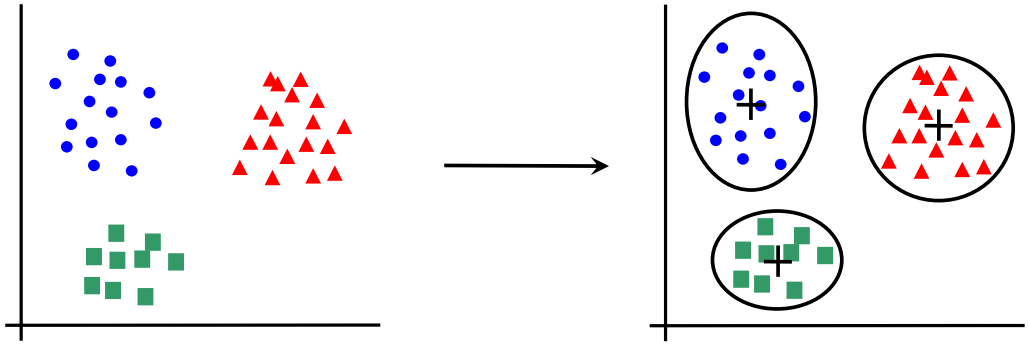
# Lecture 8: Clustering & Mixture Models

C4B Machine Learning Hilary 2011 A. Zisserman

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

- clustering

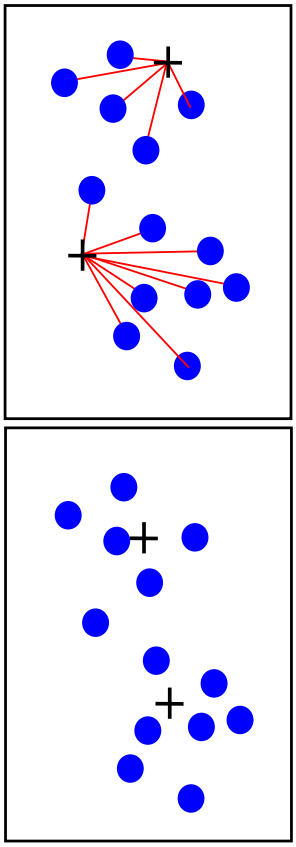


# K-means algorithm

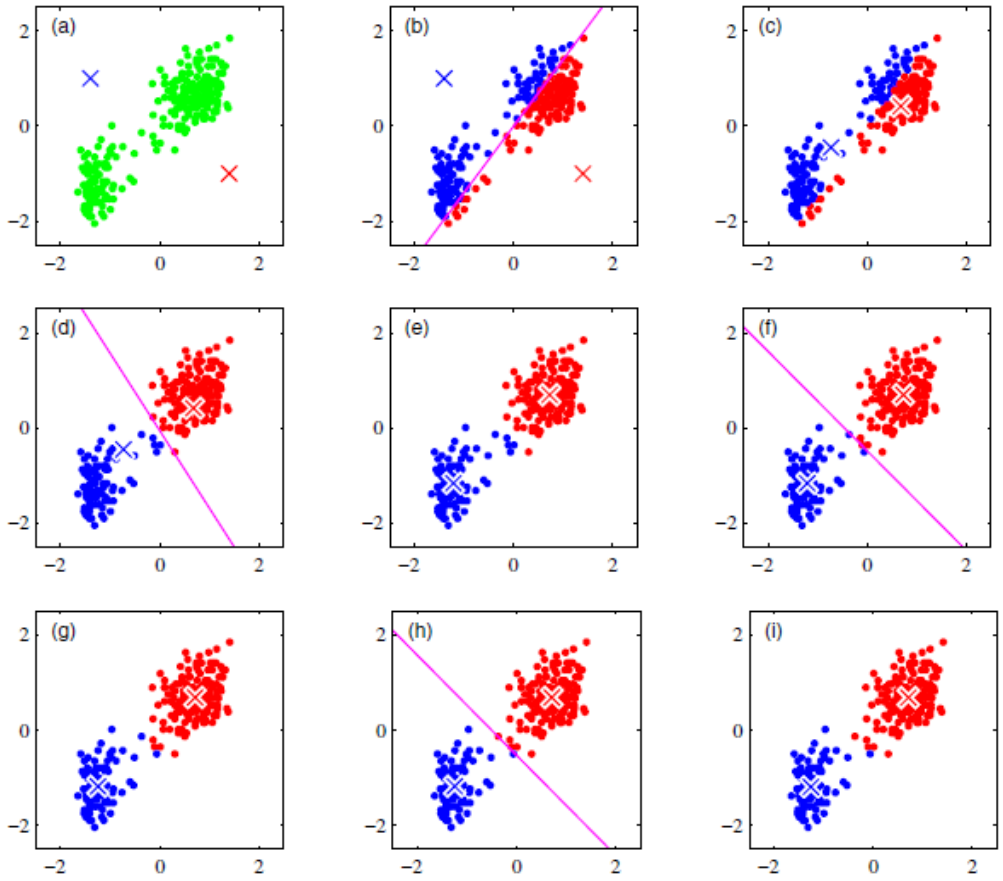
## K-means algorithm

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 $\mathbf{c}_k$

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

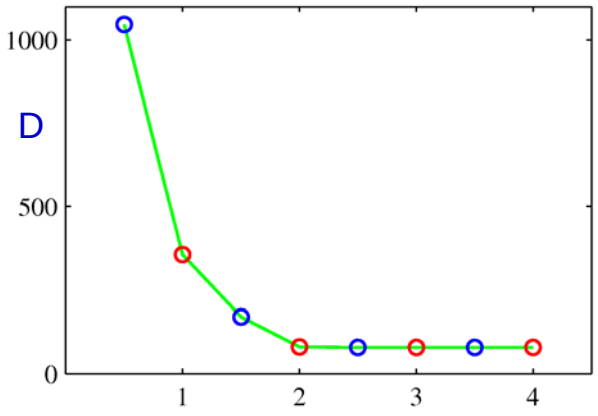
Hence

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

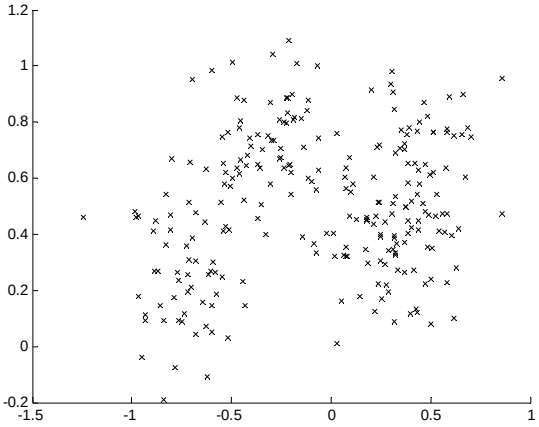
i.e. $\mathbf{c}_k$ is the mean (centroid) of the vectors assigned to it.

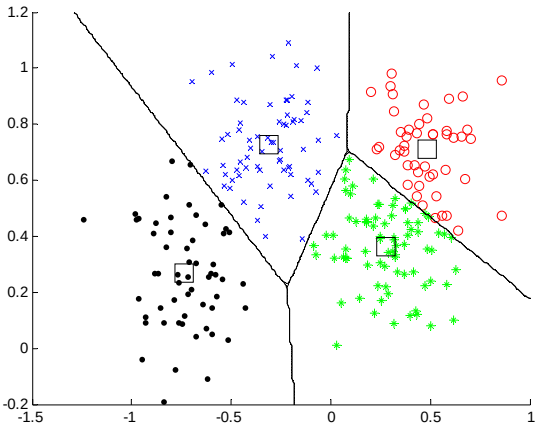
Note, since both steps decrease the cost $D$, the algorithm will converge – but it can converge to a local rather than global minimum.

## Decrease in distortion cost with iterations

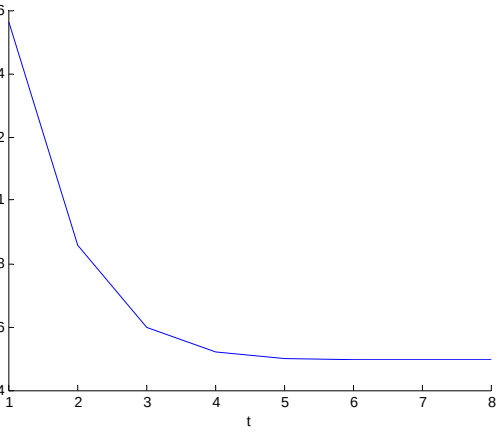


## Sensitive to initialization

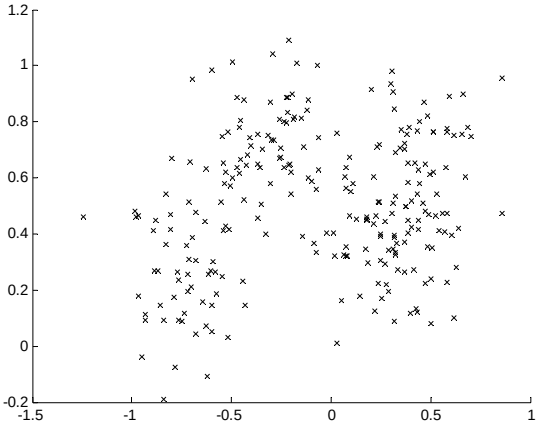


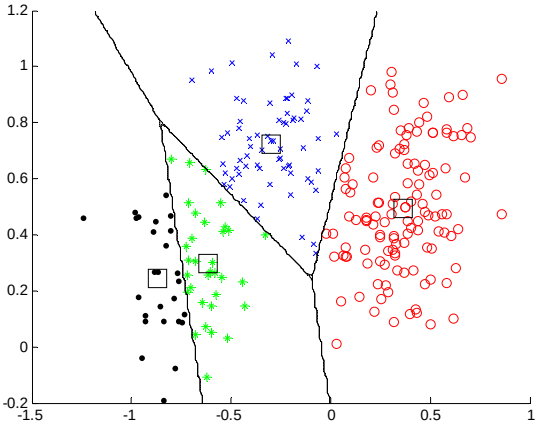




D

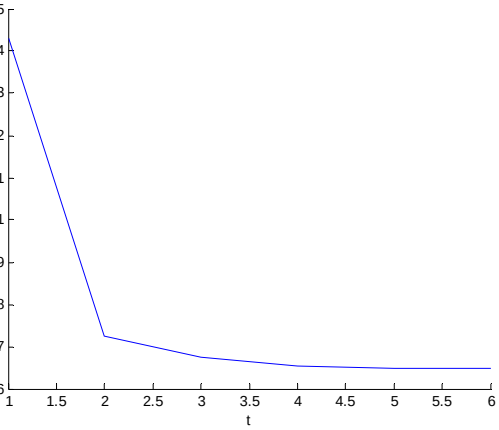



## Sensitive to initialization




D



## Practicalities

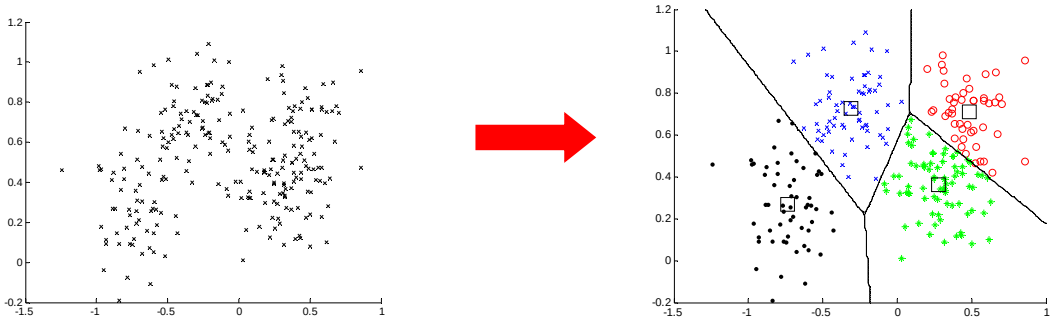
- always run algorithm several times with different initializations and keep the run with lowest cost

- 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 …

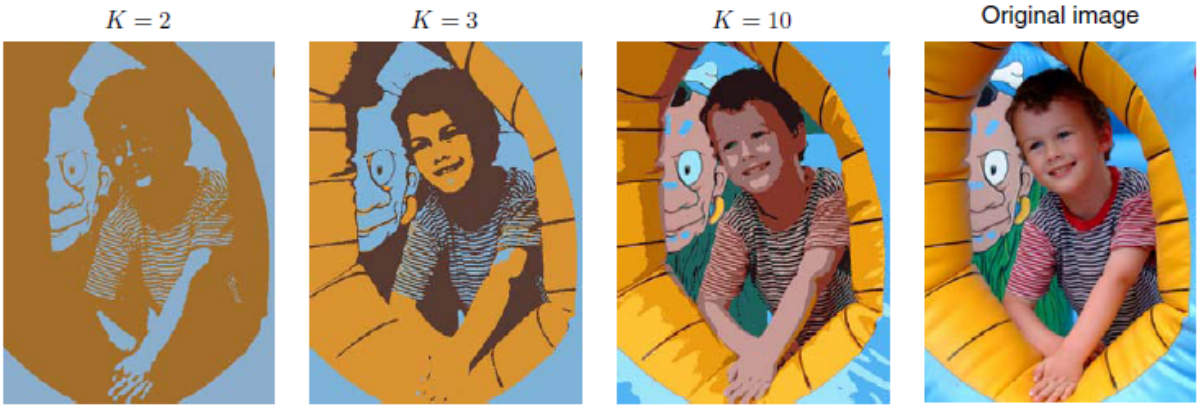
# Example application 1: vector quantization



- 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



# 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

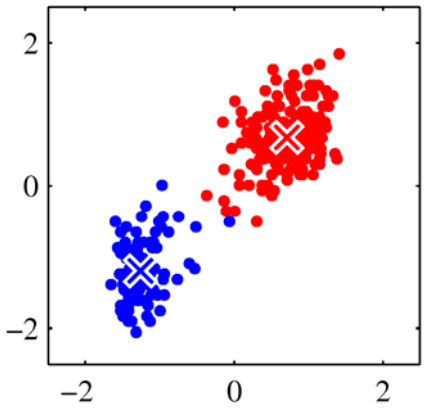
# Gaussian Mixture Models

# 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)

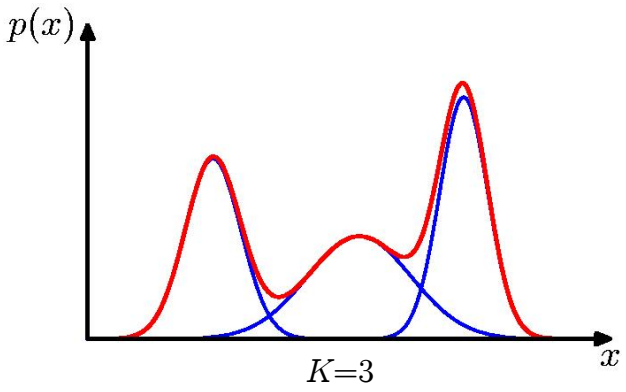
**Combine simple models into a complex model:**

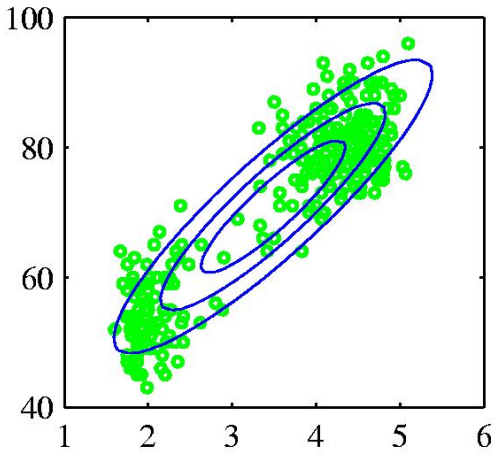
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Component

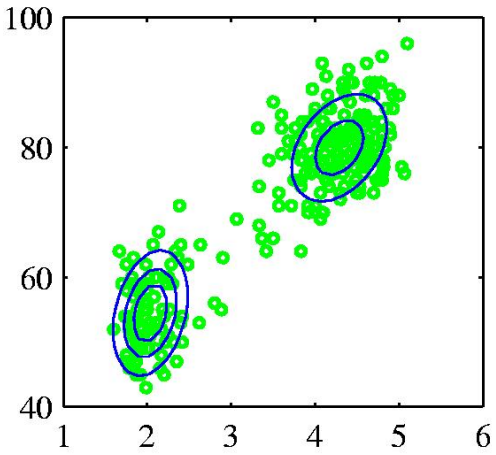
Mixing coefficient

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






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 $\theta$ are the parameters we wish to estimate (i.e. $\boldsymbol{\mu}_k$ and $\Sigma_k$ 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

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

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

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

Differentiating wrt $\Sigma_k$ 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 $\pi_k$ (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)

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

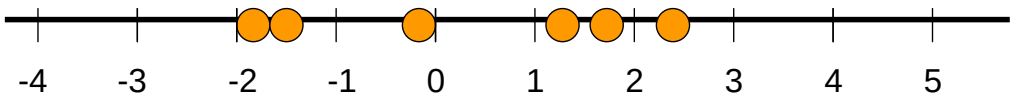
Step 2 Maximization: Re-estimate parameters using computed responsibilities

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

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

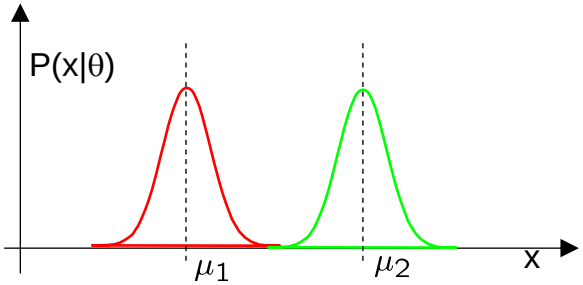
Model:

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

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

keep $\pi, \sigma$ fixed

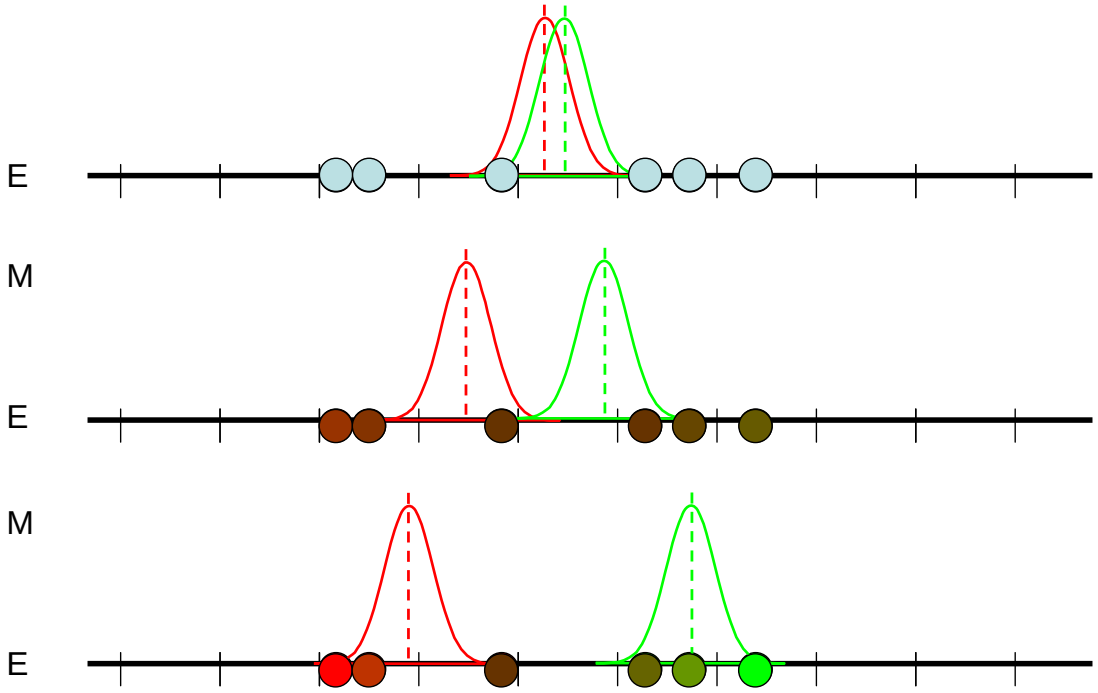
i.e. only estimate $\mu$



# Intuition of EM

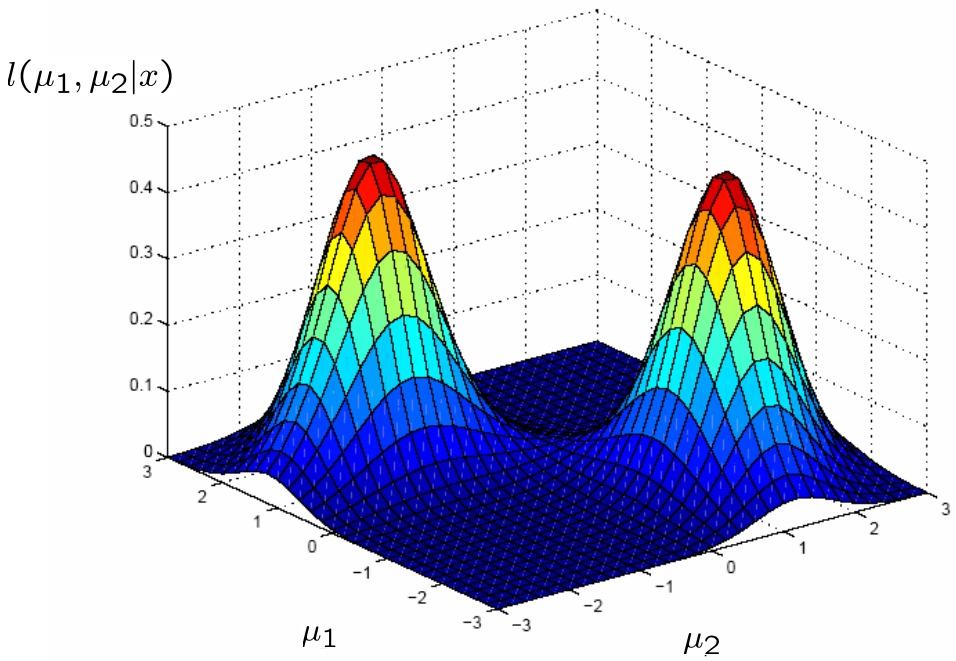
**E-step**: Compute soft assignment of the points, using current parameters

**M-step**: Update parameters using current responsibilities




# Likelihood function

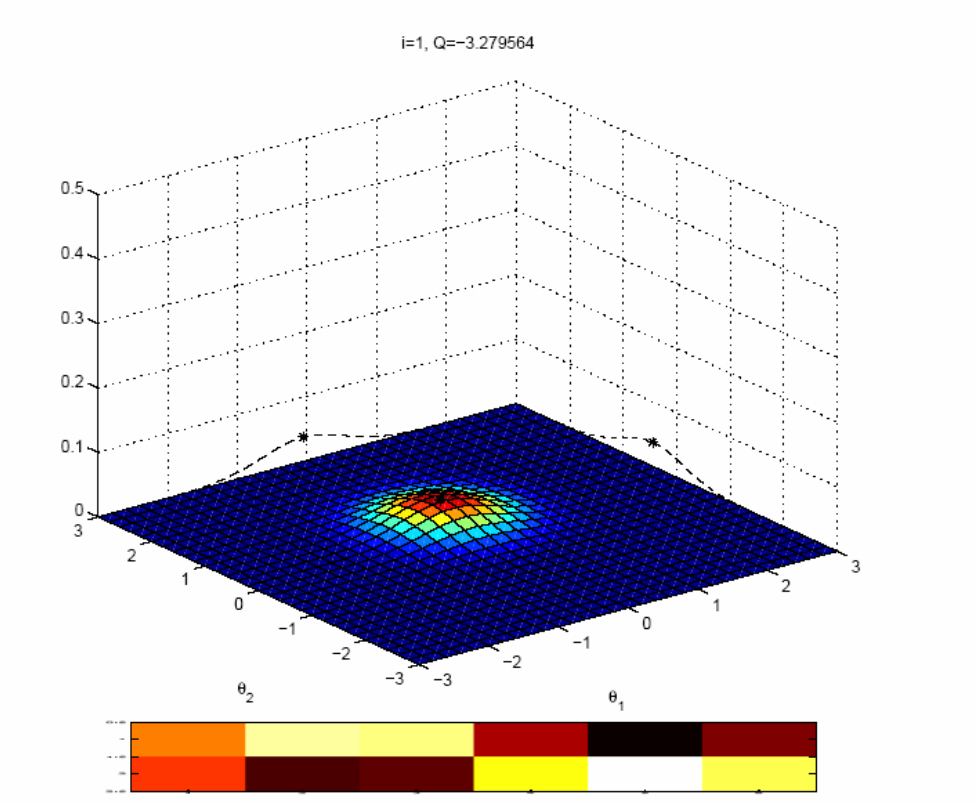
Likelihood is a function of parameters, $\theta$
Probability is a function of r.v. x

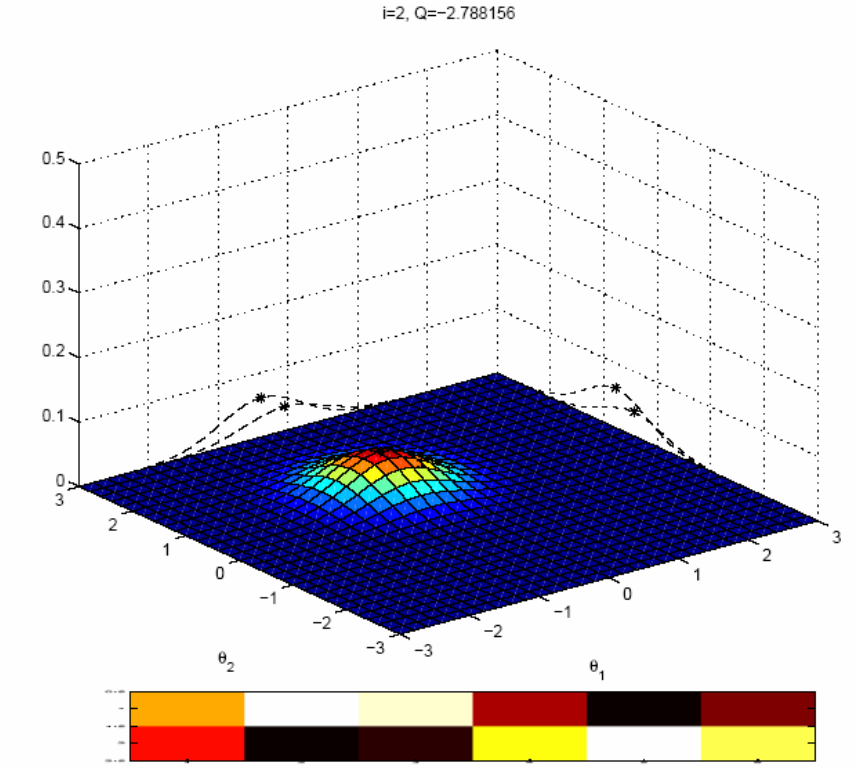


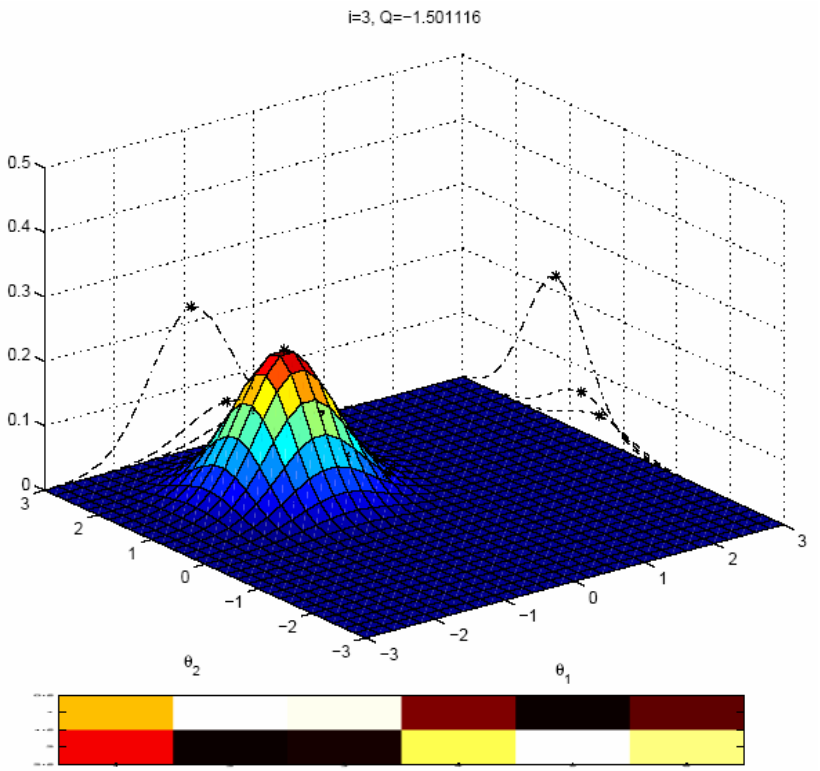
# E-step: What do we actually compute?

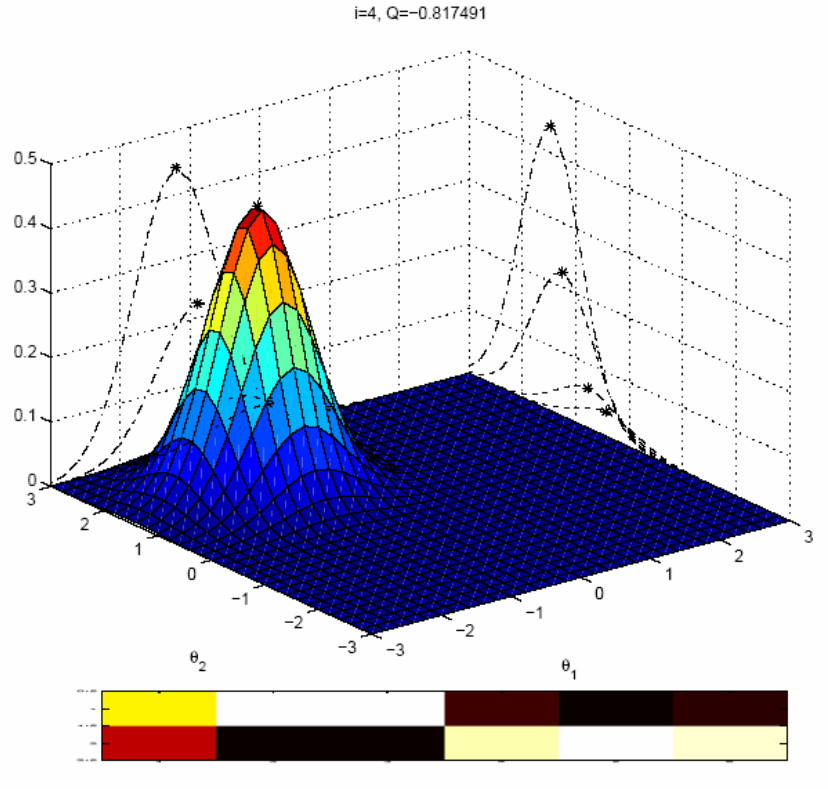
nComponents x nPoints matrix (columns sum to 1):

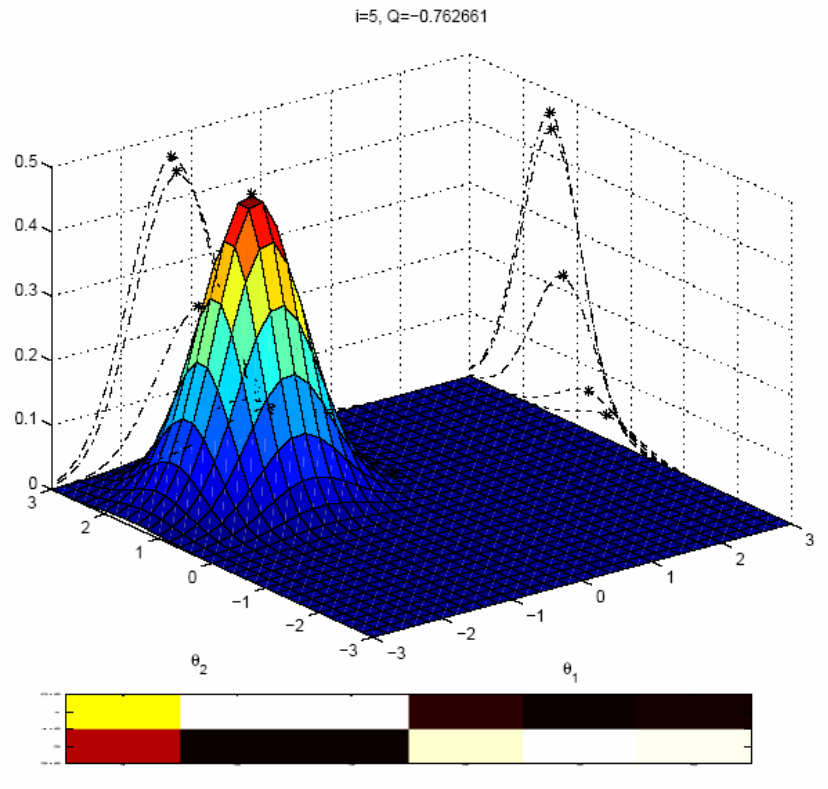
| | Point 1 | Point 2 | | | | Point 6 |
|---|---|---|---|---|---|---|
| Component 1 | | | | | | |
| Component 2 | | | | | | |



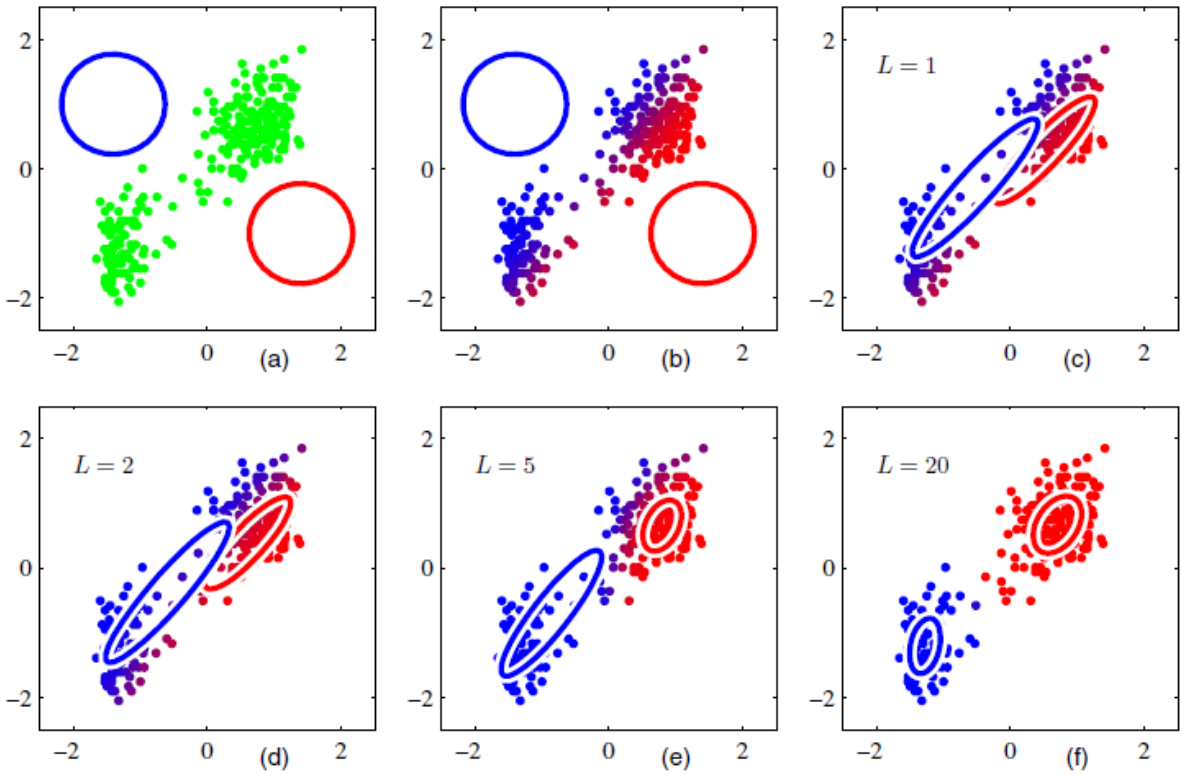

i=2, Q=−2.788156
0.5
0.4
0.3
0.2
0.1
0
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
θ2
θ1


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


i=5, Q=−0.762661
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

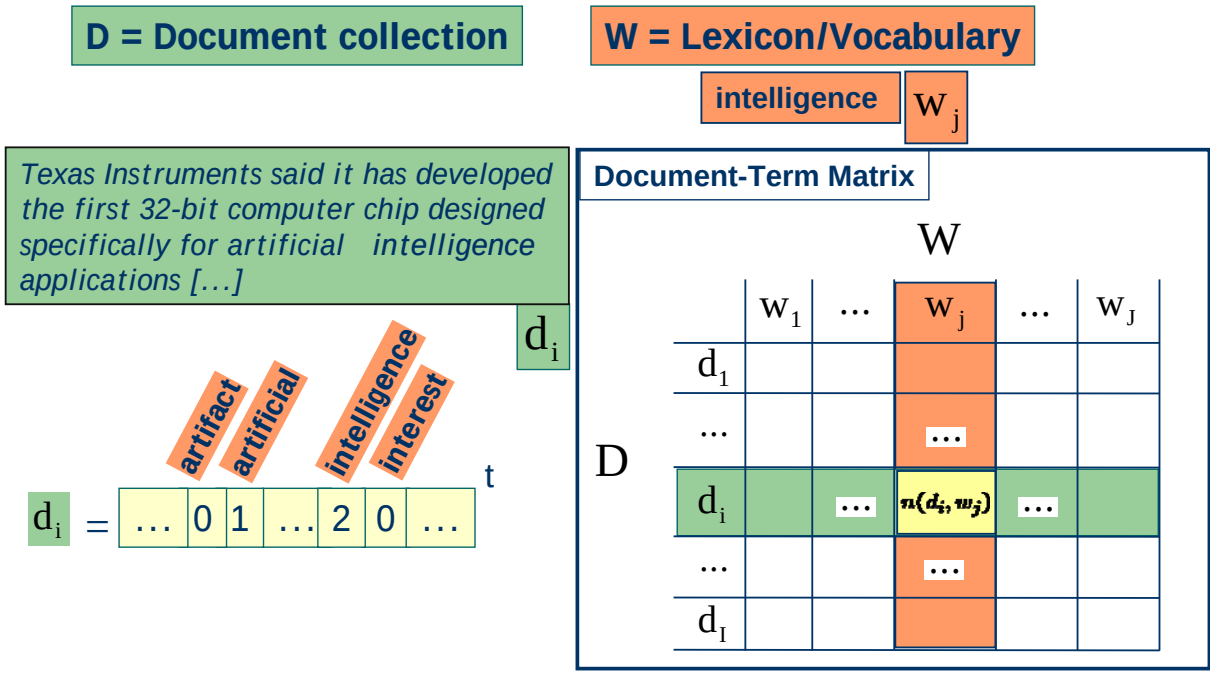
# Probabilistic Latent Semantic Analysis (pLSA)

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)

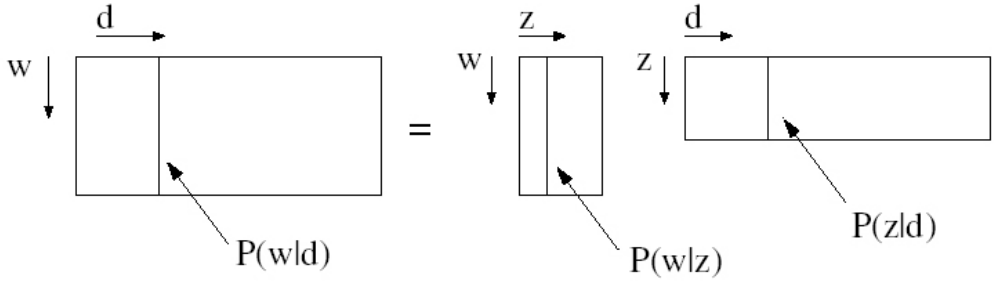
[Hofmann '99]

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)

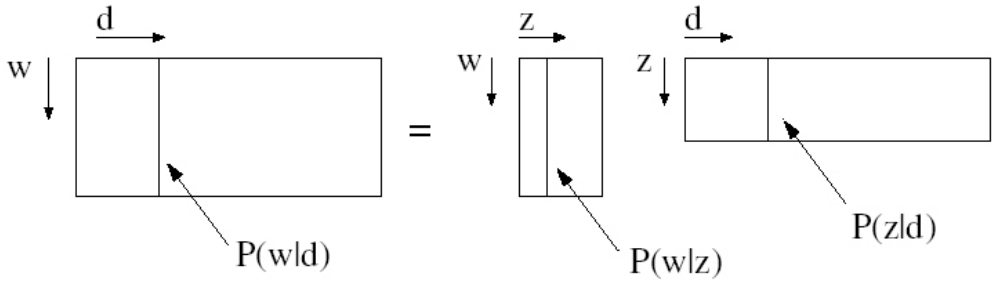
[Hofmann '99]

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

$$P(w|z;\theta) \propto \sum_{d} n(d,w)P(z|d,w), \qquad P(z|d;\pi) \propto \sum_{w} n(d,w)P(z|d,w)$$

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

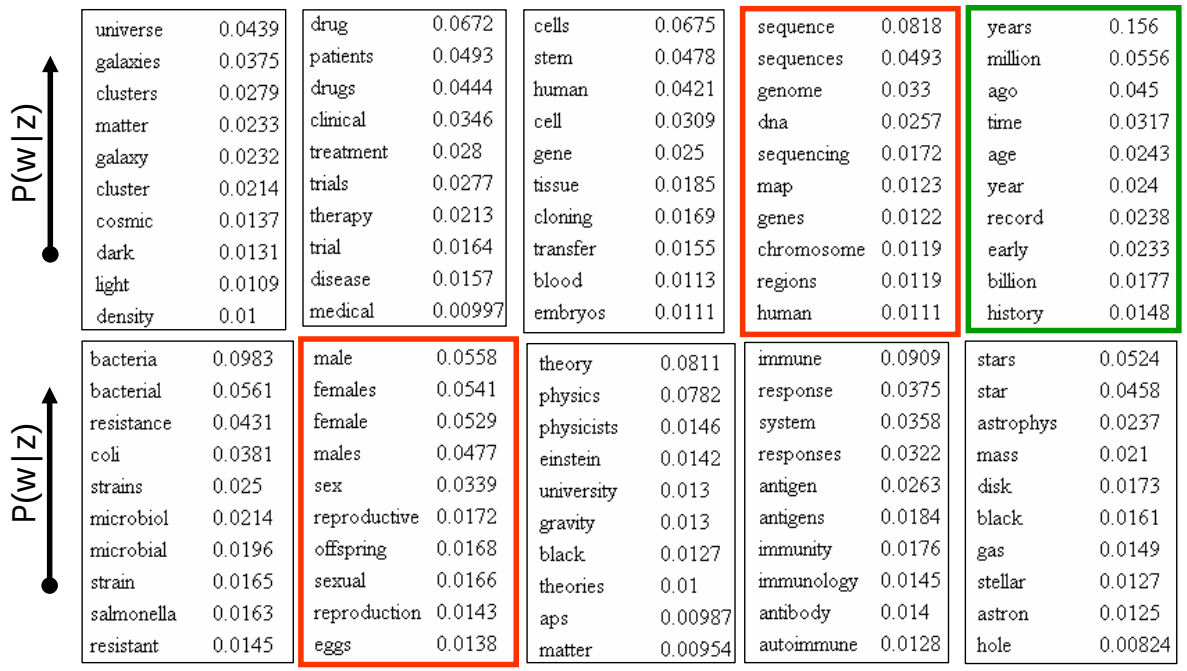
| Topic 1 | |
|---|---|
| securities | 94.96324 |
| firm | 88.74591 |
| drexel | 78.33697 |
| investment | 75.51504 |
| bonds | 64.23486 |
| sec | 61.89292 |
| bond | 61.39895 |
| junk | 61.14784 |
| milken | 58.72266 |
| firms | 51.26381 |
| investors | 48.80564 |
| lynch | 44.91865 |
| insider | 44.88536 |
| 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)




# Background reading

- Bishop, chapter 9.1 – 9.3.2

- Other topics you should know about:
  - random forest classifiers and regressors
  - semi-supervised learning
  - collaborative filtering

- More on web page:
  http://www.robots.ox.ac.uk/~az/lectures/ml