WIRTSCHAFTS
UNIVERSITÄT
WIEN

Finite Mixture Modelling
Model Specification, Estimation & Application

Bettina Grün

Department of Statistics and Mathematics

Research Seminar, November 23 2007

Finite mixture models

The finite mixture distribution is given by

$$H(\boldsymbol{y}|\boldsymbol{x},\Theta) = \sum_{k=1}^{K} \pi_k F_k(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\vartheta}_k)$$

with

$$\sum_{k=1}^{K} \pi_k = 1 \quad \wedge \quad \pi_k > 0 \,\forall k.$$

In the following it is assumed that the component specific density functions f_k exist and determine the mixture density h.

Finite mixture models

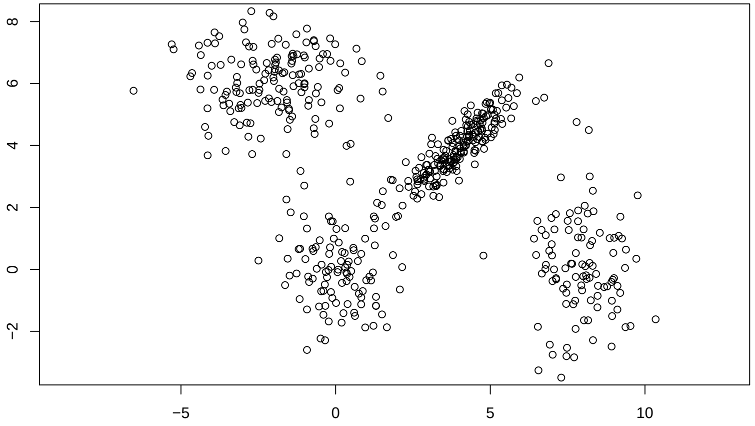
Types of applications:

- semi-parametric tool to estimate general distribution functions
- modeling unobserved heterogeneity

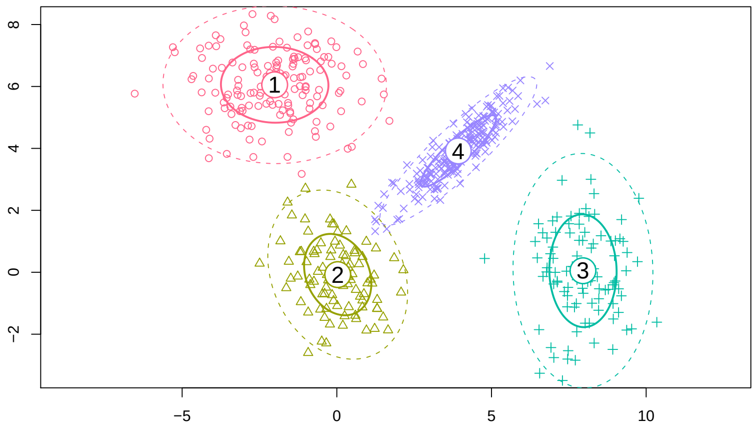
Special cases:

- model-based clustering
- mixtures of regression models

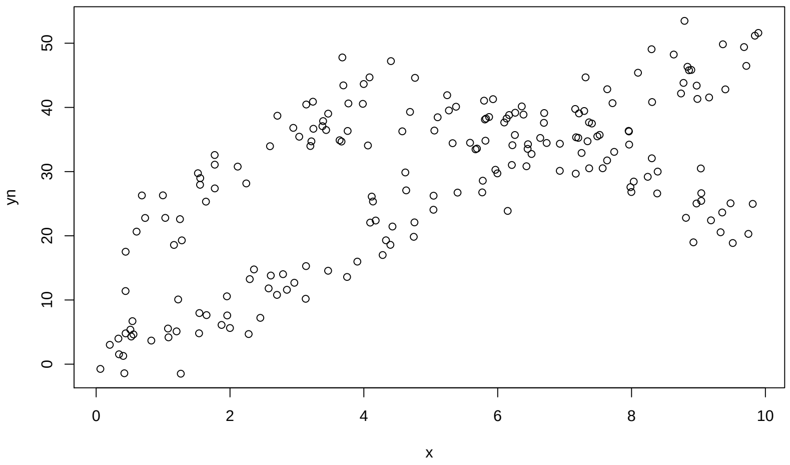
Finite mixture models



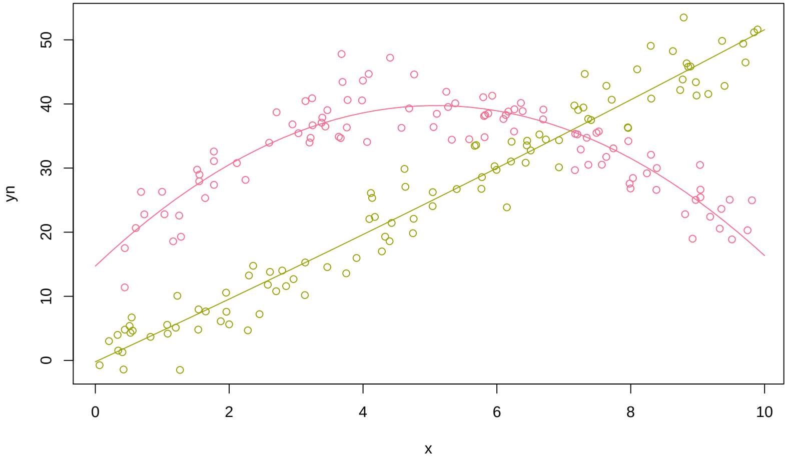
Finite mixture models



Finite mixture models



Finite mixture models



Estimation

Maximum-Likelihood: Expectation-Maximization (EM) Algorithm (Dempster, Laird and Rubin, 1977)

- General method for ML estimation in models with unobserved latent variables: The complete likelihood containing the observed and unobserved data is easier to estimate.
- Iterates between
  - E-step, which computes the expectation of the complete likelihood, and
  - M-step, where the expected complete likelihood is maximized.

Bayesian: Gibbs sampling (Diebolt and Robert, 1994)

- Markov Chain Monte Carlo algorithm
- Applicable when the joint posterior distribution is not known explicitly, but the conditional posterior distributions of each variable/subsets of variables are known.

Missing data

The component-label vectors $\boldsymbol{z}_n = (z_{nk})_{k=1,\dots,K}$ are treated as missing data. It holds that

- $z_{nk} \in \{0,1\}$ and
- $\sum_{k=1}^{K} z_{nk} = 1$ for all $k = 1,\dots,K$.

The complete log-likelihood is given by

$$\log L_c(\Theta) = \sum_{k=1}^{K} \sum_{n=1}^{N} z_{nk} \left[\log \pi_k + \log f_k(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\vartheta}_k)\right].$$

EM algorithm: E-step

Given the current parameter estimates $\boldsymbol{\Theta}^{(i)}$ replace the missing data z_{nk} by the estimated a-posteriori probabilities

$$\hat{z}_{nk}^{(i)} = \mathbb{P}(k|\boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{\Theta}^{(i)}) = \frac{\pi_k^{(i)} f_k(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\vartheta}_k^{(i)})}{\sum\limits_{u=1}^{K} \pi_u^{(i)} f_k(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\vartheta}_u^{(i)})}.$$

The conditional expectation of $\log L_c(\Theta)$ at the i^{th} step is given by

$$\begin{aligned} Q(\Theta; \Theta^{(i)}) &= \mathbb{E}_{\Theta^{(i)}}\left[\log L_c(\Theta)|\boldsymbol{y}, \boldsymbol{x}\right] \\ &= \sum_{k=1}^{K} \sum_{n=1}^{N} \hat{z}_{nk}^{(i)} \left[\log \pi_k + \log f_k(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\vartheta}_k)\right]. \end{aligned}$$

EM algorithm: M-step

The next parameter estimate is given by:

$$\Theta^{(i+1)} = \arg\max_{\Theta} Q(\Theta; \Theta^{(i)}).$$

The estimates for the prior class probabilities are given by:

$$\pi_k^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} \hat{z}_{nk}^{(i)}.$$

The component specific parameter estimates are determined by:

$$\boldsymbol{\vartheta}_k^{(i+1)} = \arg\max_{\boldsymbol{\vartheta}_k} \sum_{n=1}^{N} \hat{z}_{nk}^{(i)} \log(f_k(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{\vartheta}_k)).$$

⇒ weighted ML estimation of the component specific model.

M-step: Mixtures of Gaussian distributions

The solutions for the M-step are given in closed form:

$$\boldsymbol{\mu}_k^{(i+1)} = \frac{\sum_{n=1}^{N} \hat{z}_{nk}^{(i)} \boldsymbol{y}_n}{\sum_{n=1}^{N} \hat{z}_{nk}^{(i)}}$$

$$\Sigma_k^{(i+1)} = \frac{\sum_{n=1}^{N} \hat{z}_{nk}^{(i)} (\boldsymbol{y}_n - \boldsymbol{\mu}_k^{(i+1)})(\boldsymbol{y}_n - \boldsymbol{\mu}_k^{(i+1)})'}{\sum_{n=1}^{N} \hat{z}_{nk}^{(i)}}$$

Estimation: EM algorithm

Advantages:

- The likelihood is increased in each step → EM algorithm converges for bounded likelihoods.
- Relatively easy to implement:
  - Different mixture models require only different M-steps.
  - Weighted ML estimation of the component specific model is sometimes already available.

Disadvantages:

- Standard errors have to be determined separately as the information matrix is not required during the algorithm.
- Convergence only to a local optimum
- Slow convergence

$\Rightarrow$ variants such as Stochastic EM (SEM) or Classification EM (CEM)

EM algorithm: Number of components

Information criteria: e.g. AIC, BIC, ICL

Likelihood ratio test statistic: Comparison of nested models where the smaller model is derived by fixing one parameter at the border of the parameter space.

$\Rightarrow$ Regularity conditions are not fulfilled.

The asymptotic null distribution is not the usual χ^2-distribution with degrees of freedom equal to the difference beween the number of parameters under the null and alternative hypotheses.

- distributional results for special cases
- bootstrapping

Bayesian estimation

Determine the posterior density using Bayes' theorem

$$p(\Theta|\boldsymbol{Y}, \boldsymbol{X}) \propto h(\boldsymbol{Y}|\boldsymbol{X}, \Theta)p(\Theta),$$

where $p(\Theta)$ is the prior and $\boldsymbol{Y} = (\boldsymbol{y}_n)_n$ and $\boldsymbol{X} = (\boldsymbol{x}_n)_n$.

Standard prior distributions:

- Proper priors: Improper priors give improper posteriors.
- Independent priors for the component weights and the component specific parameters.
- Conjugate priors for the complete likelihood
  - Dirichlet distribution $\mathcal{D}(e_{0,1}, \dots, e_{0,K})$ for the component weights which is the conjugate prior for the multinomial distribution.
  - Priors on the component specific parameters depend on the underlying distribution family.
- Invariant priors, e.g. the parameter for the Dirchlet prior is constant over all components: $e_{0,k} \equiv e_0$.

Estimation: Gibbs sampling

Starting with $\boldsymbol{Z}^0 = (\boldsymbol{z}_n^0)_{n=1,\dots,N}$ repeat the following steps for $i = 1, \dots, I_0, \dots, I + I_0$.

1. Parameter simulation conditional on the classification $\boldsymbol{Z}^{(i-1)}$:
   (a) Sample $\pi_1, \dots, \pi_K$ from $\mathcal{D}((\sum_{n=1}^N z_{nk}^{(i-1)} + e_{0,k})_{k=1,\dots,K})$.
   (b) Sample component specific parameters from the complete-data posterior $p(\vartheta_1, \dots, \vartheta_K|\boldsymbol{Z}^{(i-1)}, \boldsymbol{Y})$
   Store the actual values of all parameters $\Theta^{(i)} = (\pi_k^{(i)}, \boldsymbol{\vartheta}_k^{(i)})_{k=1,\dots,K}$.
2. Classification of each observation $(\boldsymbol{y}_n, \boldsymbol{x}_n)$ conditional on knowing $\Theta^{(i)}$:
   Sample $\boldsymbol{z}_n^{(i)}$ from the multinomial distribution with parameter equal to the posterior probabilities.

After discarding the burn-in draws the draws $I_0 + 1, \dots, I + I_0$ can be used to approximate all quantities of interest.

Example: Gaussian distribution

Assume an independence prior

$$p(\boldsymbol{\mu}_k, \Sigma_k^{-1}) \sim f_N(\boldsymbol{\mu}_k; \boldsymbol{b}_0, \boldsymbol{B}_0) f_W(\Sigma_k^{-1}; c_0, \boldsymbol{C}_0).$$

1. Parameter simulation conditional on the classification $\boldsymbol{Z}^{(i-1)}$:
   (a) Sample $\pi_1^{(i)}, \dots, \pi_K^{(i)}$ from $\mathcal{D}((\sum_{n=1}^N z_{nk}^{(i-1)} + e_{0,k})_{k=1,\dots,K})$.
   (b) Sample $(\Sigma_k^{-1})^{(i)}$ in each group k from a Wishart $\mathcal{W}(c_k(\boldsymbol{Z}^{(i-1)}), \boldsymbol{C}_k(\boldsymbol{Z}^{(i-1)}))$ distribution.
   (c) Sample $\boldsymbol{\mu}_k^{(i)}$ in each group k from a $\mathcal{N}(\boldsymbol{b}_k(\boldsymbol{Z}^{(i-1)}), \boldsymbol{B}_k(\boldsymbol{Z}^{(i-1)}))$ distribution.

2. Classification of each observation $\boldsymbol{y}_n$ conditional on knowing $\ominus^{(i)}$:

$$\mathbb{P}(z_{nk}^{(i)} = 1 | \boldsymbol{y}_n, \ominus^{(i)}) \propto \pi_k f_N(\boldsymbol{y}_n; \boldsymbol{\mu}_k, \Sigma_k)$$

Estimation: Gibbs sampling

Advantages:

- Relatively easy to implement
  - Different mixture models differ only in the parameter simulation step.
  - Parameter simulation conditional on the classification is sometimes already available.

Disadvantages:

- Might fail to escape the attraction area of one mode → not all posterior modes are visited.

Gibbs sampling: Number of components

- Bayes factors

- Sampling schemes with a varying number of components
  - reversible-jump MCMC
  - inclusion of birth-and-death processes

Label switching

The posterior distribution is invariant under a permutation of the components with the same component-specific model.

⇒ Determine a unique labelling for component-specific inference:

- Impose a suitable ordering constraint, e.g. $\pi_s < \pi_t \; \forall s, t \in \{1, \dots, S\}$ with $s < t$.
- Minimize the distance to the Maximum-A-Posteriori (MAP) estimate.
- Fix the component membership for some observations.
- Relabelling algorithms.

Initialization

- Construct a suitable parameter vector $\Theta^{(0)}$.
  - random
  - other estimation methods: e.g. moment estimators

- Classify observations/assign a-posteriori probabilities to each observation.
  - random
  - cluster analysis results: e.g. hierarchical clustering, k-means

Extensions and special cases

- Model-based clustering:
  - Latent class analysis: multivariate discrete observations where the marginal distributions in the components are independent.
  - mixtures of factor analyzers
  - mixtures of t-distributions

- Mixtures of regressions:
  - mixtures of generalized linear models
  - mixtures of generalized linear mixed models

- Covariates for the component sizes: concomitant variable models

- Impose equality constraints between component-specific parameters

Software in R

- Model-based clustering:
  - **mclust** (Fraley and Raftery, 2002) for Gaussian mixtures:
    - ∗ specify different models depending on the structure of the variance-covariance matrices (volume, shape, orientation)

$$\Sigma_k = \lambda_k D_k \text{diag}(\boldsymbol{a}_k) D_k'$$

    - ∗ initialize EM algorithm with the solution from an agglomerative hierarchical clustering algorithm

- Clusterwise regression:
  - **flexmix** (Leisch, 2004)

See also CRAN Task View “Cluster Analysis & Finite Mixture Models”.

Software: FlexMix

- The function `flexmix()` provides the E-step and all data handling.
- The M-step is supplied by the user similar to `glm()` families.
- Multiple independent responses from different families
- Currently bindings to several GLM families exist (Gaussian, Poisson, Gamma, Binomial)
- Weighted, hard (CEM) and random (SEM) classification
- Components with prior probability below a user-specified threshold are automatically removed during iteration

FlexMix Design

- Primary goal is extensibility: ideal for trying out new mixture models.
- No replacement of specialized mixtures like `mclust()`, but complement.
- Usage of S4 classes and methods
- Formula-based interface
- Multivariate responses:

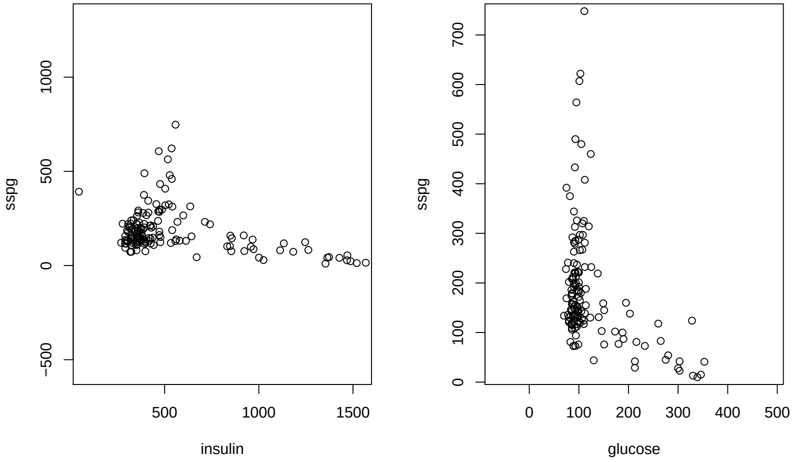
  combination of univariate families: assumption of independence (given x), each response may have its own model formula, i.e., a different set of regressors

  multivariate families: if family handles multivariate response directly, then arbitrary multivariate response distributions are possible

Example: Clustering

```
> library("flexmix")
> data("diabetes", package = "mclust")
> diabetes_data <- as.matrix(diabetes[, 2:4])
```

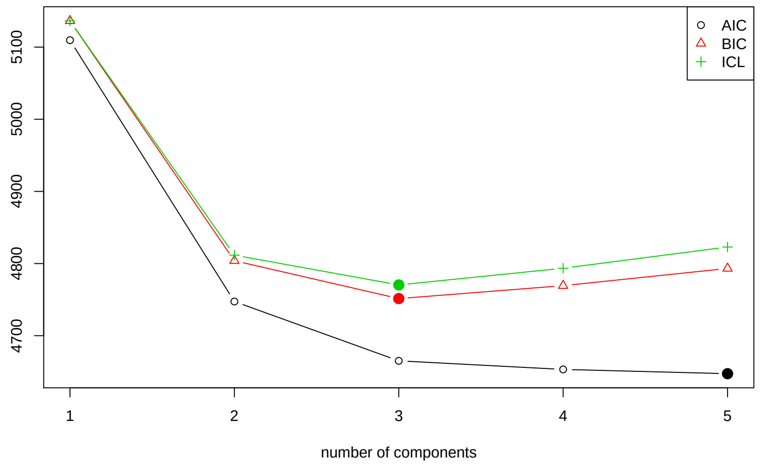
Example: Clustering



Example: Clustering

```
> (mix <- stepFlexmix(diabetes_data  ~ 1, k = 1:5,
+                        model = FLXMCmvnorm(diag = FALSE),
+                        nrep = 10))
1 : * * * * * * * * * *
2 : * * * * * * * * * *
3 : * * * * * * * * * *
4 : * * * * * * * * * *
5 : * * * * * * * * * *

Call:
stepFlexmix(diabetes_data ~ 1, model = FLXMCmvnorm(diag = FALSE),
    k = 1:5, nrep = 10)

  iter converged k k0    logLik      AIC      BIC      ICL
1    2      TRUE 1  1 -2545.833 5109.666 5136.456 5136.456
2   12      TRUE 2  2 -2354.674 4747.347 4803.905 4811.644
3   24      TRUE 3  3 -2303.557 4665.113 4751.439 4770.353
4   36      TRUE 4  4 -2287.605 4653.210 4769.302 4793.502
5   60      TRUE 5  5 -2274.655 4647.309 4793.169 4822.905

> plot(mix)
```

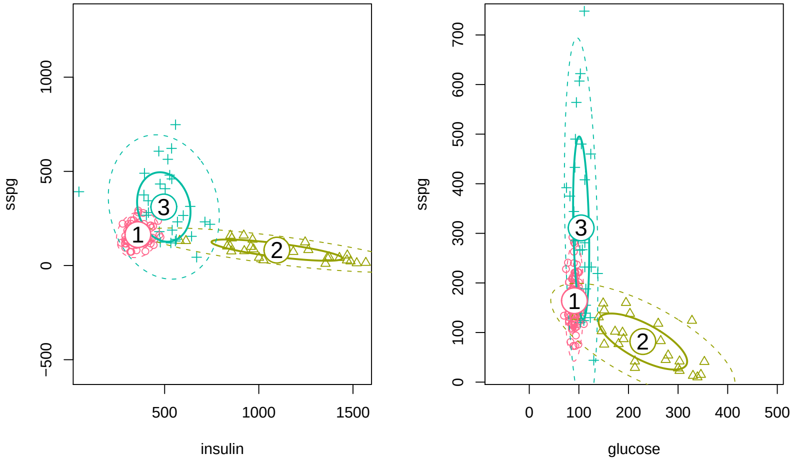
Example: Clustering



Example: Clustering

```
> (mix_best <- getModel(mix))
Call:
stepFlexmix(diabetes_data ~ 1, model = FLXMCmvnorm(diag = FALSE),
    k = 3, nrep = 10)

Cluster sizes:
 1  2  3
82 28 35

convergence after 24 iterations
> summary(mix_best)
Call:
stepFlexmix(diabetes_data ~ 1, model = FLXMCmvnorm(diag = FALSE),
    k = 3, nrep = 10)

       prior size post>0 ratio
Comp.1 0.540   82    101 0.812
Comp.2 0.199   28     96 0.292
Comp.3 0.261   35    123 0.285

’log Lik.’ -2303.557 (df=29)
AIC: 4665.113   BIC: 4751.439
```

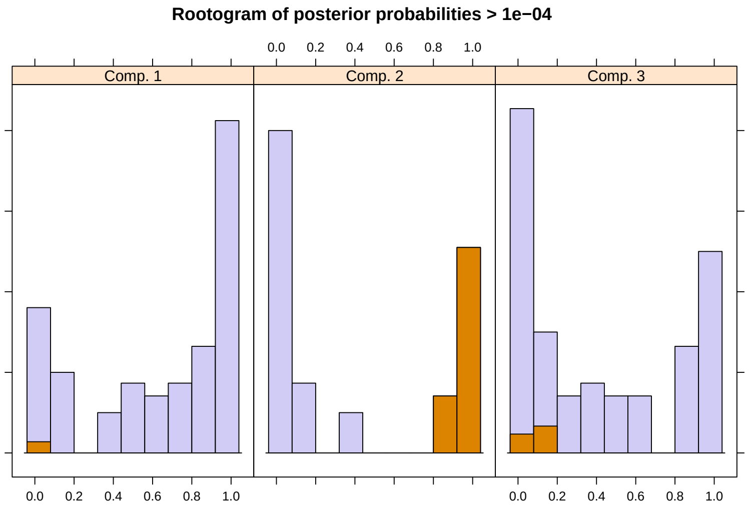
Example: Clustering



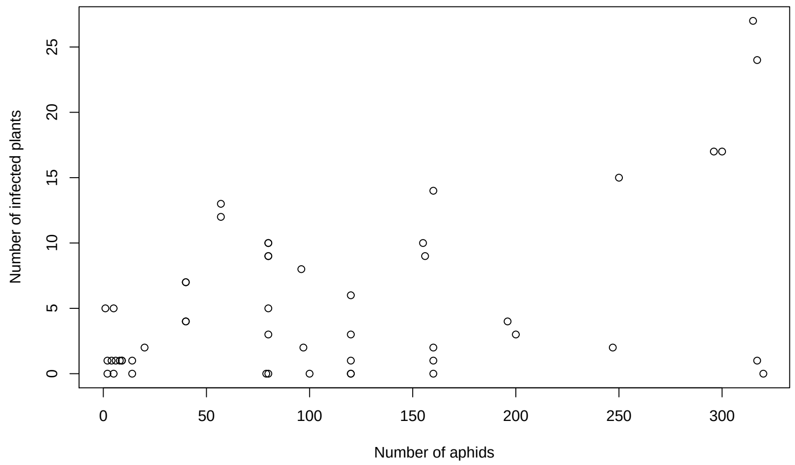
Example: Clustering

```
> table(cluster(getModel(mix)), diabetes$class)
    chemical normal overt
  1       10     72     0
  2        1      0    27
  3       25      4     6
> parameters(mix_best, component = 1, simplify = FALSE)
$center
  glucose   insulin      sspg
 91.00937 358.19098 164.14443

$cov
         glucose   insulin      sspg
glucose 58.21456   80.1404   16.8295
insulin 80.14039 2154.9810  347.6972
sspg    16.82950  347.6972 2484.1538
> plot(mix_best, mark = 2)
```

Example: Clustering



Example: Regression



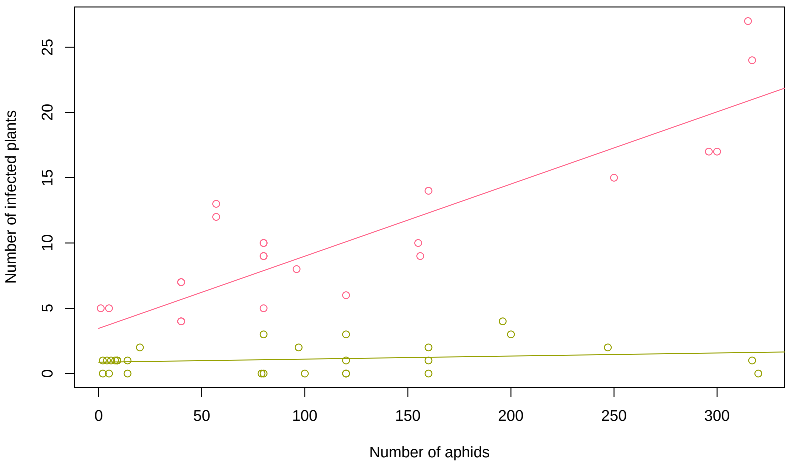
Example: Regression

```
> data("aphids", package = "mixreg")
> (mix <- stepFlexmix(n.inf ~ n.aphids, k = 2, data = aphids,
+                          nrep = 10))
2 : * * * * * * * * * *

Call:
stepFlexmix(n.inf ~ n.aphids, data = aphids, k = 2, nrep = 10)

Cluster sizes:
 1  2
23 28

convergence after 17 iterations
```

Example: Regression

```
> posterior(mix)[1:4,]
          [,1]        [,2]
[1,] 0.9949732 0.005026814
[2,] 0.9949769 0.005023128
[3,] 0.2098020 0.790198026
[4,] 0.2050383 0.794961704
> predict(mix, newdata = data.frame(n.aphids = c(0, 300)))
$Comp.1
       [,1]
1  3.458813
2 20.047842

$Comp.2
       [,1]
1 0.8679776
2 1.5740946
```

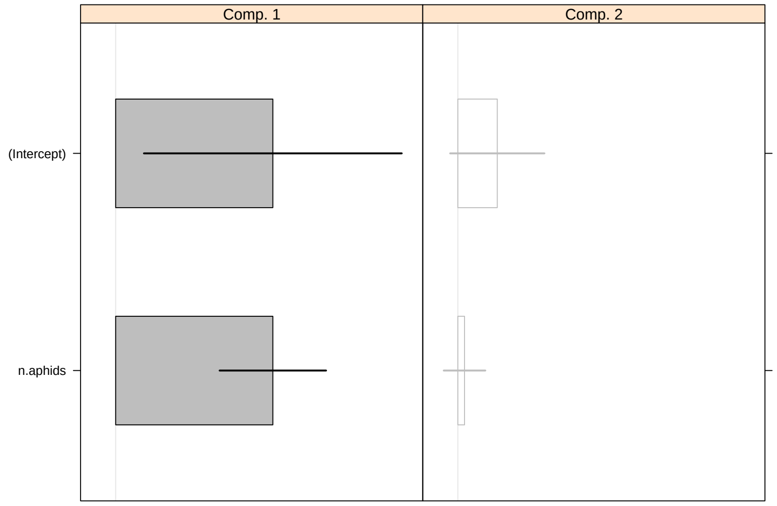
Example: Regression



Example: Regression

```
> refit(mix)
Call:
refit(mix)

Number of components: 2

$Comp.1
             Estimate Std. Error z value  Pr(>|z|)
(Intercept) 3.4585759  1.3730364  2.5189   0.01177
n.aphids    0.0552974  0.0090624  6.1019 1.048e-09

$Comp.2
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 0.8679003  0.5017007  1.7299  0.08365
n.aphids    0.0023539  0.0035375  0.6654  0.50578
> plot(refit(mix))
```

Example: Regression



Applications

Market segmentation: find groups of customers who share

- characteristics: e.g. groups of tourists with similar behaviours at their destination
- reactions: e.g. customers with similar price and other marketing mix elasticities in choice models

⇒ account for heterogeneity between customers

⇒ develop segment-specific marketing strategies

Monographs

D. Böhning.*Computer Assisted Analysis of Mixtures and Applications: Meta-Analysis, Disease Mapping, and Others*. Chapman & Hall/CRC, London, 1999.

S. Frühwirth-Schnatter. *Finite Mixture and Markov Switching Models*. Springer Series in Statistics. Springer, New York, 2006.

B. G. Lindsay. *Mixture Models: Theory, Geometry, and Applications*. The Institute for Mathematical Statistics, Hayward, California, 1995.

G. J. McLachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, 2000.

References

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM-algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

J. Diebolt and C. P. Robert. Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society B*, 56:363–375, 1994.

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. *Journal of the American Statistical Association*, 97 (458):611–631, 2002.

F. Leisch. FlexMix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, 11(8), 2004. URL `http://www.jstatsoft.org/v11/i08/`.