WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Finite mixtures of generalized linear regression models

Bettina Grün

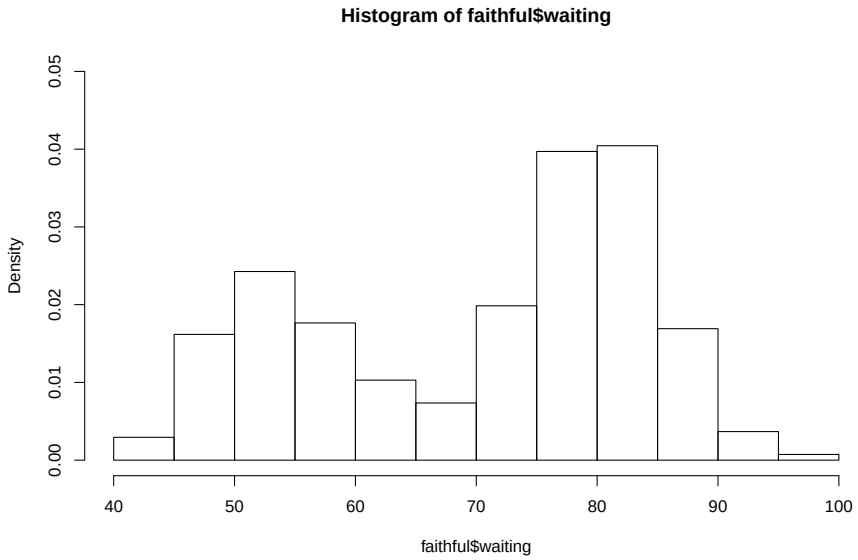
http://www.aasc.or.at/mixtures

Finite mixture models

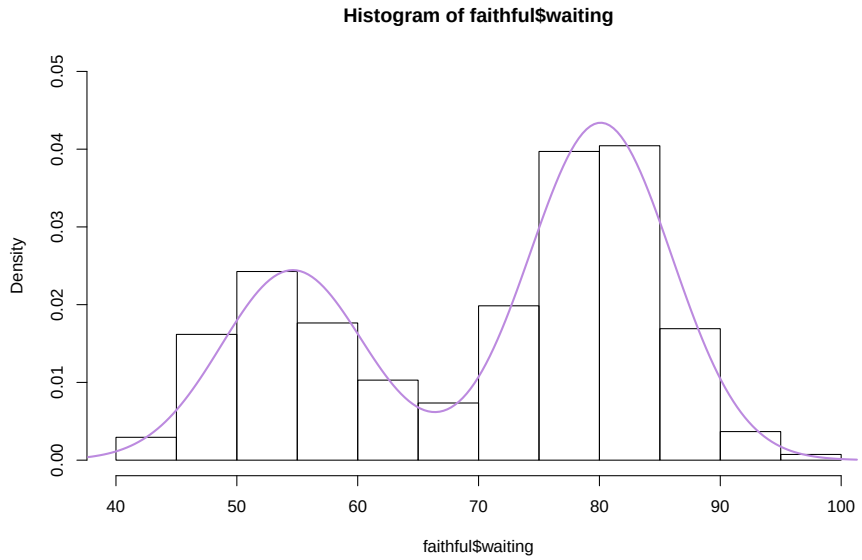
Flexible model class with special models for different kinds of data.

- Types of application:
  - semi-parametric tool to estimate general distribution functions
  - modeling unobserved heterogeneity / finding groups in data
- Areas of application:
  - astronomy
  - biology
  - economics
  - . . .

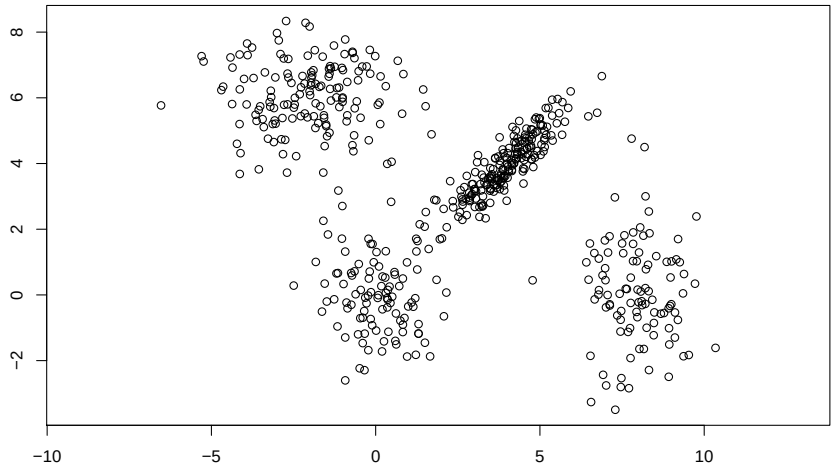
Finite mixture models / 2



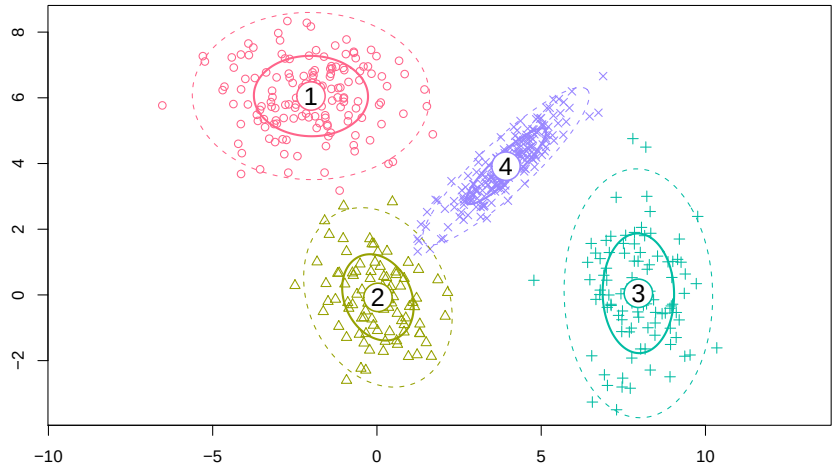
Finite mixture models / 3



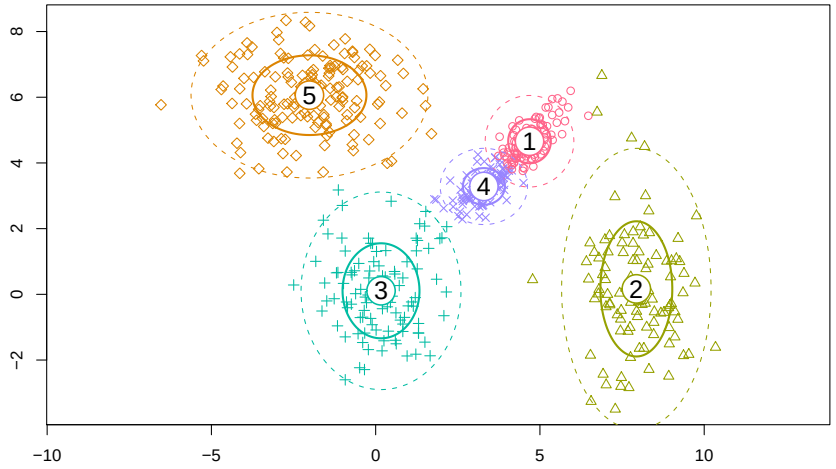
Finite mixture models / 4



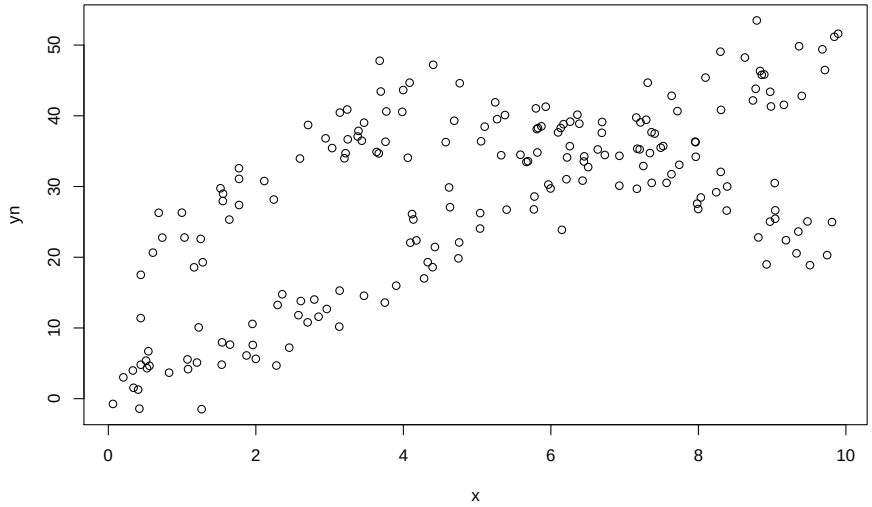
Finite mixture models / 5



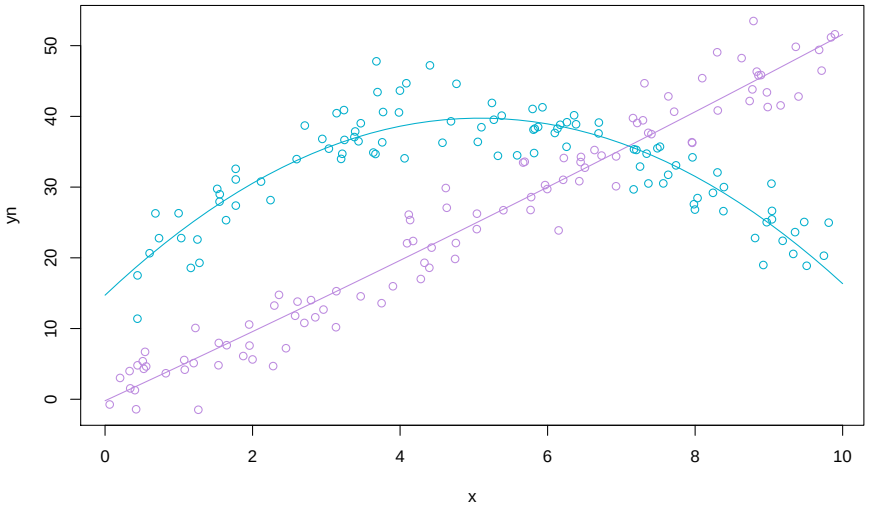
Finite mixture models / 6



Finite mixture models / 7



Finite mixture models / 8



Model definition

The finite mixture distribution is given by

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

with

$$\sum_{k=1}^{K} \pi_k(\boldsymbol{w}, \boldsymbol{\alpha}) = 1 \quad \text{and} \quad \pi_k(\boldsymbol{w}, \boldsymbol{\alpha}) > 0 \, \forall k.$$

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

Estimation

- Maximum-Likelihood estimation:
  - Direct optimization of likelihood (mostly in simpler cases)
  - Expectation-Maximization (EM) algorithm for more complicated models (Dempster et al., 1977)
  - EM followed by direct optimization for estimate of Hessian
  - ...
- Bayesian estimation:
  - MCMC, Gibbs-Sampling

EM algorithm

- 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.

Missing data

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

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

The complete log-likelihood is given by

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

EM algorithm: E-step

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

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

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

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

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 component sizes are given by:

$$\boldsymbol{\alpha}^{(i+1)} = \arg\max_{\boldsymbol{\alpha}} \sum_{n=1}^{N} \hat{z}_{nk}^{(i)} \log \pi_k(\boldsymbol{w}_n, \boldsymbol{\alpha}).$$

$\Rightarrow$ weighted ML estimation of the concomitant variable model.
If the component sizes are assumed to be constant, they are given by

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

EM algorithm: M-step / 2

The component specific parameter estimates are determined by:

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

$\Rightarrow$ weighted ML estimation of the component specific model.

EM algorithm

Advantages:

- The likelihood is increased in each step $\rightarrow$ 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 in standard software.

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

EM algorithm: variants

- **Classification EM (CEM):** assigns each observation to the component with the maximum a-posteriori probability.
  - In general faster convergence than EM.
  - Convergence to the classification likelihood.
- **Stochastic EM (SEM):** assigns each observation to one component by drawing from the multinomial distribution induced by the a-posteriori probabilities.
  - Does not converge to the "closest" local optimum given the initial values.
  - If started with too many components, some components will eventually become empty and will be eliminated.

Selecting the number of components

- **A-priori known**
- **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 between the number of parameters under the null and alternative hypotheses:
  - distributional results available for some special cases
  - bootstrapping

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, spectral clustering
- Use short runs of EM, CEM or SEM with different initializations (Biernacki et al., 2003).
- Use different subsets of the complete data: sampling (Wehrens et al., 2004) and incremental method (Fraley et al., 2005).

$\Rightarrow$ Scharl et al. (2010) recommend short runs of EM for mixtures of regression models.

Model diagnostics using resampling

- Testing for the number of components: e.g. likelihood ratio test using the parametric bootstrap (McLachlan, 1987)
- Standard deviations of parameter estimates: e.g. with parametric bootstrap with initialization in solution (Basford et al., 1997)
- Identifiability: e.g. by testing for unimodality of the component specific parameter estimates using empirical or parametric bootstrap with random initialization
- Stability of induced partitions: e.g. by comparing results using class agreement measures as the Rand index corrected for agreement by chance using empirical or parametric bootstrap with random initialization

Software in R

Model-based clustering:

- **mclust** (Fraley and Raftery, 2002) for Gaussian mixtures:
  - the structure of the variance-covariance matrices can be specified via volume, shape, and orientation
  - initialize EM algorithm with the solution from an agglomerative hierarchical clustering algorithm
- **flexmix** for binary data and mixed-mode data (Leisch, 2004; Grün and Leisch, 2008a)

Mixtures of regression models:

- **flexmix** for mixtures of GLMs

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

Mixtures of regression models

- aka Clusterwise Regression
- Regression models are fitted in each component.
  $\Rightarrow$ Weighted ML estimation of linear and generalized linear models required.
- Heterogeneity between observations with respect to regression parameters.
- Random effects can be estimated in a semiparametric way.

Possible models:

- mixtures of linear regression models
- mixtures of generalized linear models
- mixtures of generalized linear mixed models
- . . .

Identifiability

Three kinds of identifiability issues (Frühwirth-Schnatter, 2006):

- **Label switching:** impose constraint on components, as e.g., that the a-priori probabilities π_k are ascending
- **Overfitting:** leads to empty components or components with equal parameters
- **Generic unidentifiability**

Generic identifiability of mixtures of distributions

- **identifiable:** (multivariate) normal, gamma, exponential, Cauchy and Poisson component distribution functions
- **not identifiable:** continuous or discrete uniform component distribution functions
- **identifiable under certain conditions:** mixtures with binomial and multinomial component distribution functions are identifiable if

$$N \geq 2K - 1$$

  where N is the number of trials.

E.g., in Titterington et al. (1985), McLachlan & Peel (2000).

Generic identifiability of mixtures of regressions

Influencing factors:

- component distribution (see mixtures of distributions)
- covariate matrix
- repeated observations / labelled observations

Identifiability: covariate matrix

If there is only one measurement per person, full rank of the regressor matrix is not sufficient.

- **linear models:** Mixtures of linear regression models with Gaussian noise are identifiable, if the number of components K is smaller than the minimal number of (feasible) hyperplanes necessary to cover all covariate points (without intercept). (Hennig, 2000)
- **generalized linear models:** Analogous condition for linear predictor, additional conditions depending on distribution of response (especially for binomial and multinomial logit models, see Grün and Leisch, 2008b)

$\Rightarrow$ **Coverage condition**

Identifiability: repetitions / labellings

- At least one of the hyperplanes in the coverage condition has to cover all of the repeated / labelled observations where the component membership is fixed.

Violation of the coverage condition leads to:

Intra-component label switching: If the labels are fixed in one covariate point according to some ordering constraint, then labels may switch in other covariate points for different parameterizations of the model.

Illustration: binomial logit models

We consider a mixture density

$$h(y|x, \Theta) = \pi_1 f(y|N, (1, x)\beta_1) + (1 - \pi_1) f(y|N, (1, x)\beta_2)$$

with binomial logit models in the components and parameters

$$\pi_1 = 0.5 \qquad \beta_1 = (-2, 4)' \qquad \beta_2 = (2, -4)'.$$

Even if $N \geq 3$ the mixture is not identifiable if there are only 2 covariate points available. The second solution is then given by

$$\pi_1^{(2)} = 0.5 \qquad \beta_1^{(2)} = (-2, 0)' \qquad \beta_2^{(2)} = (2, 0)'.$$

Illustration: binomial logit models / 2

Simulation design:

- Number of repetitions $N \in \{1, 10\}$ in the same covariate point
- Number of covariate points: $\#x \in \{2, 5\}$, equidistantly spread across $[0, 1]$
- 100 samples with 100 observations are drawn from the model for each combination of N and $\#x$.
- Finite mixtures with 2 components are fitted to each sample.
- Solutions after imposing an ordering constraint on the intercept are reported.

Illustration: binomial logit models / 3

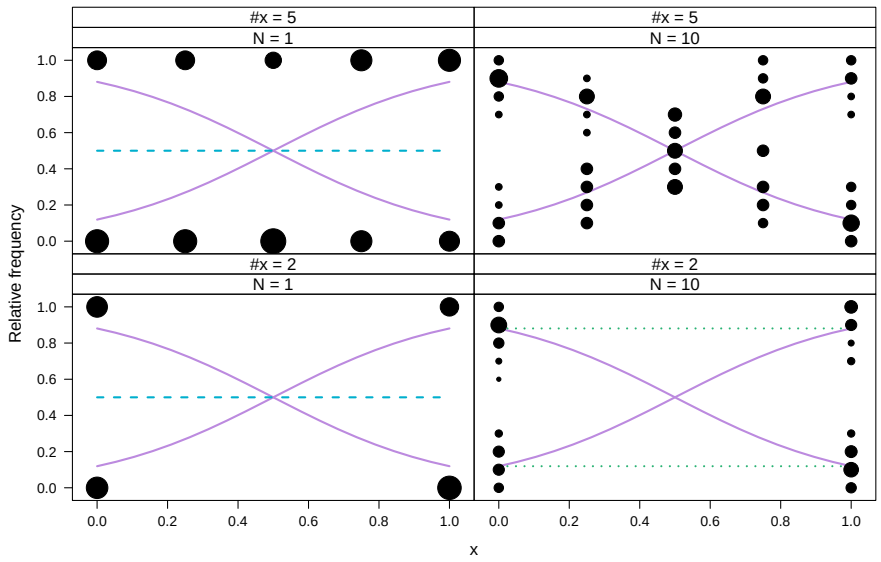


Illustration: binomial logit models / 4

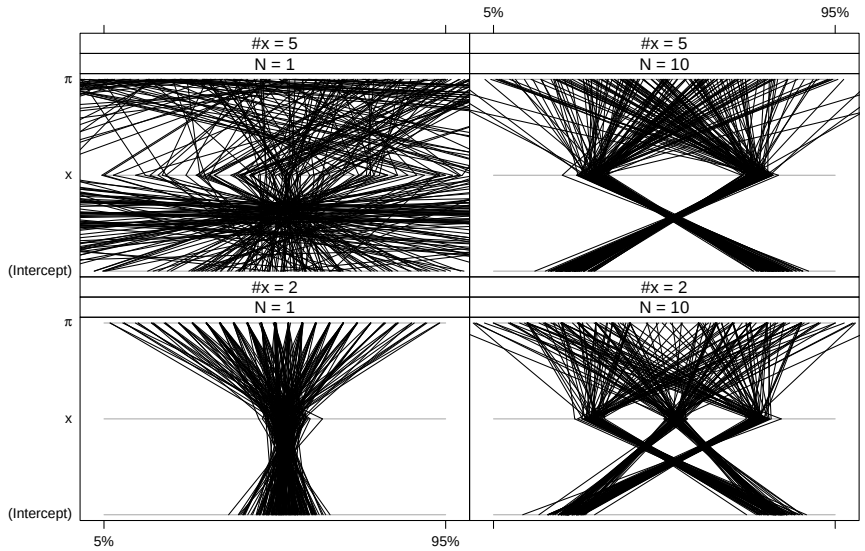
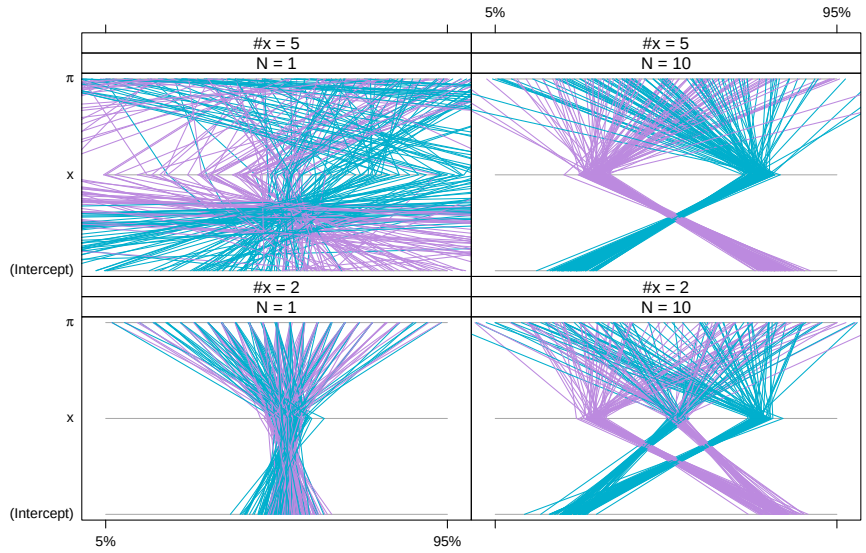


Illustration: binomial logit models / 5



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

Fit function `flexmix()`

`flexmix()` takes the following arguments:

- **formula:** A symbolic description of the model to be fit. The general form is `y~x|g` where `y` is the response, `x` the set of predictors and `g` an optional grouping factor for repeated measurements.
- **data:** An optional data frame containing the variables in the model.
- **k:** Number of clusters (not needed if `cluster` is specified).
- **cluster:** Either a matrix with `k` columns of initial cluster membership probabilities for each observation; or a factor or integer vector with the initial cluster assignments of observations.
- **model:** Object of class `"FLXM"` or list of these objects.
- **concomitant:** Object of class `"FLXP"`.
- **control:** Object of class `"FLXcontrol"` or a named list.

- repeated calls of `flexmix()` with `stepFlexmix()`
- returns an object of class `"flexmix"`

Controlling the EM algorithm

`"FLXcontrol"` for the overall behaviour of the EM algorithm:

- **iter.max:** Maximum number of iterations
- **minprior:** Minimum prior probability for components
- **verbose:** If larger than zero, then `flexmix()` gives status messages each verbose iterations.
- **classify:** One of “auto”, “weighted”, “CEM” (or “hard”), “SEM” (or “random”).

For convenience `flexmix()` also accepts a named list of control parameters with argument name completion, e.g.

```
flexmix(..., control=list(class="r"))
```

Variants of mixture models

- **Component specific models:** `FLXMxxx()`
  - Model-based clustering: `FLXMCxxx()`
    - `FLXMCmvnorm()`
    - `FLXMCmvbinary()`
    - `FLXMCmvcombi()`
    - `FLXMCmvpois()`
    - `...`
  - Clusterwise regression: `FLXMRxxx()`
    - `FLXMRglm()`
    - `FLXMRglmfix()`
    - `FLXMRziglm()`
    - `...`
- **Concomitant variable models:** `FLXPxxx()`
  - `FLXPconstant()`
  - `FLXPmultinom()`

Methods for `"flexmix"` objects

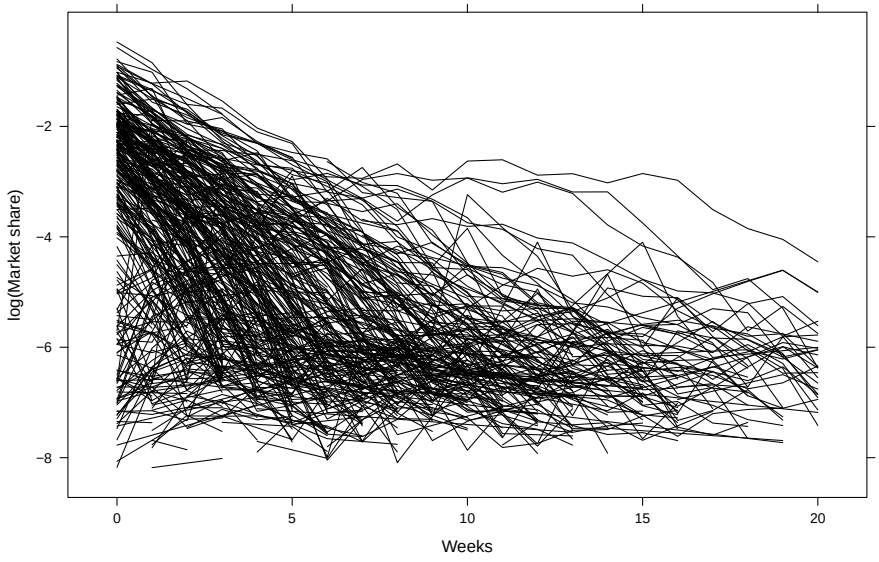
- `show()`, `summary()`: some information on the fitted model
- `plot()`: rootogram of posterior probabilities
- `refit()`: refits an estimated mixture model to obtain other additional information, such as for example the variance-covariance matrix
- `logLik()`, `BIC()`, `...`: obtain log-likelihood and model fit criteria
- `parameters()`, `prior()`: obtain component specific or concomitant variable model parameters and prior class probabilities / component weights
- `posterior()`, `clusters()`: obtain a-posteriori probabilities and assignments to the maximum a-posteriori probability
- `fitted()`, `predict()`: fitted and predicted (component-specific) values

Example: market share patterns of movies

- Box office and theaters data for 407 movies playing between May 5, 2000 and December 7, 2001 were collected from a popular website of movie records (`www.the-number.com`, cp. Krider et al., 2005).
- The gross box-office takings for the 40 most popular movies for each weekend in the time period are recorded.
- The gross takings are transformed into market shares to account for the difference in volume between weekends.
- The market share is used as dependent variable and the number of weeks since release of the movie as covariate.
- The data is restricted to the first 20 weeks after release. This reduces the number of movies in the data set to 394.
- On average 8 observations are available for each movie. In total 3149 observations are available.

A similar analysis is described in Jedidi et al. (1998).

Example: market share patterns of movies / 2



Example: market share patterns of movies / 3

- Most movies exhibit an exponential decay in market share.
  $\Rightarrow$ Log of market share is modelled.
- The component membership is fixed over the weeks for each movie.

$$h(\log(\texttt{share})|\texttt{week}, \Theta) = \sum_{k=1}^{K} \pi_k f_N(\log(\texttt{share})|\mu_k(\texttt{week}), \sigma_k^2),$$

with the mean given by

$$\mu_k = \beta_{1k} + \texttt{week}\beta_{2k}.$$

Example: market share patterns of movies / 4

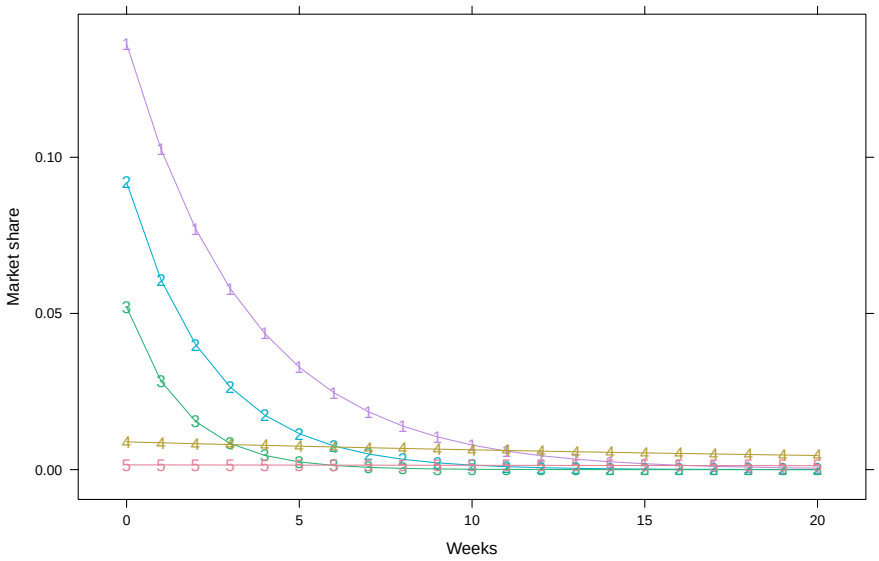
Estimation:

- Finite mixtures are fitted with 1 to 10 components.
- For each number of components the EM algorithm is repeated 10 times with random initialization.
- Components with a weight of less than 0.1 are omitted during the run of the algorithm.
- The BIC criterion is used to determine the optimal number of components.

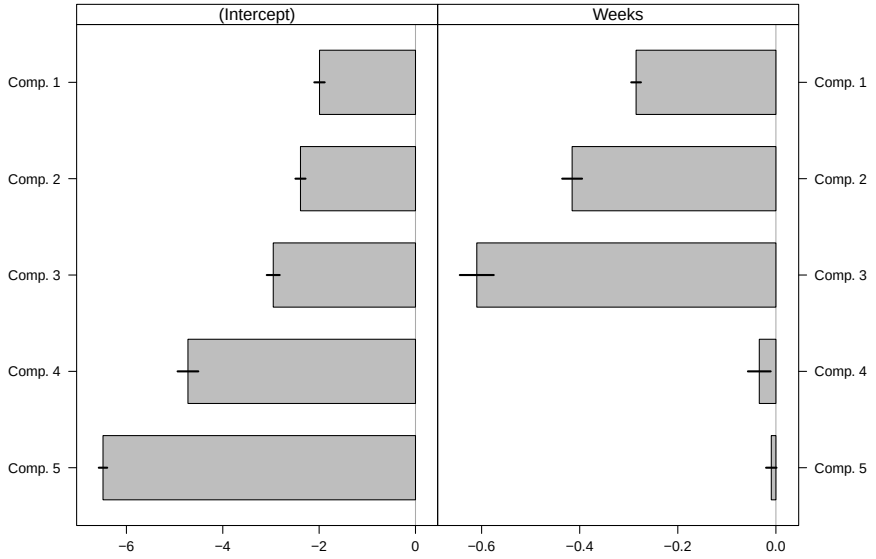
Results:

- The EM algorithm did not converge to a mixture with more than 5 components.
- The BIC suggests 5 components.

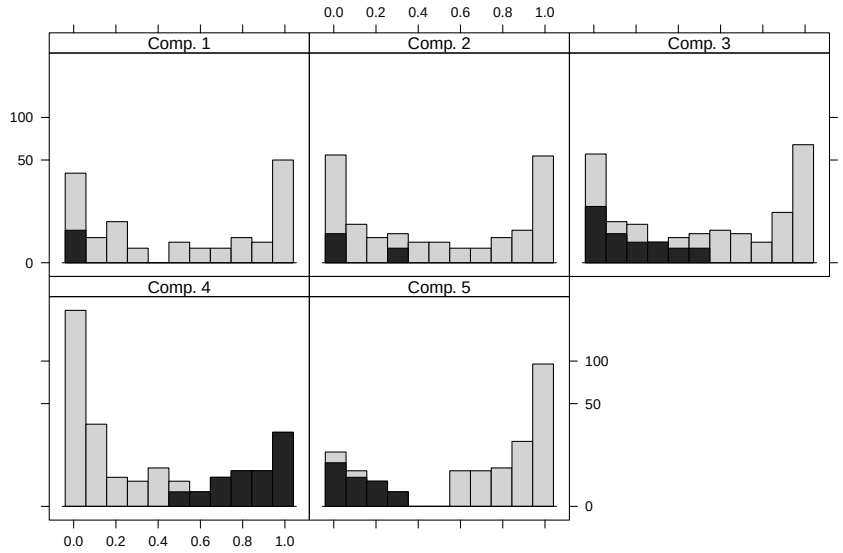
Example: market share patterns of movies / 5



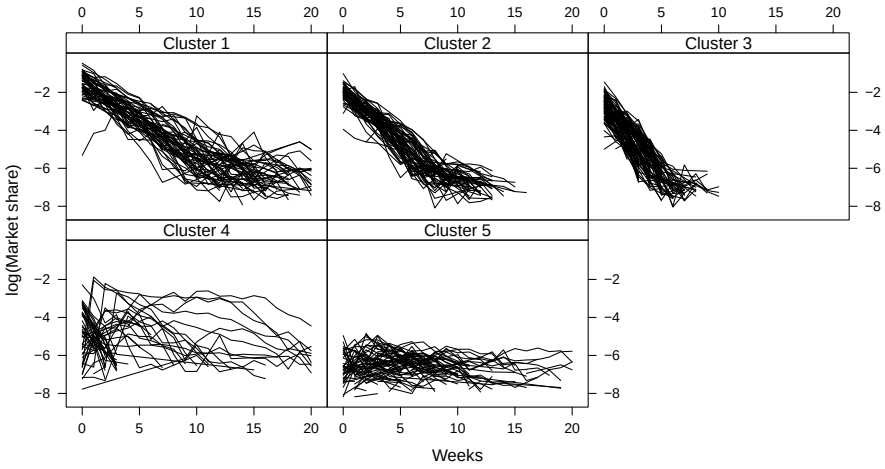
Example: market share patterns of movies / 6



Example: market share patterns of movies / 7



Example: market share patterns of movies / 8



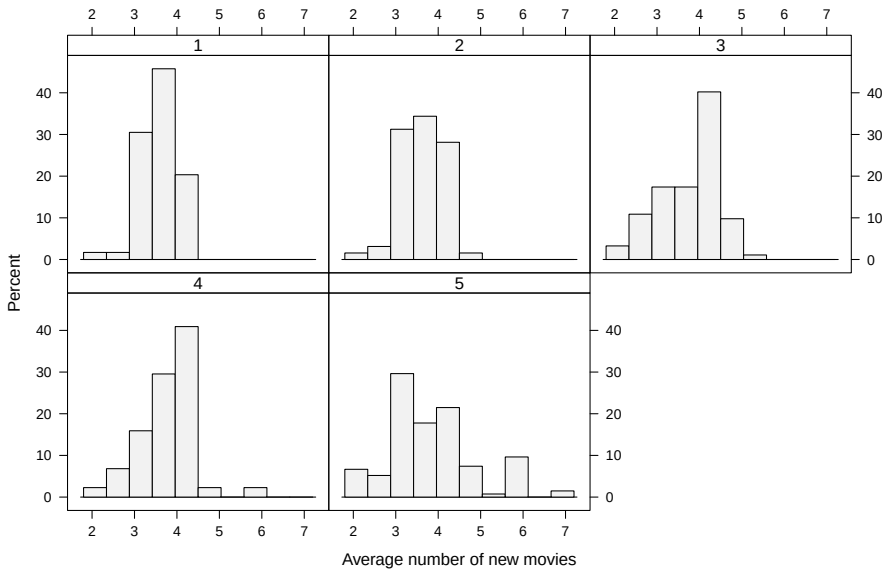
Example: market share patterns of movies / 9

Characteristics of movies in each of the 5 clusters with respect to competitive intensity and distribution.

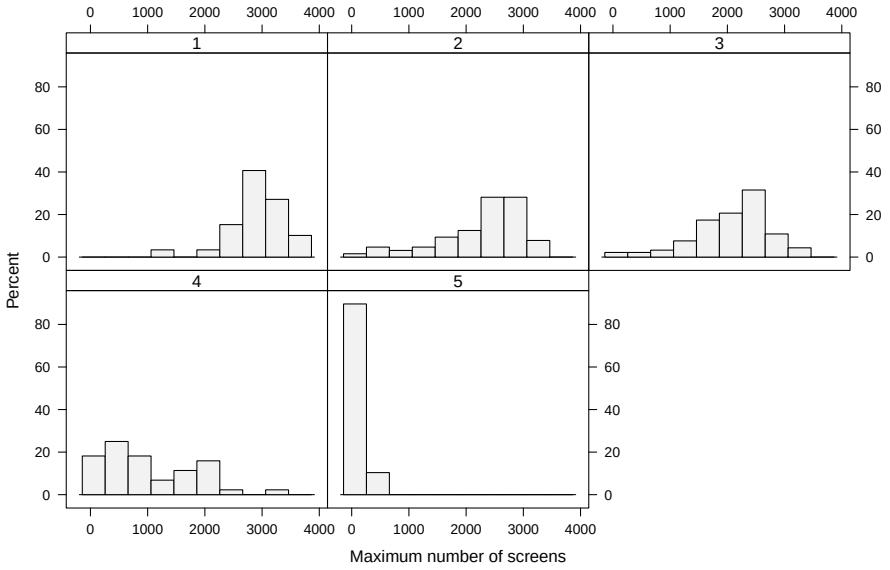
- **Average number of new movies** opening per week over the entire run of the movie.
- **Maximum number of screens** the movie was played in any week over its run.

	Cluster				
	1	2	3	4	5
Avg. # of new movies	3.6	3.6	3.8	3.7	3.8
Max. # screens	2821.4	2269.9	2032.6	1017.0	106.5

Example: market share patterns of movies / 10



Example: market share patterns of movies / 11



Summary

- Finite mixture models are a flexible model class.
  $\Rightarrow$ Different component specific models are possible.
- For estimating new mixture models the M-step needs to be available, i.e. weighted ML estimation of the component-specific model and the concomitant variable model.
- (Practical) identifiability problems are possible, especially for mixtures of regression models.
- Package **FlexMix** is available from CRAN

  `http://cran.r-project.org/package=flexmix`.

References

K.E. Basford, D.R. Greenway, G.J. McLachlan, and D. Peel. Standard errors of fitted component means of normal mixtures. *Computational Statistics*, 12: 1–17, 1997.

C. Biernacki, G. Celeux, and G. Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, 41:561–575, 2003.

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.

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.

C. Fraley, A.E. Raftery and R. Wehrens. Incremental model-based clustering for large datasets with small clusters. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2005.

References / 2

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

B. Grün and F. Leisch. FlexMix version 2: finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4), 2008a. URL `http://www.jstatsoft.org/v28/i04/`.

B. Grün and F. Leisch. Identifiability of finite mixtures of multinomial logit models with varying and fixed effects. *Journal of Classification*, 25(2): 225–247, 2008b.

C. Hennig. Identifiability of models for clusterwise linear regression. *Journal of Classification*, 17(2):273–296, 2000.

K. Jedidi, R.E. Krider, and C.B. Weinberg. Clustering at the movies. *Marketing Letters*, 9(4):393–405, 1998.

R.E. Krider, T. Li, Y. Liu, and C.B. Weinberg. The lead-lag puzzle of demand and distribution: a graphical method applied to movies. *Marketing Science*, 24 (4):635–645, 2005.

References / 3

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/`.

G.J. McLachlan. On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Applied Statistics*, 36(3):318–324, 1987.

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

T. Scharl, B. Grün and F. Leisch. Mixtures of regression models for time-course gene expression data: evaluation of initialization and random effects. *Bioinformatics*, 26(3):370–377.

D.M. Titterington, A.F.M. Smith and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.

R. Wehrens, L.M.C. Buydens, C. Fraley, and A.E. Raftery. Model-based clustering for image segmentation and large datsets via sampling. *Journal of Classification*, 21:231–253, 2004.