Model Based Clustering

Model-based clustering

- *Model-based clustering* assumes that the data were generated by a model and tries to recover the original model from the data. The model that we recover from the data then defines clusters and an assignment of objects to clusters.

A finite mixture model

Given data $\mathbf{y}$ with independent multivariate observations $\mathbf{y}_1, \ldots, \mathbf{y}_n$, the likelihood for a mixture model with G components is

$$\mathcal{L}_{MIX}(\theta_1, \ldots, \theta_G; \tau_1, \ldots, \tau_G \mid \mathbf{y}) = \prod_{i=1}^{n} \sum_{k=1}^{G} \tau_k f_k(\mathbf{y}_i \mid \theta_k), \quad (1)$$

- G: the number of groups (clusters)
- τ_k : the (prior) probability an object belongs to the kth group
- f_k the density of the kth group, with parameters θ_k

Common example

- f_k is multivariate normal ϕ_k
- θ_k is (μ_k , Σ_k)

$$\phi_k(\mathbf{y}_i \mid \mu_k, \Sigma_k) \equiv \frac{\exp\{-\frac{1}{2}(\mathbf{y}_i - \mu_k)^T \Sigma_k^{-1}(\mathbf{y}_i - \mu_k)\}}{\sqrt{\det(2\pi\Sigma_k)}}$$

- That is, the kth cluster centers at μ_k
- And its shape, orientation and tightness described by Σ_k

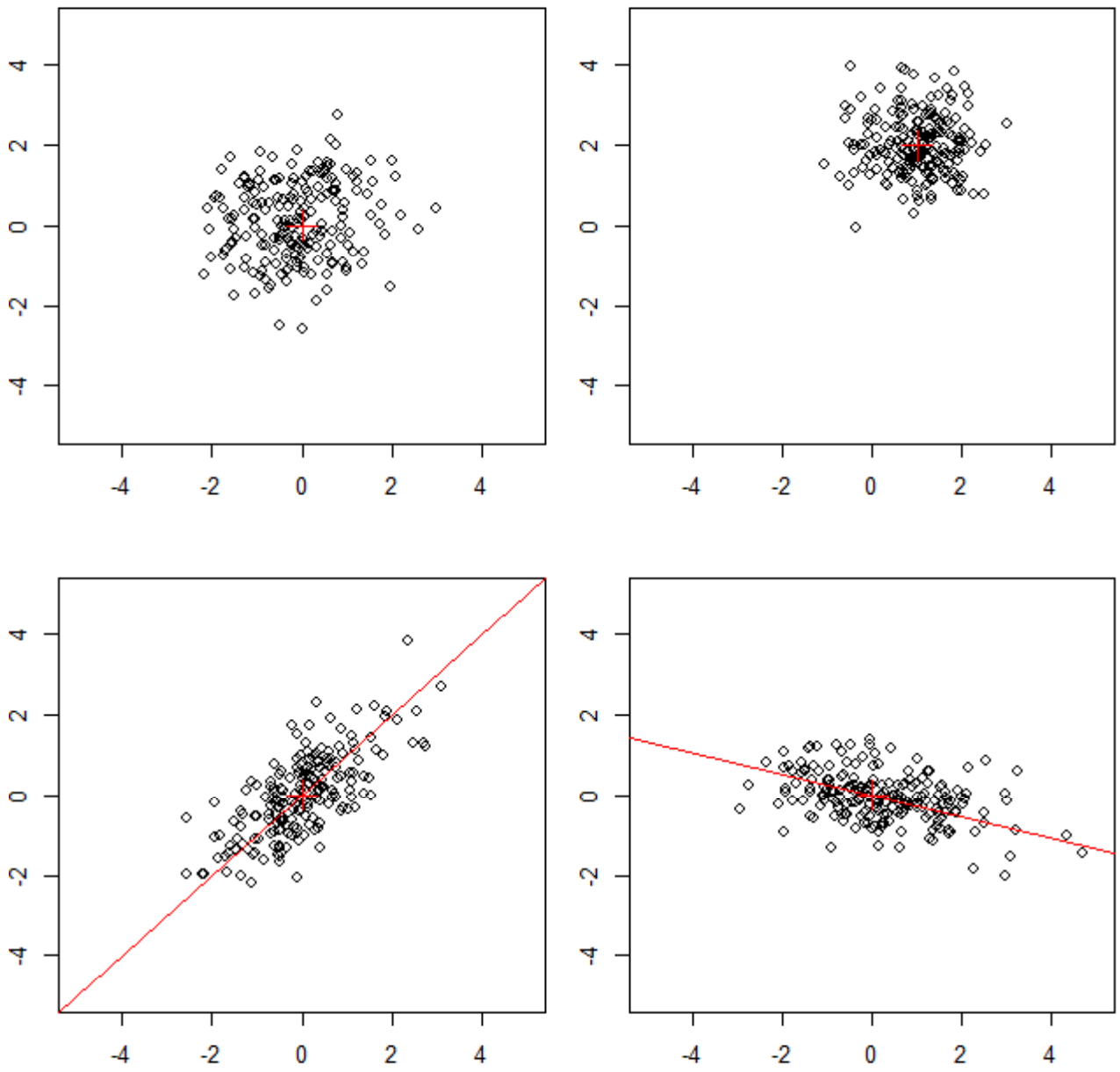
Some Special cases

- $\Sigma_k = \lambda I$: all groups are spherical and of the same tightness
  - Some references say they are of the same “size”, technically this is correct when you interpret it correctly
- $\Sigma_k = \lambda_k I$ all groups are spherical but of different tightness
- $\Sigma_k = \Sigma$: all groups share the same shape (variance-covariance structure) and size
- Σ_k : each group can have different size and shape

examples

```
library(MASS)
mu=c(0,0)
sigma=diag(2)
N=200
## sphere
x1=mvrnorm(N,mu,sigma)
mu2=mu+c(1,2)
## sphere, center moved
x2=mvrnorm(N,mu2,sigma/2)
sigma3=matrix(c(1,.7,.7,1),2,2)
## ellipse
x3=mvrnorm(N,mu,sigma3)
## rotate its angle
rotate=function(sigma,theta){
  R=matrix(c(cos(theta),-
      sin(theta),sin(theta),cos(theta)),2,2)
  R%*%sigma%*%t(R)}
sigma4=rotate(sigma3,pi/3)
x4=mvrnorm(N,mu,sigma4)
```

```
par(mfrow=c(2,2),mai=c(.5,.3,.3,.2))
plot(x1,xlim=c(-5,5),ylim=c(5,5))
points(mu[1],mu[2],col=2,pch=3,cex=2)
plot(x2,xlim=c(-5,5),ylim=c(-5,5))
points(1,2,col=2,pch=3,cex=2)
plot(x3,xlim=c(-5,5),ylim=c(-5,5))
points(0,0,col=2,pch=3,cex=2)
abline(0,1,col=2)
plot(x4,xlim=c(-5,5),ylim=c(-5,5))
points(0,0,col=2,pch=3,cex=2)
abline(0,sin(pi/4-pi/3)/cos(pi/4-
      pi/3),col=2)
```


4
2
0
-2
-4
-4
-2
0
2
4

A general framework

$$\Sigma_k = \lambda_k D_k A_k D_k^T$$

- Eigen vectors in D_k control the orientation
- Eigen values in A_k control the shape
- Size(tightness) by λ_k control the volume

Latent class model and EM algorithm

- Expectation-maximization
- Consider n observations (potentially multivariate) y_i that comes from a class defined by z_i

$$z_{ik} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ belongs to group } k \\ 0 & \text{otherwise.} \end{cases}$$

  - For the first class z=c(1,0,0,…,0)
  - For the third class z=c(0,0,1,0,..0)
- If we know z=c(0,0,1,0,..0)
  - $f(y_i | z_{i3=1}) = f_3(y_i)$

  So in general we have $\prod_{k=1}^{G} f_k(\mathbf{y}_i \mid \theta_k)^{z_{ik}}$

- The complete data (including the unobserved latent z) likelihood

$$L_c(y_i, z_i \mid \theta) = \prod_{i=1}^{n} f(y_i, z_i \mid \theta)$$

- The observed data likelihood

$$L_o(y_i \mid \theta) = \int L(y_i, z_i \mid \theta) dz$$

- The E step computes the conditional expectation of the log L_c given the observed data and the current parameter estimates
- The M step maximizes that expectation

$$\prod_{k=1}^{G} f_k(\mathbf{y}_i \mid \theta_k)^{z_{ik}}$$

$$l(\theta_k, \tau_k, z_{ik} \mid \mathbf{x}) = \sum_{i=1}^{n} \sum_{k=1}^{G} z_{ik} \log[\tau_k f_k(\mathbf{y}_i \mid \theta_k)] \quad *$$

The E step: expectation of z_{ik} is $\dfrac{\hat{\tau}_k f_k(\mathbf{y}_i \mid \hat{\theta}_k)}{\sum_{j=1}^{G} \hat{\tau}_j f_j(\mathbf{y}_i \mid \hat{\theta}_j)}$

The M step: maximize * by after plugging in the expectation of z_{ik}

The connection to K-means

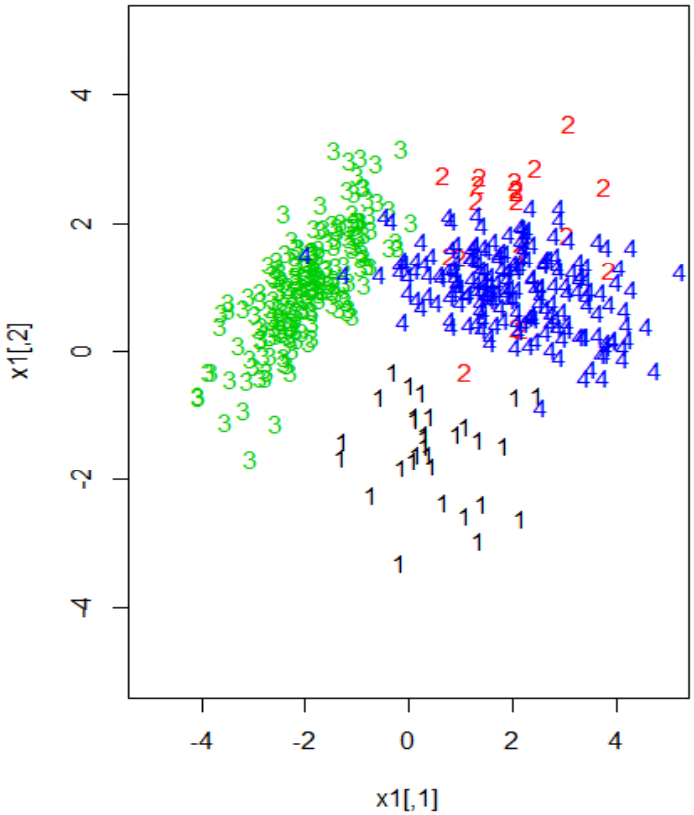
- In the E step, we compute a conditional expectation of z_{ik}: that is, given the current parameter values, what do we think the probabilities that the ith object belonging to each of the k clusters are. Though z_{ik} is discrete(multinomial with size 1), its expectation is continuous.
- If we
  - Assume $\Sigma_k = \lambda I$
  - Replace the expectation with our “best guess”, that is, assign the ith object to the k-th cluster, then iterate

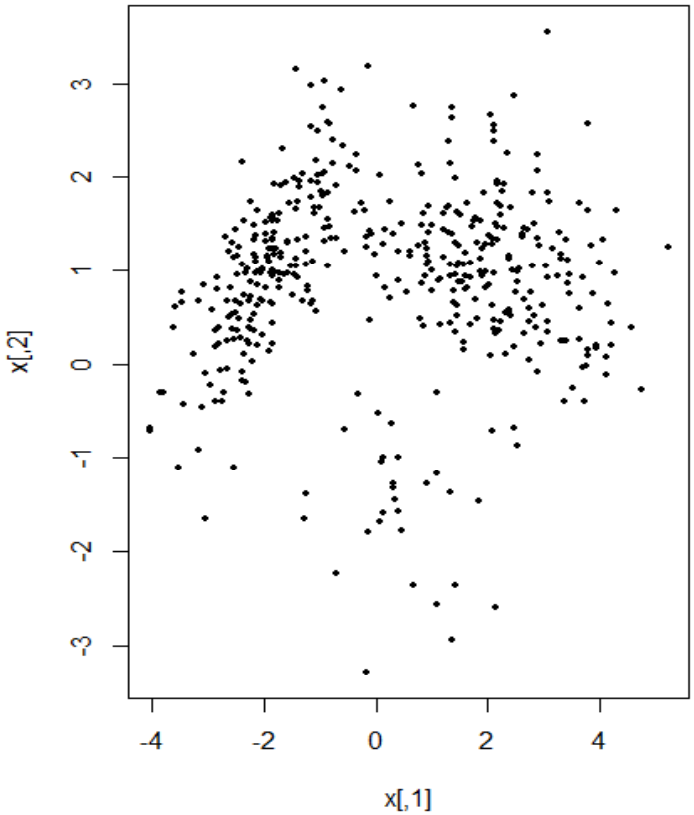
  It becomes k-means

Truth: complete data

$y_i=(y_{i1},y_{i2})$ **Z: color**

observed : $y_i=(y_{i1},y_{i2})$





Data generation

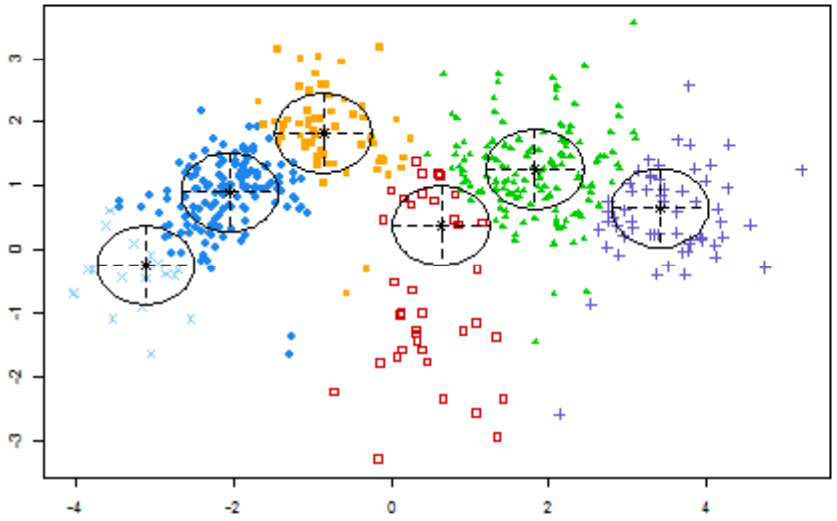
```
set.seed(2014)
x1=mvrnorm(30,c(0,-1.5),sigma)
mu2=mu+c(1,2)
x2=mvrnorm(20,c(2,2),sigma/2)
sigma3=matrix(c(1,.7,.7,1),2,2)
x3=mvrnorm(N,c(-2,1),sigma3*.7)
x4=mvrnorm(N,c(2,1),sigma4)
par(mfrow=c(1,2))
plot(x1,xlim=c(-10,10)/2,ylim=c(-5,5),pch="1")
points(x2,xlim=c(-5,5),ylim=c(-5,5),col=2,pch="2")
points(x3,xlim=c(-5,5),col=3,pch="3")
points(x4,col=4,pch="4")
x=rbind(x1,x2,x3,x4)
plot(x， pch=16,cex=.5）
```

Model based clustering

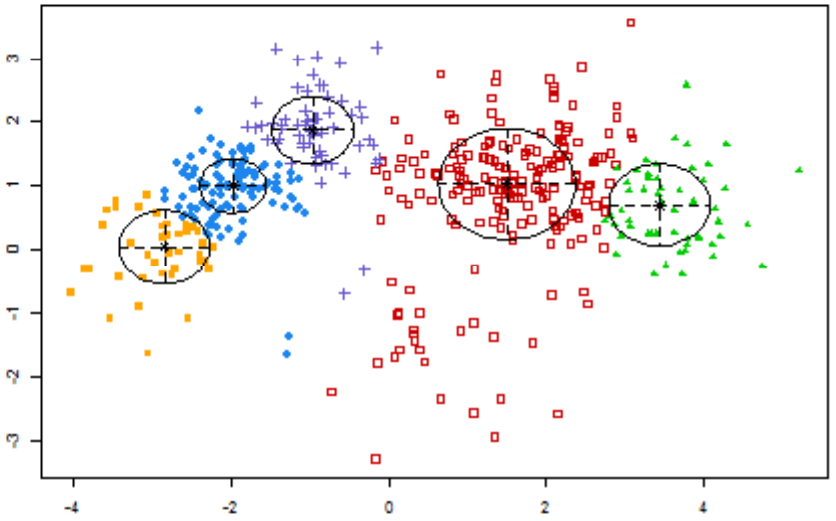
```
# equal variance, spherical
m0=Mclust(x,modelNames="EII")
#spherical, unequal volume
m1=Mclust(x,modelNames="VII")
#ellipsoidal, equal volume, shape, and orientation
m2=Mclust(x,modelNames="EEE")
#ellipsoidal, varying volume, shape, and orientation
m3=Mclust(x,modelNames="VVV")

par(mfrow=c(2,2))
par(cex=.5)
mclust2Dplot(x,parameters=m0$parameters,z=m0$z,wh
      at = "classification", identify = TRUE)
mclust2Dplot(x,parameters=m1$parameters,z=m1$z,wh
      at = "classification", identify = TRUE)
mclust2Dplot(x,parameters=m2$parameters,z=m2$z,wh
      at = "classification", identify = TRUE)
mclust2Dplot(x,parameters=m3$parameters,z=m3$z,wh
      at = "classification", identify = TRUE)
```

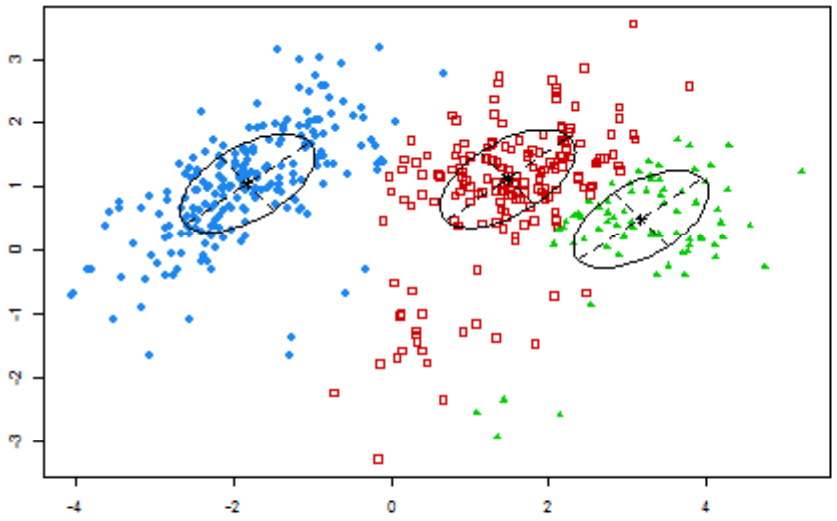
EII



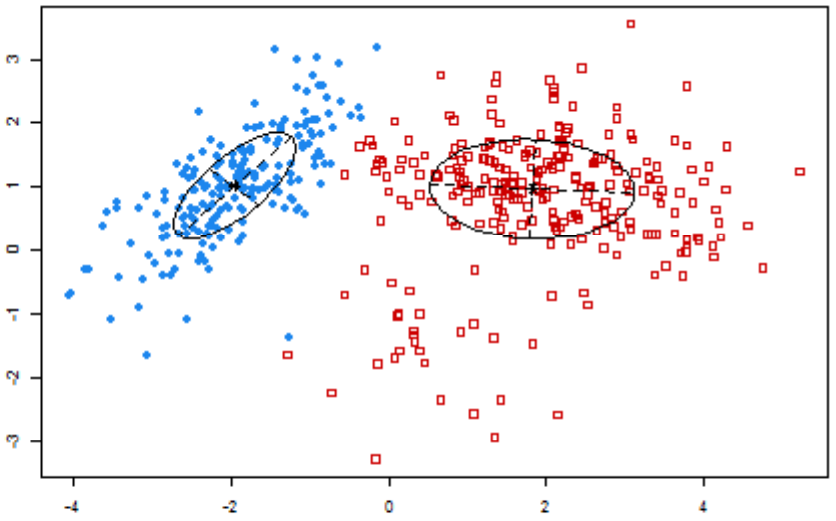
VII



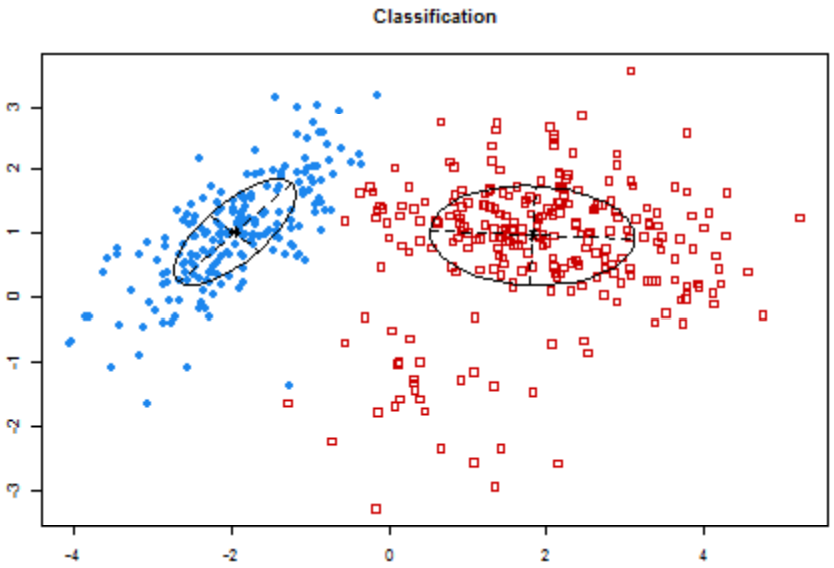
EEE

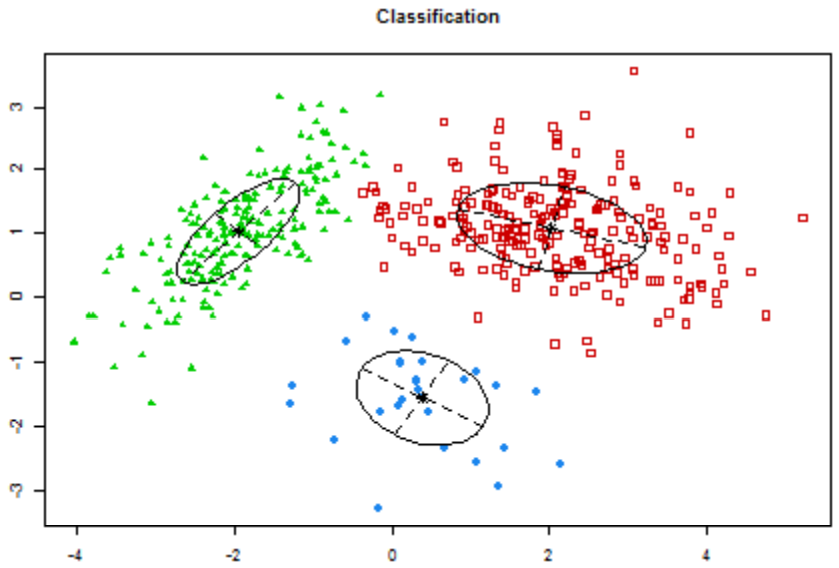


VVV



```
m4=Mclust(x)
mclust2Dplot(x,parameters=m3$parameters,z=m3$z,what = "classification", identify = TRUE)
mclust2Dplot(x,parameters=m4$parameters,z=m4$z,what = "classification", identify = TRUE)
 m4b$BIC
        EII       VII       EEI       VEI       EVI       VVI       EEE       EEV       VEV       VVV
1 -3521.545 -3521.545 -3275.289 -3275.289 -3275.289 -3275.289 -3280.654 -3280.654 -3280.654 -3280.654
2 -3173.360 -3158.895 -3178.155 -3162.973 -3182.757 -3165.015 -3145.365 -3151.408 -3082.133 -3036.738
3 -3112.911 -3121.744 -3094.451 -3105.347 -3093.578 -3103.623 -3085.357 -2981.287 -2961.844 -2961.248
4 -3088.796 -3070.012 -3066.859 -3055.443 -3075.221 -3060.863 -3073.106 -2968.546 -2982.859 -2994.388
5 -3022.854 -3041.013 -3027.418 -3044.589 -3048.509 -3062.117 -3008.185 -2993.164 -3004.847 -3018.715
6 -3016.160 -3001.849 -3021.221 -3006.035 -3050.636 -3027.093 -3015.753 -3017.388 -3018.828 -3043.862
7 -3022.920 -3021.108 -3024.845 -3023.416 -3053.099 -3050.122 -3026.500 -3021.109 -3039.867 -3063.791
8 -3029.988 -3029.294 -3035.582 -3037.397 -3067.618 -3078.757 -3033.420 -3045.572 -3063.640 -3095.861
9 -3046.615 -3051.502 -3052.596 -3055.257 -3088.796 -3104.180 -3047.826 -3069.998 -3094.454 -3130.590
```






```
Summary(m3)
Mclust VVV (ellipsoidal, varying volume,
    shape, and orientation) model with 2
    components:

 log.likelihood   n df       BIC
      -1400.247 450 11 -2867.695

Clustering table:
  1   2
201 249
```

```
m4=Mclust(x)
Summary(m4)
Mclust VVV (ellipsoidal, varying volume, shape,
    and orientation) model with 3 components:

 log.likelihood   n df       BIC
      -1427.818 450 17 -2959.493

Clustering table:
  1   2   3
 28 221 201
```

Model based hierarchical clustering

- Starting from treating each object as a singleton clusters
- Merge pairs of clusters corresponding to the greatest increase in *classification likelihood* among all possible pairs

$$\mathcal{L}_{CL}(\theta_1, \ldots, \theta_G; \ell_1, \ldots, \ell_n \mid \mathbf{y}) = \prod_{i=1}^{n} f_{\ell_i}(\mathbf{y}_i \mid \theta_{\ell_i}).$$

- Note here each object *i* is classified to a class l_i

Example: recursive-partitioning

- Houseman, E. Andres, et al. "Model-based clustering of DNA methylation array data: a recursive-partitioning algorithm for high-dimensional data arising as a mixture of beta distributions." *BMC bioinformatics* 9.1 (2008): 365.
- Data: n subjects described by J features

$$\mathbf{Y}_i = (Y_{i1}, \ldots Y_{iJ}) \qquad C_i \in \{1, \ldots, K\}$$

- Y_{ij} follows beta distribution with parameters

α_{kj} β_{kj}

- Consider I as subject and j for locus, Y_{ij} as methylation proportion

$$f(Y_{ij} = \gamma \mid C_i = k) = \mathrm{B}(\alpha_{kj}, \beta_{kj})^{-1} \gamma^{\alpha_{kj}-1} (1-\gamma)^{\beta_{kj}-1}$$

- Consider the likelihood

$$f(\mathrm{Y}_i = \mathrm{y}_i) = \sum_{k=1}^{K} \eta_k \prod_{j=1}^{J} \mathrm{B}(\alpha_{kj}, \beta_{kj})^{-1} \gamma_{ij}^{\alpha_{kj}-1} (1-\gamma_{ij})^{\beta_{kj}-1}$$

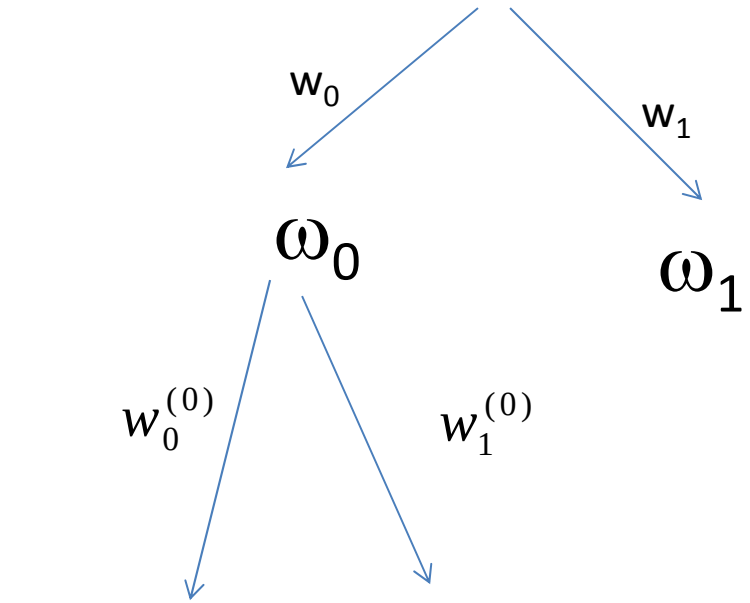
- An EM algorithm can be used as in the mixture normal example

$$\ell(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}) = \sum_{i=1}^{n} \log\{f(\mathrm{Y}_i = \mathrm{y}_i)\}.$$

- In the EM algorithm an expectation of the class probability w_{ik} given the current parameter is computed
- For easier computation, consider a weighted version

$$\ell^{(\omega)}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\eta}; \boldsymbol{\omega}) = \sum_{i=1}^{n} \omega_i \log\{f(\mathrm{Y}_i = \mathrm{y}_i)\}$$

Partitioning weight



At each node, compare the current model and the next split:

$$\mathrm{wtdBIC}_2(r) = (4J+1)\log\left(\sum_{i=1}^{n}\omega_i^r\right) - 2\ell^{(\omega)}(\boldsymbol{\alpha}^{(r)}, \boldsymbol{\beta}^{(r)}, \boldsymbol{\eta}^{(r)}; \boldsymbol{\omega}^{(r)}),$$

$$\mathrm{wtdBIC}_1(r) = 2J\log\left(\sum_{i=1}^{n}\omega_i^r\right) - 2\ell^{(\omega)}(\boldsymbol{\alpha}^{*(r)}, \boldsymbol{\beta}^{*(r)}, \boldsymbol{\eta}^{*(r)}; \boldsymbol{\omega}^{*(r)}),$$

If wtdBIC_2 is greater than wtdBIC_1 (note here the definition is on -2logLikelihood, so smaller is better), it is not worth splitting any more: Terminate the recursion at node r.

- Model based clustering can be powerful
  - "The power of model based methods is incredible"
- Even if there is a true model, it may not be well identified
- The certainty of the model is hard to evaluate, though models can be compared
- The certainty of cluster membership of each subject is different
- If there is truly a hierarchical structure, then many levels of clustering can be "correct"