Advances in Data Analysis and Classification
https://doi.org/10.1007/s11634-018-0331-4

REGULAR ARTICLE

Assessing trimming methodologies for clustering linear regression data

Francesca Torti[1] · **Domenico Perrotta**[1] · **Marco Riani**[2] · **Andrea Cerioli**[2]

Received: 20 March 2017 / Revised: 23 April 2018 / Accepted: 20 July 2018

Abstract
We assess the performance of state-of-the-art robust clustering tools for regression structures under a variety of different data configurations. We focus on two methodologies that use trimming and restrictions on group scatters as their main ingredients. We also give particular care to the data generation process through the development of a flexible simulation tool for mixtures of regressions, where the user can control the degree of overlap between the groups. Level of trimming and restriction factors are input parameters for which appropriate tuning is required. Since we find that incorrect specification of the second-level trimming in the Trimmed CLUSTering REGression model (TCLUST-REG) can deteriorate the performance of the method, we propose an improvement where the second-level trimming is not fixed in advance but is data dependent. We then compare our adaptive version of TCLUST-REG with the Trimmed Cluster Weighted Restricted Model (TCWRM) which provides a powerful extension of the robust clusterwise regression methodology. Our overall conclusion is that the two methods perform comparably, but with notable differences due to the inherent degree of modeling implied by them.

Keywords Robust clustering · Clusterwise regression · Mixture modeling · TCLUST-REG · TCWRM · Monte Carlo experiment · MixSimReg

Mathematics Subject Classification 62-07 · 62-09 · 62Jxx

1 Introduction

In regression model-based clustering, outliers and noise can be handled in different ways. For example, one approach is to represent them with one (or more) finite mixture

✉ Francesca Torti
francesca.torti@ec.europa.eu

[1] Joint Research Centre (JRC), European Commission, Via Enrico Fermi 2749, 21027 Ispra, VA, Italy

[2] Department of Economics and Management, University of Parma, Via Kennedy 6, 43125 Parma, Italy

Published online: 30 July 2018

model component(s) additional to those for the meaningful part of the data (e.g. Poisson and t components are used by Banfield and Raftery 1993; Campbell et al. 1997; Dasgupta and Raftery 1998; Peel and McLachlan 2000). Then, least squares (LS) or maximum likelihood (ML) methods are applied component-wise to estimate all parameters.

Alternatively, it is possible to rely on normally distributed variables and downweight the contribution of atypical observations using, e.g., M-estimation. For example, Campbell (1984) follows this approach to update the components of a Gaussian mixture in the M step of the EM algorithm. In the same spirit, Hennig (2003) uses M-estimation in clusterwise regression, in combination with an iteratively reweighted algorithm with zero weight for the outliers.

In this paper we focus on a third approach based on two key ideas. One is the "impartial trimming" framework of Gordaliza (1991), consisting in removing from the dataset a fraction α of the "most outlying" data units, so that to obtain a trimmed set with lowest possible variation. The second idea, which distinguishes the approach from other trimming-based methods (e.g. Neykov et al. 2007), is to constrain the group scatters in order to make the optimization of the likelihood (which is unbounded otherwise) well-posed and to reduce the possibility of spurious solutions.

Trimming and constraints can be easily incorporated in a classical EM-type mixture estimation/classification algorithm avoiding (unlike the previous two approaches) specific distributions for the noise component or ad hoc solutions. This trimmed EM-type algorithm is clearly related to the "concentration steps" introduced by Rousseeuw and Van Driessen (1999) in the Fast algorithm of the Minimum Covariance Determinant estimator (FAST-MCD). The MCD estimator, proposed in the seminal work of Rousseeuw (1984), is one of the first affine equivariant and highly robust estimators of multivariate location and scatter, but it is thanks to the fast algorithm that the MCD found concrete applicability in problems of a certain size and complexity.

This computationally efficient framework based on trimming and scatter constraints was introduced for multivariate clustering by García-Escudero et al. (2008), in the TCLUST method. Then, more recently the method was extended to clusterwise regression in TCLUST-REG (García-Escudero et al. 2010), with the addition of a *second trimming* step introduced to mitigate the effect of high leverage points affecting the mixture components with extreme values in some of the explanatory variables. More precisely, to remove the effect of leverage points, in each step of the maximization procedure of TCLUST-REG a fixed proportion of observations that are most outlying in the regression space are trimmed. In many applications, the space of the explanatory variables can possibly include dummy variables. Although the TCLUST-REG algorithm (at least in our implementation) can fit a model with dummy variables, here we will not address this possibility, as the model properties would require a separate careful study and complicate the discussion of the results and the comparison with the standard case. For an overview of robust regression methods which treat dummies we recommend Perez et al. (2014), while Cerasa and Cerioli (2017) address the selection of dummy variables in an application framework similar to that analyzed in Sect. 5.6.

An alternative novel and attractive approach to attack the problems caused by remote observations in the space of the explanatory variables, is to assume a parametric specification for such variables and to incorporate it inside the likelihood, so that

leverage points are automatically removed. This leads to the so called *Trimmed Cluster Weighted Restricted Model* (TCWRM), where a Gaussian specification is generally assumed (the model, by García-Escudero et al. 2016, is illustrated in Sect. 2).

While there is a vast literature on TCLUST for multivariate observations, the properties of TCLUST-REG and TCWRM have received much less attention. In particular, in the context of TCLUST-REG, the positive effects of the second trimming step are known only for specific data configurations, but the benefits under general settings are less clear. For example it is not known if, and when, the second trimming step can be simply replaced by an increase of the percentage of units trimmed at the first step. Similarly, in the context of TCWRM, the robustness of the procedure to departures from the assumed distribution of the explanatory variables is not known. A first objective of this paper is to explore these and other properties of the two approaches with a simulation experiment, in which we consider the relation between the values of the key model parameters and some relevant data features. The datasets differ for the number of groups, degree of overlap between the groups, type of outliers, noise contamination schemes and distribution chosen for the explanatory variables. The parameters studied are the two trimming percentages and the restriction factor imposed on the ratio between the error variances of each pair of groups, for which we study the joint effect on the bias of the estimated model parameters and on the classification error of the final clustering.

A main indication emerging from our simulations and the analysis of some benchmark datasets is that the use of TCLUST-REG with a wrong percentage of observations removed at the second trimming step may deteriorate both the model estimates and the classification performance. This can happen for insufficient trimming in presence of concentrated bad leverage contamination, but also for excessive trimming if non-harmful or even good leverage points are removed. To address the problem we have introduced a new methodological option in TCLUST-REG, through the possibility to regulate the percentage of second level trimming during the estimation steps. This is done through an adaptive approach based either on the Forward Search (Riani et al. 2009), or on the Finite Sample Re-weighted MCD method of Cerioli (2010).

While the approach of TCWRM enables us to avoid the use of a second level of trimming, its higher flexibility is counterbalanced by the need of specifying a distribution for the covariates. Another major purpose of this paper is to compare TCWRM with the adaptive TCLUST-REG approach, in presence of different outlier patterns, possible misspecification of the distribution of the explanatory variables and different schemes for leverage points.

To the growth of the TCLUST literature has certainly contributed the availability of a comprehensive and well documented R package (Fritz et al. 2012). The same cannot be said for TCLUST-REG, although some R code is available on the web-site of the authors,[1] or under the TCWRM framework, for which R code at present is only available upon request from the authors of the method. To make TCWRM and TCLUST-REG accessible to a wider statistical community we provide a MATLAB implementation of the method in our FSDA toolbox (Riani et al. 2012, 2015) where, by simply using an option `alphaX`, the user can easily switch from TCWRM to TCLUST-

[1] http://www.eio.uva.es/~langel/software/tclustReg.r.

REG (see last paragraph of Sect. 2.3 for details). We also offer the possibility to choose between classification and mixture likelihood models within the same framework, by setting the parameter `mixt` respectively to $mixt = 0$ or $mixt \geq 1$. In addition, we are working on the integration of TCLUST-REG and TCWRM in a R interface to the main MATLAB FSDA functions for regression and multivariate analysis. We published in CRAN the first release of this R package, called fsdaR, in December 2017 (https://cran.r-project.org/web/packages/).

In our work, we have given special attention to the generation of the data for the simulation experiments. In order to control precisely the degree of overlap between the different regression hyperplanes of the generating mixture, we have extended *MixSim* to clusterwise regression. *MixSim* is a general, flexible and mathematically well founded framework originally introduced to generate mixtures of Gaussian distributions (Maitra and Melnykov 2010; Melnykov et al. 2012). We have implemented the new simulation framework, *MixSimReg*, in MATLAB and made it available in the FSDA toolbox together with a previous implementation of the original multivariate counterpart, already presented in Riani et al. (2015). Our implementations of *MixSim* and *MixSimReg* also include several data contamination schemes and other enrichments.

The structure of the paper is as follows. In Sect. 2 we recall the crucial ingredients of TCWRM, discuss its relationships with TCLUST-REG, and adaptive TCLUST-REG and illustrate how these procedures are implemeted inside toolbox FSDA. *MixSimReg* is described in Sect. 3. In Sect. 4 we give a simulation study in order to appreciate the role of the restriction factor, which controls the maximum allowed ratio among the group scatters of the residuals, and its relationships with the different types of trimming. In Sect. 5 we compare our adaptive TCLUST-REG with TCWRM in presence of correct and mispecified distribution of the explanatory variables, different degree of overlapping among components and different outlying schemes. Some brief conclusions are provided in Sect. 6.

2 TCWRM and adaptive TCLUST-REG: theoretical and computational aspects

So far, following a chronological approach, we focused on TCLUST-REG. In this section we start instead from Cluster Weighted Modeling (CWM), we then review the need for restrictions and trimming, leading to TCWRM, and we finally obtain TCLUST-REG as a particular case in which the distribution of the explanatory variables is not specified. In such a way, we are able to obtain a unified view of robust clustering for regression structures, where the two competing methods are related by the choice of one specific modelling option.

CWM is a mixture approach regarding the modelisation of the joint probability of data coming from a heterogeneous population which includes as special cases mixtures of regressions. In this approach both the explanatory variables (X) and the response (Y) are treated as random variates with joint probability density function, $p(y, x)$. This formulation was originally proposed by Gershenfeld (1997) and was developed in the context of media technology, in order to build a digital violin. CWM

was initially introduced under Gaussian and linear assumptions (Gershenfeld et al. 1999). The extension to other distributions is treated for example in Ingrassia et al. (2012).

More formally, let (X, Y) be the pair of random vector X and random variable Y defined on the probabilistic space Ω with joint probability distribution $p(x, y)$, where X is a p-dimensional input vector with values in some space $\mathcal{X} \subseteq R^p$ and Y a response variable having values in $\mathcal{Y} \subseteq R^1$. Thus, $(x, y) \in \mathcal{X} \times \mathcal{Y} \subseteq R^{p+1}$. If we suppose that the probabilistic space Ω can be partitioned into G disjoint groups, say $\Omega_1, \Omega_2, \ldots, \Omega_G$, CWMs belong to the family of mixture models and have density which can be written as

$$p(x, y, \theta) = \sum_{g=1}^{G} p(y|x, \theta_{y,g}) p(x, \theta_{x,g}) \pi_g, \tag{1}$$

where $p(y|x, \theta_{y,g})$ is the conditional density of Y given x in Ω_g which depends on the vector of parameters $\theta_{y,g}$, $p(x, \theta_{x,g})$ is the marginal density of X in Ω_g which depends on the vector of parameters $\theta_{x,g}$, and π_g reflects the importance of Ω_g in the mixture with the usual constraints $\pi_g > 0$ and $\sum_{g=1}^{G} \pi_g = 1$. Vector θ denotes the full parameters set $\theta = (\theta_{y,g}^T \ \theta_{x,g}^T)^T$. It is customary to assume that in each group g the conditional relationship between Y and x is

$$Y = \beta_g^0 + x^T \beta_g + \epsilon_g, \tag{2}$$

where $\epsilon_g \sim N(0, \ \sigma_g^2)$. β_g^0, $\beta_g = (\beta_{1g}, \beta_{2g}, \ldots, \beta_{pg})^T$ and σ_g are respectively the $p + 1$ regression parameters and the scale parameter referred to component g. With the linearity and normality assumption, the first two conditional moments of Y given x can be written as $E(Y|x, \beta_g^0, \beta_g, \sigma_g) = \beta_g^0 + x^T \beta_g$, $var(Y|x, \beta_g^0, \beta_g, \sigma_g) = \sigma_g^2$. If, in addition, we also assume that the X distribution is multivariate normal, that is

$$p(x; \theta_{x,g}) = \phi_p(x; \mu_g, \Sigma_g),$$

where $\phi_p(x, \mu_g, \Sigma_g)$ denotes the density of a p-variate Gaussian distribution, with mean vector μ_g and covariance Σ_g, model (1) becomes the so called linear Gaussian CWM and can be written as

$$p(x, y; \theta) = \sum_{g=1}^{G} \phi(y|\beta_g^0 + \beta_g^T x, \sigma_g^2) \phi_p(x; \mu_g, \Sigma_g) \pi_g. \tag{3}$$

It is interesting to notice that clustering around regression (DeSarbo and Cron 1988) can be seen as a special case of Eq. (3) by setting $\phi_p(x; \mu_g, \Sigma_g) = \phi_p(x; \delta)$, that is assuming the same distribution of X for all the components. In other words, in clustering around regression only the conditional distribution of $p(y|x)$ is specified while the distribution of the regressors is ignored. Equation (3) corresponds to a mixture of regressions with weights $\phi_p(x; \mu_g, \Sigma_g)$ depending not only on π_g but also on the covariate distribution in each component g. Let $\{x_i, y_i\}$, $i = 1, 2, \ldots, n$,

represent a i.i.d. random sample of size n drawn from (X, Y). This leads to define the following log-likelihood function to be maximized (mixture log-likelihood $L_{mixt}(\theta)$)

$$L_{mixt}(\theta) = \sum_{i=1}^{n} \log \left[\sum_{g=1}^{G} \phi(y_i | b_g^0, b_g^T x, s_g^2) \phi_p(x_i, m_g, S_g) p_g \right], \tag{4}$$

where $\theta = (p_1, \ldots, p_G, b_1^0, \ldots, b_G^0, b_1, \ldots, b_G, s_1^2, \ldots, s_G^2, m_1, \ldots, m_G, S_1, \ldots, S_G)$ is the set of parameters satisfying $p_g \geq 0$ and $\sum_{g=1}^{G} p_g = 1$, $b_g^0 \in R^1$ $b_g \in R^p$, $s_g^2 \in R^+$, $m_j \in R^p$ and S_j a p.s.d. symmetric $p \times p$ matrix. The optimal set of parameters based on this likelihood is

$$\widehat{\theta}_{\text{mixt}} = \arg\max_{\theta} L_{mixt}(\theta). \tag{5}$$

Once $\widehat{\theta}_{\text{Mixt}} = (\widehat{p}_1, ..., \widehat{p}_G, \widehat{b}_1^0, ..., \widehat{b}_G^0, \widehat{b}_1, ..., \widehat{b}_G, \widehat{s}_1^2, ..., \widehat{s}_G^2, \widehat{m}_1, ..., \widehat{m}_G, \widehat{S}_1, ..., \widehat{S}_G)$ is obtained, the observations in the sample are divided into G clusters by using posterior probabilities. That is, observation (x_i, y_i) is assigned to cluster g, if $g = \arg\max_l \phi(y_i | \widehat{b}_l^0, \widehat{b}_l^T x, \widehat{s}_l^2) \phi_p(x_i; \widehat{m}_l, \widehat{S}_l) \widehat{p}_l$.

In the so-called classification framework of model based clustering, the classification log-likelihood ($L_{cla}(\theta)$) to be maximized is defined as

$$L_{cla}(\theta) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig}(\theta) \log \phi(y_i | b_g^0, b_g^T x, s_g^2) \phi_p(x_i, m_g, S_g) p_g, \tag{6}$$

where

$$z_{ig}(\theta) = \begin{cases} 1 \text{ if } g = \arg\max_l \phi(y_i | \beta_l^0, \beta_l^T x, \sigma_l^2) \phi_p(x_i, \mu_l, \Sigma_l) \pi_l, & l = 1, 2, \ldots, G, \\ 0 \text{ otherwise.} \end{cases}$$

In this case, the optimal set of estimates is

$$\widehat{\theta}_{\text{cla}} = \arg\max_{\theta} L_{cla}(\theta) \tag{7}$$

and observation $(x_i \ y_i)$ is now classified into cluster g if $z_{ig}(\widehat{\theta}_{\text{cla}}) = 1$.

In the MATLAB toolbox FSDA, the user can easily specify, using input parameter `mixt`, which of the two likelihoods (4) or (6) is to be maximized. However, both these likelihoods suffer from three major problems: unboundedness, lack of robustness and presence of several local maxima. In the three subsections below, we tackle these problems and illustrate how the solution to these issues has been implemented inside FSDA.

2.1 Unboundedness

The target functions (4) and (6) are unbounded when no constraints are imposed on the scatter parameters. It is necessary therefore to define constraints on the maximization on the set of eigenvalues $\{\lambda_r(S_g)\}, r = 1, \ldots, p$, of the scatter matrices S_g by imposing

$$\lambda_{l_1}(S_{g_1}) \leq c_X \lambda_{l_2}(S_{g_2}) \quad \text{for every} \quad 1 \leq l_1 \neq l_2 \leq p \quad \text{and} \quad 1 \leq g_1 \neq g_2 \leq G$$

and to the variances s_g^2 of the regression error terms, by requiring

$$s_{g_1}^2 \leq c_y s_{g_2}^2 \qquad \text{for every} \qquad 1 \leq g_1 \neq g_2 \leq G$$

The constants $c_X \geq 1$ and $c_y \geq 1$ are real numbers (not necessarily equal) which guarantee that we are avoiding the cases $|S_g| \to 0$ and $\sigma_g^2 \to 0$. This makes the likelihood bounded; as a consequence spurious solutions are reduced as shown in García-Escudero et al. (2010). Inside FSDA these restrictions are controlled by using the (required input) parameter named `restrfact`. If `restrfact` is a scalar, it refers to c_y and controls the differences among group scatters of the residuals. If `restrfactor` is a vector with two elements, the first element refers to c_y and the second to c_X. The algorithm used to impose the restrictions is an efficient vectorized version without loops of the procedure described in Fritz et al. (2013).

2.2 Local maxima

In order to avoid to be trapped into local maxima, we start from several different initial random subsets and bring each of them to convergence. Each subset is obtained by generating $p \times G$ natural numbers from 1 to n and extracting the corresponding rows from the original set of data. For example, if $(p + 1) = 2$, $G = 3$ and $n = 100$, we randomly generate $2 \times 3 = 6$ natural numbers in the interval $1 \leq n \leq 100$; if the generated numbers are [5, 36, 58, 71, 80, 95], the subset will be formed by the rows in the original dataset with these six indexes. The number of subsets can be controlled in FSDA using the input parameter `nsamp`, which by default is equal to the minimum between 300 and n choose $(p + 1) \times G$. An optional parameter, `refsteps`, lets the user specify the maximum allowed number of iterations (concentration steps).

For each subset we immediately apply the eigenvalue restrictions in order to be sure that we are using an admissible value of the set of parameters θ. In order to let the user have a feeling about the stability of the obtained solution, we also provide in output the value of the target function in correspondence of each subset. Finally, if the user wishes to compare the results using different values of the restriction factors, our routine makes use of parallel computing tools and enables to preextract the list of subsets without having to recalculate them for each new value of `restrfactor`.

2.3 Lack of robustness and an adaptive trimming proposal

In the literature of robust regression it is widely known the effect of both vertical outliers in Y and outliers in X. Robustness can be achieved by discarding in each step of the maximization procedure a proportion of units equal to α_1, associated with the smallest contributions to the target likelihood. More precisely, for example in the context of mixture modeling, the TCWRM parameter estimates are based on the maximization, over the parameters $p_g, m_g, S_g, b_g^0, b_g, s_g^2$, of the following trimmed likelihood function

$$L_{mixt}(\theta|\alpha_1, c_y, c_X) = \sum_{i=1}^{n} z^*(x_i, y_i) \log \left[\sum_{g=1}^{G} \phi(y_i|b_g^0, b_g^T x, s_g^2) \phi_p(x_i, m_g, S_g) p_g \right] \quad (8)$$

In 8, $z^*(\cdot, \cdot)$ is a 0-1 trimming indicator function which tells us whether observation $(x_i \ y_i)$ is trimmed off $(z^*(x_i \ y_i) = 0)$ or not $(z^*(x_i \ y_i) = 1)$. A fixed fraction α_1 of observations can be unassigned by setting $\sum_{i=1}^{n} z(x_i \ y_i) = [n(1 - \alpha_1)]$. TCLUST-REG (García-Escudero et al. 2010) can be considered as a particular case of TCWRM in which the contribution to the likelihood of $\phi_p(x_i, m_g, S_g)$ is set equal to 1. In other words, in TCLUST-REG only the conditional distribution of $p(y|x)$ is modelled/specified.

However, if the component $\phi_p(x_i, m_g, S_g)$ is discarded, α_1 just protects against vertical outliers in Y, since these data points have small $\phi(y_i|b_g^0, b_g^T x, s_g^2) p_g$ values, but it has no effect in diminishing the effect of outliers in the X space. Therefore, if we adopt a TCLUST-REG approach, it is necessary to consider [as done by García-Escudero et al. (2010)] a second trimming step, which discards a proportion α_2 of the observations after taking into account the values of the explanatory variables of the observations surviving to the first trimming step. More in detail, the second trimming step applies MCD on the explanatory variables space so that to trim a fraction α_2 of observations with the largest robust distances. The usual solution in TCLUST-REG is to fix α_2 in advance, although there is no established indication of the link between this proportion and the breakdown properties of the overall methodology. Furthermore, in the following sections we show that we may end up in a serious deterioration of the model parameter estimates and in an increase of the classification error if we impose a value of α_2 which is not well tuned. To improve the performance of TCLUST-REG, we instead propose to select α_2 adaptively from the data. This means that the robust distances are compared with the confidence bands at a selected confidence level, and the observations with distances exceeding the bands are trimmed. In this case the multivariate outlier detection procedure proposed by Cerioli (2010), based on the reweighted MCD estimator (Rousseeuw and Van Driessen 1999), or the Forward Search (Riani et al. 2009) can be used at each concentration step of each starting subset. The observations surviving to the two trimming steps are then used for updating the regression coefficients, weights and scatter matrices. We refer to the new version of method as *adaptive TCLUST-REG*. As suggested by one of the referees, a similar adaptive approach could be applied also to the first trimming step (Dotto et al. 2018, discuss the case in the multivariate context). We prefer leaving this interesting

extension to future work, so that to focus on the second trimming step, which is less studied in the literature.

Clearly, TCWRM enables us to model the marginal distribution of X, provides high flexibility to the model and automatically enables us to discard the observations which are atypical also in the space of the explanatory variables, because they will have a very small value of $\phi_p(x_i, m_g, S_g)$ and thus a small likelihood contribution $\phi(y_i|b_g^0, b_g^T x, s_g^2)\phi_p(x_i, m_g, S_g)p_g$. The higher flexibility of TCWRM, however, is counterbalanced by the additional complexity of the model, and the need of specifying a distribution for X. TCWRM seems to be more suitable when the sample size of the components is large. In the next sections we will see an example of cases in which, due to the low sample size (and natural holes in the distribution) the use of TCWRM may lead to find spurious components. One of the purposes of our work is thus to compare the TCWRM approach with adaptive TCLUST-REG.

In FSDA α_1 is a required input parameter, called `alphaLik` to stress that it is referred to the likelihood contribution. Parameter α_2 is called `alphaX` in order to stress that it is referred to outliers in the X space. If $0 \leq$ `alphaX` ≤ 0.5, TCLUST-REG is used and this parameter indicates the fixed proportion of units subject to second level trimming. In particular, if `alphaX` $= 0$ there is no second-level trimming. If `alphaX` is in the interval (0.5, 1), adaptive TCLUST-REG is used and this parameter indicates a Bonferronized confidence level to be used to identify the units subject to second level trimming. If $p > 1$, the default estimator which is used is the forward search, on the other hand, if $p = 1$ we use a reweighted MCD as modified by Cerioli (2010). Finally, if `alphaX` is equal to 1, TCWRM is used and the user can supply the value of c_X as the second element of the other input parameter `restrfact`.

2.4 Choice of function parameters and tuning constants

These methods entail suitable values for key parameters such as G, c_y, c_X, α_1 and α_2, and algorithmic tuning constants that are often overlooked. The latter include `nsamp`, `refsteps` and convergence tolerances used to attain the desired restrictions or check when a change in the objective function is small enough to stop the optimization process. Cerioli et al. (2018) have recently proposed a fully automatic approach to choose G and other key parameters. Nevertheless, the choice should always exploit possible subject matter knowledge about problem and data, as we will see in our motivating examples and case studies (Sects. 4, 5). The experience in our application domain (case study 5.6) is that the clustering obtained with a reasonable inflation of the number of groups is in general as informative as a clustering with the "correct" number of groups. The results seem rarely sensitive to the choice of the tuning constants, which in our FSDA implementation are chosen to cover the most typical scenarios. Frameworks for monitoring the effects of parameters and tuning constants have been discussed in clustering by Cerioli et al. (2017) and some discussants (García-Escudero et al. 2017b; Farcomeni and Dotto 2018; Perrotta and Torti 2018).

3 Simulating regression mixture data with MixSimReg

Our simulations use regression mixture data generated with an approach that allows to control pre-specified levels of average or/and maximum overlap between pairs of mixture components. The distinctive aspects of this approach are that the pairwise overlap has a natural formulation in terms of sum of the two misclassification probabilities, and that the generating model parameters are automatically derived to satisfy the prescribed overlap values instead of being given explicitly by the experimenter.

The approach, known as *MixSim* (Maitra and Melnykov 2010; Melnykov et al. 2012), was originally introduced in the multivariate context to generate samples from Gaussian mixture models $\sum_{g=1}^{G} \pi_g \phi(y; \mu_g, \Sigma_g)$ defined in a v-variate space, for given data vector y, group occurrence probabilities (or mixing proportions) π_g, group centroids μ_g and group covariance matrices Σ_g. If i and j ($i \neq j = 1, ..., G$) are clusters indexed by $\phi(y; \mu_i, \Sigma_i)$ and $\phi(y; \mu_j, \Sigma_j)$ with occurrence probabilities π_i and π_j, then the *misclassification probability with respect to cluster i* (i.e. conditionally on y belonging to cluster i) is defined as

$$w_{j|i} = Pr[\pi_i \phi(y; \mu_i, \Sigma_i) < \pi_j \phi(y; \mu_j, \Sigma_j)]. \tag{9}$$

Similarly for $w_{i|j}$, the *overlap between groups i and j* is then given by

$$w_{j|i} + w_{i|j} \qquad i, j = 1, 2, ..., G.$$

This section illustrates our extension of *MixSim* to regression mixtures and the implementation of the new *MixSimReg* framework in our FSDA toolbox. The starting point is the redefinition of the misclassification probability (9), which in clusterwise regression (given that y is univariate) becomes

$$w_{j|i} = Pr[\pi_i \phi(y; \mu_i, \sigma_i^2) < \pi_j \phi(y; \mu_j, \sigma_j^2)]. \tag{10}$$

The group centers are now defined as $\mu_i = \overline{x}_i' \beta_i$ and $\mu_j = \overline{x}_j' \beta_j$, where $\overline{x}_i$ ($\overline{x}_j$) is the expected value of the explanatory variable distribution for group i (j) and β_i (β_j) is the vector of regression coefficients for group i (j).

In our implementation, the distribution of the elements of vectors β_i (β_j) can be Normal (with parameters μ_β and σ_β), HalfNormal (with parameter σ_β) or Uniform (with parameters a_β and b_β). Similarly for the distribution of the elements of x_i (x_j). However, while the parameters of the distributions are the same for all elements of β in all groups, the parameters of the distribution of the elements of vectors x_i (x_j) can vary for each group and each explanatory variable. For example, it is possible to specify that the distribution of the second explanatory variable in the first group is $U(2, 3)$ while the distribution of the third explanatory variable in the second group is $U(2, 10)$.

The key result of Maitra and Melnykov (2010) is a closed expression for the probability of overlapping $w_{j|i}$ defined in Eq. (9), which is shown to be (for multivariate Gaussian mixtures) the cumulative distribution function (cdf) of a linear combination

of non central χ^2 distributions U_l with 1 degree of freedom plus a linear combination of $W_l \sim N(0, 1)$ random variables:

$$\omega_{j|i} = Pr_{N_p(\mu_i,\Sigma_i)}\left[\sum_{\substack{l=1\\ l:\lambda_l\neq 1}}^{v}(\lambda_l - 1)U_l + 2\sum_{\substack{l=1\\ l:\lambda_l=1}}^{v}\delta_l W_l \leq \sum_{\substack{l=1\\ l:\lambda_l\neq 1}}^{v}\frac{\lambda_l\delta_l^2}{\lambda_l - 1} - \sum_{\substack{l=1\\ l:\lambda_l=1}}^{v}\delta_l^2 + \log\frac{\pi_j^2|\Sigma_i|}{\pi_i^2|\Sigma_j|}\right] \tag{11}$$

The cdf is evaluated in a point c, which is the second term of the inequality. The expression and, in particular, the non-centrality parameter of the non central-χ^2 distributions U_l,

$$\lambda_l^2\delta_l^2(\lambda_l - 1)^2 \quad \text{with} \quad \delta_l = \gamma_l^{'}\Sigma_i^{-0.5}(\mu_i - \mu_j),$$

depend on the eigenvalues λ_l and eigenvectors γ_l of the spectral decomposition of matrix $\Sigma_{j|i} = \Sigma_i^{0.5}\Sigma_j^{-1}\Sigma_i^{0.5}$.

Let us now simplify the framework to clusterwise regression models with one response variable. The model becomes univariate, therefore in Eq. (11) v reduces to 1 and the summations disappear. The dimension reduction implies these additional simplifications:

- There is only one eigenvalue $\lambda_l = \sigma_i^2/\sigma_j^2 \neq 1$ and one eigenvector $\gamma_l = 1$;
- $\delta_l = \dfrac{\mu_i - \mu_j}{\sigma_i}$;
- There is a single non-central χ^2 to compute (lower or upper tail of the cdf):

$$U_l \sim \chi^2\left(1, \sigma_i^2\left[\frac{\mu_i - \mu_j}{\sigma_i^2 - \sigma_j^2}\right]^2\right).$$

This is a considerable simplification, because the computation of the linear combination of non-central χ^2 in Eq. (11) uses the expensive algorithm AS 155 of Davies (1980), discussed also in Riani et al. (2015).

With these simplifications Eq. (11), for the general *non homogeneous clusters case* in which $\sigma_i^2 \neq \sigma_j^2$, reduces to

$$\begin{aligned}\omega_{j|i} &= Pr_{N(\bar{x}'\beta_i,\sigma_i)}\left[\left(\frac{\sigma_i^2}{\sigma_j^2} - 1\right)U_l \leq \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 - \sigma_j^2} + \log\frac{\pi_j^2\sigma_i^2}{\pi_i^2\sigma_j^2}\right] \\ &= Pr_{N(\bar{x}'\beta_i,\sigma_i)}\left[U_l \leq \frac{\sigma_j^2(\mu_i - \mu_j)^2}{(\sigma_i^2 - \sigma_j^2)^2} + \frac{\sigma_i^2}{\sigma_i^2 - \sigma_j^2}\log\frac{\pi_j^2\sigma_i^2}{\pi_i^2\sigma_j^2}\right].\end{aligned} \tag{12}$$

Note that the equation holds for $\sigma_i^2 > \sigma_j^2$ and that the inequality inverts for the symmetric case where $\sigma_i^2 < \sigma_j^2$. Now, if we assume *homogeneous clusters*, i.e. if $\sigma_i^2 = \sigma_j^2$, the eigenvalue becomes 1 and therefore the contribution to both sides of the general equation comes only from the second sum term. With this additional simplification Eq. (12) reduces to:

$$\begin{aligned} \omega_{j|i} &= Pr_{N(\overline{x}'\beta_i,\sigma_i)}\left[2\delta_l N(0,1) \leq -\delta_l^2 + \log\frac{\pi_j^2}{\pi_i^2}\right] \\ &= \Phi\left[-\frac{1}{2}\left[\frac{|\mu_i - \mu_j|}{\sigma}\right] + \log\left(\frac{\pi_j}{\pi_i}\right)(1/|\mu_i - \mu_j|)\right] \end{aligned} \tag{13}$$

Note that in this case there is only the cdf of a normal distribution to compute.

Our software implementation of the framework is very flexible. We briefly discuss here the main options and parameters. One of the key output produced by *MixSimReg*, once the user specifies G, p and the presence of the intercept, is the matrix $(G \times G)$ containing the misclassification probabilities $w_{j|i}$, called `OmegaMap`. Its diagonal elements are equal to 1 while those for $i \neq j$ are `OmegaMap(i,j)` $= w_{j|i}$. The user typically specifies as input a desired average or maximum overlap, which are respectively `BarOmega` (defined as the sum of the off diagonal elements of `OmegaMap` divided by $G(G-1)/2$) and `MaxOmega` (defined as $\max(w_{j|i} + w_{i|j})$, for $i \neq j = 1, 2, ..., G$). Together with the average or maximum overlap, optionally the user can also specify a desired standard deviation for the overlap, `StdOmega`. The important restriction factor, specifying the maximum ratio to allow between the largest σ_j^2 and the smallest σ_j^2, redwith $j = 1, \ldots, G$, which are generated, is given in option `restrfactor` as scalar in the interval $[1, \infty]$.

The output produced by *MixSimReg* includes the vector of length G containing the mixing proportions, `Pi`, the $((p+intercept) \times G)$ matrix containing (in each column) the regression coefficients for each group, `Beta` and the $(G \times G)$ matrix containing the variances for the G groups, `S`. These mixture model parameters provided by *MixSimReg* are the key input variables of function `simdatasetreg`, which generates a simulated dataset with the desired statistics. Component sample sizes are produced as a realization from a multinomial distribution with probabilities given by mixing proportions `Pi`. The function `simdatasetreg` also requires the specification of a structure `Xdistrib` specifying how to generate each explanatory variable inside each group, as commented above, and of course a desired number of data points n.

To make a dataset more challenging for clustering, a user might want to simulate noise variables or outliers. Parameter `nnoise` specifies the desired number of noise variables. If an interval `int` is specified, noise will be simulated from a Uniform distribution on the interval given by `int`. Otherwise, noise will be simulated uniformly between the smallest and largest coordinates of mean vectors. `nout` specifies the number of observations outside (1 - `alpha`) ellipsoidal contours for the weighted component distributions. Outliers are simulated on a hypercube specified by the interval `int`. A user can apply an inverse Box-Cox transformation of y providing a coefficient `lambda`. The value 1 implies that no transformation is used for the response.

```
n = 100;          % number of units
G = 2;            % number of components
intercept = 0;    % no intercept
p = 1;            % number of explanatory variables without intercept
bo = 0.01;        % average overlap (BarOmega)
rfy = 100;        % restriction factor for y
betadistrib = 1;  % regression coefficients distribution is N(0, 1)
Xdistrib = struct;              % initialize Xdistrib
Xdistrib.type = 'Uniform';      % x-distribution is uniform
Xdistrib.intercept = intercept; % change default intercept setting

MSout    = MixSimreg(G, p+intercept, 'BarOmega', bo, 'Xdistrib', Xdistrib, ...
                     'betadistrib', betadistrib, 'restrfactor',rfy);
[y,X,id] = simdatasetreg(n,MSout.Pi, MSout.Beta, MSout.S, MSout.Xdistrib);
```

Fig. 1 Code used to generate the mixture data of Fig. 3

```
n = 200; G = 3; bo = 0.1;       % redefinition of common parameters
intercept =  1;                 % intercept is now present
Xdistrib = struct;              % re-initialize Xdistrib
Xdistrib.type = 'Normal';       % x-distribution is now Normal
Xdistrib.intercept = intercept; % redundant because default is 1
nnoiseunits = n*(7.5/100);      % number of outliers is 15

MSout      = MixSimreg(G, p+intercept, 'BarOmega', bo, 'Xdistrib', Xdistrib,...
                       'betadistrib', betadistrib, 'restrfactor', rfy);
[y, X, id] = simdatasetreg(n, MSout.Pi, MSout.Beta, MSout.S, MSout.Xdistrib,...
                       'noiseunits', nnoiseunits);
```

Fig. 2 Code used to generate the mixture data and the outliers of Fig. 10

Figures 1 and 2 report code fragments used to generate the data of Figs. 3 and 10, with option `nnoiseunits` used in the latter to contaminate the data.

4 Motivating examples

This section prepares the ground for the next central one, with an illustration of the role of the restriction factor and its relation with the two types of trimming. In fact, in order to conduct a fair assessment of the performances of different scatter-constrained methods, it is crucial to define a proper setting for the relative cluster scatters, i.e. to rely on reasonable values for the restriction factor. This is the objective of Sect. 4.1. Then, the relationship of the restriction factor with the different types of trimming is illustrated in Sect. 4.2. The examples are based on different simulated data configurations.

4.1 The restriction factor

The two scatterplots of Fig. 3 represent regression mixture data generated using *MixSimReg* (see first code fragment in Sect. 3) for a model without intercept, two components, a restriction factor which does not have to exceed 100 and average over-

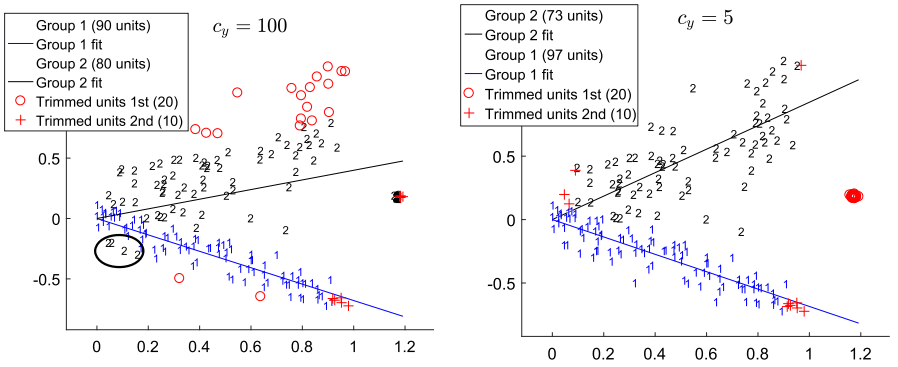

Fig. 3 Regression mixture data with a (upper) component more dispersed than the other and a concentrated contamination between them. The dataset is analysed with TCLUST-REG (on 300 subsets) with a large and a small restriction factor: $c_y = 100$ (left panel) and $c_y = 5$ (right panel). This and similar results indicate that large values of c_y can disrupt the main trimming step and deteriorate the model estimates and the final classification

lap $\bar{\omega} = 10\%$. A 10% concentrated contamination of potential high leverage units has been added between the two components. The empirical ratio between the two residual variances is 4.41 (true c_y).

The classifications in the two panels are obtained by TCLUST-REG with the first and second trimming levels set to $\alpha_1 = 10\%$ and $\alpha_2 = 5\%$ respectively. In the left panel TCLUST-REG was run with restriction factor $c_y = 100$, while in the right panel a much lower value ($c_y = 5$) was used. There is a visible side effect in using $c_y = 100$. The variability granted to the upper component is so large that some units that are clearly part of the more concentrated lower group (identified by a black ellipse) are wrongly assigned. The same happens to some contaminant units. As a consequence, the fit of the resulting Group '2' drops and a strip of units located in the upper part of the plot are trimmed (red circles or light grey, for prints in grey-scale). We have observed that, for this dataset, very similar (if not identical) bad classifications are obtained already for $c_y \geq 9$ (approximately twice the value of the true one).

This example indicates that in clusterwise regression the choice of the restriction factor requires attention, and that if parameter c_y is left too high it may lead to undesirable solutions. Of course, results also depend on the combined effects of the restriction factor and the level of the two trimming steps, which in our opinion the literature has not sufficiently studied so far. This is what we address in the next section.

4.2 Trimming

García-Escudero et al. (2010) motivated the need of a second trimming step for TCLUST-REG in relation to data patterns like in Fig. 4, formed by non-overlapping groups generated in almost disjoint ranges (of the explanatory variables), by lines (or hyperplanes, in more dimensions) intersecting at angles not necessarily close to orthogonal. Without a second trimming step (left panel) TCLUST-REG wrongly classifies, perforce, a group of units located at the intersection of the two fitted lines. The

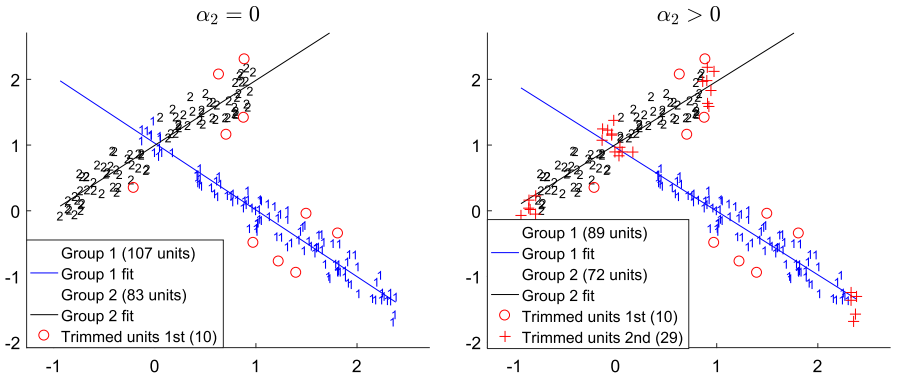

Fig. 4 TCLUST-REG applied to non-overlapping data groups generated by almost perpendicular linear components (García-Escudero et al. 2010). The second trimming step (right panel) avoids classifying incorrectly units in the intersection of the two fitted lines

number of misclassified units typically increases for angles departing from orthogonality and larger component variances. The second trimming step (right panel) draws these units out of the estimation and classification phases. In more general settings, where data groups overlap, the effect of the second trimming step has not been studied yet in the literature. This section presents findings based on a benchmark experiment, for different data configurations and parameter settings.

The data are generated with *MixSimReg* from a mixture of $G = 2$ components with average overlap $\bar{\omega} = 0.01$. Table 1 reports the mean misclassification rate reduction (ΔCE) determined by the application of a second trimming step fixed at level $\alpha_2 = 12\%$, after a first trimming of $\alpha_1 = 12\%$ or $\alpha_1 = 20\%$ (column α_1). The mean classification errors obtained after both trimming steps are also reported (CE columns); for each run the error is computed as the fraction of incorrectly assigned units among those generated by the mixture model (therefore the contaminants are not counted). This scheme is repeated for a small and a large restriction factor value (columns $c_y = 5$ and $c_y = 100$); the same restriction factor value is used both to generate the data and to apply TCLUST-REG. The rows of the table refer to different positions of a 10% contamination added to the data 'between' or 'below' the two component lines and with either large or small leverage, i.e. for independent variable values 'far away' from, or 'close' to, those of the data mixture. More precisely, 'far away' and 'close' mean that contamination is positioned respectively 200% and 20% far from the maximum x value of the original (uncontaminated) mixture data. Note that the contamination percentage (10% of the original data) is lower than the first trimming level ($\alpha_1 = 12\%, 20\%$) set for TCLUST-REG in each simulation. The main conclusions that can be drawn from Table 1 are that:

– In all simulation settings TCLUST-REG produces consistent results in terms of final classification errors (the values of CE are comparable). This means that TCLUST-REG, from a clustering perspective, is resilient to slight modifications of function parameters, tuning constants and the two trimming levels.

Table 1 TCLUST-REG simulation for 1000 datasets of 100 units generated from a mixture of 2 components with $\bar{\omega} = 0.01$ average overlap

Row	Contamination position y	Contamination position x	trim α_1	CE: $c_y = 100$ (%)	CE: $c_y = 5$ (%)	Δ *CE*: $c_y = 100$ (%)	Δ *CE*: $c_y = 5$ (%)
1	Below lines	Close	0.12	6.21	6.37	15.06	15.76
2	Below lines	Close	0.20	6.79	6.92	13.77	14.80
3	Below lines	Far away	0.12	6.38	6.33	13.25	17.59
4	Below lines	Far away	0.20	7.19	6.99	15.94	13.40
5	Between lines	Close	0.12	6.83	6.57	7.96	13.28
6	Between lines	Close	0.20	7.23	6.94	16.70	15.71
7	Between lines	Far away	0.12	6.08	6.27	15.69	15.87
8	Between lines	Far away	0.20	6.39	6.81	16.37	14.57

The table rows differ for the position of a 10% contamination, added 'between' or 'below' the two mixture lines and with either large or small leverage (i.e. 'far away' or 'close' w.r.t. the maximum of the data mixture x-range). ΔCE is the mean misclassification rate reduction obtained by applying a fixed $\alpha_2 = 12\%$ second trimming step after a first trimming at level $\alpha_1 = 12\%$ or $\alpha_1 = 20\%$ (column 'trim α_1'). CE is the mean classification error obtained after both trimming steps. Columns with $(c_y = 5)$ and $(c_y = 100)$ refer to the restriction factor chosen in the simulation

Table 2 Misclassification rate reduction ΔCE obtained by passing from ($\alpha_1 = 20\%, \alpha_2 = 0\%$) to ($\alpha_1 = 12\%, \alpha_2 = 12\%$)

Contamination position y	Contamination position x	ΔCE ($c_y = 100$) (%)	ΔCE ($c_y = 5$) (%)
Below lines	Close	21.36	20.23
Below lines	Far away	20.49	21.40
Between lines	Close	21.05	21.59
Between lines	Far away	25.46	21.58

Classification errors are larger when trimming is polarized at the first step

- The classification error CE is systematically lower for $\alpha_1 = 12\%$ than for $\alpha_1 = 20\%$. Given that the true contamination level is 10%, this implies that better results are obtained using α_1 close to the true one.
- The ΔCE values are all positive. This means that the second trimming step in general improves the classification.

One could also wonder if the second trimming can be replaced by a generous first trimming step. Table 2 shows, for some of the data settings of Table 1, that a 20% trimming polarized at the first step ($\alpha_1 = 20\%, \alpha_2 = 0\%$) is considerably less effective than more balanced combinations of first and second trimming steps, such as the ($\alpha_1 = 12\%, \alpha_2 = 12\%$) pair considered in Table 2.

Figure 5 shows the boxplots of the 1000 values of the classification error (left panels) and regression slopes bias (right panels) obtained in the simulation, for two data configurations of Table 1 (chosen because prototypical of the simulation study): top panels refer to the configuration in row 5 of the table; bottom panels refer to the configuration in row 6 of table. Each panel contains the boxplots for the two restriction factors used. The bias is computed for each group. The distribution of the values is very asymmetric. Remarkably, the larger restriction factor ($c_y = 100$) produces more cases of large classification errors and slope biases (TCLUST-REG failures) than the smaller restriction factor ($c_y = 5$). This is in line with the motivating example of Sect. 4.1 where it was shown that in some cases large values of c_y produce degenerated model estimates and classifications. For this reason, in the simulation experiments of the next section we fix the restriction factor to 5.

5 Adaptive TCLUST-REG vs TCWRM

We have seen that in TCLST-REG the second trimming step in general has beneficial effects on the classification. However, setting α_2 requires to know with good approximation the true data contamination, in particular the percentage of the high leverage units to be trimmed. The TCWRM approach of García-Escudero et al. (2017a) (Sect. 2) is a solution that moves the focus from prior knowledge on the contamination percentages to prior knowledge on the distribution of the covariates. In the adaptive TCLUST-REG approach that we propose in this work, instead of trimming a fixed percentage α_2 of observations associated with the largest robust Mahalanobis distances in the X space, we trim those lying outside a Bonferroni-corrected confidence band,

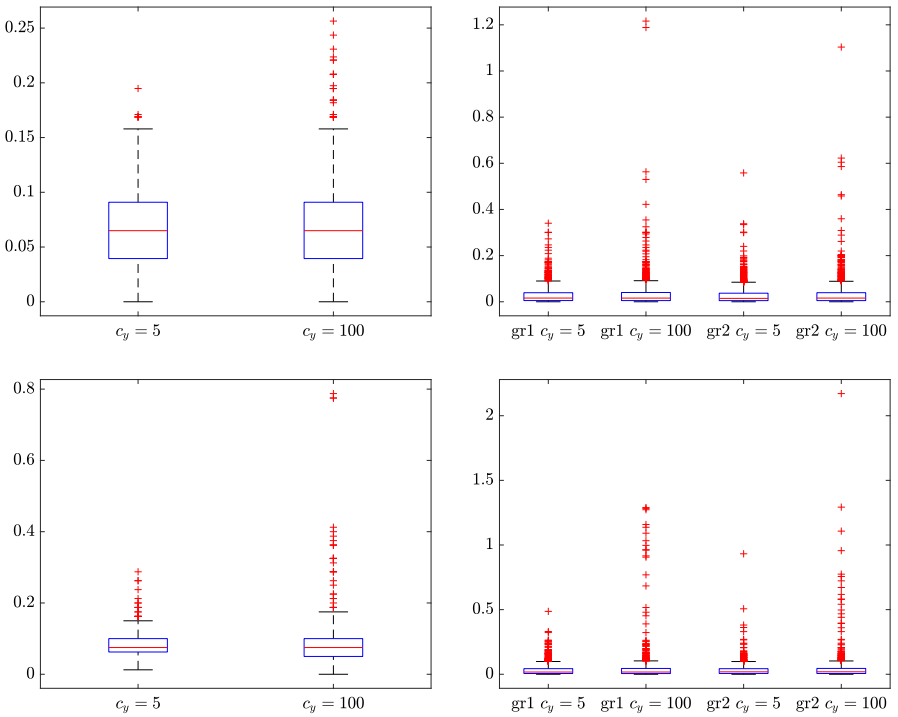

Fig. 5 Boxplots of the classification errors (left panels) and group-wise slope biases (right panels) obtained in the 1000 data configurations corresponding to lines 5 (top panels) and 6 (bottom panels) of Table 1

calculated at a confidence level specified by the user. The identification of the units is done using either the Finite Sample Re-weighted MCD rule (Cerioli 2010) or the Forward Search (Riani et al. 2009) for their good trade-off between robustness and efficiency. TCLUST (García-Escudero et al. 2008), which in the univariate case is equivalent to the MCD, can be also used.

In this section we want to illustrate the properties of the adaptive TCLUST-REG approach in comparison with TCWRM. Our aim is not to establish the superiority of a specific method under very general settings, which would require many simulation exercises based on tables of relevant summary statistics across many values of the model parameters, tuning constants and, for TCWRM, different model assumptions. To be exhaustive, this would require a separate thorough study. We therefore discuss the key properties of the methods and some crucial differences using results obtained in:

- a series of five focused case studies, based on simulated data patterns of increasing complexity and one real dataset;
- a classical simulation exercise, conceived to confirm and generalize the main conclusions of the case studies. In line with the objective of the paper, the focus is on the capacity of the methods to treat leverage observations.

5.1 Case studies setting

Four case studies are based on experiments of 100 replicates, using *MixSimReg* to generate general data patterns (case studies 1 and 2) or mimicking known datasets from the literature (case studies 3 and 4, used by García-Escudero et al. 2017a, to demonstrate the good behavior of TCWRM). The fifth case study is based on real data (case study 5). More precisely:

- Case studies 2 and 4 are more complex variants of case studies 1 and 3, respectively. In fact, case study 2 is designed with three components instead of two, and with a higher level of overlap among the components.
- In case study 4, the distribution of the independent variable is $\chi 2$ distributed. This is done to test the capacity of TCWRM to cope with deviations from the normality assumed in the current implementation.

In the following we present in details the five case studies. In addition, Table 3 summarizes their distinctive features and the main results.

Adaptive TCLUST-REG and TCWRM models are both run by optimizing the classification likelihood function (`mixt=0`) using 300 random subsets (`nsamp=300`) and 10 concentration steps (`refsteps=10`). In order to avoid potential confounding effects, the number of groups and the restriction factor, which are common to both models, are set to the true values, i.e., for case studies 1-4, those used by *MixSimReg* for generating the data. In particular, the restriction factor is $c_y = 5$. In case studies 1, 3 and 4 the first trimming level is equal to the contamination percentage; a larger value is used in case study 2. The confidence level of the flexible second trimming step, specific of the new adaptive TCLUST-REG, is set to `alphaX = 0.99`. The distribution of the explanatory variable in TCWRM is set differently in each case study, while the restriction imposed on the explanatory variable (which distinguishes TCWRM) is set to $c_X = 5$ (`restrfact(2)=5`). To evaluate the overall performances of the methods in the case studies, we compute the Adjusted Rand Index (ARI) between the true partition and the classifications produced by Adaptive TCLUST-REG and TCWRM in the 100 replicates.

5.2 Case study 1

We start with a simple data configuration setting, where 100 datasets of 100 units each are generated with *MixSimReg* from a 2-components mixture model with one Uniformly distributed explanatory variable and a $\bar{\omega} = 0.01$ average overlap.

Ten per cent of the data are contaminated, producing 10 outliers positioned at the top right part of the xy-data range. The top panels of Fig. 6 show two of the 100 datasets (titled A and B) generated in the simulation experiment, with the true classification of the units. These datasets have been chosen for their representativeness of the overall simulation study. The structure of dataset A (left panels) is well captured by TCWRM, while Adaptive TCLUST-REG fails to trim the contaminants. With dataset B both methods remove the contaminants and TCWRM produces a slightly better fit.

Figure 7 reports the boxplots of the ARI values (left panel) and the difference between the ARI values obtained with TCWRM and Adaptive TCLUST-REG (right

Table 3 Main features of the 5 case studies numbered in Column 1. Column 2: total number of observations, including outliers

Case study	Total obs.	Outl.	G	$\bar{\omega}$	X	α_1	Notes
1	100	10	2	0.01	U(0,1)	0.10	Quite simple case. Best ARI values with Adaptive TCLUST-REG and TCWRM. Slightly smaller ARI for TCLUST-REG.
2	215	15	3	0.1	N(0,1)	0.10	Similar to case 1 but more complex. Slightly better ARI for Adaptive TCLUST-REG.
3	200	20	2	.	$N(3.2, 4.4)$ and $N(2, 2.2)$	0.10	Case taken from García-Escudero et al. (2017a). Slightly better ARI for TCWRM.
4	200	20	2	.	$2.1\chi^2(1) - 1$ and $1.5\chi^2(1) - 4$	0.10	Similar to case 3, but with $X \sim \chi^2(1)$. ARI is considerably better for Adaptive TCLUST-REG
5	196	10	3	.	.	0.05	International trade data. Considerably better results obtained with Adaptive TCLUST-REG.

Column 3: number of outliers. Column 4: number of linear components G. Column 5: average overlap level $\bar{\omega}$. Column 6: distribution of the explanatory variable (we report G distributions if the components follow different distributions over X, as in case studies 3 and 4, otherwise, we report the common distribution only once, as in case studies 1 and 2). Column 7: first trimming level α_1. Column 8: notes and main results

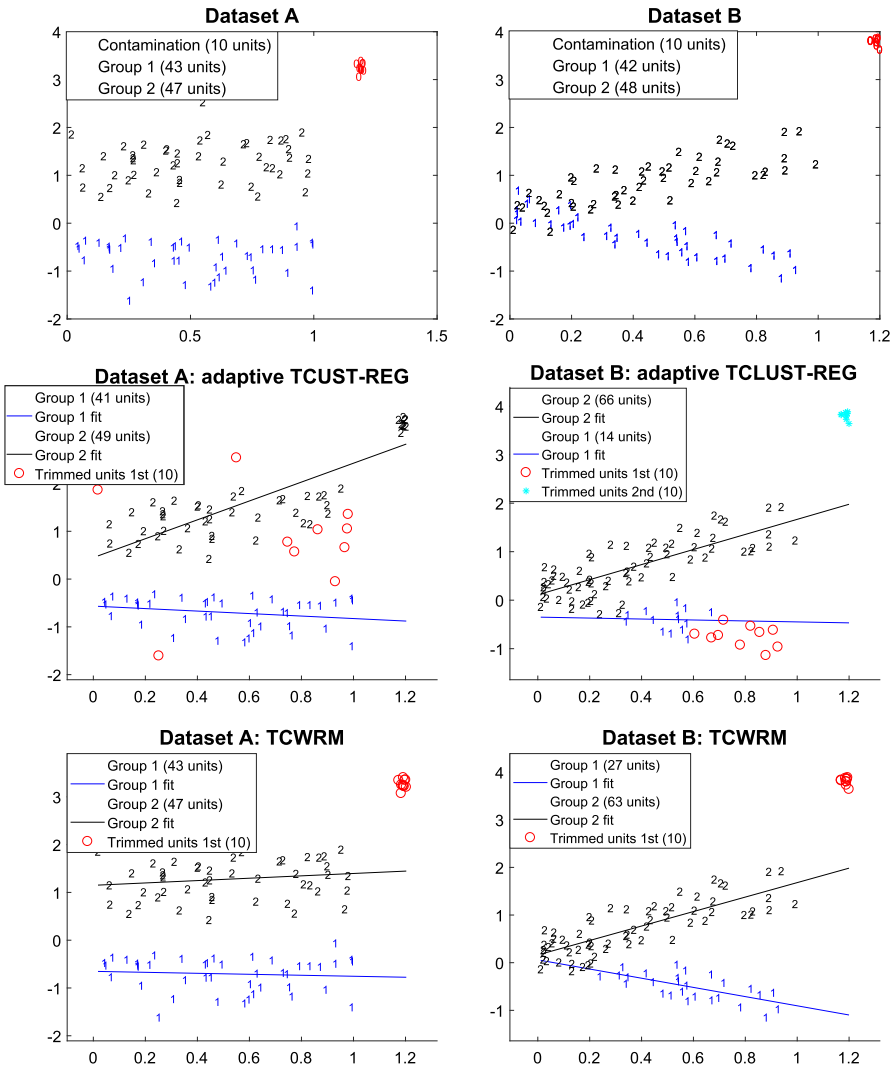

Fig. 6 Case study 1. TCWRM and Adaptive TCLUST-REG on two generic datasets (named A and B) used in the simulation

panel). The distribution around the median of the differences (which is practically 0) shows good symmetry, with only 11 values below -0.05 and 17 above 0.05. This is an indication that the two approaches perform similarly. The median ARI values of Adaptive TCLUST-REG and TCWRM are both very high, approximately equal to 0.93. As expected, they are considerably larger than the median ARI value obtained for the standard version of TCLUST-REG, for which the boxplot is also displayed in the left panel of Fig. 7 as a reference. The boxplot whiskers suggest a slightly smaller variability of the TCWRM results, but the outlying (small) ARI values of TCWRM are also more extreme. The conclusion in case study 1 is that the two approaches perform

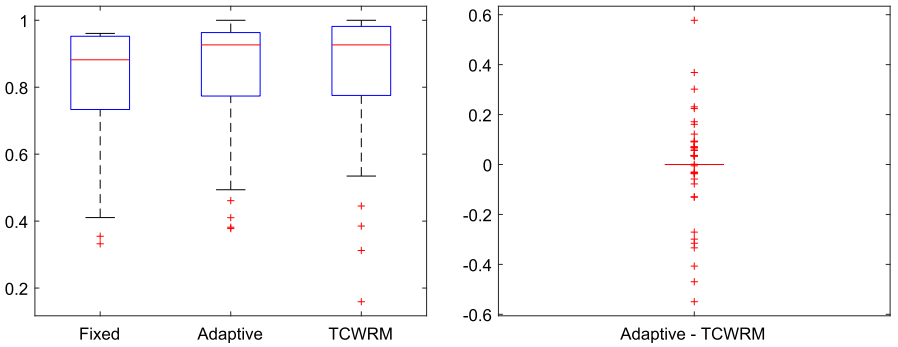

Fig. 7 Case study 1. Left panel: Adjusted Rand Index values between the true partition and the classifications given by fixed-trimming TCLUST-REG, TCWRM and Adaptive TCLUST-REG, for the 100 replicates of the simulations experiment. Right panel: Adjusted Rand Index differences for TCWRM and Adaptive TCLUST-REG

similarly, but TCWRM has been slightly penalised by drawing the explanatory variable values from a Uniform distribution.

5.3 Case study 2

Now we increase the complexity of the data structure by generating 100 datasets of 200 units each from a 3-components mixture model, with a larger average overlap, $\bar{\omega} = 10\%$. The explanatory variable values are now normally distributed, along the TCWRM model assumptions. 15 outliers are added with option `noiseunits` of function `simdatasetreg`. They are generated from the Uniform between the minimum and maximum value of the dependent and independent variables, in such a way that the squared residual from each group is larger than the $1 - 0.999$ quantile of the χ^2 distribution with 1 degree of freedom. The contamination level is therefore approximately 7% (15/215). Two of the 100 simulated datasets are shown at the top of Fig. 8. In this case study, the first trimming level is set slightly larger than the true contamination, i.e. $\alpha_1 = 10\%$. Again, there is no major differences between the two methods in this specific example. On the other hand, the ARI values in the 100 replicates now suggest for Adaptive TCLUST-REG an overall more stable response. In fact, the distribution around the median of the ARI values differences (right panel of Fig. 9), which is about 0.02, is asymmetric and in favor of the Adaptive TCLUST-REG classifications.

The median ARI values in the boxplots of the left panel are respectively 0.62 for the Adaptive TCLUST-REG and 0.6 for TCWRM. Besides, the overall variability is smaller for the Adaptive TCLUST-REG. In spite of the fact that in this example we have correctly guessed the distribution of the explanatory variable, TCWRM produces slightly worse classifications compared to adaptive TCLUST-REG.

5.4 Case study 3

In this case study the 100 simulated datasets mimic an example used by García-Escudero et al. (2017a) (see Figure 3 in their paper) to illustrate the good performances

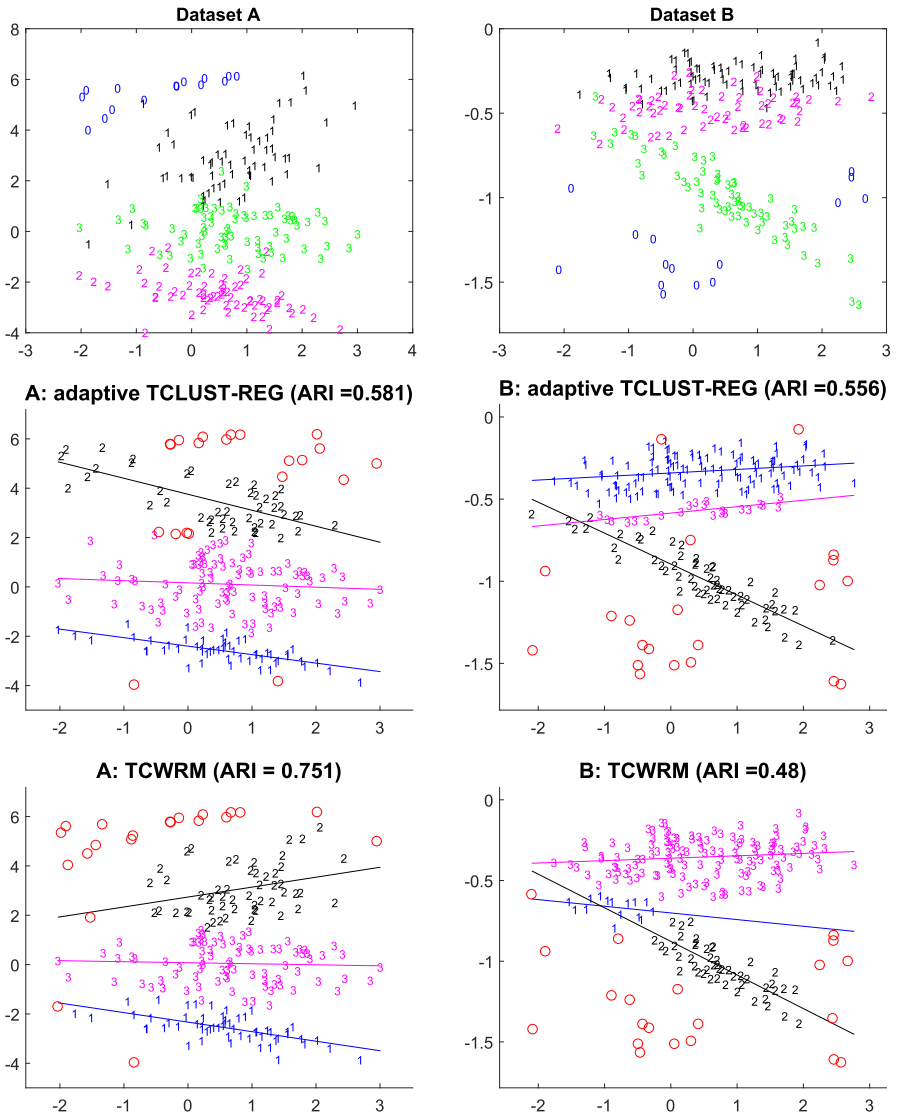

Fig. 8 Case study 2. TCWRM and Adaptive TCLUST-REG on two generic datasets (named A and B) used in the simulation

of TCWRM. An example is represented in the top-left panel of Fig. 10. The datasets are generated from a 2-components mixture model with $n = 180$. A set of 20 outliers is added above the top component. The contamination rate is therefore 10%. Note that the values of the explanatory variable in the mixture model are generated from a Normal distribution.

The boxplot of the Adjusted Rand Index values obtained in the 100 replicates of the simulation (top-right panel of the same Figure) clearly shows that TCWRM now

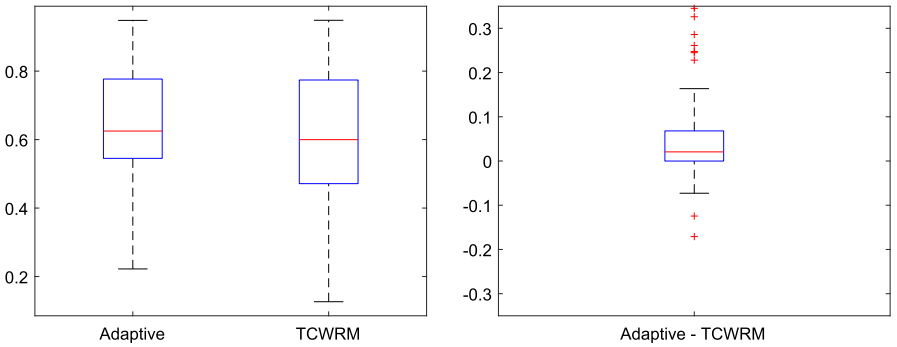

Fig. 9 Case study 2. Left panel: Adjusted Rand Index values between the true partition and the classifications given by TCWRM and Adaptive TCLUST-REG, for the 100 replicates of the simulations experiment. Right panel: Adjusted Rand Index differences. Adaptive TCLUST-REG shows a larger median Adjusted Rand Index value (0.6249 vs 0.5998 for TCWRM) and a smaller variability of the classification results

produces better classifications compared to adaptive TCLUST-REG. The main reason for the improved performance of TCWRM with respect to case study 2 is that in the present example not only the distribution of the explanatory variable is set correctly but also the two mixture components show minor overlap, with good separation from the concentrated outliers. In this "ideal" framework, robust modeling of the distribution of X outperforms the traditional regression approach adopted by adaptive TCLUST-REG, whose behaviour is instead comparable to that obtained in case study 2.

5.5 Case study 4

Case study 4 differs from the third one for the distribution used to generate the explanatory variable values. Here, two χ^2 distributions with 1 degree of freedom are used to concentrate the components data towards specific parts of the explanatory variable domain. The contaminated units are in the range of the χ^2 distributions. The deviation from the normal model assumed in our implementation on the explanatory variable produces a clear deterioration of the TCWRM results. In fact, the boxplot in the bottom panel of Fig. 10 shows that the median of the Adjusted Rand Index values is now much larger for Adaptive TCLUST-REG (65.26%) than for TCWRM (24.54%). In addition, the spread in the plots clearly indicate that Adaptive TCLUST-REG is in general much more stable, with only 4 bad (outlying) classifications.

Figure 11 illustrates two cases where TCWRM fails to capture the true set of outliers, producing wrong fits and bad classifications. The reason of the failure is clear: the explanatory variable values are χ^2 distributed and the observations on the right tail of the distribution are "seen" by TCWRM, which assumes normality, as outliers and are therefore wrongly trimmed. Adaptive TCLUST-REG shows instead a certain robustness to the distributional form of the explanatory variable. We also note, its capacity to flexibly identify some units to trim at the second step, 7 in dataset A and 5 in dataset B. Clearly these numbers depend on the confidence level of the outlier

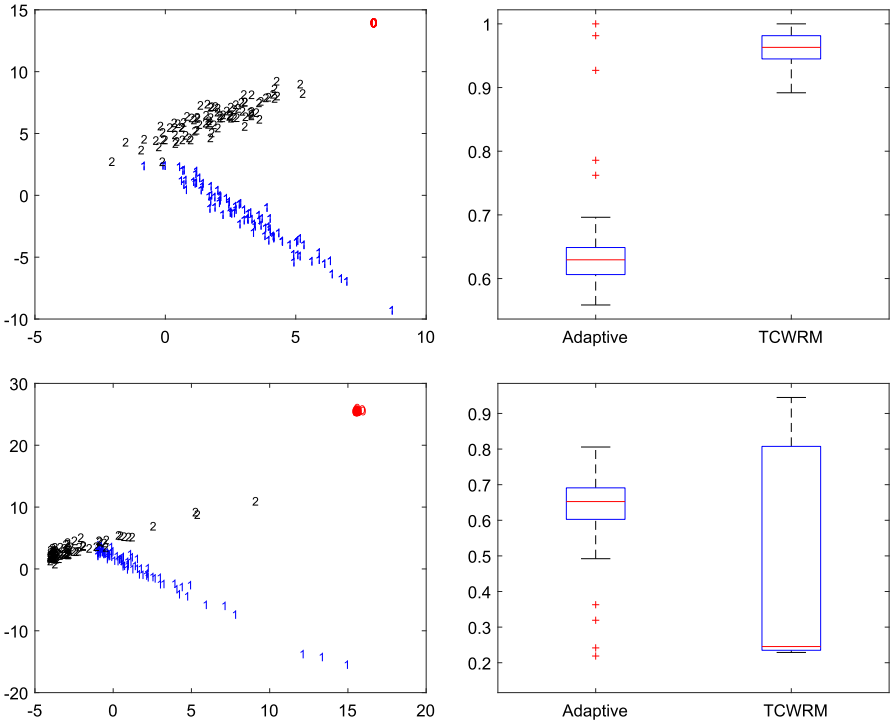

Fig. 10 Case studies 3 (top panels) and 4 (bottom panels). The two settings differentiate for the different distributions used to generate the explanatory variable values: Normal in case study 3 and χ^2 with 1 degree of freedom in case study 4. The datasets generated for the simulations mimic Figure 3 of García-Escudero et al. (2017a). The boxplots report the Adjusted Rand Index values between the true partition and the classifications given by TCWRM and Adaptive TCLUST-REG, for the 100 replicates of the simulations experiment. In case study 3, the median of the Adjusted Rand Index for TCWRM is 0.9631 and for Adaptive TCLUST-REG is 0.6294. In case study 4, the median of the Adjusted Rand Index for TCWRM is 0.2454 and for Adaptive TCLUST-REG is 0.6526

detection methodology chosen for the adaptive trimming step; choosing a different level may lead to slightly different, more or less conservative, results.

5.6 Case study 5: a real dataset from international trade

The dataset represented in Fig. 12 is a real dataset taken from the international trade. It contains values in Euros (y axis) and quantities in Kg (x axis) of 196 declarations made by an Austrian trader who imported from Israel, in a given period of time, a specific product, coded in the international Combined Nomenclature as product 6212200000: "girdles and panty girdles".

The scatterplot shows at least three linear components and, at the bottom right, a group of outliers. By zooming in the scatterplot, it becomes clear that the top and bottom components are more structured: in particular, there are two slightly separated thin components at the top (central panel) and other two at the bottom (right panel).

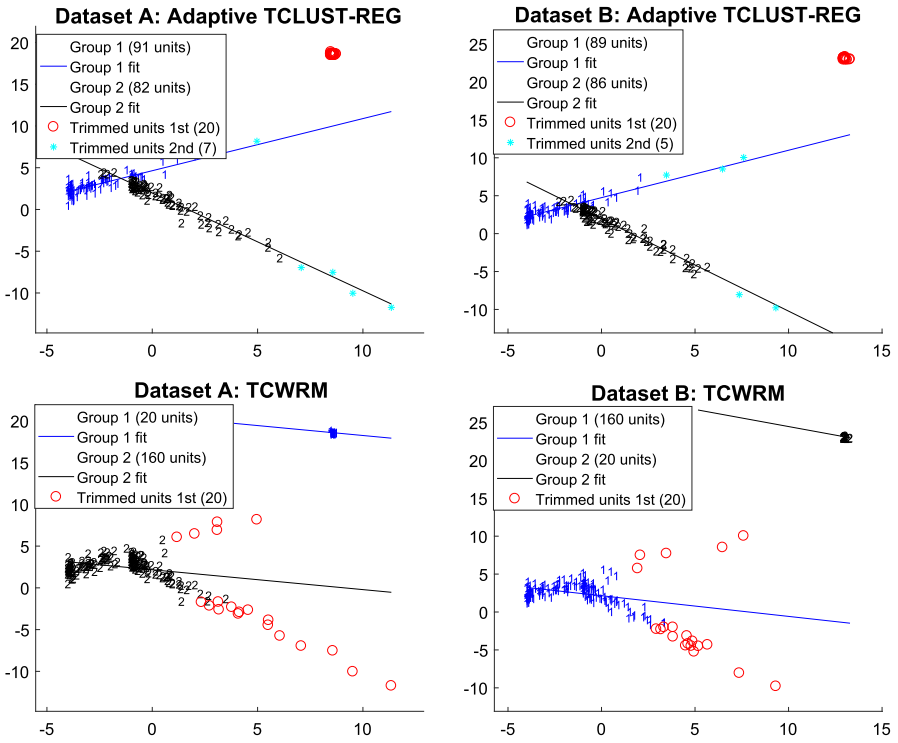

Fig. 11 Case study 4: Adaptive TCLUST-REG (top panels) and TCWRM (bottom panels) on two datasets with χ^2-distributed explanatory variable values

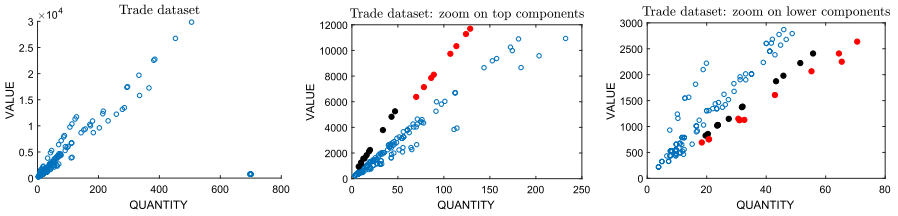

Fig. 12 Scatterplots of case study 5. Trade dataset formed by customs declarations made by an EU importer. The axes report the declared values (y-axis) and quantities (x-axis). The left panel plots the data in the original scale. The central and right panels zoom in the data to highlight the presence of components that are difficult to notice in the original scale

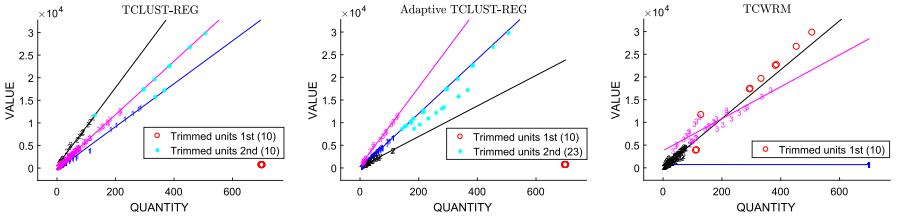

Fig. 13 Case study 5. Dataset of Figure 12 analyzed with three components ($G = 3$) with, from left to right, TCLUST-REG, Adaptive TCLUST-REG and TCWRM

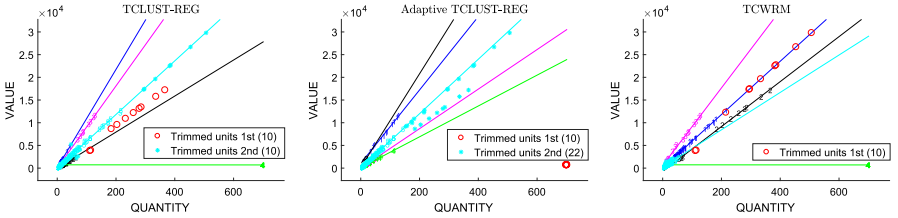

Fig. 14 Case study 5. Dataset of Figure 12 analyzed with five components ($G = 5$) with, from left to right, TCLUST-REG, Adaptive TCLUST-REG and TCWRM

Those at the top are made by two separate groups of points, one closer to the origin of the axes (black filled points) and another far from the origin (red filled points). The two groups at the bottom are also highlighted with filled points in a similar way. It is worth stressing that the lowest group corresponds to very low priced declarations, which might be of interest for anti-fraud purposes (Cerioli and Perrotta 2014, have treated this application domain also in the clusterwise regression framework).

We applied TCLUST-REG, adaptive TCLUST-REG and TCWRM to the dataset by choosing a first trimming level α_1 equal to 5%, which is the percentage of the extreme outliers in the bottom right part of the scatterplot (probably due to recording errors) and a number of groups $G = 3$ (Fig. 13) or $G = 5$ (Fig. 14). For $G = 3$ (Fig. 13) TCLUST-REG and Adaptive TCLUST-REG identify very reasonable components and the outliers. However Adaptive TCLUST-REG fits much better the lowest component, which is the most important for the application. On the contrary TCWRM collapses in the central part of the data, failing to detect the outliers and the relevant regression structures. For $G = 5$ (Fig. 14), Adaptive TCLUST-REG identifies very well the five components highlighted in Fig. 12 and the outliers. On the contrary, TCLUST-REG and TCWRM fail to detect the data structure producing a spurious fit to the outliers. Besides, the results produced by TCLUST-REG in different random starts turned out to be very unstable.

This case study shows that, even with relatively simple real datasets, apparently well structured around few linear components, the departure from the distributional assumptions on the explanatory variable can penalize the performance of TCWRM. Like in case study 4, in fact, the data distribution lacks symmetry and is much more dense near the origin of the axes. For this reason, we plan extending the model to more appropriate distributions for traded quantities (Tweedie and Tempered Linnik) that were identified in Barabesi et al. (2016a, b). Instead, our adaptive version of TCLUST-REG seems sufficiently flexible to cope with departures from the normal assumption and results to be the best performing method, independently from the number of groups chosen.

5.7 Simulation exercise

Our case studies indicate that TCLUST-REG attains better performances when the second level trimming is adaptive and corroborate the expected superior properties of TCWRM when the model assumptions on the explanatory variables are respected. The

simulation study of this section aims to check whether this conclusion is confirmed under experimental settings that generalize the international trade domain of case study 5.

We consider two scenarios, with two and three mixture components respectively. Each run is based on 1000 replicates where TCLUST-REG, its adaptive version and TCWRM are applied to distinct datasets. As the accent of the paper is on the second level trimming, the contamination is formed by a group of rather concentrated normally distributed units with high leverage. The mixture data and the contaminants were both generated with *MixSimReg*, to attain an average overlap (or expected misclassification error) of 1%. The good part of each dataset is formed by $n = 200$ units, while the contaminants (generated with option `noiseunits`) are 30 (15% of n). The parameters of the mixture components are generated (using option `betadistrib`) from a Normal with mean 1.2 and standard deviation 2. A restriction factor $c_y = 5$ is imposed on regression residuals (option `restrfactor`).

We studied four cases, with explanatory variable values generated from the following distributions and re-scaled to be roughly in the same interval:

- a Uniform in the range $[-2, 10]$;
- a Normal with mean 3.2 and standard deviation 4.4;
- a χ^2 with 1 degree of freedom;
- a Beta with both parameters equal to 0.2.

Note that the parameters of the Beta are chosen identical in order to obtain a "U" shaped distribution, with lot of points concentrated at the extremes of the data interval and few at the center: a case opposed to the Normal.

TCLUST-REG, adaptive TCLUST-REG and TCWRM are then run with:

- `nsamp` $= 300$ (number of subsets);
- `restrfact(1)` $= 5$ (restriction factor for regression residuals, c_y);
- `restrfact(2)` $= 5$ (restriction factor for covariance matrix of explanatory variables, c_X, TCWRM only);
- `alphaLik` $= 0.15$ (trimming level α_1); this corresponds to the actual contamination percentage;
- `alphaX` $= 0.05$ (second level trimming α_2, TCLUST-REG only)
- `alphaX` $= 0.9$ (90% Bonferronized confidence level, adaptive TCLUST-REG only);
- `alphaX` $= 1$ (used to choose the constrained weighted model TCWRM).

Table 4 reports the Adjusted Rand Index obtained on mixtures of $G = 2$ and $G = 3$ groups. The robust linear grouping methods used are in the rows and the distributions used to generate the data for the explanatory variables are in the columns. It is very clear that, as expected, TCWRM has superior performances when the explanatory variable values are generated from a Normal distribution, the only one currently implemented in FSDA. Not surprisingly, the same happens with uniformly distributed data. On the contrary, distributions radically departing from normality (very asymmetric as the χ^2, or "U" shaped as the Beta) deteriorate considerably the classification capacity of TCWRM. Our adaptive version of TCLUST-REG confirms the good properties discussed in the case studies. However, data with "U" shaped explanatory variable

Table 4 Adjusted Rand Index obtained in a simulation exercise designed to assess the performances of the three robust linear grouping methods for different distributions of the explanatory variable values, when the mixture components are two (top panel) or three (bottom panel)

	$N(3.2, 4.4^2)$	$U(-2, 10)$	$\chi^2(1)$	$Beta(0.2, 0.2)$
G = 2				
TCLUST-REG	0.4189	0.4270	0.4612	0.5314
Adaptive TCLUST-REG	0.4995	0.4758	0.4892	0.4768
TCWRM	0.7685	0.6210	0.2726	0.2659
G = 3				
TCLUST-REG	0.4815	0.4670	0.5546	0.6089
Adaptive TCLUST-REG	0.5445	0.5213	0.5412	0.5958
TCWRM	0.5868	0.5647	0.4648	0.4880

Each simulation is based on 1000 replicates. The *MixSimReg* parameters used to generate the data and the contamination and scheme, together with the options used to run the clustering methods are detailed in the text

values are fit better by the standard TCLUST-REG with fixed second level trimming. A logical explanation is that with this distribution the contamination falls just after the good units concentrated at the right (or left) side of the $Beta$. This creates a bi-modal set of values which cannot be fit properly by the robust method (Forward Search or re-weighted MCD) applied to identify the proper number of units to trim.

6 Conclusions

Although robust clustering tools for regression data might be useful in several application domains, like international trade (Cerioli and Perrotta 2014), very little is known about their performance under different data configurations. Our work attempts to clarify this point by comparing two methodologies that use trimming and restrictions on group scatters as their main ingredients. Our assessment is based on simulation experiments run under a variety of alternative conditions and we have given particular care to the data generation process. We have thus developed, and described in the paper, a flexible simulation tool for mixtures of regressions, where the user can have a precise control of the degree of overlap between the regression hyperplanes defining the different components, as well as the choice among different options for the distributional features of the grouped data and for the contamination process.

Our first finding concerns the usefulness of the second-level trimming required by TCLUST-REG on the values of the explanatory variables. Although we have seen that this step indeed provides beneficial consequences in some situations, it is also clear that excessive trimming of non-harmful observations, or even of good leverage points, can deteriorate both the classification performance of the method and the estimates of the underlying model parameters. We have thus proposed an improvement of the methodology where the degree of trimming exerted on the explanatory variables is not fixed in advance, but is allowed to vary according to the specific data configuration.

We have then compared our flexible and adaptive version of TCLUST-REG with TCWRM, which provides an important and powerful extension of the robust cluster-

wise regression methodology. Our overall conclusion is that the two methods perform comparably, but with some notable differences due to the inherent degree of modeling implied by them. Since TCWRM exploits the full distributional structure of the explanatory variables, its notable advantage is not surprising when this structure is correctly specified and, moreover, is far from that of the contaminant distribution. On the other hand, Adaptive TCLUST-REG turns out to be less sensitive to how data are distributed in the explanatory variables, when instead TCWRM can have poor performance. After all, what we have seen is just another instance of the longstanding antinomy between robustness and efficiency: it is clearly less dangerous to make only a few mistakes in the outlier detection step, thanks to our flexible trimming approach, than to incorrectly specify the covariate part of the model. However, the availability of prior information on the data generating mechanism, or at least a good guess of it, considerably improves the results also in a clustering framework.

Acknowledgements We thank Luis Angel Garcia Escudero, Alfonso Gordaliza and Agustin Mayo-Iscar (University of Valladolid) for the discussions had in many occasions on the algorithmic and implementation details of their trimming and restrictions based methods. The work has been partially supported by the European Commission's Hercule III programme 2014-2020 through the Automated Monitoring Tool project. This research benefits from the HPC (High Performance Computing) facility of the University of Parma, Italy. M.R. gratefully acknowledges support from the CRoNoS project, reference CRoNoS COST Action IC1408. M.R. and A.C. would like to thank the European Union's Horizon 2020 Research and Innovation Programme for its financial support of the PrimeFish project, Grant Agreement No. 635761.

References

Banfield J, Raftery A (1993) Model-based gaussian and non-gaussian clustering. Biometrics 49(3):803–821

Barabesi L, Cerasa A, Cerioli A, Perrotta D (2016a) A new family of tempered distributions. Electron J Stat 10:1031–1043

Barabesi L, Cerasa A, Perrotta D, Cerioli A (2016b) Modeling international trade data with the tweedie distribution for anti-fraud and policy support. Eur J Oper Res 248(3):1031–1043

Campbell J (1984) Mixture models and atypical values. Math Geol 16:465–477

Campbell J, Fraley C, Murtagh F, Raftery A (1997) Linear flaw detection in woven textiles using model-based clustering. Pattern Recognit Lett 18(14):1539–1548

Cerasa A, Cerioli A (2017) Outlier-free merging of homogeneous groups of pre-classified observations under contamination. J Stat Comput Simul 87(15):2997–3020

Cerioli A (2010) Multivariate outlier detection with high-breakdown estimators. J Am Stat Assoc 105(489):147–156

Cerioli A, Riani M, Atkinson AC, Corbellini A (2017) The power of monitoring: how to make the most of a contaminated multivariate sample. Stat Methods Appl. https://doi.org/10.1007/s10260-017-0409-8

Cerioli A, Garcia-Escudero LA, Mayo-Iscar A, Riani M (2018) Finding the number of normal groups in model-based clustering via constrained likelihoods. J Comput Graph Stat 27(2):404–416. https://doi.org/10.1080/10618600.2017.1390469

Cerioli A, Perrotta D (2014) Robust clustering around regression lines with high density regions. Adv Data Anal Classif 8(1):5–26

Dasgupta A, Raftery AE (1998) Detecting features in spatial point processes with clutter via model-based clustering. J Am Stat Assoc 93(441):294–302

Davies RB (1980) The distribution of a linear combination of χ^2 random variables. J R Stat Soc Ser C (Appl Stat) 29(3):323–333
DeSarbo W, Cron W (1988) A maximum likelihood methodology for clusterwise linear regression. J Classif 5(2):249–282
Dotto F, Farcomeni A, García-Escudero LA, Mayo-Iscar A (2018) A reweighting approach to robust clustering. Stat Comput 28:477–493
Farcomeni A, Dotto, F (2018) The power of (extended) monitoring in robust clustering. Stat Methods Appl. https://doi.org/10.1007/s10260-017-0417-8
Fritz H, Garca-Escudero LA, Mayo-Iscar A (2012) tclust: an R package for a trimming approach to cluster analysis. J Stat Softw 47(12):1–26
Fritz H, García-Escudero L, Mayo-Iscar A (2013) A fast algorithm for robust constrained clustering. Comput Stat Data Anal 61:124–136
García-Escudero L, Gordaliza A, Mayo-Iscar A, San Martin R (2010) Robust clusterwise linear regression through trimming. Comput Stat Data Anal 54(12):3057–3069
García-Escudero LA, Gordaliza A, Greselin F, Ingrassia S, Mayo-Iscar A (2016) The joint role of trimming and constraints in robust estimation for mixtures of gaussian factor analyzers. Comput Stat Data Anal 99:131–147
García-Escudero LA, Gordaliza A, Greselin F, Ingrassia S, Mayo-Iscar A (2017a) Robust estimation of mixtures of regressions with random covariates, via trimming and constraints. Stat Comput 27(2):377–402
García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2017b) Comments on "The power of monitoring: how to make the most of a contaminated multivariate sample". Stat Methods Appl. https://doi.org/10.1007/s10260-017-0415-x
García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2008) A general trimming approach to robust cluster analysis. Ann Stat 36(3):1324–1345
García-Escudero LA, Gordaliza A, Mayo-Iscar A, San Martín R (2010) Robust clusterwise linear regression through trimming. Comput Stat Data Anal 54(12):3057–3069
Gershenfeld N (1997) Nonlinear inference and cluster-weighted modeling. Ann N Y Acad Sci 808(1):18–24
Gershenfeld N, Schoner B, Metois E (1999) Cluster-weighted modelling for time-series analysis. Nature 397(6717):329–332
Gordaliza A (1991) Best approximations to random variables based on trimming procedures. J Approx Theory 64(2):162–180
Hennig C (2003) Clusters, outliers, and regression: Fixed point clusters. J Multivar Anal 86(1):183–212
Ingrassia S, Minotti SC, Vittadini G (2012) Local statistical modeling via a cluster-weighted approach with elliptical distributions. J Classif 29(3):63–401
Maitra R, Melnykov V (2010) Simulating data to study performance of finite mixture modeling and clustering algorithms. J Comput Graph Stat 2(19):354–376
Melnykov V, Chen W-C, Maitra R (2012) Mixsim: an R package for simulating data to study performance of clustering algorithms. J Stat Softw 51(12):1–25
Neykov N, Filzmoser P, Dimova R, Neytchev P (2007) Robust fitting of mixtures using the trimmed likelihood estimator. Comput Stat Data Anal 52(1):299–308
Peel D, McLachlan G (2000) Robust mixture modeling using the t-distribution. Stat Comput 10:335–344
Perez B, Molina I, Pena D (2014) Outlier detection and robust estimation in linear regression models with fixed group effects. J Stat Comput Simul 84(12):2652–2669
Perrotta D, Torti F (2018) Discussion of "The power of monitoring: how to make the most of a contaminated multivariate sample". Stat Methods Appl. https://doi.org/10.1007/s10260-017-0420-0
Riani M, Atkinson AC, Cerioli A (2009) Finding an unknown number of multivariate outliers. J R Stat Soc Ser B 71:447–466
Riani M, Cerioli A, Perrotta D, Torti F (2015) Simulating mixtures of multivariate data with fixed cluster overlap in FSDA library. Adv Data Anal Classif 9(4):461–481
Riani M, Perrotta D, Cerioli A (2015) The forward search for very large datasets. J Stat Softw 67(1):1–20
Riani M, Perrotta D, Torti F (2012) FSDA: A MATLAB toolbox for robust analysis and interactive data exploration. Chemom Intell Lab Syst 116:17–32
Rousseeuw PJ (1984) Least median of squares regression. J Am Stat Assoc 79:871–880
Rousseeuw PJ, Van Driessen K (1999) A fast algorithm for the minimum covariance determinant estimator. Technometrics 41(3):212–223