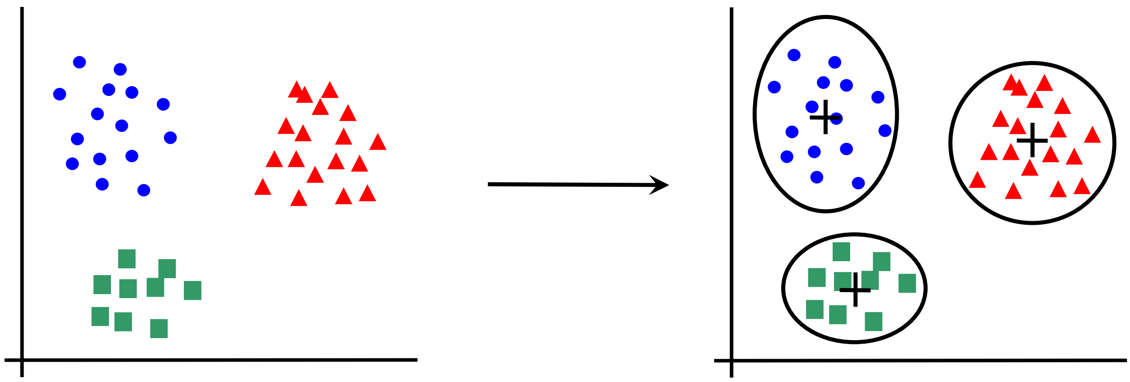
Lecture 8: Clustering & Mixture Models

C19 Machine Learning   Hilary 2013   A. Zisserman

- K-means algorithm
- GMM and the EM algorithm
- pLSA

- clustering

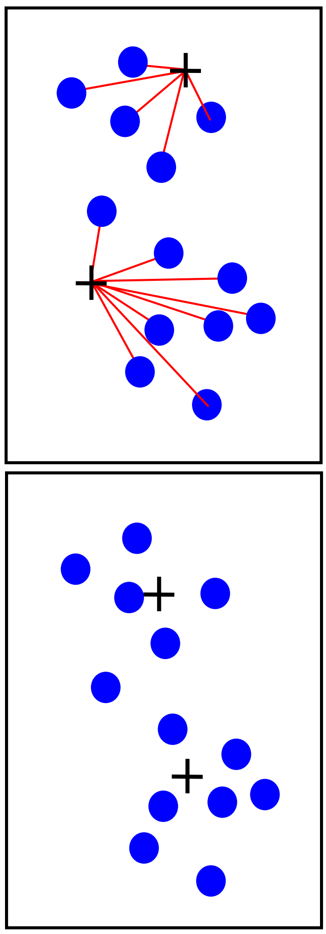


K-means algorithm

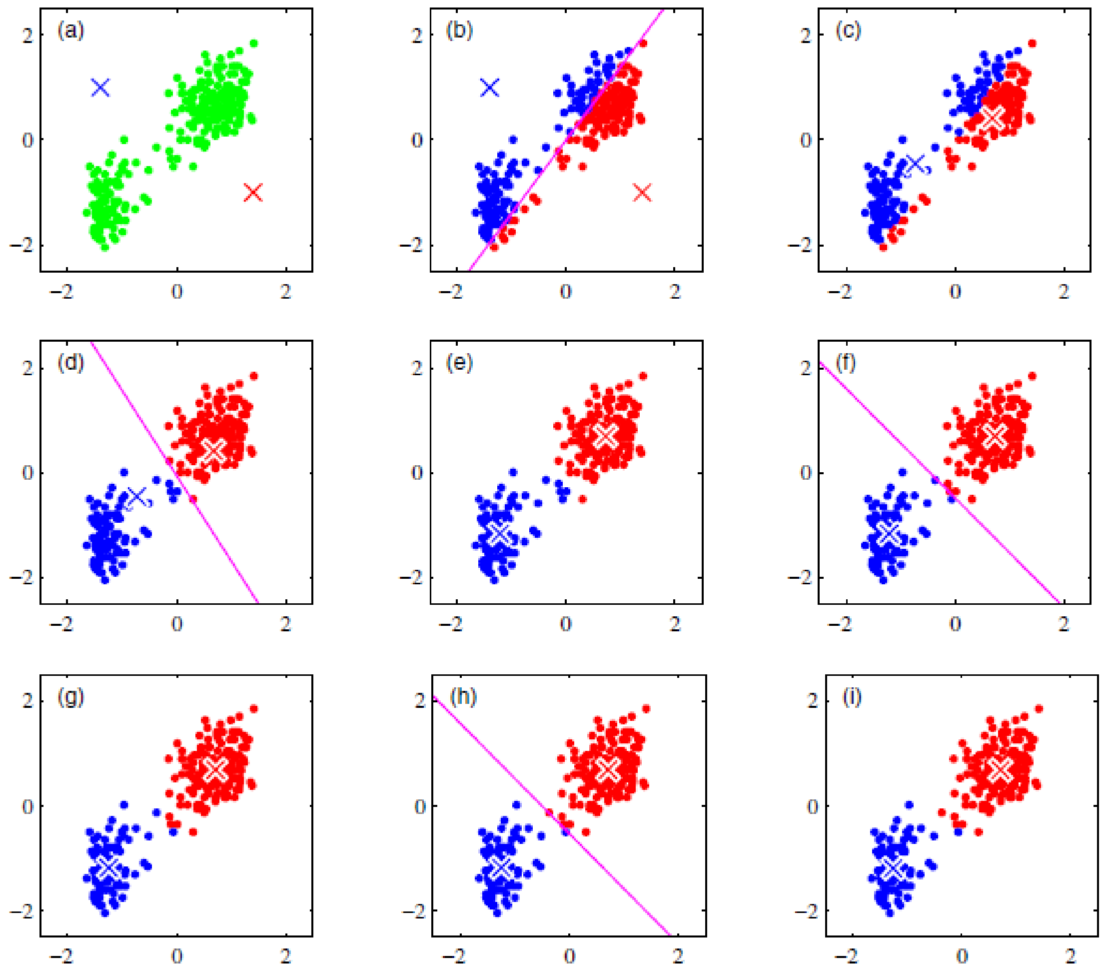
K-means algorithm

Partition data into K sets

- Initialize: choose K centres (at random)
- Repeat:
  1. Assign points to the nearest centre
  2. New centre = mean of points assigned to it
- Until no change



Example



Cost function

K-means minimizes a measure of distortion for a set of vectors $\{\mathbf{x}_i\}, i = 1, \ldots, N$

$$\mathtt{D} = \sum_{i=1}^{N} \|\mathbf{x}_i^k - \mathbf{c}_k\|^2$$

where $\mathbf{x}_i^k$ is the subset assigned to the cluster k. The objective is to find the set of centres $\{\mathbf{c}_k\}, k = 1, \ldots, K$ that minimize the distortion:

$$\min_{\mathbf{c}_k} \sum_{i=1}^{N} \|\mathbf{x}_i^k - \mathbf{c}_k\|^2$$

Introducing binary assignment variables r_{ik}, the distortion can be written as

$$\mathtt{D} = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2$$

where if $\mathbf{x}_i$ is assigned to cluster k then

$$r_{ij} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$$

Minimizing the Cost function

We want to determine

$$\min_{\mathbf{c}_k, r_{ik}} D = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2$$

Step 1: minimize over assignments r_{ik}

Each term in $\mathbf{x}_i$ can be minimized independently by assigning $\mathbf{x}_i$ to the closest centre $\mathbf{c}_k$

Step 2: minimize over centres $\mathbf{c}_k$

$$\frac{d}{d\mathbf{c}_k} \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2 = 2 \sum_{i=1}^{N} r_{ik} (\mathbf{x}_i - \mathbf{c}_k) = \mathbf{0}$$

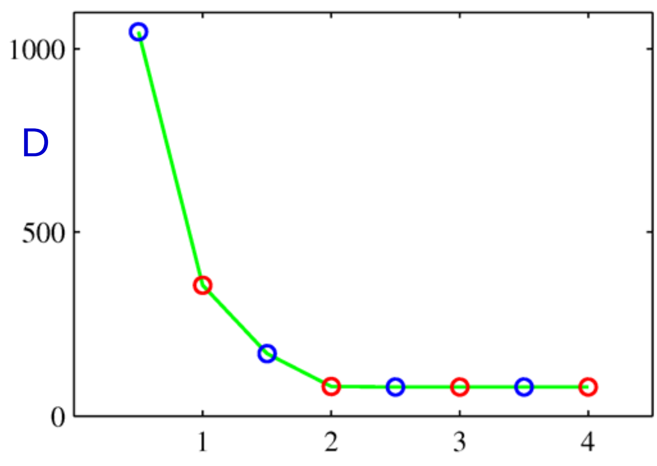
Hence

$$\mathbf{c}_k = \frac{\sum_{i=1}^{N} r_{ik} \mathbf{x}_i}{\sum_{i=1}^{N} r_{ik}}$$

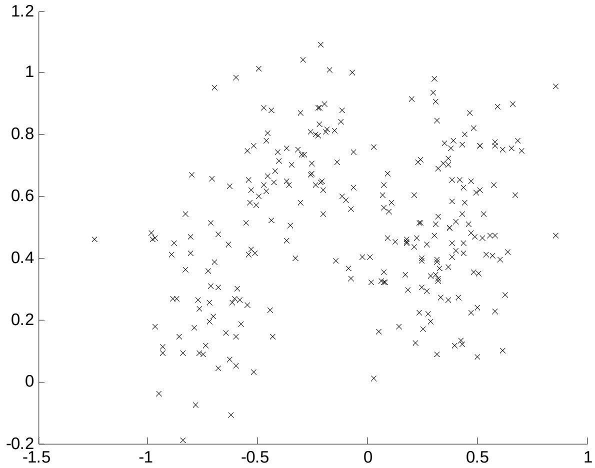
i.e. $\mathbf{c}_k$ is the mean (centroid) of the vectors assigned to it.

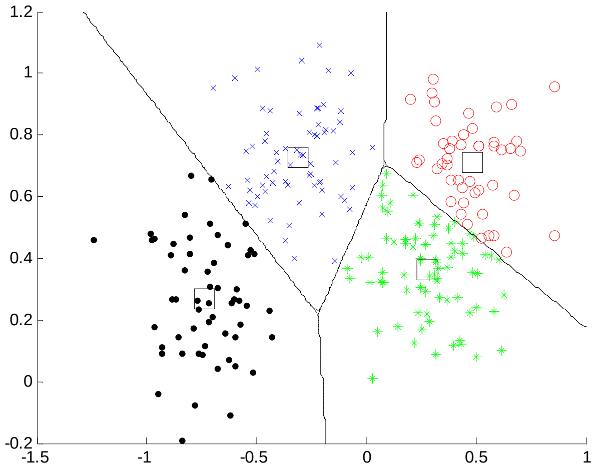
Note, since both steps decrease the cost D, the algorithm will converge – but it can converge to a local rather than global minimum.

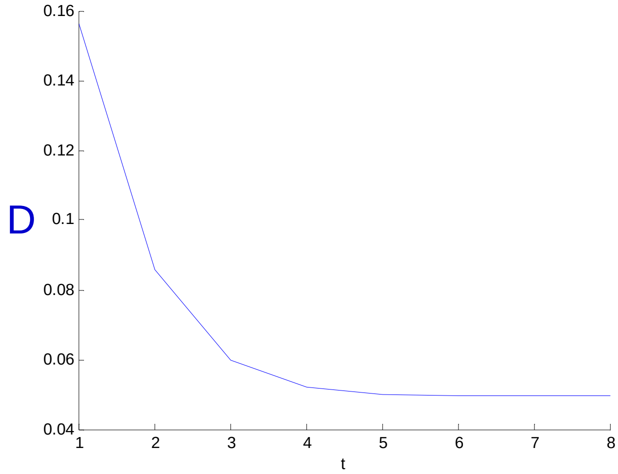
Decrease in distortion cost with iterations



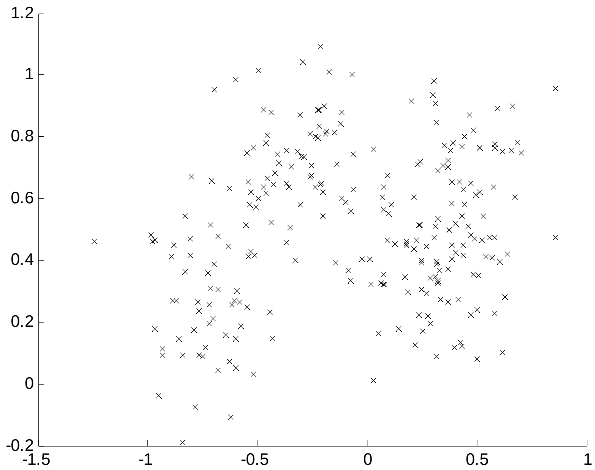
Sensitive to initialization

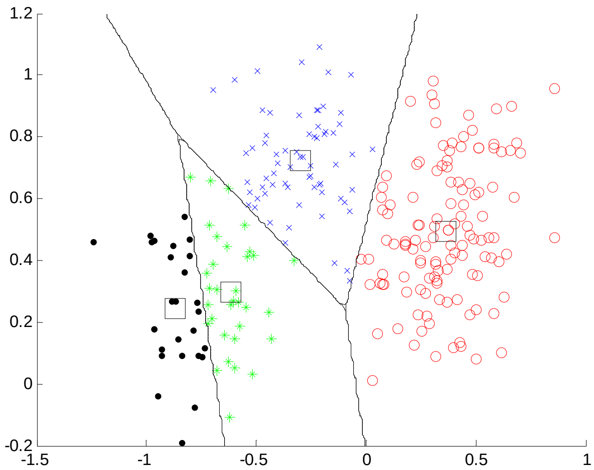


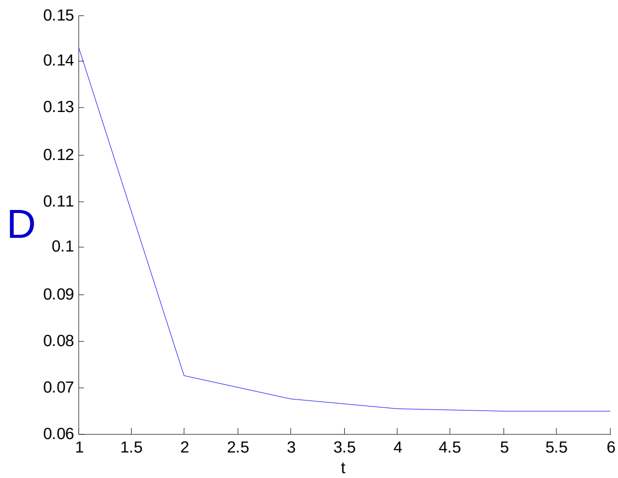




Sensitive to initialization







Practicalities

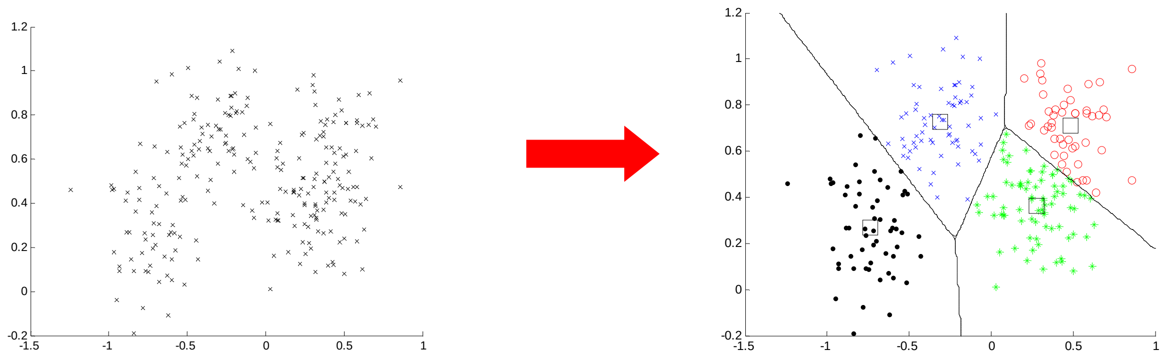
• always run algorithm several times with different initializations and keep the run with lowest cost

• choice of K

• suppose we have data for which a distance is defined, but it is non-vectorial (so can't be added). Which step needs to change?

• many other clustering methods: hierarchical K-means, K-medoids, agglomerative clustering …

Example application 1: vector quantization



• all vectors in a cluster are considered equivalent
• they can be represented by a single vector – the cluster centre
• applications in compression, segmentation, noise reduction

Example: image segmentation

- K-means cluster all pixels using their colour vectors (3D)
- assign pixels to their clusters
- colour pixels by their cluster assignment

$K = 2$ $K = 3$ $K = 10$ Original image

Example application 2: face clustering

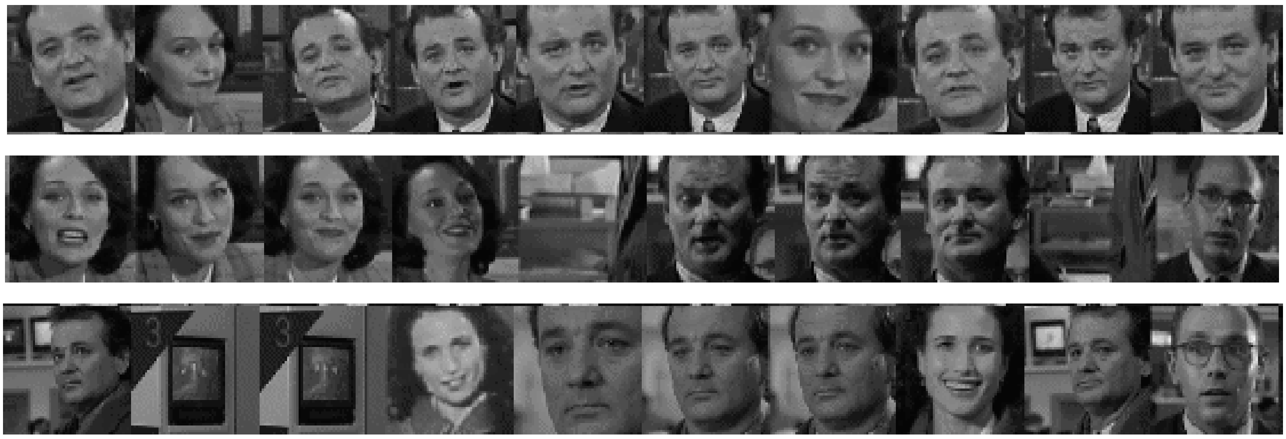
- Determine the principal cast of a feature film
- Approach: view this as a clustering problem on faces

Algorithm outline

1. Detect faces for every fifth frame in the movie
2. Describe the face by a vector of intensities
3. Cluster using a K-means algorithm

Example – "Ground Hog Day" 2000 frames

Subset of detected faces in temporal order



Clusters for K = 4

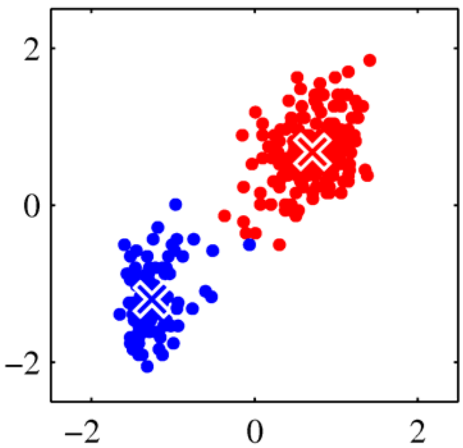
Gaussian Mixture Models

Hard vs soft assignments

• In K-means, there is a hard assignment of vectors to a cluster

• However, for vectors near the boundary this may be a poor representation

• Instead, can consider a soft-assignment, where the strength of the assignment depends on distance



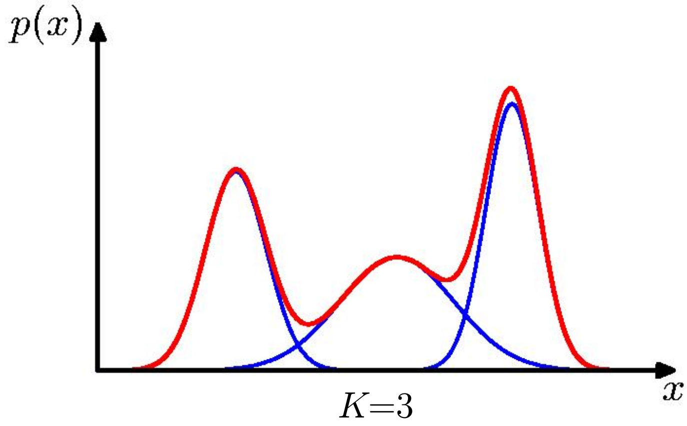
Gaussian Mixture Model (GMM)

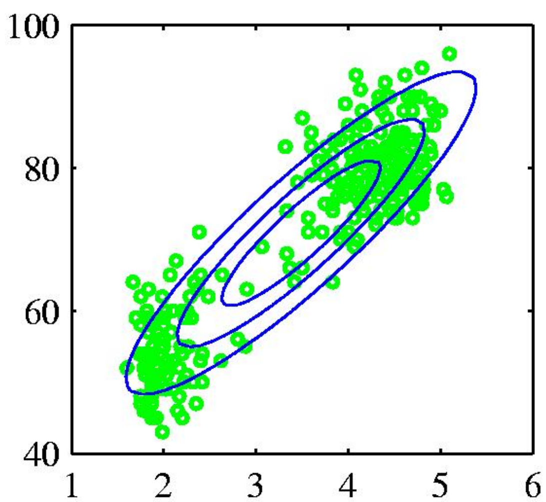
Combine simple models into a complex model:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \underbrace{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Component}}$$

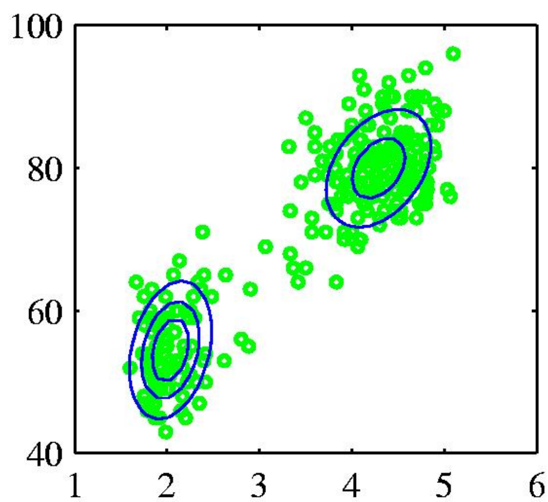
Mixing coefficient

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$





Single Gaussian



Mixture of two Gaussians

Cost function for fitting a GMM

For a point $\mathbf{x}_i$

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \Sigma_k)$$

The likelihood of the GMM for N points (assuming independence) is

$$\prod_{i=1}^{N} p(\mathbf{x}_i) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \Sigma_k)$$

and the (negative) log-likelihood is

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \Sigma_k)$$

where θ are the parameters we wish to estimate (i.e. $\boldsymbol{\mu}_k$ and Σ_k in this case).

To minimize $\mathcal{L}(\theta)$, differentiate first wrt $\boldsymbol{\mu}_k$

$$\frac{d\mathcal{L}(\theta)}{d\boldsymbol{\mu}_k} = \sum_{i=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_j, \Sigma_j)}}_{\gamma_{ik}} \Sigma_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)$$

Rearranging

$$\sum_{i=1}^{N} \gamma_{ik} \boldsymbol{\mu}_k = \sum_{i=1}^{N} \gamma_{ik} \mathbf{x}_i$$

and hence

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} \mathbf{x}_i$$ weighted mean

where

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_j, \Sigma_j)} \qquad N_k = \sum_{i=1}^{N} \gamma_{ik}$$

and γ_{ik} are the responsibilities of mixture component k for vector $\mathbf{x}_i$. N_k is the effective number of vectors assigned to component k.

γ_{ik} play a similar role to the assignment variables r_{ik} in K-means, but γ_{ik} is not binary, $0 \leq \gamma_{ik} \leq 1$

Differentiating wrt Σ_k gives

weighted covariance

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top$$

and wrt π_k (enforcing the constraint that $\sum_k \pi_k = 1$ with a Lagrange multiplier) gives

$$\pi_k = \frac{N_k}{N}$$

which is the average responsibility for the component

Now, … an algorithm for minimizing the cost function

Expectation Maximization (EM) Algorithm

Step 1 Expectation: Compute responsibilities using current parameters $\boldsymbol{\mu}_k, \Sigma_k$ (assignment)

$$\gamma_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_j, \Sigma_j)}$$

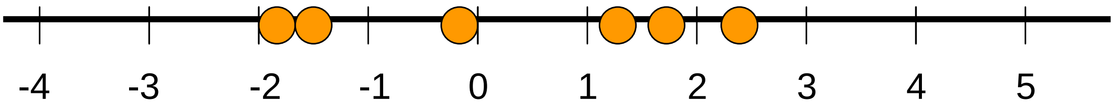
Step 2 Maximization: Re-estimate parameters using computed responsibilities

$$\begin{aligned}
\boldsymbol{\mu}_k &= \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} \mathbf{x}_i \\
\Sigma_k &= \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top \\
\pi_k &= \frac{N_k}{N} \quad \text{where } N_k = \sum_{i=1}^{N} \gamma_{ik}
\end{aligned}$$

Repeat until convergence

Example in 1D

Data: $x = (x_1, x_2, \ldots, x_N)$



OBJECTIVE: Fit mixture of Gaussian model with K=2 components

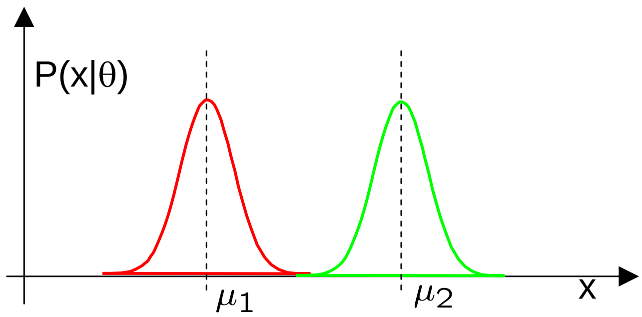
Model:

$$p(x_i|\theta) = \sum_k \pi_k \, N(x_i|\mu_k, \sigma_k) \quad \text{where} \quad \sum_{k=1}^{K} \pi_k = 1$$

Parameters: $\theta = \{\pi, \mu, \sigma\}$

keep π, σ fixed

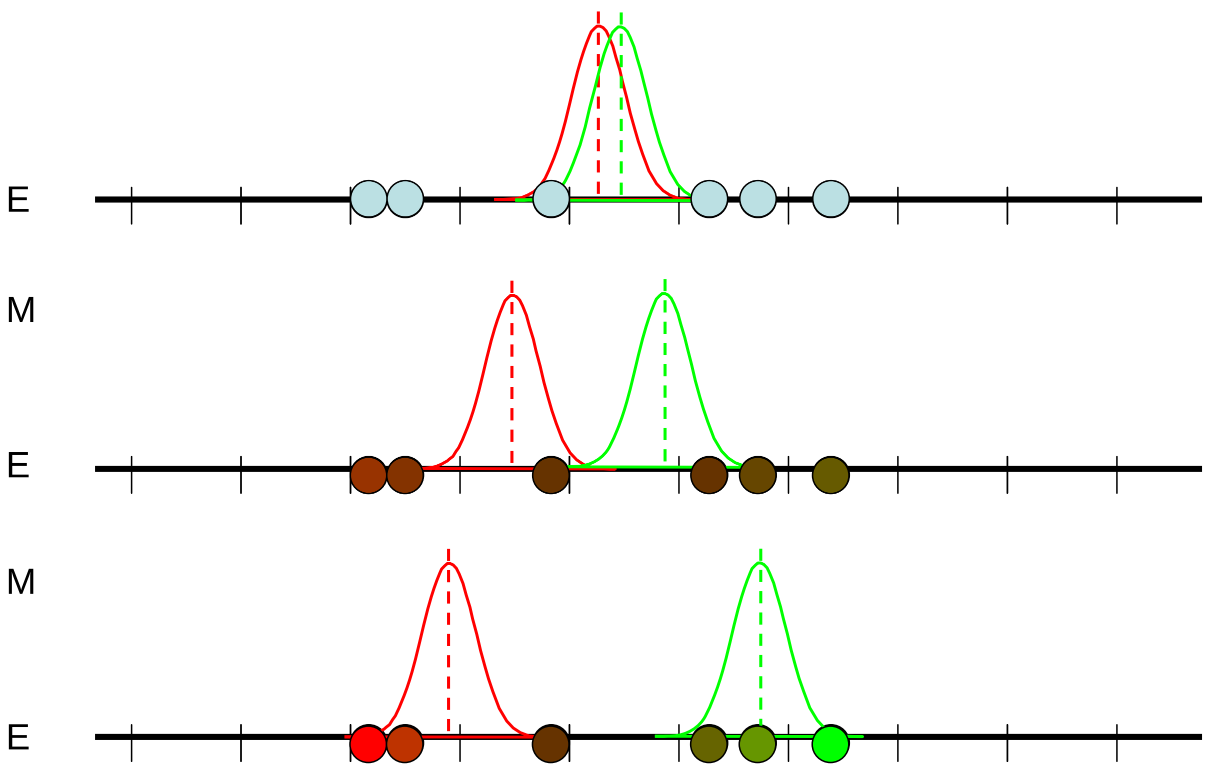
i.e. only estimate μ



Intuition of EM

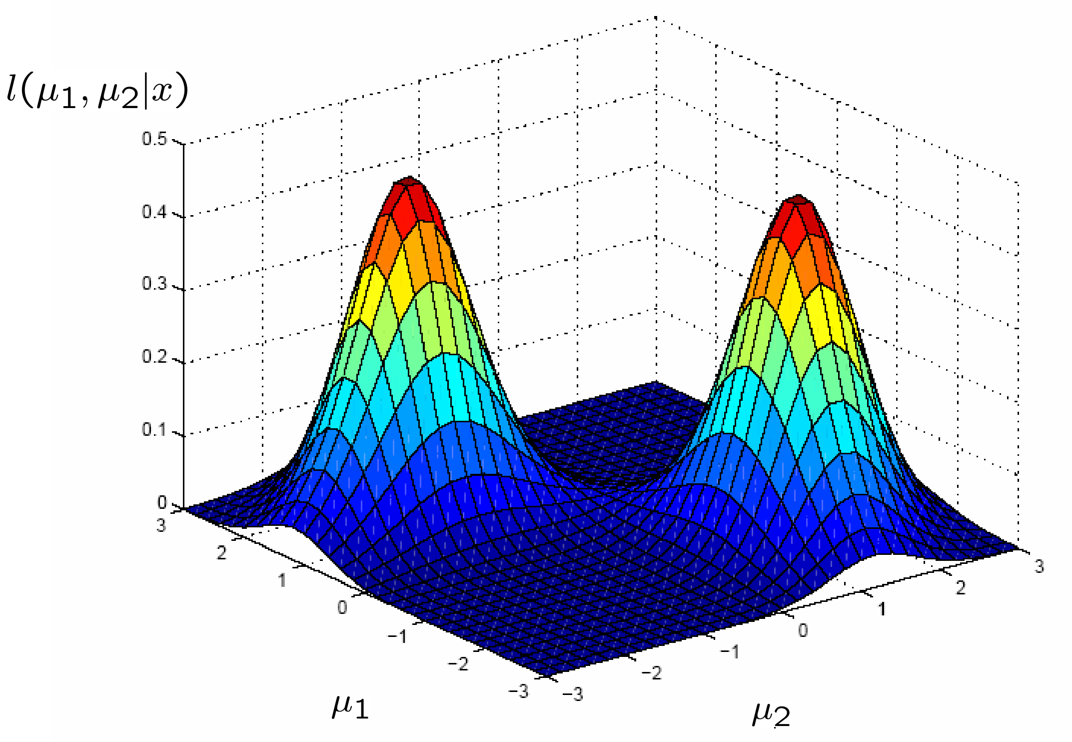
E-step: Compute soft assignment of the points, using current parameters

M-step: Update parameters using current responsibilities



Likelihood function

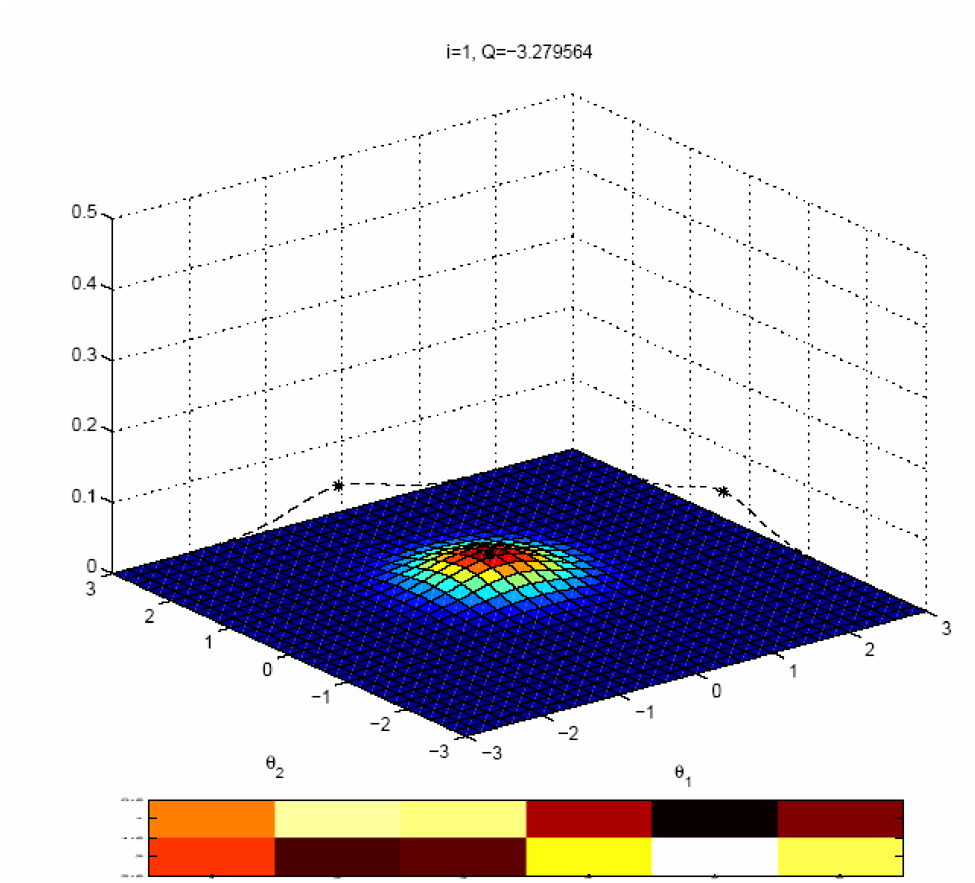
Likelihood is a function of parameters, θ
Probability is a function of r.v. x

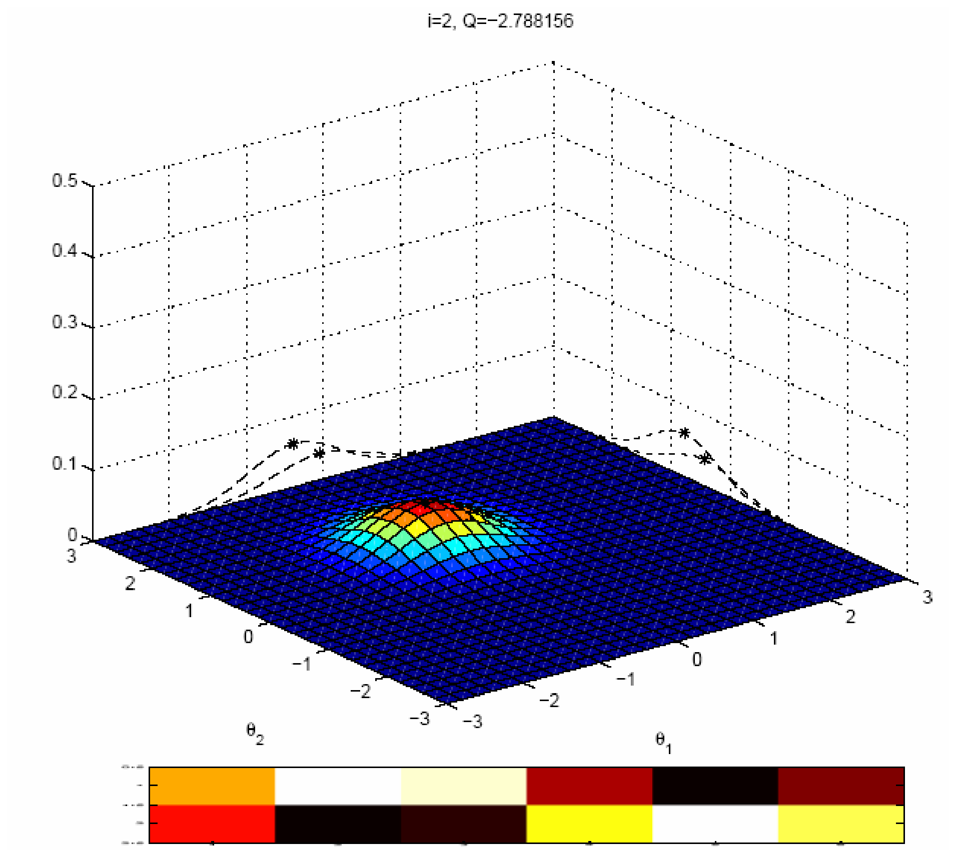


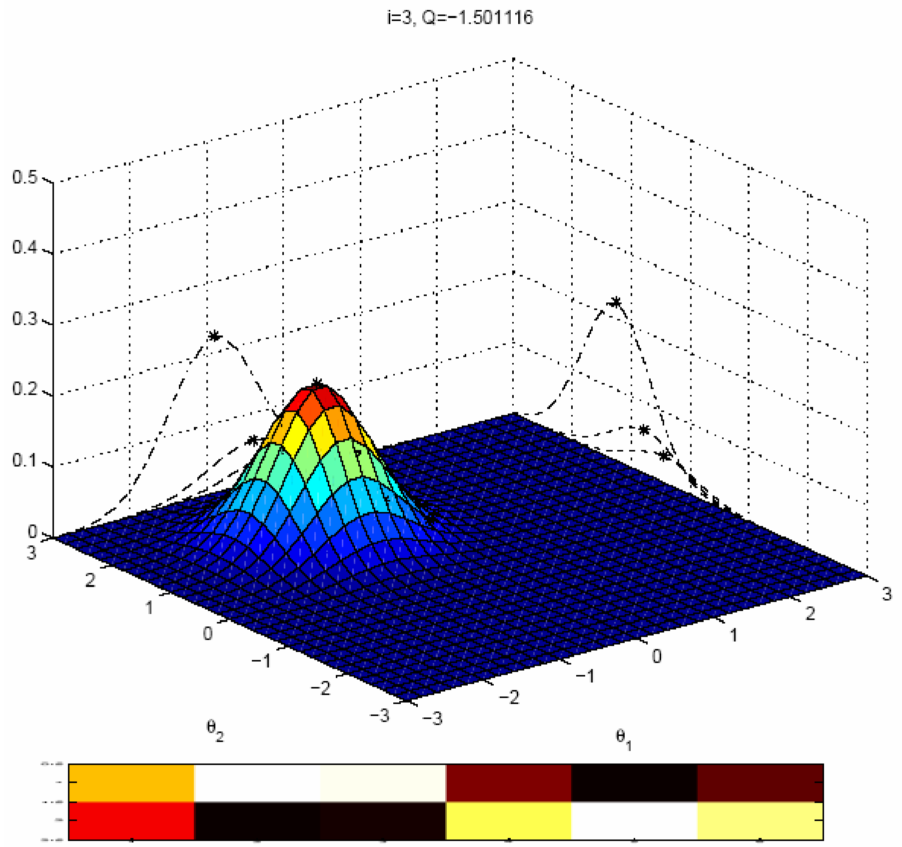
E-step: What do we actually compute?

nComponents x nPoints matrix (columns sum to 1):

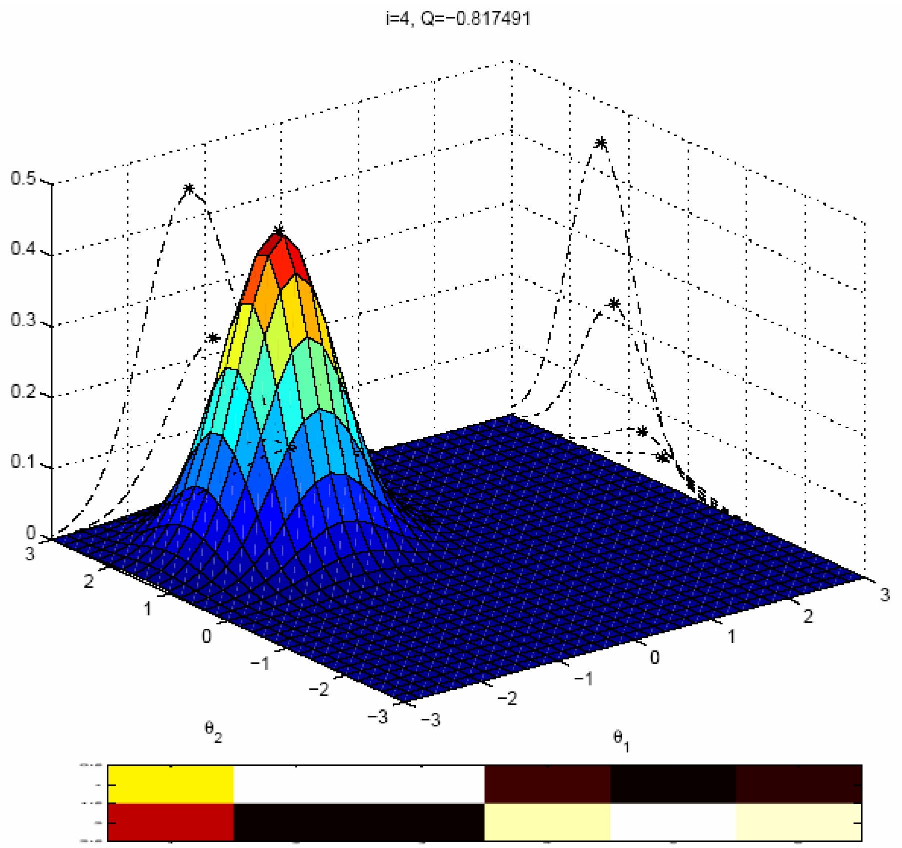
	Point 1	Point 2				Point 6
Component 1						
Component 2						


i=1, Q=−3.279564
0.5
0.4
0.3
0.2
0.1
0
3
2
1
0
−1
−2
−3
θ2
−3
−2
−1
0
1
2
3
θ1

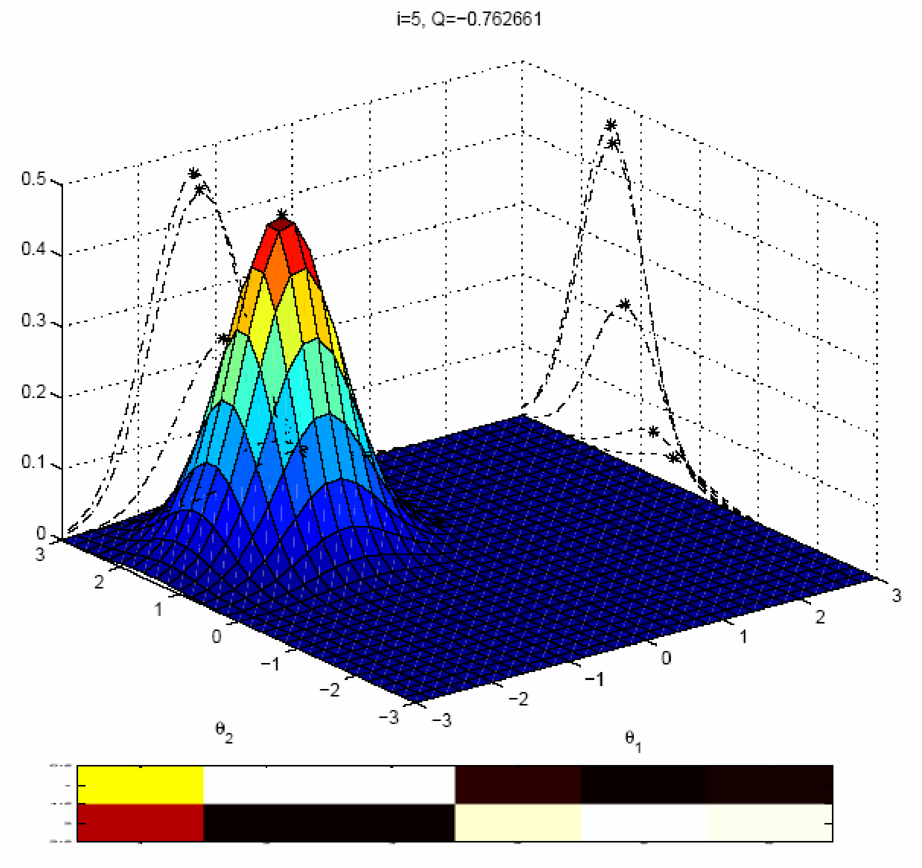

i=2, Q=−2.788156
0.5
0.4
0.3
0.2
0.1
0
3
2
1
0
−1
−2
−3
θ2
−3
−2
−1
0
1
2
3
θ1



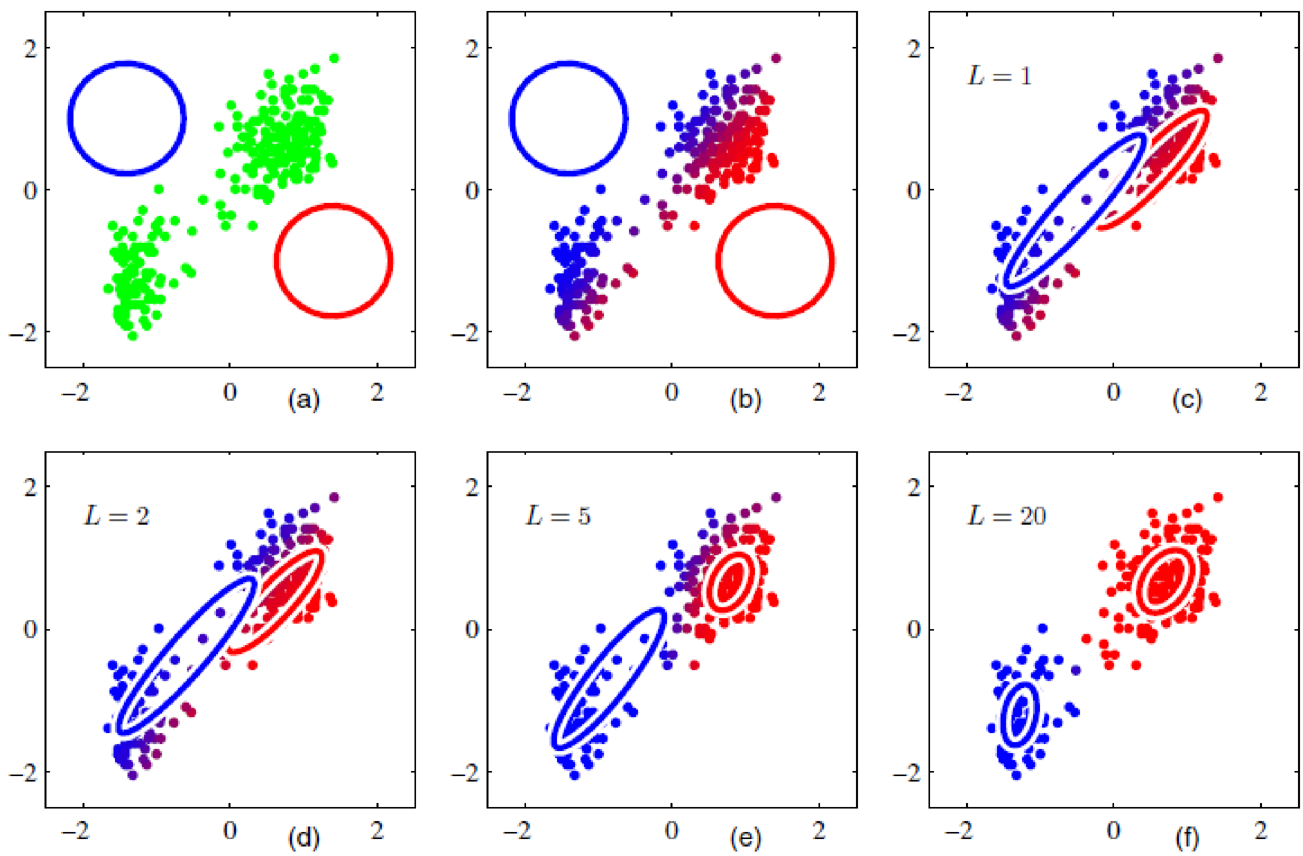
i=3, Q=−1.501116
0.5
0.4
0.3
0.2
0.1
0
3
2
1
0
−1
−2
−3
θ2
−3
−2
−1
0
1
2
3
θ1



i=4, Q=−0.817491
0.5
0.4
0.3
0.2
0.1
0
3
2
1
0
−1
−2
−3
θ2
−3
−2
−1
0
1
2
3
θ1



2D example: fitting means and covariances

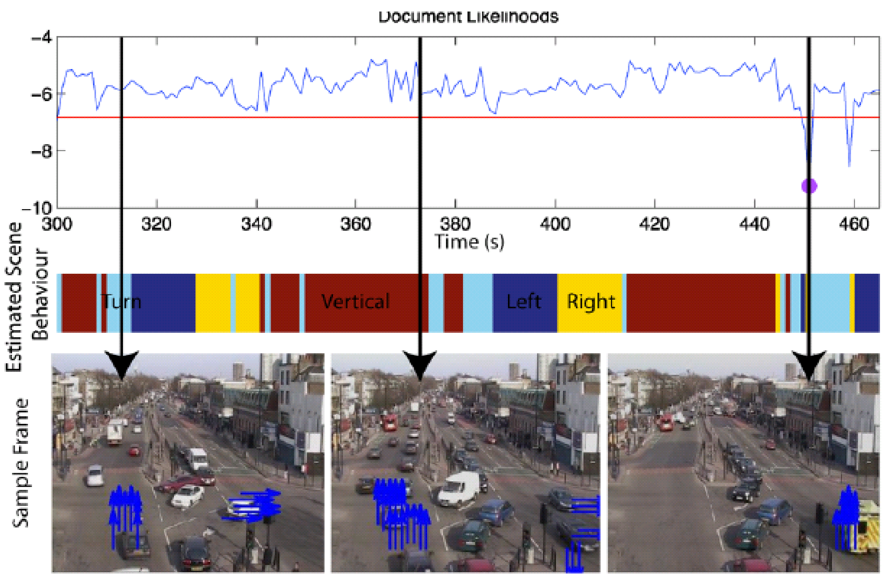


Practicalities

- Usually initialize EM algorithm using K-means
- Choice of K
- Can converge to a local rather than global minimum

Application: detecting unusual behaviour

- Model distribution of `usual' activities in a video of a street scene using a mixture model
- `Unusual' activities are then unlikely under the model



"Video Behaviour Mining Using a Dynamic Topic Model", Hospedales, Gong and Xiang, IJCV 2011

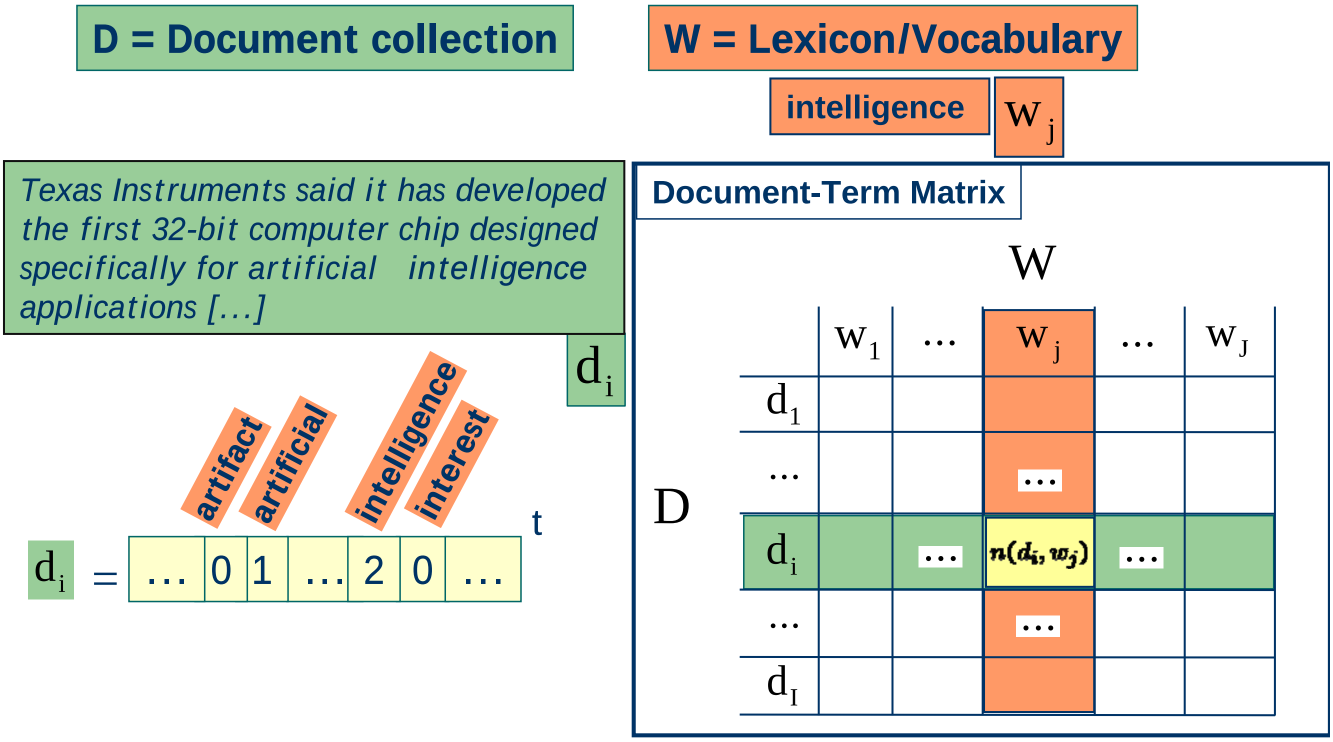
Probabilistic Latent Semantic Analysis (pLSA)

non-examinable

Unsupervised learning of topics

- Given a large collections of text documents (e.g. a website, or news archive)
- Discover the principal semantic topics in the collection
- Hence can retrieve/organize documents according to topic
- Method involves fitting a mixture model to a representation of the collection

Document-Term Matrix - bag of words model



Probabilistic Latent Semantic Analysis (pLSA)

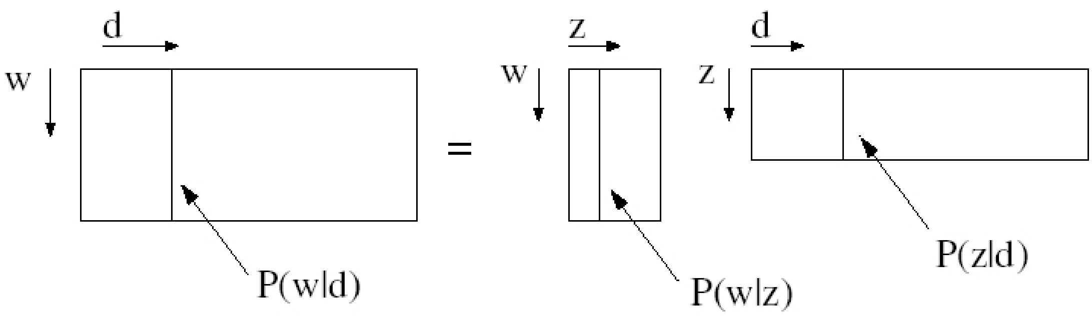
[Hofmann '99]

d … documents

w … words

z … topics

$$P(w_i|d_j) = \sum_{k=1}^{K} P(z_k|d_j)P(w_i|z_k)$$



Model fitting: find topic vectors P(w|z) common to all documents, and mixture coefficients P(z|d) specific to each document.

Probabilistic Latent Semantic Analysis (pLSA)

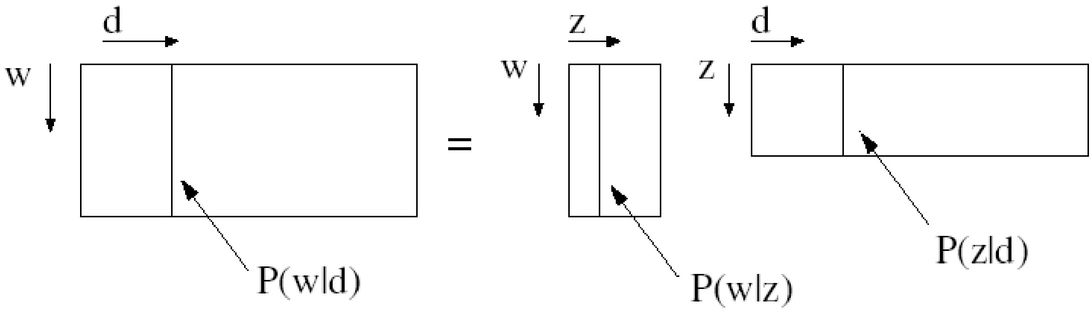
[Hofmann '99]

d … documents

w … words

z … topics

$$P(w_i|d_j) = \sum_{k=1}^{K} P(z_k|d_j)P(w_i|z_k)$$



- P(w|z) are the latent aspects
- Non-negative matrix factorization
- each document histogram explained as a sum over topics

Fitting pLSA parameters

Observed counts of word *i* in document *j*

$$L = \prod_{i=1}^{M} \prod_{j=1}^{N} P(w_i|d_j)^{n(w_i,d_j)}$$

$$\sum_{k=1}^{K} P(z_k|d_j)P(w_i|z_k)$$

Maximize likelihood of data using EM

M … number of words

N … number of documents

Expectation Maximization Algorithm for pLSA

E step: posterior probability of latent variables (“topics”)

$$P(z|d,w) = \frac{P(z|d;\pi)P(w|z;\theta)}{\sum_{z'} P(z'|d;\pi)P(w|z';\theta)}$$

Probability that the occurence of term w in document d can be “explained“ by topic z

M step: parameter estimation based on “completed” statistics

$$P(w|z;\theta) \propto \underbrace{\sum_d n(d,w)P(z|d,w)}_{\text{how often is term } w \text{ associated with topic } z\text{ ?}}, \qquad P(z|d;\pi) \propto \underbrace{\sum_w n(d,w)P(z|d,w)}_{\text{how often is document } d \text{ associated with topic } z\text{ ?}}$$

how often is term w associated with topic z ?

how often is document d associated with topic z ?

Example (1)

Topics (3 of 100) extracted from Associated Press news

Topic 1	
securities	94.96324
firm	88.74591
drexel	78.33697
investment	75.51504
bonds	64.23486
sec	61.89292
bond	61.39895
junk	61.14784
milken	58.72266
firms	51.26381
investors	48.80564
lynch	44.91865
insider	44.88536
shearson	43.82692
boesky	43.74837
lambert	40.77679
merrill	40.14225
brokerage	39.66526
corporate	37.94985
burnham	36.86570

Topic 2	
ship	109.41212
coast	93.70902
guard	82.11109
sea	77.45868
boat	75.97172
fishing	65.41328
vessel	64.25243
tanker	62.55056
spill	60.21822
exxon	58.35260
boats	54.92072
waters	53.55938
valdez	51.53405
alaska	48.63269
ships	46.95736
port	46.56804
hazelwood	44.81608
vessels	43.80310
ferry	42.79100
fishermen	41.65175

Topic 3	
india	91.74842
singh	50.34063
militants	49.21986
gandhi	48.86809
sikh	47.12099
indian	44.29306
peru	43.00298
hindu	42.79652
lima	41.87559
kashmir	40.01138
tamilnadu	39.54702
killed	39.47202
india's	39.25983
punjab	39.22486
delhi	38.70990
temple	38.38197
shining	37.62768
menem	35.42235
hindus	34.88001
violence	33.87917

Example (2)

Topics (10 of 128) extracted from Science Magazine articles (12K)

P(w|z)

universe	0.0439	drug	0.0672	cells	0.0675	sequence	0.0818	years	0.156
galaxies	0.0375	patients	0.0493	stem	0.0478	sequences	0.0493	million	0.0556
clusters	0.0279	drugs	0.0444	human	0.0421	genome	0.033	ago	0.045
matter	0.0233	clinical	0.0346	cell	0.0309	dna	0.0257	time	0.0317
galaxy	0.0232	treatment	0.028	gene	0.025	sequencing	0.0172	age	0.0243
cluster	0.0214	trials	0.0277	tissue	0.0185	map	0.0123	year	0.024
cosmic	0.0137	therapy	0.0213	cloning	0.0169	genes	0.0122	record	0.0238
dark	0.0131	trial	0.0164	transfer	0.0155	chromosome	0.0119	early	0.0233
light	0.0109	disease	0.0157	blood	0.0113	regions	0.0119	billion	0.0177
density	0.01	medical	0.00997	embryos	0.0111	human	0.0111	history	0.0148

P(w|z)

bacteria	0.0983	male	0.0558	theory	0.0811	immune	0.0909	stars	0.0524
bacterial	0.0561	females	0.0541	physics	0.0782	response	0.0375	star	0.0458
resistance	0.0431	female	0.0529	physicists	0.0146	system	0.0358	astrophys	0.0237
coli	0.0381	males	0.0477	einstein	0.0142	responses	0.0322	mass	0.021
strains	0.025	sex	0.0339	university	0.013	antigen	0.0263	disk	0.0173
microbiol	0.0214	reproductive	0.0172	gravity	0.013	antigens	0.0184	black	0.0161
microbial	0.0196	offspring	0.0168	black	0.0127	immunity	0.0176	gas	0.0149
strain	0.0165	sexual	0.0166	theories	0.01	immunology	0.0145	stellar	0.0127
salmonella	0.0163	reproduction	0.0143	aps	0.00987	antibody	0.014	astron	0.0125
resistant	0.0145	eggs	0.0138	matter	0.00954	autoimmune	0.0128	hole	0.00824

There is more …

- Bishop, chapter 9.1 – 9.3.2
- Other topics you should know about:
  - random forest regressors
  - Kernel PCA
  - semi-supervised learning
  - active learning
  - collaborative filtering
- More on web page:
  http://www.robots.ox.ac.uk/~az/lectures/ml