Some additions: **Cluster Analysis**

(F. Chiaromonte)

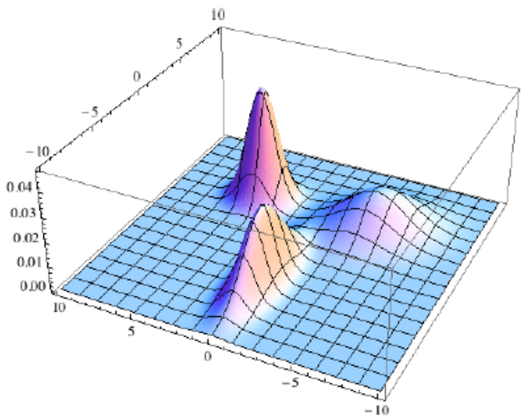
Clustering with Gaussian Mixtures (model-based soft partitioning)

Postulated **stochastic mechanism**:
each data point in $\mathbf{R}^p$ is drawn from

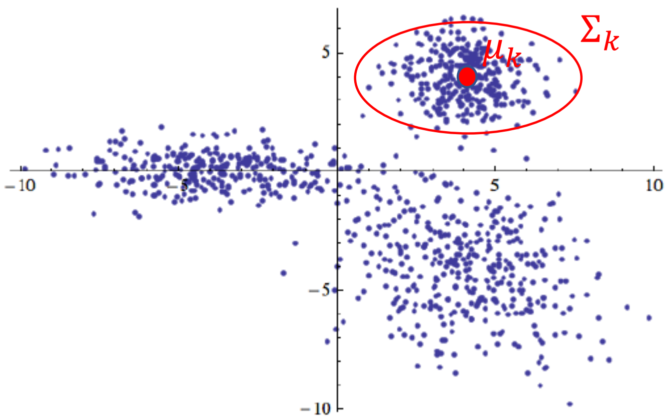
$$f(x) = \sum_{k=1}^{K} \pi_k f(x|z=k) = \sum_{k=1}^{K} \pi_k \underbrace{\varphi(x;\mu_k,\Sigma_k)}_{\text{Gaussian components}}$$

prior probabilities $Pr\{z=k\}$

latent component labels



(a) A probability distribution on $\mathbb{R}^2$.



(b) Data sampled from this distribution.

$$\theta = \{(\pi_k;\mu_k;\Sigma_k), k = 1 \dots K\}$$

PARAMETERS

- Prevalence
- Center (location)
- Shape and orientation

of each cluster

$$P = \{\mathrm{p}_{i,k}, i = 1 \dots n, k = 1 \dots K\}$$

POSTERIOR PROBABILITIES
For each data point and component
$Pr\{z_i=k \mid x_i\}$
non-negative, rows add up to 1

The Expectation-Maximization (EM) algorithm

A very important algorithm to fit models comprising latent (unobservable) variables – here, the component labels.

Produces estimates for

- all parameters – prevalence, location and shape/orientation of the clusters
- and for the posterior probabilities – which express a soft, probabilistic partition of the data points

Similar to the *k-means algorithm*, EM iteratively seeks (an) optimum of an objective function; namely, a maximum for the likelihood on the log scale. After an initialization (of ~~centroids~~ parameters or ~~memberships~~ posterior probabilities), the algorithm iterates between two steps until convergence:

E-step: compute the expectation of the log-likelihood, evaluated given the observable variables and the current parameter estimates

$$h_t(\theta) = E[logL(\theta|\boldsymbol{X}, \boldsymbol{Z})|\boldsymbol{X}, \theta_t]$$

(compute the ~~memberships~~ posterior probabilities, given the current ~~centroids~~ parameters)

M-step: compute parameter estimates maximizing the expected log-likelihood found in the E step.

$$\theta_{t+1} = argmax_\theta \, h_t(\theta)$$

(compute the ~~centroids~~ parameters given the current ~~memberships~~ posterior probabilities)

The Rand Index

Comparing two partitions. Evaluating a clustering solution against a known partition.

- Set of elements {1, 2… n} (the data points)
- A first partition ***A***, e.g., as generated by a clustering algorithm (in r groups)
- A second partition ***B***, e.g., known and used for benchmarking ***A*** (in s groups)

Share of agreement between the two partitions, Rand Index $0 \leq \mathrm{RI} \leq 1$

$$RI = \frac{\#\{(i,j) together\ in\ \boldsymbol{A}\ and\ in\ \boldsymbol{B}\} + \#\{(i,j)\ not\ together\ in\ \boldsymbol{A}\ and\ in\ \boldsymbol{B}\}}{\binom{n}{2}}$$

If r and s are different, maximal agreement cannot be 1. Adjusted Rand index

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}$$

CONTINGENCY TABLE

n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
b_1	b_2	$\cdots$	b_s	