Intro to ML

October 25th, 2021

CHAPTER 6:

Dimensionality Reduction

Why Reduce Dimensionality?

- Reduces time complexity: Less computation
- Reduces space complexity: Fewer parameters
- Saves the cost of observing the feature
- Simpler models are more robust on small datasets
- More interpretable; simpler explanation
- Data visualization (structure, groups, outliers, etc) if plotted in 2 or 3 dimensions

Factor Analysis

- Find a small number of unobservable 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, $j = 1,\ldots,k$ are the latent factors with

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

Factors are:
Unit normal and uncorrelated

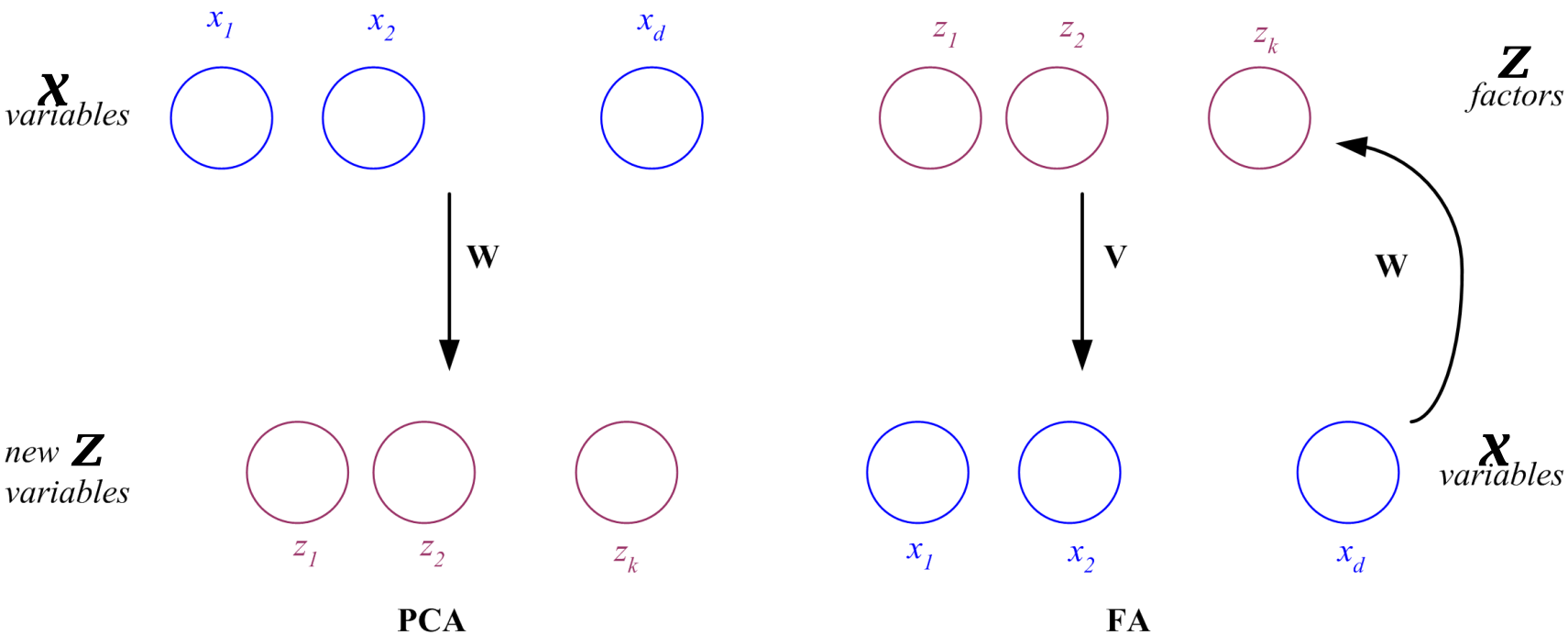
ε_i are the noise sources

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

and v_{ij} are the factor loadings

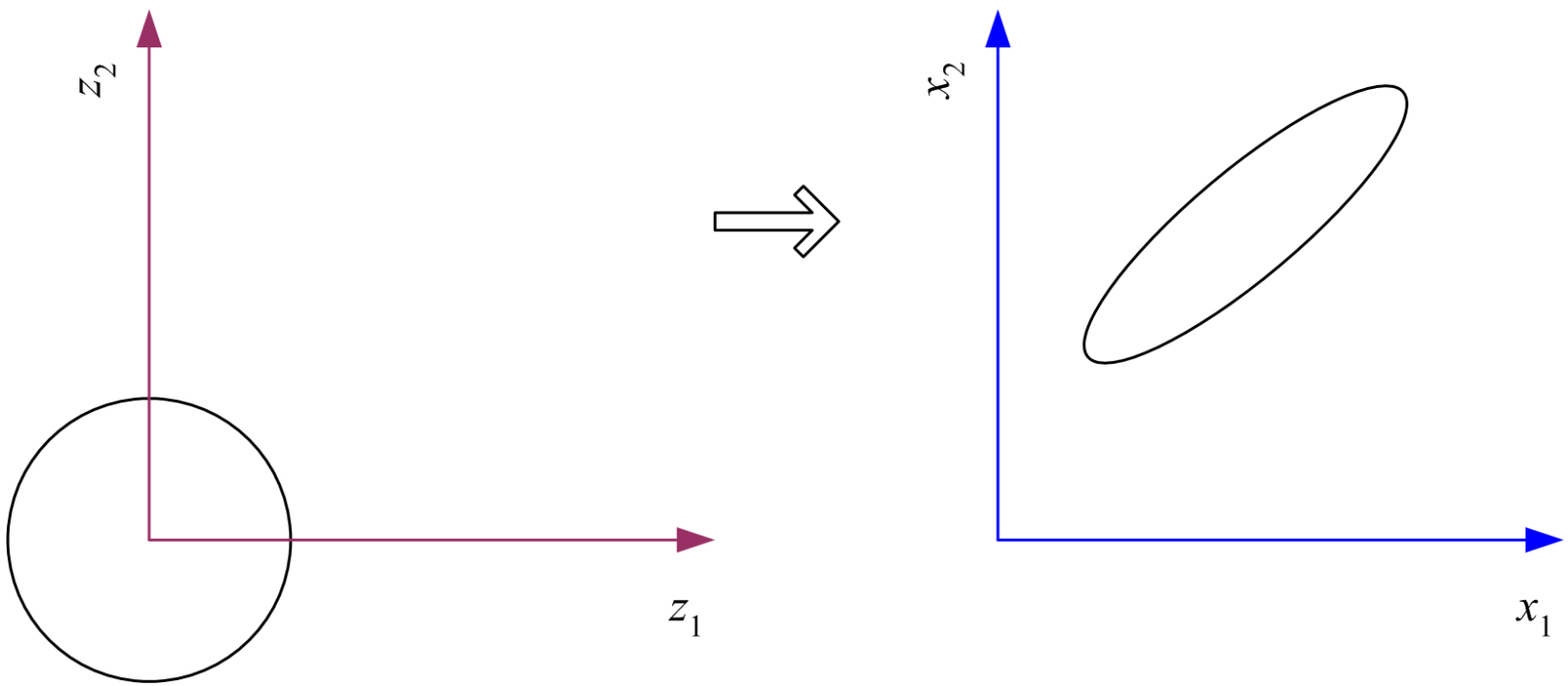
PCA vs FA

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



Factor Analysis

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



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

$Var(Xi) = V_{i1}\text{^}2 + V_{i2}\text{^}2 + \ldots \Psi_i$

- Variance of Xi attributed to common factor 1-k and noise

$$\sum = \text{cov}(X) = \text{cov}(Vz + \varepsilon)$$
$$= \text{cov}(Vz) + \text{cov}(\varepsilon)$$
$$= V\,\boxed{\text{cov}(z)}\,V^T + \Psi$$
$$= VV^T + \boxed{\Psi}$$

Identity matrix

Diagonal matrix

Assume there are two 'hidden factors', and two observable features

$\text{Cov}(x1,x2) = v_{11}v_{21} + v_{12}v_{22}$

- $Cov(x1,x2) = v_{11}v_{21} + v_{12}v_{22}$
- If x1 x2 have high covariance due to the first 'factor'
  - The $v_{11}v_{21}$ will be high or $v_{12}v_{22}$ will be high
  - Either way the above terms will be high
- If they depend on different factors, one term high one term low, this summation will be small

$Cov(x_1,z_2) = Cov(v_{12}z_{2,}\ z_2) = v_{12}$

- $Cov(x,z) = V$
- Loading represent correlation between variables and the factors

- S estimate $\sum$
- Factor analysis basically solves the following equation

$S=VV^T+ \Psi$

Linear Discriminant Analysis

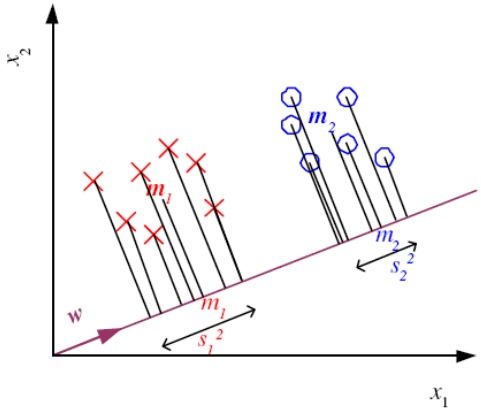
- Find a low-dimensional space s.t. when ***x*** is projected classes are as well separated as possible (supervised method)
  - $z=W^Tx$: projection of x onto w, a dimensional from d -> 1
- **m_1: original sample mean,** m_1: after projection
- **m_2: original sample mean,** m_2: after projection
- For a sample, $X=\{x^t,r^t)$, $r^t=1$ is x^t=class 1

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

After projection for m1, s1 (scatter)

- ind ***w*** that maximizes

LDA



- The goal is to let mean as far apart as possible and let the scatter for each class to be as clustered as possible
- $|m1-m2|^2$ large, s_1^2 + $s_2$2 as small

Fisher's discriminant, find w, such that

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

Is maximized

- Between-class scatter:

$$
\begin{aligned}
(m_1 - m_2)^2 &= \left(\mathbf{w}^T\mathbf{m}_1 - \mathbf{w}^T\mathbf{m}_2\right)^2 \\
&= \mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w} \\
&= \mathbf{w}^T\mathbf{S}_B\mathbf{w} \text{ where } \mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T
\end{aligned}
$$

- Within-class scatter:

$$
\begin{aligned}
s_1^2 &= \sum_t \left(\mathbf{w}^T\mathbf{x}^t - m_1\right)^2 r^t \\
&= \sum_t \mathbf{w}^T\left(\mathbf{x}^t - \mathbf{m}_1\right)\left(\mathbf{x}^t - \mathbf{m}_1\right)^T\mathbf{w}r^t = \mathbf{w}^T\mathbf{S}_1\mathbf{w}
\end{aligned}
$$

where $\mathbf{S}_1 = \sum_t \left(\mathbf{x}^t - \mathbf{m}_1\right)\left(\mathbf{x}^t - \mathbf{m}_1\right)^T r^t$

$$s_1^2 + s_1^2 = \mathbf{w}^T\mathbf{S}_W\mathbf{w} \text{ where } \mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$$

Fisher's Linear Discriminant

- Find ***w*** that max

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} = \frac{\left|\mathbf{w}^T (\mathbf{m}_1 - \mathbf{m}_2)\right|^2}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

- LDA soln: $\mathbf{w} = c \cdot \mathbf{S}_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$

Only direction matters, set c=1

- For more than 2 classes (k>2)

$z = W^T x$

- z is k-dimensional, W is dxk

K>2 Classes

For each class i, r_i^t is 1 is r_i^t belongs to class i

- Within-class scatter:

Total scatter

$$\mathbf{S}_W = \sum_{i=1}^{K} \mathbf{S}_i \qquad \mathbf{S}_i = \sum_t r_i^t \left(\mathbf{x}^t - \mathbf{m}_i\right)\left(\mathbf{x}^t - \mathbf{m}_i\right)^T$$

- Between-class scatter:

$$\mathbf{S}_B = \sum_{i=1}^{K} N_i \left(\mathbf{m}_i - \mathbf{m}\right)\left(\mathbf{m}_i - \mathbf{m}\right)^T \qquad \mathbf{m} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{m}_i$$

- Find W matrix that max $J(\mathbf{W}) = \dfrac{\left|\mathbf{W}^T \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_W \mathbf{W}\right|}$

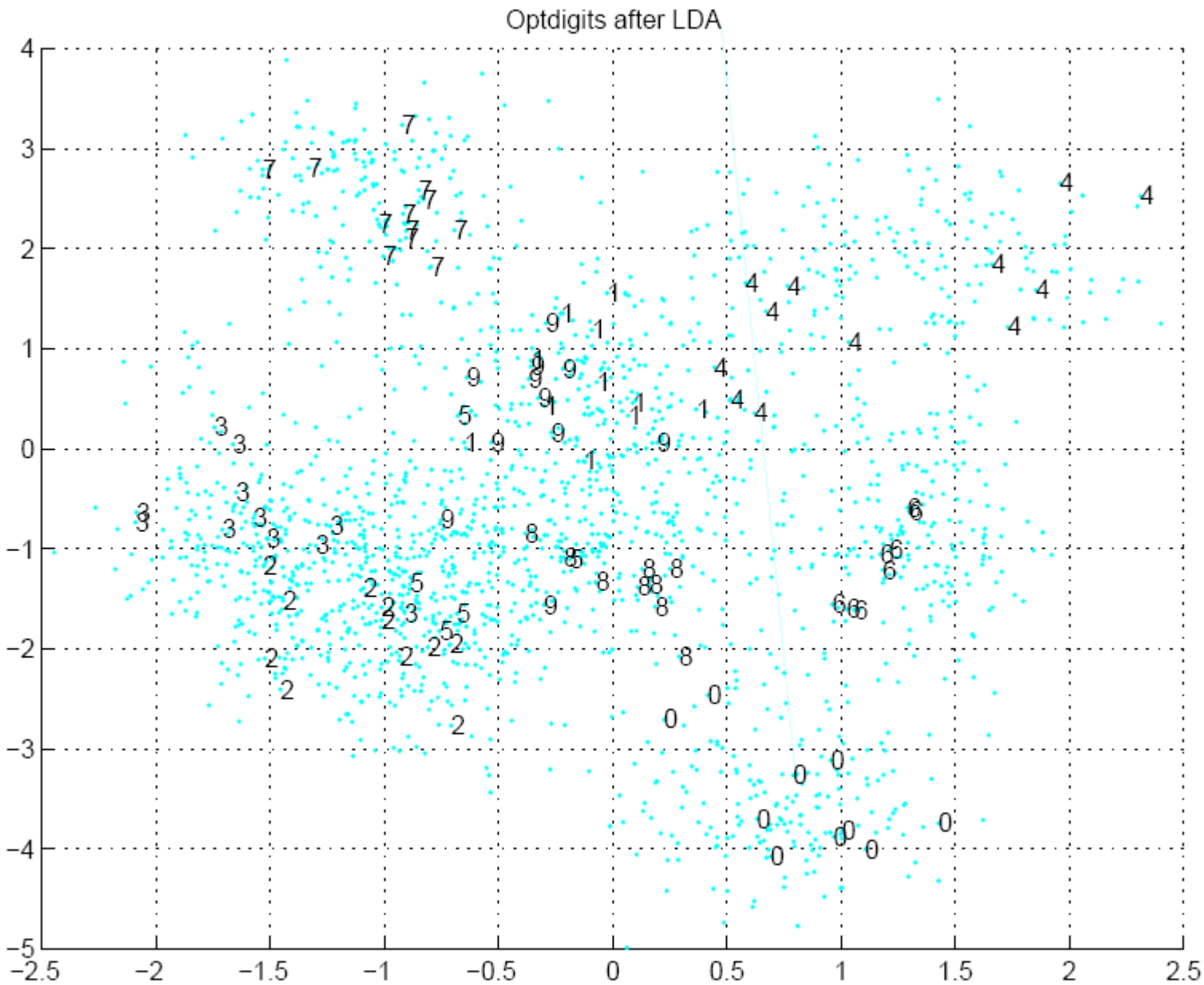
Want class means to be as far apart, and samples from the same class are close to their mean

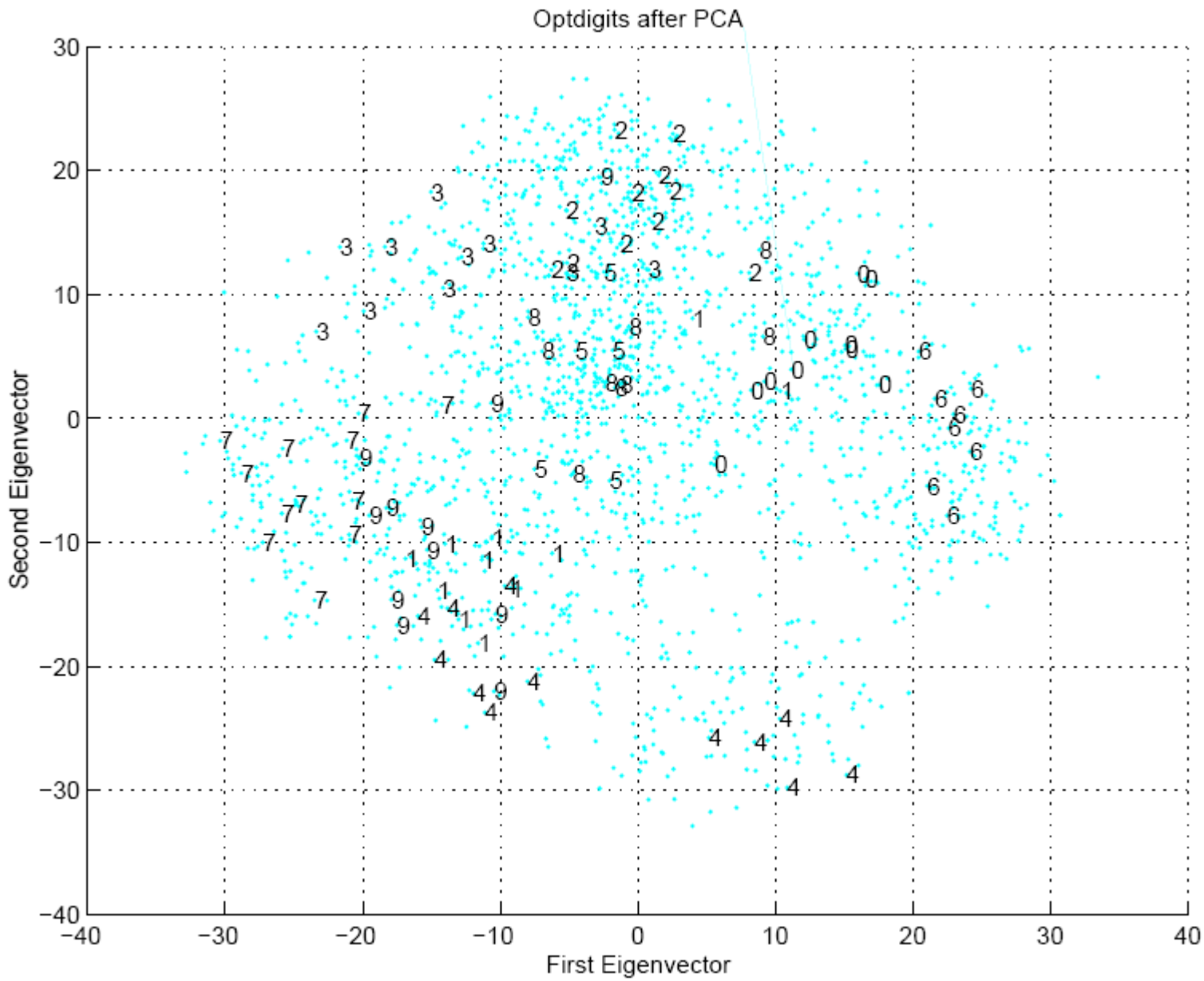
LDA (k>2)

- The largest eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$ is the solution
- S_B is the sum of K matrices
  - Hence, maximum rank of K-1 (only K-1 independent)
  - k is K-1

- LDA only works better since it makes use of the label information (do it in the training set)

- This solution is workable is S_w is invertible, if not, use PCA to get rid of singularity then LDA


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

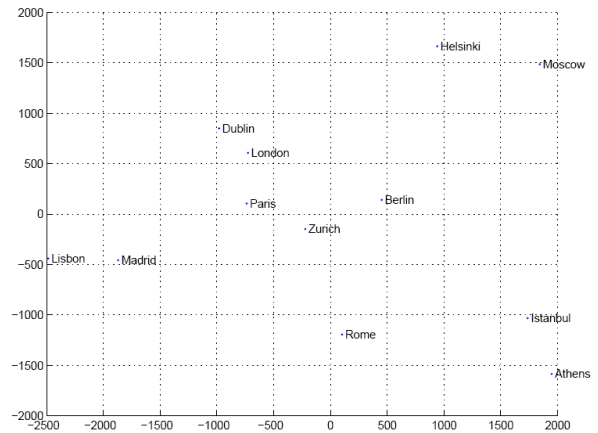


Visual display

Other important in the chapter

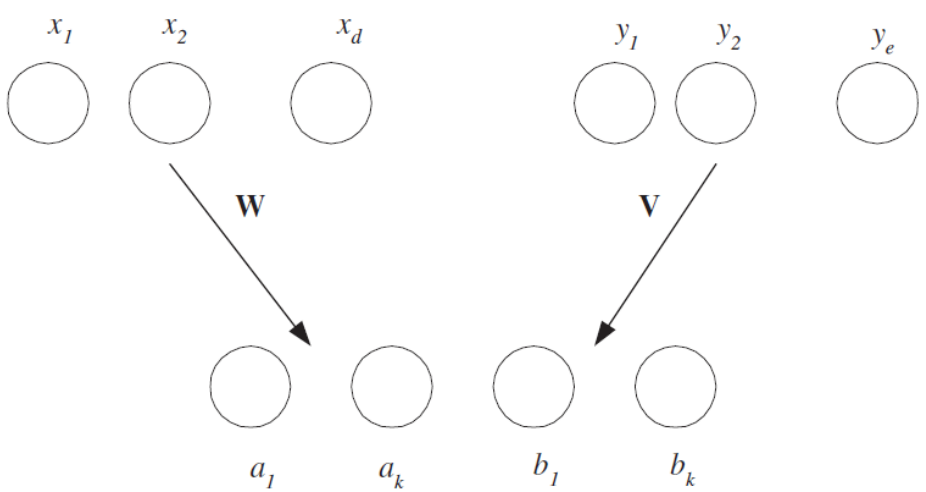
- Multidimensional scaling
  - Given pairwise distances between N points,

    d_{ij}, i,j =1,...,N place on a low-dim map s.t. distances are preserved
  - Good for visualization with proper distancing
- Canonical correlation
  - $X=\{x^t,y^t\}_t$; two sets of variables x and y x
  - We want to find two projections w and v st when x is projected along w and y is projected along v, the correlation is maximized
  - Work in cases where x, and y are different dimensions



Map of Europe by MDS
(from road travel distances)

CCA



CHAPTER 7:

Clustering

Semiparametric Density Estimation

- Parametric: Assume a single model for $p(\boldsymbol{x} \mid C_i)$ (Chapters 4 and 5)
- Semiparametric: $p(\boldsymbol{x} \mid C_i)$ is a mixture of densities

  Multiple possible explanations/prototypes:

  Different handwriting styles, accents in speech

  Not a single Gaussian distribution for the group of interest
- Nonparametric: No model; data speaks for itself (Chapter 8)

Mixture Densities

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

where G_i the components/groups/clusters,

$P(G_i)$ mixture proportions (priors),

$p(\boldsymbol{x} \mid G_i)$ component densities

Gaussian mixture where $p(\boldsymbol{x} \mid G_i) \sim N(\boldsymbol{\mu}_i, \sum_i)$ parameters $\Phi = \{P(G_i), \boldsymbol{\mu}_i, \sum_i\}_{i=1}^{k}$

unlabeled sample $X=\{\boldsymbol{x}^t\}_t$ (unsupervised learning)

Classes vs. Clusters

- Supervised: $X = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}_t$
- Classes C_i $i=1,\ldots,K$

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x} \mid C_i) P(C_i)$$

where $p(\boldsymbol{x} \mid C_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$

- $\Phi = \{P(C_i), \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^{K}$

$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N} \quad \mathbf{m}_i = \frac{\sum_t r_i^t \mathbf{x}^t}{\sum_t r_i^t}$$

$$\mathbf{S}_i = \frac{\sum_t r_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^T}{\sum_t r_i^t}$$

- Unsupervised : $X = \{\boldsymbol{x}^t\}_t$
- Clusters G_i $i=1,\ldots,k$

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

where $p(\boldsymbol{x} \mid G_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$

- $\Phi = \{P(G_i), \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^{k}$

Labels $\boldsymbol{r}^t_i$?

Through clustering

Imagine a case

- A image, 24 bits/pixel, ~ 16 million colors
- Say we have a screen with 8 bits/pixel
  - 256 colors only

- Find the best colors among 16 million colors so that the image look as close to the original image as possible -> color quantization

- Vector quantization ->continuous value to discrete space

k-Means Clustering

- Find k reference vectors (prototypes/codebook vectors/codewords) which best represent data
- Reference vectors, $\boldsymbol{m}_j$, $j = 1,\ldots,k$
- Use nearest (most similar) reference:

find a reference vector that is close to the x

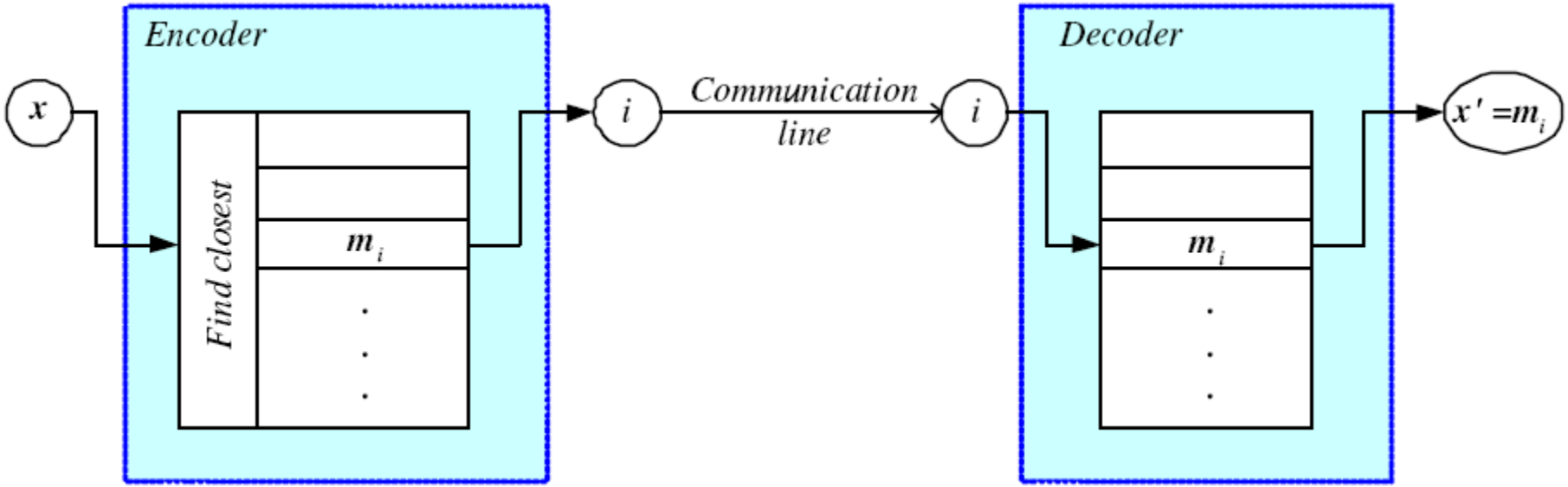
$$\left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\|$$

- Reconstruction error

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

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

Encoding/Decoding



Vector quantization saves bit, at the receiving end, there is an error

k-means Clustering

No analytical solution, since it's about finding that reference vector
Results in an iterative approach

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

Repeat

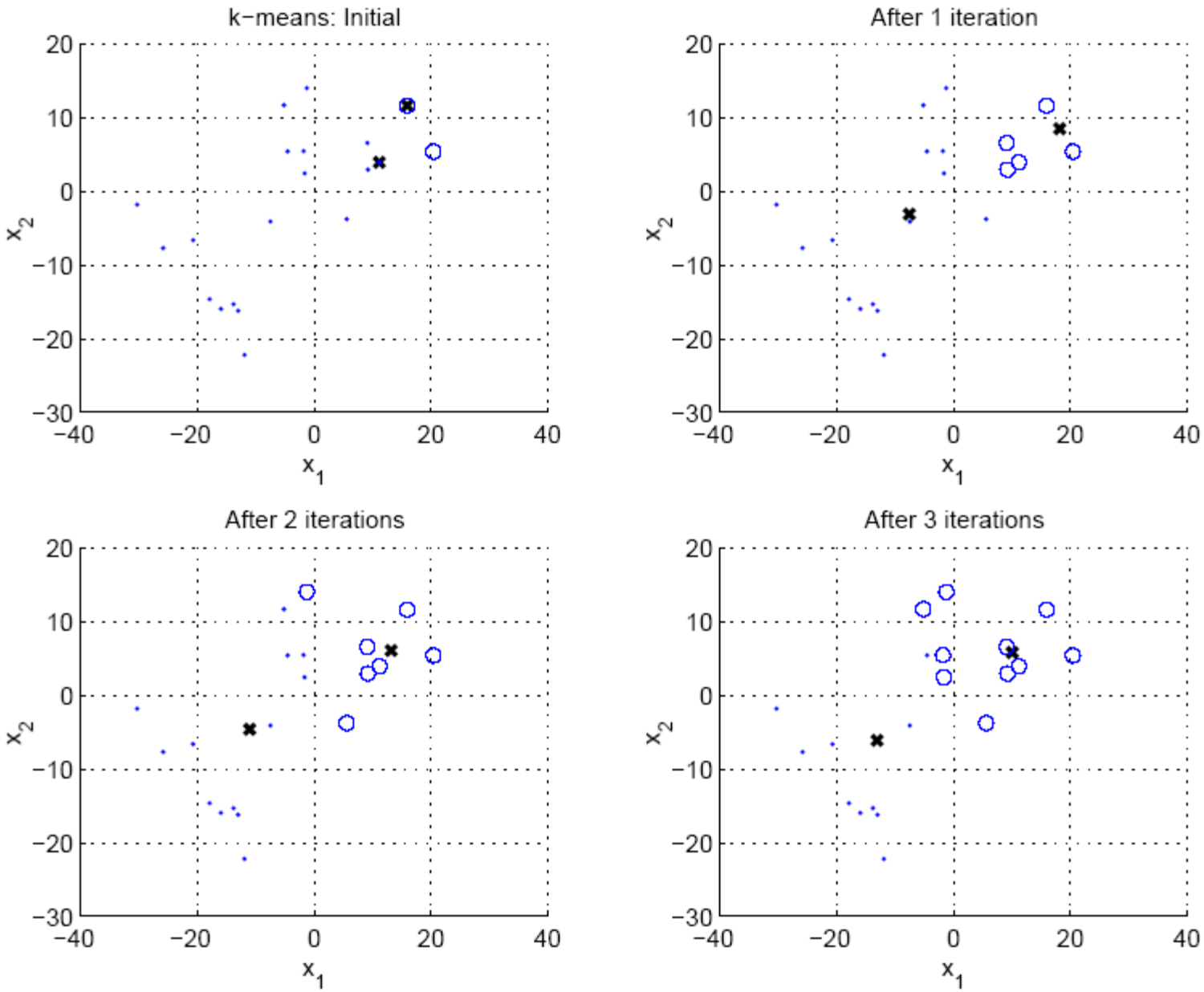
For all $\boldsymbol{x}^t \in \mathcal{X}$

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

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



K-means aim at finding codebook that minimizes reconstruction error

Expectation-Maximization (EM)

- Log likelihood with a mixture model

$$\mathcal{L}(\Phi \mid \mathcal{X}) = \log \prod_t p\left(\mathbf{x}^t \mid \Phi\right)$$

$$= \sum_t \log \sum_{i=1}^{k} p\left(\mathbf{x}^t \mid G_i\right) P\left(G_i\right)$$

Unknown, no analytical solution

EM Algorithm, core concept

- Assume there exist hidden variables *z*, which when known, make optimization much simpler
- Complete likelihood, $L_c(\Phi \mid X,Z)$, in terms of ***x*** and ***z***
- Incomplete likelihood, $L(\Phi \mid X)$, in terms of ***x***

E- and M-steps

Model parameter

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.

$$\text{E-step}: \mathcal{Q}\left(\Phi \mid \Phi^{l}\right) = E\left[\mathcal{L}_c\left(\Phi \mid \mathcal{X}, \mathcal{Z}\right) \mid \mathcal{X}, \Phi^{l}\right]$$

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

An increase in Q increases incomplete likelihood

$$\mathcal{L}\left(\Phi^{l+1} \mid \mathcal{X}\right) \geq \mathcal{L}\left(\Phi^{l} \mid \mathcal{X}\right)$$

There is proof beyond this class

EM in Gaussian Mixtures

Mixture of Guassian distribution

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

- $z^t_i = 1$ if $\boldsymbol{x}^t$ belongs to G_i, 0 otherwise; assume $p(\boldsymbol{x} \mid G_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$

- E-step
  - Expectation, find the expected value of the current model parameters under the current parameter set
- M-step
  - Maximize these parameters, and iterate

$$P(\mathbf{z}^t) = \prod_{i=1}^{k} \pi_i^{z_i^t} \tag{7.7}$$

The likelihood of an observation $\mathbf{x}^t$ is equal to its probability specified by the component that generated it:

$$p(\mathbf{x}^t|\mathbf{z}^t) = \prod_{i=1}^{k} p_i(\mathbf{x}^t)^{z_i^t} \tag{7.8}$$

$p_i(\mathbf{x}^t)$ is shorthand for $p(\mathbf{x}^t|\mathcal{G}_i)$. The joint density is

$$p(\mathbf{x}^t, \mathbf{z}^t) = P(\mathbf{z}^t)p(\mathbf{x}^t|\mathbf{z}^t)$$

and the complete data likelihood of the iid sample $\mathcal{X}$ is

$$\begin{aligned}
\mathcal{L}_c(\Phi|\mathcal{X}, \mathcal{Z}) &= \log \prod_t p(\mathbf{x}^t, \mathbf{z}^t|\Phi) \\
&= \sum_t \log p(\mathbf{x}^t, \mathbf{z}^t|\Phi) \\
&= \sum_t \log P(\mathbf{z}^t|\Phi) + \log p(\mathbf{x}^t|\mathbf{z}^t, \Phi) \\
&= \sum_t \sum_i z_i^t [\log \pi_i + \log p_i(\mathbf{x}^t|\Phi)]
\end{aligned}$$

$t \quad i$

E-step: We define

$$
\begin{aligned}
\mathcal{Q}(\Phi|\Phi^l) &\equiv E\left[\log P(X,Z)|\mathcal{X},\Phi^l\right] \\
&= E\left[\mathcal{L}_c(\Phi|\mathcal{X},\mathcal{Z})|\mathcal{X},\Phi^l\right] \\
&= \sum_t \sum_i E[z_i^t|\mathcal{X},\Phi^l][\log \pi_i + \log p_i(\boldsymbol{x}^t|\Phi)]
\end{aligned}
$$

where

$$
\begin{aligned}
E[z_i^t|\mathcal{X},\Phi^l] &= E[z_i^t|\boldsymbol{x}^t,\Phi^l] \quad \boldsymbol{x}^t \text{ are iid} \\
&= P(z_i^t = 1|\boldsymbol{x}^t,\Phi^l) \quad z_i^t \text{ is a 0/1 random variable} \\
&= \frac{p(\boldsymbol{x}^t|z_i^t = 1,\Phi^l)P(z_i^t = 1|\Phi^l)}{p(\boldsymbol{x}^t|\Phi^l)} \quad \text{Bayes' rule} \\
&= \frac{p_i(\boldsymbol{x}^t|\Phi^l)\pi_i^l}{\sum_j p_j(\boldsymbol{x}^t|\Phi^l)\pi_j^l} \\
&= \frac{p(\boldsymbol{x}^t|\mathcal{G}_i,\Phi^l)P(\mathcal{G}_i)}{\sum_j p(\boldsymbol{x}^t|\mathcal{G}_j,\Phi^l)P(\mathcal{G}_j)} \\
&= P(\mathcal{G}_i|\boldsymbol{x}^t,\Phi^l) \equiv h_i^t
\end{aligned}
\tag{7.9}
$$

- E-step:
  - Expected value is the posterior probability of the sample coming from that mixture (cluster)
  - It’s between 0-1
  - We can also think about it as soft assignment of cluster for each sample

  - In E step, given data X, compute the complete data likelihood given the current model parameters

M-step: We maximize $\mathcal{Q}$ to get the next set of parameter values Φ^{l+1}:

$$\Phi^{l+1} = \arg\max_{\Phi} \mathcal{Q}(\Phi|\Phi^l)$$

which is

$$\begin{aligned}\mathcal{Q}(\Phi|\Phi^l) &= \sum_t \sum_i h_i^t [\log \pi_i + \log p_i(\boldsymbol{x}^t|\Phi)] \\ &= \sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p_i(\boldsymbol{x}^t|\Phi)\end{aligned}$$

The second term is independent of π_i and using the constraint that $\sum_i \pi_i = 1$ as the Lagrangian, we solve for

$$\nabla_{\pi_i} \sum_t \sum_i h_i^t \log \pi_i - \lambda \left(\sum_i \pi_i - 1 \right) = 0$$

and get

$$\pi_i^{l+1} = \frac{\sum_t h_i^t}{N}$$

which is analogous to the calculation of priors in equation 7.2.

Similarly, the first term of equation 7.10 is independent of the components and can be dropped while estimating the parameters of the components. We solve for

$$\nabla_{\Phi} \sum_t \sum_i h_i^t \log p_i(\boldsymbol{x}^t|\Phi) = 0$$

Complete steps for GMM

- If assume Gaussian component (each mixture is a Gaussian distribution)

$$P(\mathcal{G}_i) = \frac{\sum_t h_i^t}{N} \qquad \mathbf{m}_i^{l+1} = \frac{\sum_t h_i^t \mathbf{x}^t}{\sum_t h_i^t}$$

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

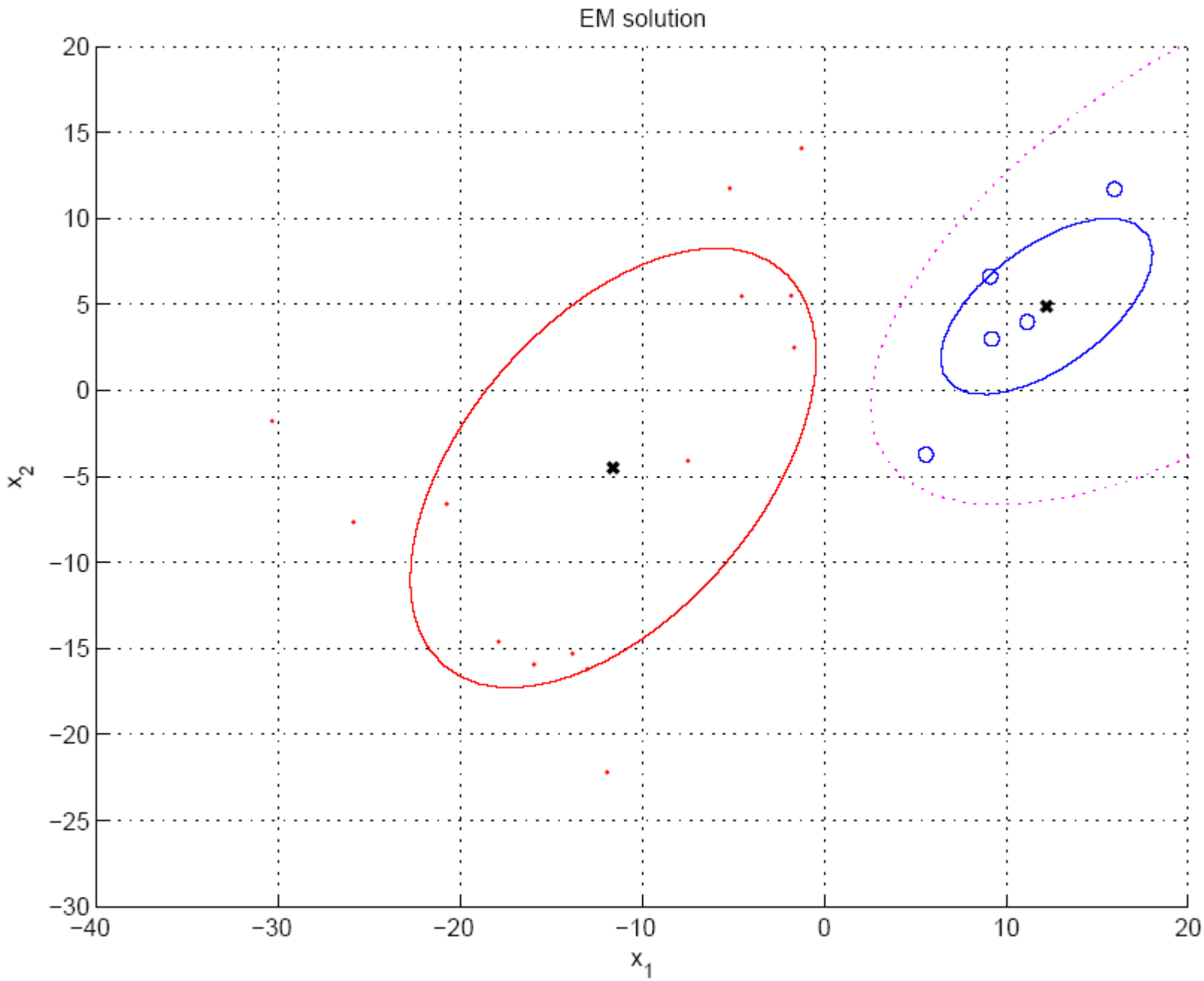
Soft assignment of a sample to a class

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

Practice implementation of GMM

- EM is initialized by k-means
- Once done, use m_i and samples associated with each cluster as to estimate the initial parameters used for mixture of Gaussian distributions

- Once done-learning, the GMM model can be used for ‘clustering’ of samples


EM solution
20
15
10
5
0
−5
−10
−15
−20
−25
−30
−40
−30
−20
−10
0
10
20
x_1
x_2

After Clustering

- Dimensionality reduction methods find correlations between features and group features
- Clustering methods find similarities between instances and group instances
- Allows knowledge extraction through

  number of clusters,

  prior probabilities,

  cluster parameters, i.e., center, range of features.

Example: CRM, customer segmentation