Scale–constrained approaches for maximum likelihood estimation and model selection of clusterwise linear regression models

Roberto Di Mari*[1] and Roberto Rocci§[2] Stefano Antonio Gattone‖[3]

[1]Department of Economics and Business, University of Catania, Italy
[2]Department of Economics and Finance, University of Rome Tor Vergata, Italy
[3]Department of Philosophical and Social Sciences, Economics and Quantitative Methods, University G. d'Annunzio, Chieti-Pescara, Italy

May 20, 2019

*roberto.dimari@unict.it
§roberto.rocci@uniroma2.it
‖gattone@unich.it

Abstract

We consider an equivariant approach imposing data-driven bounds for the variances to avoid singular and spurious solutions in maximum likelihood (ML) estimation of clusterwise linear regression models. We investigate its use in the choice of the number of components and we propose a computational shortcut, which significantly reduces the computational time needed to tune the bounds on the data. In the simulation study and the two real-data applications, we show that the proposed methods guarantee a reliable assessment of the number of components compared to standard unconstrained methods, together with accurate model parameters estimation and cluster recovery.

Key words: clusterwise linear regression, mixtures of linear regression models, data-driven constraints, equivariant estimators, computationally efficient approach, model selection.

1 Introduction

In many applications within the various fields of social and physical sciences, investigating the relationship between a response variable and a set of explanatory variables is commonly of interest. Yet, the estimation of a single set of regression coefficients for all sample observations is often inadequate. To the purpose, finite mixture of conditional normal distributions can be used to estimate clusterwise regression parameters in a maximum likelihood context. Clusterwise linear regression is also known under the names of finite mixture of linear regressions or switching regressions (Alfó and Viviani, 2016; Quandt, 1972; Quandt and Ramsey, 1978; Kiefer, 1978).

Let $\mathrm{y}_1, \ldots, \mathrm{y}_n$ be a sample of independent observations drawn from the response random variable Y_i, each respectively observed conditionally on a vector of J regressors $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Let us assume $\mathrm{Y}_i|\mathbf{x}_i$ to be distributed as a finite mixture of linear regression models, that is

$$f(\mathrm{y}_i|\mathbf{x}_i;\boldsymbol{\psi}) = \sum_{g=1}^{G} p_g \phi_g(\mathrm{y}_i|\mathbf{x}_i, \sigma_g^2, \boldsymbol{\beta}_g) = \sum_{g=1}^{G} p_g \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp\left[-\frac{(\mathrm{y}_i - \mathbf{x}_i'\boldsymbol{\beta}_g)^2}{2\sigma_g^2} \right], \tag{1}$$

where G is the total number of clusters and p_g, $\boldsymbol{\beta}_g$, and σ_g^2 are respectively the mixing proportion, the vector of $J+1$ regression coefficients including an intercept, and the variance term for the g-th cluster. The set of all model parameters to be estimated is given by

$$\boldsymbol{\psi} = \{(p_1, \ldots, p_G, \boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_G', \sigma_1^2, \ldots, \sigma_G^2)' \in \mathbb{R}^{G+(J+1)G+G} : p_1 + \cdots + p_G = 1, p_g > 0, \sigma_g^2 > 0$$

for $g = 1, \ldots, G\}$.

The likelihood function can be formulated as

$$\mathscr{L}(\boldsymbol{\psi}) = \prod_{i=1}^{n} \left\{ \sum_{g=1}^{G} p_g \frac{1}{\sqrt{2\pi {\sigma_g}^2}} \exp\left[-\frac{(\mathrm{y}_i - \mathbf{x}_i'\boldsymbol{\beta}_g)^2}{2\sigma_g^2} \right] \right\}, \tag{2}$$

which is maximized in order to estimate $\boldsymbol{\psi}$. Alternatively to direct maximization, the EM algorithm (Dempster, Laird, and Rubin, 1977) is frequently used.

A well-known complication in ML estimation of mixtures of (conditional) normals with cluster-specific variances is that the likelihood function is unbounded (Kiefer and Wolfowitz, 1956; Day, 1969). This can be seen by noting that the likelihood function goes to infinity as one mixture's

variance tends to zero and one of the sample observations has a zero residual on the corresponding component. This has two practical consequences: degeneracy of optimization algorithms and occurrence of spurious solutions. These problems have been tackled by a large number of authors and many different solutions have been proposed. A comprehensive review on the topic when estimating mixtures of elliptical distributions can be found in García-Escudero et al. (2017). See also Ritter (2014), and Rocci et al, (2018).

A classical solution of the afore mentioned problems is based on the seminal work of Hathaway (1985) which, in order to have the likelihood function of univariate mixtures of normals bounded, suggested to impose a lower bound, say c, to the ratios of the scale parameters in the maximization step. The method is equivariant under linear affine transformations of the data. That is, if the data are linearly transformed, the estimated posterior probabilities do not change and the clustering remains unaltered. Nevertheless, Hathaway's constraints are very difficult to apply within iterative procedures like the EM algorithm. In addition, how to properly choose c, which controls the strength of the constraints, is an open issue.

Clusterwise linear regression modeling is tightly linked to univariate normal mixture modeling as the response variable is (conditional on observed covariates) univariate. Recently in this context, Di Mari et al. (2017) imposed constraints on the variances of the regression error terms that are tuned on the data based on a cross-validation strategy (RGD method), hence not requiring any prior knowledge of the mixture scale balance. These constraints provide a sufficient condition for Hathaway's constraints to hold. For the extension of the method to the multivariate case see Rocci et al. (2018).

The term "spurious solutions" - i.e. non meaningful local maximizers - is widely used in the mixture modeling literature: although the concept is widely understood, we lack a rigorous characterization. To each attempt of giving a characterization corresponds a strategy on how to detect/avoid the spurious solutions. One possible path is monitoring the local maximizers in order to discard those solutions corresponding to a mixture component characterized by a small number of points and a relatively small variance (Day, 1969 and McLachlan and Peel, 2000). Seo and Kim (2012) point out that a spurious solution is typically driven by a random localized pattern of a few observations in the data. Such observations are overfitted by one component of the mixture, heavily affecting the formation of the likelihood-based solutions. They suggest to take out such, say, k observations with the highest likelihood (likelihood-based k-deleted method) and select the

solution with the highest k-deleted likelihood. A score-based k-deleted method was also proposed.

The issue of unboundedness is not only an estimation problem, but makes one of the most complicated tasks in cluster analysis, i.e. selecting the number of components, more complicated. It is common to select the number of components by means of likelihood–based information criteria, like AIC or BIC. With degenerate and spurious solutions, unduly high likelihood values are very likely to result in distorted assessments of the number of clusters.

The contribution of the present work is twofold. First, starting from a conjecture by Di Mari et al (2017), we investigate the use of the RGD method to *regularize* the log-likelihood used for the Bayesian Information Criterion. We propose to evaluate the BIC at the constrained solution as provided by the RGD approach, and we test how accurately the number of clusters are recovered over a wide set of simulation scenarios and two empirical applications. The results from our simulation study demonstrate that model selection based on a *regularized* likelihood guarantees a reliable assessment of the number of mixture components. Nevertheless, this comes at the price of a high computational cost due to the way the tuning constant is chosen. Which, all else equal, makes model selection an especially daunting task whenever, as is typically done in applied research, a suitable model candidate is to be chosen among many alternatives.

As second contribution, we present a new and computationally faster approach for selecting c based on the data, given G. The approach is designed with the aim of sensibly reducing the running time tuning procedures like cross–validation typically need. The proposal draws ideas from the k-deleted method of Seo and Kim (2012) as a root–selection approach. Within this framework a spurious solution is then characterized by the presence of overfitted observations with very small variance ratio. The resulting *accelerated* tuning approach, by means of an extensive simulation study and two real-data applications, is showed to be 1) very well suited for scale balance tuning, and to 2) do up to 10 time faster than the baseline cross-validation based approach, with no price to be paid in terms of loss of accuracy in parameters estimation and cluster recovery. In addition, whenever G is not known, we demonstrate that it can be used for reliable model selection.

The remainder of the paper is organized as follows. In Section 2, we briefly review the constrained RGD method for clusterwise regression modeling and the cross–validation strategy to tune the constraints. Section 3 describes how to carry out model selection with BIC based on the constrained estimator, and Section 4 introduces the computationally efficient alternative to the cross–validation strategy. Details on the ML estimation are given in Section 5. The proposed

methodologies are illustrated - and their performance evaluated - with a simulation study (Section 6), and two real-data examples (Section 7). Section 8 concludes with a final discussion and some ideas for future research

2 The RGD method

For univariate Gaussian mixtures, Hathaway (1985) proposed to maximize the log-likelihood under constraints of the kind

$$\min_{i \neq j} \frac{\sigma_i^2}{\sigma_j^2} \geq c \quad \text{with} \quad c \in (0, 1]. \tag{3}$$

Hathaway's approach presents a strongly consistent global solution, no singularities, and a smaller number of spurious maxima. However, there is no easy way to implement the constraints into a feasible algorithm.

For ML estimation of the clusterwise linear regression model in Equation (1), Di Mari et al. (2017) proposed relative constraints on the group conditional variances σ_g^2 of the kind

$$\sqrt{c} \leq \frac{\sigma_g^2}{\bar{\sigma}^2} \leq \frac{1}{\sqrt{c}}, \tag{4}$$

or equivalently

$$\bar{\sigma}^2 \sqrt{c} \leq \sigma_g^2 \leq \bar{\sigma}^2 \frac{1}{\sqrt{c}}. \tag{5}$$

The above constraints have the effect of shrinking the variances to a suitably chosen $\bar{\boldsymbol{\sigma}}^2$, the *target* variance term, and the level of shrinkage is given by the value of c. Easily implementable within the EM algorithm (Ingrassia, 2004; Ingrassia and Rocci, 2007), the constraints in (5) provide a sufficient condition for Hathaway's constraints - Equation (3) - to hold. This can be seen by noting that

$$\frac{\sigma_g^2}{\sigma_j^2} = \frac{\sigma_g^2/\bar{\sigma}^2}{\sigma_j^2/\bar{\sigma}^2} \geq \frac{\sqrt{c}}{1/\sqrt{c}} = c.$$

This type of constraints ensures the method to be equivariant (Di Mari et al., 2017) i.e. if the dependent variable is rescaled, the linear predictor and the error's standard deviations are both on the new response scale provided that the *target* variance $\bar{\boldsymbol{\sigma}}^2$ is rescaled accordingly. This is true if $\bar{\boldsymbol{\sigma}}^2$ is estimated from the data with a method itself equivariant. For example the variance of the homoscedastic model as well as, for instance, the OLS residual variance. Perhaps most importantly

for clustering, this leaves the estimates of the posterior probabilities unaltered, hence guaranteeing a final partition which does not depend on any previous data transformation or standardization.

A sensible choice of the tuning parameter c is needed. Selecting c jointly with the mixture parameters by maximizing the likelihood on the entire sample would trivially yield an overfitted scale balance approaching zero. Di Mari et al. (2017) proposed, for constrained estimation of clusterwise linear regression, a cross-validation strategy in order to let the data decide the optimal scale balance. The resulting scale balance can be seen as the most appropriate-to-the-data compromise between the heteroscedastic model (for $c \to 0$, $\hat{\boldsymbol{\sigma}}_g^2$ equals the unconstrained ML estimate) and the homoscedastic model (when $c = 1$, $\hat{\boldsymbol{\sigma}}_g^2 = \bar{\boldsymbol{\sigma}}^2$). In particular, they consider partitioning M times the data set $\{(\mathrm{y}_i, \mathbf{x}_i)\}_n$ at random into a training set S_m of size n_S and a test set $\bar{S}_m$ of size $n_{\bar{S}}$, where $n_S + n_{\bar{S}} = n$. For the m-th partition, let $\widehat{\boldsymbol{\psi}}(c, S_m)$ be the constrained ML estimator based on the training set and $\ell_{\bar{S}_m}(\widehat{\boldsymbol{\psi}}(c, S_m))$ be the log-likelihood function of the test set evaluated at $\widehat{\boldsymbol{\psi}}(c, S_m)$. The cross-validated log-likelihood is

$$\mathrm{CV}(c) = \sum_{m=1}^{M} \ell_{\bar{S}_m}(\widehat{\boldsymbol{\psi}}(c, S_m)), \tag{6}$$

which is the sum of the contribution of each test set to the log-likelihood. The optimal c is found as the maximizer of the function in Equation (6).

The maximization of the cross-validated log-likelihood corresponds to the minimization of an unbiased estimator of the Kullback-Leibler divergence between the *truth* and the model under consideration (Smyth, 1996; 2000). The logic behind its use is that it can be seen as function of c only, and maximizing it handles the issue of overfitting as training and test sets are independent (Arlot and Celisse, 2000). The method has shown great promise in terms of quality of model parameters estimation (Di Mari et al., 2017); in the next Section, we propose its use for selecting the number of components.

3 Regularized BIC for model selection

Likelihood–based information criteria, like the AIC and the BIC, are widely used to select the number of mixture components in model–based clustering. Leroux (1992) showed that neither of the two consistently underestimates the number of mixture components. Further studies showed

that, whereby AIC tends to overestimate the number of components (Koehler & Murphree, 1987), BIC consistently estimates it (Keribin, 2000). The BIC has two ingredients: the (negative) maximized mixture loglikelihood taking into account the overall fit of the model to the data, and a penalty term measuring model complexity and sample size. Standard BIC has the form:

$$\text{BIC} = -2\log\mathscr{L}(\widehat{\boldsymbol{\psi}}) + \eta\log(n), \tag{7}$$

where $\eta = \underbrace{J+1}_{\text{regression coeff}} + \underbrace{G}_{\text{scales}} + \underbrace{G-1}_{\text{mixing proportions}}$ represents the number of free parameters to be estimated, and measures model complexity. It is self–evident that $\widehat{\boldsymbol{\psi}}$ computed by using the unbounded likelihood could correspond to a degenerate or spurious solution, making BIC unreliable.

The constrained estimator eliminates degeneracy and reduces the number of spurious solutions (Hathaway, 1985), as the likelihood surface is regularized. How well the regularization is done depends on how the bounds are tuned: with an optimal data–driven selection strategy, we claim that the RGD approach can be used to compute the BIC for a sounder assessment of the number of components as the chance of overfitted solutions is greatly reduced. The BIC, computed at the constrained solution, is as follows:

$$\text{BIC} = -2\log\mathscr{L}(\widehat{\boldsymbol{\psi}}_c) + \eta\log(n). \tag{8}$$

Similarly, Fraley and Raftery (2007) proposed to select the number of components by evaluating the BIC at the maximum a posteriori estimate regularizing the likelihood by adding some prior distributions on the variances.

Notice that, in the BIC of Equation (8), whatever the value of c, η is fixed. In fact, different values of c should correspond to different model complexity levels. Consider the case of c close to 1: the component variances are constrained to be similar to the target variance. In other words, much of their final estimated values comes from the target variance. In the opposite situation, a value of c close to zero allows the component variances to (almost freely) vary. Based on similar considerations, Cerioli et al (2017)'s proposal amounts, in a clusterwise linear regression context, to measure the effective complexity due to the scales as the fraction $(1-c)\times G$, yielding the following modified BIC

$$\text{BIC}_{\text{mod}} = -2 \log \mathscr{L}(\widehat{\boldsymbol{\psi}}_c) + \eta^* \log(n), \tag{9}$$

where $\eta^* = \underbrace{J+1}_{\text{regression coeff}} + \underbrace{(1-c) \times G}_{\text{free scales}} + \underbrace{G-1}_{\text{mixing proportions}}$. Note that, with $c = 1$, the component variances are shrunk towards the target which is taken as input, so no scale is actually estimated.

In the simulation study and the empirical applications, we will illustrate both model selection criteria of Equations (8) and (9) under different scenarios.

4 A computationally efficient constrained approach

In this Section, we first sketch the k-deleted method of Seo and Lindsay (2010), Seo and Kim (2012), and Kim and Seo (2014) in its naïve formulation, i.e. the likelihood-based k-deleted method. Then, starting from their baseline idea, we propose a new, computationally faster, data driven method to tune c.

4.1 The likelihood-based k-deleted method

Singular or spurious solutions are characterized by one or a few observations having overly large log-likelihood terms compared to the rest of the sample. In such cases, these sample points end up dominating the overall log-likelihood. In order to identify such k dominating observations, Seo and Kim (2012) suggested to use the individual log-likelihood terms, and then define the so called k-deleted log-likelihood as follows

$$\ell_{-k}(\boldsymbol{\psi}) = \log \mathscr{L}(\boldsymbol{\psi}) - \sum_{d=1}^{k} \log f(\mathrm{y}_{(n-d+1)}; \boldsymbol{\psi}) \tag{10}$$

where $f(\mathrm{y}_{(1)}; \boldsymbol{\psi}) \leq \cdots \leq f(\mathrm{y}_{(n)}; \boldsymbol{\psi})$ are the ordered values of the individual likelihood terms evaluated at $\boldsymbol{\psi}$. Given the set of local maximizers $\boldsymbol{\Psi} = \{\widehat{\boldsymbol{\psi}}^{(s)}; \quad s = 1, \ldots, S\}$ previously found, the k-deleted log-likelihood is used as a criterion to select the root such that

$$\widehat{\boldsymbol{\psi}}_{-k} = \arg\max_{\boldsymbol{\psi} \in \boldsymbol{\Psi}} \{\ell_{-k}(\boldsymbol{\psi})\}. \tag{11}$$

In words, the very appealing feature of the (likelihood-based) k-deleted method is that it selects a solution among the ones already computed. The quantity in Equation (10) represents how well the rest of the data are fitted after one removes the possible effect of overfitting a single or a few observations (Seo and Lindsay, 2010). On the other hand, whether effectively the method discards the spurious solutions in favor of the *correct* one depends on actually computing the largest number of solutions possible. Exploring complicated likelihood surfaces will hence require well refined initialization strategies - possibly consisting of large sets of different starts.

4.2 An efficient RGD approach

For a given value of the tuning parameter c, let $\widehat{\boldsymbol{\psi}}_c$ be the maximizer found maximizing (2) subject to (5). The constant c is selected as follows

$$c = \underset{0<c\leq 1}{\arg\max}\{\ell_{-k}(\widehat{\boldsymbol{\psi}}_c)\}. \tag{12}$$

In words, c is chosen by maximizing (10) with respect to (c, ψ)., The negative term in Equation (10) can be thought as a sort of penalty for spurious solutions. This term also eliminates the overfitting - in the same spirit as in Seo and Lindsay (2010), where it was used for selecting the bandwidth of their smoothed ML estimator.

In addition, by implicitly selecting a maximizer for the constrained ML problem among constrained solutions, the method we propose does not strongly depend on the initialization strategies employed as the number of spurious maximizers is already reduced in a constrained setup (Hathaway, 1985). Stating it differently, it applies a root selection approach - the k-deleted method - to a setup which already guarantees a smaller number of solutions to the ML problem.

5 ML estimation

Once a data–driven choice of c is available, the RGD method requires a target variance as input. The most natural candidate, as argued in Di Mari et al. (2017), is the homoscedastic normal variance as, for $c = 1$, the RGD method's output is then simply the homoscedastic model. Notice that, if the target was chosen from another equivariant method, the RGD approach with scale balance equal to one would be estimating a common-variance model, with common variance equal

Algorithm	Description	Component-scales
HomN	Homoscedastic Normal	$\sigma_g^2 = \sigma^2$
HetN	Heteroscedastic Normal	$\sigma_g^2 \geq 0$
ConC	Constrained algorithm: c selected by crossvalidation	$\bar{\sigma}^2\sqrt{c} \leq \sigma_g^2 \leq \bar{\sigma}^2\frac{1}{\sqrt{c}}$
ConK	Constrained algorithm: c selected with the likelihood-based k-deleted method	$\bar{\sigma}^2\sqrt{c} \leq \sigma_g^2 \leq \bar{\sigma}^2\frac{1}{\sqrt{c}}$

Table 1: Algorithms used in the simulation study

to the target and the other model parameters estimated at their conditional ML values.

ML parameter estimation, after the data–driven selection step, is done by means of Ingrassia and Rocci (2007)'s constrained EM (for details on the steps for clusterwise linear regression, see Di Mari et al, 2017).

6 Numerical study

6.1 Design

The algorithms compared in the numerical study are listed in table 1. The purpose of this simulation study is to address the following issues:

- how sensitive the efficient RGD approach (ConK) is to different choices of k;
- how ConK compares with the standard RGD method (ConC), and with the homoscedastic (HomN) and heteroscedastic (HetN) models;
- how the choice of the target impacts the overall accuracy of the proposed methodologies;
- how the two reformulations of the BIC (Equations (8) and (9)) perform under different scenarios, and how reliably they allow selecting the number of components compared to BIC computed at HomN and HetN solutions.

Concerning the choice of k (ConK approach), Seo and Kim (2012), for P-variate Gaussian mixtures, suggest choosing it between P - where only one component is degenerate (or spurious)

in all P dimensions - and $P \times (G-1)$ - where $G-1$ components are degenerate in all P dimensions. In the first part of the simulation study, we assess ConK in terms of accuracy of parameter estimates (MSE of regression coefficients and component variances) and cluster recovery (adjusted *Rand* index, Adj-Rand, of Hubert and Arabie, 1985) for $k = \{1, 2, (J+1) \times (G-1), n/10, n/5, n/2, n/1.25, n/1.11\}$, where $J+1$ is the number of regressors (including the intercept). We expect very similar results for the different k's as the k-deleted method we implement in this paper is not a root selection method, but rather implemented to select a tuning parameter - similarly to what is done in Seo and Lindsay (2010) for bandwidth selection in their smoothed ML estimator. However note that we also test for values of k very large compared to the sample size (e.g. $k = n/1.11$) to exclude the possibility that any $k \leq n$ works.

In the second part of our simulation study, the performance of ConK is compared with: 1) ConC, 2) the unconstrained algorithm with common (homoscedastic) component-scales (HomN), and 3) the unconstrained algorithm with different (heteroscedastic) component-scales (HetN). By taking the number of components as known, we evaluate the computation time, the accuracy of the model parameter estimates and of cluster recovery for all 4 methods. The target measures used for the comparisons are MSE of the regression coefficients (averaged across regressors and groups) and of the component variances (averaged across groups) for estimation accuracy, and the adjusted *Rand* index for cluster recovery.

In the third part of our simulation study, we investigated what are the effects of changing the target variance on overall estimation results. The alternative target we used for comparison is the Ordinary Least Squares residuals variance $\hat{\sigma}^2_{\text{OLS}} = \frac{1}{n-J-1} \sum_{i=1}^{n} (\mathrm{y}_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}_{\text{OLS}})^2$.

Lastly, we take the number of components as unknown, and let each method select G between 1 and $G^* + 2$ (where G^* is the true number of groups) as the one for which the BIC is the lowest. For the constrained methods ConC and ConK, we do this exercise using the BIC reformulations of Equations (8) and (9).

The data were generated from a clusterwise linear regression with 3 regressors and intercept, with 2, 3, and 4 components and sample sizes of 100 and 200. The class proportions considered were, respectively, $(0.5, 0.5)'$ and $(0.2, 0.8)'$, and $(0.2, 0.4, 0.4)'$ and $(0.2, 0.3, 0.5)'$, and $(0.25, 0.25, 0.25, 0.25)'$ and $(0.1, 0.2, 0.3, 0.4)'$. Regressors have been drawn from 3 independent standard normals, whereas regression coefficients have been drawn from U$(-1.5, 1.5)$ and intercepts are $(4, 9)'$, $(4, 9, 16)'$, and $(4, 9, 16, 25)'$ for the 2-component, 3-component and 4-component

models respectively. The component variances have been drawn from Inv-Gamma$(3, 1)$.

For each of the 12 combinations sample size $\times$ mixing proportion, we generated 250 samples: for each sample and each algorithm - HomN, HetN, ConC, and ConK - we select the best solution (highest likelihood) out of 10 random starts*. For a single run of the RGD algorithm a unique starting partition is used. In particular, the constant c is chosen by profiling the log-likelihood with respect to ψ. The maximizations are done by using the same starting partition as obtained from the constrained maximization of the log-likelihood for a pre-specified $c = 0.1$. An alternative, but computationally cumbersome, option would be to use different initializations to profile the log-likelihood. In our experience, this usually adds little in terms of accuracy of the final estimates given the high computation cost.

6.2 Results

6.2.1 Sensitivity analysis: impact of k in the k-deleted likelihood approach

Tables 2 and 3 show results of the ConK algorithm for $n = 100$ and $n = 200$ respectively, for 8 different values of k. Whereby we observe some variation across conditions with $n = 100$, results are qualitatively the same for $n = 200$, except for the inadmissible $k = n/1.11$. For sake of conciseness, in subsequent analysis we focus on the perhaps most representative scenarios of $k = 1$ and $k = n/5$, respectively *small* and *large* k.

6.2.2 Regression parameters estimation and cluster recovery

In Tables 4 and 5 we display results for all approaches in terms of mean and median of average MSE of model parameters, adj-Rand index, CPU time and selected c for, respectively, $n = 100$ and $n = 200$. Similarly to what Di Mari et al. (2017) found, with two components and $n = 100$, the difference between the unconstrained and the constrained approaches is small in terms of accuracy of regression parameter estimates. With $G = 3$ and $G = 4$ the gain of using a constrained approach is neater, especially with uneven mixing proportions. We observe that, together with keeping up very well with the cross-validated constrained approach, the proposed accelerated method performs equally well given the two alternative k's, and does from twice up to nearly ten times faster than ConC. Interestingly, ConK$_{k=n/5}$'s solutions are closer to ConC in terms of selected scale balance

*Computer programs are available from the corresponding author upon request.

k	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	time	c	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	time	c
	Mixing proportions $(0.5, 0.5)$					Mixing proportions $(0.2, 0.8)$				
1	0.014	0.007	0.962	0.059	0.302	0.037	0.015	0.973	0.073	0.231
	(0.013)	(0.012)	(0.054)	(0.022)	(0.285)	(0.129)	(0.031)	(0.051)	(0.136)	(0.266)
2	0.014	0.007	0.962	0.062	0.332	0.0365	0.0152	0.9726	0.1521	0.2365
	(0.013)	(0.012)	(0.055)	(0.025)	(0.293)	(0.129)	(0.031)	(0.051)	(0.121)	(0.273)
$(J+1)\times(G-1)$	0.013	0.007	0.962	0.065	0.367	0.037	0.015	0.973	0.071	0.241
	(0.013)	(0.012)	(0.055)	(0.026)	(0.300)	(0.129)	(0.031)	(0.051)	(0.113)	(0.272)
$n/10$	0.013	0.007	0.962	0.081	0.536	0.037	0.015	0.973	0.074	0.247
	(0.013)	(0.013)	(0.055)	(0.032)	(0.280)	(0.129)	(0.031)	(0.051)	(0.127)	(0.274)
$n/5$	0.014	0.009	0.961	0.087	0.628	0.036	0.015	0.973	0.074	0.247
	(0.013)	(0.015)	(0.056)	(0.036)	(0.262)	(0.128)	(0.031)	(0.051)	(0.119)	(0.275)
$n/2$	0.014	0.011	0.960	0.090	0.653	0.036	0.017	0.973	0.075	0.256
	(0.013)	(0.019)	(0.057)	(0.040)	(0.207)	(0.129)	(0.038)	(0.051)	(0.115)	(0.282)
$n/1.25$	0.013	0.009	0.962	0.085	0.557	0.037	0.015	0.973	0.093	0.430
	(0.013)	(0.018)	(0.053)	(0.036)	(0.221)	(0.128)	(0.032)	(0.048)	(0.136)	(0.332)
$n/1.11$	0.013	0.010	0.962	0.082	0.5418	0.037	0.019	0.973	0.097	0.529
	(0.012)	(0.025)	(0.054)	(0.036)	(0.242)	(0.129)	(0.038)	(0.045)	(0.120)	(0.339)
	Mixing proportions $(0.2, 0.4, 0.4)$					Mixing proportions $(0.2, 0.3, 0.5)$				
1	0.201	0.069	0.952	0.226	0.120	0.126	0.045	0.969	0.198	0.090
	(0.891)	(0.330)	(0.109)	(0.231)	(0.168)	(0.668)	(0.171)	(0.058)	(0.182)	(0.133)
2	0.233	0.092	0.945	0.232	0.124	0.125	0.045	0.969	0.201	0.099
	(0.953)	(0.421)	(0.131)	(0.254)	(0.178)	(0.668)	(0.170)	(0.058)	(0.191)	(0.147)
$(J+1)\times(G-1)$	0.225	0.085	0.946	0.246	0.164	0.125	0.044	0.969	0.215	0.123
	(0.907)	(0.365)	(0.130)	(0.260)	(0.210)	(0.667)	(0.168)	(0.058)	(0.202)	(0.170)
$n/10$	0.240	0.085	0.946	0.251	0.200	0.124	0.043	0.970	0.228	0.180
	(0.973)	(0.377)	(0.130)	(0.249)	(0.237)	(0.665)	(0.166)	(0.057)	(0.199)	(0.185)
$n/5$	0.208	0.078	0.947	0.858	0.242	0.093	0.040	0.970	0.228	0.180
	(0.811)	(0.336)	(0.128)	(2.003)	(0.265)	(0.402)	(0.142)	(0.055)	(0.199)	(0.224)
$n/2$	0.199	0.087	0.946	0.277	0.303	0.092	0.046	0.970	0.242	0.280
	(0.767)	(0.331)	(0.127)	(0.262)	(0.254)	(0.394)	(0.151)	(0.053)	(0.191)	(0.258)
$n/1.25$	0.201	0.127	0.947	0.289	0.331	0.088	0.086	0.968	0.254	0.329
	(0.804)	(0.388)	(0.121)	(0.321)	(0.253)	(0.356)	(0.211)	(0.053)	(0.292)	(0.260)
$n/1.11$	0.218	0.158	0.945	0.287	0.329	0.109	0.121	0.966	0.246	0.354
	(0.834)	(0.447)	(0.121)	(0.317)	(0.253)	(0.471)	(0.247)	(0.057)	(0.199)	(0.274)
	Mixing proportions $(0.25, 0.25, 0.25, 0.25)$					Mixing proportions $(0.1, 0.2, 0.3, 0.4)$				
1	1.074	0.246	0.908	0.601	0.078	2.334	0.351	0.921	0.530	0.028
	(2.634)	(0.595)	(0.145)	(0.349)	(0.124)	(3.397)	(0.637)	(0.105)	(0.267)	(0.068)
2	1.117	0.275	0.905	0.619	0.075	2.293	0.344	0.921	0.543	0.027
	(2.665)	(0.652)	(0.148)	(0.367)	(0.122)	(3.367)	(0.632)	(0.104)	(0.293)	(0.068)
$(J+1)\times(G-1)$	1.110	0.253	0.907	0.656	0.116	2.253	0.352	0.922	0.575	0.035
	(2.726)	(0.591)	(0.146)	(0.378)	(0.171)	(3.310)	(0.686)	(0.103)	(0.282)	(0.082)
$n/10$	1.107	0.253	0.907	0.659	0.121	2.217	0.348	0.923	0.577	0.036
	(2.720)	(0.591)	(0.146)	(0.367)	(0.177)	(3.300)	(0.684)	(0.103)	(0.288)	(0.083)
$n/5$	1.067	0.228	0.910	0.675	0.171	2.086	0.307	0.929	0.614	0.055
	(2.667)	(0.535)	(0.142)	(0.392)	(0.214)	(3.252)	(0.623)	(0.099)	(0.320)	(0.098)
$n/2$	1.019	0.245	0.909	0.716	0.247	1.981	0.238	0.934	0.688	0.149
	(2.552)	(0.557)	(0.143)	(0.455)	(0.228)	(3.236)	(0.459)	(0.101)	(0.389)	(0.172)
$n/1.25$	1.151	0.390	0.902	0.746	0.300	2.121	0.367	0.923	0.726	0.287
	(2.818)	(0.814)	(0.150)	(0.557)	(0.242)	(3.688)	(0.633)	(0.130)	(0.412)	(0.276)
$n/1.11$	1.504	0.555	0.876	0.762	0.336	2.362	0.432	0.917	0.708	0.322
	(2.978)	(0.917)	(0.159)	(0.592)	(0.285)	(3.765)	(0.659)	(0.134)	(0.398)	(0.305)

Table 2: ConK. Results for different values of k. 250 samples, $n = 100$, 10 random starts, 3 regressors and intercept. Values averaged across samples, with standard deviations in parentheses.

than ConK$_{k=1}$'s. In terms of clusters recovery, all of the three constrained setups give better results compared to the unconstrained algorithms, and are relatively close to one another.

Considering a sample size of $n = 200$ (Table 5) boosts the performance of all methods, especially of the constrained methods while HetN and HomN show a good improvement in the $(0.25, 0.25, 0.25, 0.25)'$ condition only. The conditions with $G = 3$ and $G = 4$, especially with $G = 4$ and uneven component sizes, are where the difference between the constrained and the unconstrained methods is largest - with an average MSE for the estimated regressors and the component variances up to 10 times smaller in median. In all conditions ConC and ConK deliver the best clusters recovery.

In all the simulated scenario presented, the adj-Rand index values are quite high (almost all them greater than 0.9). Our personal simulation experience has shown that by lowering the cluster separation the behavior of all the methods, in terms of cluster recovery and regression parameters estimation, decreases while their relative performance remains almost the same.

k	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	time	c	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	time	c
	Mixing proportions $(0.5, 0.5)$					Mixing proportions $(0.2, 0.8)$				
1	0.006	0.003	0.978	0.075	0.343	0.011	0.006	0.984	0.077	0.262
	(0.006)	(0.005)	(0.029)	(0.014)	(0.300)	(0.016)	(0.012)	(0.024)	(0.029)	(0.280)
2	0.006	0.003	0.978	0.077	0.371	0.011	0.006	0.984	0.078	0.256
	(0.006)	(0.005)	(0.030)	(0.017)	(0.305)	(0.016)	(0.012)	(0.024)	(0.029)	(0.281)
$(J+1)\times(G-1)$	0.006	0.003	0.978	0.080	0.389	0.011	0.006	0.984	0.078	0.258
	(0.006)	(0.005)	(0.030)	(0.020)	(0.310)	(0.016)	(0.012)	(0.024)	(0.029)	(0.282)
$n/10$	0.006	0.003	0.978	0.107	0.606	0.012	0.006	0.984	0.079	0.269
	(0.006)	(0.005)	(0.030)	(0.036)	(0.281)	(0.016)	(0.012)	(0.024)	(0.032)	(0.290)
$n/5$	0.006	0.005	0.975	0.114	0.702	0.011	0.006	0.984	0.079	0.268
	(0.006)	(0.007)	(0.031)	(0.038)	(0.241)	(0.016)	(0.012)	(0.024)	(0.033)	(0.293)
$n/2$	0.006	0.007	0.975	0.114	0.683	0.011	0.008	0.984	0.082	0.288
	(0.007)	(0.016)	(0.033)	(0.039)	(0.195)	(0.016)	(0.023)	(0.024)	(0.035)	(0.299)
$n/1.25$	0.006	0.005	0.976	0.110	0.607	0.011	0.007	0.982	0.102	0.483
	(0.006)	(0.012)	(0.032)	(0.039)	(0.203)	(0.015)	(0.013)	(0.028)	(0.056)	(0.335)
$n/1.11$	0.006	0.005	0.977	0.108	0.595	0.011	0.012	0.981	0.106	0.558
	(0.006)	(0.011)	(0.031)	(0.040)	(0.219)	(0.015)	(0.036)	(0.032)	(0.075)	(0.343)
	Mixing proportions $(0.2, 0.4, 0.4)$					Mixing proportions $(0.2, 0.3, 0.5)$				
1	0.010	0.007	0.986	0.197	0.132	0.015	0.011	0.987	0.226	0.099
	(0.010)	(0.022)	(0.019)	(0.144)	(0.162)	(0.081)	(0.068)	(0.021)	(0.181)	(0.136)
2	0.010	0.007	0.986	0.200	0.132	0.015	0.011	0.987	0.229	0.091
	(0.010)	(0.022)	(0.019)	(0.157)	(0.163)	(0.081)	(0.068)	(0.021)	(0.181)	(0.131)
$(J+1)\times(G-1)$	0.010	0.007	0.986	0.207	0.171	0.015	0.011	0.986	0.232	0.108
	(0.010)	(0.022)	(0.019)	(0.159)	(0.201)	(0.080)	(0.068)	(0.021)	(0.173)	(0.157)
$n/10$	0.010	0.007	0.986	0.230	0.258	0.015	0.011	0.987	0.260	0.142
	(0.010)	(0.022)	(0.018)	(0.185)	(0.252)	(0.080)	(0.065)	(0.020)	(0.339)	(0.191)
$n/5$	0.010	0.009	0.986	0.253	0.307	0.015	0.011	0.987	0.265	0.166
	(0.010)	(0.023)	(0.020)	(0.272)	(0.282)	(0.079)	(0.064)	(0.020)	(0.214)	(0.226)
$n/2$	0.010	0.014	0.985	0.264	0.337	0.010	0.019	0.986	0.291	0.282
	(0.010)	(0.029)	(0.021)	(0.357)	(0.257)	(0.008)	(0.042)	(0.018)	(0.251)	(0.278)
$n/1.25$	0.011	0.036	0.983	0.258	0.378	0.011	0.075	0.983	0.296	0.330
	(0.013)	(0.099)	(0.023)	(0.268)	(0.253)	(0.009)	(0.158)	(0.021)	(0.278)	(0.269)
$n/1.11$	0.033	0.065	0.981	0.251	0.377	0.032	0.140	0.979	0.307	0.360
	(0.338)	(0.181)	(0.034)	(0.188)	(0.250)	(0.303)	(0.271)	(0.035)	(0.312)	(0.256)
	Mixing proportions $(0.25, 0.25, 0.25, 0.25)$					Mixing proportions $(0.1, 0.2, 0.3, 0.4)$				
1	0.083	0.029	0.980	0.541	0.110	0.225	0.065	0.985	0.656	0.035
	(0.699)	(0.204)	(0.041)	(0.381)	(0.110)	(1.007)	(0.280)	(0.039)	(0.521)	(0.068)
2	0.090	0.036	0.978	0.544	0.112	0.235	0.064	0.984	0.671	0.031
	(0.573)	(0.217)	(0.048)	(0.391)	(0.118)	(1.046)	(0.279)	(0.040)	(0.628)	(0.067)
$(J+1)\times(G-1)$	0.090	0.036	0.978	0.563	0.177	0.194	0.051	0.987	0.690	0.039
	(0.572)	(0.216)	(0.048)	(0.433)	(0.175)	(0.932)	(0.189)	(0.025)	(0.672)	(0.082)
$n/10$	0.097	0.037	0.978	0.590	0.254	0.186	0.051	0.987	0.754	0.056
	(0.711)	(0.218)	(0.049)	(0.451)	(0.212)	(0.895)	(0.187)	(0.025)	(0.807)	(0.117)
$n/5$	0.096	0.039	0.978	0.602	0.338	0.164	0.043	0.988	0.826	0.078
	(0.708)	(0.218)	(0.048)	(0.445)	(0.247)	(0.838)	(0.160)	(0.022)	(0.971)	(0.146)
$n/2$	0.106	0.049	0.976	0.617	0.414	0.175	0.060	0.986	0.924	0.147
	(0.748)	(0.219)	(0.047)	(0.451)	(0.224)	(0.856)	(0.164)	(0.030)	(0.956)	(0.179)
$n/1.25$	0.175	0.088	0.969	0.635	0.322	0.171	0.157	0.985	0.937	0.281
	(0.942)	(0.281)	(0.065)	(0.738)	(0.211)	(0.879)	(0.293)	(0.024)	(0.954)	(0.229)
$n/1.11$	0.513	0.166	0.952	0.605	0.309	0.256	0.236	0.981	1.067	0.332
	(1.761)	(0.384)	(0.093)	(0.532)	(0.233)	(1.179)	(0.492)	(0.044)	(1.523)	(0.268)

Table 3: ConK. Results for different values of k. 250 samples, $n = 200$, 10 random starts, 3 regressors and intercept. Values averaged across samples, with standard deviations in parentheses.

Algorithm	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	time	c	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	time	c
	Mixing proportions $(0.5, 0.5)$					Mixing proportions $(0.2, 0.8)$				
HomN	0.016	0.037	0.940	0.017	-	0.040	0.051	0.965	0.019	-
	(0.024)	(0.070)	(0.076)	(0.006)	-	(0.122)	(0.116)	(0.062)	(0.011)	-
HetN	0.014	0.008	0.962	0.017	-	0.052	0.044	0.967	0.017	-
	(0.013)	(0.012)	(0.055)	(0.008)	-	(0.187)	(0.209)	(0.073)	(0.016)	-
ConC	0.014	0.008	0.963	0.285	0.622	0.042	0.024	0.972	0.283	0.619
	(0.013)	(0.015)	(0.056)	(0.070)	(0.309)	(0.157)	(0.105)	(0.061)	(0.092)	(0.357)
ConK$_{k=1}$	0.014	0.007	0.962	0.059	0.302	0.037	0.015	0.973	0.073	0.231
	(0.013)	(0.012)	(0.054)	(0.022)	(0.285)	(0.129)	(0.031)	(0.051)	(0.136)	(0.266)
ConK$_{k=n/5}$	0.014	0.009	0.961	0.087	0.628	0.036	0.015	0.973	0.074	0.247
	(0.013)	(0.015)	(0.056)	(0.036)	(0.262)	(0.128)	(0.031)	(0.051)	(0.119)	(0.275)
	Mixing proportions $(0.2, 0.4, 0.4)$					Mixing proportions $(0.2, 0.3, 0.5)$				
HomN	0.319	0.111	0.940	0.176	-	0.248	0.090	0.955	0.187	-
	(1.171)	(0.323)	(0.116)	(0.109)	-	(0.932)	(0.269)	(0.088)	(0.356)	-
HetN	0.320	0.166	0.932	0.180	-	0.323	0.123	0.958	0.145	-
	(1.090)	(0.685)	(0.154)	(0.119)	-	(1.215)	(0.391)	(0.075)	(0.090)	-
ConC	0.212	0.092	0.948	1.436	0.409	0.105	0.046	0.969	1.293	0.361
	(0.832)	(0.372)	(0.119)	(0.776)	(0.347)	(0.472)	(0.167)	(0.055)	(0.467)	(0.337)
ConK$_{k=1}$	0.201	0.069	0.952	0.226	0.120	0.126	0.045	0.969	0.198	0.090
	(0.891)	(0.330)	(0.109)	(0.231)	(0.168)	(0.668)	(0.171)	(0.058)	(0.182)	(0.133)
ConK$_{k=n/5}$	0.208	0.078	0.947	0.858	0.242	0.093	0.040	0.970	0.228	0.180
	(0.811)	(0.336)	(0.128)	(2.003)	(0.265)	(0.402)	(0.142)	(0.055)	(0.199)	(0.224)
	Mixing proportions $(0.25, 0.25, 0.25, 0.25)$					Mixing proportions $(0.1, 0.2, 0.3, 0.4)$				
HomN	1.126	0.224	0.880	0.337	-	3.882	0.217	0.912	0.338	-
	(2.994)	(0.471)	(0.191)	(0.148)	-	(4.199)	(0.331)	(0.110)	(0.188)	-
HetN	2.567	0.620	0.824	0.335	-	3.868	0.605	0.880	0.312	-
	(3.492)	(1.051)	(0.183)	(0.111)	-	(3.650)	(0.695)	(0.110)	(0.119)	-
ConC	1.080	0.273	0.907	2.622	0.263	2.549	0.358	0.919	2.510	0.129
	(2.481)	(0.636)	(0.145)	(1.002)	(0.318)	(3.270)	(0.653)	(0.100)	(0.893)	(0.206)
ConK$_{k=1}$	1.074	0.246	0.908	0.601	0.078	2.334	0.351	0.921	0.530	0.028
	(2.634)	(0.595)	(0.145)	(0.349)	(0.124)	(3.397)	(0.637)	(0.105)	(0.267)	(0.068)
ConK$_{k=n/5}$	1.067	0.228	0.910	0.675	0.171	2.086	0.307	0.929	0.614	0.055
	(2.667)	(0.535)	(0.142)	(0.392)	(0.214)	(3.252)	(0.623)	(0.099)	(0.320)	(0.098)

Table 4: 250 samples, $n = 100$, 10 random starts, 3 regressors and intercept. Values averaged across samples, with standard deviations in parentheses.

Algorithm	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	time	c	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	time	c
	Mixing proportions $(0.5, 0.5)$					Mixing proportions $(0.2, 0.8)$				
HomN	0.007	0.028	0.957	0.097	-	0.012	0.038	0.976	0.116	-
	(0.012)	(0.068)	(0.053)	(0.020)	-	(0.016)	(0.102)	(0.039)	(0.033)	-
HetN	0.006	0.003	0.978	0.099	-	0.014	0.008	0.984	0.099	-
	(0.006)	(0.005)	(0.029)	(0.024)	-	(0.048)	(0.035)	(0.024)	(0.037)	-
ConC	0.006	0.003	0.978	2.323	0.632	0.011	0.006	0.984	2.335	0.579
	(0.006)	(0.005)	(0.030)	(0.144)	(0.299)	(0.015)	(0.009)	(0.025)	(0.191)	(0.357)
ConK$_{k=1}$	0.006	0.003	0.978	0.075	0.343	0.011	0.006	0.984	0.077	0.262
	(0.006)	(0.005)	(0.029)	(0.014)	(0.300)	(0.016)	(0.012)	(0.024)	(0.029)	(0.280)
ConK$_{k=n/5}$	0.006	0.005	0.975	0.114	0.702	0.011	0.006	0.984	0.079	0.268
	(0.006)	(0.007)	(0.031)	(0.038)	(0.241)	(0.016)	(0.012)	(0.024)	(0.033)	(0.293)
	Mixing proportions $(0.2, 0.4, 0.4)$					Mixing proportions $(0.2, 0.3, 0.5)$				
HomN	0.055	0.072	0.972	0.334	-	0.324	0.148	0.964	0.432	
	(0.507)	(0.360)	(0.061)	(0.282)	-	(1.201)	(0.405)	(0.067)	(0.629)	-
HetN	0.040	0.021	0.983	0.303	-	0.063	0.023	0.984	0.297	
	(0.422)	(0.158)	(0.035)	(0.136)	-	(0.478)	(0.145)	(0.034)	(0.188)	-
ConC	0.010	0.007	0.986	3.530	0.414	0.015	0.012	0.986	3.723	0.313
	(0.010)	(0.022)	(0.019)	(0.591)	(0.325)	(0.080)	(0.067)	(0.022)	(1.379)	(0.329)
ConK$_{k=1}$	0.010	0.007	0.986	0.197	0.132	0.015	0.011	0.987	0.226	0.099
	(0.010)	(0.022)	(0.019)	(0.144)	(0.162)	(0.081)	(0.068)	(0.021)	(0.181)	(0.136)
ConK$_{k=n/5}$	0.010	0.009	0.986	0.253	0.307	0.015	0.011	0.987	0.265	0.166
	(0.010)	(0.023)	(0.020)	(0.272)	(0.282)	(0.079)	(0.064)	(0.020)	(0.214)	(0.226)
	Mixing proportions $(0.25, 0.25, 0.25, 0.25)$					Mixing proportions $(0.1, 0.2, 0.3, 0.4)$				
HomN	0.036	0.056	0.975	0.661	-	3.029	0.248	0.948	0.742	-
	(0.261)	(0.126)	(0.038)	(0.338)	-	(4.081)	(0.299)	(0.057)	(0.622)	-
HetN	0.571	0.167	0.948	0.716	-	0.979	0.224	0.969	0.664	-
	(1.748)	(0.600)	(0.106)	(0.303)	-	(2.311)	(0.622)	(0.056)	(0.285)	-
ConC	0.121	0.038	0.978	5.634	0.350	0.1865	0.0481	0.9871	6.3252	0.133
	(0.816)	(0.220)	(0.050)	(1.711)	(0.281)	(0.708)	(0.217)	(0.048)	(1.369)	(0.219)
ConK$_{k=1}$	0.083	0.029	0.980	0.541	0.110	0.225	0.065	0.985	0.656	0.035
	(0.699)	(0.204)	(0.041)	(0.381)	(0.110)	(1.007)	(0.280)	(0.039)	(0.521)	(0.068)
ConK$_{k=n/5}$	0.096	0.039	0.978	0.602	0.338	0.164	0.043	0.988	0.826	0.078
	(0.708)	(0.218)	(0.048)	(0.445)	(0.247)	(0.838)	(0.160)	(0.022)	(0.971)	(0.146)

Table 5: 250 samples, $n = 200$, 10 random starts, 3 regressors and intercept. Values averaged across samples, with standard deviations in parentheses.

6.2.3 Target

In Table 6 the results obtained with our proposed choice for the target variance, the homoscedastic normal variance, are compared with those obtained by using a different target, the OLS residual variance. We consider the simulation scenarios $G = 4$, class proportions equal to $(0.25, 0.25, 0.25, 0.25)'$ and sample sizes of 100 and 200. Constrained approaches based on the OLS target are significantly better - in terms of parameter estimation accuracy and cluster recovery - than the unconstrained approaches (HomN and HetN). Yet, we find the OLS target to be outperformed by the homoscedastic normal target variance. Interestingly, we observe a connection between the selected c and the adopted target variance: all the c values selected under the OLS target are very close to zero. In this respect, c can also be seen as an indicator of how suitable or unsuitable a given target variance is e.g. a c very close to zero might indicate that the target carries little information on the component scales.

Algorithm	Avg MSE $\hat{\boldsymbol{\beta}}$	Avg MSE $\hat{\boldsymbol{\sigma}}^2$	Adj-Rand	c
		$n = 100$, OLS target		
ConC	1.417	0.418	0.884	0.005
	(2.799)	(0.789)	(0.160)	(0.000)
ConK$_{k=n/5}$	1.289	0.329	0.890	0.002
	(2.638)	(0.650)	(0.150)	(0.000)
		$n = 100$, HomN target		
ConC	1.080	0.273	0.907	0.263
	(2.481)	(0.636)	(0.145)	(0.318)
ConK$_{k=n/5}$	1.067	0.228	0.910	0.171
	(2.667)	(0.535)	(0.142)	(0.215)
		$n = 200$, OLS target		
ConC	0.219	0.133	0.967	0.001
	(1.101)	(0.635)	(0.073)	(0.002)
ConK$_{k=n/5}$	0.272	0.124	0.966	0.001
	(1.288)	(0.533)	(0.078)	(0.002)
		$n = 200$, HomN target		
ConC	0.121	0.038	0.978	0.350
	(0.816)	(0.220)	(0.050)	(0.281)
ConK$_{k=n/5}$	0.096	0.039	0.978	0.338
	(0.709)	(0.218)	(0.048)	(0.247)

Table 6: OLS and HomN target variances. 250 samples, $n = 100$ and $n = 200$, class proportions $(0.25, 0.25, 0.25, 0.25)'$, 10 random starts, 3 regressors and intercept. Mean values across samples, with standard deviations in parentheses.

6.2.4 Model selection: the number of clusters

In Table 7 we display the percentage of correct guesses for G delivered by each method for each of the 12 simulation conditions. The procedure minimizing the BIC computed using the solutions of HetN almost completely fails to recover the correct number of clusters. By contrast, we observe that in all conditions the modified BIC computed using the constrained approaches yields the highest number of correct guesses. Very similar performance is achieved with standard BIC computed at the ConC solution. Using standard BIC tarnishes the performance of ConK, which however outperforms HomN in setups with larger sample sizes ($n = 200$), as well as with smaller sample size ($n = 100$) but larger G^* and uneven component sizes.

Further insight can be acquired by looking at the absolute frequencies of guesses for the number of components G of each method (Figure 1 and 2). For the constrained approaches ConC and ConK, we compute the BIC based on the formulas of Equations (8) and (9). The advantage of ConK and ConC over HomN - which does slightly better than HetN - shows up in all conditions, and is more evident for $n = 200$. We observe that the correction for the different model complexities entailed by c shows an improvement in selecting the number of components for $n = 100$, which vanishes for ConC when $n = 200$, and is neater for ConK[§]. One possible explanation for this might be that the cross-validation strategy for selecting the scale balance is less affected by overspecification of the number of components - although when the number of components is well specified, ConK delivers almost no loss relative to ConC in parameter estimation.

[§]We checked that this is also the case for $n > 200$. Related figures are available from the corresponding author upon request

$n = 100$	$G = 2$		$G = 3$		$G = 4$	
	(0.5,0.5)'	(0.2,0.8)'	(0.2,0.4,0.4)'	(0.2,0.3,0.5)'	(0.25,0.25,0.25,0.25)'	(0.1,0.2,0.3,0.4)'
HomN	0.816	0.948	0.720	0.768	0.660	0.496
HetN	0.404	0.428	0.116	0.132	0.140	0.128
ConC	0.956	0.976	0.920	0.928	0.780	0.536
ConC*	0.968	0.972	0.900	0.940	0.780	0.548
ConK$_{k=1}$	0.828	0.796	0.700	0.712	0.632	0.536
ConK$^*_{k=1}$	0.852	0.832	0.744	0.772	0.692	0.540
ConK$_{k=n/5}$	0.832	0.800	0.704	0.712	0.652	0.532
ConK$^*_{k=n/5}$	0.860	0.832	0.748	0.776	0.700	0.536
$n = 200$	$G = 2$		$G = 3$		$G = 4$	
	(0.5,0.5)'	(0.2,0.8)'	(0.2,0.4,0.4)'	(0.2,0.3,0.5)'	(0.25,0.25,0.25,0.25)'	(0.1,0.2,0.3,0.4)'
HomN	0.796	0.988	0.804	0.812	0.776	0.568
HetN	0.524	0.624	0.300	0.316	0.172	0.168
ConC	0.992	0.992	0.980	0.988	0.968	0.908
ConC*	0.996	0.992	0.988	0.996	0.972	0.904
ConK$_{k=1}$	0.964	0.972	0.944	0.948	0.920	0.884
ConK$^*_{k=1}$	0.980	0.980	0.968	0.956	0.932	0.904
ConK$_{k=n/5}$	0.964	0.972	0.944	0.944	0.916	0.876
ConK$^*_{k=n/5}$	0.980	0.984	0.968	0.956	0.932	0.896

Table 7: Proportion of correct guesses for G. 250 samples, 10 random starts, 3 regressors and intercept. For each setup, lowest BIC solution selected between $1, \ldots, G^* + 2$ components. Methods with index * use the BIC formula of Equation (9).

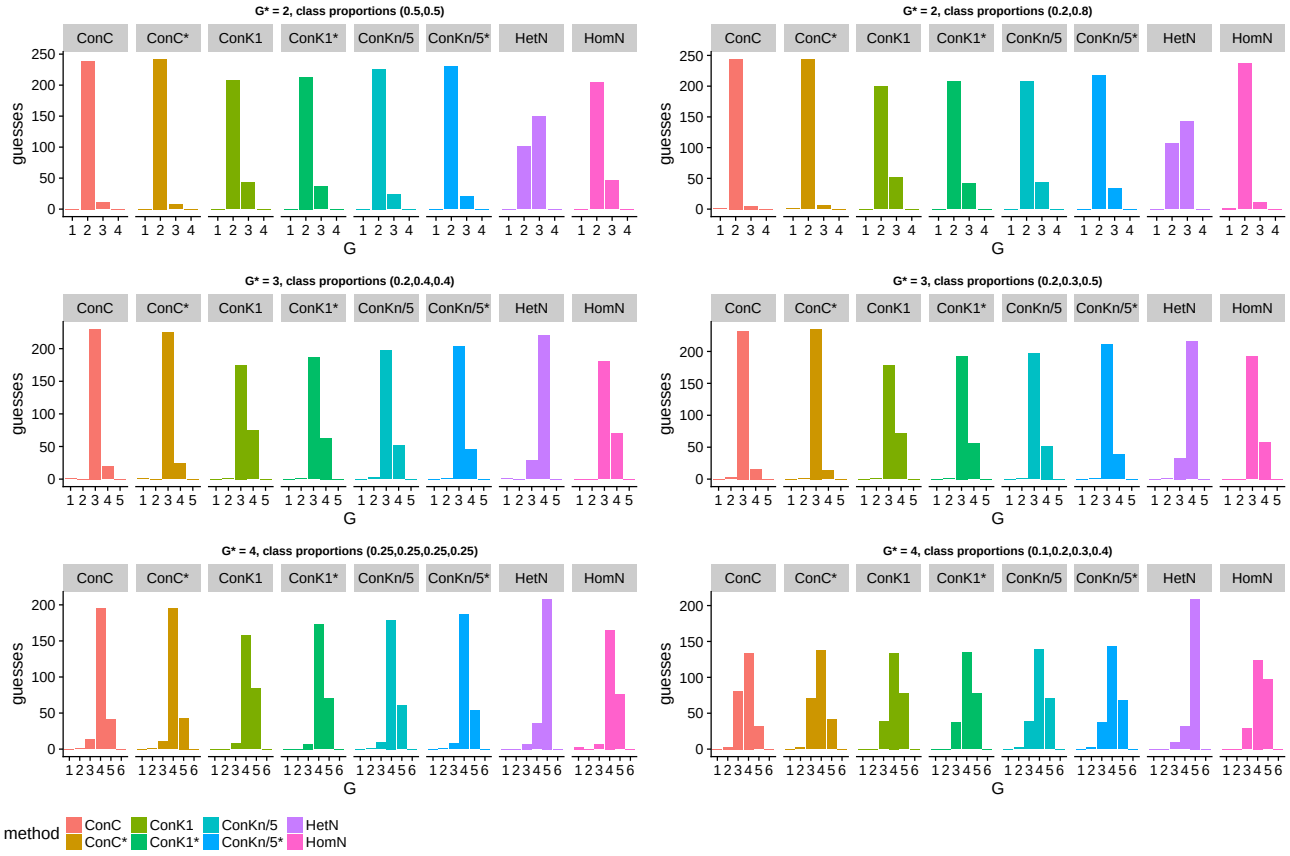

Figure 1: Number of guesses for G per method, $n = 100$. The cross-validation procedure is run with $M = n/5$ and test set of size equal to $n/10$.

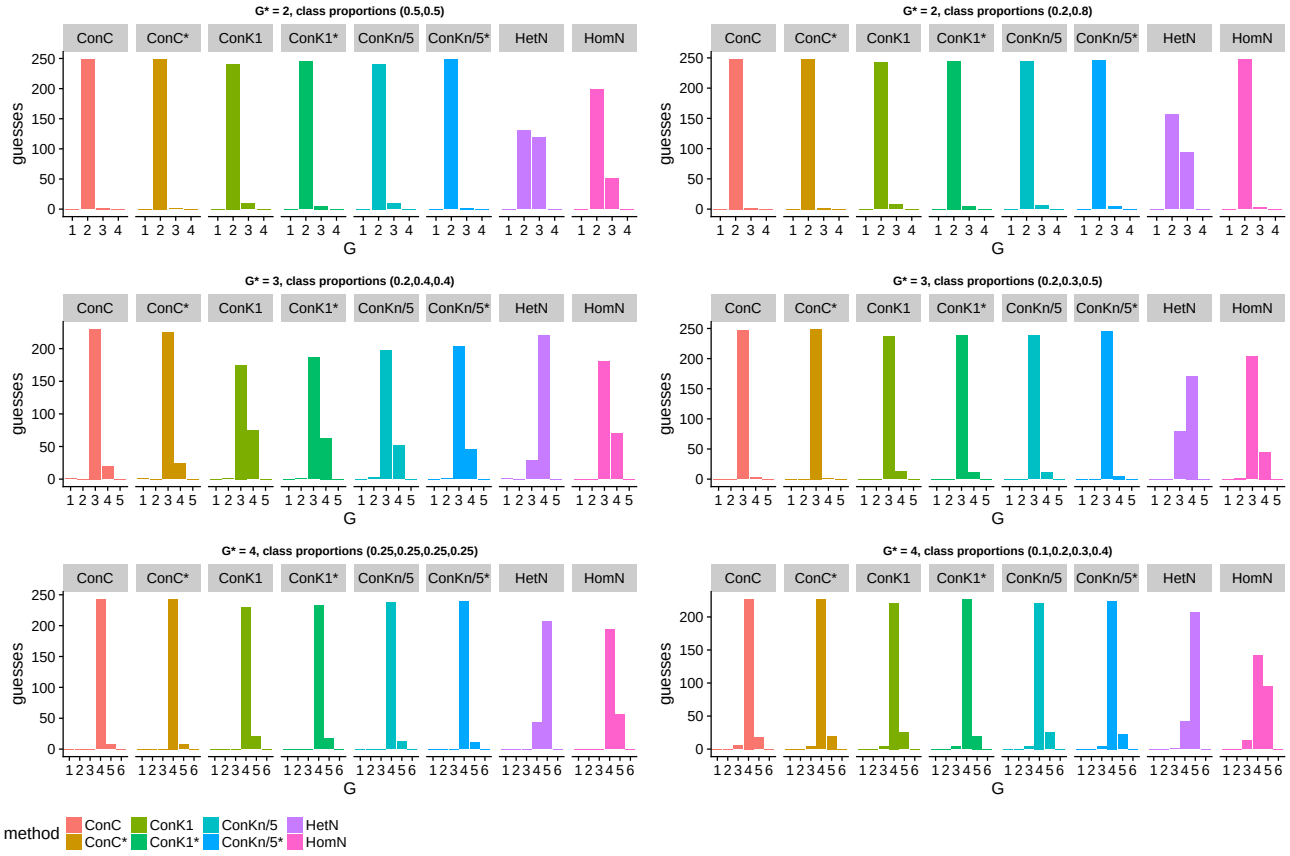

Figure 2: Number of guesses for G per method, $n = 200$. The cross-validation procedure is run with $M = n/5$ and test set of size equal to $n/10$.

Finally, in Figure 3 we plot the average Adjusted Rand index, computed using the solution with number of components as chosen by each method. For instance, if HetN has chosen 4 components when $G^* = 2$, we compute the Adj-Rand comparing the 4-component solution with the true 2-component one. Overall, by simultaneously looking at model selection and cluster recovery, all the constrained methods, with and without the correction for the number of degrees of freedom, yield very similar performances, doing always better than the unconstrained rivals.

7 Two real-data applications

In this Section we illustrate the use of the constrained approaches, ConC and ConK, and compare them with HetN and HomN.

For neither of the data sets the number of subgroups in the underlying population is known. We fitted a clusterwise linear regression, using the 3 methods under comparison, on the *CEO* data set (`http://lib.stat.cmu.edu/DASL/DataArchive.html`), with 2, 3, 4, and 5 components, and analyzed the 2-class solution - which minimized the BIC formulas of Equation (8) and (9) under ConC and ConK (for $k = 1$ and $k = n/5$), and the plain BIC under HomN - in terms of estimated model parameters and clustering. We carried out a similar exercise on the *AutoMpg* data set (available at `https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data`), where instead only the constrained methods agree on the 2-component solution - seemingly the most suited to the data in terms of clusters interpretation.

7.1 *CEO* data

This data set contains information about salary (dependent variable) and age (independent variable) of 59 CEOs from small U.S. companies. The underlying clusters structure is unknown. Among those who already analyzed this data set, Carbonneau, Caporossi, and Hansen (2011) fitted a 2-component clusterwise linear regression, whereas Bagirov, Ugon, and Mirzayeva (2013) compared the 2-component and the 4-component setups.

In Table 8 we show BIC values computed using respectively HomN and HetN, and BIC's (Equations (8) and (9)) computed using ConC and ConK (with $k = 1$ and $k = n/5$). The con-

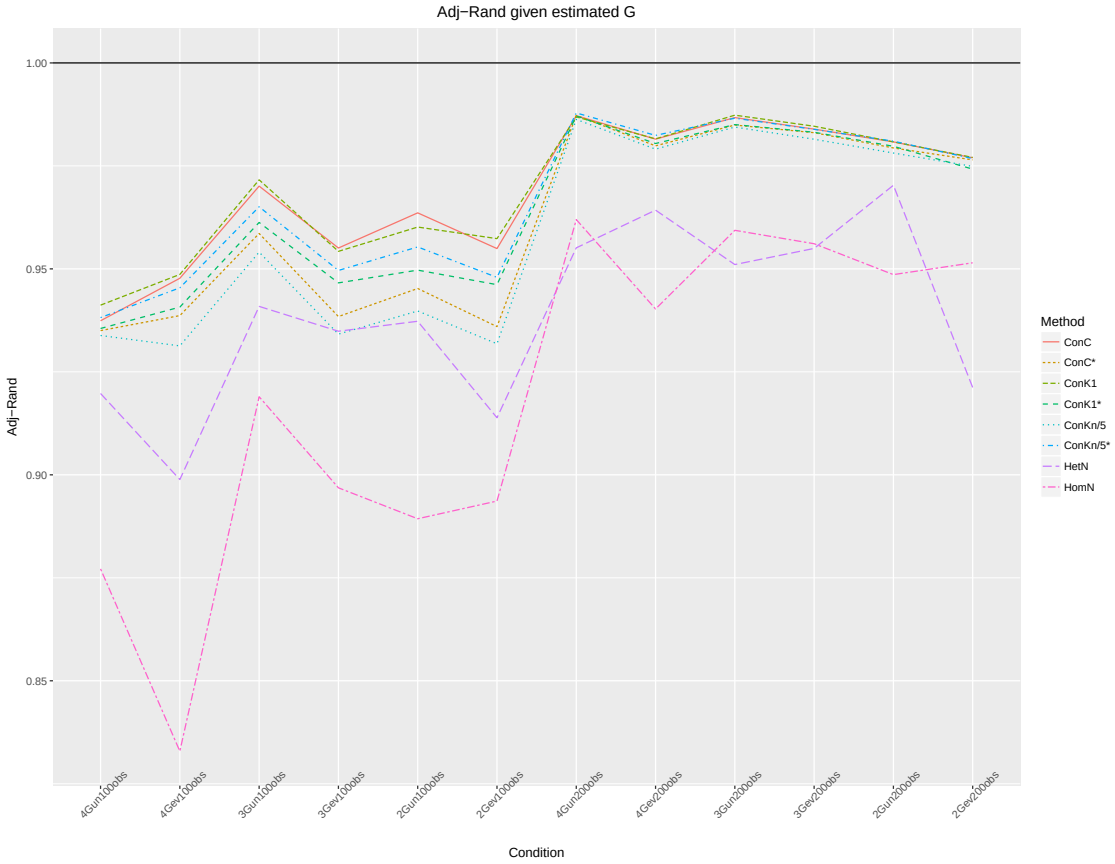

Figure 3: Average Adjusted Rand Index (Adj-Rand) for each method, in all 12 simulation conditions, with the optimal G as selected by each method. Tick marks in the x-axis to be read as number of clusters (2, 3 or 4 G), even ("ev") or uneven ("un") mixing proportions and number of observations ("100obs" and "200obs").

strained approaches and HomN all agree on the two component solution. Using BIC with HetN would lead to select the seemingly spurious 4-component solution showed in Di Mari et al. (2017).

	$G = 2$	$G = 3$	$G = 4$	$G = 5$
BIC_{HomN}	**706.30**	712.14	719.41	725.66
BIC_{HetN}	704.42	707.70	**593.94**	601.88
BIC_{ConC}	**706.86**	721.13	740.92	741.53
$\text{BIC}_{\text{ConC}^*}$	**702.12**	716.45	724.61	725.10
$\text{BIC}_{\text{ConK}_{k=1}}$	**704.42**	707.72	712.01	718.95
$\text{BIC}_{\text{ConK}^*_{k=1}}$	**703.23**	707.72	712.01	718.95
$\text{BIC}_{\text{ConK}_{k=n/5}}$	**707.25**	713.76	727.35	727.88
$\text{BIC}_{\text{ConK}^*_{k=n/5}}$	**702.12**	713.71	721.95	727.76

Table 8: *CEO* data. BIC values for $G = 2$, $G = 3$, $G = 4$, and $G = 5$. Best solutions out of 100 random starts. Minimum BIC values in bold for each method.

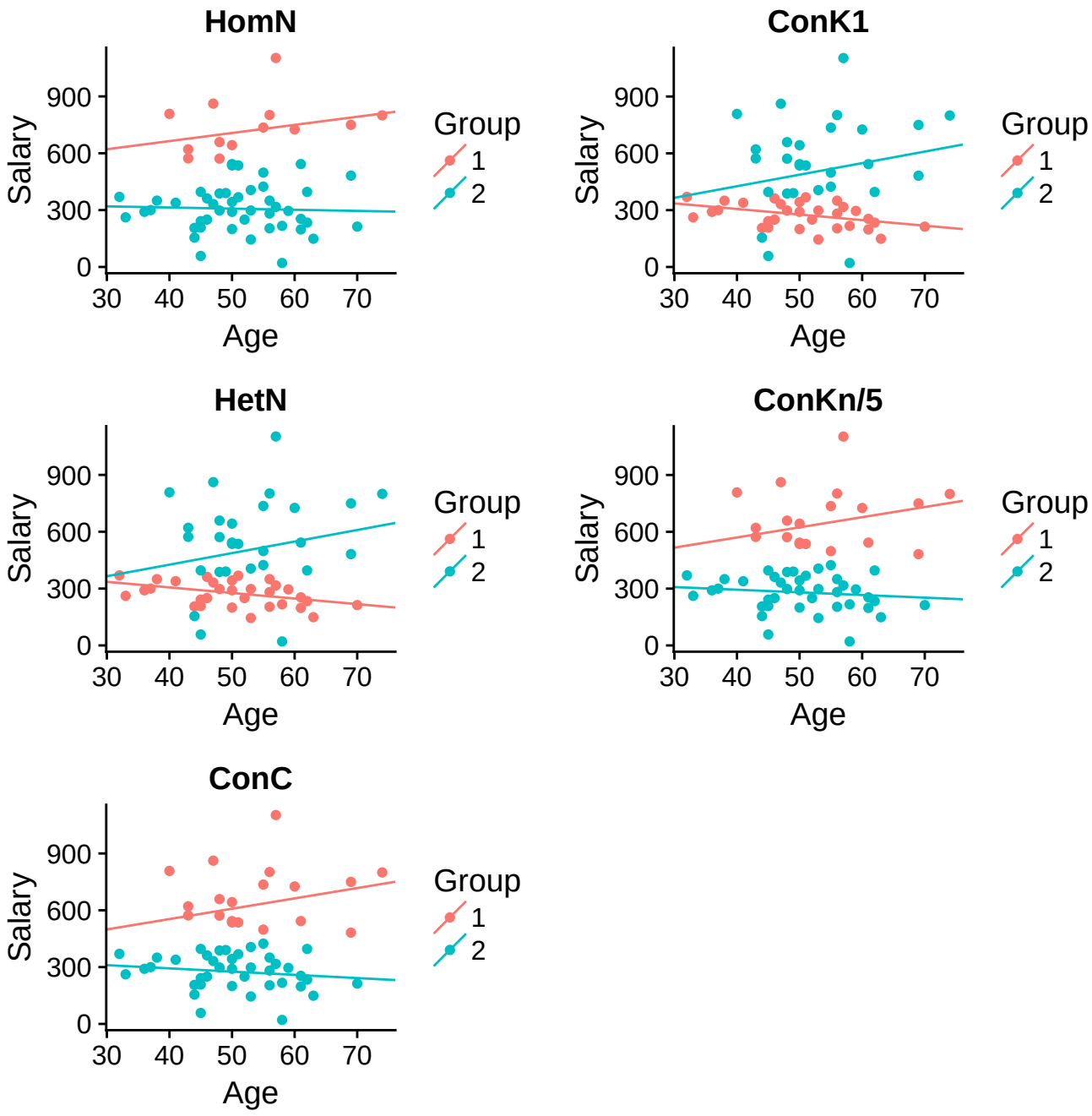

Figure 4: *CEO* data. Clusterwise regressions of salary on age of CEO's. Best solutions out of 100 random starts, $G = 2$. The cross-validation procedure is run with $M = n/5$ and test set of size equal to $n/10$.

The 2-class clusterwise linear regressions and the crisp assignments are plotted in Figure 4.

We observe that, in line with the simulation study, ConC and ConK with $k = n/5$ yield the

same clustering, and such clustering is exactly in between HomN and HetN, as well as very similar regression lines. This confirms what was found in Di Mari et al. (2017). By contrast, ConK with $k = 1$ produces a final solution which is closer to HetN.

7.2 *AutoMpg* data

This data set contains a sample of 398 vehicles, where information on city-cycle fuel consumption in miles per gallon is gathered for each vehicle, alongside with the following set of covariates (of mixed type): number of cylinders, model year, and origin, which are discrete valued; displacement, horsepower, weight, and acceleration, which are instead continuous valued. Records for horsepower were missing for six sample units. Given that the car model is available, with all relevant information, we were able to retrieve the missing values and included them in the data set.

We estimated a clusterwise linear regression model of miles per gallon on the above set of covariates. Plain BIC and modified BIC - for ConC and ConK only - values are reported in Table 9. Constrained approaches largely agree on the two-component solution (5 out of 6), whereby HomN and HetN favor respectively the 3 and the 5 component solution. Interestingly, ConK with $k = 1$ seems to need a correction for model complexity in the BIC to behave coherently with ConC and ConK with $k = 1$.

	$G = 2$	$G = 3$	$G = 4$	$G = 5$
BIC_{HomN}	1365.89	**1363.22**	1381.12	1398.56
BIC_{HetN}	1329.35	1340.55	1320.86	**1319.93**
BIC_{ConC}	**1329.74**	1340.95	1364.76	1371.26
$\text{BIC}_{\text{ConC}^*}$	**1325.03**	1336.11	1364.10	1370.17
$\text{BIC}_{\text{ConK}_{k=1}}$	1329.35	1340.55	**1328.49**	1340.18
$\text{BIC}_{\text{ConK}^*_{k=1}}$	**1326.52**	1338.08	1328.49	1340.13
$\text{BIC}_{\text{ConK}_{k=n/5}}$	**1337.27**	1356.77	1361.80	1397.75
$\text{BIC}_{\text{ConK}^*_{k=n/5}}$	**1330.49**	1344.93	1361.13	1389.29

Table 9: *Auto-Mpg* data. BIC values for $G = 2$, $G = 3$, $G = 4$, and $G = 5$. Best solutions out of 100 random starts. Minimum BIC values in bold for each method.

By looking at Table 10 we observe that acceleration ($\mathbf{x}_1$), cylinders ($\mathbf{x}_2$), and displacement ($\mathbf{x}_3$) have all positive effect on miles per gallon in the first (smaller) component, and negative effect in the second (larger) component. Cars with more horsepower, not surprisingly, tend to drive less miles per gallon - although the effect is relatively milder for the second (larger) component

- whereby a more recent model year ($\mathbf{x}_5$), all else equal, is positively associated with miles per gallon in both components - again, with a relatively milder effect on the second component.

	HomN		HetN		ConC		ConK$_{k=1}$		ConK$_{k=n/5}$	
p_g	0.2215	0.7785	0.4473	0.5527	0.4353	0.5647	0.4473	0.5527	0.3857	0.6143
intercept	-35.0716	-3.2278	-23.3485	3.7071	-23.5883	3.5861	-23.3480	3.7072	-24.7003	2.5239
β_{1g}	0.1819	-0.2530	0.1354	-0.4212	0.1383	-0.4177	0.1354	-0.4212	0.1601	-0.3867
β_{2g}	1.1272	-0.7172	0.1853	-0.9055	0.1767	-0.8938	0.1853	-0.9055	0.2145	-0.8550
β_{3g}	0.0170	0.0004	0.0362	-0.0116	0.0367	-0.0116	0.0362	-0.0116	0.0376	-0.0112
β_{4g}	-0.2113	-0.0077	-0.1188	-0.0035	-0.1211	-0.0031	-0.1188	-0.0035	-0.1286	-0.0015
β_{5g}	1.1328	0.5862	0.9546	0.4699	0.9592	0.4730	0.9546	0.4699	0.9773	0.4921
β_{6g}	-0.0070	-0.0042	-0.0084	-0.0022	-0.0084	-0.0022	-0.0084	-0.0022	-0.0085	-0.0026
β_{7g}	0.6887	1.7958	0.7283	2.6872	0.7221	2.6430	0.7283	2.6872	0.6703	2.4315
σ^2_g	2.3770	2.3770	3.1592	1.4190	3.1576	1.4906	3.1592	1.4190	3.1588	1.7886
c	-	-	-	-	0.1547	0.1547	0.0558	0.0558	0.3206	0.3206

Table 10: *Auto-Mpg* data. Covariates are acceleration ($\mathbf{x}_1$), cylinders ($\mathbf{x}_2$), displacement ($\mathbf{x}_3$), horsepower ($\mathbf{x}_4$), model year ($\mathbf{x}_5$), weight ($\mathbf{x}_6$), and origin ($\mathbf{x}_7$). Best solutions out of 100 random starts, $G = 2$. $K = n/5$, and test set size = $n/10$.

8 Discussion

In the present paper, a computationally efficient constrained approach for clusterwise regression modeling was presented. Starting from the baseline idea of Seo and Lindsay (2010) and Seo and Kim (2012), we propose a new, computationally faster, data driven method to tune c. Based on the simulation study and the two empirical applications, we have shown that the proposed method compares very well with the RGD method in terms of accuracy of parameter estimates and cluster recovery, doing from twice up to ten times faster than the RGD approach.

In addition, we have demonstrated that the issue of unboundedness is not only an estimation problem, but seriously affects also the assessment of the number of components. We have implemented and deeply tested a formulation of the BIC, in the spirit of Fraley and Raftery (2007), using the (log) likelihood evaluated at the constrained solutions. To take into account the proportion of estimated scale entailed by the constrained estimator, we have also applied a Cerioli et al (2017)–type of correction in our context, counting the number of free scales as the proportion (1-c) of unconstrained variances. In the simulation study and the empirical applications, we have shown that both approaches to compute the BIC based on the constrained estimator yield a sounder assessment of the number of components than standard unconstrained approaches. The correction for the proportion of estimated unconstrained scale seems to improve over the constrained BIC

for ConK only with $k = 1$ in the $G \geq 3$ and small sample ($n = 100$) simulation conditions. We observed a similar behavior in the the *Auto–Mpg* data application.

When comparing our benchmark target - the homoscedastic normal variance - with another target - the OLS residuals variance - we observed an interesting link between how well the target summarizes the scale information and the scale balance: under the OLS target variance, which performed relatively worse that the homoscedastic normal variance, on average the c values were relatively closer to zero. In this respect, c can also be seen as an indicator of how suitable or unsuitable a given target variance is.

Having one tuning parameter to set (k) rather than two or more - as, for instance, in cross-validation schemes - limits the users' arbitrariness. In the real-data applications we observed that different values of k might determine different conclusions on the chosen scale balance. In general, based on our results and having the cross-validated method as benchmark, larger values of k (relative to the sample size) seems to be more favorable.

The estimated number of clusters may depend on the allowed difference among scatter parameters in specific clustering applications (Hennig and Liao, 2013). In the simulation study, we found that selecting the number of components with a BIC based on the estimates of the homoscedastic normal (HomN) algorithm might work in some cases (smaller sample size and components with similar class sizes). Nevertheless, there are situations, like the one we analyzed in our second application, where the BIC based on HomN overstates the number of components. Since neither of the two scenarios can be recognized *a priori*, we suggest the use of BIC based on the constrained solutions to correctly assess the number of components.

The equivariance property of our approach comes from the fact that the constraints are centered at a *target* variance, which we re-estimate if the dependent variable is transformed. Having an equivariant method for clustering is crucial. The reason is not limited to requiring that the final clustering remains unaltered as one acts affine transformation on the variable of interest: from a statistical point of view, no matter how the data come in, affine equivariance means that there is no data transformation ensuring better results, since the method is unaffected by changes of scale in the response variable. From a numerical perspective, in our experience, large scale differences in the data can cause trouble to optimization algorithms breaking in practice the affine equivariance property. However, we do not expect this to be an issue in our context which is univariate.

The approach from Cerioli et al (2017) that we have applied to clusterwise regression modeling

is based on the consideration that by imposing constraints on the component variances, we are not estimating the full model scales, but only some fraction of them. Still, how this relates to the notion of effective degrees of freedom requires further research (see, for instance, Zou, Hastie, and Tibshirani, 2007).

References

[1] Alfó, M. & Viviani, S. (2016). Finite Mixtures of Structured Models. In "Hennig C., Meila, M., Murtagh, F., Rocci, R. (Eds.), *Handbook of Cluster Analysis*," 217–240. Chapman & Hall: Boca Raton, FL.

[2] Arlot, S., & Celisse, A. (2010). Cross-validation procedures for model selection. *Statistics Surveys, 4, 40-79*.

[3] Bagirov, A. M., Ugon, J., & Mirzayeva, H. (2013). Nonsmooth nonconvex optimization approach to clusterwise linear regression problems. *European Journal of Operational Research, 229(1), 132-142.*

[4] 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 and Data Analysis, 41(3), 561-575.*

[5] Carbonneau, R. A., Caporossi, G., & Hansen, P. (2011). Globally optimal clusterwise regression by mixed logical-quadratic programming. *European Journal of Operational Research, 212(1), 213-222.*

[6] Cerioli, A., García-Escudero, L. A., Mayo-Iscar, A., & Riani, M. (2017). Finding the number of groups in model-based clustering via constrained likelihoods. *Journal of Computational & Graphical Statistics*, DOI: 10.1080/10618600.2017.1390469.

[7] Chen, J., & Tan, X. (2009). Inference for multivariate normal mixtures. *Journal of Multivariate Analysis, 100(7), 1367-1383.*

[8] Chen, J., Tan, X., & Zhang, R. (2008). Inference for normal mixtures in mean and variance. *Statistica Sinica, 18(2), 443.*

[9] Ciuperca, G., Ridolfi, A., and Idier, J. (2003). Penalized maximum likelihood estimator for normal mixtures. *Scandinavian Journal of Statistics, 30(1), 45-59.*

[10] Day N.E. (1969). Estimating the components of a mixture of two normal distributions, *Biometrika, 56, 463-474*.

[11] Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 39, 1-38.*

[12] DeSarbo, W. S., & Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification, 5(2), 249-282.*

[13] Di Mari, R., Rocci, R., & Gattone, S. A. (2017). Clusterwise linear regression modeling with soft scale constraints. *International Journal of Approximate Reasoning, 91, 160-178.*

[14] Fraley, C., and Raftery, A. E. (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification, 24(2), 155-181.*

[15] Frühwirth-Schnatter, S. (2006). *Finite mixture and Markov switching models.* Springer Science & Business Media.

[16] García-Escudero, L. A., Gordaliza, A., Greselin, F., Ingrassia, S., and Mayo-Iscar, A. (2017). Eigenvalues and constraints in mixture modeling: geometric and computational issues. *Advances in Data Analysis and Classification. DOI: https://doi.org/10.1007/s11634-017-0293-y*.

[17] Hathaway R. J. (1985). A constrained formulation of maximum-likelihood estimation for normal mixture distributions, *The Annals of Statistics, 13, 795-800*.

[18] Hennig, C. (2000). Identifiablity of models for clusterwise linear regression. *Journal of Classification, 17(2), 273-296.*

[19] Hennig, C., & Liao, T. F. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society, Series C, 62(3), 309-369.*

[20] Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 1, pp. 221-233).*

[21] Huber, P.J. (1981). *Robust Statistics.* John Wiley and Sons, New York.

[22] Hubert L., & Arabie P. (1985). Comparing partitions. *Journal of Classification, 2, 193-218*.

[23] Ingrassia S. (2004). A likelihood-based constrained algorithm for multivariate normal mixture models, *Statistical Methods & Applications, 13, 151-166*.

[24] Ingrassia S., & Rocci, R. (2007). A constrained monotone EM algorithm for finite mixture of multivariate Gaussians. *Computational Statistics & Data Analysis, 51, 5339-5351*.

[25] James, W., & Stein, C. (1961). Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 1961, pp. 361-379).*

[26] Keribin, C. (2000). Consistent estimation of the order of mixture models. *Sankhyā: The Indian Journal of Statistics, Series A, 49-66.*

[27] Kiefer J., & Wolfowitz J. (1956). Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *Annals of Mathematical Statistics, 27, 886906*.

[28] Kiefer, N. M. (1978). Discrete parameter variation: Efficient estimation of a switching regression model. *Econometrica 46, 427-434*.

[29] Koehler, A. B., & Murphree, E. S. (1988). A comparison of the Akaike and Schwarz criteria for selecting model order. *Applied Statistics, 187-195.*

[30] Leroux B.G. (1992). Consistent estimation of a mixing distribution. *Annals of Statistics, 20, 1350–1360.*

[31] Li, M., Xiang, S., & Yao, W. (2016). Robust estimation of the number of components for mixtures of linear regression models. *Computational Statistics, 31(4), 1539-1555.*

[32] Li, Y. F., & Zhou, Z. H. (2015). Towards making unlabeled data never hurt. *IEEE transactions on pattern analysis and machine intelligence, 37(1), 175-188.*

[33] Lindsay, B. G. (1995). Mixture Models: Theory, Geometry and Applications. *NSF-CBMS Regional Conference Series in Probability and Statistics, 5, i163.*

[34] McLachlan G.J., & Peel D. (2000). *Finite Mixture Models*. John Wiley & Sons, New York.

[35] Neykov, N., Filzmoser, P., Dimova, R., & Neytchev, P. (2007). Robust fitting of mixtures using the trimmed likelihood estimator. *Computational Statistics & Data Analysis, 52(1), 299-308.*

[36] Quandt, R. E. (1972). A new approach to estimating switching regressions. *Journal of the American Statistical Association, 67(338), 306-310.*

[37] Quandt, R. E., & Ramsey,J.B. (1978). Estimating mixtures of normal distributions and switching regressions. *Journal of the American Statistical Association, 73(364), 730-738*.

[38] Ritter, G. (2014). *Robust cluster analysis and variable selection.* Monographs on Statistics and Applied Probability 137, CRC Press.

[39] Rocci, R., Gattone, S.A., & Di Mari, R. (2017). A data driven equivariant approach to constrained Gaussian mixture modeling. *Advances in Data Analysis and Classification. DOI: 10.1007/s11634-016-0279-1.*

[40] Rusakov, D., & Geiger, D. (2005). Asymptotic model selection for naive Bayesian networks. *Journal of Machine Learning Research, 6, 1-35.*

[41] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 6(2), 461-464.*

[42] Seo B., & Kim D. (2012). Root selection in normal mixture models. *Computational Statistics & Data Analysis, 56, 2454–2470*.

[43] Seo, B., & Lindsay, B. G. (2010). A computational strategy for doubly smoothed MLE exemplified in the normal mixture model. *Computational Statistics and Data Analysis, 54(8), 1930-1941.*

[44] Smyth, P. (1996). *Clustering using Monte-Carlo cross validation.* In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, AAAI Press, pp. 126133.*

[45] Smyth, P. (2000). Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing, 10(1), 63-72.*

[46] Stone, C. (1984). Cross-validatory choice and assessment of statistical predictions. *Journal of Royal Statistical Society: Series B (Statistical Methodology), 36(2), 111-147.*

[47] Zou, H., Hastie, T., & Tibshirani, R. (2007). On the "degrees of freedom" of the lasso. *The Annals of Statistics, 35(5), 2173-2192.*