JOHANNES KEPLER
UNIVERSITY LINZ

Tools and methods for model-based clustering in R

Bettina Grün

Rennes 2018

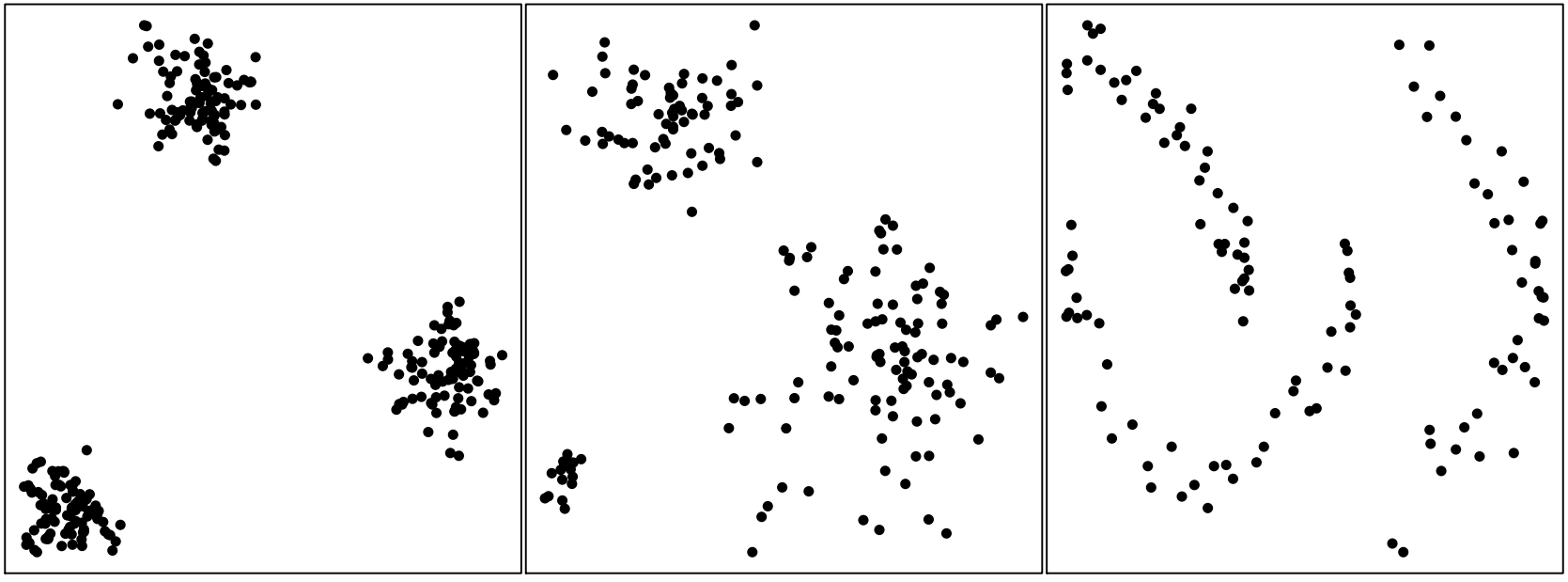
Cluster analysis

- The task of grouping a set of objects such that
  - Objects in the same group are as similar as possible and
  - Objects in different groups are as dissimilar as possible.
- The aim is to determine a partition of the given set of objects, e.g., to determine which objects belong to the same group and which to different groups.
- Statistical methods:
  - Heuristic methods: hierarchical clustering, partitioning methods (e.g., k-means).
  - Model-based methods: finite mixture models.

Specifying the cluster problem

- The cluster problem is in general perceived as ill defined.
- Different notions of what defines a cluster exist:
  - Compactness.
  - Density-based levels.
  - Connectedness.
  - Functional similarity.
- Several cluster solutions might exist for a given data set depending on which notion is used.
- The application context is important to define which clusters should be targeted and to assess the usefulness of a clustering solution.

Specifying the cluster problem / 2



Model-based clustering methods

- Model-based clustering embeds the clustering problem in a probabilistic framework.
- This implies:
  - Statistical inference tools can be used.
  - Different cluster distributions can be used depending on the cluster notion.
  - More explicit specification of what defines a cluster required than for heuristic methods.

Finite mixture models

- Generative model for observations $(\boldsymbol{y}_i, \boldsymbol{x}_i)$, $i = 1, \ldots, n$:
  1. Draw a cluster membership indicator S_i from a multinomial distribution with parameters $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_K)$.
  2. Draw $\boldsymbol{y}_i$ given $\boldsymbol{x}_i$ and S_i from the cluster distribution:

$$\boldsymbol{y}_i | \boldsymbol{x}_i \sim f_{S_i}(\boldsymbol{y}_i | \boldsymbol{x}_i).$$

- The distribution of $(\boldsymbol{y}_i, \boldsymbol{x}_i)$ is then given by

$$\boldsymbol{y}_i | \boldsymbol{x}_i \sim \sum_{k=1}^{K} \eta_k f_k(\boldsymbol{y}_i | \boldsymbol{x}_i),$$

  where
  - $\eta_k \geq 0$ for all k and $\sum_{k=1}^{K} \eta_k = 1$.
  - $f_k()$ represents the cluster distribution.

Finite mixture models / 2

Methods differ with respect to:

- Clustering kernel:
  - Specification of cluster distributions.
  - Use of additional variables $\boldsymbol{x}_i$, e.g., for regression.
- Estimation framework:
  - Maximum likelihood estimation.
  - Bayesian estimation.

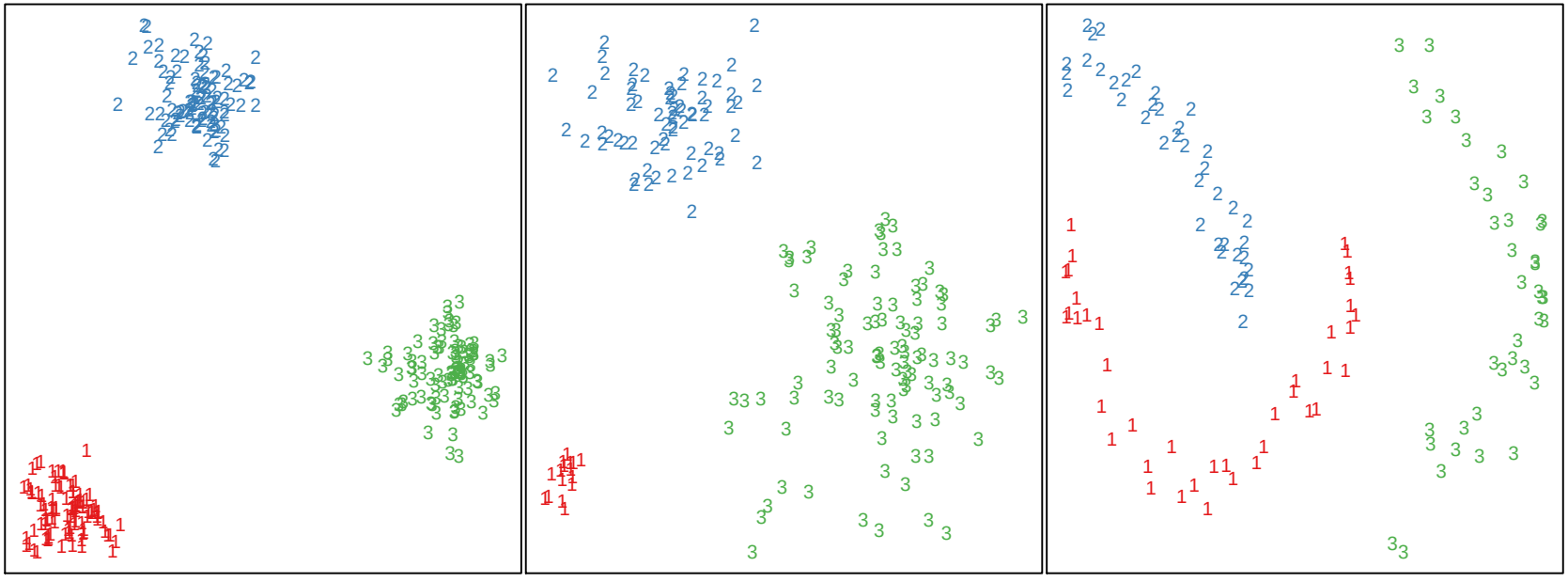
Finite mixture models / 3

- Cluster membership indicators can be inferred using the a-posteriori probabilities:

$$\mathbb{P}(S_i = k|\boldsymbol{y}_i, \boldsymbol{x}_i) \propto \eta_k f_k(\boldsymbol{y}_i|\boldsymbol{x}_i).$$

- A hard assignment can be obtained by
  - Assigning to the cluster where this probability is maximum.
  - Drawing from this probability distribution.

Finite mixture models / 4



Estimation of finite mixtures with fixed K

- Maximum likelihood estimation:
  - EM algorithm.
  - General purpose optimizers.
  - Hybrid approaches.
- Bayesian estimation:
  - MCMC sampling with data augmentation by adding S_i, $i = 1, \ldots, n$.
  - General purpose Gibbs samplers can be used, e.g., **JAGS** available in R through package **rjags** (Plummer, 2016).

EM algorithm

- Standard maximum likelihood estimation method in a missing data context.
- Guaranteed to converge for bounded likelihoods.
- Only convergence to a local optimum.
- In general slow convergence behavior.
- Consists of E- and M-step:
  - E-step requires determining the a-posteriori probabilities.
  - M-step requires weighted maximum likelihood estimation of the cluster distributions.

MCMC sampling

- Determination of the a-posteriori probabilities required to draw S_i, $i = 1, \ldots, n$ from a multinomial distribution.
- Conditional on S_i, $i = 1, \ldots, n$ drawing from the posterior of the cluster-specific parameters is the same as if the cluster-specific distribution is used for the complete data set.
- Often poor mixing observed.
- For symmetric priors the posterior is also symmetric and thus multimodal.

Determining the number of clusters

- No generally accepted solution available.
- Suggested methods include:
  - Information criteria: AIC, BIC, ICL.
  - Likelihood ratio test with distribution under the null determined using sampling methods.
  - Marginal likelihoods in Bayesian estimation.

Clustering kernel

- **Components corresponding to clusters:**
  In general using parametric distributions for the components and thus also for the clusters.
  - Multivariate continuous data.
  - Multivariate categorical data.
  - Multivariate mixed data.
  - Multivariate data with regression structure.
- **Combining components to clusters:**
  I.e., the cluster distributions are mixture distributions.
  - Two-step procedures.
  - Simultaneous estimation using constraints or informative priors.

In the following these variants are investigated for maximum likelihood estimation.

Multivariate continuous data

- The standard model is a mixture of multivariate Gaussians.
- The model-based clustering model is given by

$$\boldsymbol{y}_i \sim \sum_{k=1}^{K} \eta_k \phi(\boldsymbol{y}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

- For K clusters and d-dimensional observations $\boldsymbol{y}_i$ the number of estimated parameters corresponds to

$$K \cdot (d + d(d+1)/2) + K - 1.$$

Multivariate continuous data / 2

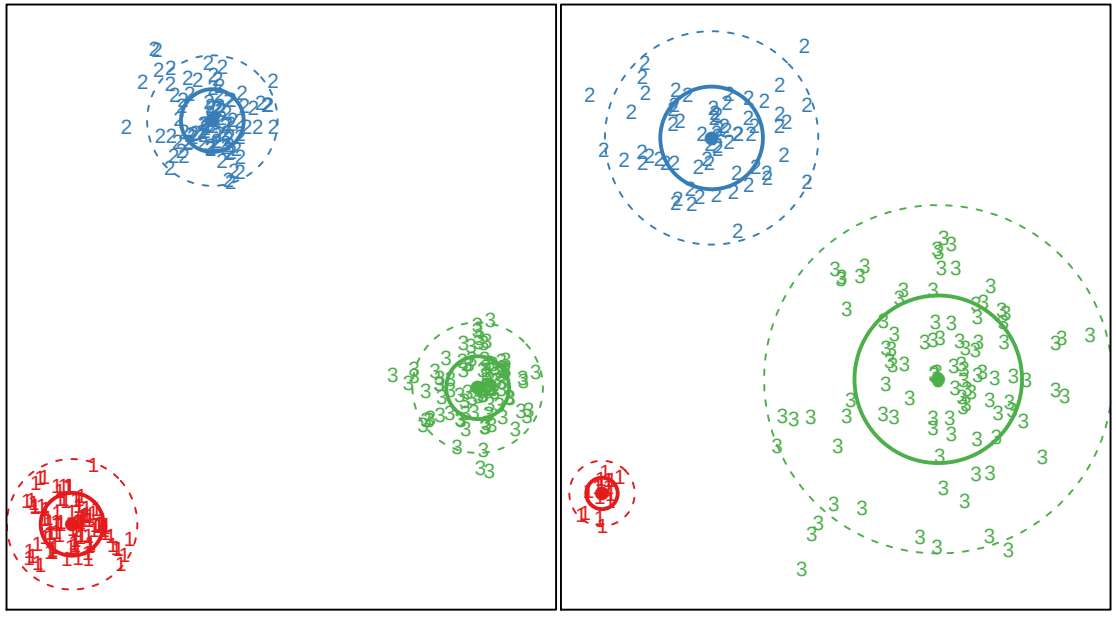
- Parsimonity is achieved based on the decomposition of the variance-covariance matrix into
  - Volume λ.
  - Orientation D.
  - Shape A.

  given by

$$\Sigma_k = \lambda_k D_k A_k D_k^\top .$$

- 14 different models emerge by imposing different constraints on the variance-covariance matrices within or across clusters.
- Available packages in R, e.g.:
  - **mclust** (Scrucca et al., 2016),
  - **mixture** (Browne et al., 2018),
  - **Rmixmod** (Lebret et al., 2015).

Multivariate continuous data / 3



Multivariate continuous data / 4

- Alternative approaches to achieve parsimonity are mixtures of factor analyzers.
  - E.g., package **pgmm** (McNicholas et al., 2018) in R.
- If the cluster shapes are not symmetric and light tailed, alternative cluster kernels are:
  - *t*-distributions (e.g., package **teigen**; Andrews et al. 2018).
  - Skewed and / or heavy tailed distributions: e.g.,
    - **EMMIXcskew** (Lee and McLachlan, 2018),
    - **MixSAL** (Franczak et al., 2018),
    - **mixsmsn** (Prates et al., 2013).

Multivariate categorical data

- Often also referred to as latent class analysis.
- Clusters induce a dependency between variables, while variables are independent within clusters.
  $\Rightarrow$ Local independency assumption.
- The model-based clustering model is given by

$$\boldsymbol{y}_i \sim \sum_{k=1}^{K} \eta_k \left[\prod_{j=1}^{d} \text{Multinomial}(y_{ij} | \pi_k^j) \right]$$

  for d-dimensional observations.
- Available packages in R: e.g.,
  - **poLCA** (Linzer and Lewis, 2011)
  - **Rmixmod** (Lebret et al., 2015)

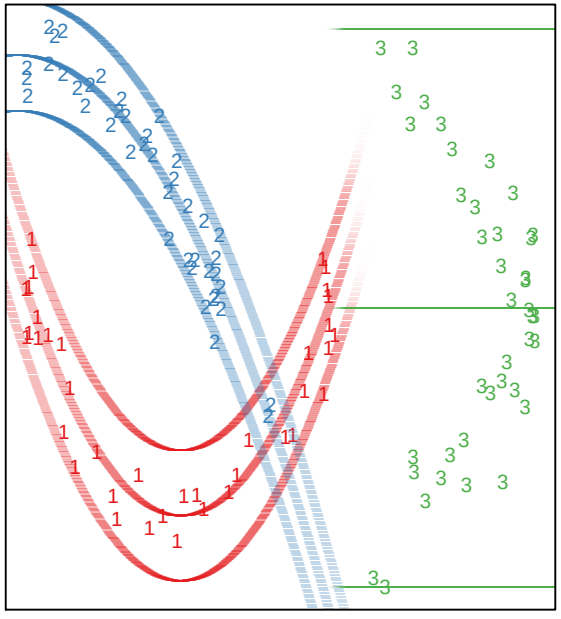
Multivariate data with regression structure

- Often also referred to as clusterwise regression.
- The model-based clustering model is given by

$$\boldsymbol{y}_i|\boldsymbol{x}_i \sim \sum_{k=1}^{K} \eta_k f(\boldsymbol{y}_i|\boldsymbol{\mu}_k(\boldsymbol{x}_i), \phi_k).$$

- Different regression models possible:
  - Generalized linear models.
  - Generalized linear mixed-effects models.
- Available packages in R: e.g.,
  - **flexmix** (Leisch, 2004; Grün and Leisch, 2008)
  - **mixtools** (Benaglia et al., 2009)

Multivariate data with regression structure / 2



Combining components to clusters

- Two-step procedures:
  1. Fit a mixture model as semi-parametric tool for density estimation.
  2. Combine components of the mixture model to form clusters based on some criterion.

  Available packages in R, e.g.:
  - **mclust** uses entropy as criterion (Baudry et al., 2010).
  - **fpc** (Hennig, 2018) provides several variants as proposed in Hennig (2010).
- Simultaneous estimation using informative priors in Bayesian estimation can be used in combination with standard estimation methods.

Post-processing tools

- Inference on partitions.
- Inference on cluster-specific parameters:
  In particular for Bayesian estimation the label switching problem needs to be resolved.
- Assigning new observations to clusters:
  Cluster predictions possible.
- Assessing cluster quality.

Assessing cluster quality

- Agreement measures between cluster assignments and true classes available as in a supervised setting:
  - Label-invariant measures:
    - Rand index (corrected for agreement by chance).
    - Jaccard index.
    - Purity.
  - Label-specific measures:
    - Misclassification rate.
- Available packages in R:
  - Package **clue** (Hornik, 2005) provides general infrastructure to assess cluster solutions:
    Function `cl_agreement` provides several methods to assess cluster agreement.

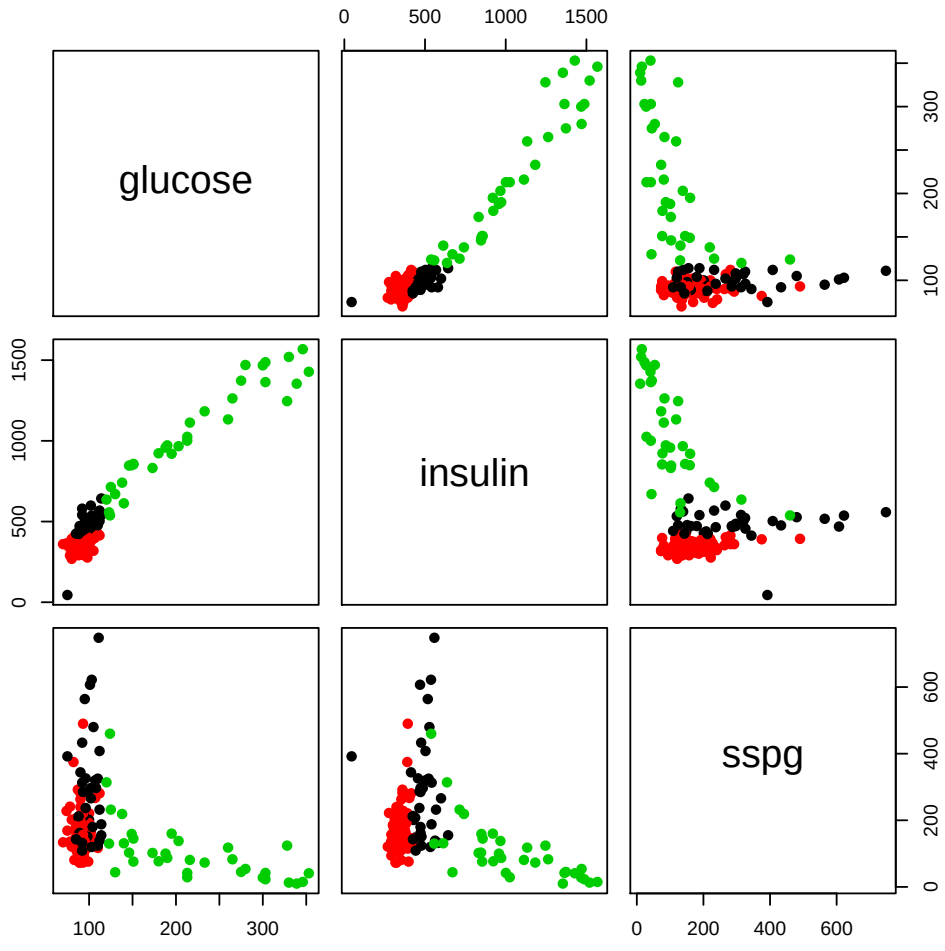
Comparing packages

- Use a classification data set with 3 known classes and three continuous predictors.
- Compare the three packages for fitting mixtures of Gaussians with variance-covariance matrix decomposition.
- Fit all models for $K = 1, \ldots, 9$ and select the best according to the BIC.
- Extract the cluster assignments and assess cluster quality in comparison to the true classification.

Comparing packages: data

```
> data("diabetes", package = "mclust")
> library("clue")
> class <- as.cl_partition(diabetes$class)
> X <- diabetes[, -1]
> pairs(X, col = diabetes$class, pch = 19)
```

Comparing packages: data / 2



Comparing packages: mclust

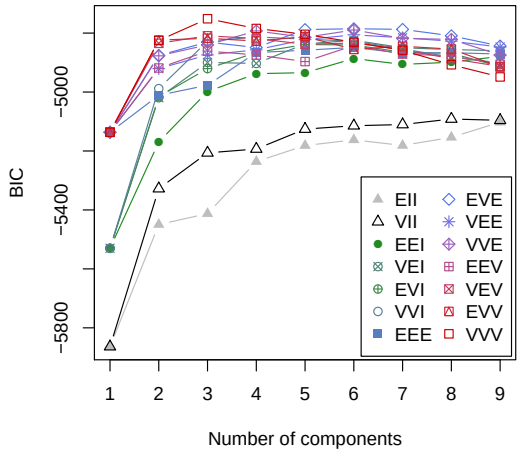
```
> library("mclust")
> mclust.sol <- Mclust(X, G = 1:9)
> class(mclust.sol)

[1] "Mclust"

> mclust.sol[c("G", "modelName")]

$G
[1] 3

$modelName
[1] "VVV"

> c(logLik = logLik(mclust.sol), BIC = BIC(mclust.sol))

   logLik       BIC
-2303.493  4751.311
```

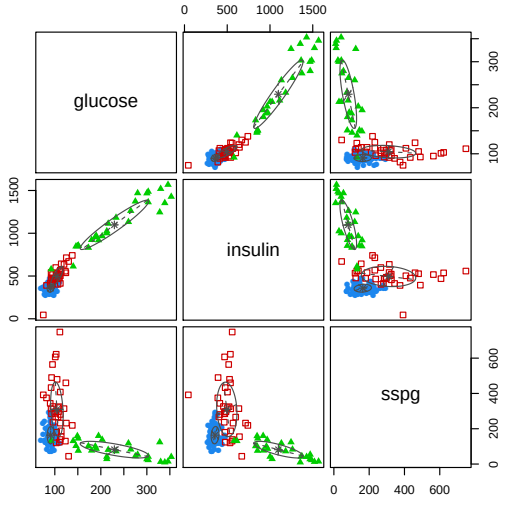
Comparing packages: mclust / 2

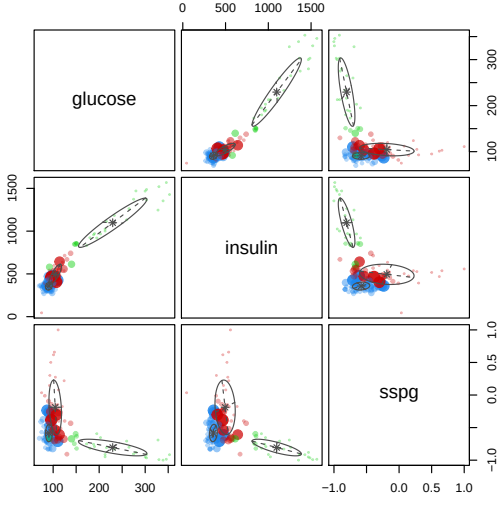
Standard visualization:

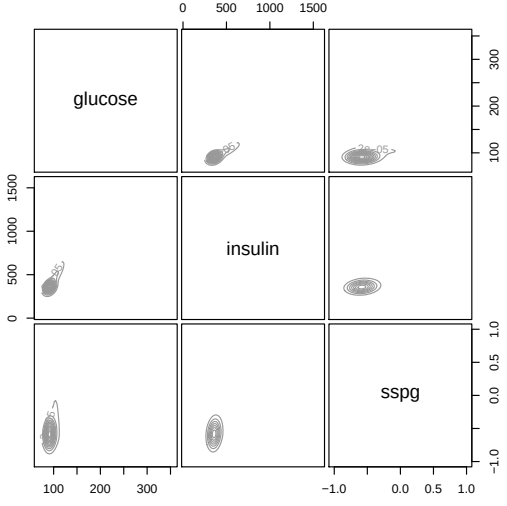
```
> plot(mclust.sol, "BIC")
> plot(mclust.sol, "classification")
> plot(mclust.sol, "uncertainty")
> plot(mclust.sol, "density")
```

Comparing packages: mclust / 3









Comparing packages: mclust / 4

Class agreement:

```
> cl_agreement(as.cl_hard_partition(mclust.sol), class,
+   method = "crand")

Cross-agreements using corrected Rand index:
          [,1]
[1,] 0.6640181
```

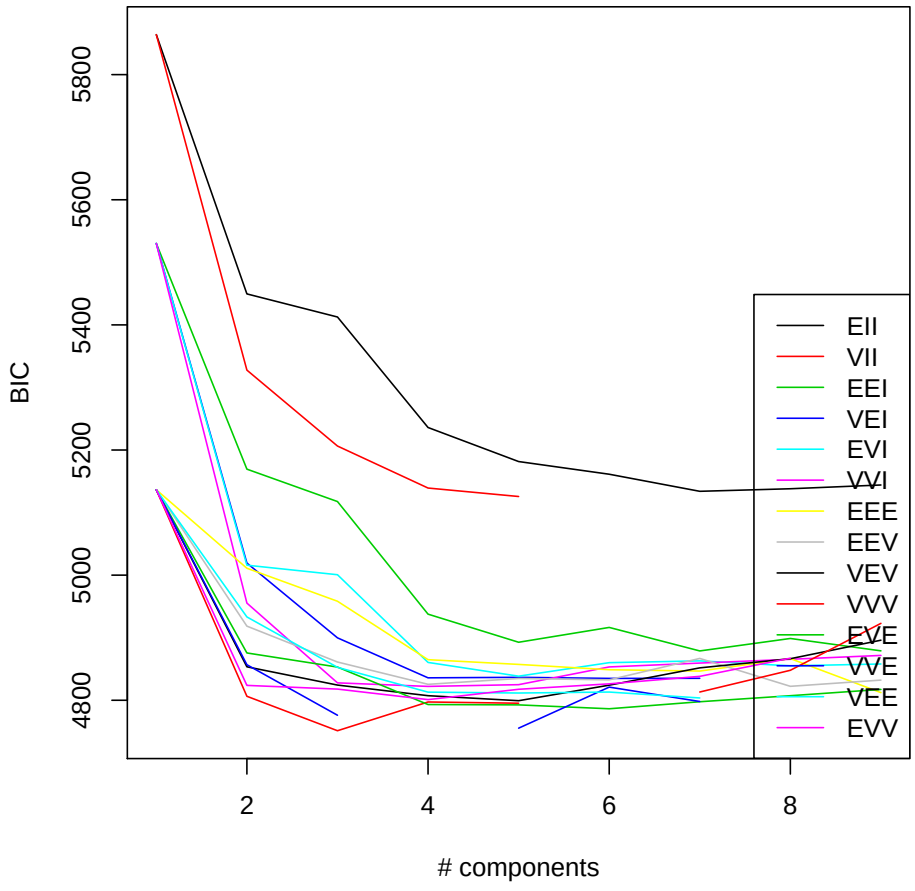
Comparing packages: mixture

```
> set.seed(1234)
> library("mixture")
> mixture.sol <- gpcm(as.matrix(X), G = 1:9)
> class(mixture.sol)

[1] "gpcm"

> (best <- mixture.sol$bicModel[c("G", "covtype")])

$G
[1] 3

$covtype
[1] "VVV"

> mixture.sol$BIC[best$G, best$covtype, c("loglik", "BIC")]

   loglik       BIC
-2303.492  4751.309
```

Comparing packages: mixture / 2

Standard visualization:

```
> plot(mixture.sol)
```



Comparing packages: mixture / 3

Class agreement:

```
> mixture.preds <- mixture.sol$map
> cl_agreement(as.cl_partition(mixture.preds), class,
+   method = "crand")

Cross-agreements using corrected Rand index:
           [,1]
[1,] 0.6640181
```

Comparing packages: Rmixmod

```
> library("Rmixmod")
> mixmod.sol.1 <- mixmodCluster(
+   data = X, nbCluster = 1,
+   models = mixmodGaussianModel(listModels =
+     c("Gaussian_pk_Lk_Ck", "Gaussian_pk_Lk_Bk",
+       "Gaussian_pk_Lk_I")))
> mixmod.sol.1["bestResult"]@criterionValue

[1] 5136.446

> mixmod.sol <- mixmodCluster(
+   data = X, nbCluster = 2:9,
+   models = mixmodGaussianModel(equal.proportions = FALSE),
+   strategy = mixmodStrategy(nbTryInInit = 20), seed = 10)
> class(mixmod.sol)

[1] "MixmodCluster"
attr(,"package")
[1] "Rmixmod"
```

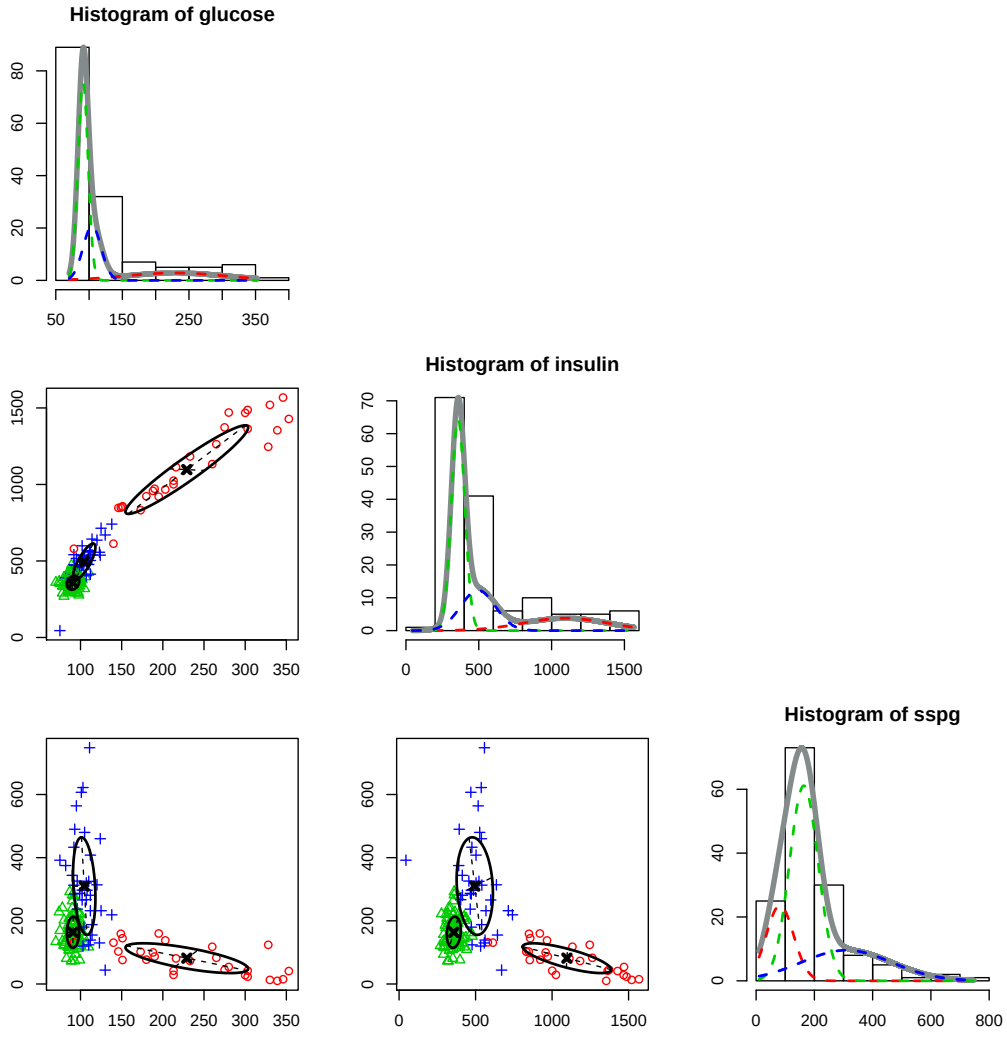
Comparing packages: Rmixmod / 2

```
> mixmod.sol["bestResult"]@nbCluster

[1] 3

> mixmod.sol["bestResult"]@model

[1] "Gaussian_pk_Lk_Ck"

> c(logLik = mixmod.sol["bestResult"]@likelihood,
+   BIC = mixmod.sol["bestResult"]@criterionValue)

   logLik       BIC
-2303.493  4751.311
```

Standard visualization:

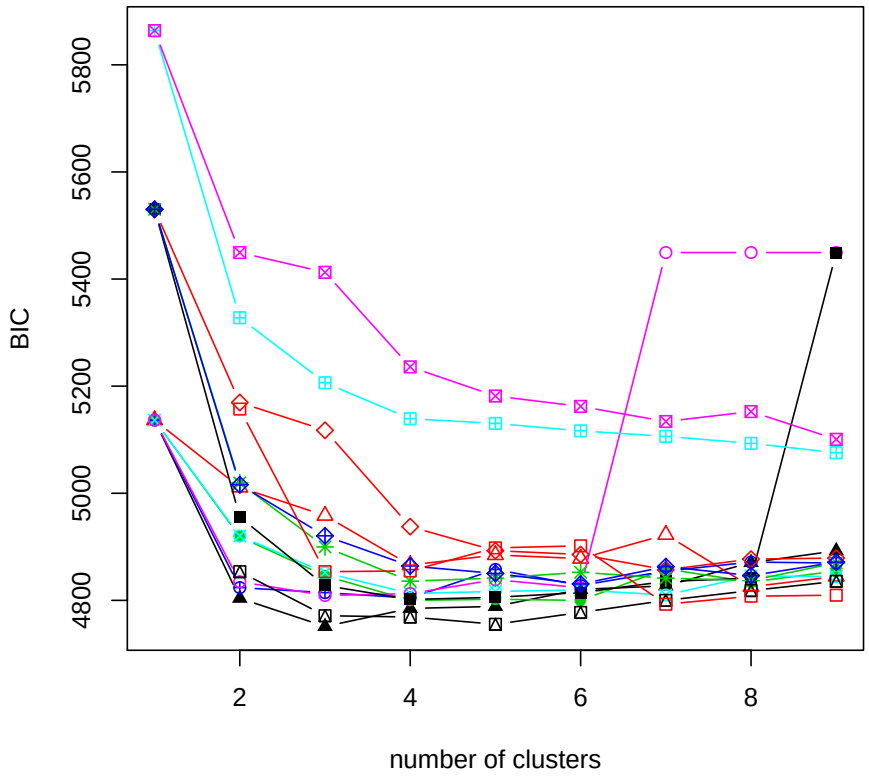
```
> plot(mixmod.sol)
```

Comparing packages: Rmixmod / 3



Comparing packages: Rmixmod / 4

Some coding also allows to obtain the BIC values for all fitted models ...



Comparing packages: Rmixmod / 5

Class agreement:

```
> mixmod.preds <- mixmod.sol["bestResult"]@partition
> cl_agreement(as.cl_partition(mixmod.preds), class,
+   method = "crand")

Cross-agreements using corrected Rand index:
          [,1]
[1,] 0.6640181
```

Comparing packages

- All packages implement the EM algorithm and allow to specify the number of clusters and which variance-covariance structures should be fitted.
- By default all packages use the BIC to select the best fitting model.
- For this simple example all packages arrive at the same solution.
- However, the packages use different initialization strategies.
- The return objects differ as well as the associated methods available to inspect the obtained solutions.
- Class assignments can be extracted and used in combination with package **clue** to assess cluster quality.

Summary

- Model-based clustering is a versatile method for clustering.
- Different variants exist depending on
  - Clustering kernel.
  - Estimation methods.
- A large number of R packages are available where each covers a certain set of mixture models.
- The same functionality is often covered by several packages.
- Packages vary in the range of models covered and what additional functionality is provided for fitted models.
- Common standards and infrastructure are mostly lacking.
- For more information see the CRAN Task View: Cluster Analysis & Finite Mixture Models:
  `https://CRAN.R-project.org/view=Cluster`

References

J. L. Andrews, J. R. Wickins, N. M. Boers, and P. D. McNicholas. teigen: An R package for model-based clustering and classification via the multivariate *t* distribution. **Journal of Statistical Software**, 83(7):1–32, 2018. doi: 10.18637/jss.v083.i07.

J.-P. Baudry, A. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. **Journal of Computational and Graphical Statistics**, 2(19):332–353, 2010. doi: 10.1198/jcgs.2010.08111.

T. Benaglia, D. Chauveau, D. R. Hunter, and D. Young. mixtools: An R package for analyzing finite mixture models. **Journal of Statistical Software**, 32(6):1–29, 2009. doi: 10.18637/jss.v032.i06.

R. P. Browne, A. ElSherbiny, and P. D. McNicholas. **Mixture: Mixture Models for Clustering and Classification**, 2018. URL `https://CRAN.R-project.org/package=mixture`. R package version 1.5.

References / 2

B. C. Franczak, R. P. Browne, P. D. McNicholas, and K. L. Burak. **MixSAL: Mixtures of Multivariate Shifted Asymmetric Laplace (SAL) Distributions**, 2018. URL `https://CRAN.R-project.org/package=MixSAL`. R package version 1.0.

B. Grün and F. Leisch. FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters. **Journal of Statistical Software**, 28(4):1–35, 2008. doi: 10.18637/jss.v028.i04.

C. Hennig. Methods for merging Gaussian mixture components. **Advances in Data Analysis and Classification**, 4(1):3–34, 2010. doi: 10.1007/s11634-010-0058-3.

C. Hennig. **fpc: Flexible Procedures for Clustering**, 2018. URL `https://CRAN.R-project.org/package=fpc`. R package version 2.1-11.

K. Hornik. A CLUE for CLUster Ensembles. **Journal of Statistical Software**, 14(12):1–25, 2005. doi: 10.18637/jss.v014.i12.

References / 3

R. Lebret, S. Iovleff, F. Langrognet, C. Biernacki, G. Celeux, and G. Govaert. Rmixmod: The R package of the model-based unsupervised, supervised, and semi-supervised classification Mixmod library. **Journal of Statistical Software**, 67(6):1–29, 2015. doi: 10.18637/jss.v067.i06.

S. Lee and G. McLachlan. EMMIXcskew: An R package for the fitting of a mixture of canonical fundamental skew *t*-distributions. **Journal of Statistical Software**, 83(3):1–32, 2018. doi: 10.18637/jss.v083.i03.

F. Leisch. FlexMix: A general framework for finite mixture models and latent class regression in R. **Journal of Statistical Software**, 11(8):1–18, 2004. doi: 10.18637/jss.v011.i08.

D. A. Linzer and J. B. Lewis. poLCA: An R package for polytomous variable latent class analysis. **Journal of Statistical Software**, 42(10):1–29, 2011. doi: 10.18637/jss.v042.i10.

P. D. McNicholas, A. ElSherbiny, A. F. McDaid, and T. B. Murphy. **pgmm: Parsimonious Gaussian Mixture Models**, 2018. URL `https://CRAN.R-project.org/package=pgmm`. R package version 1.2.2.

References / 4

M. Plummer. **rjags: Bayesian Graphical Models Using MCMC**, 2016. URL `https://CRAN.R-project.org/package=rjags`. R package version 4-6.

M. O. Prates, C. R. B. Cabral, and V. H. Lachos. mixsmsn: Fitting finite mixture of scale mixture of skew-normal distributions. **Journal of Statistical Software**, 54(12):1–20, 2013. doi: 10.18637/jss.v054.i12.

L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery. mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. **The R Journal**, 8(1):205–233, 2016. doi: 10.21236/ada459792.