

CG, Fall 2011-12

# Clustering gene expression data & the EM algorithm



# How Gene Expression Data

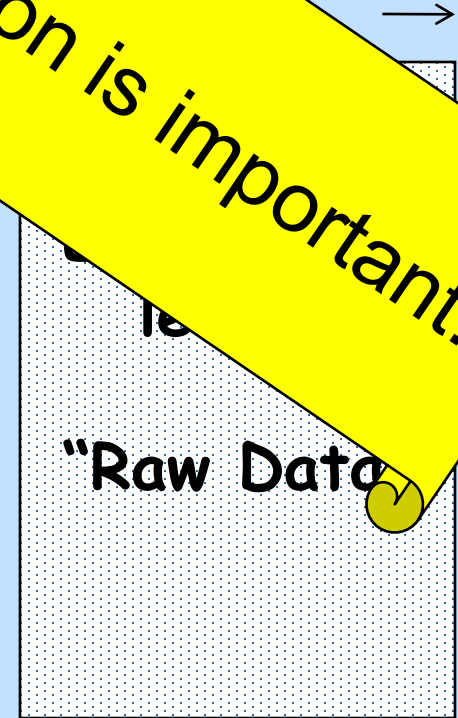
Log

Entries of the Raw Data matrix

- Ratio values
- Absolute values
- ...
- Row = gene's **expression pattern / fingerprint vector**
- Column = experiment/condition's **profile**

Normalization is important!!

genes



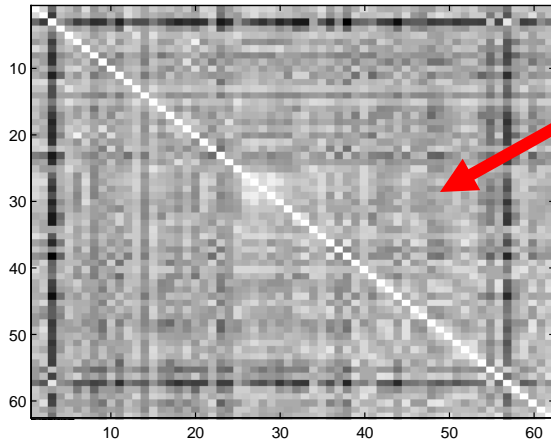
# Data Preprocessing

- **Input:** Real-valued raw data matrix.
- **Compute the similarity matrix** (dot product/correlation/...)
- Alternatively - distances

conditions →

genes ↓

Expression levels,  
"Raw Data"



From the Raw Data matrix we compute the **similarity matrix**  $S$ .  $S_{ij}$  reflects the similarity of the expression patterns of gene  $i$  and gene  $j$ .

# DNA chips: Applications

- Deducing functions of unknown genes (similar expression pattern → similar function)
- Identifying disease profiles
- Deciphering regulatory mechanisms (co-expression → co-regulation).
- Classification of biological conditions
- Drug development
- ...

Analysis requires **clustering** of genes/conditions.

# Clustering: Objective

Group elements (genes) to clusters satisfying:

- **Homogeneity**: Elements inside a cluster are highly similar to each other.
- **Separation**: Elements from different clusters have low similarity to each other.
- Needs formal objective functions.
- Most useful versions are NP-hard.

# The Clustering Bazaar



# Hierarchical clustering



# An Alternative View

Instead of partition to clusters -  
Form a tree-hierarchy of the input  
elements satisfying:

- More similar elements are placed closer along the tree.
- Or: Tree distances reflect element similarity



# Hierarchical Clustering: Average Linkage

Sokal & Michener 58, Lance & Williams 67

- Input: Distance matrix ( $D_{ij}$ )
- Iterative algorithm. Initially each element is a cluster.  $n_r$ - size of cluster  $r$ 
  - Find min element  $D_{rs}$  in  $D$ ; merge clusters  $r,s$
  - Delete elements  $r,s$ ; add new element  $t$  with
$$D_{it}=D_{ti}=\frac{n_r}{(n_r+n_s)}\cdot D_{ir} + \frac{n_s}{(n_r+n_s)}\cdot D_{is}$$
  - Repeat

# A General Framework

Lance & Williams 67

- Find min element  $D_{rs}$ , merge clusters r,s

- Delete elems. r,s, add new elem. t with

$$D_{it}=D_{ti}=\alpha_r D_{ir} + \alpha_s D_{is} + \gamma |D_{ir}-D_{is}|$$

- Single-linkage:  $D_{it}=\min\{D_{ir},D_{is}\}$

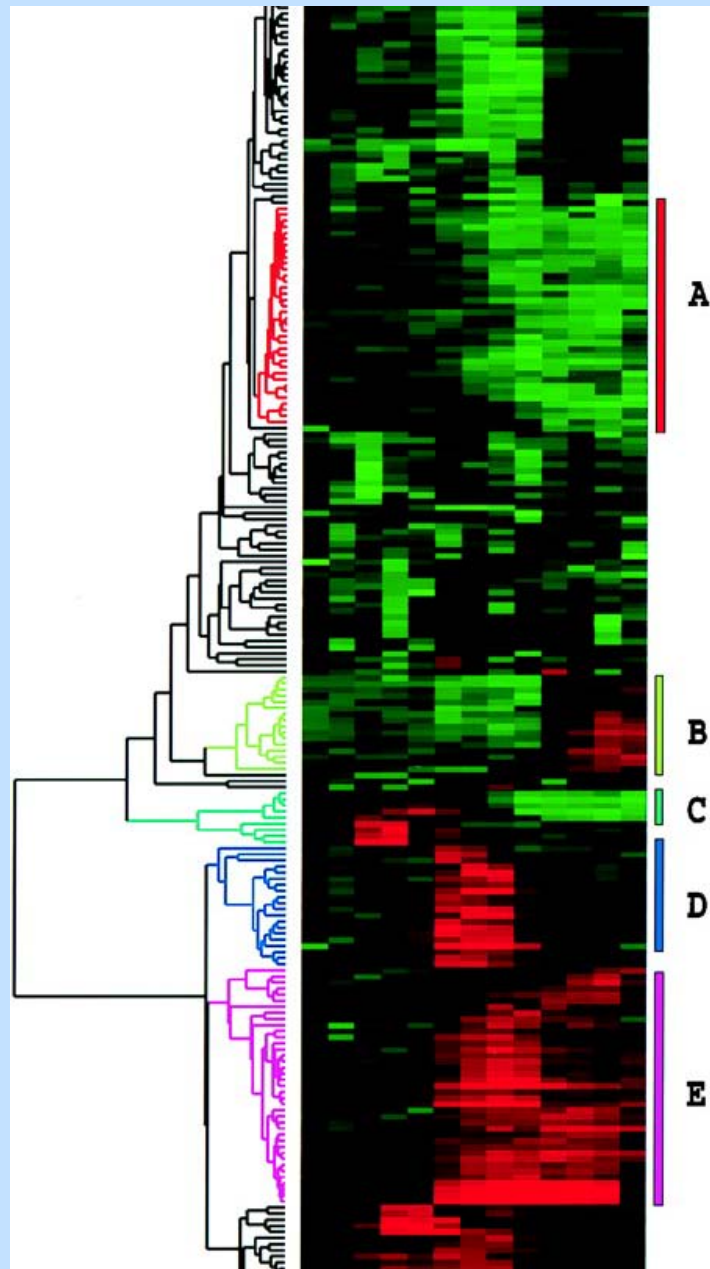
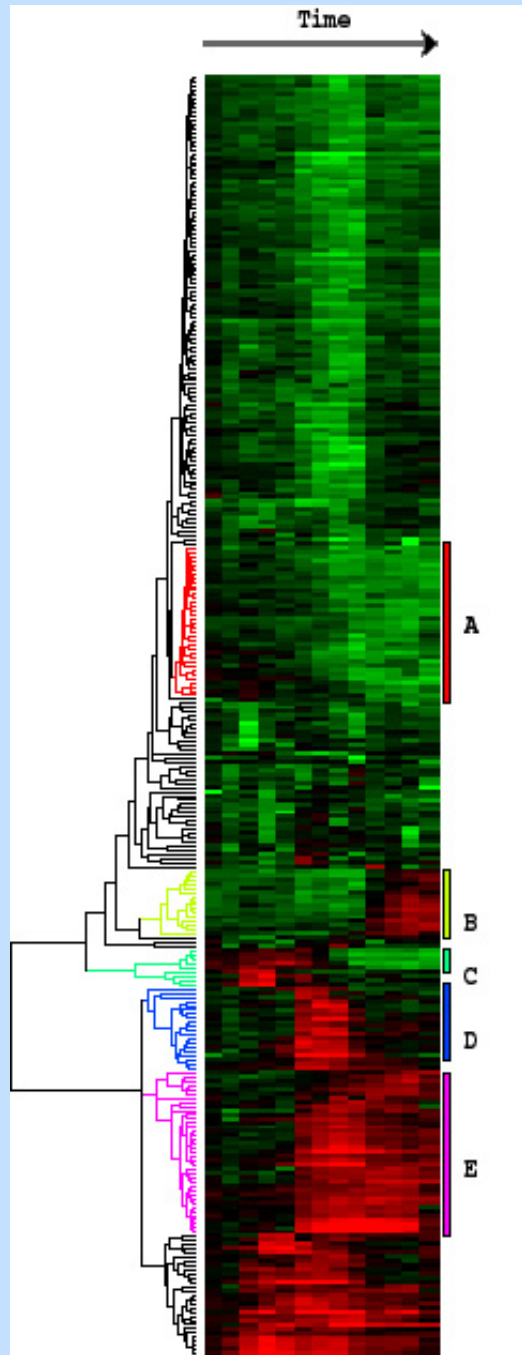
- Complete-linkage:  $D_{it}=\max\{D_{ir},D_{is}\}$

- Note: analogous formulation in terms of similarity matrix (rather than distance)

# Hierarchical clustering of GE data

Eisen et al., PNAS 1998

- Growth response: Starved human fibroblast cells, added serum
- Monitored 8600 genes over 13 time-points
- $t_{ij}$  - fluorescence level of gene  $i$  in condition  $j$ ;  $r_{ij}$  - same for reference
- $s_{ij} = \log(t_{ij}/r_{ij})$
- $S_{kl} = (\sum_j s_{kj} \cdot s_{lj}) / [||s_k|| ||s_l||]$  (cosine of angle)
- Applied average linkage method
- Ordered leaves by increasing element weight: average expression level, time of maximal induction, or other criteria



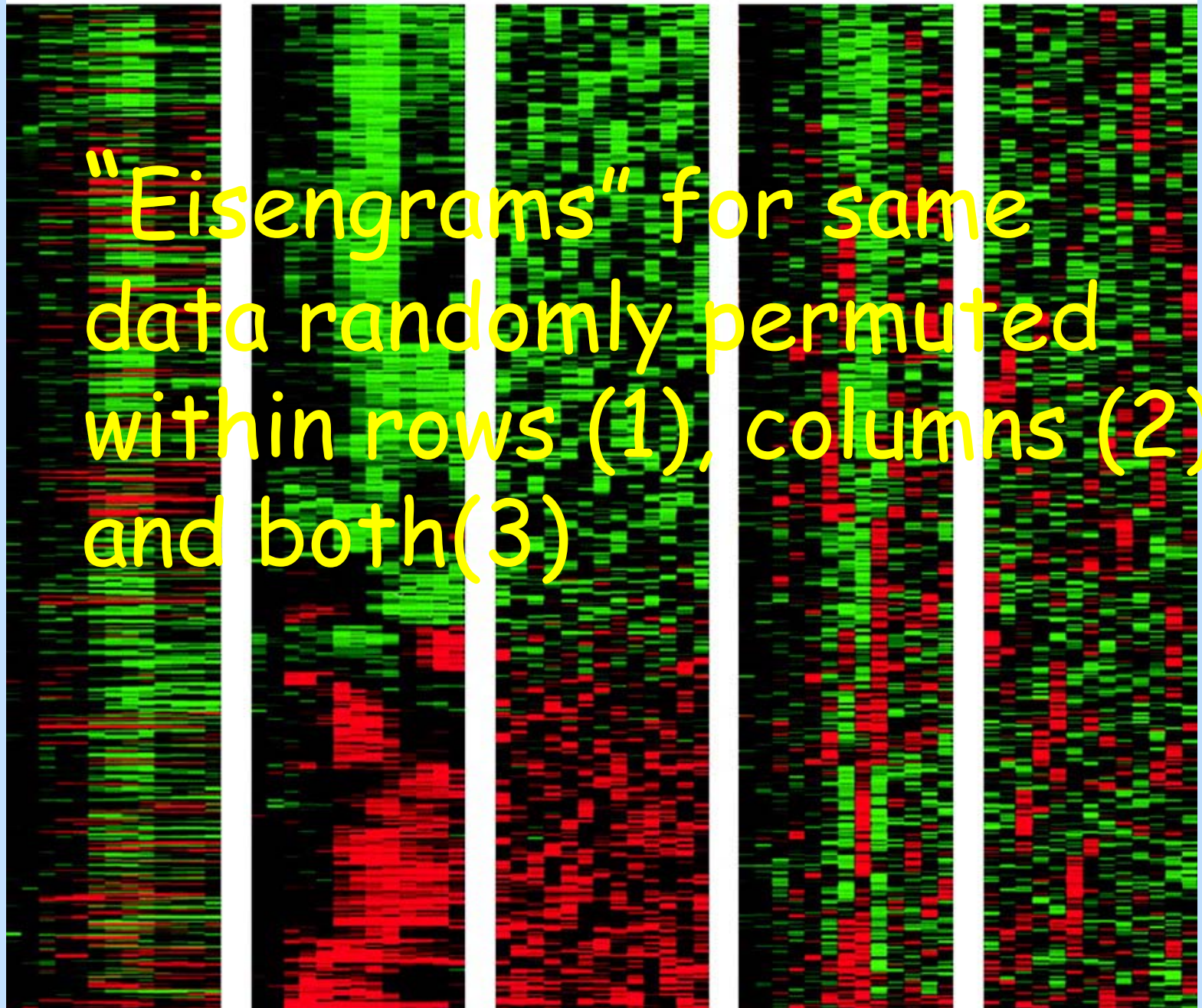
start

clustered

random1

random2

random3



"Eisengrams" for same data randomly permuted within rows (1), columns (2) and both(3)



# Comments

- Distinct measurements of same genes cluster together
- Genes of similar function cluster together
- Many cluster-function specific insights
- Interpretation is a REAL biological challenge

# More on hierarchical methods

- **Agglomerative** vs. the “more natural” **divisive**.
- **Advantages:**
  - gives a single coherent global picture
  - Intuitive for biologists (from phylogeny)
- **Disadvantages:**
  - No single partition; no specific clusters
  - Forces all elements to fit a tree hierarchy

# Non-Hierarchical Clustering



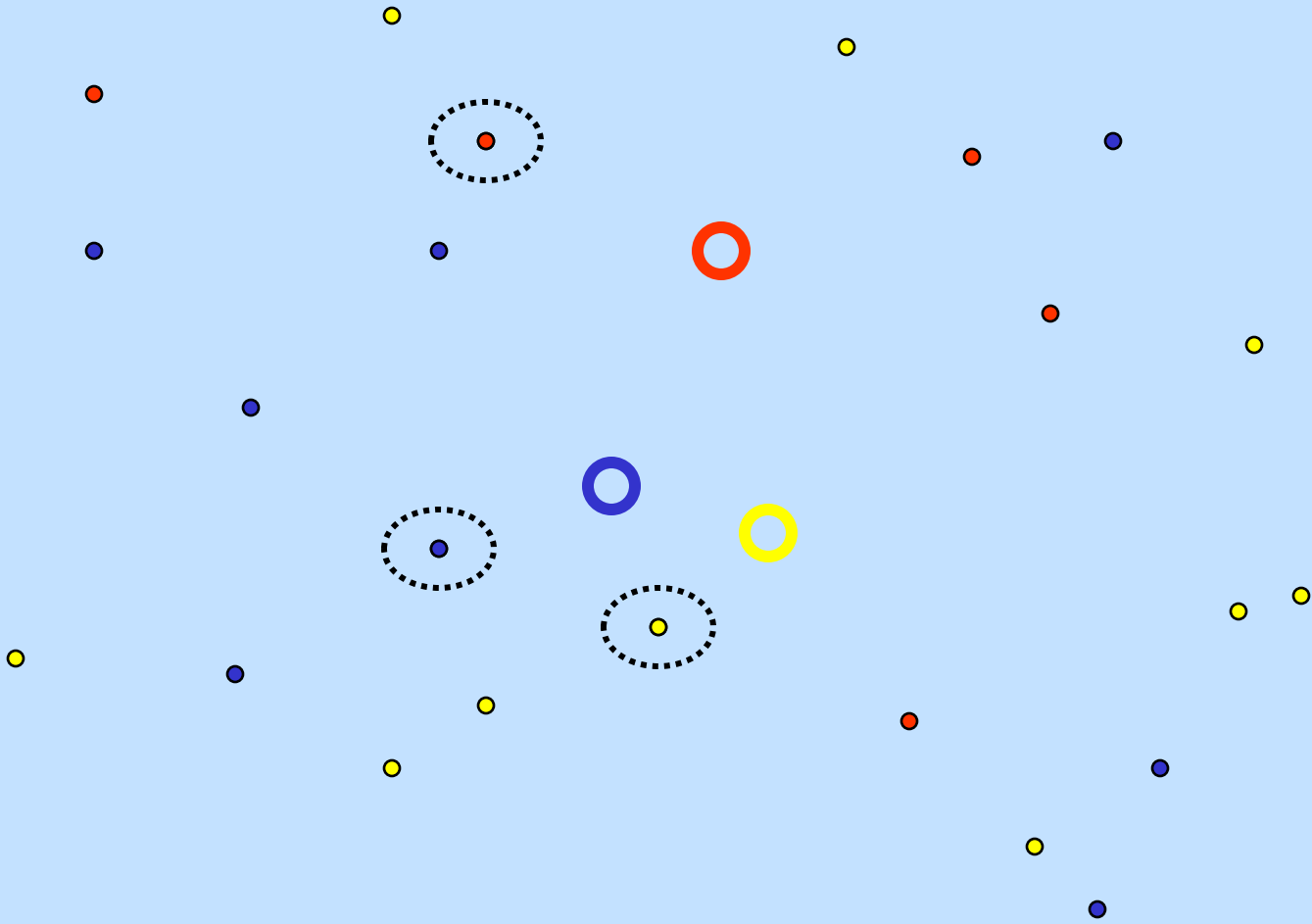
# K-means

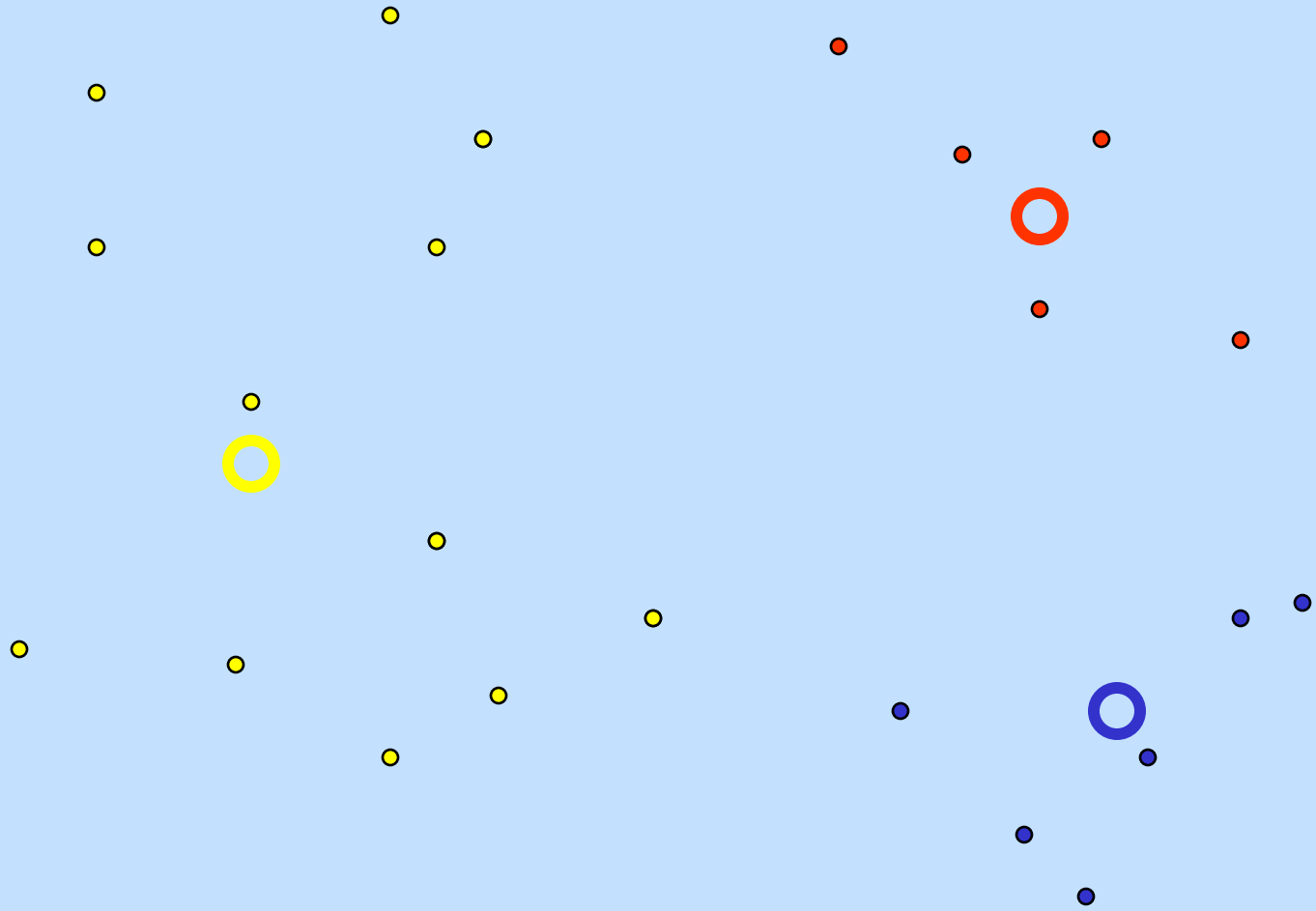
(Lloyd' 57, Macqueen '67)

- Input: vector  $v_i$  for each element  $i$ ;  
#clusters= $k$
- Define a **centroid**  $c_p$  of a cluster  $C_p$  as its average vector.
- Goal: minimize  $\sum_{clusters p} \sum_{i \text{ in cluster } p} d(v_i, c_p)$
- Objective = homogeneity only ( $k$  fixed)
- NP-hard already for  $k=2$ .

# K-means alg.

- Initialize an arbitrary partition  $P$  into  $k$  clusters.
- Repeat the following till convergence:
  - Update centroids (max  $c$ ,  $P$  fixed)
  - Assign each point to its closest centroid (max  $P$ ,  $c$  fixed)
- Can be shown to have poly expected time under various assumptions on data distribution.
- A variant: perform a single best modification (that decreases the score the most).





# A Soft Version

- Based on a probabilistic model of data as coming from a mixture of Gaussians:  $P(z_i = j) = \pi_j$   
 $P(x_i | z_i = j) \sim N(\mu_j, \sigma I)$
- Goal: evaluate the parameters  $\theta$  (assume  $\sigma$  is known).
- Method: apply EM to maximize the likelihood of data.

$$L(\theta) \propto \prod_i \sum_j \pi_j \exp\left(-\frac{d(x_i, \mu_j)^2}{2\sigma^2}\right)$$

# EM, soft version

- Iteratively, compute soft assignment and use it to derive expectations of  $\pi$ ,  $\mu$ :

$$w_{ij}^{(t)} = p(z = j | \mathbf{x}_i, \theta^{(t)}) = \frac{\pi_j^{(t)} p(\mathbf{x}_i | z_i = j, \theta^{(t)})}{\sum_{k=1}^n \pi_k^{(t)} p(\mathbf{x}_i | z_i = k, \theta^{(t)})}$$

$$\pi_i^{(t+1)} = \frac{1}{N} \sum_{i=1}^N w_{ij}^{(t)}$$

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^N w_{ij}^{(t)} \mathbf{x}_i}{\sum_{i=1}^N w_{ij}^{(t)}}$$

# Soft vs. hard clustering

Soft version minimizes:

$$L(\theta) \propto \prod_i \sum_j \pi_j \exp\left(-\frac{d(x_i, \mu_j)^2}{2\sigma^2}\right)$$

If we assume that each element is in one cluster (hard assignment) then:

$$-\log \mathbb{E}(\theta) \propto \sum_i d(x_i, \mu_{c(i)})^2$$

This is exactly the k-means criterion!

# Expectation-maximization: The probabilistic setting

Input: data  $x$  coming from a probabilistic model with hidden information  $y$

Goal: Learn the model's parameters so that the likelihood of the data is maximized.

Example: a mixture of two Gaussians

$$P(y_i = 1) = p_1 ; P(y_i = 2) = p_2 = 1 - p_1$$

$$P(x_i | y_i = j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x_i - \mu_j)^2}{2\sigma^2}\right)$$




# The likelihood function

$$P(y_i = 1) = p_1 ; P(y_i = 2) = p_2 = 1 - p_1$$

$$P(x_i | y_i = j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x_i - \mu_j)^2}{2\sigma^2}\right)$$

$$L(\theta) = \prod_i P(x_i | \theta) = \prod_i \sum_j P(x_i, y_i = j | \theta)$$

$$\log L(\theta) = \sum_i \log \left( \sum_j \frac{p_j}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x - \mu_j)^2}{2\sigma^2}\right) \right)$$




# The EM algorithm

**Goal:  $\max \log P(\mathbf{x}|\theta) = \log (\sum P(\mathbf{x}, \mathbf{y}|\theta))$**

Assume we have a model  $\theta^t$  which we wish to improve.

Note:  $P(\mathbf{x}|\theta) = P(\mathbf{x}, \mathbf{y}|\theta) / P(\mathbf{y}|\mathbf{x}, \theta)$

$$P(y|x, \theta^t) \cdot \log P(x|\theta) = P(y|x, \theta^t) \cdot \log P(x, y|\theta) - P(y|x, \theta^t) \cdot \log P(y|x, \theta)$$
$$\sum_y P(y|x, \theta^t) \cdot \log P(x|\theta) = \sum_y P(y|x, \theta^t) \cdot \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \cdot \log P(y|x, \theta)$$

$$\log P(x|\theta) = \sum_y P(y|x, \theta^t) \cdot \log P(x, y|\theta) - \sum_y P(y|x, \theta^t) \cdot \log P(y|x, \theta)$$

$$\log P(x|\theta^t) = \sum_y P(y|x, \theta^t) \cdot \log P(x, y|\theta^t) - \sum_y P(y|x, \theta^t) \cdot \log P(y|x, \theta^t)$$

$$\Delta = \underbrace{Q(\theta|\theta^t) - Q(\theta^t|\theta^t)}_{\text{Constant}} + \sum_y P(y|x, \theta^t) \cdot \log \frac{P(y|x, \theta^t)}{P(y|x, \theta)} \geq 0$$



# The EM algorithm (cont.)

Main component:

$$Q(\theta | \theta^t) = \sum_y P(y | x, \theta^t) \cdot \log P(x, y | \theta) = E^t [\log P(x, y | \theta)]$$

is the expectation of  $\log P(x, y | \theta)$  over the distribution of  $y$  given by the current parameters  $\theta^t$

The algorithm:

- E-step: Calculate the Q function
- M-step: Maximize  $Q(\theta | \theta^t)$  with respect to  $\theta$



# Application to the mixture model

$$Q(\theta | \theta^t) = \sum_y P(y | x, \theta^t) \cdot \log P(x, y | \theta) = E^t [\log P(x, y | \theta)]$$

$$P(x, y | \theta) = \prod_i P(x_i, y_i = j | \theta) = \prod_i \prod_j P(x_i, y_i = j | \theta)^{y_{ij}}$$

$$y_{ij} = \begin{cases} 1 & y_i = j \\ 0 & y_i \neq j \end{cases}$$

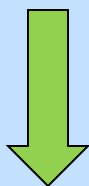
$$\log P(x, y | \theta) = \sum_i \sum_j y_{ij} \log P(x_i, y_i = j | \theta)$$

$$E^t[\log P(x, y | \theta)] = \sum_i \sum_j E^t[y_{ij}] \log P(x_i, y_i = j | \theta)$$

# Application (cont.)

$$E^t[\log P(x, y | \theta)] = \sum_i \sum_j E^t[y_{ij}] \log P(x_i, y_i = j | \theta)$$

$$w_{ij}^t := E^t[y_{ij}] = P(y_{ij} = 1 | x_i, \theta^t) = \frac{P(x_i, y_i = j | \theta^t)}{\sum_j P(x_i, y_i = j | \theta^t)}$$



$$Q(\theta | \theta^t) = \sum_i \sum_j w_{ij}^t \left( \log \frac{1}{\sqrt{2\pi}} - \log \sigma + \log p_j - \frac{(x_i - \mu_j)^2}{2\sigma^2} \right)$$