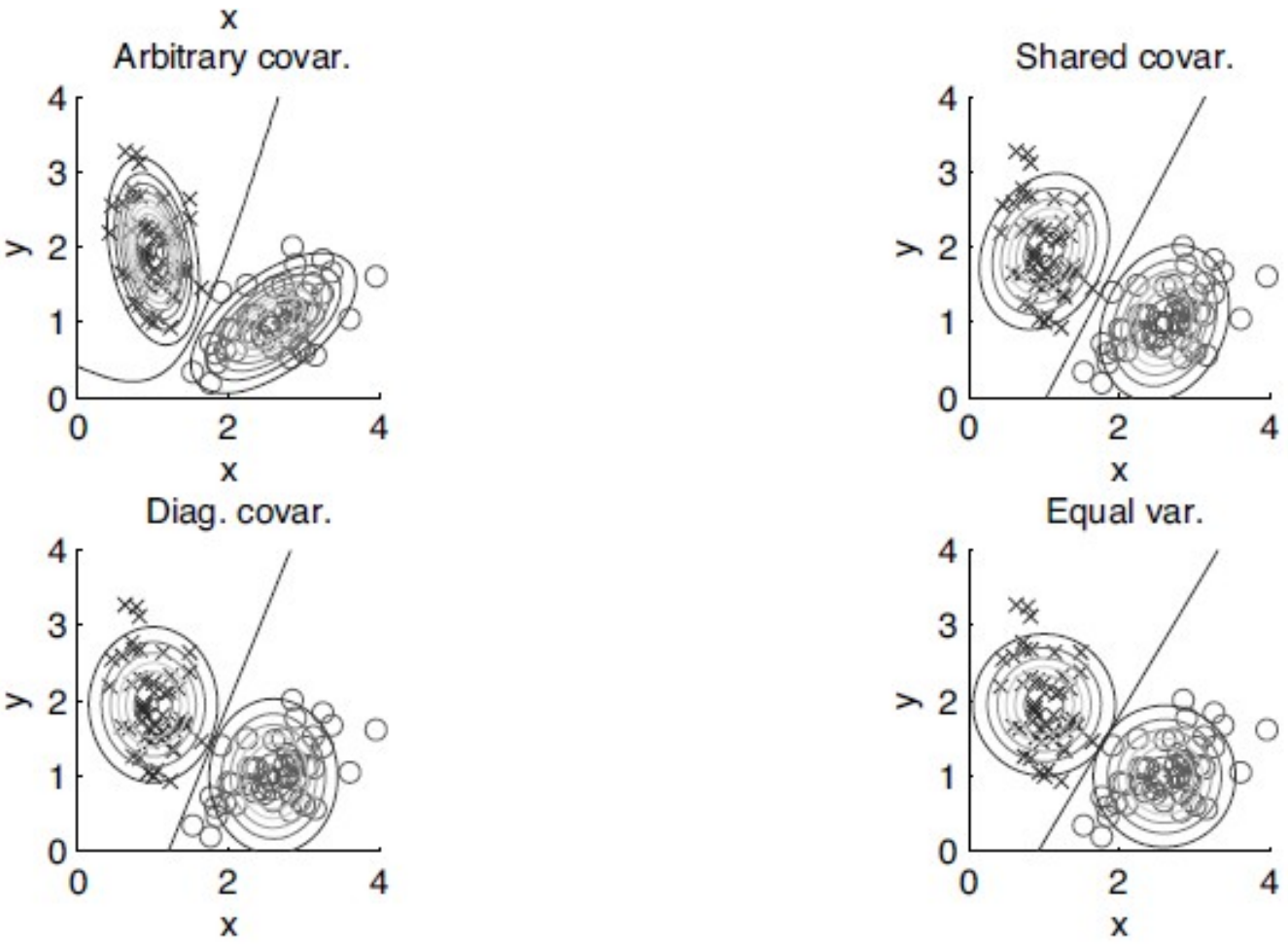
- Discrete Features
- Multivariate Regression

Ch 6: Dimensionality reduction

CHAPTER 5:

Multivariate Methods

Model Selection



Discrete Features

- Binary features: $p_{ij} \equiv p\left(x_j = 1 | C_i\right)$

  if x_j are independent (Naive Bayes')

$$p\left(\mathbf{x} | C_i\right) = \prod_{j=1}^{d} p_{ij}^{x_j} \left(1 - p_{ij}\right)^{\left(1 - x_j\right)}$$

  the discriminant is linear

$$g_i\left(\mathbf{x}\right) = \log p\left(\mathbf{x} | C_i\right) + \log P\left(C_i\right)$$
$$= \sum_j \left[x_j \log p_{ij} + \left(1 - x_j\right) \log\left(1 - p_{ij}\right)\right] + \log P\left(C_i\right)$$

Estimated parameters $\hat{p}_{ij} = \frac{\sum_t x_j^t r_i^t}{\sum_t r_i^t}$

Multivariate Regression

$$r^t = g\left(x^t \mid w_0, w_1, \ldots w_d\right) + \varepsilon$$

- Multivariate linear model

$$w_0 + w_1 x_1^t + w_2 x_2^t + \cdots + w_d x_d^t$$

- $$E\left(w_0, w_1, \ldots w_d \mid \mathrm{X}\right) = \frac{1}{2}\sum_t \left[r^t - w_0 - w_1 x_1^t - \cdots - w_d x_d^t\right]^2$$

Multivariate Regression

$$w_0 + w_1 x_1^t + w_2 x_2^t + \cdots + w_d x_d^t$$

- $$E(w_0, w_1, \ldots w_d \mid X) = \frac{1}{2}\sum_t \left[r^t - w_0 - w_1 x_1^t - \cdots - w_d x_d^t\right]^2$$

$$\begin{aligned}
\sum_t r^t &= N w_0 + w_1 \sum_t x_1^t + w_2 \sum_t x_2^t + \cdots + w_d \sum_t x_d^t \\
\sum_t x_1^t r^t &= w_0 \sum_t x_1^t + w_1 \sum_t (x_1^t)^2 + w_2 \sum_t x_1^t x_2^t + \cdots + w_d \sum_t x_1^t x_d^t \\
\sum_t x_2^t r^t &= w_0 \sum_t x_2^t + w_1 \sum_t x_1^t x_2^t + w_2 \sum_t (x_2^t)^2 + \cdots + w_d \sum_t x_2^t x_d^t \\
&\vdots \\
\sum_t x_d^t r^t &= w_0 \sum_t x_d^t + w_1 \sum_t x_d^t x_1^t + w_2 \sum_t x_d^t x_2^t + \cdots + w_d \sum_t (x_d^t)^2
\end{aligned}$$

CHAPTER 6:

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)
  - Factor Analysis (FA)
  - Linear Discriminant Analysis (LDA)

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|θ)

“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”

Subset Selection

- There are 2^d subsets of d features
- Forward search: Add the best feature at each step
  - Set of features F initially $\emptyset$.
  - At each iteration, find the best new feature

$j = \text{argmin}_i E (F \cup x_i)$

    - Add x_j to F if $E (F \cup x_j) < E (F)$

- $O(d^2)$ algorithm does not guarantee optimal
- Backward search: Start with all features and remove one at a time, if possible.
-

Subset Selection

Backward search: Start with all features and remove one at a time, if possible.

- Set of features F
- At each iteration, remove a feature that does not decrease the error

$j = \text{argmin}_i E (F - x_i)$

- Remove x_j to F if $E (F - x_j) < E (F)$

- O(d^2) algorithm does not guarantee optimal

Subset-selection (recap)

- 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

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

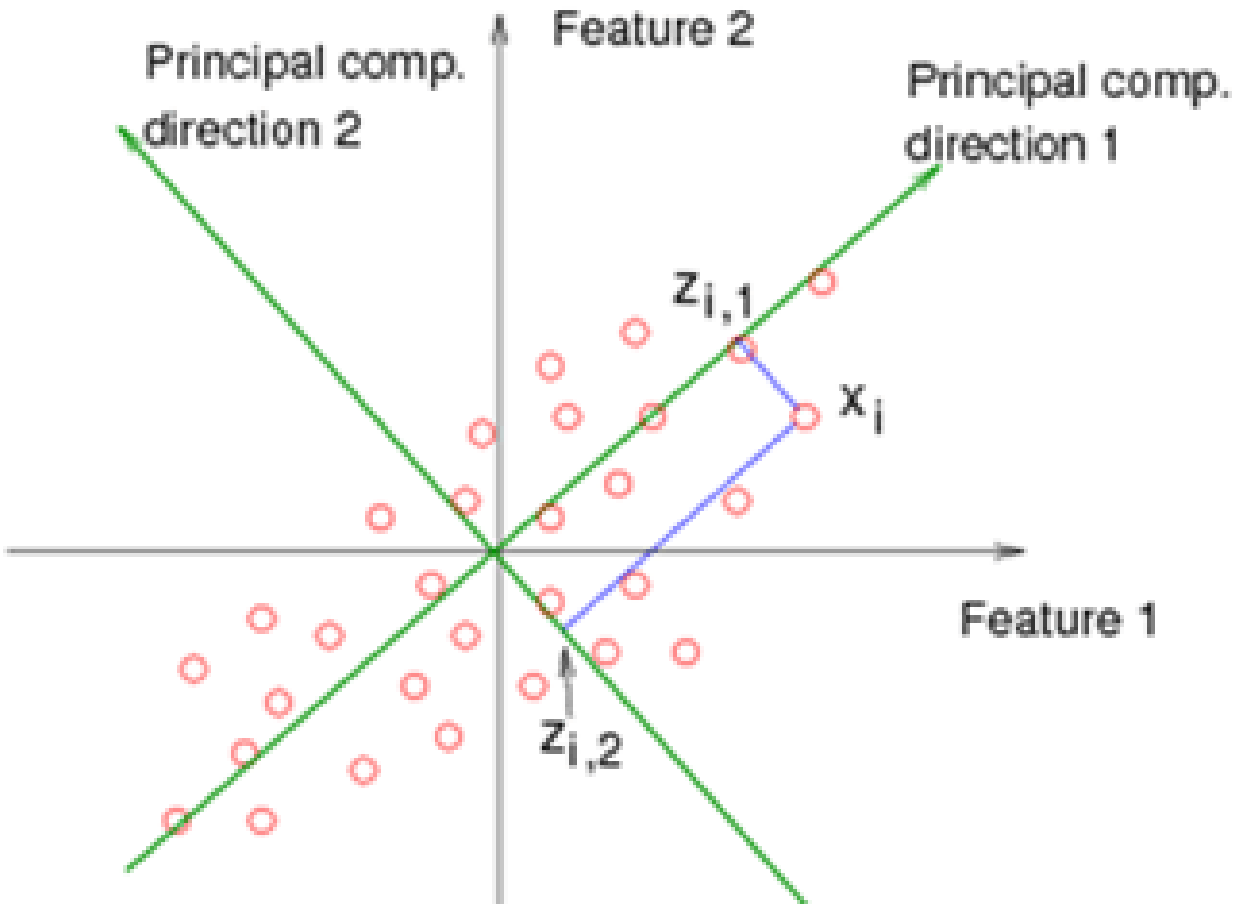
- Find a projection matrix w from d-dimensional to k-dimensional vectors that keeps error low

$$z = w^T x$$

PCA: Motivation

- Assume that d observables are linear combination of k<d vectors
- 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 or otherwise bear no information

PCA: Motivation



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

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

- Taking Derivatives

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

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

PCA

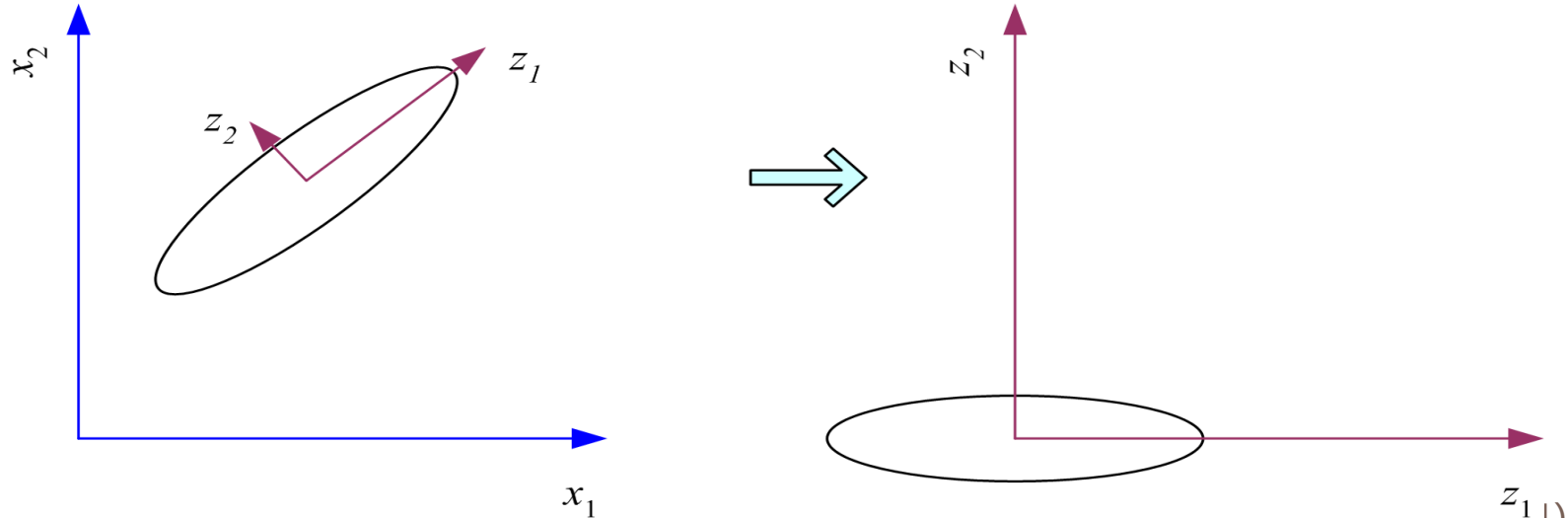
- d-dimensional feature space
- d by d symmetric covariance matrix estimated from samples $\text{Cov}(\boldsymbol{x}) = \boldsymbol{\Sigma}$,
- Select k largest eigenvalue of the covariance matrix and associated k eigenvectors
- The first eigenvector will be a direction with largest variance

What PCA does

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

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

Centers the data at the origin and rotates the axes



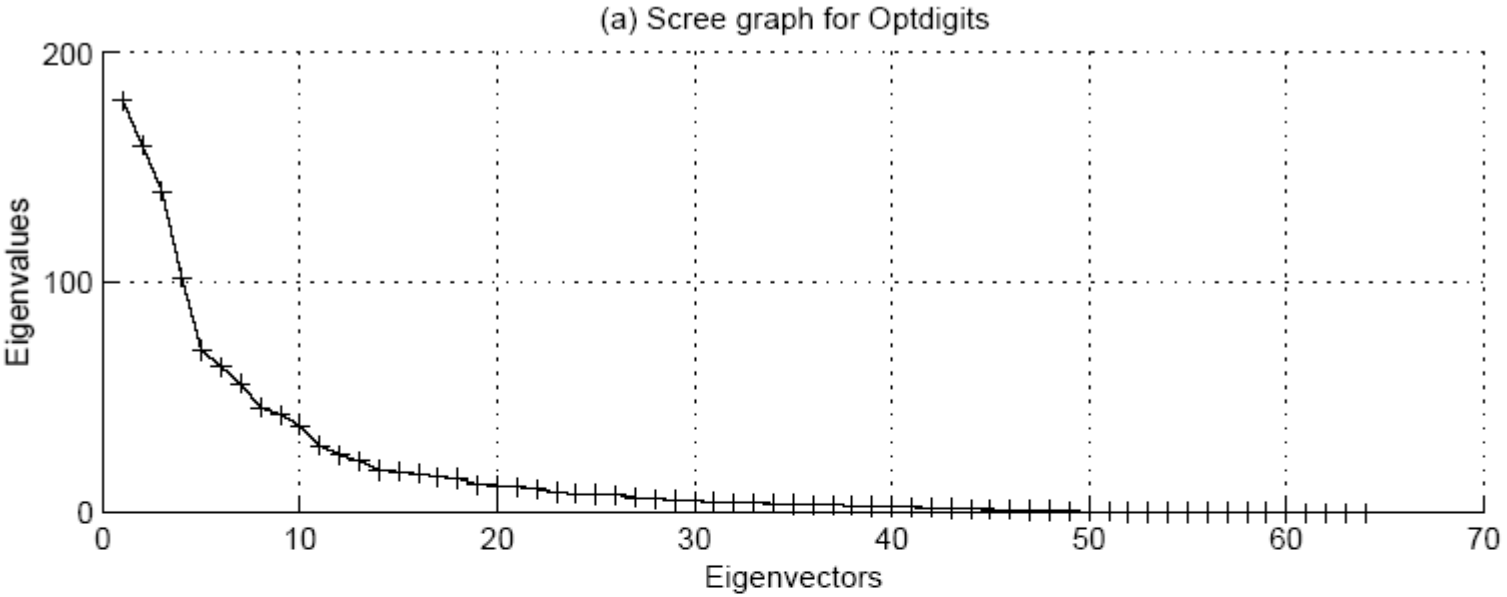
How to choose k ?

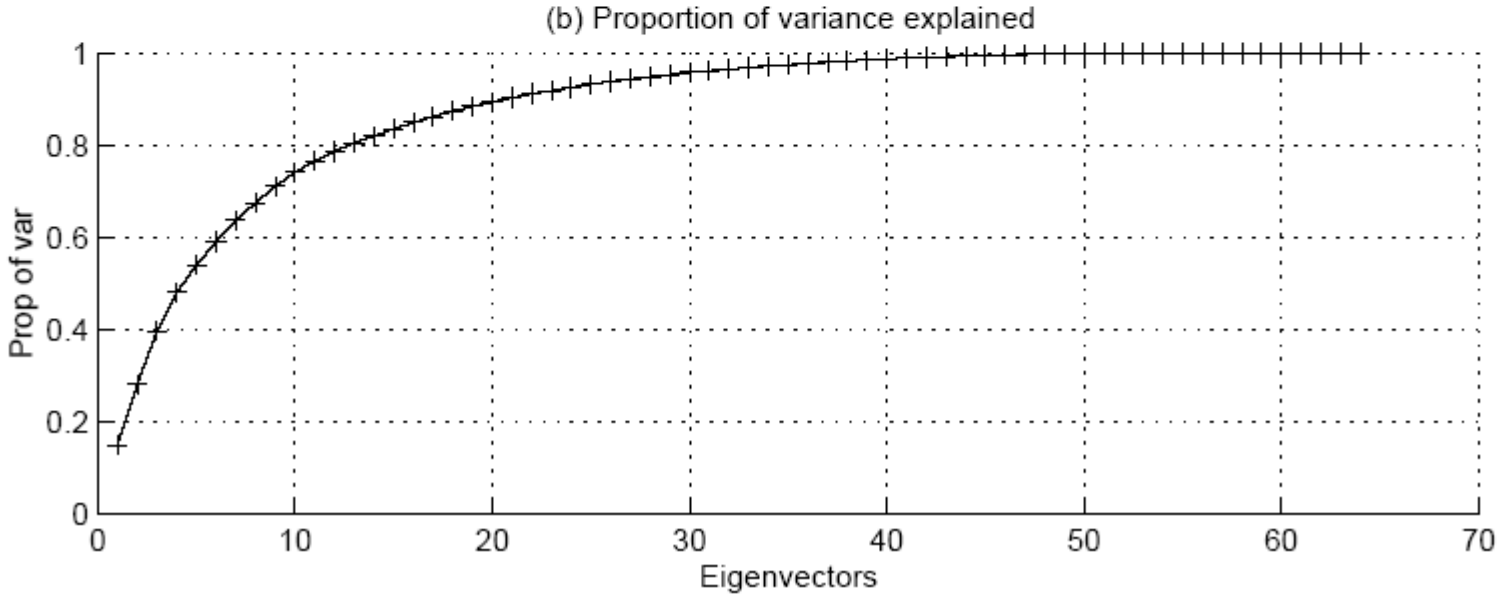
- Proportion of Variance (PoV) explained

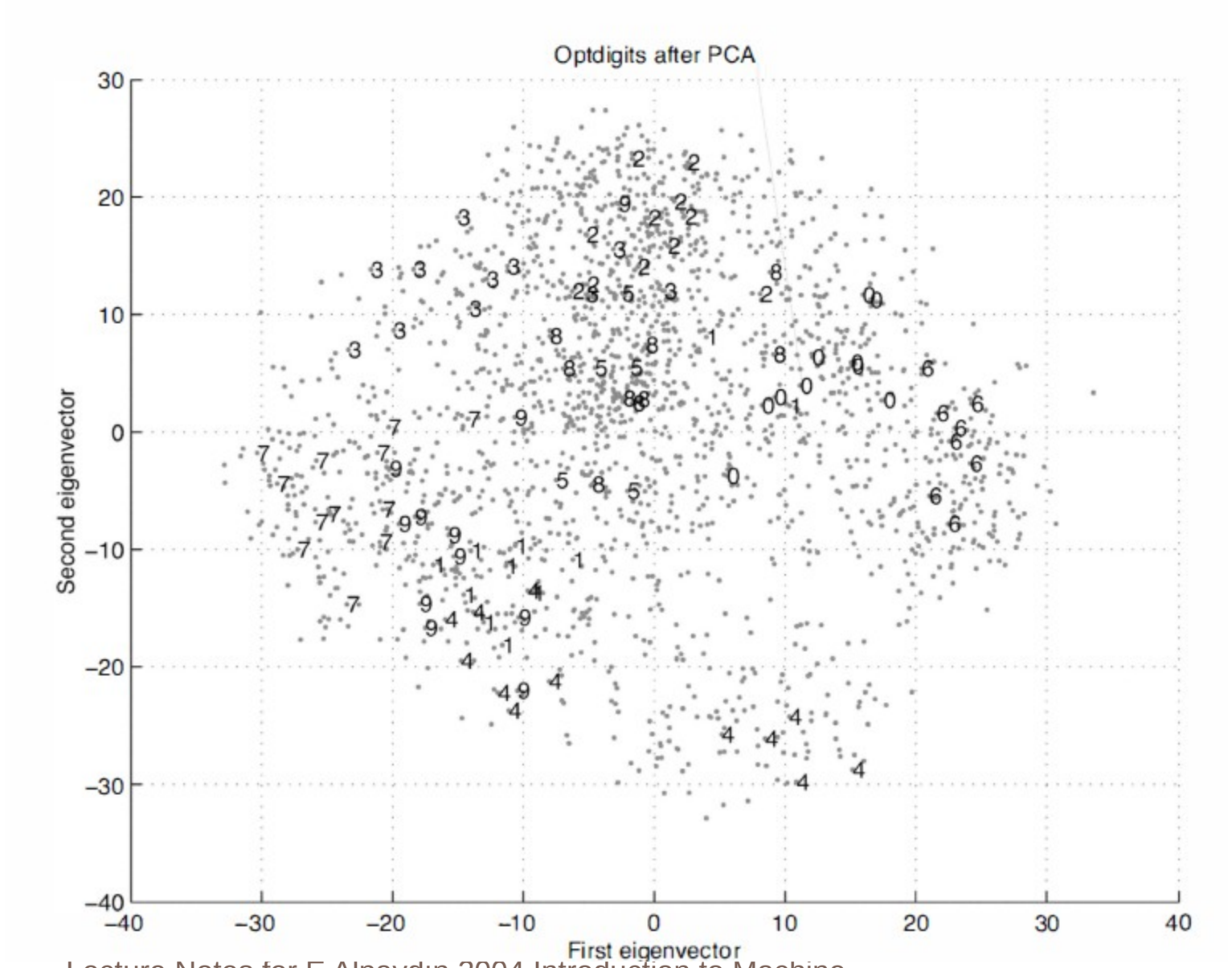
$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

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


Optdigits after PCA
Second eigenvector
First eigenvector
30
20
10
0
−10
−20
−30
−40
−40
−30
−20
−10
0
10
20
30
40

PCA

-
- 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

- PCA is unsupervised (does not take into account class information)
- 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 $\boldsymbol{z}$, which when combined generate $\boldsymbol{x}$:

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

where z_j, j =1,...,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

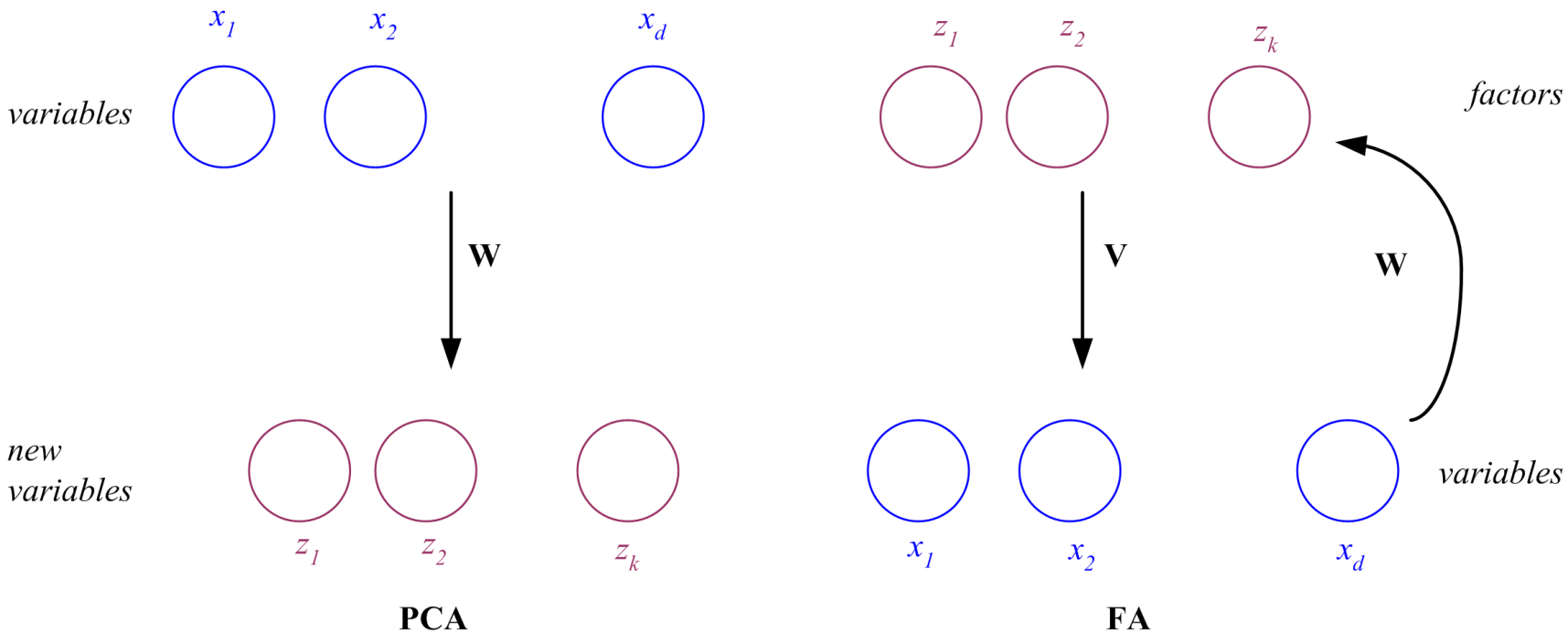
$$E[\, \varepsilon_i \,]= \psi_i,\ \mathrm{Cov}(\varepsilon_i , \varepsilon_j) =0,\ i \neq j,\ \mathrm{Cov}(\varepsilon_i , z_j) =0$$

,

and v_{ij} are the factor loadings $\boldsymbol{x} - \boldsymbol{\mu} = \mathbf{V}\mathbf{z} + \boldsymbol{\epsilon}$

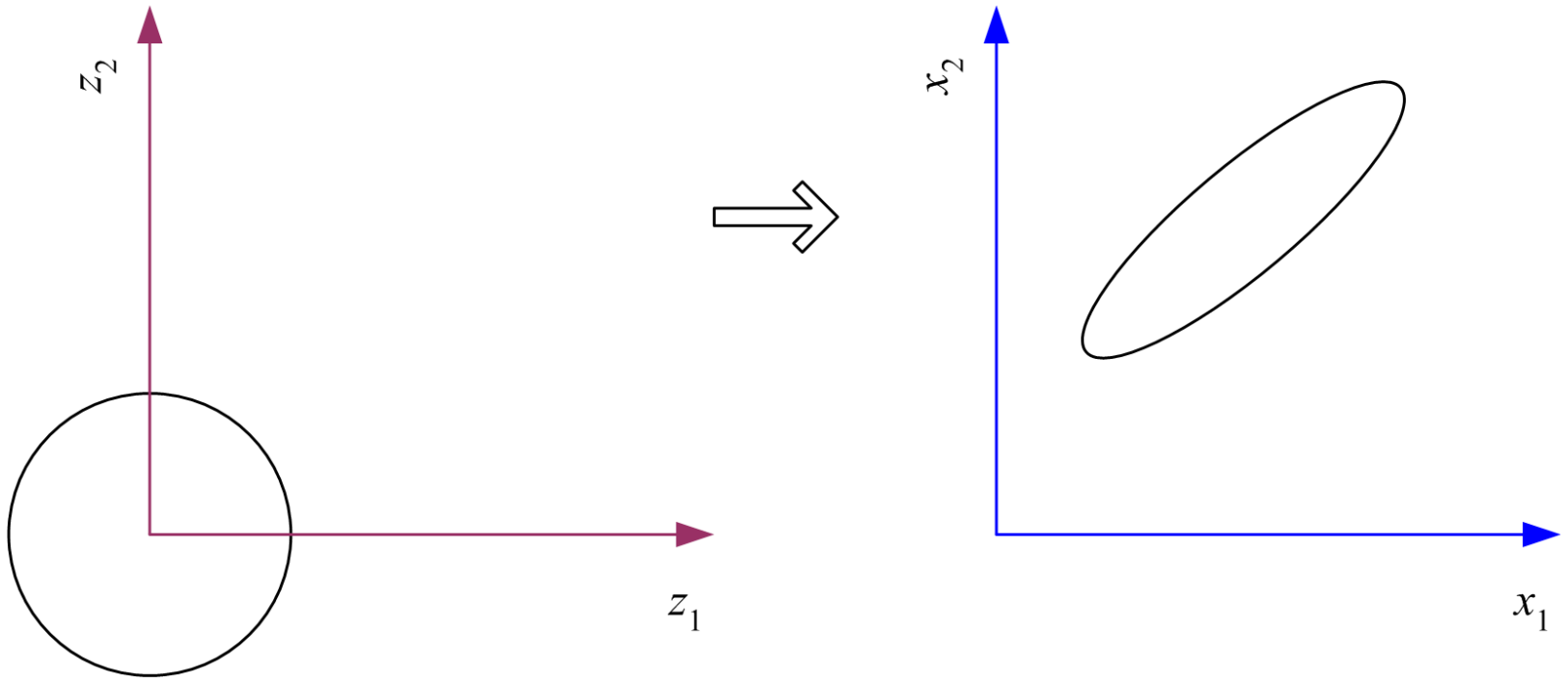
PCA vs FA

- PCA From $\boldsymbol{x}$ to $\boldsymbol{z}$ $\quad \boldsymbol{z} = \mathbf{W}^T(\boldsymbol{x} - \boldsymbol{\mu})$
- FA From $\boldsymbol{z}$ to $\boldsymbol{x}$ $\quad \boldsymbol{x} - \boldsymbol{\mu} = \mathbf{V}\boldsymbol{z} + \boldsymbol{\varepsilon}$



Factor Analysis

- In FA, factors z_j are stretched, rotated and translated to generate $\boldsymbol{x}$

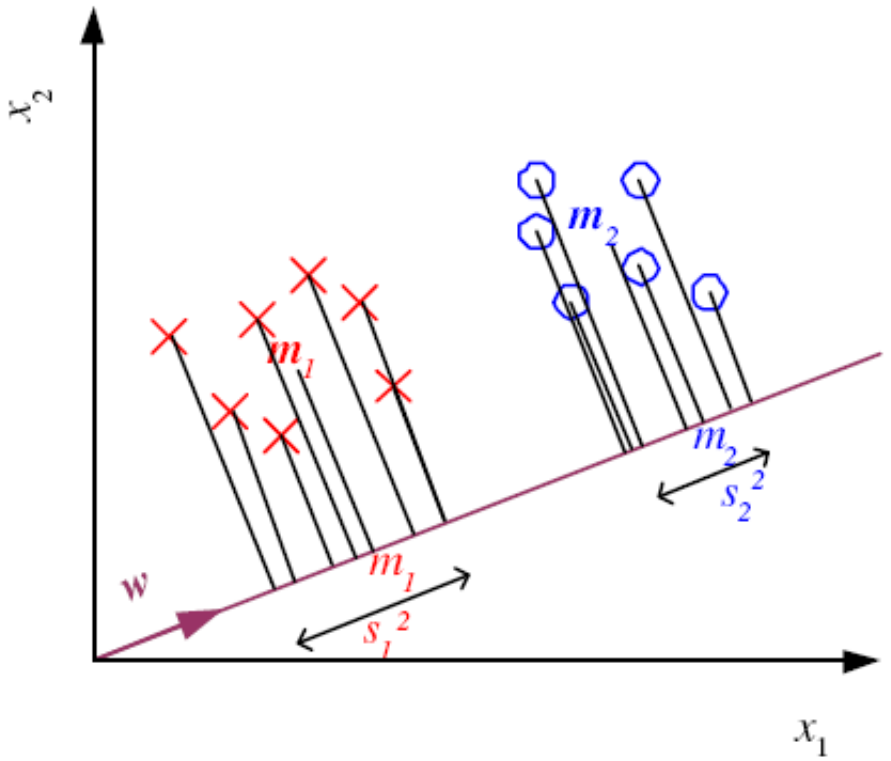


FA Usage

- 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 $\boldsymbol{x}$ is projected, classes are well-separated



Means and Scatter after projection

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

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

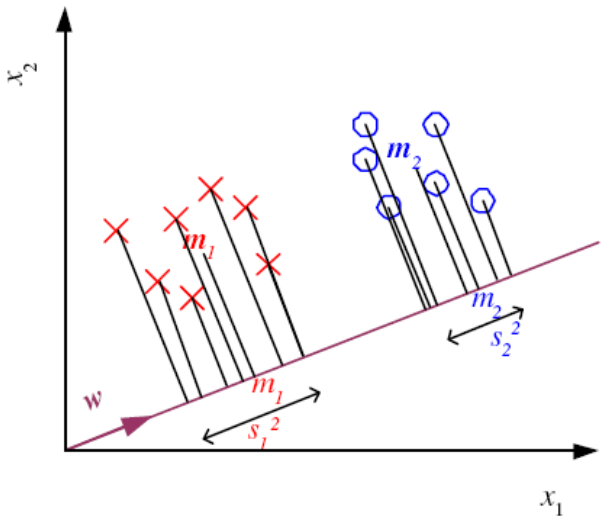
$$s_1^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_1)^2 r^t$$

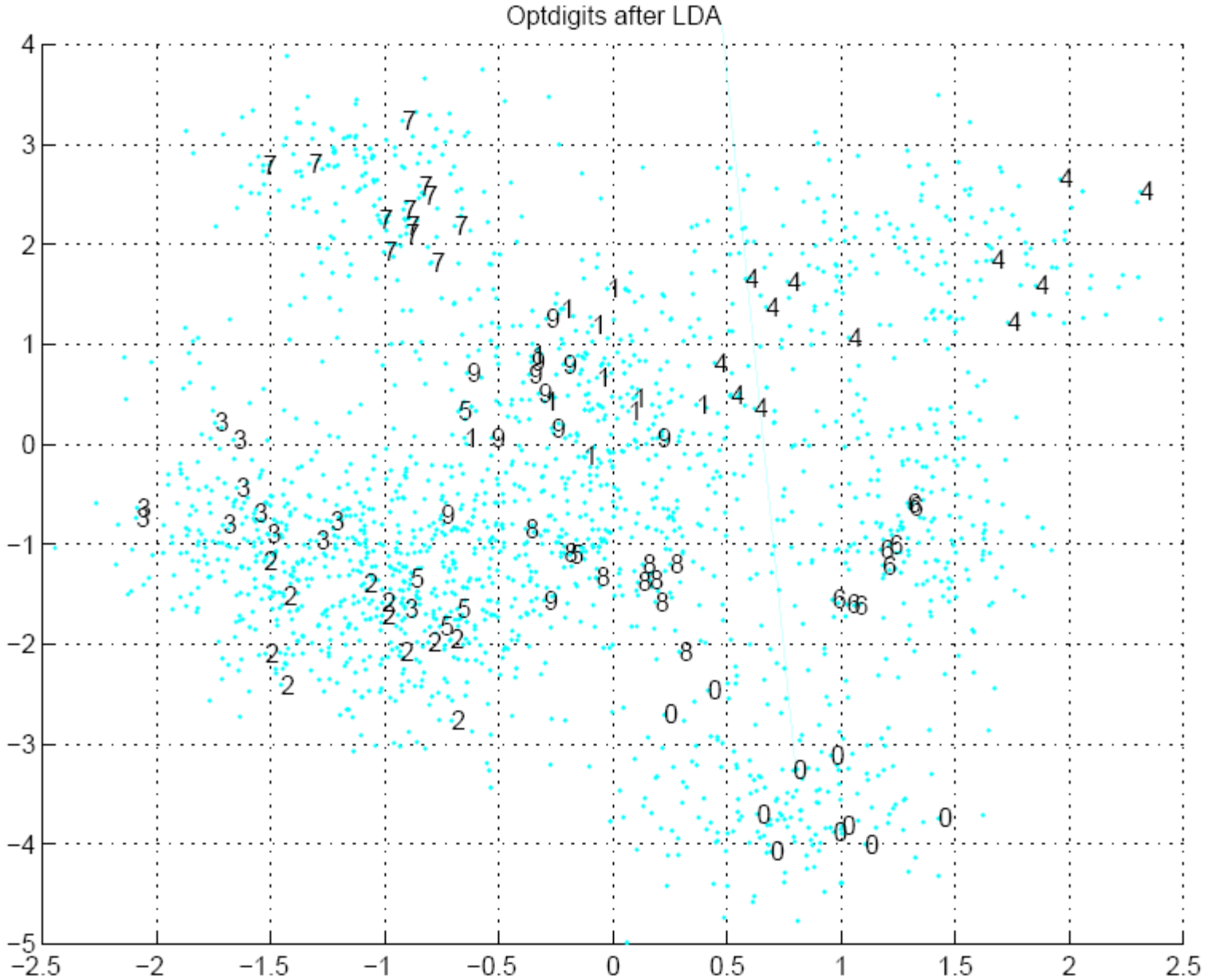
$$s_2^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_2)^2 (1 - r^t)$$

Good Projection

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

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




Optdigits after LDA
4
3
2
1
0
−1
−2
−3
−4
−5
−2.5
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5

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