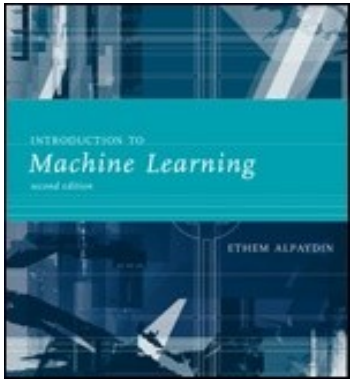
Lecture Slides for

INTRODUCTION TO

Machine Learning

2nd Edition



ETHEM ALPAYDIN, modified by Leonardo Bobadilla and some parts from http://www.cs.tau.ac.il/~apartzin/MachineLearning/ and www.cs.princeton.edu/courses/archive/fall01/cs302/notes/11.../EM.ppt

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

Outline

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

CHAPTER 7:

Clustering

Clustering: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

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

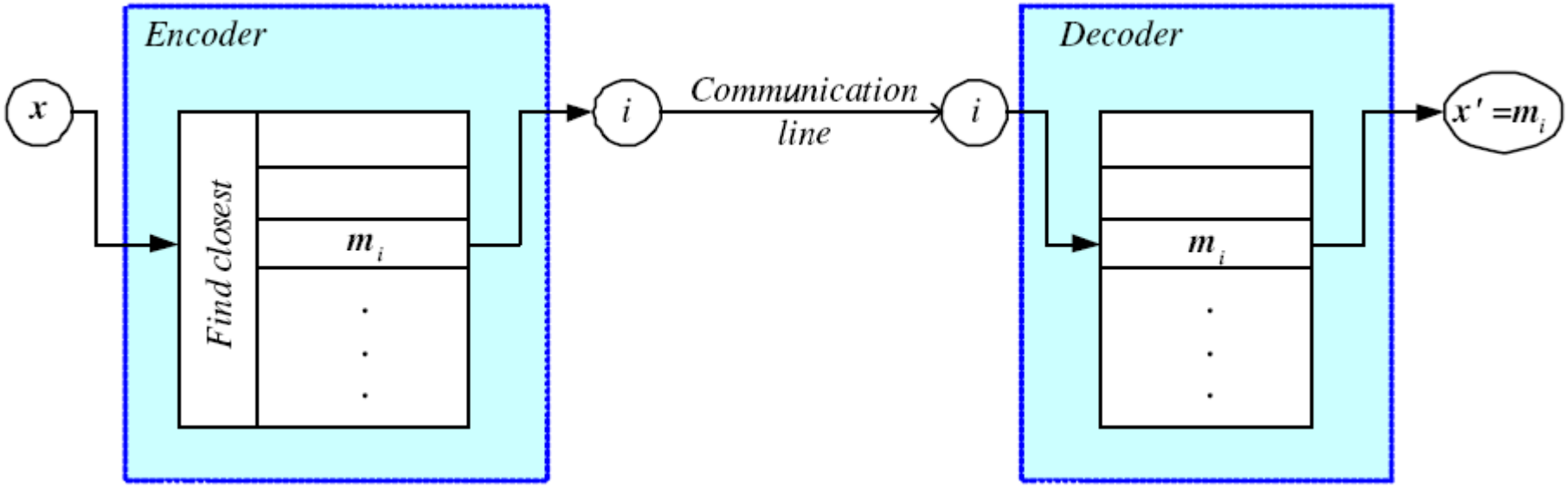
- Sample (pixels): $\mathcal{X} = \{\boldsymbol{x}^t\}_{t=1}^N$
- k reference vectors (palette): $\boldsymbol{m}_j, j = 1, \ldots, k.$
- Select reference vector for each pixel:

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

- 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

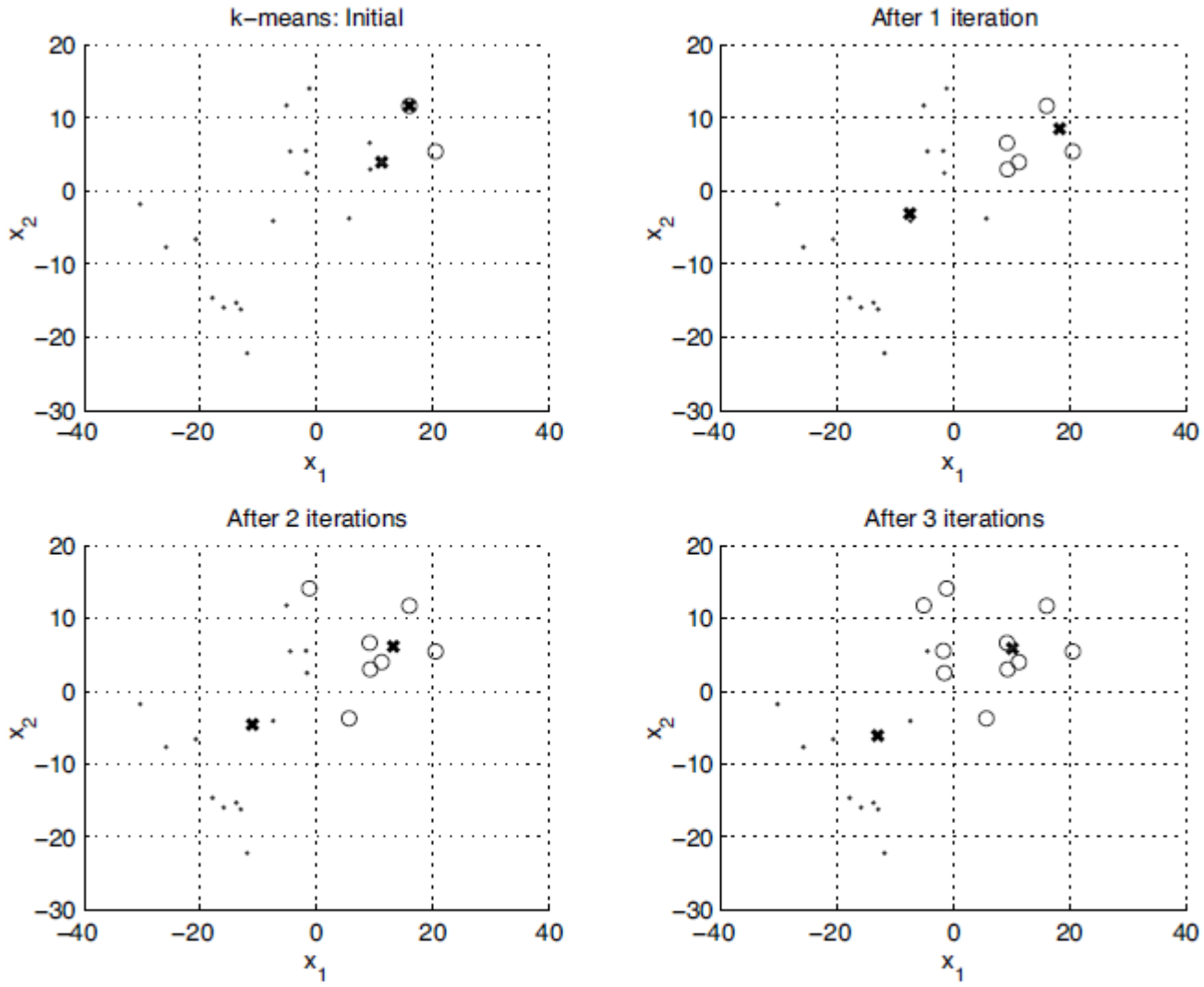


Figure 7.2 Evolution of *k*-means. Crosses indicate center positions. Data points are marked depending on the closest center.

Expectation Maximization: Learning from Data

We want to learn a model with a set of parameter values Φ

We are given a set of data *X*.

An approach: $\text{argmax}_{\Phi} \Pr(X|\Phi)$

This is the *maximum likelihood* model (ML).

Super Simple Example

Coin I and Coin II. (biased.)

Pick a coin at random (uniform).

Flip it 4 times.

Repeat.

What are the parameters of the model?

Data

Coin I	Coin II
HHHT	TTTH
HTHH	THTT
HTTH	TTHT
THHH	HTHT
HHHH	HTTT

Probability of X Given Φ

p: Probability of H from Coin I

q: Probability of H from Coin II

Let's say h heads and t tails for Coin I. h' and t' for Coin II.

$\Pr(X|\Phi) = p^h (1-p)^t q^{h'} (1-q)^{t'}$

How maximize this quantity?

Maximizing p

Use maximum likelihood.

$h/(t+h) = p$

Missing Data

HHHT

TTTH

THTT

TTHT

THHH

HTTH

HTHH

HTTT

HHHH

HTHT

Oh Boy, Now What!

If we knew the labels (which flips from which coin), we could find ML values for p and q.

What could we use to label?

p and q!

Computing Labels

p = ¾, q = 3/10

Pr(Coin I | HHTH)

= Pr(HHTH | Coin I) Pr(Coin I) / c

= $(3/4)^3(1/4)\ (1/2)/c$ = .052734375/c

Pr(Coin II | HHTH)

= Pr(HHTH | Coin II) Pr(Coin II) / c

= $(3/10)^3(7/10)\ (1/2)/c$= .00945/c

Expected Labels

	I	II		I	II
HHHT	.85	.15	HTTH	.44	.56
TTTH	.10	.90	HTHH	.85	.15
THTT	.10	.90	HTTT	.10	.90
TTHT	.10	.90	HHHH	.98	.02
THHH	.85	.15	HTHT	.44	.56

Wait, I Have an Idea

Pick some mode Φ_0

Expectation

- Compute expected labels via Φ_i

Maximization

- Compute ML model Φ_{i+1}

Repeat

Could This Work?

Expectation-Maximization (EM)

$\Pr(X|\Phi_i)$ will not decrease.

Sound familiar? Type of search.

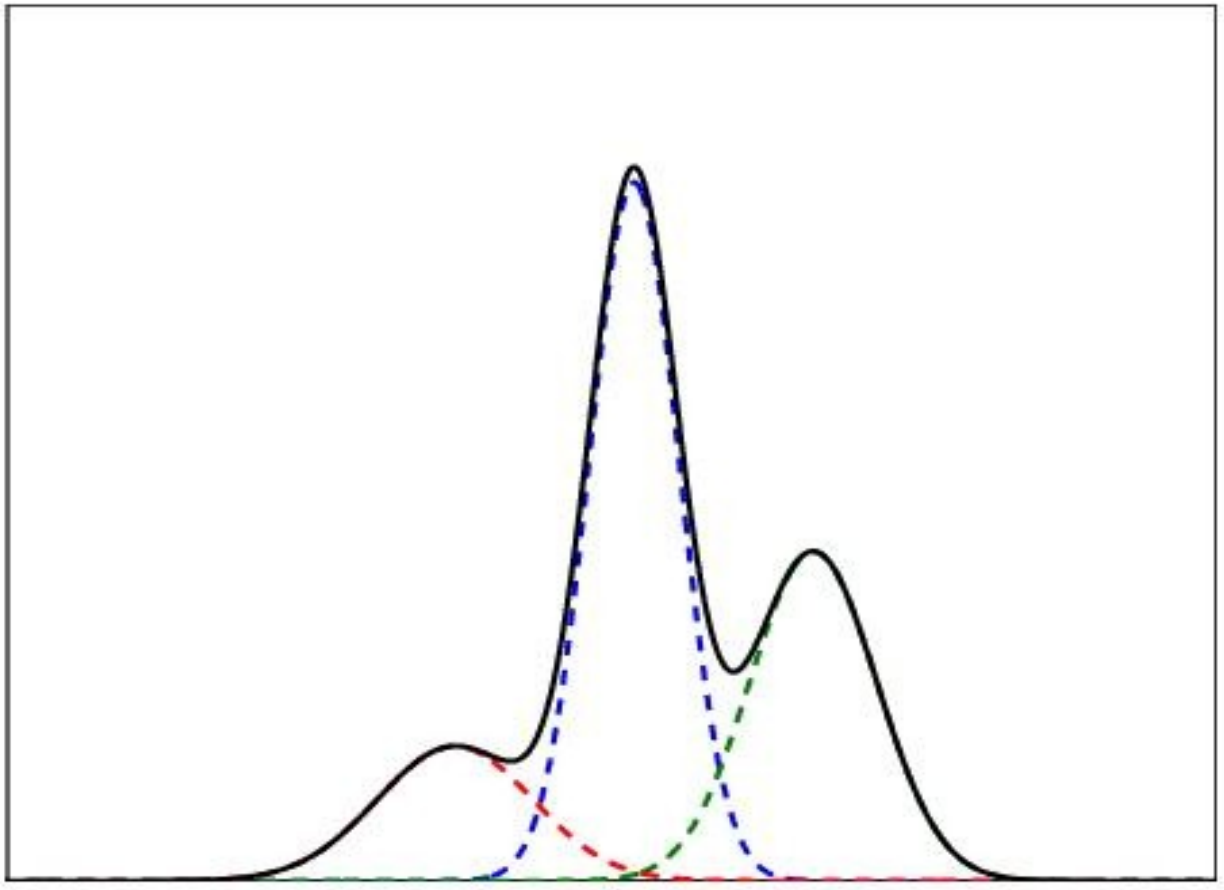
Mixture Densities

$$p(x) = \sum_{i=1}^{k} p(x|G_i) P(G_i)$$

- where G_i the components/groups/clusters,
  $P(G_i)$ mixture proportions (priors),
  $p(\boldsymbol{x}|G_i)$ component densities

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

Example

- 

Expectation Maximization(EM): Motivation

- Date came from several distributions
- Assume each distribution is known up to parameters
- If we would know for each data instance from what distribution it came, could use parametric estimation
- Introduce unobservable (latent) variables which indicate source distribution
- Run iterative process
  - Estimate latent variables from data and current estimation of distribution parameters
  - Use current values of latent variables to refine parameter estimation

EM

- Log-Likelihood $$L(\Phi \mid X) = \log \prod_t p(\boldsymbol{x}^t \mid \Phi)$$
$$= \sum_t \log \sum_{i=1}^{k} p(\boldsymbol{x}^t \mid G_i) P(G_i)$$
- Assume hidden variables Z, which when known, make optimization much simpler
- Complete likelihood, $L_c(\Phi \mid X,Z)$, in terms of $\boldsymbol{X}$ and $\boldsymbol{Z}$
- Incomplete likelihood, $L(\Phi \mid X)$, in terms of $\boldsymbol{X}$

Latent Variables

- Unknown
- Can’t compute complete likelihood $L_c(\Phi \mid X,Z)$
- Can compute its expected value

$$\text{E-step:} Q\left(\Phi|\Phi^l\right) = E\left[L_C\left(\Phi|X,Z\right)|X,\Phi^l\right]$$

E- and M-steps

- Iterate the two steps:

1. E-step: Estimate z given X and current Φ
2. M-step: Find new Φ' given z, X, and old Φ.

$$\text{E-step: } Q\left(\Phi|\Phi^l\right) = E\left[L_C\left(\Phi|X,Z\right)|X,\Phi^l\right]$$

$$\text{M-step: } \Phi^{l+1} = \arg\max_{\Phi} Q\left(\Phi|\Phi^l\right)$$

Example:

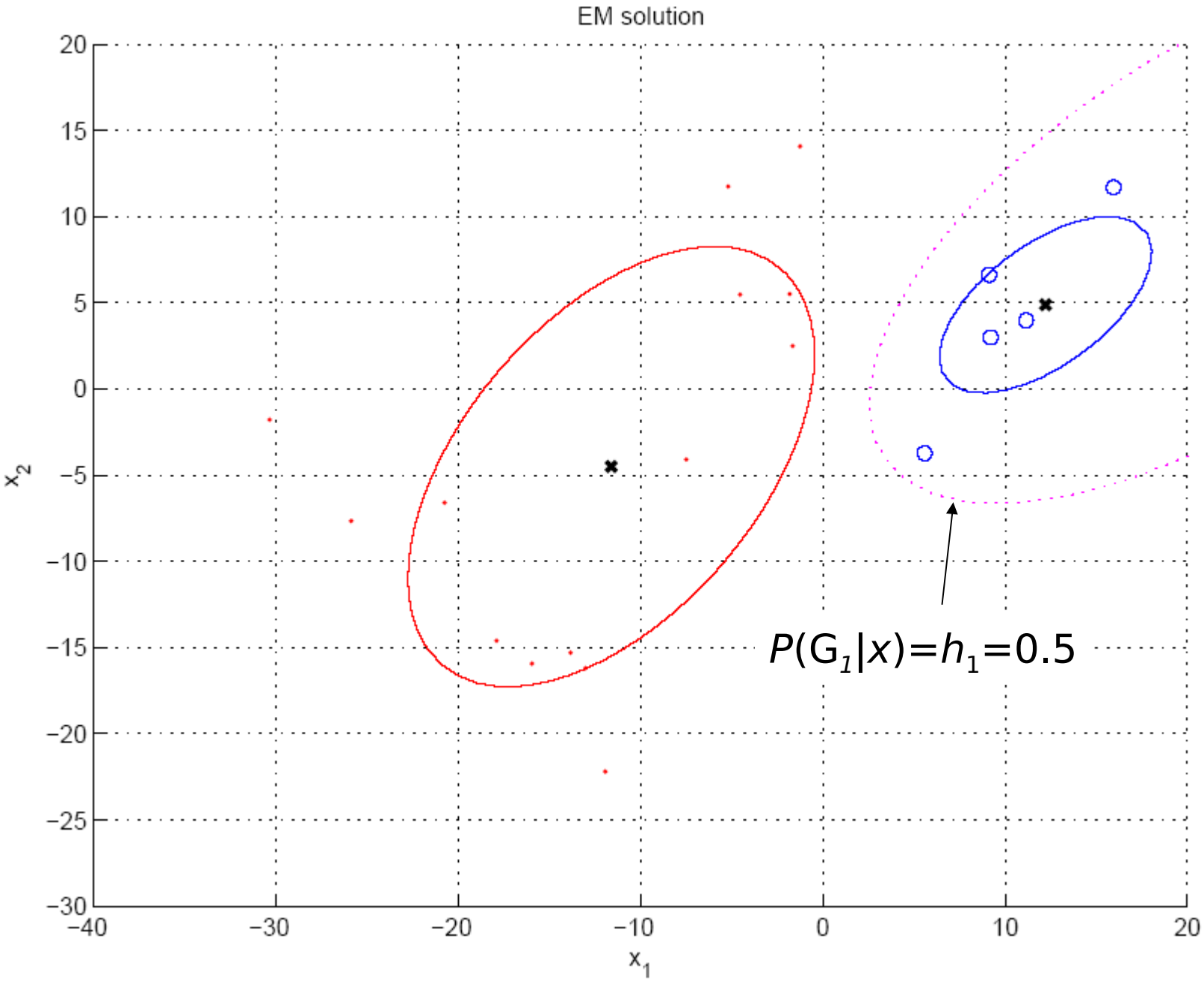
- Data came from mix of Gaussians $\hat{p}_i(\mathbf{x}^t|\Phi) \sim \mathcal{N}(\mathbf{m}_i, \mathbf{S}_i)$,
- Maximize likelihood assuming we know latent "indicator variable"

$$\mathbf{m}_i^{l+1} = \frac{\sum_t h_i^t \mathbf{x}^t}{\sum_t h_i^t}$$

$$\mathbf{S}_i^{l+1} = \frac{\sum_t h_i^t (\mathbf{x}^t - \mathbf{m}_i^{l+1})(\mathbf{x}^t - \mathbf{m}_i^{l+1})^T}{\sum_t h_i^t}$$

- E-step: expected value of indicator variables

$$h_i^t = \frac{\pi_i |\mathbf{S}_i|^{-1/2} \exp[-(1/2)(\mathbf{x}^t - \mathbf{m}_i)^T \mathbf{S}_i^{-1}(\mathbf{x}^t - \mathbf{m}_i)]}{\sum_j \pi_j |\mathbf{S}_j|^{-1/2} \exp[-(1/2)(\mathbf{x}^t - \mathbf{m}_j)^T \mathbf{S}_j^{-1}(\mathbf{x}^t - \mathbf{m}_j)]}$$


EM solution
20
15
10
5
0
−5
−10
−15
−20
−25
−30
−40
−30
−20
−10
0
10
20
x2
x1
P(G1|x)=h1=0.5

.1)

EM for Gaussian mixtures

- Assume all groups/clusters are Gaussians
- Multivariate Uncorrelated
- Same Variance
- Harden indicators
  - EM: expected values are between 0 and 1
  - K-means: 0 or 1
- Same as k-means

Dimensionality Reduction vs. Clustering

- Dimensionality reduction methods find correlations between features and group features
  - Age and Income are correlated

- Clustering methods find similarities between instances and group instances
  - Customer A and B are from the same cluster

Clustering: Usage for supervised learning

- Describe data in terms of cluster
  - Represent all data in cluster by cluster mean
  - Range of attributes

- Map data into new space(preprocessing)
  - d- dimension original space
  - k- number of clusters
  - Use indicator variables as data representations
  - k might be larger then d

Mixture of Mixtures

- In classification, the input comes from a mixture of classes (supervised).
- If each class is also a mixture, e.g., of Gaussians, (unsupervised), we have a mixture of mixtures:

$$p(\boldsymbol{x}|\mathrm{C}_i) = \sum_{j=1}^{k_i} p(\boldsymbol{x}|\mathrm{G}_{ij}) P(\mathrm{G}_{ij})$$

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} p(\boldsymbol{x}|\mathrm{C}_i) P(\mathrm{C}_i)$$

Hierarchical Clustering

- Probabilistic view
  - Fit mixture model to data
  - Find codewords minimizing reconstruction error

- Hierarchical clustering
  - Group similar items together
  - No specific model/distribution
  - Items in groups is more similar to each other than instances in different groups

Hierarchical Clustering

Minkowski (L_p) (Euclidean for $p = 2$)

$$d_m\left(\boldsymbol{x}^r, \boldsymbol{x}^s\right) = \left[\sum_{j=1}^{d}\left(x_j^r - x_j^s\right)^p\right]^{1/p}$$

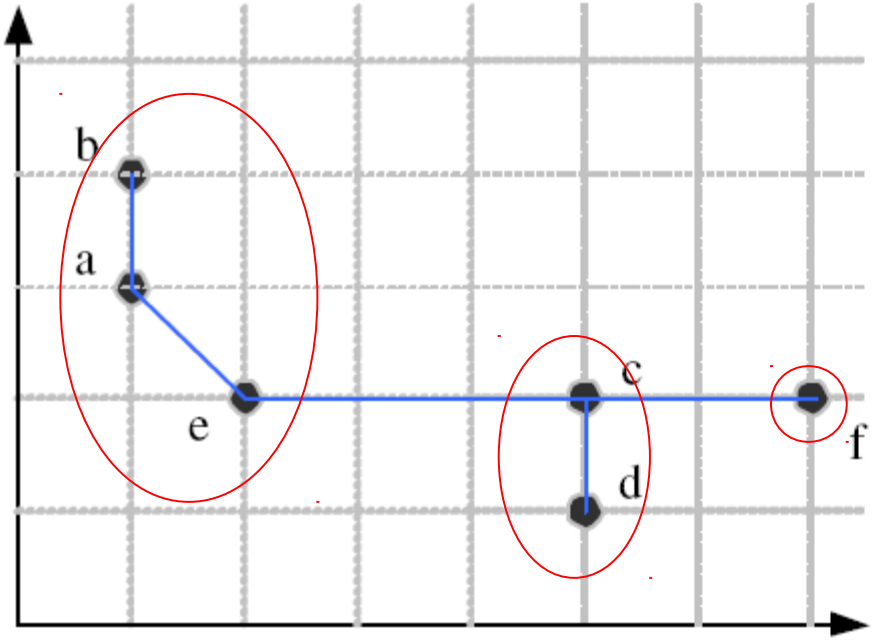
City-block distance

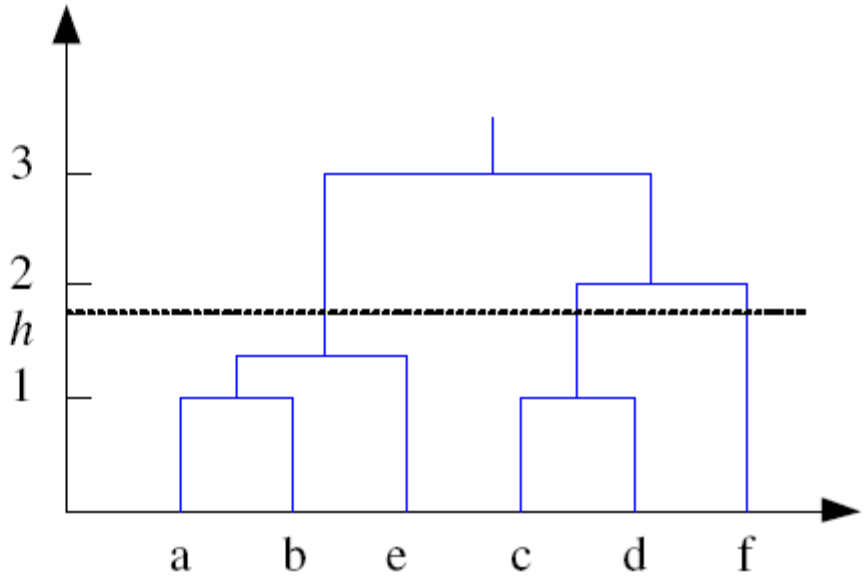
$$d_{cb}\left(\boldsymbol{x}^r, \boldsymbol{x}^s\right) = \sum_{j=1}^{d}\left|x_j^r - x_j^s\right|$$

Agglomerative Clustering

- Start with clusters each having single point
- At each step merge similar clusters
- Measure of similarity
  - Minimal distance(single link)
    - Distance between closest points in 2 groups
  - Maximal distance(complete link)
    - Distance between most distant points in 2 groups
  - Average distance
    - Distance between group centers

Example: Single-Link Clustering





Dendrogram

Choosing k

- Defined by the application, e.g., image quantization
- Plotting data in two dimensions using PCA

- Incremental (leader-cluster) algorithm: Add one at a time until “elbow” (reconstruction error/log likelihood/intergroup distances)