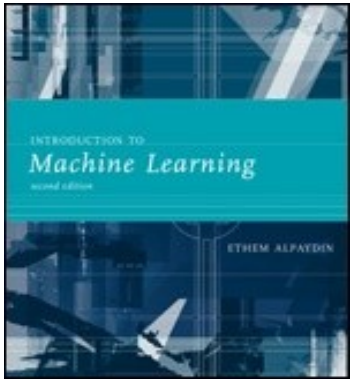
Lecture Slides for

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
Ch 6: Dimensionality reduction

This class:
Ch 6: Dimensionality reduction
Ch 7: Clustering

# Outline

Previous class
Ch 6: Dimensionality reduction
Ch 7: Clustering
This class:
Ch 6: Dimensionality reduction

CHAPTER 6:

# Dimensionality Reduction

# Projection

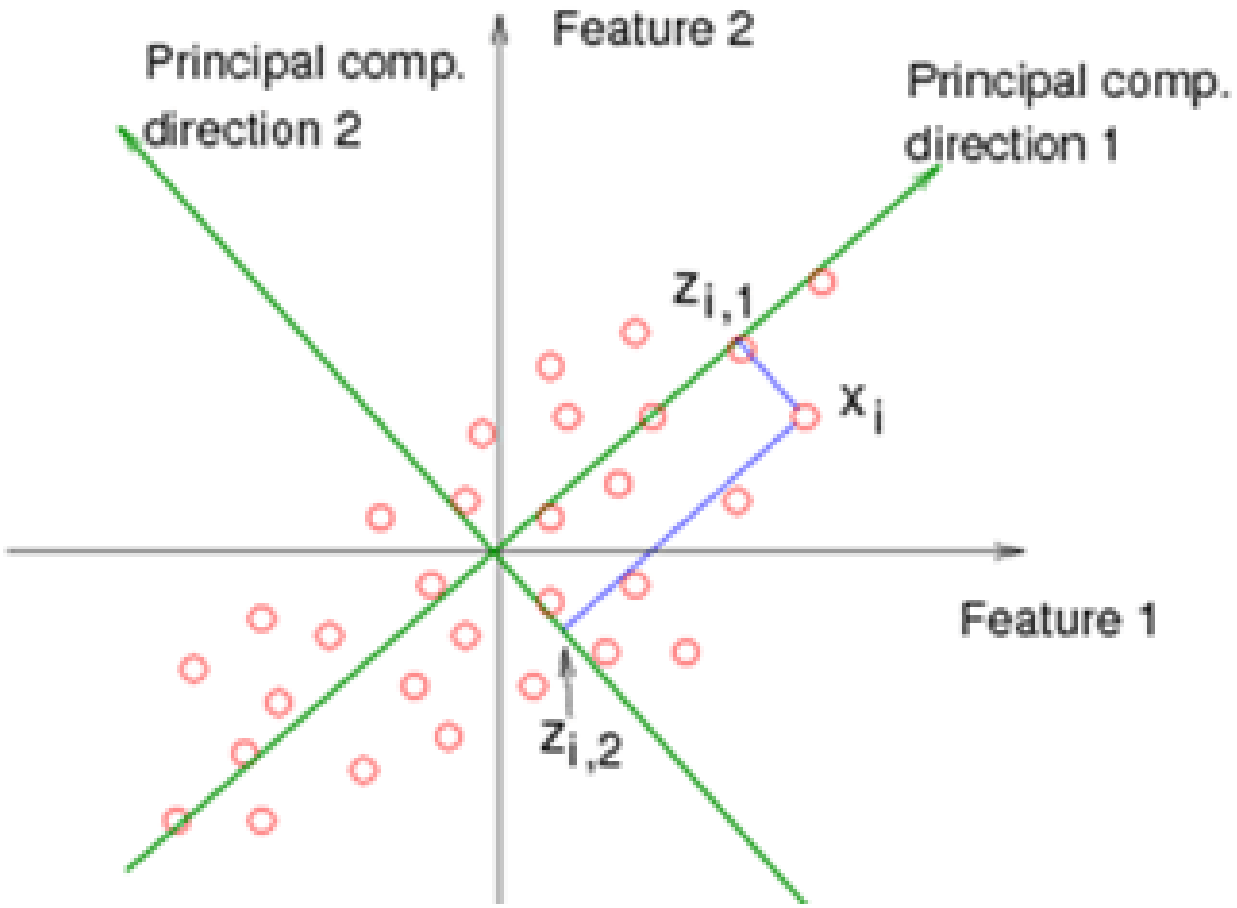
- Find a projection matrix w from d-dimensional to k-dimensional vectors that keeps error low

$$z = w^T x$$

# 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

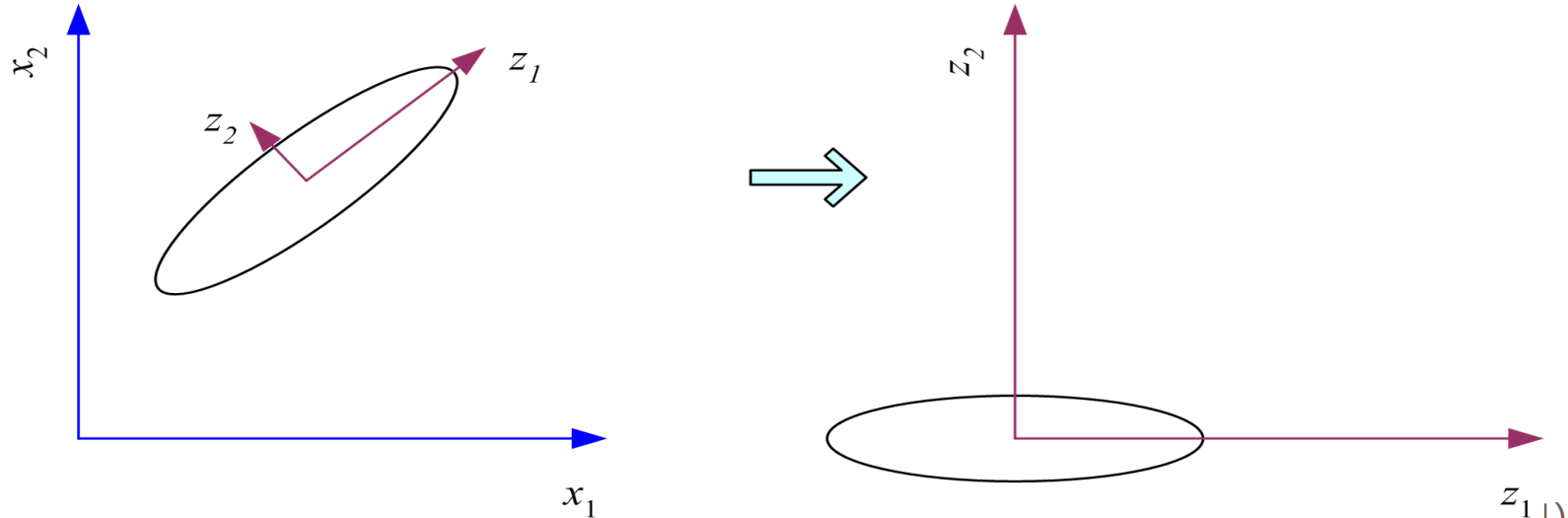
- d-dimensional feature space
- d by d symmetric covariance matrix estimated from samples $\text{Cov}(\boldsymbol{x}) = \boldsymbol{\Sigma}$,
- Select k largest eigenvalue of the covariance matrix and associated k eigenvectors
- The first eigenvector will be a direction with largest variance

# 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 (show code)

Centers the data at the origin and rotates the axes



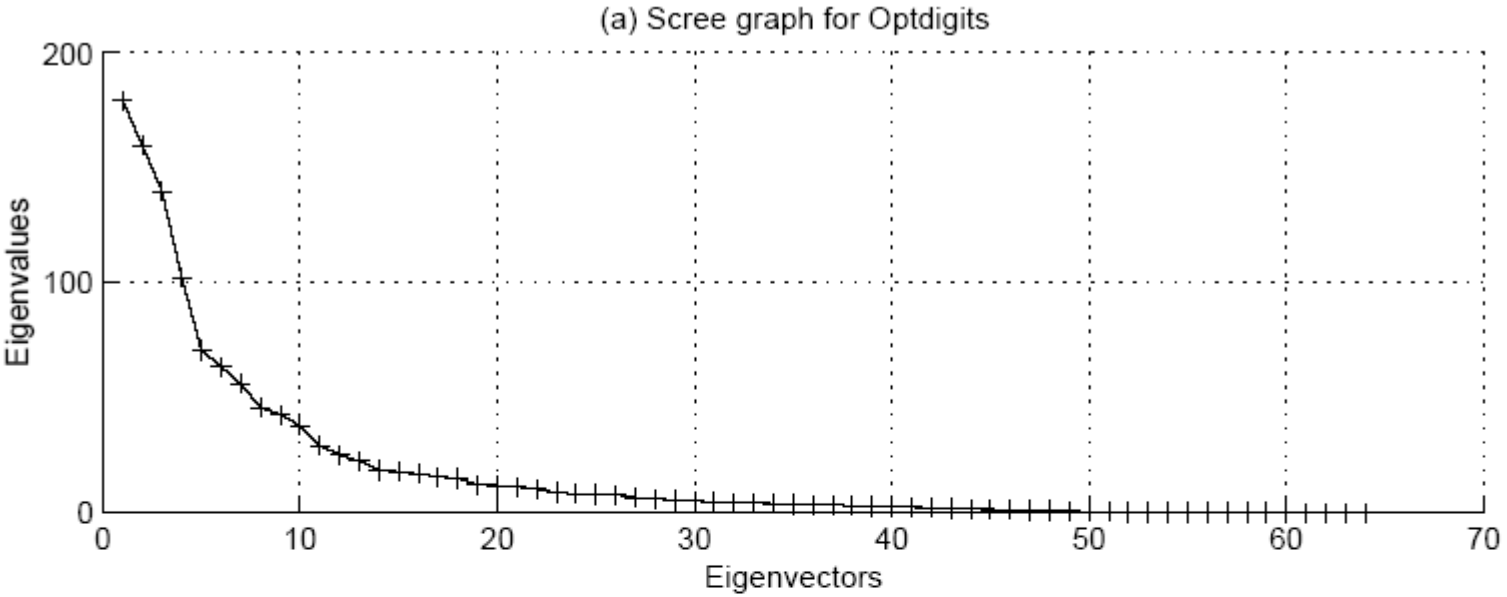
# How to choose k ?

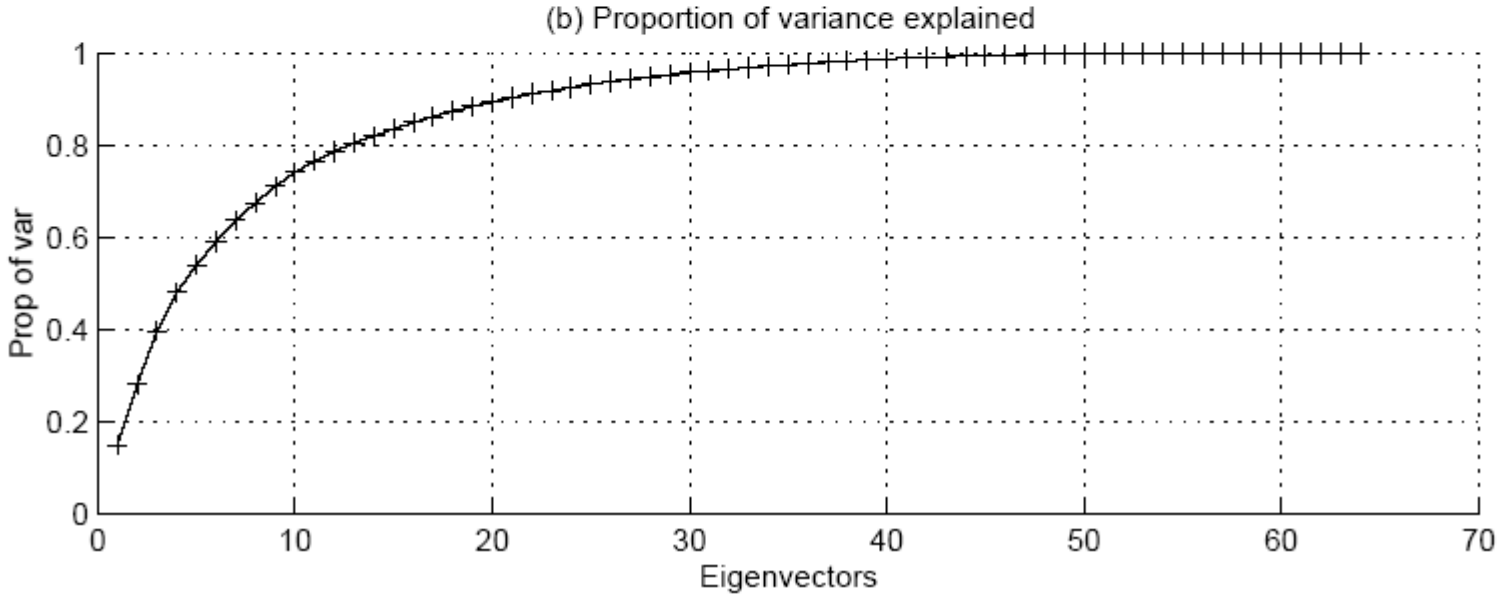
- Proportion of Variance (PoV) explained

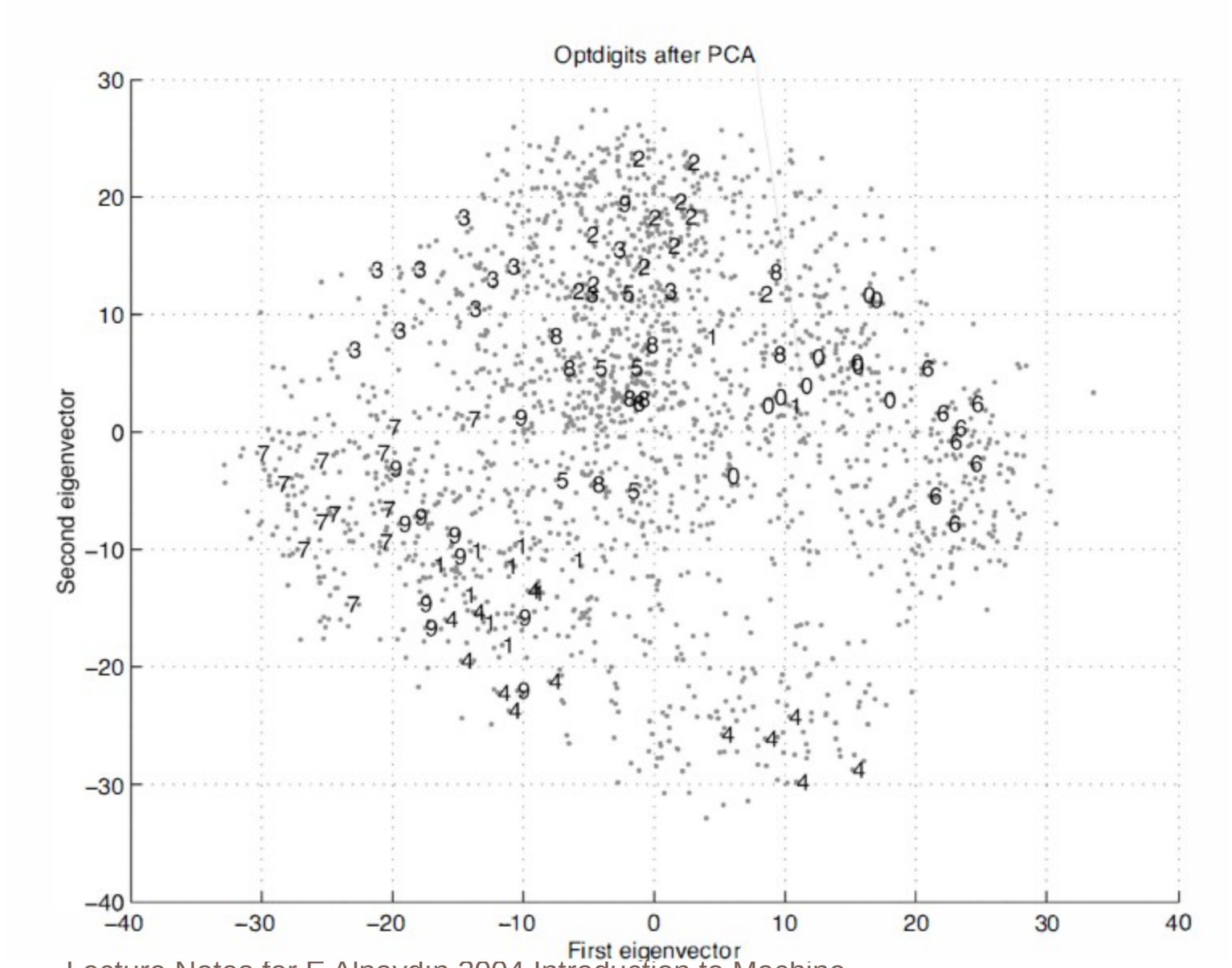
$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

when $\lambda_i$ 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

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

$\varepsilon_i$ are the noise sources

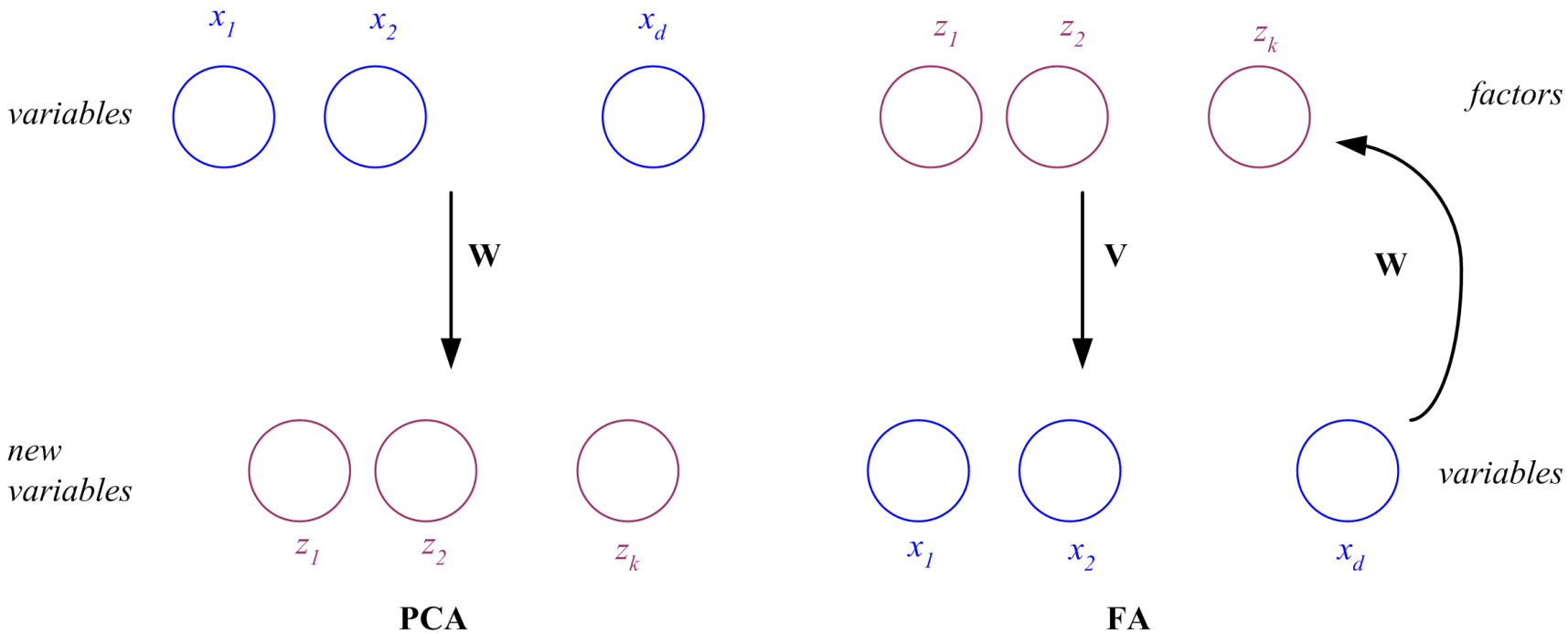
$$E[\, \varepsilon_i \,]= \psi_i,\ \mathrm{Cov}(\varepsilon_i , \varepsilon_j) =0,\ i \neq j,\ \mathrm{Cov}(\varepsilon_i , z_j) =0$$

,

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

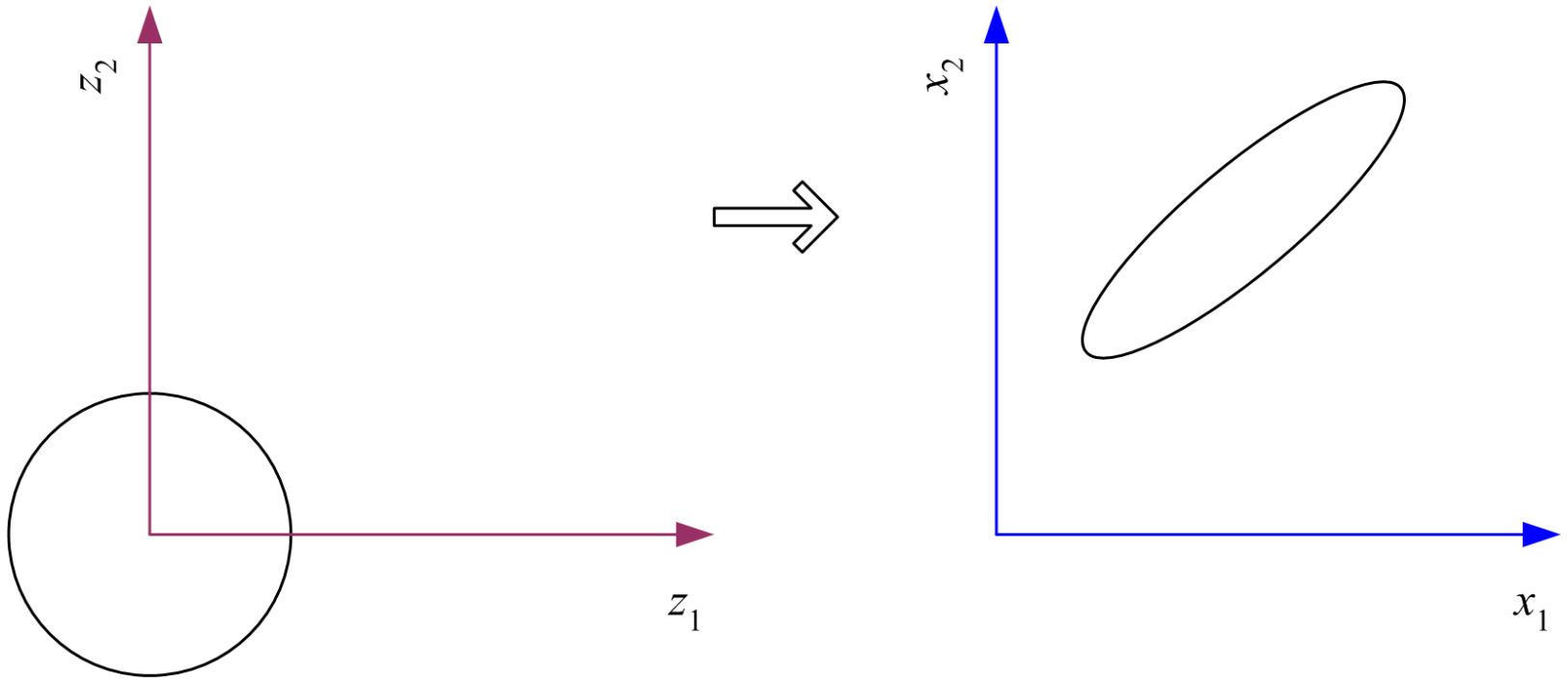
# PCA vs FA

- 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 $\boldsymbol{x}$

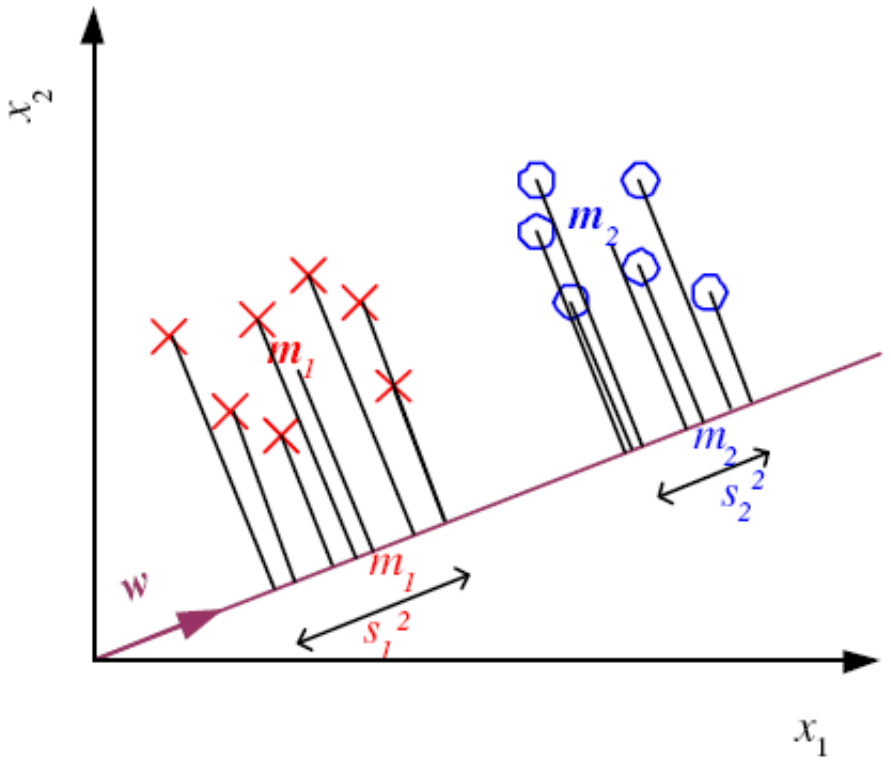


# FA Usage

- 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

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

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

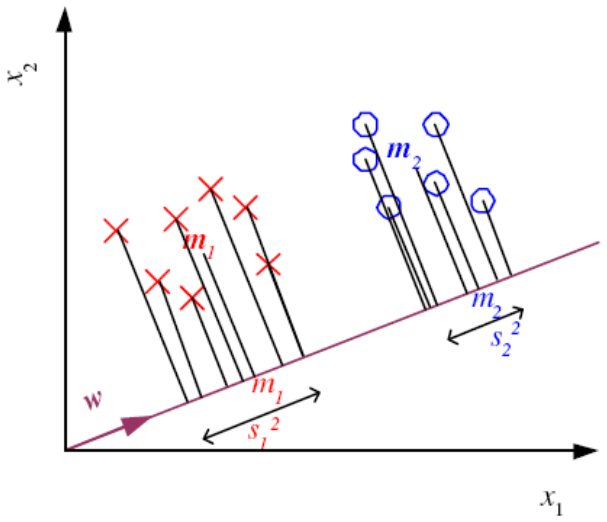
$$s_1^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_1)^2 r^t$$

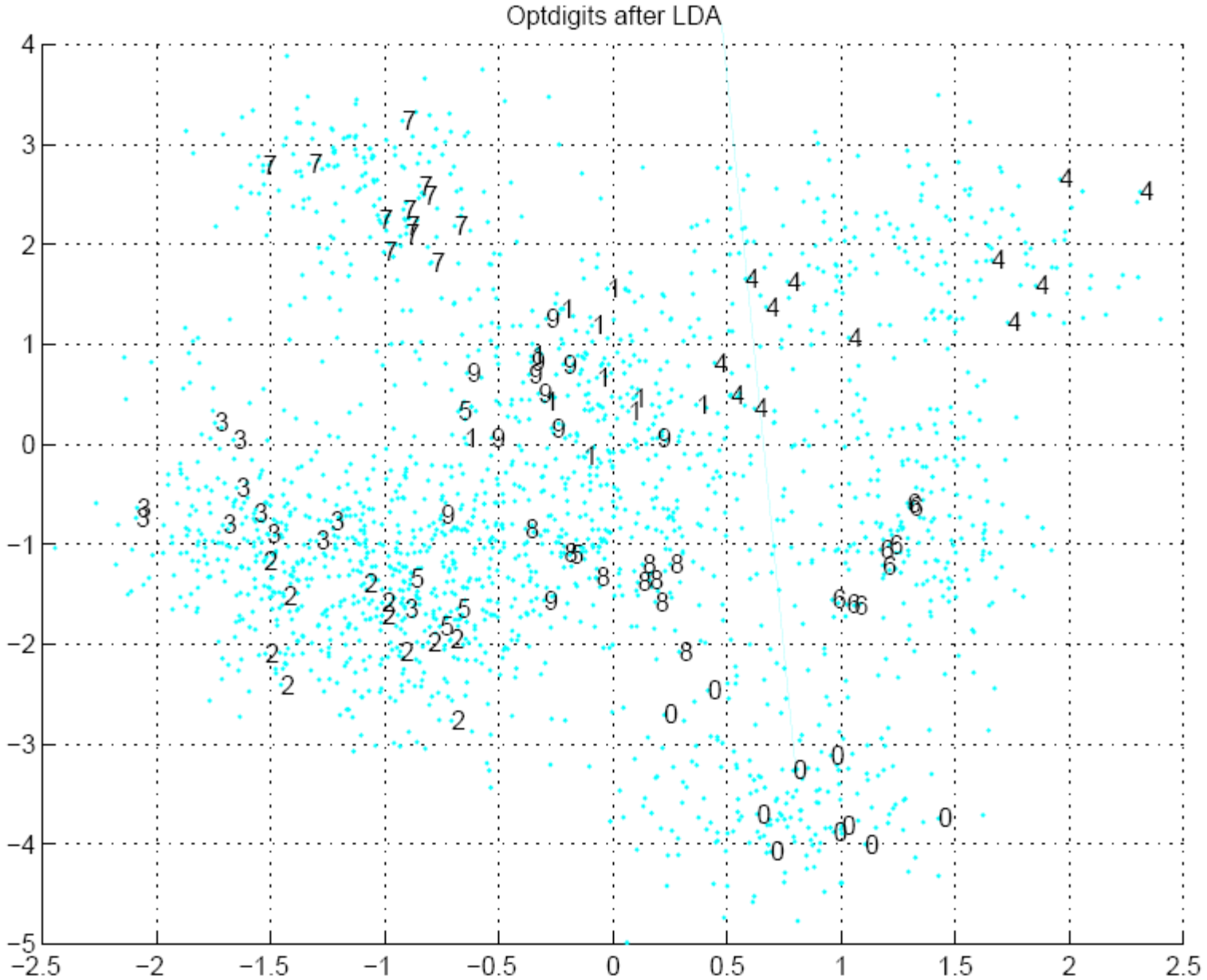
$$s_2^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_2)^2 (1 - r^t)$$

# 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
  - Unsupervised
    - PCA: explain most of the total variability
    - FA: explain most of the common variability
  - Supervised
    - LDA: best separate class instances

CHAPTER 7:

# Clustering

# Motivation

- Classification problem:
  - Need P(C|X)
  - Bayes: can be computed from P(x|C)
  - Need to estimation P(x|C) from data
  - Assume a model (e.g. normal distribution) up to parameters
  - Compute estimators(ML, MAP) for parameters from data

- Regression
  - Need to estimate joint P(x,r)
  - Bayes: can be computed from P(r|x)
  - Assume model up to parameters (e.g. linear)
  - Compute parameters from data (e.g. least squares)

# Motivation

- Not always can assume that data came from single distribution/model
- Nonparametric method: don’t assume any model, compute probability of new data directly from old data
- Semi-parametric/mixture models: assume data came from a unknown mixture of known models

# Motivation

- Optical Character Recognition
  - Two ways to write 7 (w/o horizontal bar)
  - Can’t assume single distribution
  - Mixture of unknown number of templates

- Compared to classification
  - Number of classes is known
  - Each training sample has a label of a class
  - Supervised Learning

# Mixture Densities

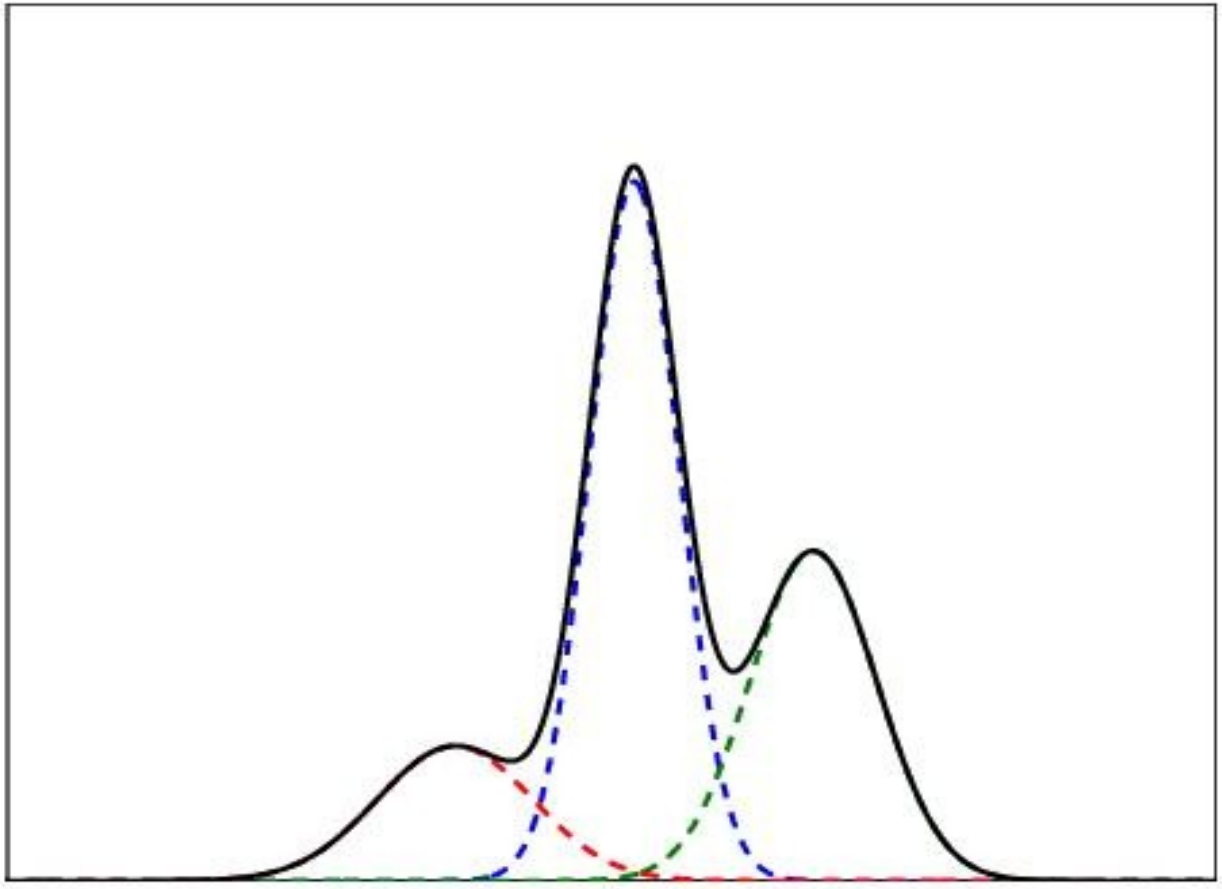
$$p(\boldsymbol{x}) = \sum_{i=1}^{k} p(\boldsymbol{x} \mid \mathrm{G}_i) P(\mathrm{G}_i)$$

- where $\mathrm{G}_i$ the components/groups/clusters,
  $P(\mathrm{G}_i)$ mixture proportions (priors),
  $p(\boldsymbol{x} \mid \mathrm{G}_i)$ component densities

- Gaussian mixture where $p(\boldsymbol{x}|\mathrm{G}_i) \sim \mathrm{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ parameters $\Phi = \{P(\mathrm{G}_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{k}$
  unlabeled sample $X=\{\boldsymbol{x}^t\}_t$ (unsupervised learning)

# Example

- C



# Example : Color quantization

- Image: each pixels represented by 24 bit color
- Colors come from different distribution (e.g. sky, grass)
- Don’t have labeling for each pixels if it’s sky or grass
- Want to use only 256 colors in palette to represent image as close as possible to original
- Quantize uniformly: assign single color to each 2^24/256 interval
- Waste values for rarely occurring intervals

# Quantization

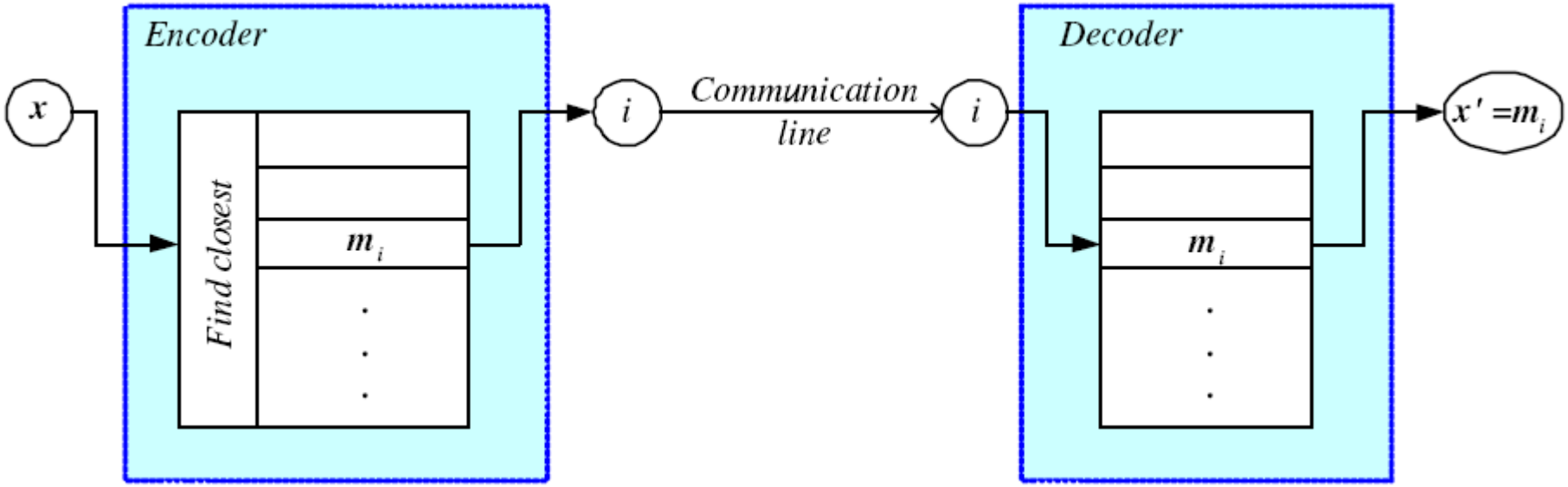
- Sample (pixels): $\mathcal{X} = \{\boldsymbol{x}^t\}_{t=1}^N$
- k reference vectors (palette): $\boldsymbol{m}_j, j = 1, \ldots, k.$
- Select reference vector for each pixel:

$$\|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\|$$

- Reference vectors: codebook vectors or code words
- Compress image $E\left(\{\boldsymbol{m}_i\}_{i=1}^k | \mathcal{X}\right) = \sum_t \sum_i b_i^t \|\boldsymbol{x}^t - \boldsymbol{m}_i\|$
- Reconstruction error

$$b_i^t = \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

# Encoding/Decoding



# K-means clustering

- Minimize reconstruction error

$$E\left(\{\mathbf{m}_i\}_{i=1}^{k} \mid \mathcal{X}\right) = \sum_t \sum_i b_i^t \left\| \mathbf{x}^t - \mathbf{m}_i \right\|$$

- Take derivatives and set to zero

$$\boldsymbol{m}_i = \frac{\sum_t b_i^t \boldsymbol{x}^t}{\sum_t b_i^t}$$

- Reference vectors is the mean of all instances it represents

# K-Means clustering

- Iterative procedure for finding reference vectors
- Start with random reference vectors
- Estimate labels b
- Re-compute reference vectors as means
- Continue till converge

# k-means Clustering

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to $k$ random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

For all $\boldsymbol{m}_i, i = 1, \ldots, k$

$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

Until $\boldsymbol{m}_i$ converge

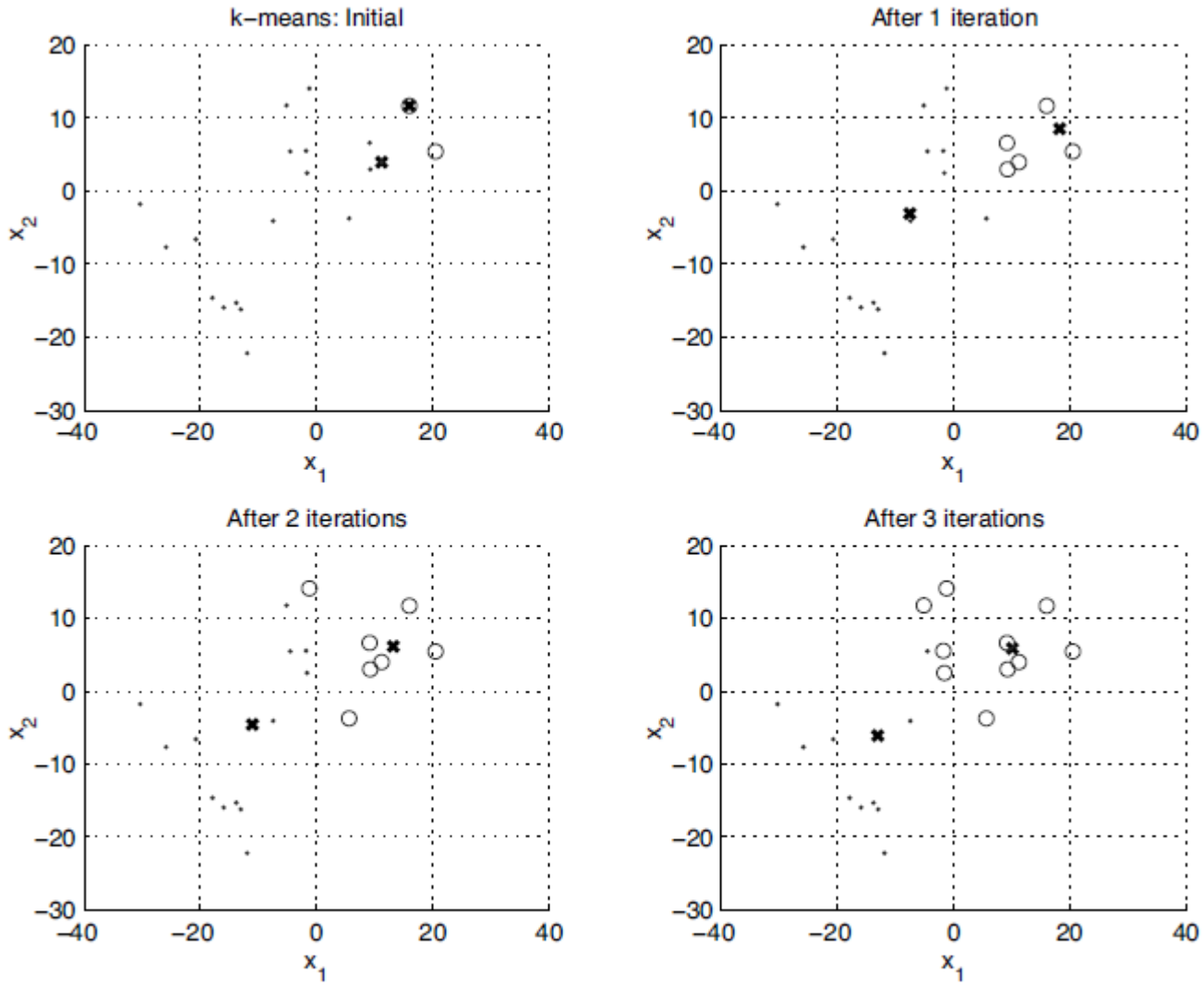



**Figure 7.2** Evolution of *k*-means. Crosses indicate center positions. Data points are marked depending on the closest center.