Fitting Finite Mixtures of Generalized Linear Regressions in R

Bettina Grün Vienna University of Technology

Friedrich Leisch University of Munich

This is a preprint of an article accepted for publication in:
Computational Statistics and Data Analysis, 2006.
`http://www.elsevier.com/locate/csda`

Abstract

R package **flexmix** provides flexible modelling of finite mixtures of regression models using the EM algorithm. Several new features of the software such as fixed and nested varying effects for mixtures of generalized linear models and multinomial regression for a-priori probabilities given concomitant variables are introduced. The use of the software in addition to model selection is demonstrated on a logistic regression example.

Key words: concomitant variable, finite mixture, fixed effect, generalized linear model, R.

1 Introduction

Finite mixtures of regression models are a popular method to model unobserved heterogeneity or to account for overdispersion in data. They are flexible models and in theory it is easy to modify and extend them by using more complex models for the component distribution functions and estimate the corresponding parameters, e.g., using the EM algorithm.

R (R Development Core Team, 2006) features several extension packages for estimation of mixture regression models, e.g., **fpc** for mixtures of linear regression models (Hennig, 2000) and **mmlcr** for mixed-mode latent class regression (Buyske, 2006). However, like virtually all other (non-R) implementations, they consider only a few particular types of mixture models and do not reflect the generality of the theoretical model class in the software design. R package **flexmix** (Leisch, 2004) tries to fill this gap by encapsulating the abstract statistical objects of interest into S4 classes and methods such that the resulting software can be easily extended.

This paper is organized as follows: Section 2 gives notation and the model class, the main new functions of **flexmix** are presented in Section 3, and we end with a short demonstration in Section 4. The latest development version of the package sources and all R code necessary to reproduce the results in this article are available from `http://www.ci.tuwien.ac.at/research/mixtures`.

2 Model specification

We consider finite mixtures of regression models of form

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

where $\boldsymbol{\Theta}$ denotes the vector of all parameters, y the dependent, $\boldsymbol{x}$ the independent, $\boldsymbol{w}$ the concomitant variable, and F is the component specific distribution function. For component-wise generalized linear models (GLMs), F must be a member of the exponential family (McCullagh and Nelder, 1989). The component specific parameters are the regression coefficients $\boldsymbol{\beta}_k$ and dispersion parameters ϕ_k. The component weights π_k need to satisfy

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

where $\boldsymbol{\alpha}$ are the parameters of the concomitant variable model.

Different concomitant variable models are possible to determine the component weights (Dayton and Macready, 1988), as the mapping function only has to fulfill condition (1). In the following a multinomial logit model for the π_k is assumed with the first component as baseline.

This class of finite mixtures of generalized linear models with concomitant variable models is given in McLachlan and Peel (2000, p. 145). Special cases are for example random intercept models (see Follmann and Lambert, 1989; Aitkin, 1999) where the coefficients of all independent variables are assumed to be equal over the mixture components.

Our software implementation allows to specify such equality constraints for parameters over mixture components: $(\boldsymbol{\beta}'_k, \phi_k)$ may be restricted to be equal over all components, to vary between groups of components, or to be different for all components. Variation between groups is referred to as varying effects with one level of nesting. In addition, each (group of) components may use different sets of covariates. Due to space restrictions we cannot give full details of parameter estimation, but extension from standard linear models (Grün and Leisch, 2006) to GLMs is rather straightforward.

3 Design principles

Functions and model formulae are first class objects in the S language, which allows in combination with the lexical scoping rules of R (Gentleman and Ihaka, 2000) for very modular software design. Rather than using text mode arguments used as switches within function bodies, **flexmix** uses driver functions to specify all aspects of the mixture model. Users can either use the growing collection of drivers distributed as part of **flexmix**, or write and use their own drivers.

In a first step the (unfitted) component specific model $F(y|\boldsymbol{x}, \boldsymbol{\beta}_k, \phi_k)$ and the concomitant variable model $\pi(\boldsymbol{w}, \boldsymbol{\alpha})$ have to be specified. For this no data are needed, only the names of the independent and dependent variables and their respective interaction structure are defined.

`FLXglm()` only allows varying effects for the coefficients and the dispersion parameters. In this case the likelihood can be maximized separately for each component in the M-step of the EM algorithm. If there are also fixed and nested varying effects for the regression coefficients and dispersion parameters, our new driver `FLXglmFix()` has to be used and the likelihood is maximized simultaneously for all components. The design matrix is constructed by replicating the observations K times with suitable columns of zeros added. Model formulae for the varying, nested varying and fixed effects have to be provided. These are evaluated by successively updating the formula of the random effects with the formula for the fixed and then the nested varying effects.

The concomitant variable model is specified in a similar fashion. The default dummy driver `FLXconstant()` uses no concomitant variables and acts only as a placeholder. For multinomial logistic regression our new function `FLXmultinom()` can be used (see example section). The main estimation engine of **flexmix** has changed to be able to use the new functionality, however this are changes behind the scenes in unexported functions, all existing user code should run unaffected.

By default, EM is initialized using random assignment of observations to mixture components, function `stepFlexmix()` can be used to automatically try out several initializations. We now also provide the choice of starting EM with user-specified posteriors or (more common) posteriors from a previous run. To select a model with a suitable number of components information criteria such as the Akaike information criterion (AIC), the Bayesian information criterion (BIC) and the integrated completed likelihood information criterion (ICL; Biernacki et al., 2000) can be used.

`flexmix()` returns an object of class `flexmix` and methods defined for this class include `show()`, `summary()` and `plot()`. `show()` gives the call, the table of cluster assignments and the number of iterations until convergence. Further details are given by `summary()` which provides the prior probabilities together with the table of cluster assignments, the number of observations with a-posteriori probability larger than `eps` and the ratio of these numbers, which indicates how well separated the components are. In addition the likelihood (with degrees of freedom used), the AIC and the BIC are printed. The default plot is a rootogram of the a-posteriori probabilities for each component. In addition there are accessor functions for the component specific parameters (`parameters()`), for the a-posteriori probabilities (`posterior()`), the maximum a-posteriori class assignments (`cluster()`) and the fitted values for each component (`fitted()`). More information on the estimated parameters of the component specific and concomitant variable models can be obtained using `refit()` and the corresponding `summary()` method (see example section).

4 Logistic Regression Example

We now illustrate model fitting and model selection in R on simple artificial data from a mixture of binomial regression models. More examples for both logistic regression and other members of the GLM family are provided as part of the software package through a collection of artificial and real world data sets, most of which have been previously used in the literature (see references in the online help pages). Each data set can be loaded to R with `data(name)` and the fitting of the proposed models can be replayed using `example(name)`. Further details on these examples are given in a user guide which can be accessed using `vignette("regression-examples", package="flexmix")` from within R.

The artificial data considered here are sampled from a mixture distribution with three components and with varying effects for the intercept and nested varying effects for covariate x. The mixture distribution is given by:

$$H(y|x, w, \mathbf{\Theta}) = \sum_{s=1}^{3} \pi_s(w, \boldsymbol{\alpha}) \mathrm{Bi}(y|T, \theta_s)$$

where $\mathrm{Bi}(\cdot|T, \theta)$ denotes the binomial distribution with success probability θ and number of trials T. The success probabilities are given by

$$\begin{aligned}
\mathrm{logit}(\theta_1) &= x\beta_{2,1} + \beta_{3,1} \\
\mathrm{logit}(\theta_2) &= x\beta_{2,1} + \beta_{3,2} \\
\mathrm{logit}(\theta_3) &= x\beta_{2,2} + \beta_{3,3}
\end{aligned}$$

where $\beta_{2,\cdot} = (2, 0)$ and $\beta_{3,\cdot} = (-4, 1, 3)$. The component weights depend on the variable w and are determined by

$$\begin{aligned}
&\text{Class 2: } \mathrm{logit}[\pi_2(w, \boldsymbol{\alpha})] = 1 - w \\
&\text{Class 3: } \mathrm{logit}[\pi_3(w, \boldsymbol{\alpha})] = w.
\end{aligned}$$

A random sample with 200 observations is drawn from this mixture distribution for $T = 20$, x standard Gaussian and w from the set $\{0, 1\}$ with equal probability (and independent of x). The observations are plotted separately for the two levels of w in Figure 1, the plotting symbol corresponds to the true component membership. It can be clearly seen that most observations are from Class 2 for $w = 0$ and from Class 3 for $w = 1$.

In practice the true structure of the data is unknown, so we start by fitting a full model with different parameters for each component, the corresponding R code is shown in Figure 2. After loading package and data as well as setting a random seed, we define the concomitant variable model and store it in object `Conc`. Then we define the full model using function `FLXglm()` and store it in `Model.1`, note that the actual data have not been used so far. Finally, we fit a 3-component mixture model using `nrep=5` replications of the EM algorithm and store the best in `Fitted.1`. Selection of the correct number of components using AIC or BIC is straightforward in this simple example, consult the user guide cited above for details. Instead, we will concentrate on determining the correct structure for the fixed and varying effects in this paper.

Figure 3 depicts the values of the intercept and coefficients for covariate x together with 95% confidence intervals. The intercepts in the three components are all different and the confidence intervals do not overlap. For x we get a completely different picture: The coefficient for Component 3 is almost zero (and hence greyed out), and the confidence intervals for the other two

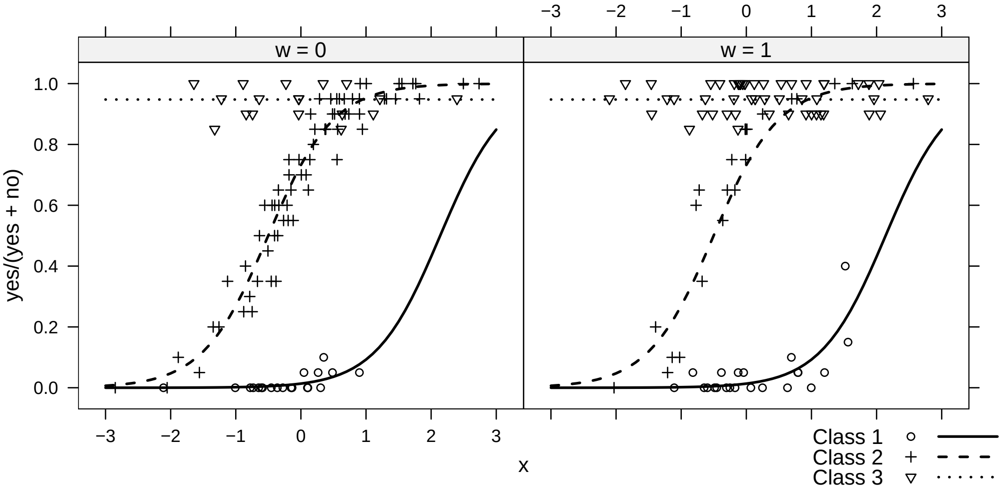

Figure 1: Sample with 200 observations from a mixture of binomial regression models. The plotting symbols correspond to the true component memberships and the lines are the fitted values.

```
> library(flexmix)
> data(BregFix)
> set.seed(4)
> Conc <- FLXmultinom(~w)
> Model.1 <- FLXglm(~x, family = "binomial")
> Fitted.1 <- stepFlexmix(cbind(yes, no) ~ 1, data = BregFix,
+     model = Model.1, k = 3, concomitant = Conc, nrep = 5)
> Model.2 <- FLXglmFix(~1, nested = list(formula = c(~x,
+     ~0), k = c(2, 1)), family = "binomial")
> Fitted.2 <- flexmix(cbind(yes, no) ~ 1, data = BregFix,
+     model = Model.2, concomitant = Conc, cluster = posterior(Fitted.1))
```

Figure 2: Fitting mixtures of binomial regression models without constraints (`Model.1`) and with grouped varying effects (`Model.2`).

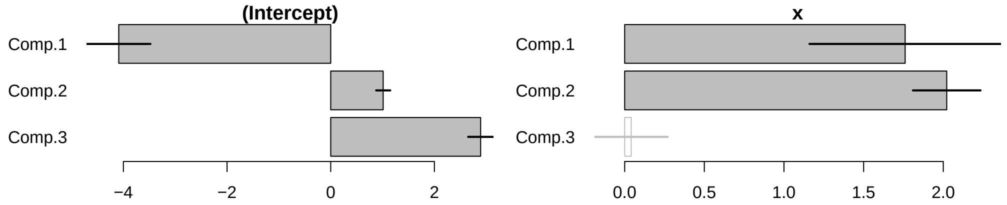

Figure 3: Coefficients of the larger `Model.1`.

components overlap. Note that the confidence intervals are not taking into account that the components have been estimated simultaneously and are not independent, hence overlaps with zero or other components should only be interpreted as hints for model selection, not as formal significance tests.

We now use function `FLXglmFix()` to specify a more parsimonious model: We have a varying effect for the intercept (formula `~1`) and restrict the first two components to have the same coefficient for x (nested formula `~x`) and the third component to have only the intercept (nested formula `~0`). The concomitant variable model remains unchanged. To get the same ordering of the components and speed up computations we initialize EM with the posteriors of the first model. `Fitted.1` has a BIC of 903.57, while the BIC of `Fitted.2` is 893.58, so the smaller model is prefered (AIC and ICL lead to the same result). Details of the smaller model are shown in Figure 4, all coefficients differ from zero and between components. Thus the correct model would have been obtained even without knowledge of the true data generating process. Figure 1 shows the corresponding predicted values as lines.

5 Summary & outlook

R package **flexmix** provides functionality for fitting models from a general class of mixtures of regressions. This class includes popular special cases as for example random intercept models. Model specification and estimation is possible with a consistent and convenient user interface as differences in estimation are hidden from the user.

In the future we want to work on high-level tools for model selection and diagnostics. An expedient extension of the provided model class is to allow for component specific offsets instead of only an overall offset, as this would allow to fit zero-inflated models. Another line of research are more robust models, e.g. by using a background noise component. For large numbers of observations and/or components the block matrices needed for the design of

```
> summary(refit(Fitted.2))

Call:
refit(Fitted.2)

Model:

Component 1 :
            Estimate Std. Error z value  Pr(>|z|)
x            2.00419    0.10094  19.856 < 2.2e-16
(Intercept) -4.28641    0.22399 -19.137 < 2.2e-16

-------------
Component 2 :
            Estimate Std. Error z value  Pr(>|z|)
x           2.004194   0.100936  19.856 < 2.2e-16
(Intercept) 1.005767   0.068437  14.696 < 2.2e-16

-------------
Component 3 :
            Estimate Std. Error z value  Pr(>|z|)
(Intercept)  2.89578    0.12147  23.840 < 2.2e-16


Concomitant Variables:

Component 2 :
            Estimate Std. Error z value  Pr(>|z|)
(Intercept)  1.19611    0.25460  4.6979 2.628e-06
w1          -1.23517    0.39169 -3.1535  0.001614

-------------
Component 3 :
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.031747   0.318045 -0.0998  0.92049
w1           0.783617   0.406279  1.9288  0.05376
```

Figure 4: Summary and parameters of the model with nested varying effects.

nested effects may use too much memory and using sparse matrix algebra may help to reduce the effect.

Acknowledgemnts

This research was supported by the Austrian Academy of Sciences (ÖAW) through a DOC-FFORTE scholarship for Bettina Grün and the Austrian Science Foundation (FWF) under grant P17382.

References

Aitkin, M., September 1999. Meta-analysis by random effect modelling in generalized linear models. Statistics in Medicine 18 (17–18), 2343–2351.

Biernacki, C., Celeux, G., Govaert, G., July 2000. Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (7), 719–725.

Buyske, S., 2006. R Package `mmlcr`: Mixed-Mode Latent Class Regression. Version 1.3.5.
URL `http://www.stat.rutgers.edu/~buyske/software.html`

Dayton, C. M., Macready, G. B., March 1988. Concomitant-variable latent-class models. Journal of the American Statistical Association 83 (401), 173–178.

Follmann, D. A., Lambert, D., March 1989. Generalizing logistic regression by non-parametric mixing. Journal of the American Statistical Association 84 (405), 295–300.

Gentleman, R., Ihaka, R., 2000. Lexical scope and statistical computing. Journal of Computational and Graphical Statistics 9 (3), 491–508.

Grün, B., Leisch, F., 2006. Fitting finite mixtures of linear regression models with varying & fixed effects in R. In: Rizzi, A., Vichi, M. (Eds.), Compstat 2006—Proceedings in Computational Statistics. Physica Verlag, Heidelberg, Germany, pp. 853–860.

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

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

McCullagh, P., Nelder, J. A., 1989. Generalized Linear Models, 2nd Edition. Chapman and Hall.

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

R Development Core Team, 2006. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org