The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–33

Cluster-weighted models using Stata

Daniele Spinelli (corresponding author)
Department of Statistics and Quantitative Methods
University of Milano-Bicocca
Milan, Italy
daniele.spinelli@unimib.it

Salvatore Ingrassia
Department of Economics and Business
University of Catania
Catania, Italy
salvatore.ingrassia@unict.it

Giorgio Vittadini
Department of Statistics and Quantitative Methods
University of Milano-Bicocca
Milan, Italy
giorgio.vittadini@unimib.it

Abstract. The cluster-weighted model (CWM) is a member of the family of mixture of regression models and is also known as mixtures of regressions with random covariates. CWMs refer to the framework of model-based clustering and have their natural application when the research interest requires modeling the relationship between a response variable and a set of covariates using a regression-based approach such as a generalized linear model and the sample is suspected to be composed by heterogeneous latent classes. Software for estimating these models is not yet available in Stata. The aim of this article is to introduce the Stata package `cwmglm`, which allows fitting CWMs based on the most common generalized linear models with random covariates. Moreover, `cwmglm` allows the estimation of parsimonious models of Gaussian distributions, with the parametrization of the variance-covariance matrix based on the eigenvalue decomposition. These features are completely new for Stata users. The `cwmglm` package features goodness-of-fit, bootstrapping and model selection tools. We illustrate the use of `cwmglm` with real and simulated datasets.

Keywords: st0001, `cwmglm`, finite mixtures of regressions with random covariates, model-based clustering, Gaussian parsimonious models

1 Introduction

The Cluster-Weighted Model (CWM) is a member of the family of mixtures of regression models, and it is also known in the literature as mixture of regressions with random covariates and as as saturated mixture regression model (Wedel 2002). The model has been first proposed in the context of media technology under Gaussian assumptions

(Gershenfeld 1997, 1999; Gershenfeld et al. 1999; Schöner 2000). In a sequence of papers, starting from Ingrassia et al. (2012), the CWM was formulated in the statistical framework and the main statistical properties were established.

CWMs refer to the framework of model-based clustering (McNicholas 2016) and have their natural application when the research interest requires modeling the relationship between a response variable Y and a set of covariates $\boldsymbol{X} = (X_1, \ldots, X_p)'$ and the sample is suspected to be composed by heterogeneous latent (i.e., unobserved) classes. In particular, CWM is a mixture approach modeling the joint probability of the response variable and of the explanatory variables for data characterized by unobserved subpopulations.

For the application context of CWMs, approaches based on a mixture of regressions (FMR, McLachlan and Peel 2000; Früwirth-Schnatter 2005) or based on mixture of regressions with concomitant variables (FMRC, Dayton and Macready 1988; Wedel 2002) are also available: these models are estimable in Stata using `fmm` and concern mixtures of conditional distributions of Y given $\boldsymbol{X}$. The CWM is a very flexible model-based clustering approach because it parametrically models the marginal distribution of the covariates and the conditional distribution of the response given the covariates along with latent heterogeneity. This means that, other than estimating latent class-specific regression parameters like FMRs (i.e., the parameters of the conditional distributions), the model estimates the parameters related to the marginal distributions of the covariates (e.g., means and variances) in each latent class. In this framework, CWMs have the advantage of overcoming the intrinsic limitation of FMRs and FMRCs that is the *assignment independence assumption* which hypothesizes that the assignment of the data to the latent classes in the sample is independent of the covariate distribution (Hennig 2000). In other words, this means that the covariates values are assumed *not* to affect the allocation of observations to the latent classes. This assumption might be too restrictive or the allocation mechanism with respect to the covariates might not be known in advance by the researcher and, thus, a more flexible and general model would be necessary. On the contrary, CWMs assumes random covariates and allows assignment *dependence*: the covariate distributions are assumed to vary between latent classes and to affect the allocation of the data points to the latent classes themselves; this results in better classification performance (Punzo 2014).

Therefore,the CWM is a more general approach than FMR and FMRC. As a matter of fact, in Ingrassia et al. (2012), it is shown that under Gaussian assumptions, the CWM includes mixtures of distributions, FMR and FMRC as special cases. A further extension is proposed in Ingrassia et al. (2015) concerning a broad family of CWMs to model discrete responses in which the component conditional densities are assumed to belong to the exponential family and the covariates are allowed to be categorical or numeric.

Flexibility of CWMs has been widely shown in literature. In the framework of of health-care quality assessment, Berta et al. (2016) proposed a multilevel cluster-weighted model for handling hierarchical data for the purpose of measuring the effectiveness of diagnostic procedures or specific treatment episodes with respect to healthcare outcomes.

Very recently, in Berta et al. (2024) the CWM has been taken into consideration for modeling unobserved heterogeneity related to COVID-19 in-hospital mortality during the early stages of the pandemic in Italy (January 2020 to June 2020). In this context, patients have been stratified into 3 classes using a CWM. Such classes differ in terms of mortality (response variable), modeled with a logistic regression as a function of admission day, age, sex and comorbidities (covariates). Within each class, logistic regression highlighted 3 different patterns of mortality risk during the evolution of the pandemic. The first class had constant and low mortality over time, the second class had intermediate risk at baseline and reached its mortality peak around March 2020 and the third class had the largest mortality at baseline but exhibited a declining pattern over time. Concerning covariates, assignment to the latent groups depends on age, admission week and presence of comorbidities. For instance, male and females have an equal probability to belong to class 1, while male have a larger probabilities of being allocated to class 2. Other recent applications of CWMs have included the analysis of administrative data on hospital admissions in Italy (Berta and Vinciotti 2019), studies of Italian tourism data (Soffritti 2021), sales of canned tuna (Diani et al. 2022) and detection of latent classes in data about voles (Subedi et al. 2013).

From the software point of view, Mazza et al. (2018) underlined the scarcity of packages to estimate CWMs; the same authors developed `flexcwm` for R. To our knowledge no other software is currently available. The official Stata command (`fmm`) can be used to fit FMRs; however, it is not capable of estimating CWMs, as it does not model random covariates. Other community-contributed packages (Hernández Alava and Wailoo 2015; Gray and Alava 2018; Jenkins and Rios-Avila 2023; Huismans et al. 2022) address model-based clustering and mixture of regressions under different perspectives but there are no commands able to estimate CWMs.

Hence, the aim of this article is to address the above lack of software availability by introducing `cwmglm`, a Stata package focused on CWMs. Our package is based on the framework of Ingrassia et al. (2012), Ingrassia et al. (2015), Ingrassia and Punzo (2020) and Di Mari et al. (2023) and allows fitting CWMs as mixtures of regressions based on generalized linear models (GLMs) with random covariates. The supported families are Gaussian, Poisson and binomial, while the allowed marginalizations for the covariates are multivariate Gaussian, multinomial, binomial, and Poisson. Multivariate Gaussian models are addressed using the parsimonious mixtures related to the eigenvalue decomposition of the variance-covariance matrix (Banfield and Raftery 1993; Celeux and Govaert 1995), which is currently not estimable in Stata. Other than extending the possibility estimating CWMs to Stata users, `cwmglm` introduces new internal validity measures based on the generalized coefficient of determination and on the three-term decompositions of the total sum of squares and of the total deviance (Di Mari et al. 2023; Ingrassia and Punzo 2020), model selection and bootstrap-based inference. These features are not available in other software estimating CWMs.

The rest of the article is organized as follows. Section 2 outlines the theoretical foundations of our package, section 3 describes `cwmglm`, sections 4 illustrates the use of `cwmglm` in practice, with examples using both real data (Covid-19 admissions and students) and simulated data. Section 5 concludes.

2 Statistical framework

Assume we are provided with a sample of size n $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ concerning a response variable Y and a set of p covariates $\boldsymbol{X} = (X_1, \ldots, X_p)'$ coming from a heterogeneous population formed by K homogeneous latent classes. In the framework of Ingrassia et al. (2012) and Ingrassia et al. (2015), the CWM models the distribution of $(\boldsymbol{X}, Y)$ is as follows:

$$f(\boldsymbol{x}, y; \boldsymbol{\theta}) = \sum_{j=1}^{K} \pi_j p(y|\boldsymbol{x}; \boldsymbol{\zeta}_j) q(\boldsymbol{x}; \boldsymbol{\psi}_j) \tag{1}$$

Here, π_j is the mixing proportion of the latent classes, $p(y|\boldsymbol{x}; \boldsymbol{\zeta}_j)$ is the class j-specific conditional density of the response variable and $q(\boldsymbol{x}; \boldsymbol{\psi}_j)$ is the marginal density of $\boldsymbol{X}$ in class j. Vector $\boldsymbol{\theta}$ includes all the parameters of the conditional density and of the marginal density, which are vectors $\boldsymbol{\zeta}_j$ and $\boldsymbol{\psi}_j$ $(\forall j = 1, \ldots, K)$, respectively. These parameters are to be estimated. The identifiability of the model is established in Ingrassia et al. (2015).

Models are usually selected according to metrics like the Bayesian information criterion (BIC) and the Akaike information criterion (AIC). The optimal model is the one corresponding to the minimum value of AIC or BIC, depending on the adopted criterion. To select among different models, the expression for the most general log-likelihood should be used for the nested model. For instance, comparing a CWM with a FMR would require that the log-likelihood of the FMR is recalculated considering the FMR as a CWM with group-invariant parameters of the covariates; the same applies to FMRC. In `cwmglm`, users can do these comparisons using with the postestimation command `cwmcompare`.

2.1 Models for the conditional density

In our framework, the conditional density belongs to the exponential family and it is modeled as a generalized linear model (GLM, McCullagh and Nelder 2019). In `cwmglm`, the conditional density of the CWM to be modeled can be selected according to the Gaussian, binomial, or Poisson distribution. For example, in the Gaussian case we set $\boldsymbol{\zeta}_j = (\boldsymbol{\beta}, \sigma)$ and $y|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}\boldsymbol{\beta'}_g, \sigma_g^2)$. Thus, the conditional density $p(y|\boldsymbol{x}; \boldsymbol{\zeta}_j)$ is as follows:

$$p(y|\boldsymbol{x}; \boldsymbol{\zeta}_j) = p(y|\boldsymbol{x}; \boldsymbol{\beta}_j; \sigma_j) = \phi(y, \boldsymbol{x}\boldsymbol{\beta}'_j, \sigma_j^2) \tag{2}$$

In the binomial case, the response variable is binary (i.e., $Y \in \{0, 1\}$) and it is assumed that the conditional probability $p(y|\boldsymbol{x}; \boldsymbol{\zeta}_j)$ is characterized by regression coefficients $\boldsymbol{\zeta}_j = \boldsymbol{\beta}_j$ and the expected value $\mu_j(\boldsymbol{x}, \boldsymbol{\beta}_j) = \exp(\boldsymbol{x}\boldsymbol{\beta'}_j)/(1 + \exp(\boldsymbol{x}\boldsymbol{\beta'}_j))$:

$$p(y|\boldsymbol{x}; \boldsymbol{\zeta}_j) = p(y|\boldsymbol{x}; \boldsymbol{\beta}_j) = [\mu_j(\boldsymbol{x}, \boldsymbol{\beta}_j)]^y [1 - \mu_j(\boldsymbol{x}, \boldsymbol{\beta}_j)]^{1-y} \tag{3}$$

For count response variables, assuming that $y \in \mathbb{N}$ and that $Y|\boldsymbol{x} \sim \text{Pois}(\mu_g)$ leads

to the Poisson model in equation (4) with expected value $\mu_j(\boldsymbol{x}, \boldsymbol{\beta}_j) = \exp(\boldsymbol{x}\boldsymbol{\beta'}_j)$.

$$p(y|\boldsymbol{x}; \boldsymbol{\zeta}_j) = p(y|\boldsymbol{x}; \boldsymbol{\beta}_j) = \frac{\mu_j(\boldsymbol{x}, \boldsymbol{\beta}_j)^y \exp\left[\mu_j(\boldsymbol{x}, \boldsymbol{\beta}_j)\right]}{y!} \tag{4}$$

2.2 Models for the marginal density

Assume that the covariates are formed by continuous and discrete variables such that $\boldsymbol{X} = (\boldsymbol{U}', \boldsymbol{V}', \boldsymbol{W}', \boldsymbol{Z}')'$, where $\boldsymbol{U}$ includes continuous covariates, $\boldsymbol{V} \in \mathbb{N}$, $\boldsymbol{W}$ includes binary covariates, and all the elements of $\boldsymbol{Z}$ are unordered categorical variables. In `cwmglm`,the marginal distribution of these covariates can be modeled as Gaussian ($\boldsymbol{U}$), Poisson ($\boldsymbol{V}$), binomial ($\boldsymbol{W}$), or multinomial ($\boldsymbol{Z}$). In general, the variables in $\boldsymbol{X}$ are not required to coincide with the covariates of the GLM underlying the conditional density. The possible discrepancy between the variables in the marginal density and the covariates of the GLM can be viewed as the results of some constraints on the conditional or on the marginal density. For instance, omitting a covariate from the marginalization is equivalent to constraining that the marginal density of such covariate have the same parameters in all of the latent classes.

For d-variate Gaussian marginals, we have $q(\boldsymbol{u}; \boldsymbol{\psi}_j) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ where $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are the mean vector and the variance covariance matrix respectively of the j-th component. In `cwmglm` the estimation of $\boldsymbol{\Sigma}_j$ considers the eigendecomposition of Celeux and Govaert (1995).

$$\boldsymbol{\Sigma}_j = \lambda_j \boldsymbol{D}_j \boldsymbol{A}_j \boldsymbol{D}'_j \tag{5}$$

From a geometrical point of view, classes can be represented as ellipsoids centered at the mean vector $\boldsymbol{\mu}_j$, where $\lambda_j = |\boldsymbol{\Sigma}_j|^{\frac{1}{d}}$ is the volume of class j, $\boldsymbol{D}_j$ represents the orientation and $\boldsymbol{A}_j$ represents the shape ($|\boldsymbol{A}_j| = 1$). Depending on constraints on λ_j, $\boldsymbol{D}_j$ and $\boldsymbol{A}_j$, latent classes may have equal or variable volume and spherical, equal or variable shape; the orientation may be axis-aligned, equal or variable. Spherical shape means that the multivariate Gaussian variables underlying $\boldsymbol{\Sigma}$ are homoskedastic and uncorrelated (i.e., $\boldsymbol{A}_j = \boldsymbol{D}_j = \boldsymbol{I}\ \forall j = 1, \ldots, K$). Equal orientation means that latent classes are constrained to have the same orientation, variable orientation means that orientation is class-specific; the same concept applies to volume and shape. Axis-aligned orientation refers to the situation in which the ellipsoid representing the covariance matrix have their major axes aligned to the axes of the coordinates system related to the variables that are characterized by the same covariance matrix. The combinations of these assumptions on λ, $\boldsymbol{D}$ and $\boldsymbol{A}$ lead to 14 parsimonious models which are detailed in table 1.

Table 1: Summary of the normal parsimonious models

Volume	**Shape**	**Orientation**	**Model name**	$\boldsymbol{\Sigma}_j$	**N. parameters**
Equal	Spherical		EII	$\lambda \boldsymbol{I}$	1
Variable	Spherical		VII	$\lambda_j \boldsymbol{I}$	K
Equal	Equal	Axis-Aligned	EEI	$\lambda \boldsymbol{A}$	q
Variable	Equal	Axis-Aligned	VEI	$\lambda_j \boldsymbol{A}$	$K + d - 1$
Equal	Variable	Axis-Aligned	EVI	$\lambda \boldsymbol{A}_j$	$1 + K(d-1)$
Variable	Variable	Axis-Aligned	VVI	$\lambda_j \boldsymbol{A}_j$	Kd
Equal	Equal	Equal	EEE	$\lambda \boldsymbol{DAD'}$	$d(d+1)/2$
Variable	Equal	Equal	VEE	$\lambda_j \boldsymbol{DAD'}$	$K + d - 1 + d(d-1)/2$
Equal	Variable	Equal	EVE	$\lambda \boldsymbol{DA}_j \boldsymbol{D'}$	$1 + K(d-1) + d(d-1)/2$
Variable	Variable	Equal	VVE	$\lambda_j \boldsymbol{DA}_j \boldsymbol{D'}$	$Kd + d(d-1)/2$
Equal	Equal	Variable	EEV	$\lambda \boldsymbol{D}_j \boldsymbol{A} \boldsymbol{D}_j'$	$d + Kd(d-1)/2$
Variable	Equal	Variable	VEV	$\lambda_j \boldsymbol{D}_j \boldsymbol{A} \boldsymbol{D}_j'$	$K + d - 1 + kD(D-1)/2$
Equal	Variable	Variable	EVV	$\lambda \boldsymbol{D}_j \boldsymbol{A}_j \boldsymbol{D}_j'$	$1 + K(d-1) + Kd(d-1)/2$
Variable	Variable	Variable	VVV	$\lambda_j \boldsymbol{D}_j \boldsymbol{A}_j \boldsymbol{D}_j'$	$Kd(d+1)/2$

K: number of latent classes
d: number of parameters in $\boldsymbol{x}$

Regarding binomial covariates, their density can be obtained by replacing y with v in equation (3) and by assuming, in the same equation, that the expected value μ_j is constant. The same reasoning can be applied to Poisson covariates. Multinomial variables $\boldsymbol{Z}$ can be represented as $\boldsymbol{z}^r = (z_{r1}, \ldots, z_{rl})$ where $z^{rs} = \mathbb{1}(z_r = s)$ and $s \in \{1, \ldots, l\}$ and modeled as independent binomial terms.

2.3 EM estimation

The log-likelihood $l(\boldsymbol{\theta})$ corresponding to equation (1) is as follows:

$$l(\boldsymbol{\theta}) = \sum_{j=1}^{K}\sum_{i=1}^{N} \tau_{ij} \ln \pi_j + \sum_{j=1}^{K}\sum_{i=1}^{N} \tau_{ij} \ln[p(y_i|\boldsymbol{x}_i; \boldsymbol{\zeta}_j)] + \sum_{j=1}^{K}\sum_{i=1}^{N} \tau_{ij} \ln[q(\boldsymbol{x}_i; \boldsymbol{\psi}_j)] \quad (6)$$

where τ_{ij} is an indicator such that $\tau_{ij} = 1$ if observation i belongs to class j and $\tau_{ij} = 0$ otherwise. It is maximized using the EM algorithm (Dempster et al. 1977). In the t-th iteration, the E-step consists of calculating $\hat{\tau}_{ij}$ the posterior probability related to τ_{ij} (equation (7)) given the current expectation of $\boldsymbol{\theta}^t$.

$$\hat{\tau}_{ij}^t = \frac{\pi_j^t p(y_i|\boldsymbol{x}_i; \boldsymbol{\zeta}_j^t) q(\boldsymbol{x}_i; \boldsymbol{\psi}_j^t)}{f(\boldsymbol{x}, y, \boldsymbol{\theta}^t)} \quad (7)$$

In the M-step, the value of $\hat{\tau}_{ij}$ is plugged into equation (6) and then the current log-likelihood is maximized, obtaining the new estimates of the parameters, see Ingrassia

et al. (2015) for details. The maximization of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ implies that: $\pi_j = n^{-1} \sum_{i=1}^{n} \tau_{ij}$. Maximizing the conditional and marginal parts of (6) with respect to $\boldsymbol{\zeta}_j$ and $\boldsymbol{\psi}_j$ $(\forall j = 1, \ldots, K)$ is equivalent to independently maximizing the expressions for the additive components of equation (6). To establish convergence **cwmglm** applies the Aitken acceleration (Aitken 1927) and the procedure stops when $l_{\infty}^{t+2} - l^{t+1} < \epsilon$ where the asymptotic log-likelihood estimate is:

$$l_{\infty}^{t+2} = l^{r+1} + \frac{l^{t+2} - l^{t+1}}{1 - a^{t+1}} \tag{8}$$

Once the posterior probabilities $\hat{\tau}_{ij}$ are obtained from the EM algorithm, observations can be allocated to latent classes using the maximum *a posteriori* (MAP) classification. The posterior probabilities are also termed as "soft" group membership because all of the observations have a positive probability of belonging to all of the classes. The MAP classification is represented in equation (9) and is also referred as "hard" group allocation since it assigns each observation to a single class (Di Mari et al. 2023) and, therefore, it is a discrete variable.

$$MAP(\hat{\tau}_{ig}) = \begin{cases} 1 & \text{if } \max_j(\hat{\tau}_{ij}) \text{ occurs in g} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

2.4 Measures of fit

Information criteria

For model selection, `cwmglm` implements the two standard information criteria in the maximum likelihood framework: the AIC and the BIC. Given a sample of N observations and a CWM characterized by r estimated parameters and the maximized value of the log-likelihood $\hat{l}$ the criteria are as follows:

$$\begin{aligned} AIC &= 2r - 2\hat{l} \\ BIC &= r \ln N - 2\hat{l} \end{aligned} \tag{10}$$

The lower AIC or BIC, the better is the model fit.

Deviance decomposition and generalized R^2

To evaluate the goodness of fit of the GLM underlying the conditional density in equation (1), `cwmglm` adopts the measures of fit outlined by Di Mari et al. (2023) which extend previous results in the framework of Gaussian models given in Ingrassia and Punzo (2020). These measures, in turn, extend deviance-based measures of lack of fit proposed by Cameron and Windmeijer (1996) to clusterwise regressions and are defined both at the class level (i.e., locally) and at the whole sample level (i.e., globally).

From the log-likelihood in equation (6) it is convenient to express the second additive component as follows:

$$\sum_{j=1}^{K}\sum_{i=1}^{N}\tau_{ij}\ln[p(y_i|\boldsymbol{x}_i;\boldsymbol{\beta}_j,\phi_j)]=\sum_{j=1}^{K}\sum_{i=1}^{N}\tau_{ij}\ln[p(y_i;\mu_{ij};\phi_j)]=\sum_{j=1}^{K}l(\boldsymbol{\mu}_j,\phi_j) \tag{11}$$

where $\mu_{ij}=E(Y_i|\boldsymbol{X}_i;\boldsymbol{\beta}_j)$ is the expected value for the i-th observation in the j-th latent class of the GLM underlying equation (11) and ϕ_j is the dispersion parameter. For each group, the total within deviance WD_j can be decomposed into two terms: the local explained deviance EWD_j and the local residual deviance RWD_j.

$$\begin{aligned} WD_j &= [l(\boldsymbol{y},\hat{\phi}_j)-l(\bar{\boldsymbol{y}}_j,\hat{\phi}_j)]=EWD_j+RWD_j \\ EWD_j &= [l(\hat{\boldsymbol{\mu}}_j,\hat{\phi}_j)-l(\bar{\boldsymbol{y}}_j,\hat{\phi}_j)] \\ RWD_j &= [l(\boldsymbol{y},\hat{\phi}_j)-l(\hat{\boldsymbol{\mu}}_j,\hat{\phi}_j)] \end{aligned} \tag{12}$$

Considering the whole sample, the total deviance TD can be additively decomposed into the within-group deviance ($WD=\sum_{j=1}^{K}WD_j$) and the between-group deviance BD:

$$\begin{aligned} BD &= \sum_{j=1}^{K}BD_j \\ BD_j &= [l(\bar{\boldsymbol{y}}_j,\hat{\phi}_j)-l(\bar{\boldsymbol{y}},\hat{\phi}_j)]. \end{aligned} \tag{13}$$

The former measures the class-specific dispersion of the observations with respect to the latent class averages, while the latter contains information about the separation between classes in terms of units of the response variable.

The global total deviance TD can be decomposed as follows:

$$\begin{aligned} TD = WD+BD &= EWD+RWD+BD \\ EWD &= \sum_{j=1}^{K}EWD_j \\ RWD &= \sum_{j=1}^{K}RWD_j \end{aligned} \tag{14}$$

where EWD is the overall explained within deviance and RWD is the global residual within deviance. From the decomposition outlined in equation (12)-(14) it is possible to define a generalized coefficient of determination both locally (R^2_j) and globally (R^2). Di Mari et al. (2023) define the former as the normalized local deviance $R^2_j=EWD_j/WD_j$ and the latter as $R^2=EWD/WD$.

In particular, R_j^2 can be seen as the proportion of the local deviance in the j-th group that cannot be explained by the intercept-only GLM in that group, but which can be explained by the linear predictor $\eta_{ij} = \widehat{\boldsymbol{\beta}}_j \boldsymbol{x}_i$ of the GLM. The overall R^2 can be interpreted as the proportion of the within deviance explained by the fitted mixture of GLMs.

Furthermore, from equation (14) it is possible to define normalized indicators by dividing both sides by TD (Ingrassia and Punzo 2020). Three measures are then obtained: $NEWD = EWD/TD$, $NRWD = RWD/TD$ and $NBD = BD/TD$. The first is the proportion of the deviance explained by the inclusion of the covariates in the conditional model (i.e., in the GLM), the second represents the proportion of total deviance left unexplained by the regressions, and NBD is a measure of association between the response variable and the latent group variable (i.e., the proportion of the deviance explained by the classes).

3 The cwmglm package

The `cwmglm` package requires Stata 16 or newer versions and allows fitting CWMs based on mixtures of the most common GLMs with random covariates. The supported families are Gaussian, Poisson and binomial and the marginal distributions allowed for the covariates are multivariate Gaussian, multinomial, binomial, and Poisson. The syntax of `cwmglm` is designed to maximize flexibility in model specification. In particular, users can fit both mixtures of distributions and FMR, are nested in equation (1).
In `cwmglm`, the only mandatory option is `posterior`, and the general syntax is as follows:

```
cwmglm [depvar indepvars] [if] [in], [k(#)] [glm options] [marginalization
options] [inizialization options] [maximization options] [display options]
```

After the command statement, users may optionally specify the response variable (*depvar*) and the covariates (*indepvars*) of the conditional part of the CWM. If they are specified, the conditional part of the CWM is a GLM of the family defined by the `family` option (see section 3.1). Otherwise, the conditional part of the model is not considered: this is equivalent to setting $\pi_j p(y|\boldsymbol{x};\boldsymbol{\zeta}_j) = 1 \ \forall j$ in equation (1) or assuming that the second addend in the right-hand side of equation (6) is equal to zero.

3.1 Options

The option `k(#)` sets the number of latent classes, which is parameter K in equation (1). Leaving this option unspecified leads to the estimation of a CWM with $K = 2$.

GLM options

The only option controlling the conditional density of the response variable that can be specified is `family(familyname)`. That option specifies the distribution of the response

variable *depvar* for the GLM. The default is `family(gaussian)` (link identity, conditional density in equation (2)). The other allowed distributions are `family(binomial)` (link logit, conditional density in equation (3)) and `family(poisson)` (log link , conditional density in equation (4)).

Marginalization options

The marginalization options control the specification of the marginal density $q(\boldsymbol{x}_i; \boldsymbol{\psi}_j)$. The `cwmglm` package allows covariates to be multivariate Gaussian, Poisson, binomial and multinomial distributed. The detailed options are as follows:

- `xnormal(`*varlist*`)` specifies that the variables in `varlist` follow a (multivariate) Gaussian distribution. If this option is specified, users can model the variance-covariance matrix of normal covariates using one of the fourteen parsimonious models of Celeux and Govaert (1995). The options are based on the taxonomy of table 1 and are `eii`, `vii`, `eei`, `vei`, `evi`, `vvi`, `eee`, `vee`, `eve`, `vve`, `eev`, `vev`, `evv` and `vvv`. If exactly one variable is included in `xnorm(`*varlist*`)`, the possible options are equal variance (using `eee`) or different variances (option `vvv`). Estimates for the models EVE and VVE are obtained using the minorization-majorization algorithm (Browne and McNicholas 2014; Sarkar et al. 2020).

- `xpoisson(`*varlist*`)` specifies that the variables in `varlist` follow independent Poisson distributions

- `xbinomial(`*varlist*`)` specifies that the variables in `varlist` follow independent binomial distributions

- `xmultinomial(`*varlist*`)` specifies that the variables in `varlist` follow independent multinomial distributions. Factor variable syntax is not allowed. Categories are detected automatically.

Let us consider a theoretical example for the sake of clarity. Let us assume that the covariates of a model to be estimated are `X` (continuous), `U`, and `V` (both binary). Users modeling X as multivariate Gaussian distributed with equal volume, variable shape and variable orientation (EVV) and modeling U and V as binomial covariates, should set the marginalization options in the following way:

```
cwmglm ..., posterior(stub) xnormal(X) evv xbinomial(U V)
```

This is equivalent to assuming that the parameters to be estimated in the marginal distribution in equation (1) are given by $\boldsymbol{\psi}_j = (\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}, p_{uj}, p_{vj})$, and that the underlying marginal density in latent class j would be:

$$q([\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{V}]; \boldsymbol{\psi}_j) = \phi(\boldsymbol{x}, \mu_j, \boldsymbol{\Sigma_j})[p_{uj}]^u[1-p_{uj}]^{1-u}[p_{vj}]^v[1-p_{vj}]^{1-v} \qquad (15)$$

where $\phi(\boldsymbol{x}, \mu_j, \boldsymbol{\Sigma_j})$ is the density of the bivariate Gaussian distribution with expected value $\boldsymbol{\mu}_j$ and variance-covariance matrix $\boldsymbol{\Sigma_j}$. The parsimonious model EVV implies that $\boldsymbol{\Sigma_j} = \lambda \boldsymbol{D}_j \boldsymbol{A}_j \boldsymbol{D}_j'$. Scalars p_{uj} and p_{vj} are the parameters for the underlying binomial distributions related to $\boldsymbol{U}$ and $\boldsymbol{V}$ in latent class j.

Initialization options

The initialization options control the initial values of the CWM. The main option is `start(svmethod)`, which that specifies how the initial component membership (*hard* class memberships) or component membership probabilities (*soft* class memberships) to be supplied to the EM algorithm are obtained. This option can be specified as follows:

- `start(kmeans)` specifies that the starting values are computed by assigning each observation to an initial latent class that is determined by running a kmeans cluster analysis; this is the default.
- `start(randomid)` specifies that the starting values are computed by randomly assigning observations to initial classes.
- `start(randompr)` specifies that the starting values are computed by randomly assigning initial class probabilities.
- `ndraws(#)`, applies only if `start(randompr)` or `start(randomid)` are specified. It specifies #, the number of random draws for used to select the starting values(initial class memberships or probabilities); among the # runs, starting values are selected if they have the highest log-likelihood. The default value is 10.
- `start(custom)` causes `cwmglm` to initialize the EM algorithm with user-specified initial class memberships or probabilities. If this option is chosen, users must specify the `initial` option.
- `initial(`*`varlist`*`)` is a variable set containing the starting values for `start(custom)`. The user must specify a k-dimensional *varlist*, where k is the number of latent classes. The *`varlist`* in this option may represent soft or hard group memberships; it applies only if `start(custom)` is specified.

Maximization options

These options control the settings of the iterative maximization procedures occurring in `cwmglm`

- `iterate(#)`, is the maximum number of EM iterations. The default is 1200.
- `iteratexnorm(#)` is the maximum number of iterations to be used for the parsimonious models. It only affects the estimations of VEE, EVE, VVE, VEV and VEI models. Default is 1200.

- `convcrit(#)` is the stopping criterion for the Aitken acceleration procedure. The default threshold is 1e-5.

Display options

These options reduce the output in the results window.

- `nolog` suppresses the iteration log. Reports only the required iterations to converge and the log-likelihood at convergence.
- `noclustertable` forces `cwmglm` not to display the clustering table. If this option is specified the package supresses the output related to the number of observations allocated to each latent class and prior class probabilities $\boldsymbol{\pi} = \pi_1, \ldots, p_K$.
- `nodeviance` forces `cwmglm` not to display the deviance measures outlined in section **cwmglm**.
- `nomarginal` forces **cwmglm** not to display the parameters of the marginal densities such as mean vectors and covariances of multivariate normal covariates.
- `noregtable` forces `cwmglm` not to display regression table related the parameters of the generalized linear model underlying the conditional part of the CWM.

3.2 Saved results

The following scalars are returned by `cwmglm`:

- `e(N)`, the number of observations used for the estimation;
- `e(dof)`, the number of estimated parameters;
- `e(ll)`, the value of the maximized log likelihood;
- `e(bic)`, the BIC;
- `e(aic)`, the AIC;
- `e(converged)`, a binary indicator that is equal to 1 if the EM has reached convergence and 0 otherwise.

The `cwmglm` package returns the following matrices:

- `e(b)`, the coefficients vector of the GLM;
- `e(V)`, the variance-covariance matrix of the GLM;
- `e(phi0)`, the dispersion parameter of the GLM;

- `e(globaldeviance)`, a 2×4 matrix containing the overall residual deviance, the overall explained deviance, the between deviance, and the total deviance, as defined by equation (14). These measures are both in their natural units and normalized.
- `e(localdeviance)`, a $4 \times (K + 1)$ matrix including the within deviance decomposition $WD = EWD + RWD$ of the GLMs and the generalized R^2 (see equation (12)-14).
- `e(R2)` a vector including the group-specific (R_j^2) and overall (R^2) generalized coefficients of determination for the GLM;
- `e(cl_table)`, clustering table that contains the estimated size for each latent class based on MAP group memberships (equation (9));
- `e(prior)`, estimates of π_j the prior latent classes weights (equation (1)).
- `e(mu)`, the mean vector of the variables marginalized as multivariate Gaussian distributed (`xnormal` option).
- `e(sigma)`, the variance-covariance matrices of the variables marginalized as multivariate Gaussian distributed (`xnormal` option).
- `e(lambda)`, a $m_{Poisson} \times K$ matrix of the parameters the covariates with Poisson marginalization, where $m_{Poisson}$ is the number of variables declared in the `xpoisson` option.
- `e(p_binomial)`, a $m_{binomial} \times K$ matrix of the parameters the covariates with binomial marginalization, where $m_{binomial}$ is the number of variables declared in the `xbinomial` option.
- `e(p_multi_#)` probabilities for the variable marginalized as multinomial distributed. It returns m matrices named `e(p_multi_1)`, ..,`e(p_multi_m)`, where m is the number of multinomial variables declared in `xmultinomial`.
- `e(ic)`, a vector containing the AIC and the BIC.

`cwmglm` returns the following macros:

- `e(depvar)`, the response variable of the GLM;
- `e(indepvars)`, the list of covariates used in the GLM;
- `e(cmd)`, cwmglm;
- `e(xnormal)`, a `varlist` containing the variables with normal marginalization;
- `e(xnormodel)` the parsimonious model used for the normal marginalization (e.g., `vvv`, `eee`);

- `e(xpoisson)`, a `varlist` containing the variables with poisson marginalization;
- `e(xbinomial)`, a `varlist` containing the variables with binomial marginalization;
- `e(xmultinomial)` , a `varlist` containing the variables with multinomial marginalization;
- `e(xmultinomial_fv)` , a `varlist` containing the variables with multinomial marginalization in factor variable notation;
- `e(glmcmd)`, the command to fit the GLM within each EM iteration.

The only function returned by `cwmglm` is `e(sample)`, which marks the estimation sample.

3.3 Postestimation commands

Here we outline the syntax for prediction (the `predict` command), bootstrapping (`cwmbootstrap` command) and model comparison (the `cwmcompare` command).

Syntax for predict

After `cwmglm`, the `predict` command can be used to classify observations into the K latent classes. To calculate the posterior class memberships (i.e., soft class membership) the following syntax is used:

`predict` *stub*, `posterior`

This command creates K new variables with a prefix given by *stub*. For instance, if $K = 3$ and *stub* is `z`, Stata would create 3 new variables: `z1`, `z2` and `z3`. The prediction of hard group memberships (i.e., discrete allocation of observations to classes) is also possible using the MAP. The syntax is as follows:

`predict` *newvarname*, `map`

This creates a new variable *varname* that assigns hard group membership according to the *maximum a posteriori probability*. This means that observation i is assigned to latent class j if $\hat{z}_{ij} = \max_{h=1..k}(\hat{z}_{ih})$.

Syntax for cwmbootstrap

This postestimation command `cwmbootstrap` uses the results returned by `cwmglm` (see section 3.2) to estimate bootstrap standard errors for the following estimates:

- `e(b)` , the coefficient vector of the GLM;
- `e(p_multi_#)`, the probabilities of a each outcome for the xmultinomial variables;

- `e(p_binomial)`, the probabilities of a positive outcome for the xbinomial variables;
- `e(lambda)`, the mean of the xpoisson variables;
- `e(mu)`, the mean of the xnorm variables.

`cwmbootstrap` returns only matrices related to the inference on the above-mentioned parameters. Such matrices are:

- `r(b)`, the inference table for e(b);
- `r(p_multi)`, the inference table for e(p_multi_#);
- `r(p_binomial)`, the inference table for e(p_binomial);
- `r(lambda)`, the inference table for e(lambda);
- `r(mu)`, the inference table for e(mu).

The only option for `cwmbootstrap` is `nreps` , an integer that sets the number of bootstrap replications. The syntax is as follows:

```
cwmbootstrap, reps(#)
```

Syntax for cwmcompare

The `cwmcompare` command uses the AIC and BIC to compare different moldes obtained using from `cwmglm`. The syntax is as follows:

`cwmcompare` *`namelist`*

where *`namelist`* is a list of estimates saved using `estimates store`. The `cwmcompare` command recalculates the information criteria using the most general specification from the members of *`namelist`* as well as suitable constraints in order to make estimates comparable. The returned results are as follows:

- `r(table)` is a matrix containing the AIC and BIC for the model in *`namelist`*;
- `r(bestAIC)` and `r(bestBIC)` are macros containing the members of *`namelist`* corresponding to the models minimizing the AIC and BIC respectively.

4 Examples

This section provides three examples to illustrate the use `cwmglm` in the empirical setting. The first two are based on real data, while the last one is based on artificial datasets. Additional examples are available in the online appendix.

4.1 The covid dataset

The first real dataset includes a random sample of administrative data regarding 1,000 hospital admissions during the first COVID-19 wave (February 2020 to May 2020) in a single Italian province. The COVID-19 dataset is used to illustrate how to build CWMs with the `cwmglm` command. The variables are as follows:

- `los`, the length of stay (LOS) (in days);
- `admday`, the day of admission (standardized variable);
- `age`, the patient's age (standardized variable);
- `mortrisk`, the standardized in-hospital mortality risk score based on comorbidities(see Fabbian et al. (2017) for details);
- `n1`, the number of procedures that the patient underwent during their hospital stay;
- `female`, =1 if the patient is female, or =0 otherwise.

The empirical strategy is concerned with estimating CWMs for different numbers of latent classes ($K = 2, 3, 4$) and comparing their fit. The conditional part is aimed at explaining the LOS (`los`) conditional on the period of admission (`admday`), on the patient's demographic characteristics (age and sex) and on patient complexity as proxied by the number of procedures characterizing the hospital stay (`n1`) and mortality risk (`mortrisk`). This section reports the estimation results only for $K = 2$ and the summary of the fit of each models. For further details, we refer readers to the appendix (see online supplementary material).

Since `los` is an integer representing the number of days spent in the hospital, the conditional part of the CWM is based on the Poisson family. In this framework, the basic syntax for `cwmglm` is as follows:

```
cwmglm los admday age mortrisk n1 female, k(2) family(poisson)
```

In its current state, the command above would not model the marginal density of the covariates; it would assume $q(\boldsymbol{X}; \boldsymbol{\psi}_j) = 1$ in equation (1) and fit a FMR. Thus, to fit a CWM (a mixture of regressions with random covariates), the command must be updated with the marginalization options. The continuous covariates (`admday age mortrisk`) are modeled as a multivariate Gaussian distribution with a VVV variance-covariance matrix, the most general parsimonious model. Thus, the option `xnormal(admday age mortrisk)` must be added; although the VVV model is the default, the `vvv` option is specified for the sake of illustration. Omitting `vvv` would estimate the same model. Since `n1` is a count variable it is marginalized as Poisson distributed. Gender (the variable `female`) is modeled using the binomial distribution. Overall, the command would be updated as follows:

```
cwmglm los admday age mortrisk n1 female, k(2) family(poisson) xnormal(admday
age mortrisk) vvv xpoisson(n1) xbinomial(female)
```

The command above is also used as the building block to estimate CWM with $K = \{3, 4\}$ by simply changing the value of `k()`. Executing it in Stata with the addition of the `nolog nomarg` options to avoid excessive tables would lead to the following output:

```
. cwmglm los admday age mortrisk n1 female, xnormal(admday age mortrisk) vvv  ///
> xpoisson(n1) xbin(female) k(2) family(poisson) nolog nomarg
initializing EM...
EM iteration       129:log-likelihood=  -9883.11853
Prior Probabilities
```

g1	g2
.3094271	.6905729

Clustering Table

g1	g2
306	694

Information criteria

AIC	BIC
19832.24	19994.19

Local Deviance

	g1	g2	Overall
WD	2487.11	2084.844	4571.954
RWD	1099.886	1717.78	2817.666
EWD	1387.225	367.063	1754.288
R^2	.5577657	.1760626	.3837064

Global Deviance

	RWD	EWD	BD	TD
Deviance	1754.288	2817.666	2919.108	7491.062
Normalized~e	.2341841	.3761371	.3896788	1

los	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
g1						
admday	-.0138496	.0083668	-1.66	0.098	-.0302482	.002549
age	.4370916	.0098406	44.42	0.000	.4178043	.4563789
mortrisk	-.332969	.009551	-34.86	0.000	-.3516887	-.3142493
n1	.2316298	.0054555	42.46	0.000	.2209373	.2423223
female	-.2845163	.0144488	-19.69	0.000	-.3128355	-.2561971
_cons	2.49537	.0231239	107.91	0.000	2.450048	2.540692
g2						
admday	.0412585	.010733	3.84	0.000	.0202222	.0622948
age	.2730241	.0338367	8.07	0.000	.2067054	.3393429

```
    mortrisk |  -.3688652   .0345665   -10.67   0.000    -.4366143   -.3011161
          n1 |   .1556781   .0091091    17.09   0.000     .1378246    .1735316
      female |  -.1753901   .0251741    -6.97   0.000    -.2247305   -.1260498
       _cons |   1.615551   .0371954    43.43   0.000      1.54265    1.688453
------------------------------------------------------------------------------
```

The first table in the Stata output above shows that the estimated groups (`g1` and `g2`) have prior probabilities (π from equation (1)) equal to $\hat{\pi}_1 = 0.309$ and $\hat{\pi}_2 = 0.691$, while the second one reports that hard group membership would correspond to allocating 306 and 694 observations to `g1` and `g2`, respectively. Overall, the GLM is characterized by a generalized determination coefficient $R^2 = 0.38$ (from the fourth table in the Stata output), this is equivalent to a 38 % proportionate increase in the local explained deviance due to the inclusion of the covariates in the regression models (Cameron and Windmeijer 1997; Brilleman 2011). The same figure is equal to $R^2_1 = 0.56$ for `g1` and $R^2_2 = 0.18$ for `g2`. This implies that the GLM in the first latent class fits the data better than `g2`. The between deviance is a separation measure: it indicates the degree of separation of the latent class along the response variable axis (Di Mari et al. 2023). The normalized between deviance is $2919.108/7491.062 = 0.389$, which means that 38.9% of the total deviance is explained by the difference in the dependent variable between latent classes. Specifically, the separation of the groups explains the deviances as well as the regression models since $NEWD = 0.376$.

The estimation results on the regression coefficients are $\boldsymbol{\beta}_1 = (-0.013, 0.437, -0.333, 0.231, -0.284, 2.495)$ for latent class `g1` and $\boldsymbol{\beta}_2 = (0.041, 0.273, -0.369, 0.155, -0.175, 1.61)$ for `g2`. The parameters of the marginal densities of the covariates can be accessed from the returned results (see section 3.2; the command `ereturn list` can be used to access the complete list of returned results). For multivariate Gaussian covariates `admday`, `age`, and `mortrisk`, the parameters are stored in matrices `e(mu)` and `e(sigma)`, shown in the Stata output below.

```
.
. matlist e(mu), title("mean of multivariate Gaussian covariates ")
mean of multivariate Gaussian covariates
             |        g1         g2
-------------+--------------------
      admday | -.0438391   .0196431
         age | -.1848609   .0828312
    mortrisk |  .1941918  -.0870121
.
. matlist e(sigma),title("variance-covariance matrix of multivariate Gaussian covariates")
variance-covariance matrix of multivariate Gaussian covariates
             |                g1                          |                g2
             |     admday       age     mortrisk          |     admday       age     mortrisk
-------------+----------------------------------------------+----------------------------------
      admday |  .6904834   .1316384    .281698             |  1.135991   .1382855   .1698366
         age |  .1316384   1.484113   1.063809             |  .1382855   .7594605   .6944169
    mortrisk |   .281698   1.063809   1.506549             |  .1698366   .6944169   .7471128
```

The latent class `g1` is characterized by a mean vector equal to $\boldsymbol{\mu}_1 = (-0.044, -0.189, 0.194)$,

the corresponding vector in `g2` is $\boldsymbol{\mu}_2 = (0.0196, 0.082, -0.087)$. This means that in `g1`, patients have been admitted earlier (`admday`), are younger (`age`) but have a higher risk of mortality at the time of admission (`mortrisk`). The variance-covariance matrices highlight that the normal covariates are positively correlated in both latent classes. The latent classes have a rather similar expected value of the only Poisson covariate (`n1`): $\lambda_1 = 3.480$ in `g1` and $\lambda_2 = 3.515$ in `g2`. Concerning sex, females have a larger probability to belong to the first latent class ($p_1 = 0.424$) than to `g2` ($p_2 = 0.295$).

```
. matlist e(lambda), title("mean of Poisson covariates")
mean of Poisson covariates
             |        g1         g2
-------------+--------------------
          n1 |  3.480433    3.51456
.
. matlist e(p_binomial), title("mean of binomial covariates")
mean of binomial covariates
             |        g1         g2
-------------+--------------------
      female |  .4240503    .295099
```

To estimate models with K greater than 2, users may simply edit the `k` option in the command (see section 3.1) with the desired number of latent classes. A summary of the models for $K = 2, 3, 4$ is presented in table 2. Overall, the results suggest that models with larger K values fit the data better than the model with $K = 2$. The AIC and BIC suggest selecting the model with $K = 4$; however, the differences in AIC and BIC are marginal and model selection would be between $K = 3$ and $K = 4$ as they involve a trade-off between regression fit (indicated by R^2) and class separation (summarized by NBD). The CWM with $K = 2$ is discarded.

Table 2: Summary of the CWM applied to the covid dataset

	$K = 2$	$K = 3$	$K = 4$
AIC	19,832.24	19,560.91	19,053.67
BIC	19,994.19	19,801.39	19,372.68
R^2	0.384	0.435	0.573
NEWD	0.234	0.202	0.281
NBD	0.389	0.536	0.510

4.2 The students dataset

The second real dataset is based on a survey of 270 students attending the University of Catania. For each respondent, the variables include information about height (`height`), father's height (`heightf`), weight (`weight`), and gender (`gender`).

Estimation and bootstrapping

In this section, we replicate the model presented in section 5.3 of Mazza et al. (2018) and integrate it with bootstrapping using the postestimation command `cwmbootstrap`. The underlying modeling strategy is to treat gender as an unobserved and evaluate whether the CWM is able to discriminate between females and males. This model specified multivariate normal covariates with equal size, equal shape and equal orientation (EEE) and a Gaussian GLM. The dependent variable is `weight` while the covariates are `height` and `heightf`.

```
. cwmglm weight height heightf,  k(2)  xnormal(height heightf) eee
initializing EM...
(output omitted)
EM iteration        17:log-likelihood=  -2648.16080
```

Prior Probabilities

g1	g2
.4369902	.5630098

Clustering Table

g1	g2
117	153

Information criteria

AIC	BIC
5328.322	5385.896

Local Deviance

	g1	g2	Overall
WD	169.0922	228.9385	398.0306
RWD	116.9874	151.0126	268
EWD	52.1048	77.92581	130.0306
R^2	.3081444	.3403789	.3266849

Global Deviance

	RWD	EWD	BD	TD
Deviance	130.0306	268	204.7742	602.8048
Normalized~e	.2157093	.4445884	.3397023	1

mean vectors of the Gaussian variables

	g1	g2
height	177.5373	161.7553

```
    heightf |   174.1353   175.6054
```

variance matrices of the Gaussian variables

	g1		g2	
	height	heightf	height	heightf
height	27.8441	22.04352	27.8441	22.04352
heightf	22.04352	34.69652	22.04352	34.69652

weight	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
g1						
height	.7612449	.1082026	7.04	0.000	.5491717	.9733182
heightf	-.0088664	.0939404	-0.09	0.925	-.1929862	.1752534
_cons	-57.28365	12.3717	-4.63	0.000	-81.53174	-33.03556
g2						
height	.8983653	.0912865	9.84	0.000	.719447	1.077284
heightf	-.1442846	.0838213	-1.72	0.085	-.3085712	.0200021
_cons	-54.08234	12.12518	-4.46	0.000	-77.84725	-30.31742

Compared to Mazza et al. (2018), the two groups have the same dimensions and the regression coefficients are very close in value. For instance the coefficient of height in the largest group obtained by Mazza et al. (2018) is equal to 0.897714 while the one given by `cwmglm` is equal to 0.8983653. Moreover, for the smallest group, the obtained mean vector is (177.5373 , 174.1353), while Mazza et al. (2018) report (177.54,174.14).

```
.
. predict group, map
(maximum posterior probability group allocation)
. tab group gender
```

group	Gender F	Gender M	Total
1	149	4	153
2	2	115	117
Total	151	119	270

```
.
. tw (scatter height weight if group==1 & gender=="M", mlcolor(gs9) mcolor(gs9) msymbol(T)) ///
>  (scatter height weight if group==2 & gender=="M", mlcolor(gs9) mcolor(gs16) msymbol(T)) ///
>  (scatter height weight if group==1 & gender=="F", mlcolor(gs9) mcolor(gs16) msymbol(O)) ///
>  (scatter height weight if group==2 & gender=="F", mlcolor(gs9) mcolor(gs9) msymbol(O)), ///
>  legend(off) saving(g1,replace)
file g1.gph saved

.
.  tw (scatter height heightf if group==1 & gender=="M", mlcolor(gs9) mcolor(gs9) msymbol(T)) ///
>  (scatter height heightf if group==2 & gender=="M", mlcolor(gs9) mcolor(gs16) msymbol(T)) ///
>  (scatter height heightf if group==1 & gender=="F", mlcolor(gs9) mcolor(gs16) msymbol(O)) ///
>  (scatter height heightf if group==2 & gender=="F", mlcolor(gs9) mcolor(gs9) msymbol(O)), ///
>  legend(off) saving(g2,replace)
```

```
file g2.gph saved
.
.  tw (scatter heightf weight if group==1 & gender=="M", mlcolor(gs9) mcolor(gs9) msymbol(T)) ///
>  (scatter heightf weight if group==2 & gender=="M", mlcolor(gs9) mcolor(gs16) msymbol(T)) ///
>  (scatter heightf weight if group==1 & gender=="F", mlcolor(gs9) mcolor(gs16) msymbol(O)) ///
>  (scatter heightf weight if group==2 & gender=="F", mlcolor(gs9) mcolor(gs9) msymbol(O)), ///
>  legend(off) saving(g3,replace)
file g3.gph saved
.  graph combine g1.gph g2.gph g3.gph, rows(1)
```

In figure 1 triangles represent males and circles represent females; empty markers indicate correctly classified observations, while full markers represent classification errors.

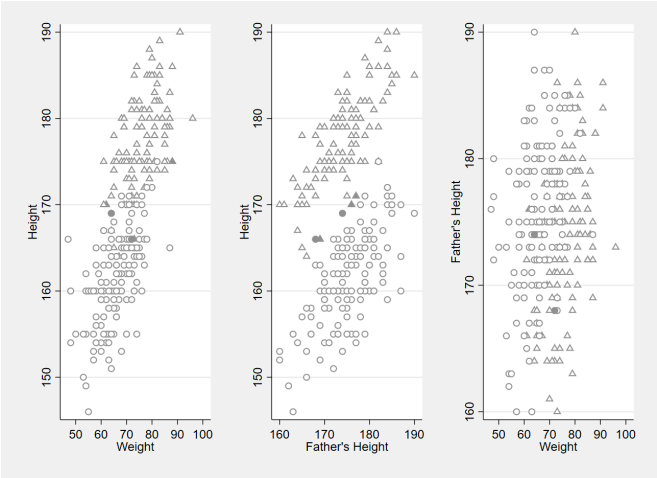

Figure 1: Scatter plot of model variables

For inference purposes, we use the postestimation command `cwmbootstrap` to obtain the standard errors of the estimated parameters. Consistently with standard Stata estimation commands, `cwmbootstrap` displays and returns the inference tables for the estimated parameters. In this example, `cwmbootstrap` returns a matrix `r(b)` containing the inference table for GLM and a matrix `r(mu)` containing bootstrap estimates for the means of the multivariate Gaussian covariates. Our test rejects the null hypothesis that the coefficient of height is zero for both groups, while it fails to reject the one regarding the father's height.

```
. set seed 67788
. cwmbootstrap, reps(100)
Bootstrap replications (100)
..................................................    50
..................................................   100

GLM estimates
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
g1						
height	.7612449	.1857554	4.10	0.000	.397171	1.125319
heightf	-.0088664	.1590014	-0.06	0.956	-.3205035	.3027706
_cons	-57.28365	15.7494	-3.64	0.000	-88.15189	-26.4154
g2						
height	.8983653	.1410497	6.37	0.000	.6219129	1.174818
heightf	-.1442846	.100023	-1.44	0.149	-.3403261	.0517569
_cons	-54.08234	13.39161	-4.04	0.000	-80.3294	-27.83527

Mean of Gaussian covariates (marginal distribution)

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
g1						
height	177.5373	.5050874	351.50	0.000	176.5473	178.5272
heightf	174.1353	.4082369	426.55	0.000	173.3351	174.9354
g2						
height	161.7553	.5161487	313.39	0.000	160.7436	162.7669
heightf	175.6054	.4450325	394.59	0.000	174.7331	176.4776

Comparing CWM and FMR

In this section, we use `cwmcompare` to select the best model among two nested alternatives. The benchmark model (saved as `cwm` using `estimates store`) is the CWM from section 4.2. The competing model (saved as `fmm`) is a finite mixture of Gaussian GLMs with the same response variable as the previous one. The number of latent classes is held fixed at $k = 2$. Since the models maximize different log-likelihoods the AIC and BIC values obtained from `cwmglm` are not directly comparable but must be adjusted. The model `cwm` is the most general, its log-likelihood is as follows:

$$\sum_{i=1}^{N}\sum_{g=1}^{2}\tau_{ig}\left\{\ln\pi_{ig}+\ln\left[\phi(weight_i-\boldsymbol{X}_i\boldsymbol{\beta}_g',\sigma_g^2)\right]+\ln\left[\phi_2(\boldsymbol{X}_i-\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g)\right]\right\} \tag{16}$$

while the model `fmm` maximizes

$$\sum_{i=1}^{N}\sum_{g=1}^{2}\tau_{ig}\left\{\ln\pi_{ig}+\ln\left[\phi(weight_i-\boldsymbol{X}_i\boldsymbol{\beta}_g',\sigma_g^2)\right]\right\} \tag{17}$$

where $\phi_2()$ is the bivariate Gaussian PDF and $\boldsymbol{X}_i = (height_i, heightf_i)$. The information criteria calculated using log-likelihoods from equations (16) and (17) are not comparable. The postestimation command `cwmcompare` adjusts equation (17) using

the expression in equation (16) and the constraints $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$ (i.e., the means and covariance matrices of the covariates are constrained to be group-invariant). The Stata output below reports the results. Note that, in the second call of `cwmglm`, the option `xnormal` have been omitted. This omission allows to estimate the FMR in equation (17) instead of the CWM in equation (16). The order of magnitude of the information criteria is rather different between models (the AIC is 5,328 vs 1,937 in `cwm` and `fmm`) due to the difference in the formula used for the log-likelihood. Selecting the model without adjustment would lead to the (incorrect) selection of model `fmm`. After the adjustment using `cwmcompare` the AIC- and BIC-minimizing model is `cwm`.

```
. cwmglm weight height heightf,  k(2)  xnormal(height heightf) eee  nolog ///
>         nocluster nodev noregt nomarginal
initializing EM...
EM iteration        17:log-likelihood=  -2648.16080
Information criteria
─────────────────────
      AIC        BIC
─────────────────────
 5328.322   5385.896
─────────────────────

. estimates store cwm
.
. cwmglm weight height heightf,  k(2)  eee  nolog nocluster nodev noregt
initializing EM...
EM iteration       855:log-likelihood=   -959.66820
Information criteria
─────────────────────
      AIC        BIC
─────────────────────
 1937.336   1969.722
─────────────────────

. estimates store fmm
. cwmcompare cwm fmm
information criteria for cwmglm estimates
               │        AIC        BIC
───────────────┼──────────────────────
         cwm   │   5328.322   5385.896
         fmm   │   5629.813   5680.191
the model with the minimum AIC is  cwm
the model with the minimum BIC is  cwm
```

4.3 The multinorm dataset

This simulated example illustrates the use of `cwmglm` as a tool for estimating parsimonious mixtures of multivariate normal distributions, as well as how to use loops along with `cwmglm` and the postestimation command `cwmcompare` to automate the model selection process based on AIC and BIC. The dataset is generated with the code below.

```
. clear
. set seed 234567
. matrix C1 = (1351, -358\-358, 136)
. matrix mu1=(59,68)
. drawnorm x1 x2, n(1000) cov(C1) means(mu1)
(obs 1,000)
. tempfile temp
. gen group=1

. preserve
. clear
. matrix C1 = (47, -12\ -12, 378)
. matrix mu1=(8,61)
. drawnorm x1 x2, n(200) cov(C1) means(mu1)
(obs 200)
. gen group=2
. save `temp´, replace
. restore
. append using `temp´
.
. preserve
. clear
. matrix mu1=(124,40)
. matrix C = (7407, 1033\ 1033, 728)
. drawnorm x1 x2, n(720) cov(C1) means(mu1)
(obs 720)
. gen group=3
. save `temp´, replace
. restore
. append using `temp´
. save multinorm,replace
.
. tw (scatter x1 x2 if group==1) ///
> (scatter x1 x2 if group==2) ///
> (scatter x1 x2 if group==3), ///
> legend(order (1 "comp. 1" 2 "comp. 2" 3 "comp. 3")) ///
> legend( rows(1)) title(Artificial Data)
```

The 1,920 observations are labeled by the variable `group`, which identifies three subpopulations (A, B, and C). In this dataset, there are two continuous variables, `admday` and `age`, which are drawn from a multivariate normal distribution. The classes are shown in figure 2. The variables `admday` and `age` have different covariance matrices with variable shape, orientation, and volume (VVV). The characteristic means,

variance-covariance matrices, and sample sizes are as follows:

$$\begin{aligned}\mu_A = (59, 68), \Sigma_A = \left(\begin{smallmatrix} 1351 & -358 \\ -358 & 136 \end{smallmatrix}\right), N_A = 1000 \\ \mu_B = (8, 61), \Sigma_B = \left(\begin{smallmatrix} 47 & -12 \\ -12 & 378 \end{smallmatrix}\right), N_B = 200 \\ \mu_C = (124, 40), \Sigma_C = \left(\begin{smallmatrix} 7407 & 1033 \\ 1033 & 728 \end{smallmatrix}\right), N_C = 720\end{aligned} \tag{18}$$

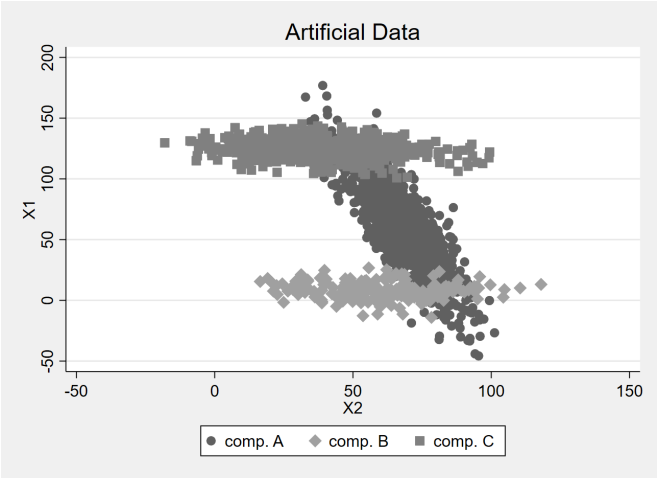

Figure 2: Clusters in the multinorm dataset

In the following code, we run a `foreach` loop on all the possible combinations of parsimonious models and number of components. The outer `foreach` loop cycles over different specifications of the multivariate Gaussian parsimonious models listed in local macro `models` (e.g., VVV,VVE,EEI) and the inner `forvalues` loop cycles over $k = 2, \ldots, 5$. Inside the concatenated loops, convergence is checked and the estimates of the converging model are saved using `estimates store`. Model selection is carried out using `cwmcompare` on the saved estimates.

```
. tw (scatter x1 x2 if group=="A") ///
> (scatter x1 x2 if group=="B") ///
> (scatter x1 x2 if group=="C"), ///
> legend(order (1 "comp. A" 2 "comp. B" 3 "comp. C")) ///
> legend( rows(1)) title(Artificial Data)
.
. graph export fig3.png, as(png) replace
file fig3.png saved as PNG format
. global CWMs //initializing the estimates list
. local models vev evv vvv eei vei evi ///
> vvi eii vii eee vee eve vve eev // list of the 14 parsimonious models
.
. foreach model of local models { // looping over parsimonious models
  2.         forval i=2/5 { // looping over number of clusters
  3.
```

```
.                 qui cwmglm, xnorm(x1 x2) k(`i´) `model´
  4.                 if (e(converged)==1) {
  5.                 estimates store `model´`i´ //saving estimates for converged models
  6.                 global CWMs $CWMs `model´`i´ //updating the estimates list
  7.                 }
  8.                 else di in red ///
>                 "model `model´ with `i´ mixture component did not converge"
  9.         }
 10. }
model evv with 5 mixture component did not converge
model vvv with 4 mixture component did not converge
model eve with 4 mixture component did not converge
model eev with 5 mixture component did not converge

.
. cwmcompare $CWMs
information criteria for cwmglm estimates
```

	AIC	BIC
vev2	37030.31	37085.91
vev3	36057.32	36140.72
vev4	36100.53	36211.73
vev5	35549.92	35688.92
evv2	35457.97	35513.57
evv3	34558.02	34641.42
evv4	35244.29	35355.49
vvv2	35182.28	35243.44
vvv3	34505.37	34599.89
vvv5	36197.52	36358.76
eei2	36137.08	36176.01
eei3	35713.89	35769.49
eei4	36098.1	36170.38
eei5	35960.8	36049.76
vei2	36124.26	36168.74
vei3	35574.39	35641.11
vei4	35619.72	35708.68
vei5	35539.86	35651.06
evi2	35560.15	35604.63
evi3	35200.26	35266.98
evi4	34959.24	35048.2
evi5	35199.14	35310.35
vvi2	35252.24	35302.28
vvi3	35159.25	35237.09
vvi4	35045.69	35151.33
vvi5	35086.71	35220.15
eii2	36219.56	36252.92
eii3	35790.79	35840.83
eii4	36116.7	36183.42
eii5	36056.54	36139.94
vii2	36165.59	36204.51
vii3	35579.66	35640.82
vii4	35635.12	35718.52
vii5	35326.92	35432.56
eee2	36012.86	36057.34
eee3	35608.79	35669.95
eee4	35969.97	36047.81
eee5	35862.86	35957.38
vee2	36000.59	36050.63
vee3	35495.59	35567.87
vee4	35549.8	35644.32

```
        vee5 |  35221.59   35338.35
        eve2 |  35540.84   35590.88
        eve3 |  34889.31   34961.59
        eve5 |  35855.42   35972.18
        vve2 |  35267.77   35323.37
        vve3 |  34869.07   34952.47
        vve4 |  35158.68   35269.88
        vve5 |  35832.35   35971.35
        eev2 |  35451.93   35501.97
        eev3 |  34724.96   34797.24
        eev4 |  35160.93   35255.45
the model with the minimum AIC is  vvv3
the model with the minimum BIC is  vvv3
```

As displayed in the output above, the model that minimizes both the AIC and the BIC is characterized by three latent classes; the variance covariance matrix is characterized by variable orientation, variable volume, and variable shape (VVV), which is the most general type of model. As can be observed in figure 3 and from the Stata output of the `tab` command, the mixture of parsimonious multivariate normal models predict the classes well. Specifically, the MAP classes obtained through `predict` correctly classifies 1,762 observations out of 1,920 (92%). Latent class A largely overlaps with estimated latent class `g3`, B is assigned to `g2`, and C is assigned to `g1` (output of `tab map group`). The means (`e(mu)`) and variance-covariance (`e(sigma)`) matrices are very similar to the data-generating ones in values and order of magnitude. For instance, group `g3` is characterized by an estimated vector of means $\hat{\boldsymbol{\mu}}_3 = (60.279, 67.529)$ which is remarkably similar to the corresponding vector from the data-generating process $\boldsymbol{\mu}_A = (59, 68)$.

```
. ***** activating the estimates from the best model
. estimates restore `r(bestAIC)´
(results vvv3 are active now)
. matlist e(mu)
             |        g1         g2         g3
-------------+------------------------------------
          x1 |  8.044332   124.2815   60.27864
          x2 |  60.91576   39.82697   67.52907
. matlist e(sigma)
             |        g1                  |        g2                  |        g3
             |        x1         x2       |        x1         x2       |        x1         x2
-------------+----------------------------+----------------------------+----------------------------
          x1 |  50.51839  -8.000548       |  49.85068  -23.94966       |  1299.013  -341.5892
          x2 | -8.000548    399.377       | -23.94966   379.8781       | -341.5892   131.7972
. predict _tau, posterior
(posterior probabilities)
. predict map, map
(maximum posterior probability group allocation)
.
. tw (hist x1) (kdensity x1 [aw=_tau1]) (kdensity x1 [aw=_tau2] ///
> ) (kdensity x1 [aw=_tau3]), ///
> legend(rows(1)) ///
> legend(order(1 "Observed PDF" 2 "comp.1" 3 "comp.2" 4 "comp.3")) ///
> saving(gg1,replace) title(x1)
file gg1.gph saved
```

```
.
. tw (hist x2) (kdensity x2 [aw=_tau1]) (kdensity x2 [aw=_tau2]) (kdensity x2 [aw=_tau3]), ///
> legend(rows(1)) ///
> legend(order(1 "Observed PDF" 2 "comp.1" 3 "comp.2" 4 "comp.3")) ///
> saving(gg2,replace) title(x2)
file gg2.gph saved

.
. tab map group
```

map	Cluster identifier A	B	C	Total
1	15	160	0	175
2	78	0	701	779
3	907	40	19	966
Total	1,000	200	720	1,920

```
. tw (scatter x1 x2 if map==1) (scatter x1 x2 if map==2) ///
> (scatter x1 x2 if map==3), ///
> legend(order (1 "CWM Comp. 1" 2 "CWM Comp. 2" 3 "CWM Comp. 3")) ///
> legend( rows(1)) title(Artificial Data) subtitle(Estimated components)
. graph combine gg1.gph gg2.gph , rows(1) ycommon
```

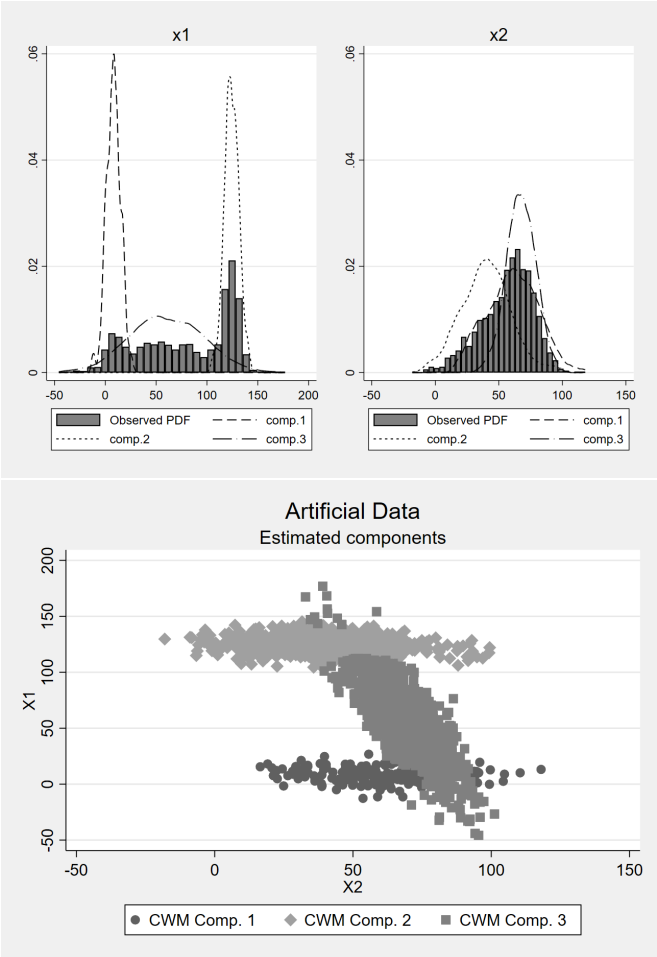

Figure 3: Estimated classes in the multinorm dataset

5 Conclusion

In this article, we introduced `cwmglm`, a new Stata command that supports CWMs, general class of mixture models that includes FMRs and mixture of distributions. Models for the marginal and conditional densities are based the most common distributions used in GLMs. A possible development of `cwmglm` would be its extension to other less common models to be estimated within each latent class such as those related to the mixtures of Student-t distributions or to regressions with multivariate response variables; this is left to future research.

6 References

Aitken, A. C. 1927. XXV.—On Bernoulli's Numerical Solution of Algebraic Equations. *Proceedings of the Royal Society of Edinburgh* 46: 289–305.

Banfield, J. D., and A. E. Raftery. 1993. Model-based Gaussian and non-Gaussian clustering. *Biometrics* 803–821.

Berta, P., S. Ingrassia, A. Punzo, and G. Vittadini. 2016. Multilevel cluster-weighted models for the evaluation of hospitals. *Metron* 74(3): 275–292.

Berta, P., S. Ingrassia, G. Vittadini, and D. Spinelli. 2024. Latent heterogeneity in COVID-19 hospitalisations: a cluster-weighted approach to analyse mortality. *Australian & New Zealand Journal of Statistics* .

Berta, P., and V. Vinciotti. 2019. Multilevel Logistic Cluster-Weighted Model for Outcome Evaluation in Health Care. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 12(5): 434–443.

Brilleman, S. 2011. DEVR2: Stata module to compute Cameron and Windmeijer's deviance based R-squared measure. https://EconPapers.repec.org/RePEc:boc:bocode:s457340.

Browne, R. P., and P. D. McNicholas. 2014. Estimating Common Principal Components in High Dimensions. *Advances in Data Analysis and Classification* 8(2): 217–226.

Cameron, A. C., and F. A. Windmeijer. 1996. R-squared Measures for Count Data Regression Models With Applications to Health-Care Utilization. *Journal of Business & Economic Statistics* 14(2): 209–220.

———. 1997. An R-Squared Measure of Goodness of Fit for Some Common Nonlinear Regression Models. *Journal of econometrics* 77(2): 329–342.

Celeux, G., and G. Govaert. 1995. Gaussian Parsimonious Clustering Models. *Pattern recognition* 28(5): 781–793.

Dayton, C. M., and G. B. Macready. 1988. Concomitant-Variable Latent-Class Models. *Journal of the American Statistical Association* 83(401): 173–178.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: B (Methodological)* 39(1): 1–22.

Di Mari, R., S. Ingrassia, and A. Punzo. 2023. Local and Overall Deviance R-Squared Measures for Mixtures of Generalized Linear Models. *Journal of Classification* 40(2): 233–266.

Diani, C., G. Galimberti, and G. Soffritti. 2022. Multivariate Cluster-Weighted Models Based On Seemingly Unrelated Linear Regression. *Computational Statistics & Data Analysis* 171: 107451.

Fabbian, F., A. De Giorgi, E. Maietti, M. Gallerani, M. Pala, R. Cappadona, R. Manfredini, and U. Fedeli. 2017. A Modified Elixhauser Score for Predicting In-Hospital Mortality in Internal Medicine Admissions. *European Journal of Internal Medicine* 40: 37–42.

Früwirth-Schnatter, S. 2005. *Finite Mixture and Markov Switching Models*. Heidelberg: Springer.

Gershenfeld, N. 1997. Nonlinear Inference and Cluster-Weighted Modeling. *Annals of the New York Academy of Sciences* 808(1): 18–24.

———. 1999. *The Nature of Mathematical Modelling*. Cambridge: Cambridge University Press.

Gershenfeld, N., B. Schöner, and E. Metois. 1999. Cluster-Weighted Modelling for Time-Series Analysis. *Nature* 397: 329–332.

Gray, L. A., and M. H. Alava. 2018. A command for fitting mixture regression models for bounded dependent variables using the beta distribution. *The Stata Journal* 18(1): 51–75.

Hennig, C. 2000. Identifiablity of Models for Clusterwise Linear Regression. *Journal of classification* 17(2).

Hernández Alava, M., and A. Wailoo. 2015. Fitting adjusted limited dependent variable mixture models to EQ-5D. *Stata Journal* 15(3): 737–750.

Huismans, J., J. W. Nijenhuis, and A. Sirchenko. 2022. A mixture of ordered probit models with endogenous switching between two latent classes. *The Stata Journal* 22(3): 557–596.

Ingrassia, S., S. Minotti, and G. Vittadini. 2012. Local Statistical Modeling via the Cluster-Weighted Approach with Elliptical Distributions. *Journal of Classification* 29(3): 363–401.

Ingrassia, S., and A. Punzo. 2020. Cluster Validation for Mixtures of Regressions Via the Total Sum of Squares Decomposition. *Journal of Classification* 37(2): 526–547.

Ingrassia, S., A. Punzo, G. Vittadini, and S. C. Minotti. 2015. The Generalized Linear Mixed Cluster-Weighted Model. *Journal of Classification* 32(1): 85–113.

Jenkins, S. P., and F. Rios-Avila. 2023. Finite mixture models for linked survey and administrative data: Estimation and postestimation. *The Stata Journal* 23(1): 53–85.

Mazza, A., A. Punzo, and S. Ingrassia. 2018. `flexCWM`: a Flexible Framework for Cluster-Weighted Models. *Journal of Statistical Software* 86(2): 1–30.

McCullagh, P., and J. A. Nelder. 2019. *Generalized linear models*. Routledge.

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

McNicholas, P. D. 2016. Model-based clustering. *Journal of Classification* 33: 331–373.

Punzo, A. 2014. Flexible mixture modelling with the polynomial Gaussian cluster-weighted model. *Statistical Modelling* 14(3): 257–291.

Sarkar, S., X. Zhu, V. Melnykov, and S. Ingrassia. 2020. On Parsimonious Models for Modeling Matrix Data. *Computational Statistics & Data Analysis* 142: 106822.

Schöner, B. 2000. Probabilistic Characterization and Synthesis of Complex Data Driven Systems. Technical report, Ph.D. Thesis, MIT.

Soffritti, G. 2021. Estimating the Covariance Matrix of the Maximum Likelihood Estimator Under Linear Cluster-Weighted Models. *Journal of Classification* 38(3): 594–625.

Subedi, S., A. Punzo, S. Ingrassia, and P. D. McNicholas. 2013. Clustering and Classification via Cluster-Weighted Factor Analyzers. *Advances in Data Analysis and Classification* 7(1): 5–40.

Wedel, M. 2002. Concomitant Variables in Finite Mixture Models. *Statistica Neerlandica* 56(3): 362–375.

Acknowledgements

Earlier versions of this package were presented at the 2022 Stata Conference in Florence (Italy), at the 2023 Royal Statistical Society congress in Harrogate (United Kingdom) and at the 2023 meeting of Classification and Data Analysis Group (ClaDAG) of the Italian Statistical Society held in Salerno, Italy. We are grateful to Piergiorgio Lovaglio, to the editor and to an anonymous referee. Salvatore Ingrassia acknowledges financial support from PNRR MUR project PE0000013-FAIR.

About the authors

Daniele Spinelli is a Research Fellow in Statistics at the University of Milano-Bicocca
Salvatore Ingrassia is a Full Professor of Statistics at the University of Catania
Giorgio Vittadini is a Full Professor of Statistics at the University of Milano-Bicocca