Clustering & Expectation-Maximization

STAT261: Introduction to Machine Learning

Lecture 10, April April 27

Outline

- Factor Analysis
- LDA – review last time, multiple vectors
- Clustering
  - K-means
  - Hierarchical Clustering - brief description
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

Factor Analysis

- Find a small number of factors $\boldsymbol{z}$, which when combined generate $\boldsymbol{x}$:

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

where z_j, $j = 1,\dots,k$ are the latent factors with

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

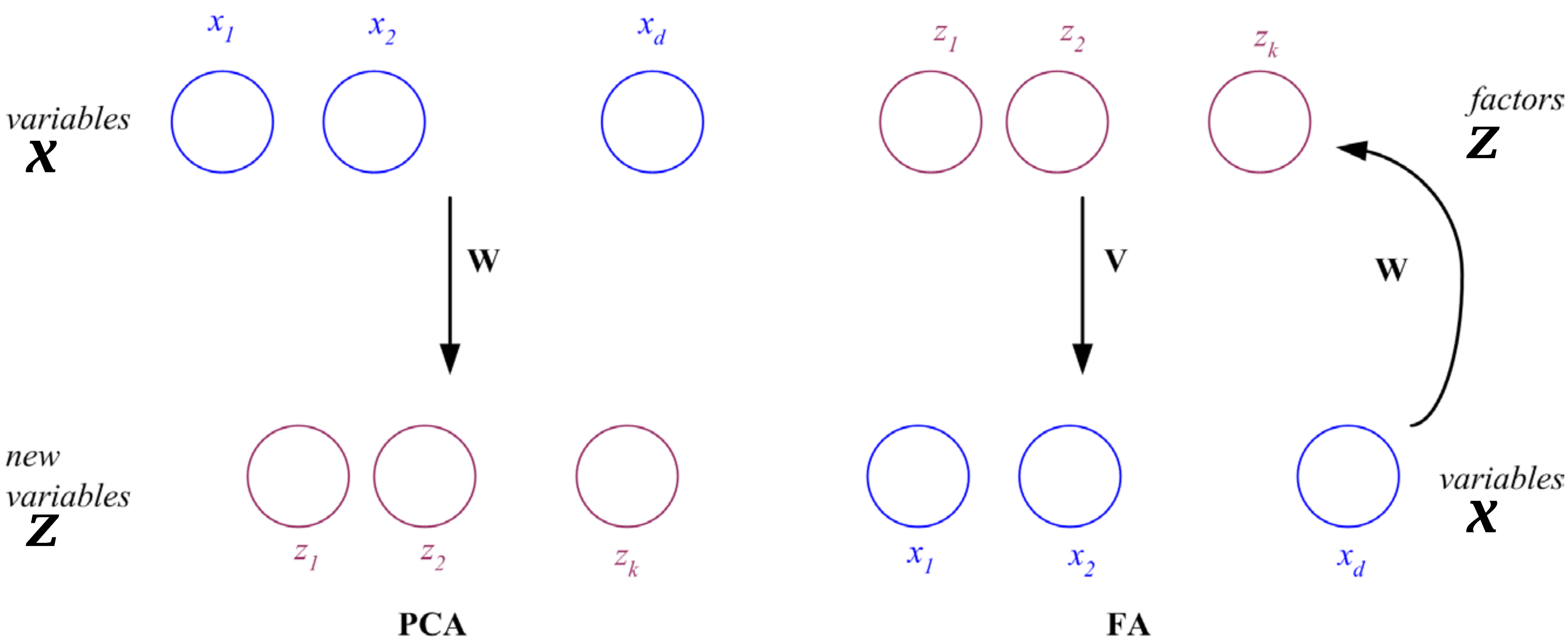
ε_i are the noise sources

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

and v_{ij} are the factor loadings

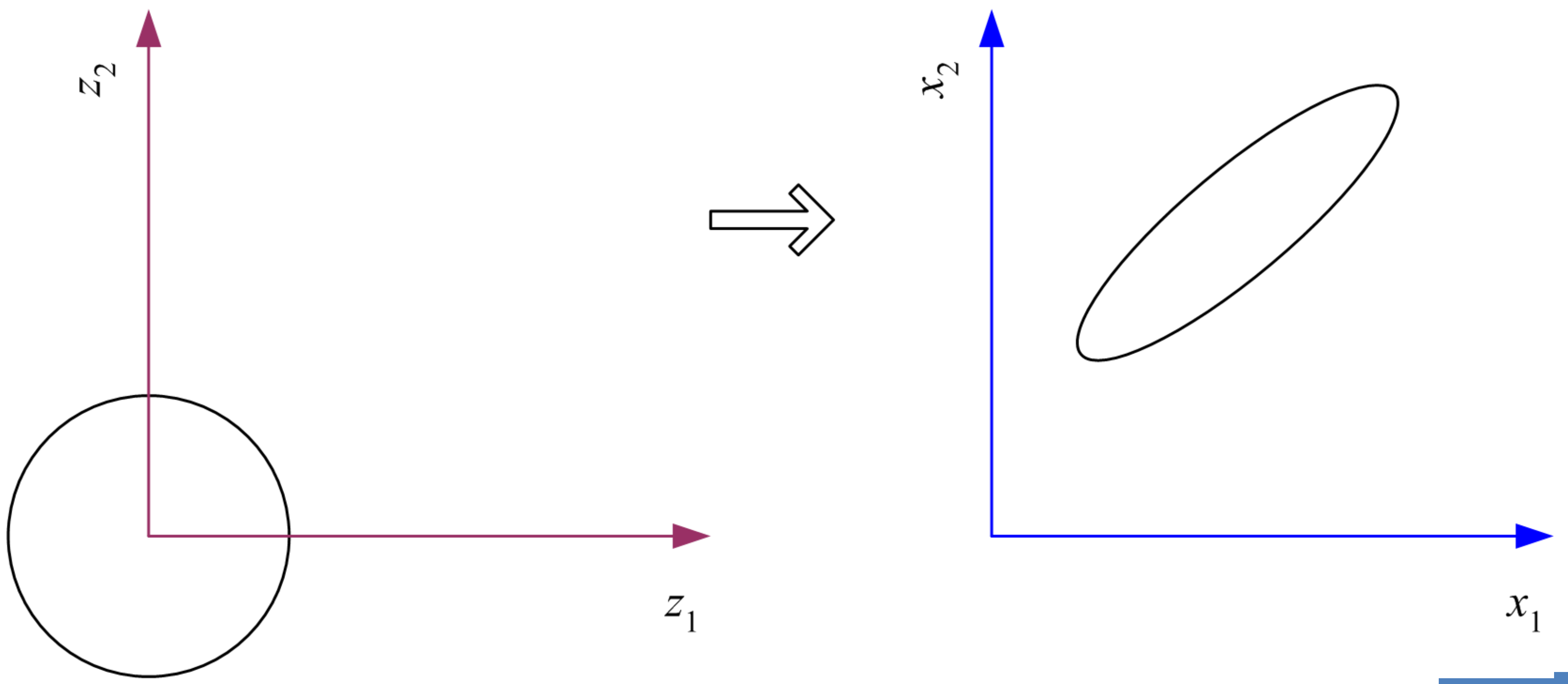
PCA vs FA

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

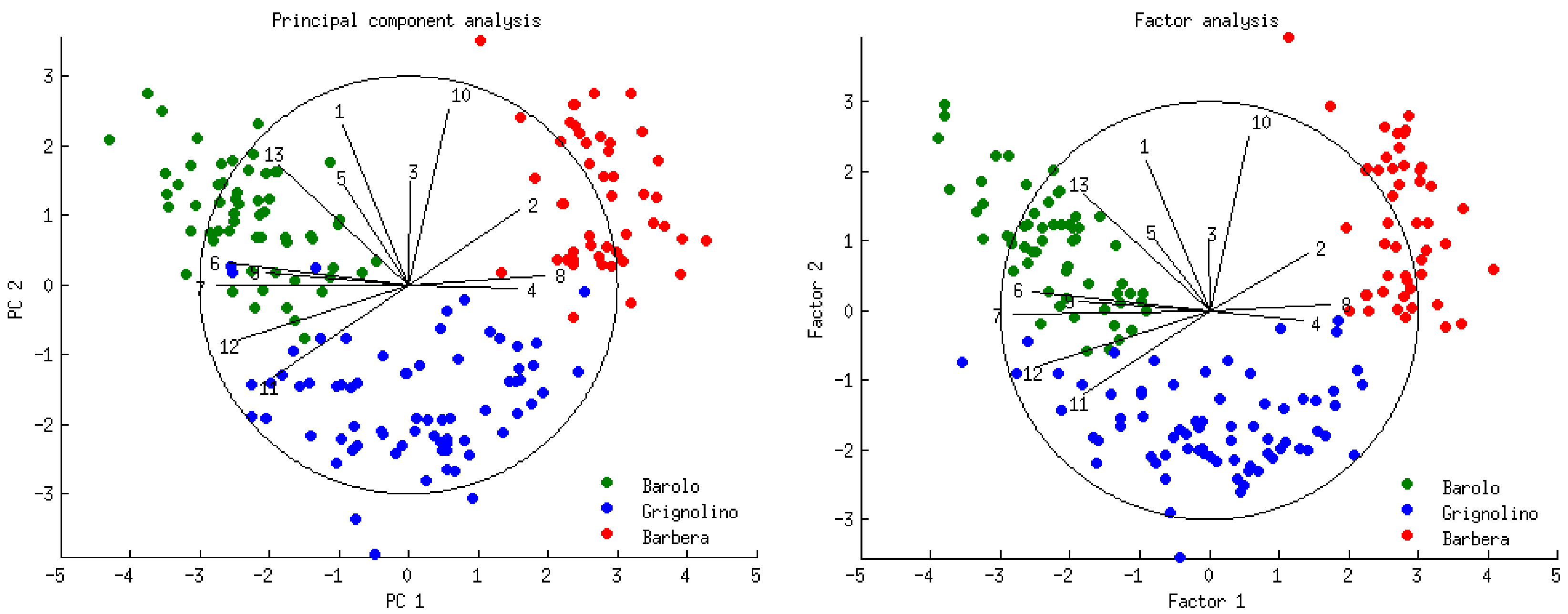


Factor Analysis

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



PCA vs FA



Principal component analysis
PC 1
PC 2
Factor analysis
Factor 1
Factor 2
Barolo
Grignolino
Barbera

Outline

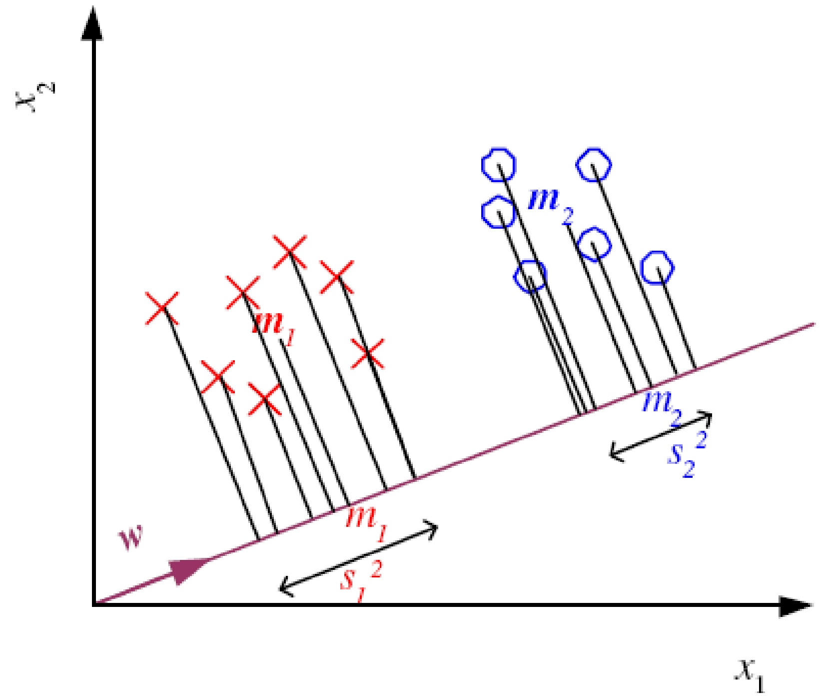
- Factor Analysis
- LDA – multiple vectors
- Clustering
  - K-means
  - Hierarchical Clustering - brief description
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

Linear Discriminant Analysis

- Lower dimension--
  preserves class separation.

  (hence discriminant)
- Project data on vector w again (hen
- Find ***w*** that maximizes

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



It turns out that that w is the normal vector to the plane that best separates the classes

Problem Set Up

- X_1, X_2: Data matrices from two classes
- Sample mean and covariance in each data set, before projections
  - μ_ℓ, S_ℓ
- Given vector w define:
  - Mean in each class: $m_\ell = w^T \mu_\ell$, after projection
  - Sample variance in each class: $s_\ell^2 = w^T S_\ell w$, after projection
  - Number of samples in each class: N_ℓ
- Fisher Criteria: Find direction w to maximize:

$$J = \frac{N|m_1 - m_2|^2}{N_1 s_1^2 + N_2 s_2^2}$$

  - Average squared difference normalized by the variance

Solution to the Fisher Criteria (K=2 classes)

- Matrix form of Fisher criteria:

$$J = \frac{N|m_1 - m_2|^2}{N_1 s_1^2 + N_2 s_2^2} = \frac{N\left|w^T(\mu_1 - \mu_2)\right|^2}{N_1 w^T S_1 w + N_2 w^T S_2 w}$$

$$= \frac{w^T S_B w}{w^T S_W w}$$

- $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$, between class scatter, before projection
- $S_W = \frac{N_1}{N} S_1 + \frac{N_2}{N} S_2$ weighted sum in-class scatter, before projection

Solution to the Fisher Criteria (K=2 classes)

- Optimization of Fisher

- Take derivative and set to zero:
  - $(w^T S_B w) S_W w = (w^T S_W w) S_B w$
  - $(w^T S_B w) S_W w = \left(w^T S_W w\right)(\mu_1 - \mu_2)^T w (\mu_1 - \mu_2)$
  - $S_W w = c(\mu_1 - \mu_2)$

- LDA solution: $w = c S_w^{-1}(\mu_1 - \mu_2)$

K>2 Classes

- With multiple classes, let
  - Overall sample mean: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$
  - Sample mean and covariance in each class: μ_ℓ, S_ℓ
- Define:
  - Cross-class variances: $S_B = \sum_{\ell=1}^{K} N_\ell(\mu - \mu_\ell)(\mu - \mu_\ell)^T$
  - In-class scatter: $S_W = \sum_{\ell=1}^{K} N_\ell S_\ell$
- LDA components are $K-1$ eigenvectors of $S_W^{-1}S_B$
- Or Find W that maximizes

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T\mathbf{S}_B\mathbf{W}\right|}{\left|\mathbf{W}^T\mathbf{S}_W\mathbf{W}\right|}$$

The largest eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$

Maximum rank of K-1

Fisher's Linear Discriminant

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

- LDA soln: $K - 1$ eigenvectors of $S_W^{-1} S_B$

The largest eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$
Maximum rank of K-1

- Parametric soln:
$$\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2)$$
when $p(\mathbf{x} \mid C_i) \sim \mathcal{N}(\mu_i, \Sigma)$

Outline

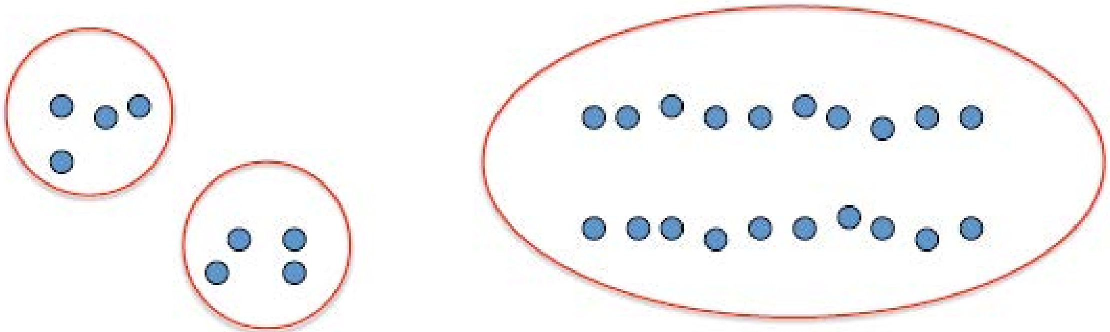
- Factor Analysis
- LDA – multiple vectors
- Clustering
  - K-means
  - Hierarchical Clustering - brief description
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

Clustering: Unsupervised Learning

- Unsupervised learning: Requires data, but no labels
- Finds groups / clusters in data
    - Goal: Automatically segment data into groups of similar points
- Question: When and why would we want to do this?
- Useful for:
    - Don't know what we are looking for
    - Automatically organizing data
    - Understanding hidden structure in some data
    - Representing high-dimensional data in a low-dimensional space
- Examples:
    - Customers shopping patterns & regionalities
    - Genes according to expression profile
    - Search results according to topic
    - A museum catalog according to image similarity

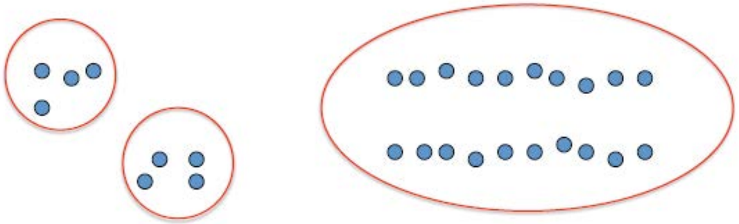
Simple Example

- Basic idea: group together similar instances
- Example: 2D point patterns

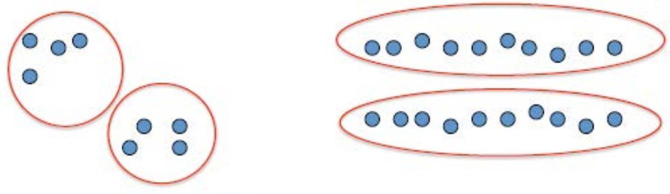


Simple Example – Cont'd

- What does "similar" mean?
- Two possible clusters
  - Option 1



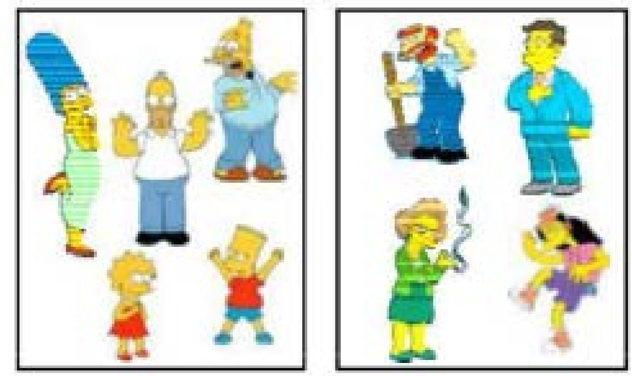
  - Option 2

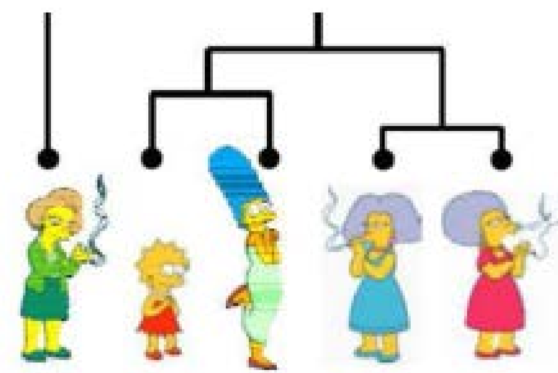


  - Which is right?
- Depends on how we define "similar"
  - And what we consider close?
- Will use Euclidean distance $d_{ij} = \|x_i - x_j\|^2$
  - Can use any similarity "metric" or distance you like

Clustering Algorithms

- Partition algorithms (flat)
  - K-means
  - Gaussian mixture models
  - Spectral methods
- Hierarchical clustering
  - Bottom up: Agglomerative
  - Top down: Divisive





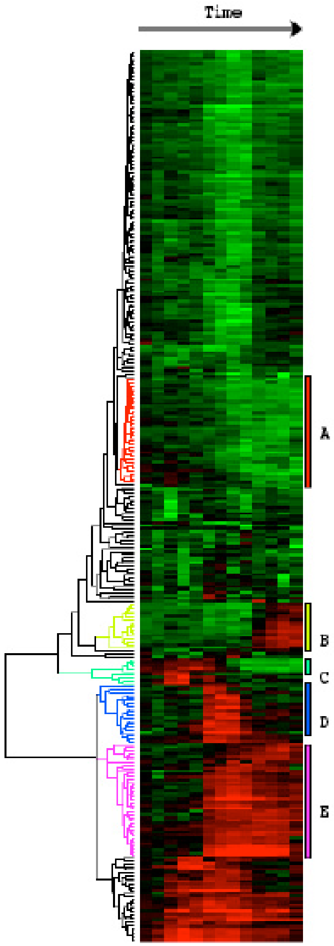
Example: Image Segmentation

- Goal:
  - Break up image
  - Meaningful or perceptually similar regions

[Slide from James Hayes]

Example: Gene Expression Data

- Gene believed to work in groups
- Each gene has "expression level"
  - Amount of protein produced
- Collect expression levels at different times
  - Measured in microarray
- Identify clusters:
  - Genes in same cluster will express together



Outline

- Factor Analysis
- LDA – multiple vectors
- Clustering
  - K-means
  - Hierarchical Clustering - brief description
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

K-means: Simple iterative clustering

- Simple iterative algorithm to:
  - μ_k = mean of each cluster (hence "K-means")
  - $C_n \in \{1, \ldots K\}$ = cluster of sample x_n
- Step 0: Start with guess at centroids: μ_k
  - Random ok, often "smart" heuristic, distances, etc
- Step 1: Assign cluster to sample x_n

$$C_n = \arg\min_k \|x_n - \mu_k\|^2$$

  - Select cluster with closest mean
- Step 2: Update mean of each cluster:

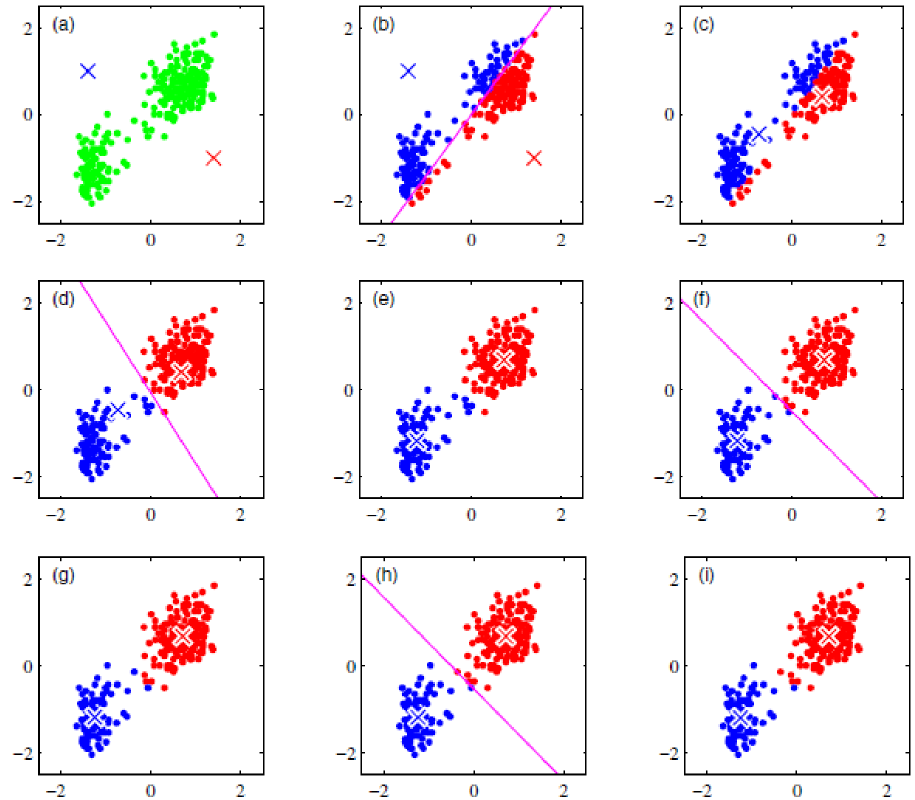
$$\mu_k = \text{average of } x_n \text{ for } x_n \; with \; C_n = k$$

- Return to step 1

Old Faithful K-Means illustrated

y-axis : time

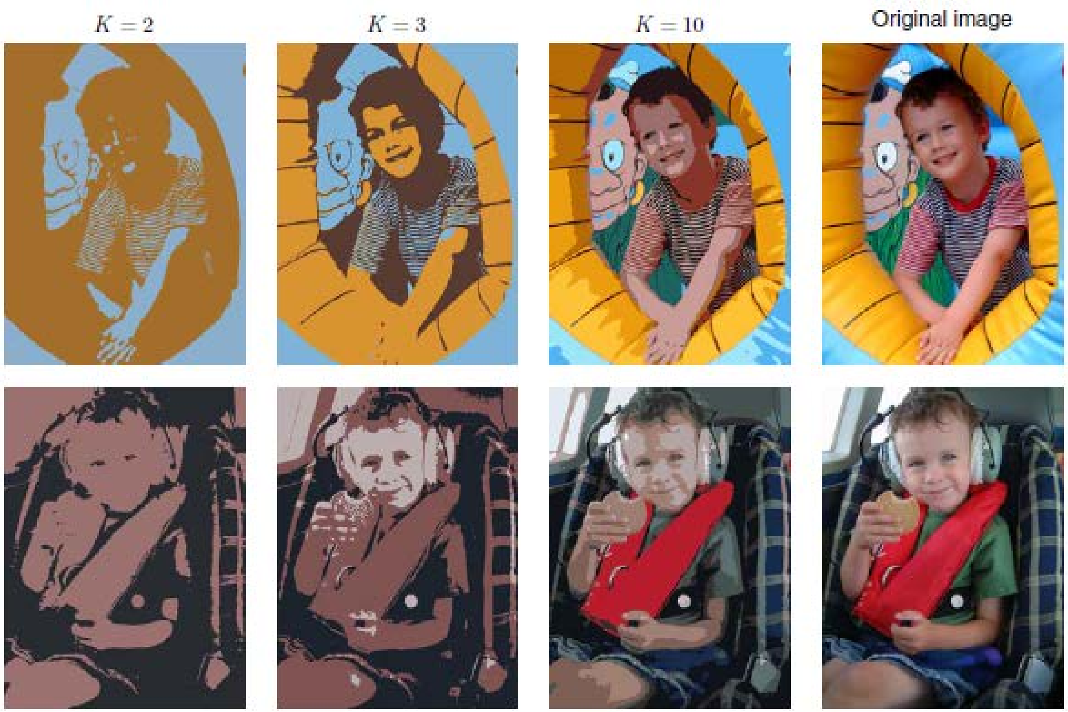
until the next eruption



x-axis : duration of the eruption

- From Bishop, Chapter 9. K-Means on “old faithful” data set

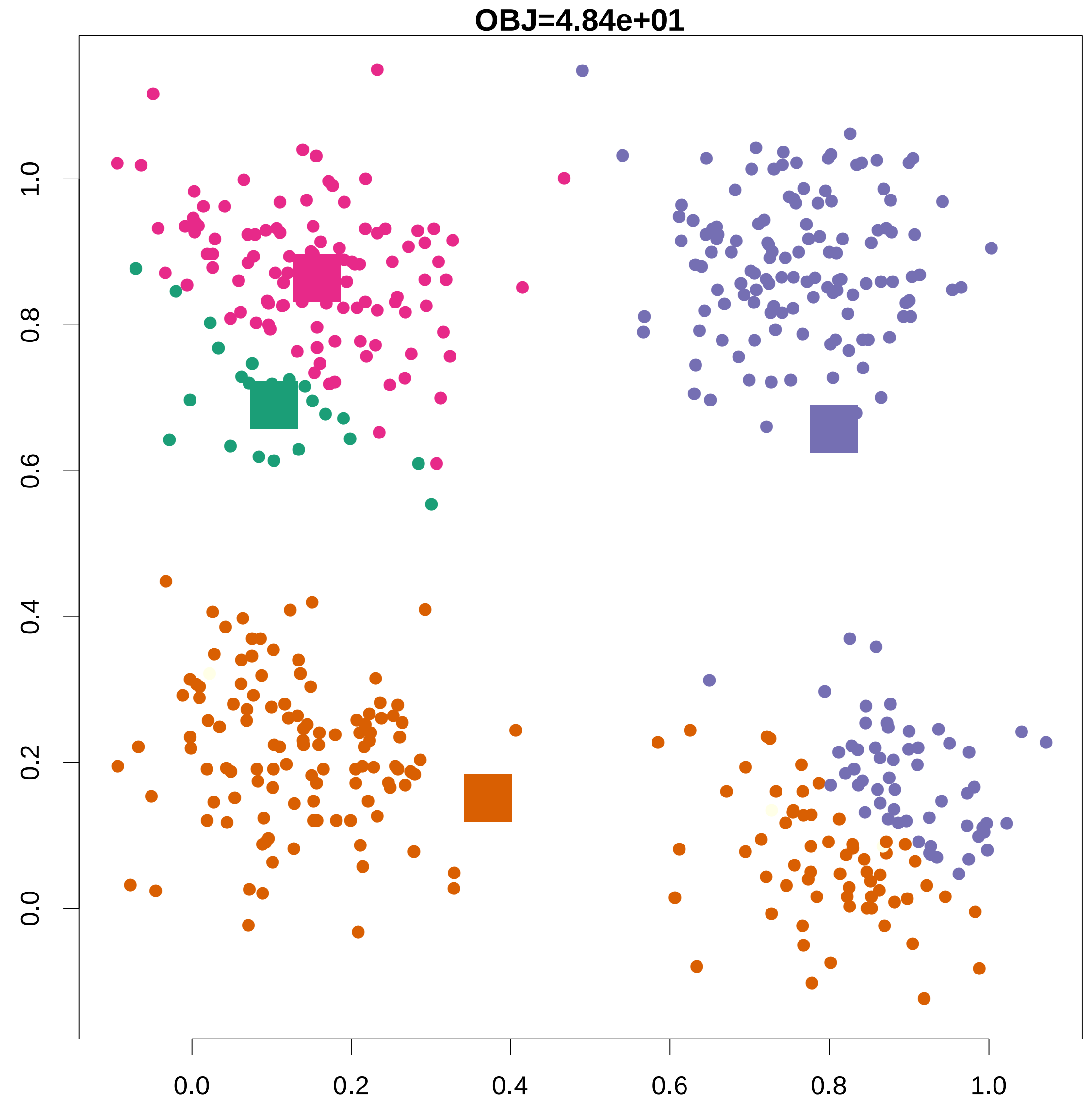
Image Segmentation

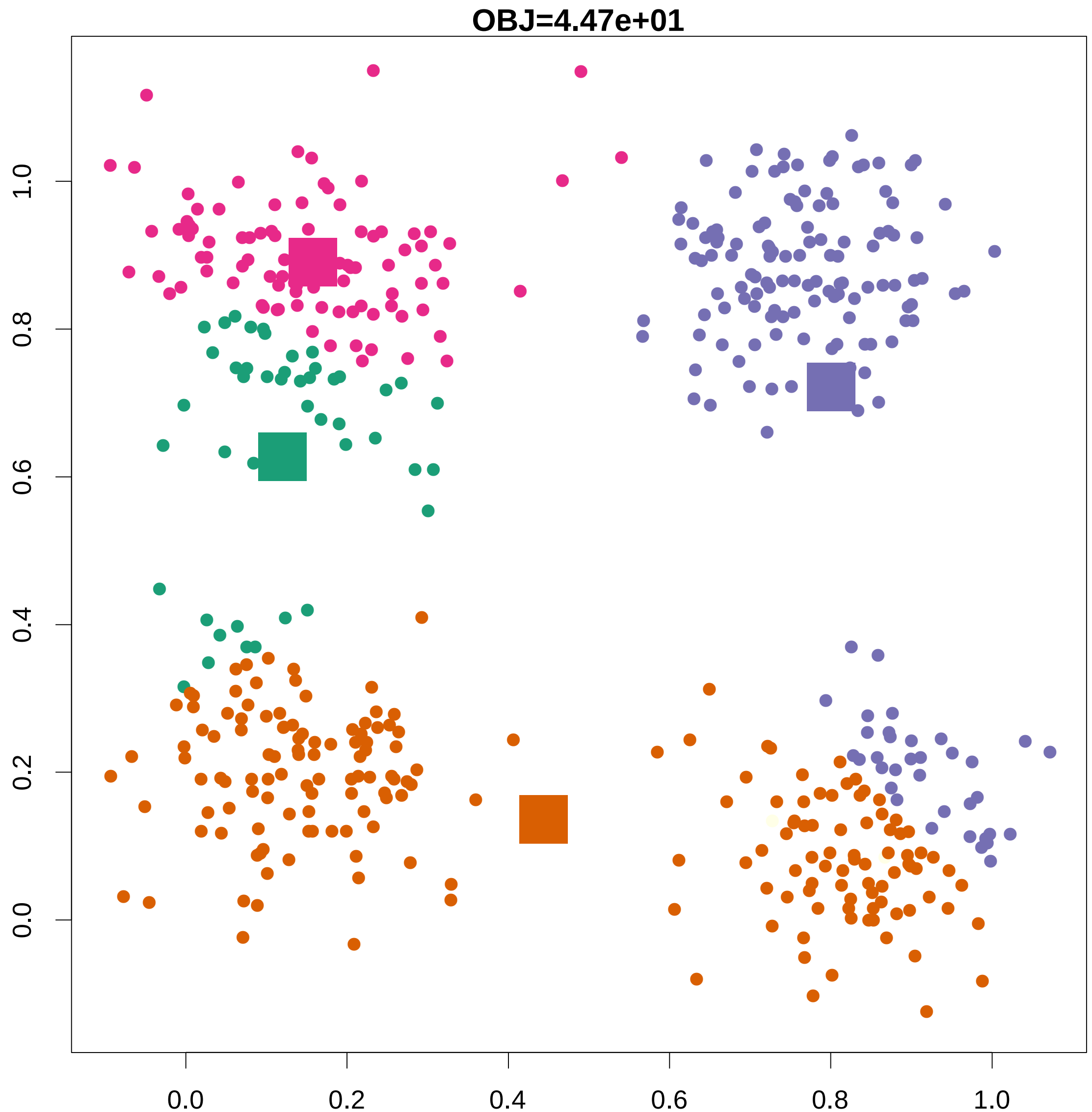


- Color based segmentation
- RGB (0,255)
- Euclidean "pixel-colour" distance
- Typically locality in images matter
- Proximity based clustering
- Textures, edges, shapes

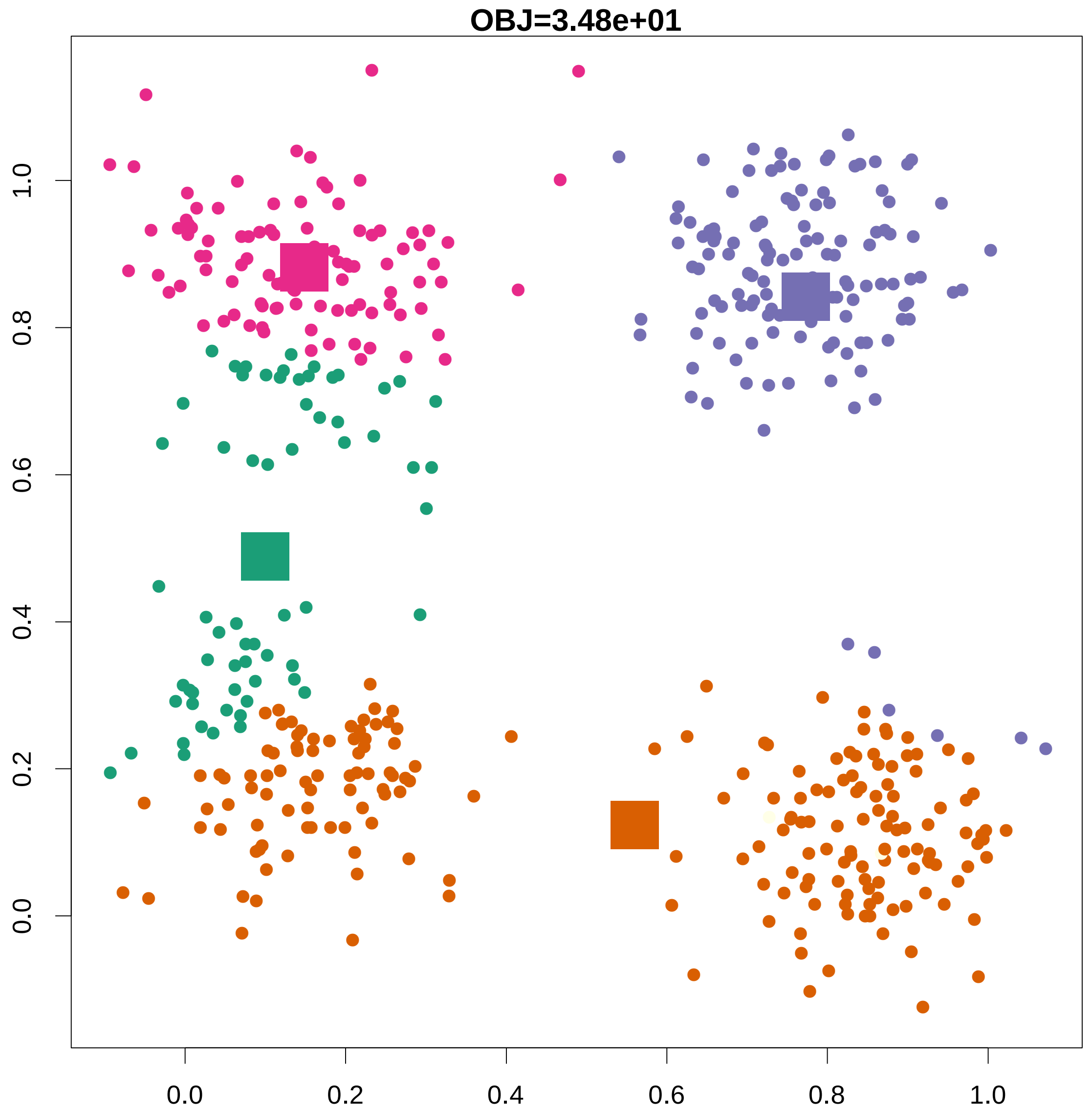
- Use K-means on the RGB values (dimension = 3)
- Replace each pixel by the colour that of the centroid of the cluster

Kmeans





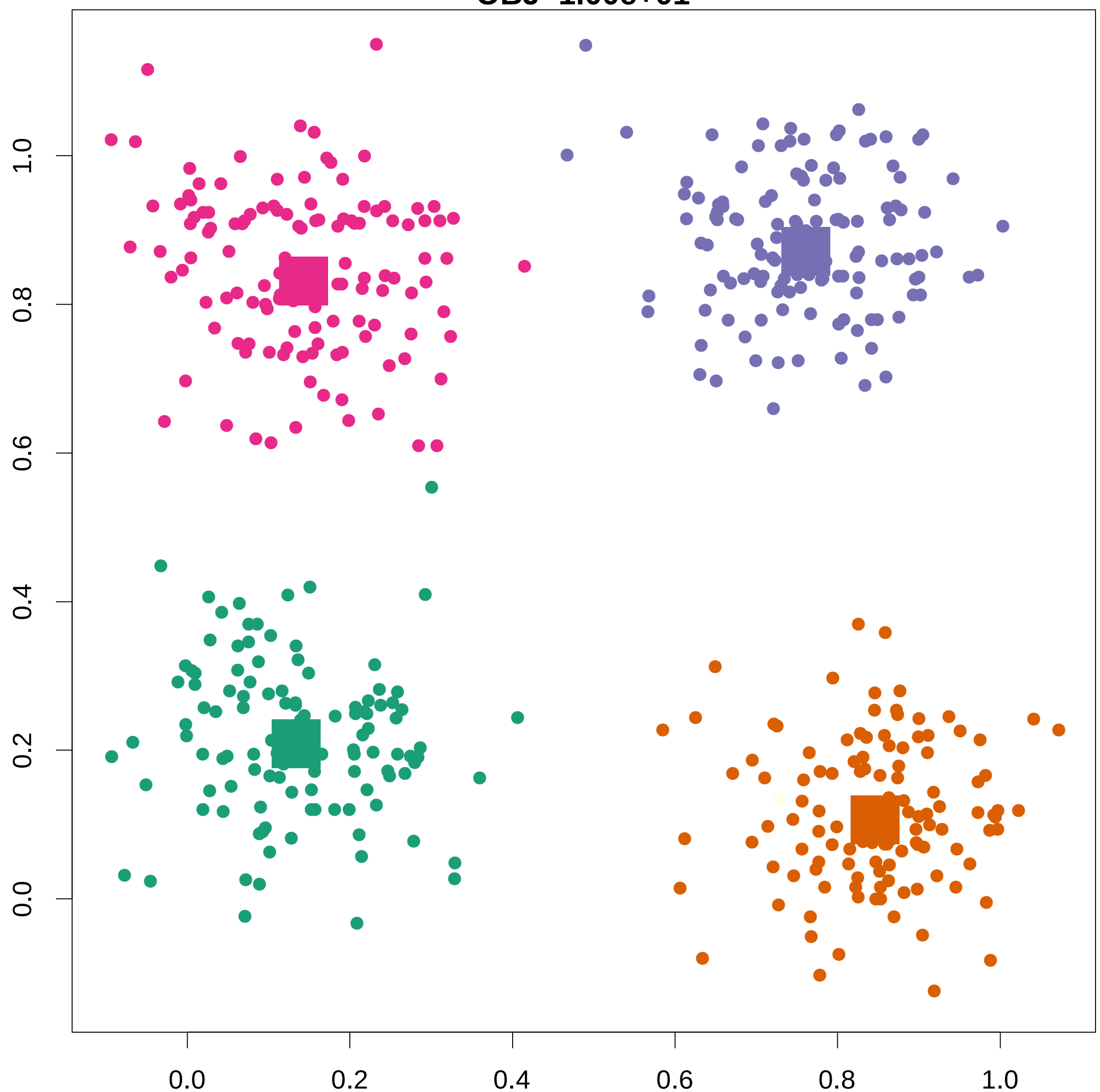
OBJ=4.47e+01
0.0
0.2
0.4
0.6
0.8
1.0



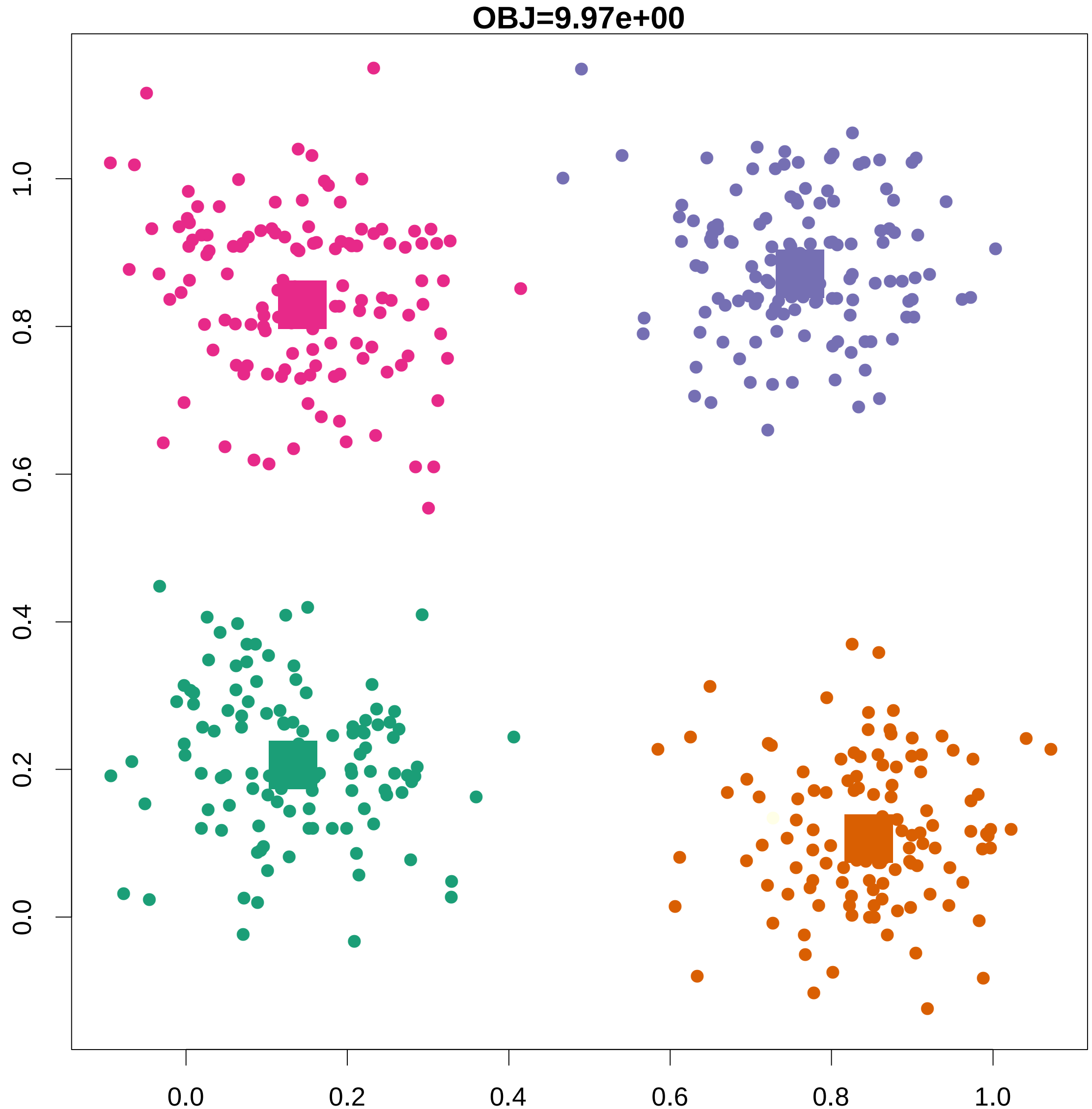
OBJ=3.48e+01
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

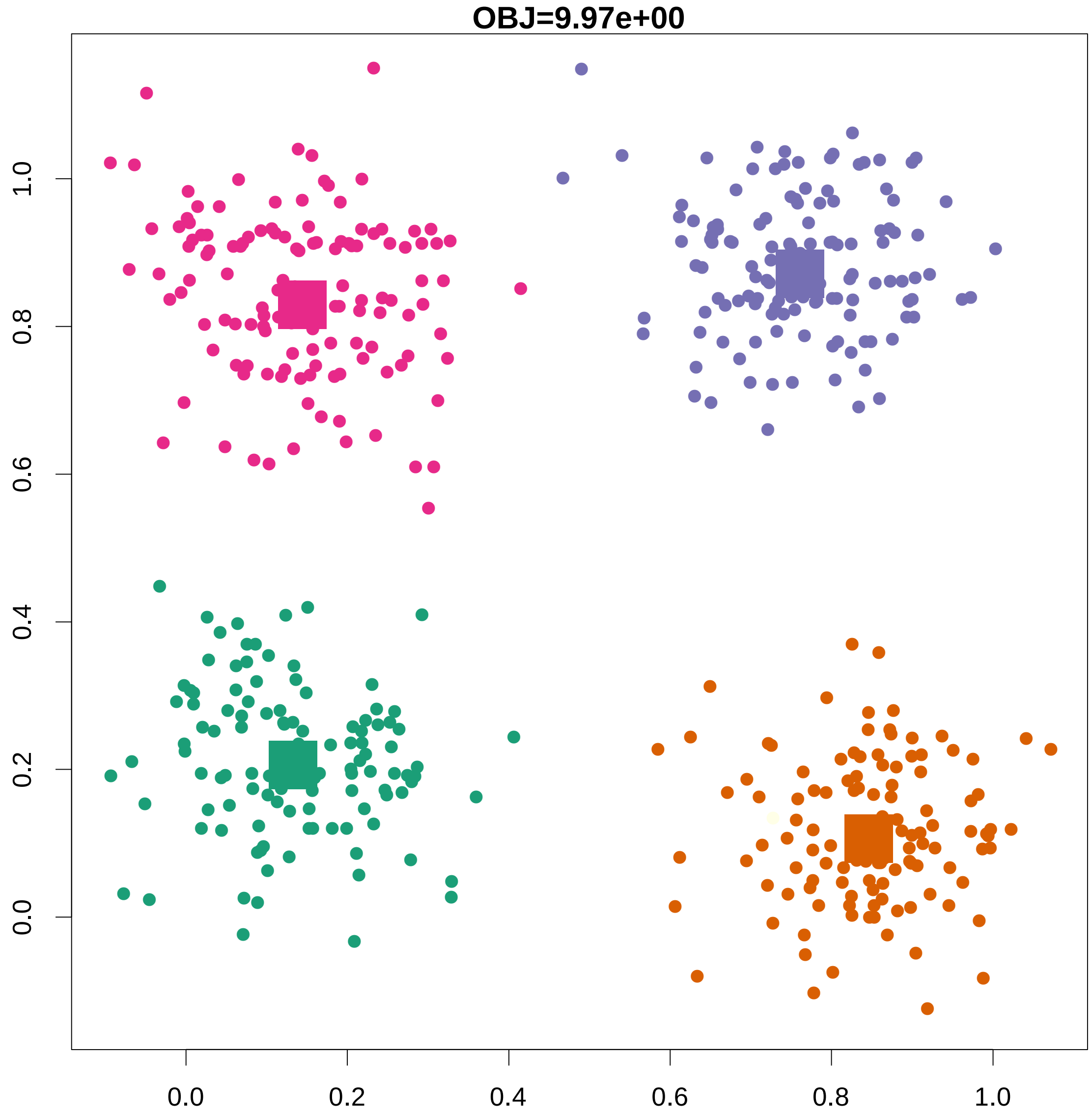

OBJ=1.73e+01
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

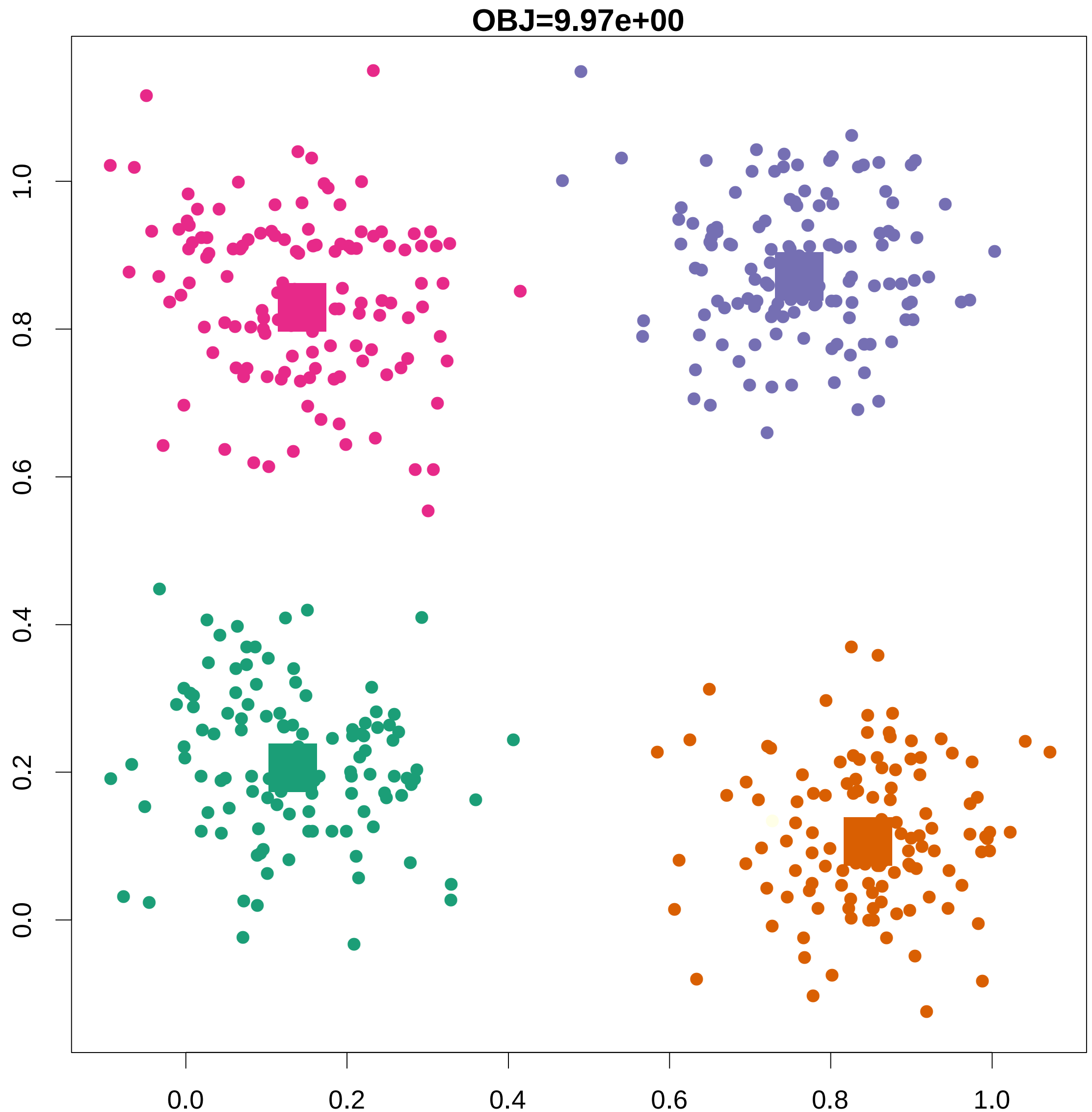
OBJ=1.00e+01




OBJ=9.97e+00
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


OBJ=9.97e+00
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


OBJ=9.97e+00
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


OBJ=9.97e+00
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

K-means Convergence

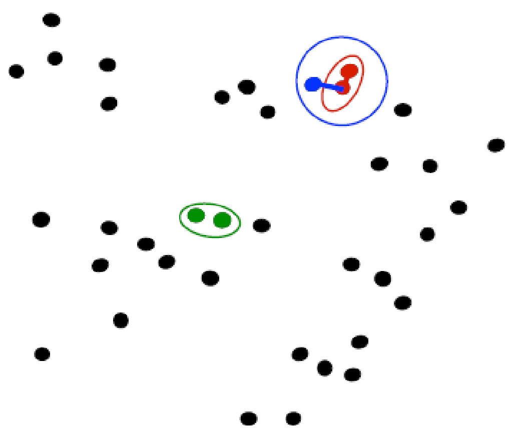
- Will converge to some cluster
- Will always converge to a "local" minima of cost function

$$J(r_{nk}, \mu_k) = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} \|x_n - \mu_k\|^2$$

  - Subject to $r_{nk} = 0$ or 1 and $\sum_i r_{nk} = 1$
- K-means alternately decreases J
  - Fix centroids, μ_k, and update the memberships r_{nk}
- But, can get stuck in a local minima
  - May need good selection of initial condition

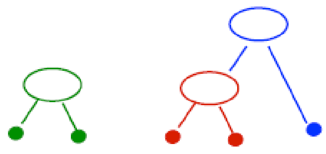
Agglomerative Clustering

- Agglomerative clustering:
  - First merge very similar instances
  - Incrementally build larger clusters out of smaller clusters



- Algorithm:
  - Maintain a set of clusters
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there' s only one cluster left

- Produces not one clustering, but a family of clusterings represented by a dendrogram



Outline

- Factor Analysis
- LDA – multiple vectors
- Clustering
  - K-means
  - Hierarchical Clustering - brief description
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

Data Sets from Mixtures

- Probabilistic models for clusters
- Random variable $z \in \{1, \ldots, K\}$
  - Discrete event with PMF: $P(z = i)$
  - Often not observed directly, a latent variable
- Observed variable x, can be continuous
  - Probability depends on z, $p(x|z = i)$
  - One PDF per state $z = i$, called a component
- Distribution of x can be computed via total

Distribution on Data Sets from Mixtures

- Distribution of x can be computed via total probability
  - PDF $p(x) = \sum p(x|z = i)P(z = i)$
  - CDF $F(x_0) = \sum P(x \leq x_0|z = i)P(z = i)$
- Example: Mixture of two Gaussian

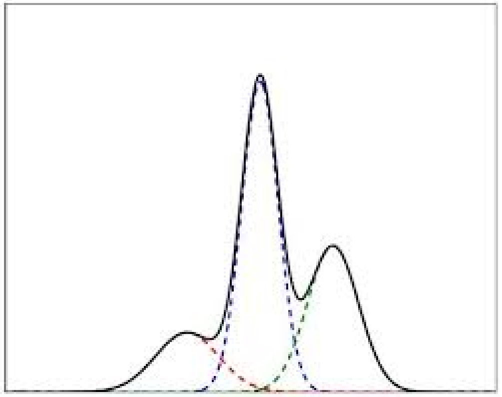
Mixture Models: Examples

- Many data occurs from underlying discrete states
- Example 1: Size of a webpage
  - $z =$ content of the webpage, e.g. number of images
- Example 2: Speech
  - $z =$ phoneme the speaker is saying
- Example 3: Image
  - $x =$ RGB values of a pixel or region of pixels
  - $z =$ one a small number of objects the pixel is part of

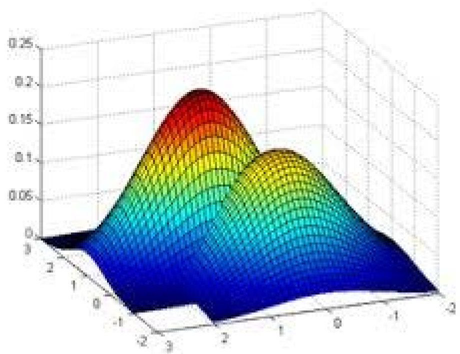
Gaussian Mixture Models

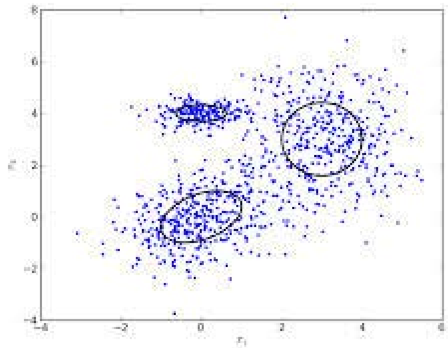
- Each $p(x|z = i)$ is a Gaussian
- Parametrized by:
  - $q_i = P(z = i) =$ Probability of each component
  - $\mu_i = E(x|z = i), P_i = var(x|z = i)$
    mean and variance in each component
- Can be vector valued

Visualizing GMMs



- 1d model with $K = 3$ components





- PDF for 2d GMM with $K = 2$ components
- Random points from a GMM with $K = 3$ components

Expectation and Variance

- Can compute expectation and variance by total probability
  - Expectation: $\mu = E(x) = \sum q_i \mu_i$
  - Variance:

$$var(x) = \sum_i \underbrace{q_i P_i}_{\text{Variance within component}} + \underbrace{q_i(\mu_i - \mu)(\mu_i - \mu)^T}_{\text{Variance between components}}$$

- Proof on board

Estimating the Latent Variable

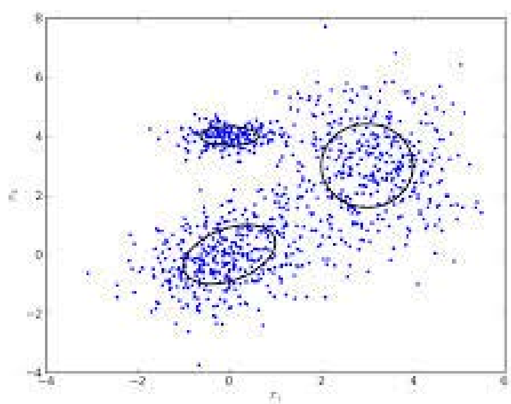
- Given x, can we estimate z?
- Use Bayes' rule:

$$P(z = i|x) = \frac{P(x|z = i)q_i}{\sum_k P(x|z = k)q_k}$$

- Example: Scalar Gaussian
  - Illustration on board

Fitting a Mixture Model

- Given data $x = (x_1, \dots, x_N)$
- Find GMM parameters
  - Mean and variance in each component
  - Probability of each component
- Can be interpreted as "clustering"
- Parametric probabilistic model versus K-means



Maximum Likelihood Estimation

- Unknown parameters in GMM:
$$\theta = (q_1, \ldots, q_K, \mu_1, \ldots, \mu_K, P_1, \ldots, P_K)$$
- Data $x = (x_1, \ldots, x_N)$
- Likelihood of x_n:
$$p(x_n|\theta) = \sum_{k=1}^{K} p(x_n|z_n = k, \theta) P(z_n = k|\theta) = \sum_{k=1}^{K} q_k N(x_n|\mu_k, P_k)$$
- Negative log likelihood of all data
$$L(\theta) = -\ln p(x|\theta) = -\sum_{n=1}^{N} \ln\left[\sum_{i=1}^{K} q_i N(x_n|\mu_i, P_i)\right]$$
- ML estimation:
$$\hat{\theta} = \arg\min_{\theta} L(\theta)$$
- Type equation here.

Outline

- Factor Analysis
- LDA – multiple vectors
- Clustering
  - K-means
  - Hierarchical Clustering - brief description
- Mixture Distributions
- Expectation Maximization Algorithm
- Convergence of EM

Expectation Maximization Algorithm

- Optimization of $L(\theta)$ is hard
  - No simple way to directly optimize
  - Likelihood is non-convex

- Expectation maximization:
  - Simple iterative procedure:
  - Generates a sequence of estimates $\hat{\theta}^0, \hat{\theta}^1, \ldots$
  - Attempts to approach MLE

$$\hat{\theta}^k \to \arg\min_{\theta} L(\theta)$$

To be continued next lecture

UCLA

EM Steps

- E-step: Estimate the latent variables
  - Find the posterior of the latent variables given $\hat{\theta}^k$

$$P(z|x,\theta = \hat{\theta}^k)$$

  - Compute function, Q, auxiliary function

$$Q\left(\theta, \hat{\theta}^k\right) := E\left[\ln p(x,z|\theta) | \hat{\theta}^k\right]$$
$$= \sum_z \ln p(x,z|\theta)\, P(z|x,\theta = \hat{\theta}^k)$$

- M-step: Update parameters

$$\hat{\theta}^{k+1} = \arg\max_{\theta} Q\left(\theta, \hat{\theta}^k\right)$$

E-Step for a GMM: Finding the posterior

- Given parameters q_i, μ_i, P_i
- Find posterior by Bayes rule

$$\gamma_{ni} = P(z_n = i|x) = \frac{P(x_n|z_n = i)q_i}{\sum_k P(x_n|z_n = k)q_k}$$

$$= \frac{N(x_n|\mu_i, P_i)q_i}{\sum_k P(x_n|\mu_k, P_k)q_k}$$

- A "soft" selection

E-Step for a GMM

- Auxilliary function separates

$$Q(\theta, \hat{\theta}^k) = E\left[\ln p(x, z) | \hat{\theta}^k\right]$$

$$= \sum_{i=1}^{K} \sum_{n=1}^{N} \gamma_{ni} \ln P(x_n, z_n = i)$$

$$= \sum_{i=1}^{K} \sum_{n=1}^{N} \gamma_{ni} [\ln q_i + \ln N(x_n | \mu_i, P_i)]$$

M-Step for the GMM

- Maximize $Q\left(\theta, \hat{\theta}^k\right)$
- Update for q_i (proof on board)

$$q_i = \frac{N_i}{\sum_j N_j}, \qquad N_i = \sum_n \gamma_{ni}$$

- Update for μ_i

$$\mu_i = \frac{1}{N_i} \sum_n \gamma_{ni}\, x_n$$

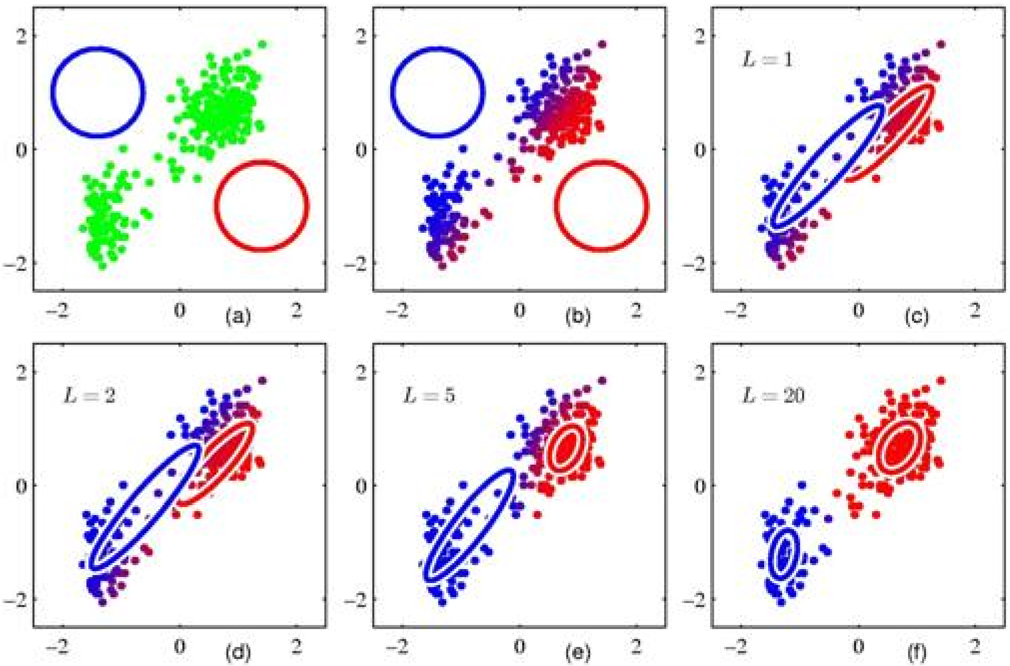
- Update for P_i

$$P_i = \frac{1}{N_i} \sum_n \gamma_{ni}\, (x_n - \mu_i)(x_n - \mu_i)^\wedge T$$

Relation to K means

- EM can be seen as a "soft" version
  - In K-Means: $\gamma_{ni} = 1$ or 0
- Variance
  - In K-means: $P_i = I$
  - In EM, this is estimated
- EM provides "scaling" of various dimensions

EM Illustrated



- Simple example with K=2 clusters
- Dimension = 2
- Convergence from a bad initial condition