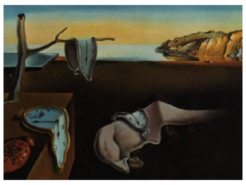


Journal of Statistical Software

August 2018, Volume 86, Issue 2. *doi: 10.18637/jss.v086.i02*

flexCWM: A Flexible Framework for Cluster-Weighted Models

Angelo Mazza
University of Catania

Antonio Punzo
University of Catania

Salvatore Ingrassia
University of Catania

Abstract

Cluster-weighted models (CWMs) are mixtures of regression models with random covariates. However, besides having recently become rather popular in statistics and data mining, there is still a lack of support for CWMs within the most popular statistical suites. In this paper, we introduce **flexCWM**, an R package specifically conceived for fitting CWMs. The package supports modeling the conditioned response variable by means of the most common distributions of the exponential family and by the t distribution. Covariates are allowed to be of mixed-type and parsimonious modeling of multivariate normal covariates, based on the eigenvalue decomposition of the component covariance matrices, is supported. Furthermore, either the response or the covariates distributions can be omitted, yielding to mixtures of distributions and mixtures of regression models with fixed covariates, respectively. The expectation-maximization (EM) algorithm is used to obtain maximum-likelihood estimates of the parameters and likelihood-based information criteria are adopted to select the number of groups and/or a parsimonious model. For the component regression coefficients, standard errors and significance tests are also provided. Parallel computation can be used on multicore PCs and computer clusters, when several models have to be fitted. To exemplify the use of the package, applications to artificial and real datasets, included in the package, are presented.

Keywords: cluster-weighted models, EM algorithm, mixture models, model-based clustering, random covariates.

1. Introduction

When data at hand are composed by a response variable Y and by a set $\boldsymbol{X}$ of d covariates, say $(\boldsymbol{X}, Y)$, and there is a latent source of heterogeneity splitting the data into groups (or clusters) possibly with different regression relationships, mixtures of regression models with fixed covariates (see, e.g., DeSarbo and Cron 1988 and Frühwirth-Schnatter 2006) constitute a reference framework of analysis. However, by assuming fixed covariates, modeling of $\boldsymbol{X}$ is not considered; furthermore, the assignment of the data points $(\boldsymbol{x}, y)$ to the clusters is required to be independent from the covariates distribution, as noted by Hennig (2000). This *assignment independence* assumption is generally not true in observational studies, and makes mixtures of regression models with fixed covariates inadequate in many real data applications. Mixtures of regression models with random covariates overcome this problem by allowing for *assignment dependence*: the component distributions for $\boldsymbol{X}$ can also be distinct and they can affect the assignment of the data points to the clusters. Therefore, they are often to be preferred in real data analyses (Hennig 2000). For a comparison between the two approaches, see also Ingrassia, Minotti, and Vittadini (2012) and Ingrassia and Punzo (2016).

A member of the class of mixtures of regression models with random covariates is the cluster-weighted model (CWM; Gershenfeld 1997). The CWM assumes a (parametric) functional relation for the local expectation of $Y|\boldsymbol{X}=\boldsymbol{x}$, and factorizes the local joint distribution $p\left(\boldsymbol{x},y\right)$ into the product between the conditional distribution of $Y|\boldsymbol{X}=\boldsymbol{x}$ and the marginal distribution of $\boldsymbol{X}$. Some recent developments in CWMs can be found in Subedi, Punzo, Ingrassia, and McNicholas (2013, 2015), Punzo (2014), Ingrassia, Minotti, and Punzo (2014), Ingrassia, Punzo, Vittadini, and Minotti (2015), Berta, Ingrassia, Punzo, and Vittadini (2016), Punzo and McNicholas (2017), and Dang, Punzo, McNicholas, Ingrassia, and Browne (2017).

In this paper, we introduce the R (R Core Team 2018) package **flexCWM**, available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=flexCWM`, specifically conceived for fitting CWMs. The package supports modeling of the conditioned response variable by means of the most common distributions of the exponential family and by the t distribution. Covariates may be of mixed-type; supported distributions are multivariate Gaussian, multinomial, binomial, and Poisson. Following Banfield and Raftery (1993) and Celeux and Govaert (1995), parsimonious modeling for multivariate normal covariates, based on the eigenvalue decomposition of the component covariance matrix, is supported (see Punzo and Ingrassia 2015). The expectation-maximization (EM) algorithm is used to obtain maximum-likelihood estimates of the parameters and several likelihood-based information criteria are adopted to select the number of groups and/or a parsimonious model. For the local regression coefficients, standard errors and significance tests are also provided.

Several CRAN packages, supporting modeling by mixtures of regressions, are available. A list of them may be found in the CRAN task view "Cluster Analysis & Finite Mixture Models" of Leisch and Grün (2018), in the section entitled "Cluster-wise Regression". Package **flexmix** is one of the most widely used packages for mixtures of regression models (Leisch 2004) and mixtures of regression models with concomitant variables (Grün and Leisch 2008); it implements an user-extensible framework for estimation, via the EM algorithm. Other packages for mixtures of regression models, include: package **fpc** for mixtures of linear regression models and fixed point clusters for linear regression (Hennig 2018), package **mixreg** for mixtures of one-variable regression models (Turner 2018), and package **mixtools**, which provides a set of functions for analyzing a variety of finite mixture models, including mixtures of regression models with fixed covariates (see Benaglia, Chauveau, Hunter, and Young 2009, Section 5). Within this context, package **flexCWM** aims at giving support for cluster-weighted modeling, providing also an alternative for estimating other classical mixture models.

The paper is organized as follows. Section 2 retraces the CWM implemented in the **flexCWM** package, Section 3 outlines the EM algorithm for maximum likelihood parameter estimation, and Section 4 illustrates further computational/operational aspects. The relevance of the package is shown, via real and artificial data sets, in Section 5, and conclusions are finally given in Section 6.

2. Model specification

The distribution of $(\boldsymbol{X}, Y)$, according to a CWM, can be written as

$$p(\boldsymbol{x}, y; \boldsymbol{\vartheta}) = \sum_{j=1}^{k} \pi_j p\left(y|\boldsymbol{x}; \boldsymbol{\beta}_j, \boldsymbol{\gamma}_j\right) p(\boldsymbol{x}; \boldsymbol{\alpha}_j), \tag{1}$$

where, for the jth component, π_j is the mixing proportion, with $\pi_j > 0$ and $\sum_{j=1}^{k} \pi_j = 1$, $p\left(y|\boldsymbol{x}; \boldsymbol{\beta}_j, \boldsymbol{\gamma}_j\right)$ is the parametric (with respect to $\boldsymbol{\beta}_j$ and $\boldsymbol{\gamma}_j$) distribution of $Y|\boldsymbol{X} = \boldsymbol{x}$, and $p(\boldsymbol{x}; \boldsymbol{\alpha}_j)$ is the parametric (with respect to $\boldsymbol{\alpha}_j$) distribution of $\boldsymbol{X}$. In (1), $\boldsymbol{\vartheta}$ contains all the model parameters. Sufficient conditions for the identifiability of model (1), under Gaussian assumptions for $Y|\boldsymbol{X} = \boldsymbol{x}$ in each mixture components, are given in Hennig (2000); see Ingrassia *et al.* (2015) in the case of $p\left(y|\boldsymbol{x}; \boldsymbol{\beta}_j, \boldsymbol{\gamma}_j\right)$ belonging to the exponential family.

The **flexCWM** package allows for the fitting of CWMs as parameterized in (1) using the expectation-maximization (EM) algorithm. The local expectations of $Y|\boldsymbol{X} = \boldsymbol{x}$ are modeled using generalized linear models, where $\boldsymbol{\beta}_j$ are the regression coefficients and $\boldsymbol{\gamma}_j$ are additional parameters related to the distribution family chosen for $p\left(y|\boldsymbol{x}; \boldsymbol{\beta}_j, \boldsymbol{\gamma}_j\right)$. The linear predictor and the link function are specified using standard R `formula` objects, in the same way used by the well-known `stats::glm()` function, and the following distribution families are available: binomial, Poisson, gamma, inverse Gaussian, normal, and, for robustness' sake, the t distribution. Covariates can be of mixed-type; supported distributions are multivariate normal, binomial, Poisson, and multinomial. Since local independence is assumed, covariates' distributions are joined together via taking the product.

The **flexCWM** package may be used for estimating latent class cluster models with mixed-type variables (see Vermunt and Magidson 2002, pp. 94–95) and mixtures of regression models with fixed covariates, since both models are special cases of (1), obtained respectively by fixing $p\left(y|\boldsymbol{x}; \boldsymbol{\beta}_j, \boldsymbol{\gamma}_j\right) \equiv 1$ and $p(\boldsymbol{x}; \boldsymbol{\alpha}_j) \equiv 1$. Furthermore, the $\boldsymbol{x}$ variables in $p(\boldsymbol{x}; \boldsymbol{\alpha}_j)$ may be partially or completely different from the covariates $\boldsymbol{x}$ in $p\left(y|\boldsymbol{x}; \boldsymbol{\beta}_j, \boldsymbol{\gamma}_j\right)$, so leading to a particular type of mixtures of regression models with concomitant variables (Dayton and Macready 1988); for this reason, the $\boldsymbol{x}$ variables in $p(\boldsymbol{x}; \boldsymbol{\alpha}_j)$ are referred to as concomitant variables in the package.

Normal concomitant variables are modeled using multivariate normal distributions with covariance matrices $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k$; so, for q normal variables, $kq(q+1)/2$ free covariance parameters have to be estimated. For sake of parsimony, based on Punzo and Ingrassia (2015) and following Celeux and Govaert (1995), the **flexCWM** package considers the eigendecomposition

$$\boldsymbol{\Sigma}_j = \lambda_j \boldsymbol{\Gamma}_j \boldsymbol{\Delta}_j \boldsymbol{\Gamma}_j^{\top}, \tag{2}$$

where $\lambda_j = |\boldsymbol{\Sigma}_j|^{1/q}$, $\boldsymbol{\Delta}_j$ is the scaled ($|\boldsymbol{\Delta}_j| = 1$) diagonal matrix of the eigenvalues of $\boldsymbol{\Sigma}_j$ sorted in decreasing order, and $\boldsymbol{\Gamma}_j$ is a $q \times q$ orthogonal matrix whose columns are the normalized eigenvectors of $\boldsymbol{\Sigma}_j$, ordered according to their eigenvalues. Each element in the right-hand side of (2) has a different geometric interpretation of the cluster with respect to the normal concomitant variables: λ_j determines the volume, $\boldsymbol{\Delta}_j$ the shape, and $\boldsymbol{\Gamma}_j$ the orientation.

Still following Celeux and Govaert (1995), the **flexCWM** package allows to impose constraints on the three components of (2) resulting in a family of fourteen parsimonious CWMs, when $q > 1$, and in a family of two CWMs when $q = 1$ (cf. Table 1).

Family	Model	Volume	Shape	Orientation	$\boldsymbol{\Sigma}_j$	# of free parameters in $\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_k$
Univariate	E	Equal	–	–	λ	1
	V	Variable	–	–	λ_j	k
Spherical	EII	Equal	Spherical	–	$\lambda\boldsymbol{I}$	1
	VII	Variable	Spherical	–	$\lambda_j\boldsymbol{I}$	k
Diagonal	EEI	Equal	Equal	Axis-Aligned	$\lambda\boldsymbol{\Delta}$	q
	VEI	Variable	Equal	Axis-Aligned	$\lambda_j\boldsymbol{\Delta}$	$k+q-1$
	EVI	Equal	Variable	Axis-Aligned	$\lambda\boldsymbol{\Delta}_j$	$1+k\left(q-1\right)$
	VVI	Variable	Variable	Axis-Aligned	$\lambda_j\boldsymbol{\Delta}_j$	kq
General	EEE	Equal	Equal	Equal	$\lambda\boldsymbol{\Gamma}\boldsymbol{\Delta}\boldsymbol{\Gamma}^\top$	$q\left(q+1\right)/2$
	VEE	Variable	Equal	Equal	$\lambda_j\boldsymbol{\Gamma}\boldsymbol{\Delta}\boldsymbol{\Gamma}^\top$	$k+q-1+q\left(q-1\right)/2$
	EVE	Equal	Variable	Equal	$\lambda\boldsymbol{\Gamma}\boldsymbol{\Delta}_j\boldsymbol{\Gamma}^\top$	$1+k\left(q-1\right)+q\left(q-1\right)/2$
	EEV	Equal	Equal	Variable	$\lambda\boldsymbol{\Gamma}_j\boldsymbol{\Delta}\boldsymbol{\Gamma}_j^\top$	$q+kq\left(q-1\right)/2$
	VVE	Variable	Variable	Equal	$\lambda_j\boldsymbol{\Gamma}\boldsymbol{\Delta}_j\boldsymbol{\Gamma}^\top$	$kq+q\left(q-1\right)/2$
	VEV	Variable	Equal	Variable	$\lambda_j\boldsymbol{\Gamma}_j\boldsymbol{\Delta}\boldsymbol{\Gamma}_j^\top$	$k+q-1+kq\left(q-1\right)/2$
	EVV	Equal	Variable	Variable	$\lambda\boldsymbol{\Gamma}_j\boldsymbol{\Delta}_j\boldsymbol{\Gamma}_j^\top$	$1+k\left(q-1\right)+kq\left(q-1\right)/2$
	VVV	Variable	Variable	Variable	$\lambda_j\boldsymbol{\Gamma}_j\boldsymbol{\Delta}_j\boldsymbol{\Gamma}_j^\top$	$kq\left(q+1\right)/2$

Table 1: Nomenclature, covariance structure, and number of free parameters in $\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_k$; in the univariate case, λ denotes the variance.

3. Maximum likelihood estimation: The EM algorithm

The **flexCWM** package adopts the EM algorithm of Dempster, Laird, and Rubin (1977) to find maximum likelihood (ML) estimates for the parameters of model (1). This algorithm is described below; further details, and simulation results about the asymptotic properties of the estimators of the local regression coefficients, can be found in Ingrassia *et al.* (2015) and Punzo and Ingrassia (2016).

Given a random sample $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ of $(\boldsymbol{X}, Y)$ from model (1), and once k has been specified, the algorithm basically takes into account the complete-data log-likelihood

$$l_c(\boldsymbol{\vartheta}) = \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij} \ln(\pi_j) + \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij} \ln\left[p\left(y_i|\boldsymbol{x}_i;\boldsymbol{\beta}_j,\boldsymbol{\gamma}_j\right)\right] + \sum_{i=1}^{n}\sum_{j=1}^{k} z_{ij} \ln\left[p\left(\boldsymbol{x}_i;\boldsymbol{\alpha}_j\right)\right], \tag{3}$$

where $z_{ij} = 1$ if $(\boldsymbol{x}_i, y_i)$ comes from component j and $z_{ij} = 0$ otherwise. The EM algorithm iterates between two steps, one E-step and one M-step, until convergence; their schematization, with respect to the CWM in its general version (1), is given below.

E-step: Given the current parameter estimates $\boldsymbol{\vartheta}^{(r)}$ of the rth iteration, simply replace each z_{ij} by the estimated posterior probability

$$z_{ij}^{(r)} = \frac{\pi_j^{(r)} p\left(y_i|\boldsymbol{x}_i;\boldsymbol{\beta}_j^{(r)},\boldsymbol{\gamma}_j^{(r)}\right) p\left(\boldsymbol{x}_i;\boldsymbol{\alpha}_j^{(r)}\right)}{p\left(\boldsymbol{x}_i, y_i;\boldsymbol{\vartheta}^{(r)}\right)}. \tag{4}$$

M-step: The obtained values of $z_{ij}^{(r)}$, which are function of $\boldsymbol{\vartheta}^{(r)}$, are substituted for z_{ij} in (3) leading to the expected complete-data log-likelihood

$$Q\left(\boldsymbol{\vartheta}|\boldsymbol{\vartheta}^{(r)}\right) = Q_1\left(\boldsymbol{\pi}|\boldsymbol{\vartheta}^{(r)}\right) + Q_2\left(\boldsymbol{\beta},\boldsymbol{\gamma}|\boldsymbol{\vartheta}^{(r)}\right) + Q_3\left(\boldsymbol{\alpha}|\boldsymbol{\vartheta}^{(r)}\right), \tag{5}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)$, and $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_k)$; the three terms on the right-hand side of (5) are ordered as the three terms on the right-hand side of (3). As the three terms on the right-hand side of (5) have zero cross-derivatives, they can be maximized separately.

The maximization of Q_1, with respect to $\boldsymbol{\pi}$ and subject to the constraints on these parameters, yields

$$\pi_j^{(r+1)} = \sum_{i=1}^{n} z_{ij}^{(r)} \Big/ n, \quad j = 1, \ldots, k. \tag{6}$$

The maximization of Q_2, with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, is equivalent to independently maximizing each of the k expressions

$$Q_{2j}\left(\boldsymbol{\beta}_j,\boldsymbol{\gamma}_j|\boldsymbol{\vartheta}^{(r)}\right) = \sum_{i=1}^{n} z_{ij}^{(r)} \ln\left[p\left(y_i|\boldsymbol{x}_i;\boldsymbol{\beta}_j,\boldsymbol{\gamma}_j\right)\right] \tag{7}$$

with respect to $\boldsymbol{\beta}_j$ and $\boldsymbol{\gamma}_j$, $j = 1, \ldots, k$. When $p\left(y|\boldsymbol{x};\boldsymbol{\beta}_j,\boldsymbol{\gamma}_j\right)$ is assumed to belong to the exponential family, the maximization of (7) is equivalent to the maximization problem of the generalized linear model (for the complete data), except that each observation $(\boldsymbol{x}_i, y_i)$ contributes to the log-likelihood with a known weight $z_{ij}^{(r)}$; see Wedel and DeSarbo (1995) and Wedel and Kamakura (2000, pp. 120–124) for details.

If the parsimonious variants of model (1) are not applied, analogously to Q_2, the maximization of Q_3, with respect to $\boldsymbol{\alpha}$, is equivalent to independently maximizing each of the k expressions

$$Q_{3j}\left(\boldsymbol{\alpha}_j|\boldsymbol{\vartheta}^{(r)}\right) = \sum_{i=1}^{n} z_{ij}^{(r)} \ln\left[p\left(\boldsymbol{x}_i; \boldsymbol{\alpha}_j\right)\right] \tag{8}$$

with respect to $\boldsymbol{\alpha}_j$, $j = 1, \ldots, k$. These updates can be obtained using weighted ML estimation. When the parsimonious variants of the CWM are used, the M-step changes only for the way the terms of the decomposition of $\boldsymbol{\Sigma}_j$ are obtained. These updates are analogous to those given by Celeux and Govaert (1995) for mixtures of multivariate Gaussian distributions with eigendecomposed covariance matrices and, within the R platform, have been implemented in the `m.step()` function of the **ContaminatedMixt** package (Punzo, Mazza, and McNicholas 2018), on which package **flexCWM** depends. Package **ContaminatedMixt** allows the fit of parsimonious mixtures of multivariate contaminated normal distributions (Punzo and McNicholas 2016).

4. Some computational and operational aspects

4.1. Initialization

Many initialization strategies have been proposed for the EM algorithm applied to mixture models (see, e.g., Biernacki, Celeux, and Govaert 2003, Karlis and Xekalaki 2003, and Bagnato and Punzo 2013). The **flexCWM** package implements the following initializations, all based on providing the initial quantities $\boldsymbol{z}_i^{(0)} = \left(z_{i1}^{(0)}, \ldots, z_{ik}^{(0)}\right)$, $i = 1, \ldots, n$, to the first E-step of the algorithm.

`"random.soft"`: The k values in $\boldsymbol{z}_i^{(0)}$ are randomly generated by a uniform distribution – via the `runif()` function of the **stats** package – and then normalized in order to sum to 1.

`"random.hard"`: Each $\boldsymbol{z}_i^{(0)}$ is substituted by a single observation randomly generated – via the `rmultinom()` function of the **stats** package – from a multinomial distribution with probabilities $(1/k, \ldots, 1/k)$.

`"manual"`: The (soft or hard) values of $\boldsymbol{z}_i^{(0)}$ are provided by the user.

Furthermore, random initializations can be repeated from different random positions; in this case, the solution maximizing the observed-data log-likelihood among these runs is selected.

For `numeric` variables only, two additional initialization strategies have been made available:

`"kmeans"`: Hard values for $\boldsymbol{z}_i^{(0)}$, $i = 1, \ldots, n$, are provided by a preliminary run of the k-means algorithm, as implemented by the `kmeans()` function of the **stats** package.

`"mclust"`: Soft values for $\boldsymbol{z}_i^{(0)}$, $i = 1, \ldots, n$, are computed by a preliminary fit of an unconstrained mixture of Gaussian distributions as implemented by the `Mclust()` function of the **mclust** package (Fraley, Raftery, Scrucca, Murphy, and Fop 2017).

4.2. Convergence criterion

The Aitken acceleration (Aitken 1926) is used to estimate the asymptotic maximum of the log-likelihood at each iteration of the EM algorithm. Based on this estimate, we can decide whether or not the algorithm has reached convergence, i.e., whether or not the log-likelihood is sufficiently close to its estimated asymptotic value. The Aitken acceleration at iteration $r + 1$ is given by

$$a^{(r+1)} = \frac{l^{(r+2)} - l^{(r+1)}}{l^{(r+1)} - l^{(r)}},$$

where $l^{(r+2)}$, $l^{(r+1)}$, and $l^{(r)}$ are the observed-data log-likelihood values from iterations $r + 2$, $r + 1$, and r, respectively. Then, the asymptotic estimate of the log-likelihood at iteration $r + 2$ is given by

$$l_{\infty}^{(r+2)} = l^{(r+1)} + \frac{1}{1 - a^{(r+1)}} \left(l^{(r+2)} - l^{(r+1)} \right)$$

(cf. Böhning, Dietz, Schaub, Schlattmann, and Lindsay 1994). The EM algorithm can be considered to have converged when $l_{\infty}^{(r+2)} - l^{(r+1)} < \epsilon$ (see Lindsay 1995).

4.3. Model selection

Thus far, the number of components k and/or the covariance structure of the normal distributed concomitant variables (cf. Table 1) have been treated as *a priori* fixed. However, in most practical applications, they are unknown, so it is common practice to select them by evaluating a convenient (likelihood-based) model selection criterion over a reasonable range of possible options. The **flexCWM** package supports the information criteria listed in Table 2, where $l\left(\widehat{\boldsymbol{\vartheta}}\right)$ is the observed-data log-likelihood, m is the number of free parameters, and $\mathrm{MAP}\left(\widehat{z}_{ij}\right)$ is the *maximum a posteriori* probability operator – assuming value 1 if $\max_{h=1,\ldots,k}\left\{\widehat{z}_{ih}\right\}$ occurs at component j and 0 otherwise – and $\widehat{\boldsymbol{\vartheta}}$ and $\widehat{z}_{ij}$ are the estimates for $\boldsymbol{\vartheta}$ and z_{ij} obtained with the EM algorithm. Simulation studies aimed at comparing the behavior of the criteria in Table 2 for some particular variants of the CWM are provided in Ingrassia *et al.* (2015) and Punzo and Ingrassia (2016). For an alternative approach, based on a testing procedure, to select the best parsimonious model between those of the "general" family in Table 1, see Punzo, Browne, and McNicholas (2016).

5. Package description and illustrative examples

In this section we provide a description of the main capabilities of the **flexCWM** package, along with illustrations via real and artificial datasets contained in the package.

5.1. Package description

The **flexCWM** package is developed in an object-oriented design, using the standard S3 paradigm. Its main function, `cwm()`, fits the CWM(s) and returns an object of class '`cwm`'; the

Criterion	Definition	Reference
AIC	$2l\left(\widehat{\boldsymbol{\vartheta}}\right) - 2m$	Akaike (1973)
AIC$_3$	$2l\left(\widehat{\boldsymbol{\vartheta}}\right) - 3m$	Bozdogan (1994)
AICc	$\text{AIC} - 2\dfrac{m(m+1)}{n-m-1}$	Hurvich and Tsai (1989)
AICu	$\text{AICc} - n\ln\dfrac{n}{n-m-1}$	McQuarrie, Shumway, and Tsai (1997)
AWE	$2l\left(\widehat{\boldsymbol{\vartheta}}\right) - 2m\left(\dfrac{3}{2} + \ln n\right)$	Banfield and Raftery (1993)
BIC	$2l\left(\widehat{\boldsymbol{\vartheta}}\right) - m\ln n$	Schwarz (1978)
CAIC	$2l\left(\widehat{\boldsymbol{\vartheta}}\right) - m(1+\ln n)$	Bozdogan (1987)
ICL	$\text{BIC} + 2\sum_{i=1}^{n}\sum_{j=1}^{k}\text{MAP}\left(\widehat{z}_{ij}\right)\ln\widehat{z}_{ij}$	Biernacki, Celeux, and Govaert (2000)

Table 2: Definition and key reference for the implemented model selection criteria.

arguments of this function, along with their description, are listed in Table 3. In addition, the package contains several methods that allow for data extraction and visualization. Extractors for '`cwm`' class objects are illustrated in Table 4.

When several models have been fitted, extractor functions consider the best model according to the information criterion in `criterion`, within the subset of estimated models having a number of components among those in `k`, a conditional distribution for Y among those in `familyY`, and a parsimonious model among those in `modelXnorm`. Note that `getIC()` and `whichBest()` have an argument `criteria`, in substitution to `criterion`, which allows to select more than one criterion.

Finally, the package includes some methods for '`cwm`' class objects: `plot()`, to display classification results in terms of both scatter plots and univariate distribution plots, `summary()`, to visualize the estimated parameters and further inferential details, and `print()`, to print the selected model(s) according to the information criteria in Table 2. As usual, further details can be found in the functions' help pages.

5.2. How to fit the different types of mixture models

In the following, we will show how to fit several types of mixture models available in the literature. Table 5 schematizes how to fit mixtures of univariate distributions, assuming that the data are in a vector named `Y`. As we can see from the table, there are two ways for fitting binomial, Poisson, and (unconstrained) Gaussian mixtures. Furthermore, the **flexCMM** package allows to fit:

- All the fourteen parsimonious mixtures of multivariate Gaussian distributions of Celeux and Govaert (1995). They are obtained by setting `modelXnorm` to the parsimonious model(s) to be fitted and by passing the data available using `Xnorm`.
- Latent class cluster models for mixed-type variables (see Vermunt and Magidson 2002, pp. 94–95). They are obtained by passing variables via `Xnorm`, `Xbin`, `Xpois` and `Xmult`

Arguments	Description
`formulaY`	An optional object of class '`formula`'. A symbolic description of the model to be fitted.
`familyY`	An optional object of class '`family`'. A description of the conditional distribution(s) of Y in each mixture component. It can be: `gaussian`, `poisson`, `binomial`, `Gamma`, `inverse.gaussian` (use `?stats::family` for details about default/possible links), and `student.t` with default `link = identity`. Default value is `gaussian`.
`data`	An optional `data.frame`, `list`, or `environment` with the variables needed to evaluate `formulaY`.
`Xnorm, Xbin, Xpois, Xmult`	Optional matrices containing concomitant variables having normal, binomial, Poisson, and multinomial distributions, respectively.
`modelXnorm`	An optional vector of character strings indicating the parsimonious models to be fitted (cf. Table 1). The default is `c("E", "V")` when `ncol(Xnorm) == 1`, and `c("EII", "VII", "EEI", "VEI", "EVI", "VVI", "EEE", "VEE", "EVE", "EEV", "VVE", "VEV", "EVV", "VVV")` when `ncol(Xnorm) > 1`.
`Xbtrials`	An optional vector containing the number of trials for each column in `Xbin`. If omitted, the maximum of each column in `Xbin` is chosen.
`k`	A vector containing the values of k to be fitted.
`initialization`	Initialization strategy for the EM algorithm. It can be: `"random.soft"` (default), `"random.hard"`, `"manual"`, `"kmeans"`, and `"mclust"` (see Section 4.1 for details).
`start.z`	$(n \times k)$ matrix of soft or hard classification; it is used only if `initialization = "manual"`.
`seed`	Seed for the random number generator, when random initializations are used; if `NULL` (default), the current seed is not changed.
`maxR`	Number of random EM initializations. Default value is `1`.
`iter.max`	Maximum number of EM iterations. Default value is `200`.
`threshold`	Value of ϵ in the Aitken acceleration procedure (see Section 4.2). Default value is `1.0e-04`.
`eps`	Smallest value for the eigenvalues of $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_k$. It used to prevent the EM algorithm to be affected by local maxima or degeneracy of the likelihood (see Hathaway 1986 and Ingrassia 2004). Default value is `1e-100`.
`parallel`	When `TRUE`, the package **parallel** is used for parallel computation. The number of cores to use may be set with the global option `cl.cores`; default value is detected using `detectCores()`.

Table 3: List of arguments for the function `cwm()`.

Extractors	Description
`getBestModel()`	A '`cwm`' class object containing the best model only.
`getCluster()`	Classification vector.
`getIC()`	Values for the considered criteria in `criteria`.
`getPar()`	Estimated parameters (i.e., $\widehat{\boldsymbol{\vartheta}}$).
`getParPrior()`	Estimated mixture weights (i.e., $\widehat{\boldsymbol{\pi}}$).
`getParGLM()`	Estimated regression coefficients (i.e., $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\gamma}}$).
`getParConcomitant()`	Estimated parameters of the concomitant variables (i.e., $\widehat{\boldsymbol{\alpha}}$). It has an additional optional argument, `name`, which allows to specify the names of distribution families (`"multinomial"`, `"poisson"`, `"normal"`, `"binomial"`) for the concomitant variables; if `NULL` (default) parameters estimated for all concomitant variables are returned.
`getParXnorm()`	Estimated parameters for `Xnorm`.
`getParXbin()`	Estimated parameters for `Xbin`.
`getParXpois()`	Estimated parameters for `Xpois`.
`getParXmult()`	Estimated parameters for `Xmult`.
`getPosterior()`	Estimated posterior probabilities (i.e., $\widehat{z}_{ij}$, $i = 1, \dots, n$ and $j = 1, \dots, k$).
`getSize()`	Estimated group sizes (from the hard classification induced by the MAP operator).
`whichBest()`	Position of the model, in the '`cwm`' class object, for the criteria specified in `criteria`.

Table 4: Extractors for '`cwm`' class objects.

Mixture component	`formulaY`	`familyY`	`Xnorm`	`Xbin`	`Xpois`	`Xmult`	`modelXnorm`
Binomial	`Y ~ 1`	`binomial`	`NULL`	`NULL`	`NULL`	`NULL`	`NULL`
	`NULL`	`NULL`	`NULL`	`Y`	`NULL`	`NULL`	`NULL`
Poisson	`Y ~ 1`	`poisson`	`NULL`	`NULL`	`NULL`	`NULL`	`NULL`
	`NULL`	`NULL`	`NULL`	`NULL`	`Y`	`NULL`	`NULL`
Gaussian (V)	`Y ~ 1`	`gaussian`	`NULL`	`NULL`	`NULL`	`NULL`	`"V"`
	`NULL`	`NULL`	`Y`	`NULL`	`NULL`	`NULL`	`"V"`
Gaussian (E)	`NULL`	`NULL`	`Y`	`NULL`	`NULL`	`NULL`	`"E"`
Multinomial	`NULL`	`NULL`	`NULL`	`NULL`	`NULL`	`Y`	`NULL`
Gamma	`Y ~ 1`	`Gamma`	`NULL`	`NULL`	`NULL`	`NULL`	`NULL`
Inverse Gaussian	`Y ~ 1`	`inverse.gaussian`	`NULL`	`NULL`	`NULL`	`NULL`	`NULL`
Student-t	`Y ~ 1`	`student.t`	`NULL`	`NULL`	`NULL`	`NULL`	`NULL`

Table 5: How to fit univariate mixtures with different component distributions. The vector of data is named `Y`.

according to their distribution. `modelXnorm` can be used to specify the parsimonious model(s) to be applied to `Xnorm` variables. Furthermore, one more variable, say `Y`, may be passed by setting `formulaY = Y ~ 1` and specifying its distribution using argument `familyY`; this procedure allows for modeling a variable having Gamma, inverse Gaussian or Student's *t* distribution. Note that `Y` has to be in the current environment or in a dataset passed through argument `data`.

- Mixtures of regression models with fixed covariates are obtained by passing a symbolic description of the component regression model via the argument `formulaY`, and specifying its conditional distribution and link function using argument `familyY`.
- The CWM in (1) is obtained by following the same scheme given for mixtures of regression models with fixed covariates, but with the difference of attributing each covariate to one, and only one, of the four arguments `Xnorm`, `Xbin`, `Xpois`, and `Xmult`.

5.3. The `students` dataset

The first tutorial uses the `students` dataset included in the **flexCWM** package. These data, introduced by Ingrassia *et al.* (2014), come from a survey of 270 students attending a Statistics course at the Department of Economics and Business of the University of Catania in the academic year 2011/2012 and consist of a response variable `WEIGHT` (weight of the respondent, measured in kilograms) and two covariates, `HEIGHT` (height of the respondent, measured in centimeters) and `HEIGHT.F` (height of respondent's father, measured in centimeters). Data are labeled with respect to the `GENDER` of the respondent. In the following we assume that data were unlabeled: the purpose of this tutorial is to see if the methods implemented in the **flexCWM** package are able to correctly identify the number of latent groups and distinguish between male and female subjects.

To begin the analysis, the data are loaded by

```
R> set.seed(1234)
R> library("flexCWM")
R> data("students", package = "flexCWM")
```

The command `data("students", package = "flexCWM")` loads a data frame with four variables, as we can see in the structure of the object.

```
R> str(students)

'data.frame':   270 obs. of  4 variables:
 $ GENDER  : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
 $ HEIGHT  : int  184 186 186 179 175 181 187 174 179 175 ...
 $ WEIGHT  : int  82 83 88 74 68 79 80 85 81 78 ...
 $ HEIGHT.F: int  185 180 182 177 175 176 179 172 182 174 ...
```

In order to make the next commands shorter and more readable, we now attach the student data frame to the R search path via the command

```
R> attach(students)
```

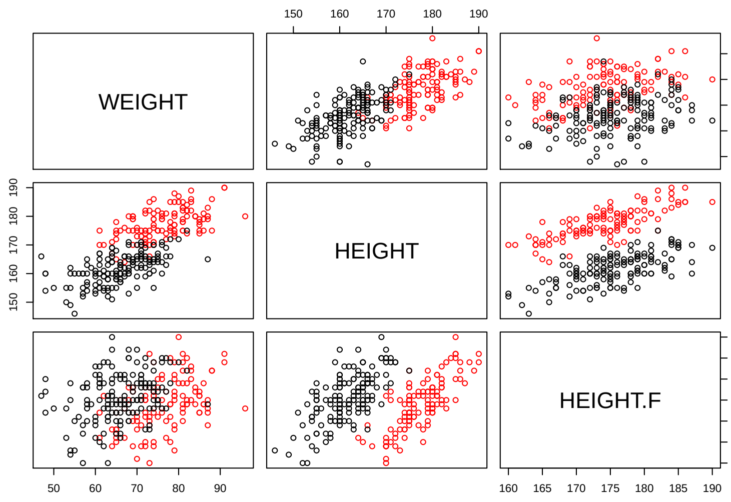


Figure 1: Dataset `students`: Matrix of scatter plots for the variables `WEIGHT`, `HEIGHT`, and `HEIGHT.F`, with colors induced by `GENDER`.

A first look at the pairwise relation between these variables can be given through the matrix of scatter plots shown in Figure 1; it is obtained via the command

```
R> pairs(cbind(WEIGHT, HEIGHT, HEIGHT.F), col = GENDER)
```

First of all, we will consider classical mixtures of linear (Gaussian) regression models with fixed covariates; however, to deal with either longer than normal tails or atypical observations, we will also consider the Student's t distribution (Yao, Wei, and Yu 2014). The following command

```
R> fit1 <- cwm(WEIGHT ~ HEIGHT + HEIGHT.F,
+    familyY = list(gaussian, student.t), k = 1:3,
+    initialization = "kmeans")

[...]

Best model according to AIC, AICc, AICu, AIC3, AWE, BIC, CAIC, ICL is
obtained with k = 1 group(s) and family gaussian(identity)
```

performs the EM-fitting of six models, because of `familyY = list(gaussian, student.t)` and `k = 1:3`, and returns an object of class '`cwm`'. Argument `initialization = "kmeans"` is used to indicate the initialization strategy to adopt for the EM algorithm (cf. Section 4.1). Because several models have to be fitted, parallel computation is convenient; it is set with the argument `parallel = TRUE`. The number of CPU cores used is printed and this is followed, after a few seconds, by a description of the best model according to each of the 8 criteria in Table 2. Here, we can note that all the criteria agree in suggesting a model with $k = 1$. From a clustering point of view, this is not a good result, since a clear group structure is visible in the scatter plot of `HEIGHT` versus `HEIGHT.F` in Figure 1. Nonetheless, because the fitted models assume fixed covariates (assignment independence), they do not identify the correct cluster structure.

In order to take the covariates distributions into account for clustering, we will now fit the CWM versions of the models previously fitted; this is done via the command

```
R> fit2 <- cwm(WEIGHT ~ HEIGHT + HEIGHT.F,
+    familyY = list(gaussian, student.t), Xnorm = cbind(HEIGHT, HEIGHT.F),
+    k = 1:3, initialization = "kmeans")
```

```
With k=1, some models in modelXnorm are equivalent, so only the first is
estimated.

[...]

Best model according to AIC, AICc is obtained with k = 3 group(s) and
parsimonious model EVV and family gaussian(identity)

Best model according to AICu, AIC3, ICL is obtained with k = 2 group(s) and
parsimonious model VVE and family gaussian(identity)

Best model according to AWE, BIC, CAIC is obtained with k = 2 group(s) and
parsimonious model EEE and family gaussian(identity)
```

The argument `Xnorm = cbind(HEIGHT, HEIGHT.F)` specifies the multivariate normal concomitant variables whose marginal distribution is considered for the definition of the CWM. Since we did not specify any value for the argument `modelXnorm`, by default all the fourteen parsimonious variants of the CWM are fitted (cf. Table 1). As we can see from the results printed, all the criteria suggest at least 2 mixture components. In particular the BIC, which is the most widely adopted criterion in the literature on mixture models, suggests an EEE model with two mixture components and a Gaussian distribution for `WEIGHT` | (`HEIGHT`, `HEIGHT.F`). To find out more about the model selected by the BIC, we run the command

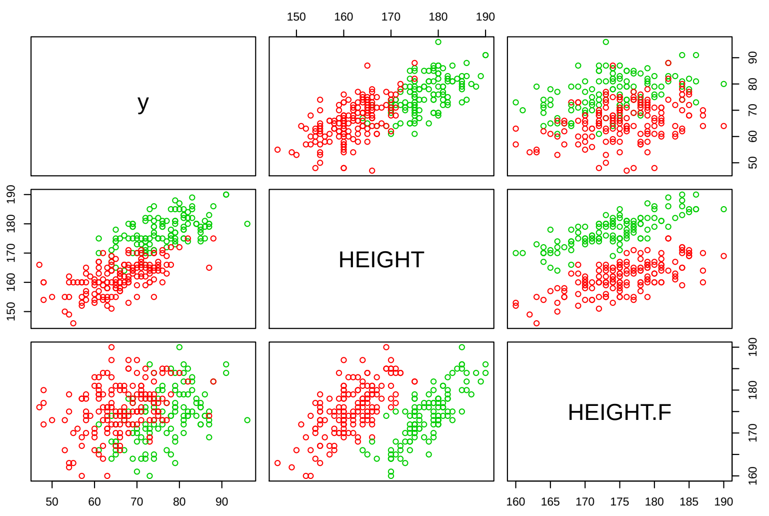
```
R> summary(fit2, criterion = "BIC", concomitant = TRUE)
```

```
----------------------------------
Best fitted model according to BIC
----------------------------------
 loglikelihood   n df     BIC
       -2638.7 270 16 -5367.1

Clustering table:
  1   2
153 117

Prior: comp.1 = 0.56295, comp.2 = 0.43705

Distribution used for GLM: gaussian(identity). Parameters:

Component 1
```

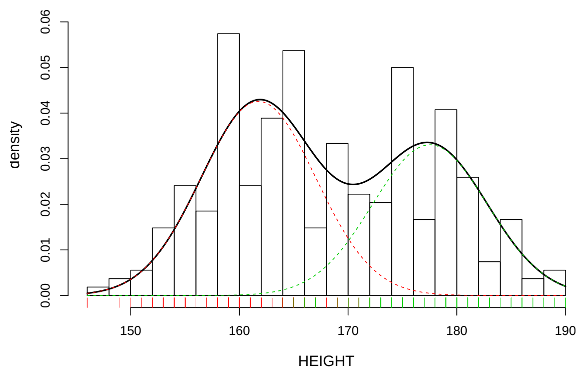
```
              Estimate Std. Error t value  Pr(>|t|)
(Intercept) -54.027136  12.126540 -4.4553 1.233e-05 ***
HEIGHT        0.897714   0.091307  9.8318 < 2.2e-16 ***
HEIGHT.F     -0.144006   0.083820 -1.7180   0.08695 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
sigma = 5.951

Component 2
               Estimate  Std. Error t value  Pr(>|t|)
(Intercept) -57.2474456  12.3740383 -4.6264 5.807e-06 ***
HEIGHT        0.7599275   0.1081866  7.0242 1.787e-11 ***
HEIGHT.F     -0.0077206   0.0939206 -0.0822    0.9345
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
sigma = 5.889

Model for normal concomitant variables: EEE (ellipsoidal, equal volume,
shape and orientation) with 2 components

Normal concomitant variables
  Means:
      group 1 group 2
X.1  161.75  177.54
X.2  175.60  174.14

  Variance-covariance matrices:
  Component 1
       X.1    X.2
X.1 27.835 22.038
X.2 22.038 34.698
  Component 2
       X.1    X.2
X.1 27.835 22.038
X.2 22.038 34.698
```

The argument `concomitant = TRUE` allows to print the estimated parameters for the concomitant variables (`Xnorm` only in this example). Note that the p values are only a raw approximation; this is because standard errors are not computed on the observed-data likelihood of model (1), but on the expression in (7) and assuming the posterior probabilities $\widehat{z}_{ij}$ as given, instead of as estimated. Furthermore, the p values are not corrected for the fact that the data are already used to determine the specific fitted model (see Grün and Leisch 2008 and Harrell 2001 for further details about this problem). So, their intended use is in an exploratory manner only. We can take a graphical glance at the classification results for the best BIC model via the command

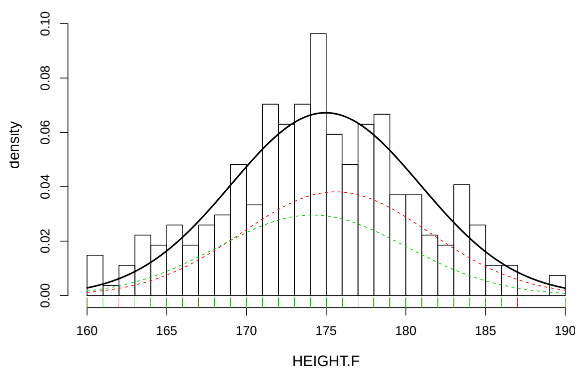
```
R> plot(fit2)
```



(a) Matrix of scatter plots with colors induced by the fitted classification.



(b) Marginal (in black) and weighted component (in green and red) fitted densities of `HEIGHT`.



(c) Marginal (in black) and weighted component (in green and red) fitted densities of `HEIGHT.F`.

Figure 2: Dataset `students`: Classification obtained by the EEE linear Gaussian CWM with $k = 2$ components.

which produces the plots in Figure 2. The marginal densities in Figure 2(b) and Figure 2(c) show the covariates' contribution to the definition of the two groups (assignment dependence), with `HEIGHT` contributing more than `HEIGHT.F`. This assignment dependence is the cause of the differences between the results obtained with this model and the ones obtained with the model with fixed covariates (`fit1`). The comparison of Figure 1 and Figure 2 discloses a good agreement between the classification obtained with that induced by `GENDER`. This agreement is also corroborated, more precisely, by the confusion matrix.

```
R> fit2clust <- getCluster(fit2)
R> table(fit2clust, GENDER)
```

```
         GENDER
fit2clust   F   M
        1 149   4
        2   2 115
```

The function `getCluster()` extracts the classification vector of the fitted model that is shown in Table 4.

5.4. The `ExCWM` dataset

In this second tutorial, we will use artificial data generated by a CWM with $k = 2$ components, of sizes $n_1 = 70$ and $n_2 = 130$, four covariates of mixed data types and with various distributions (normal, binomial and Poisson), and one binomial response variable. Our purpose is to explore the capabilities of the **flexCWM** package in fitting data of mixed types and detecting the true number of latent groups. First we will fit mixtures of distributions of variables of the same data type. Then, we will fit mixtures of distributions of variables of mixed data types. At the end, we will fit a full CWM, that takes into account the dependence between the response variable and the covariates.

The data are in a data frame named `ExCWM`, which is included in the package. The following commands load the data frame, attach it to the R search path, and display its structure.

```
R> set.seed(17)
R> data("ExCWM", package = "flexCWM")
R> attach(ExCWM)
R> str(ExCWM)
```

```
'data.frame':   200 obs. of  8 variables:
 $ Y       : int  19 20 14 11 5 17 6 20 19 20 ...
 $ Xnorm1  : num  -1.074 -0.177 -3.611 -2.285 -2.342 ...
 $ Xnorm2  : num  -2.92 -3.08 -1.65 -2.96 -2.17 ...
 $ Xbin    : int  3 5 4 3 2 3 3 5 4 3 ...
 $ Xpois   : int  1 0 0 0 1 1 0 2 0 2 ...
 $ group   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Ybtrials: int  20 20 20 20 20 20 20 20 20 20 ...
 $ Xbtrials: int  15 15 15 15 15 15 15 15 15 15 ...
```

In the following, a brief description of the 8 variables in the data frame is given. It will allow the user to interpret correctly the results of the clustering that will be shown later.

`Xnorm1`, `Xnorm2`: The first $n_1 = 70$ observations are simultaneously generated by a bivariate Gaussian distribution with mean and covariance matrix given by

$$\boldsymbol{\mu}_1 = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_1 = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix},$$

respectively, while the remaining $n_2 = 130$ observations are generated by a bivariate Gaussian distribution with mean $\mu_2 = (2, 2)^\top$ and covariance matrix $\boldsymbol{\Sigma}_2 = 2\boldsymbol{\Sigma}_1$; this corresponds to a VEE parsimonious model.

`Xbin`: It is generated by a binomial distribution, with probability 0.2 for the first $n_1 = 70$ observations and with probability 0.8 for the remaining observations. The number of trials (`Xbtrials`) is 15.

`Xpois`: It is generated by a Poisson distribution, with mean $\lambda_1 = 1$ for the first $n_1 = 70$ observations and with mean $\lambda_2 = 5$ for the remaining observations.

`Y`: It is generated by a binomial distribution, with conditional mean equal to

$$\texttt{Ybtrials}\frac{\exp\left\{1.5\texttt{Xnorm1} + \texttt{Xnorm2} + 2\texttt{Xbin} + \texttt{Xpois}\right\}}{1 + \exp\left\{1.5\texttt{Xnorm1} + \texttt{Xnorm2} + 2\texttt{Xbin} + \texttt{Xpois}\right\}},$$

for the first $n_1 = 70$ observations, and with conditional mean equal to

$$\texttt{Ybtrials}\frac{\exp\left\{-20 + 0.5\texttt{Xnorm1} + 2\texttt{Xnorm2} + 0.5\texttt{Xbin} + 1.7\texttt{Xpois}\right\}}{1 + \exp\left\{-20 + 0.5\texttt{Xnorm1} + 2\texttt{Xnorm2} + 0.5\texttt{Xbin} + 1.7\texttt{Xpois}\right\}},$$

for the remaining observations. The number of trials is `Ybtrials = 20`.

`group`: It contains the true classification.

Parsimonious mixtures of multivariate normal distributions

We start by fitting parsimonious mixtures of bivariate normal distributions to (`Xnorm1`, `Xnorm2`) via the command

```
R> resXnorm <- cwm(Xnorm = cbind(Xnorm1, Xnorm2), k = 1:3,
+    initialization = "kmeans")
```

```
With k=1, some models in modelXnorm are equivalent, so only the first is
estimated.

[...]

Best model according to AIC, AICc is obtained with k = 3 group(s) and
parsimonious model VEE

Best model according to AICu, AIC3, AWE, BIC, CAIC, ICL is obtained with
k = 2 group(s) and parsimonious model VEE
```

Apart from AIC and AICc, all the other information criteria select the true underlying VEE model with two components. Parameters estimated for normal covariates, relative to the best model according to `criterion` (`"BIC"` is the default), may be retrieved using the command

```
R> getParXnorm(resXnorm)

$normal.mu
       group 1  group 2
X.1 -2.089624 1.891098
X.2 -2.001844 1.958787

$normal.Sigma
, , group 1
```

```
           X.1        X.2
X.1  1.0063575 -0.4627117
X.2 -0.4627117  0.8355841

, , group 2

          X.1       X.2
X.1  2.335237 -1.073715
X.2 -1.073715  1.938960
```

Mixtures of binomial distributions

We now fit mixtures of binomial distributions to `Xbin`, via the command

```
R> resXbin <- cwm(Xbin = Xbin, k = 1:3, initialization = "kmeans")

Estimating model with k = 1 *
Estimating model with k = 2 ***
Estimating model with k = 3 *************************************************
*****************************************

Best model according to AIC, AICc, AICu, AIC3, AWE, BIC, CAIC, ICL is
obtained with k = 2 group(s)
```

When `parallel` is set to `FALSE`, which is the default, details on which model is being estimated are displayed during computation, and one star is printed at each iteration of the EM algorithm. All criteria succeeded in detecting the true number of components. Parameters for the binomial components can be retrieved using

```
R> getParXbin(resXbin)

$binomial.p
         comp.1    comp.2
Xbin1 0.1962726 0.7774974
```

Mixtures of Poisson distributions

We now fit mixtures of Poisson distributions to `Xpois`, via the command

```
R> resXpois <- cwm(Xpois = Xpois, k = 1:3, initialization = "kmeans")

Estimating model with k = 1 *
Estimating model with k = 2 *************************
Estimating model with k = 3 ********************************************

Best model according to AIC, AICc, AICu, AIC3, AWE, BIC, CAIC, ICL is
obtained with k = 2 group(s)
```

As before, all the criteria detected the true number of components. Parameters for the Poisson components can be retrieved using

```
R> getParXpois(resXpois)

$poisson.lambda
         comp.1   comp.2
Xpois1 1.107827 5.326135
```

Latent class cluster models for mixed-type variables

We now fit latent class cluster models, for mixed-type variables, to the four variables in (`Xnorm1`, `Xnorm2`, `Xbin`, `Xpois`)

```
R> resX <- cwm(Xnorm = cbind(Xnorm1, Xnorm2), Xbin = Xbin, Xpois = Xpois,
+    k = 1:3, initialization = "kmeans")

With k=1, some models in modelXnorm are equivalent, so only the first is
estimated.

[...]

Best model according to AIC, AICc is obtained with k = 3 group(s) and
parsimonious model VEE

Best model according to AICu, AIC3, AWE, BIC, CAIC, ICL is obtained with
k = 2 group(s) and parsimonious model VEE
```

According to the BIC, the best model has two components and a VEE covariance structure for (`Xnorm1`, `Xnorm2`). All the parameters for the component distributions can be retrieved via the command

```
R> getParConcomitant(resX)

$normal.mu
     group 1   group 2
X.1 1.891910 -2.088991
X.2 1.959559 -2.001148

$normal.Sigma
, , group 1

          X.1       X.2
X.1  2.331590 -1.073407
X.2 -1.073407  1.936516
```

```
, , group 2

           X.1        X.2
X.1  1.0079350 -0.4640285
X.2 -0.4640285  0.8371464


$binomial.p
         comp.1    comp.2
Xbin1 0.7784746 0.2000168

$poisson.lambda
         comp.1    comp.2
Xpois1 4.861538 0.9571244
```

Note that, although independence between (`Xnorm1`, `Xnorm2`), `Xbin`, and `Xpois` is assumed, estimates of component distributions' parameters are different from those obtained previously; this is due to the effect of the posterior probabilities in the E-step of the EM algorithm.

CWMs

Finally, we fit CWMs, with a `Y` response and (`Xnorm1`, `Xnorm2`, `Xbin`, `Xpois`) as covariates/concomitant variables. This is done via the command

```
R> res <- cwm(cbind(Y, Ybtrials - Y) ~ Xnorm1 + Xnorm2 + Xbin + Xpois,
+    data = ExCWM, familyY = binomial, k = 1:3, initialization = "kmeans",
+    Xnorm = cbind(Xnorm1, Xnorm2), Xbin = Xbin, Xpois = Xpois,
+    Xbtrials = Xbtrials[1])

With k=1, some models in modelXnorm are equivalent, so only the first is
estimated.

[...]

Best model according to AIC, AICc, AICu, AIC3, AWE, BIC, CAIC, ICL is
obtained with k = 2 group(s) and parsimonious model VEE and family
binomial(logit)
```

When `familyY = binomial`, the response variable in `formulaY` has to be a matrix with two columns, where the first column is the number of "successes" and the second column is the number of "failures"; in the previous command, this matrix has been built using `cbind(Y, Ybtrials - Y)`. Note that within `stats::glm()`, when the response is the proportion of successes, argument `weights` is used to give the number of trials; this way of providing binomial data is not supported by `cwm()`. According to all criteria, the best model is VEE, with two components. Summary results about the estimated parameters can be obtained via the command

```
R> summary(res, concomitant = TRUE)
```

```
----------------------------------
Best fitted model according to BIC
----------------------------------
 loglikelihood   n df     BIC
       -1746.6 200 23 -3615.1

Clustering table:
  1   2
130  70

Prior: comp.1 = 0.65, comp.2 = 0.35

Distribution used for GLM: binomial(logit). Parameters:

Component 1
              Estimate Std. Error  z value  Pr(>|z|)
(Intercept) -21.031795   1.097057 -19.1711 < 2.2e-16 ***
Xnorm1        0.472029   0.064847   7.2791  3.36e-13 ***
Xnorm2        2.055985   0.113973  18.0392 < 2.2e-16 ***
Xbin          0.589793   0.050347  11.7145 < 2.2e-16 ***
Xpois         1.684478   0.078374  21.4928 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Component 2
            Estimate Std. Error z value  Pr(>|z|)
(Intercept)  0.21206    0.39764  0.5333    0.5938
Xnorm1       1.61273    0.12609 12.7900 < 2.2e-16 ***
Xnorm2       1.07458    0.13428  8.0027 1.217e-15 ***
Xbin         2.03959    0.12072 16.8949 < 2.2e-16 ***
Xpois        0.94386    0.11489  8.2153 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Model for normal concomitant variables: VEE (ellipsoidal, equal shape and
orientation) with 2 components

Normal concomitant variables
  Means:
      group 1 group 2
X.1  1.8919 -2.0890
X.2  1.9596 -2.0011

  Variance-covariance matrices:
  Component 1
        X.1     X.2
X.1  2.3316 -1.0734
```

```
X.2 -1.0734  1.9365
  Component 2
         X.1      X.2
X.1  1.00794 -0.46403
X.2 -0.46403  0.83715

Poisson concomitant variables: lambda parameter
       comp.1  comp.2
Xpois1 4.8615 0.95712

Binomial concomitant variables: p parameter
       comp.1  comp.2
Xbin1 0.77847 0.20002
```

To take a graphical look at the obtained results in terms of classification and density estimation, we can run the command

```
R> plot(res)
```

which produces the plots in Figure 3.

5.5. Computational times

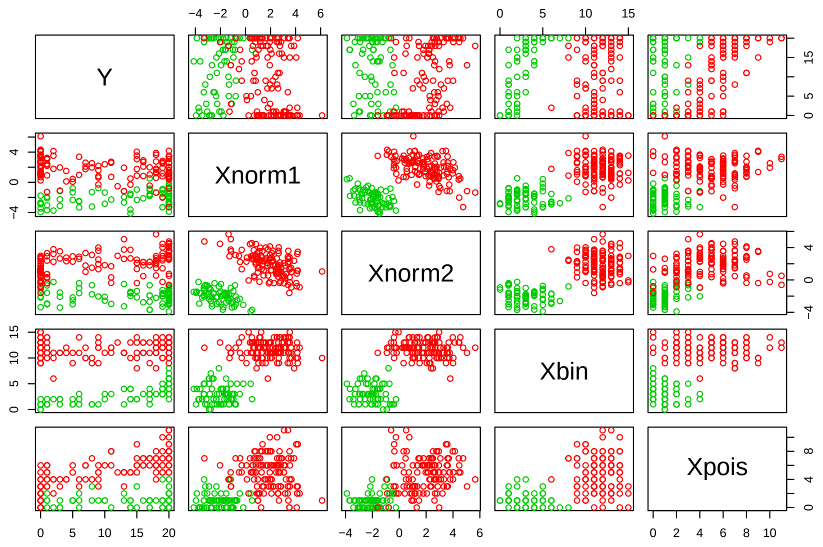
In this section we provide the results of a simulation study on the computational times required by the **flexCWM** package to fit the CWM when the size of the dataset increases. In this study we consider 3 simulation factors:

1. the sample size n (50,000 and 100,000);
2. the number d of covariates (50 and 100);
3. the number of cores (1, 2, 3, and 4).

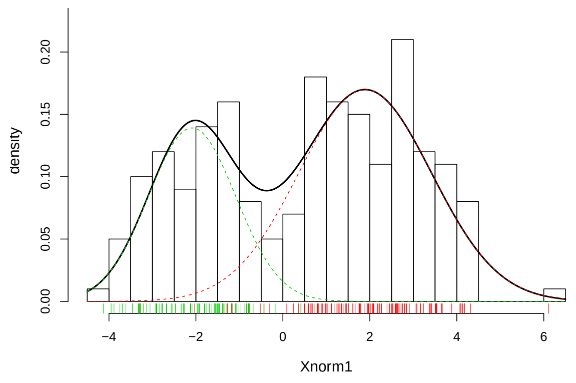
For each combination of n and d, one hundred datasets are sampled from a CWM, for a total of $2 \times 2 \times 100 = 400$ datasets. On each generated dataset, we fit the 14 parsimonious versions of the CWM; this is repeated 4 times for each considered number of cores, which is set using the `cl.cores` argument of `options()`. Computation is performed on a Windows 10 PC, with Intel i7-8550U CPU, 16.0 GB RAM, using R 64-bit, and the elapsed time is computed via the `system.time()` function of the **base** package.

For each combination of n and d, Figure 4 reports the elapsed time (in seconds) in each of the 100 replications when varying the number of cores. The code used to obtain Figure 4(a) is detailed below.

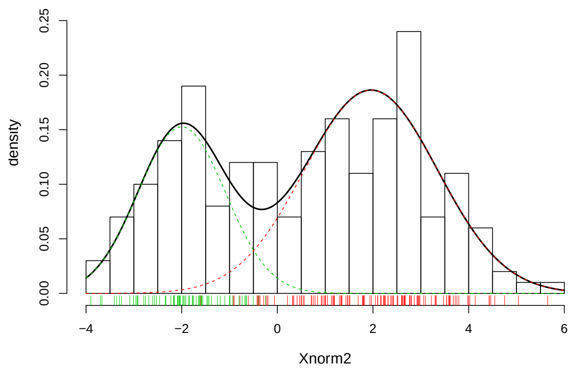
```
R> library("mnormt")
R> cores <- 1:4
R> seed <- 10
R> R <- 100
R> n <- 50000 # or 100000
```



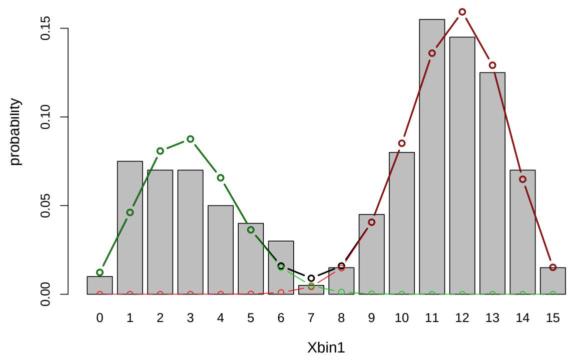
(a) Matrix of scatter plots with colors induced by the fitted classification.



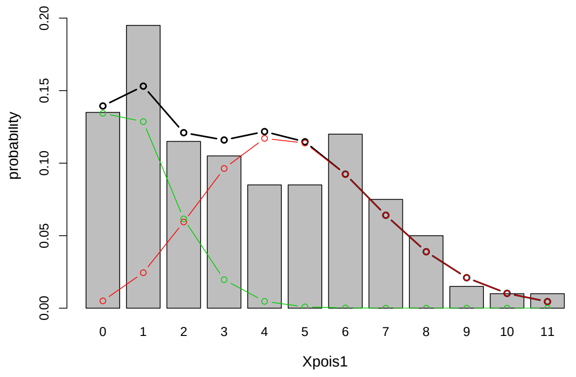
(b) Marginal (in black) and weighted component (in green and red) fitted densities of `Xnorm1`.



(c) Marginal (in black) and weighted component (in green and red) fitted densities of `Xnorm2`.

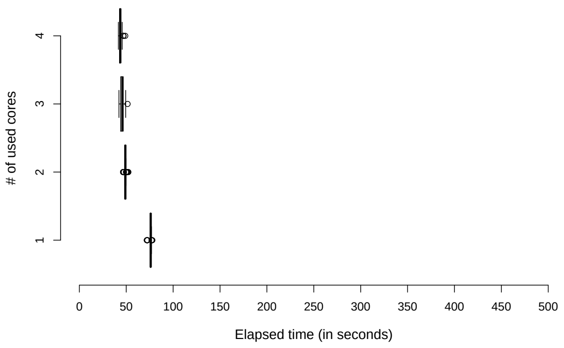


(d) Marginal (in black) and weighted component (in green and red) probability mass functions of `Xbin`.

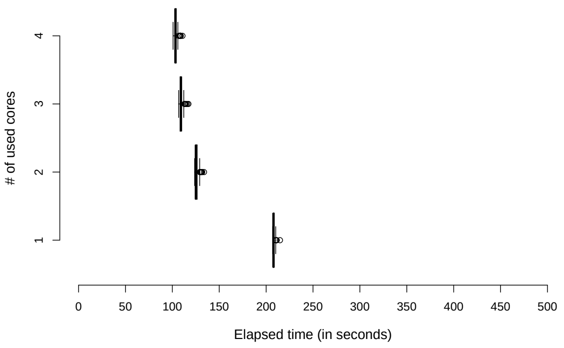


(e) Marginal (in black) and weighted component (in green and red) probability mass functions of `Xpois`.

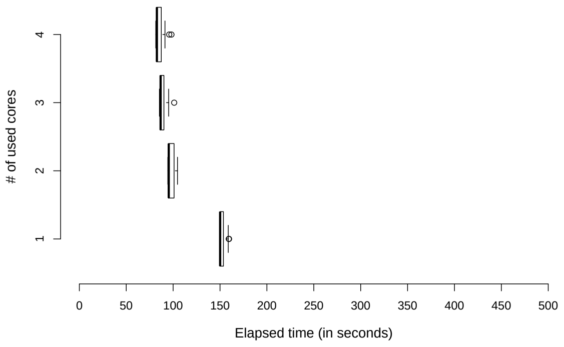
Figure 3: Dataset `ExCWM`: Classification obtained by the VEE linear binomial CWM with $k = 2$ components.



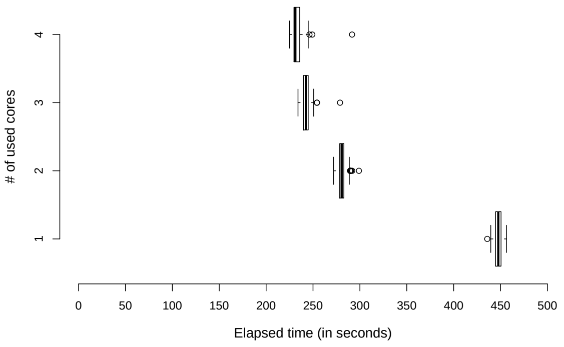
(a) $n = 50,000$ and $d = 50$.



(b) $n = 50,000$ and $d = 100$.



(c) $n = 100,000$ and $d = 50$.



(d) $n = 100,000$ and $d = 100$.

Figure 4: Elapsed time (in seconds) in each of the 100 replications when varying the number of cores.

```
R> d <- 50 # or 100
R> k <- 2
R> set.seed(seed)
R> res <- array(0, c(R, length(cores)))
R> n1 <- n2 <- n/k
R> mu1 <- rep(3, d)
R> mu2 <- -mu1
R> Sigma1 <- Sigma2 <- diag(d)
R> alpha <- 0
R> beta1 <- rep(-1, d)
R> beta2 <- rep(1, d)
R> for (r in 1:R) {
+    X1 <- rmnorm(n = n1, mean = mu1, varcov = Sigma1)
+    X2 <- rmnorm(n = n2, mean = mu2, varcov = Sigma2)
+    X <- rbind(X1, X2)
+    Y1 <- rnorm(n/k, mean = (alpha + beta1 %*% t(X[1:(n/k), ])),
+      sd = 0.5)
+    Y2 <- rnorm(n/k, mean = (alpha + beta2 %*% t(X[(n/k+1):n, ])),
+      sd = 0.5)
+     Y <- c(Y1, Y2)
+     loo <- data.frame(Y = Y, X)
+     for (m in cores) {
+       cat(paste0("---> r = ", r, " m=", m, "\n"))
+       options(cl.cores = m)
+       t <- system.time(
+         foo <- cwm(Y ~ ., data = loo, Xnorm = loo[, -1], k = k,
+           initialization = "kmeans", parallel = TRUE, seed = seed)
+       )
+       res[r, m] <- t[3]
+       cat(paste0("---> time: ", t[3]), "\n")
+     }
+   }
R> b <- ceiling(max(res)/50) * 50
R> boxplot(res, horizontal = TRUE, xlab = "Elapsed time (in seconds)",
+     ylab = "# of used cores", ylim = c(0, b), axes = FALSE)
R> axis(1, at = seq(0, b, by = 50))
R> axis(2, at = 1:4)
```

The remaining plots of Figure 4 are obtained by simply changing the values of `n` and `d` in the code above, as indicated by the comments. To have the four plots in the same scale, the upper limit `b` of the horizontal axes has been set to the highest time of the worst case scenario (`n = 100000` and `d = 100`). As expected, for each combination of n and d, the elapsed time decreases (in average) when the number of cores increases. By using 4 cores, the elapsed time ranges from 41.61 seconds when $n = 50,000$ and $d = 50$, to 291.65 seconds when $n = 100,000$ and $d = 100$.

6. Conclusions

Cluster-weighted models, eminent members of the class of mixtures of regression models with random covariates, have recently become popular in statistics and data mining. Nevertheless, there is a lack of support for them in the most used software packages for statistical computing. In this paper, we have introduced **flexCWM**, a package for the R software environment, specifically conceived for fitting and disseminating cluster-weighted models.

The package offers support for several distributions for both the response variable and the covariates, allowing the latter to be of mixed types. Parsimonious modeling for normal covariates is also supported. This allows for a certain flexibility, and it has been shown that the package can also fit various mixtures of distributions, latent class cluster models for mixed-type variables, mixtures of regression models with fixed covariates, and mixtures of regression models with concomitant variables.

The package returns the most popular information criteria and, when a comparison among different models is needed, computation can take advantage of parallelization on multicore PCs and computer clusters. This is handy when, as it is often the case in practical applications, the number of groups and/or the covariance structure of normal covariates is not *a priori* known. We believe our package may be a practical tool supporting academics and practitioners who are involved in cluster analysis applications.

References

Aitken AC (1926). "On Bernoulli's Numerical Solution of Algebraic Equations." In *Proceedings of the Royal Society of Edinburgh*, volume 46, pp. 289–305.

Akaike H (1973). "Information Theory and an Extension of Maximum Likelihood Principle." In BN Petrov, F Csaki (eds.), *Second International Symposium on Information Theory*, pp. 267–281. Akademiai Kiado, Budapest.

Bagnato L, Punzo A (2013). "Finite Mixtures of Unimodal Beta and Gamma Densities and the k-Bumps Algorithm." *Computational Statistics*, **28**(4), 1571–1597. doi:10.1007/s00180-012-0367-4.

Banfield JD, Raftery AE (1993). "Model-Based Gaussian and Non-Gaussian Clustering." *Biometrics*, **49**(3), 803–821. doi:10.2307/2532201.

Benaglia T, Chauveau D, Hunter DR, Young DS (2009). "**mixtools**: An R Package for Analyzing Mixture Models." *Journal of Statistical Software*, **32**(6), 1–28. doi:10.18637/jss.v032.i06.

Berta P, Ingrassia S, Punzo A, Vittadini G (2016). "Multilevel Cluster-Weighted Models for the Evaluation of Hospitals." *METRON*, **74**(3), 275–292. doi:10.1007/s40300-016-0098-3.

Biernacki C, Celeux G, Govaert G (2000). "Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(7), 719–725. doi:10.1109/34.865189.

Biernacki C, Celeux G, Govaert G (2003). "Choosing Starting Values for the EM Algorithm for Getting the Highest Likelihood in Multivariate Gaussian Mixture Models." *Computational Statistics & Data Analysis*, **41**(3–4), 561–575. doi:10.1016/s0167-9473(02)00163-9.

Böhning D, Dietz E, Schaub R, Schlattmann P, Lindsay BG (1994). "The Distribution of the Likelihood Ratio for Mixtures of Densities from the One-Parameter Exponential Family." *The Annals of the Institute of Statistical Mathematics*, **46**(2), 373–388. doi:10.1007/bf01720593.

Bozdogan H (1987). "Model Selection and Akaikes's Information Criterion (AIC): The General Theory and its Analytical Extensions." *Psychometrika*, **52**(3), 345–370. doi:10.1007/bf02294361.

Bozdogan H (1994). "Theory & Methodology of Time Series Analysis." In *Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach*, volume 1. Kluwer Academic Publishers, Dordrecht.

Celeux G, Govaert G (1995). "Gaussian Parsimonious Clustering Models." *Pattern Recognition*, **28**(5), 781–793. doi:10.1016/0031-3203(94)00125-6.

Dang UJ, Punzo A, McNicholas PD, Ingrassia S, Browne RP (2017). "Multivariate Response and Parsimony for Gaussian Cluster-Weighted Models." *Journal of Classification*, **34**(1), 4–34. doi:10.1007/s00357-017-9221-2.

Dayton CM, Macready GB (1988). "Concomitant-Variable Latent-Class Models." *Journal of the American Statistical Association*, **83**(401), 173–178. doi:10.1080/01621459.1988.10478584.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society B*, **39**(1), 1–38.

DeSarbo WS, Cron WL (1988). "A Maximum Likelihood Methodology for Clusterwise Linear Regression." *Journal of Classification*, **5**(2), 249–282. doi:10.1007/bf01897167.

Fraley C, Raftery A, Scrucca L, Murphy TB, Fop M (2017). ***mclust**: Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation.* R package version 5.4, URL https://CRAN.R-project.org/package=mclust.

Frühwirth-Schnatter S (2006). *Finite Mixture and Markov Switching Models.* Springer-Verlag, New York.

Gershenfeld N (1997). "Nonlinear Inference and Cluster-Weighted Modeling." *The Annals of the New York Academy of Sciences*, **808**(1), 18–24. doi:10.1111/j.1749-6632.1997.tb51651.x.

Grün B, Leisch F (2008). "**FlexMix** Version 2: Finite Mixtures with Concomitant Variables and Varying and Constant Parameters." *Journal of Statistical Software*, **28**(4), 1–35. doi:10.18637/jss.v028.i04.

Harrell FE (2001). *Regression Modeling Strategies.* Springer-Verlag, New York. doi:10.1007/978-1-4757-3462-1.

Hathaway RJ (1986). "A Constrained EM Algorithm for Univariate Normal Mixtures." *Journal of Statistical Computation and Simulation*, **23**(3), 211–230. doi:10.1080/00949658608810872.

Hennig C (2000). "Identifiablity of Models for Clusterwise Linear Regression." *Journal of Classification*, **17**(2), 273–296. doi:10.1007/s003570000022.

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

Hurvich CM, Tsai CL (1989). "Regression and Time Series Model Selection in Small Samples." *Biometrika*, **76**(2), 297–307. doi:10.1093/biomet/76.2.297.

Ingrassia S (2004). "A Likelihood-Based Constrained Algorithm for Multivariate Normal Mixture Models." *Statistical Methods and Applications*, **13**(2), 151–166. doi:10.1007/s10260-004-0092-4.

Ingrassia S, Minotti SC, Punzo A (2014). "Model-Based Clustering via Linear Cluster-Weighted Models." *Computational Statistics & Data Analysis*, **71**, 159–182. doi:10.1016/j.csda.2013.02.012.

Ingrassia S, Minotti SC, Vittadini G (2012). "Local Statistical Modeling via the Cluster-Weighted Approach with Elliptical Distributions." *Journal of Classification*, **29**(3), 363–401. doi:10.1007/s00357-012-9114-3.

Ingrassia S, Punzo A (2016). "Decision Boundaries for Mixtures of Regressions." *Journal of the Korean Statistical Society*, **45**(2), 295–306. doi:10.1016/j.jkss.2015.11.005.

Ingrassia S, Punzo A, Vittadini G, Minotti SC (2015). "The Generalized Linear Mixed Cluster-Weighted Model." *Journal of Classification*, **32**(1), 85–113. doi:10.1007/s00357-015-9175-1.

Karlis D, Xekalaki E (2003). "Choosing Initial Values for the EM Algorithm for Finite Mixtures." *Computational Statistics & Data Analysis*, **41**(3–4), 577–590. doi:10.1016/s0167-9473(02)00177-9.

Leisch F (2004). "**FlexMix**: A General Framework for Finite Mixture Models and Latent Class Regression in R." *Journal of Statistical Software*, **11**(8), 1–18. doi:10.18637/jss.v011.i08.

Leisch F, Grün B (2018). *CRAN Task View: Cluster Analysis & Finite Mixture Models*. Version 2018-03-09, URL https://CRAN.R-project.org/view=Cluster.

Lindsay BG (1995). *Mixture Models: Theory, Geometry and Applications*, volume 5. NSF-CBMS Regional Conference Series in Probability and Statistics, Institute of Mathematical Statistics, Hayward, California.

McQuarrie A, Shumway R, Tsai CL (1997). "The Model Selection Criterion AICu." *Statistics & Probability Letters*, **34**(3), 285–292. doi:10.1016/s0167-7152(96)00192-7.

Punzo A (2014). "Flexible Mixture Modelling with the Polynomial Gaussian Cluster-Weighted Model." *Statistical Modelling*, **14**(3), 257–291. doi:10.1177/1471082x13503455.

Punzo A, Browne RP, McNicholas PD (2016). "Hypothesis Testing for Mixture Model Selection." *Journal of Statistical Computation and Simulation*, **86**(14), 2797–2818. doi:10.1080/00949655.2015.1131282.

Punzo A, Ingrassia S (2015). "Parsimonious Generalized Linear Gaussian Cluster-Weighted Models." In I Morlini, T Minerva, M Vichi (eds.), *Advances in Statistical Models for Data Analysis*, Studies in Classification, Data Analysis and Knowledge Organization, pp. 201–209. Springer-Verlag, Switzerland. doi:10.1007/978-3-319-17377-1_21.

Punzo A, Ingrassia S (2016). "Clustering Bivariate Mixed-Type Data via the Cluster-Weighted Model." *Computational Statistics*, **31**(3), 989–1013. doi:10.1007/s00180-015-0600-z.

Punzo A, Mazza A, McNicholas PD (2018). "**ContaminatedMixt**: An R Package for Fitting Parsimonious Mixtures of Multivariate Contaminated Normal Distributions." *Journal of Statistical Software*, **85**(10), 1–25. doi:10.18637/jss.v085.i10.

Punzo A, McNicholas PD (2016). "Parsimonious Mixtures of Multivariate Contaminated Normal Distributions." *Biometrical Journal*, **58**(6), 1506–1537. doi:10.1002/bimj.201500144.

Punzo A, McNicholas PD (2017). "Robust Clustering in Regression Analysis via the Contaminated Gaussian Cluster-Weighted Model." *Journal of Classification*, **34**(2), 249–293. doi:10.1007/s00357-017-9234-x.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Schwarz G (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, **6**(2), 461–464. `doi:10.1214/aos/1176344136`.

Subedi S, Punzo A, Ingrassia S, McNicholas PD (2013). "Clustering and Classification via Cluster-Weighted Factor Analyzers." *Advances in Data Analysis and Classification*, **7**(1), 5–40. `doi:10.1007/s11634-013-0124-8`.

Subedi S, Punzo A, Ingrassia S, McNicholas PD (2015). "Cluster-Weighted t-Factor Analyzers for Robust Model-Based Clustering and Dimension Reduction." *Statistical Methods & Applications*, **24**(4), 623–649. `doi:10.1007/s10260-015-0298-7`.

Turner R (2018). ***mixreg**: Functions to Fit Mixtures of Regressions*. R package version 0.0-6, URL `https://CRAN.R-project.org/package=mixreg`.

Vermunt JK, Magidson J (2002). "Latent Class Cluster Analysis." In JA Hagenaars, AL McCutcheon (eds.), *Applied Latent Class Analysis*, pp. 89–106. Cambridge University Press, Cambridge.

Wedel M, DeSarbo WS (1995). "A Mixture Likelihood Approach for Generalized Linear Models." *Journal of Classification*, **12**(1), 21–55. `doi:10.1007/bf01202266`.

Wedel M, Kamakura WA (2000). *Market Segmentation: Conceptual and Methodological Foundations*. 2nd edition. Kluwer Academic Publishers, Boston.

Yao W, Wei Y, Yu C (2014). "Robust Mixture Regression Using the t-Distribution." *Computational Statistics & Data Analysis*, **71**, 116–127. `doi:10.1016/j.csda.2013.07.019`.

Affiliation:

Angelo Mazza
Department of Economics and Business
University of Catania
Corso Italia, 55, 95129 Catania, Italy
Telephone: +39/095/7537736
E-mail: `a.mazza@unict.it`
URL: `http://docenti.unict.it/a.mazza`

Antonio Punzo
Department of Economics and Business
University of Catania
Corso Italia, 55, 95129 Catania, Italy
Telephone: +39/095/7537640
E-mail: `antonio.punzo@unict.it`
URL: `http://www.economia.unict.it/punzo`

Salvatore Ingrassia
Department of Economics and Business
University of Catania
Corso Italia, 55, 95129 Catania, Italy
Telephone: +39/095/7537732
E-mail: `s.ingrassia@unict.it`
URL: `http://www.datasciencegroup.unict.it/content/salvatore-ingrassia`

Journal of Statistical Software `http://www.jstatsoft.org/`
published by the Foundation for Open Access Statistics `http://www.foastat.org/`
August 2018, Volume 86, Issue 2 *Submitted:* 2015-02-02
`doi:10.18637/jss.v086.i02` *Accepted:* 2017-07-06