Hierarchical agglomerative clustering algorithm

Input

- ($n_x p$) data matrix, **X**, with n p-dimensional multivariate observations

Init

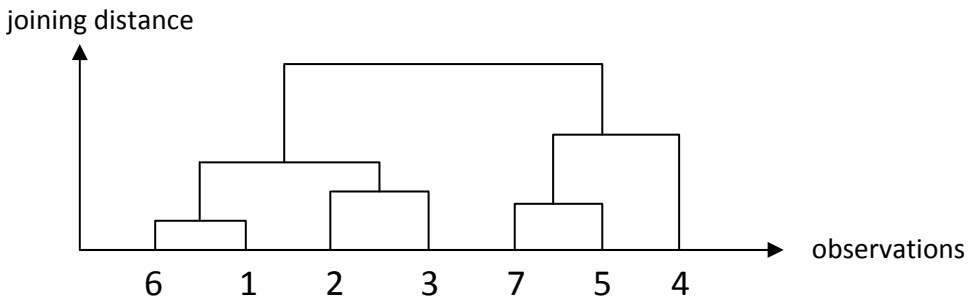
- Calculate ($n_x n$) symmetrical zero-diagonal distance matrix, **D**, where entry d_{ik} is the distance (Euclidean or other) between observations i and k
- Define n initial clusters labeled i=1,...,n each holding one observation i

Loop

- for j = 1:n-1
- find minimal entry $d_{ik} > 0$ in **D** (choose arbitrarily if there are ties)
- combine clusters i and k into cluster ik
- delete rows and columns in **D** corresponding to clusters i and k
- add a new row and column to **D** with the distance between ik and other clusters (using the chosen linkage method for inter-cluster distance)
- end

Output

- Dendrogram (tree diagram)



# of clusters ?

- Make suitable horizontal cut in dendrogram
- Look for big "jumps" in joining distance ~ significant separation of clusters

Measure of global fit of clustering

- Cophenetic correlation coefficient between original distances in **D** and joining distances in resulting dendrogram

Problems with algorithm

- Complexity $O(n^3)$
- # of clusters not explicitly output from algorithm
- Non-reversible clustering proces in algorithm

Non-hierarchical "K-means" clustering algorithm

Input

- ($n_x p$) data matrix, $\mathbf{X}$, with n p-dimensional multivariate observations $\mathbf{x}_1,...,\mathbf{x}_n$
- Definition of desired/assumed # of clusters, K

Init

- Define K initial clusters $S_1,...,S_K$ (disjoint and complete subsets of $\mathbf{x}_1,...,\mathbf{x}_n$)
- Calculate the initial centroids (sample means), $\mathbf{m}_1,...,\mathbf{m}_K$ of $S_1,...,S_K$ as $\mathbf{m}_i = \sum_{\mathbf{x}_j \in S_i} \mathbf{x}_j / N_{S_i},\ i = 1, \dots, K$
- The initialization can, e.g., either be done by arbitrarily partitioning the observations into K sets, or even simpler by defining K arbitrary p-dimensional vectors $\mathbf{m}_1,...,\mathbf{m}_K$

Loop

- repeat
- reassignments: for j = 1:n
- assign $\mathbf{x}_j$ to S_i iff $\|\mathbf{x}_j - \mathbf{m}_i\| \leq \|\mathbf{x}_j - \mathbf{m}_k\|, \forall k \neq i$
- (reassign observation to nearest centroid)
- (choose arbitrarily if there are ties)
- end
- update centroids: for i = 1:K
- $\mathbf{m}_i = \sum_{\mathbf{x}_j \in S_i} \mathbf{x}_j / N_{S_i}$
- end
- until no more reassignments can be done

Output

- K clusters $S_1,...,S_K$ (disjoint and complete subsets of $\mathbf{x}_1,...,\mathbf{x}_n$)

Pros and Cons

- \+ ($n_x n$) distance matrix D is not used, only the n observations
- \+ Complexity O(nKI), where I is number of iterations
- \- # of clusters must be predefined
- \- "some" dependency on initial clustering
- \- algorithm performs best for clusters of approximately same size

---

Measure of global fit of clustering

- The algorithm finds a local minimum (not necessarily the global minimum) of the within-cluster sum of squares $SS_W = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \mathbf{m}_i\|^2$
- This objective function can be used to compare different choices of K and different initial clusterings

---

Gaussian Mixture Model clustering algorithm

Input

- ($n_x p$) data matrix, **X**, with n p-dimensional multivariate observations $\mathbf{x}_1,\ldots,\mathbf{x}_n$
- Definition of desired/assumed # of clusters, K

Model

- Observations (the n rows in **X**) are supposed to be generated by a gaussian mixture model of the form
  $f_X(\mathbf{x}) = \sum_{k=1}^{K} p_k f_{Y_k}(\mathbf{x}), \quad \mathbf{x} = [x_1, \ldots, x_p]^T$
  where the prior probabilities obeys $\sum_{k=1}^{K} p_k = 1, \ \forall p_k \geq 0$
  and the individual MVN components obeys $Y_k \sim N_p(\mu_k, \Sigma_k)$

# of unknown parameters

- $K - 1$ prior probabilities, p_k (since they sum to 1)
- $K \cdot p$ mean values in μ_k
- $K \cdot \frac{p(p+1)}{2}$ unique variances/covariances in Σ_k
- giving a total of $N_{par}(K) = K \cdot \left[1 + \frac{p(p+3)}{2}\right] - 1$ unknown parameters

Maximum Likelihood estimation of parameters

- The likelihood of independent observations $\mathbf{x}_1,\ldots,\mathbf{x}_n$ is
  $$L\left(\{p_k\}_{k=1}^{K}, \{\mu_k\}_{k=1}^{K}, \{\Sigma_k\}_{k=1}^{K}\right) = \prod_{j=1}^{n} f_X\left(\mathbf{x}_j | \{p_k\}_{k=1}^{K}, \{\mu_k\}_{k=1}^{K}, \{\Sigma_k\}_{k=1}^{K}\right)$$
  $$= \prod_{j=1}^{n}\left[\sum_{k=1}^{K} p_k f_{Y_k}(\mathbf{x}_j | \mu_k, \Sigma_k)\right]$$
  where $f_{Y_k}(\mathbf{x}_j | \mu_k, \Sigma_k)$ is the usual p-dimensional MVN pdf
- Maximization of L by equating $\frac{\partial L}{\partial u} = 0$, for all unknown parameters "u"
- Numerical maximization is done by specialized SW, e.g. MATLAB
  giving $L_{max} = L\left(\{\hat{p}_k\}_{k=1}^{K}, \{\hat{\mu}_k\}_{k=1}^{K}, \{\hat{\Sigma}_k\}_{k=1}^{K}\right)$
- Only a local maximum is found, no guarantee for global maximum, therefore often several runs are done using different initializations
- If optimization for different values of K are to be considered, the different numbers of estimated parameters are taken into account by "penalizing" the found values of L_{max} for each value of K, e.g.
  Akaike Information Criterion maximizes $AIC = 2 \log L_{max} - 2 N_{par}(K)$
  Bayesian Information Criterion maximizes $BIC = 2 \log L_{max} - N_{par}(K) \log n$

Calculate posterior probabilities for observations $\mathbf{x}_j$, $j = 1, \ldots, n$

- $P\left(\text{cluster \# } k \,\middle|\, \mathbf{x}_j\right) = \frac{\hat{p}_k f_{Y_k}(\mathbf{x}_j | \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{i=1}^{K} \hat{p}_i f_{Y_i}(\mathbf{x}_j | \hat{\mu}_i, \hat{\Sigma}_i)}, \quad k = 1, \ldots, K$

Assign observations to clusters using MAP (Maximum Aposteriori)

- Estimated cluster for $\mathbf{x}_j = \arg\max_{k=1,\ldots,K} P\left(\text{cluster \# } k \,\middle|\, \mathbf{x}_j\right)$