N. d'ordre : 41416

Université Lille1 : Sciences et Technologies
École Doctorale Sciences Pour l'Ingénieur

Thèse

pour obtenir le grade de :

Docteur de l'Université Lille1

dans la spécialité

« Mathématiques Appliquées »

par

Loïc Yengo

Contribution à la classification de variables dans les modèles de régression en grande dimension

Thèse soutenue le 28 mai 2014 devant le jury composé de :

M.	Nicolas Wicker	(Président du jury)
M.	Grégory Nuel	(Examinateur)
M.	Steéphane Robin	(Examinateur)
M.	Avner Bar-Hen	(Examinateur)
M.	Julien Jaqcues	(Co-Directeur de Thèse)
M.	Christophe Biernacki	(Directeur de Thèse)

Laboratoire Paul Painlevé (UMR CNRS 8524)
U.F.R de Mathématiques
Université Lille1
59655 Villeneuve d'Ascq France

Contents

List of Figures

List of Tables

Acknowledgements

First of all, my most sincere acknowledgments go to Julien Jacques and Christophe Biernacki who supervised this research. Their infallible support and inspiring guidance have been priceless during this four years trip.

To be more accurate this trip did actually start five years ago in another field of applications being Toxicokinetics. I am grateful to Alexandre Péry for being my mentor during my early life as a researcher. In addition, I thank all the people in the team METO. Thanks Rémy, Céline, Florence and Enrico.

I thank Philippe Froguel for believing in me since the beginning. Welcoming me in his research team was the most exciting and enriching challenge I ever faced.

I cannot go on with these acknowledgements without mentioning the precious help from people who work with me everyday. Their support was also priceless every time I had to concentrate on my PhD. Thanks Yann, Rémy, Boris, Cécile, Dorothée, Mickaël, Marie and Ghislain.

Many others also contributed to help me completing this effort. I adress my special thanks to Sadia, Murielle, Emmanuel and Alexandre. I have you all in my heart.

I finally dedicate these last words to my parents Bernadette Nsounda and Jean-Louis Yengo: *Matondo ma sakila*[1].

[1] means "thank you very much" in Lari, my mother tongue.

Résumé

Cette thèse propose une contribution originale au domaine de la classification de variables en régression linéaire. Cette contribution se base sur une modélisation hiérarchique des coefficients de régression. Cette modélisation permet de considérer ces derniers comme des variables aléatoires distribuées selon un mélange de Gaussiennes ayant des centres différents mais des variances égales. En transferant de l'hypothèse de distribution des covariables vers les coefficients de régression, notre modèle rend la classification de variables plus facile dans les cas où les covariables ne sont pas issues d'une même famille de distributions. Par exemple, aucune hypothèse supplémentaire n'est pas à faire pour classer ensemble des covariables quantitatives et qualitatives. Nous montrons dans cette thèse que l'algorithme EM, communément utilisé pour estimer les paramètres d'un modèle hiérarchique ne peut s'appliquer. En effet, l'étape E de l'algorithme n'est pas explicite pour notre modèle. Nous avons d'abord étudié une stratégie d'estimation basée sur une approximation par simulations Monte Carlo de l'étape E. Cette stratégie, connue sous le nom d'algorithme Monte Carlo EM a toutefois été jugée trop lente. Nous avons donc proposé une approche plus efficace pour l'estimation des paramètres grâce à l'utilisation de l'algorithme SEM-Gibbs. En plus de cette amélioration computationnelle, nous avons introduit une contrainte dans le modèle pour permettre d'effectuer une sélection de variables simulanément. Notre modèle présente de très bonnes qualités prédictives relativement aux approches classiques pour la réduction de la dimension en régression linéaire. Cette thèse présente aussi une extension de notre méthodologie dans le cadre de la régression Probit pour données binaires. Cette extension a également donné des résultats encourageant en termes de qualité de prédiction. Enfin, deux autres extensions du modèle ont été étudiées. Nous avons premièrement généralisé notre modèle en relâchant l'hypothèse de l'égalité des variances pour les composantes du mélange Gaussien. Les perfor-

mances de ce modèle généralisé ont été comparées à celles du modèle initial à travers différents scénarios de simulations. Bien que moins rapide, cette généralisation a permis d'améliorer dans certains cas à la fois l'erreur de prediction et l'erreur de classification de notre modèle (capacité à inférer la partition latente des coefficients de régression). La dernière extension envisageée concerne la classfication de variables indépendamment du signe de leur effet mais uniquement sur la base de la valeur absolue de cet effet. Les performances predédictives de ce dernier modèle ont été étudiées aux travers d'expériences numériques sur des données réelles et simulées. Ces expériences revèlent qu'augmenter la flexibilité du modèle dela sorte empire les performances en grande dimension.
Ce travail de recherche a conduit au développement du paquet `R clere`. Ce dernier paquet met en oeuvre tous les les algorithmes décrits dans cette thèse.

Abstract

We proposed in this thesis an original contribution to the field of variable clustering in linear regression through a model-based approach. This contribution was made via a hierarchical modeling of the regression coefficients as random variables drawn from a mixture of Gaussian distributions with equal variances. By transferring the distributional assumption from the covariates to the regression coefficients, our model makes variable clustering easier in cases where the variables do not belong to the same family of distributions (binary and continuous covariates might be clustered together). Parameter estimation in the proposed model was shown to be challenging since the classical EM algorithm could not apply. We first studied an estimation strategy based on a Monte Carlo approximation of the EM algorithm. This strategy also known as the Monte Carlo EM (MCEM) algorithm was however found to be very slow. We then developed a more efficient algorithm for parameter estimation, through the use of the SEM-Gibbs algorithm. Along with this computational improvement, we also enhanced our model to allow variable selection. Given the good predictive performances of the CLERE method compared to standard techniques for dimension reduction, we considered an extension of the latter to binary response data. This extension was studied in the context of Probit regression. This extension also showed competitive predictive performances. We finally considered two other extensions to the CLERE methodology. First, we generalized our model by relaxing the assumption of equal variance for the components in the mixture of Gaussians. The performances of this generalization were compared to those of the initial model under different scenarios on simulated data. Although more computationally intensive, this generalization was shown to improve the classification error associated with the recovery of the latent partition of the covariates. The second extension explored was studied to improve the model parsimony and aimed at performing variable clustering regardless the sign of effect size but

only using its magnitude. We studied the predictive performances of this new model and revealed its limitation in high dimensional settings. This research led to the development of the `R` package `clere` which implements most of the algorithms described in this thesis.

Chapter 1

Introduction

The advent and rising development of technologies for massive data collection is changing the landscape of society. From ubiquitous social networks to advanced molecular Biology, all the layers of our lives are affected by what has been lately denoted as the Big data phenomenon. These changes are first of all a huge opportunity, especially for scientific research. Indeed, an incredibly large number of discoveries is associated with those new technologies. As a specific example, only 5 to 10 genetic loci were known to influence Human height before the era of Genome-wide association studies. This number was taken to 700, five years later. With this opportunity also came a number of challenges. The major one is certainly the challenge of deciphering the complexity generated by all possible interactions between those newly available data. New areas of research have therefore emerged to help social and life scientists address this issue. One question raised in this context was whether classical tools developed for low-dimensional data may be applied in high-dimensional settings. This question has been particularly studied for linear regression models. Linear regression models have been utilized in many scientific fields for centuries. In low-dimensional settings, those models are easily interpretable and yield relevant predictions. However, as long the accounted number of variables increases those two properties get lost.

The most studied solutions so far proposed for high-dimensional linear regression are based on variable selection techniques. These methods are perfectly suited in cases where the intrinsic dimension, i.e. the number of actual variables influencing the response, is small with respect to the sample size. However, it happens that some phe-

nomena are influenced by a quite large number of variables (700 so far known for Human height). More recently, strategies that mixed variable selection and variable clustering have been developed as an improvement to classical variable selection approaches. This new area of research has been notably fueled in the last five years.

Despite this rapid growth, none of the contributions proposed for variable clustering in linear regression have made use of traditional tools of model-based clustering. We therefore found relevant to propose a new method that would be based on the latter tools. This introduction has two main parts. The first is dedicated to a literature review of significant contributions in variable selection and variable clustering for linear regression. The second parts describes briefly our contribution and how it is outlined throughout this manuscript.

1.1 Literature review

We consider the usual linear regression model defined as

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \text{ with } \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \ (i = 1, \ldots, n) \tag{1.1}$$

Without loss of generality, we assume that $\beta_0 = 0$. In model (1.1), y_i stands for a measured response variable (age, viral load, incomes, etc.), x_{ij} for the j-th explanatory variable (covariate), β_j for the effect associated with the j-th covariate and ε_i is an error term.

Despite its simplicity, this model has been used in numerous scientific fields for centuries. The reason for such a success mainly relates to (1) the ease of interpretation of its parameters and sometimes to (2) its good predictive performances. Preserving those two properties of interpretability and accurate prediction is a necessary condition for a proper use of linear regression models. The number p of covariates in the model also denoted as the *model dimensionality* is known to influence the two crucial properties mentioned above. From a practical point of view, one can easily realize that the more covariates we have in the model the harder becomes the interpretation of the process

that generated the data. In addition, Stein demonstrated in [44] that the prediction error attributable to a model like (1.1) increases along with p. It comes thus naturally that building parsimonious model is a precious goal to pursue.
We present in this section main techniques for dimension reduction in the field of linear regression. This section is organized as follows. In subsection 1.1.1, we present classical techniques based on variable selection, while in subsection 1.1.2 we present alternative approaches mostly based on variable clustering.

Figure 1.1: Tag cloud representing key words from the references cited in this manuscript.

1.1.1 Variable selection in linear regression

As highlighted above, identifying the most relevant covariates is a critical question in linear regression. This issue which first raised for prediction and interpretation purposes, becomes crucial in high dimensional regression problems, i.e. when $p > n$ or $p \gg n$.

The number of possible subsets among p predictors increases exponentially with respect to p. Finding the best subset of predictors, regardless the objective criterion considered, is therefore an NP-hard combinatorial optimization problem [2]. Stepwise algorithms [20], although sub-optimal, are often proposed as a way to solve that problem. These sequential approaches are however based on greedy algorithms that make them unsuited in high dimensional settings. Moreover, those methods exhibit high variance in their estimates and weakly reduce the prediction error.

The best subset selection problem can also be formulated as the following penalized least squares problem:

$$\widehat{\boldsymbol{\beta}}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \|\boldsymbol{\beta}\|_0 \right\}, \tag{1.2}$$

where $\boldsymbol{\beta} = \left(\beta_1, \ldots, \beta_p\right)' \in \mathbb{R}^p$, λ is a tuning parameter and $\|\boldsymbol{\beta}\|_0$ is the L^0 norm of vector $\boldsymbol{\beta}$, i.e. the number of components in the vector that does not equal zero. The problem defined in Equation (1.2) is non-convex. Nevertheless, convex relaxations of this problem can be proposed by using instead of the L^0 norm, penalties based on L^q norms with $q \geq 1$. The family of convex-relaxations thus defined is known as *Bridge* regressions [47]. The most studied of these methods is obtained with $q = 1$, and mainly known as the Least Absolute and Shrinkage Operator or simply LASSO [45]. Although ridge regression [21] also belongs to the family of *Bridge* regressions (obtained with $q = 2$), it cannot be considered as a variable selection method. Indeed, the solutions it yields are not sparse. As a consequence, the latter method will not be detailed in this subsection.

The LASSO estimator $\widehat{\boldsymbol{\beta}}^{\text{LASSO}}$ of $\boldsymbol{\beta}$ therefore verifies

$$\widehat{\boldsymbol{\beta}}^{\text{LASSO}}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{1.3}$$

Efficient algorithms were developed in [15, 31] to solve the optimization problem (1.3) and thus contributed to the success of the LASSO during the last three decades. By

penalizing the absolute size of the regression coefficients, less important predictors are dropped from the model as their effect size is set to zero. The penalty parameter λ is generally chosen to minimize the out-of-sample prediction error, through cross-validation procedures.

Bayesian variable selection

The LASSO also has a Bayesian interpretation. Indeed, the penalty term $\text{pen}(\lambda) = \lambda\|\boldsymbol{\beta}\|_1$ in Equation (1.3) may be considered as the log density of $\boldsymbol{\beta}$ under the prior assumption that the β_j's are independent and identically Laplace distributed variables [33]. Other Bayesian approaches were introduced for variable selection in linear regression. Among these approaches we may refer to the spike and slab prior [30, 23] as the mostly studied method. The spike and slab prior, represented on Figure 1.2, assumes that the regression coefficients are distributed according to a mixture of two centered Gaussian distributions with different variances. One component of the mixture (the spike) is chosen to have a small variance, while the other component (the slab) is allowed to have a large variance. Variables having a large posterior probability to be assigned to the spike are dropped from the model.

1.1.2 Variable clustering in linear regression

Improvements brought to the LASSO

The idea of grouping covariates is not recent in linear regression. We found the premises of this idea in the works of Joliffe [25] who proposed to replaced the covariates with the principal components obtained from them. Although not being a proper variable clustering approach, this method led to interesting predictive improvements. More recently, this idea regained strength through the joint works of Zhou and Hastie [52]. The latter proposed to mix the L^1 penalty in the LASSO with a L^2 penalty to improve the variable selection in presence of strongly correlated predictors. This approach is known as the *Elastic net*. The *Elastic net* consequently solves the following optimization problem:

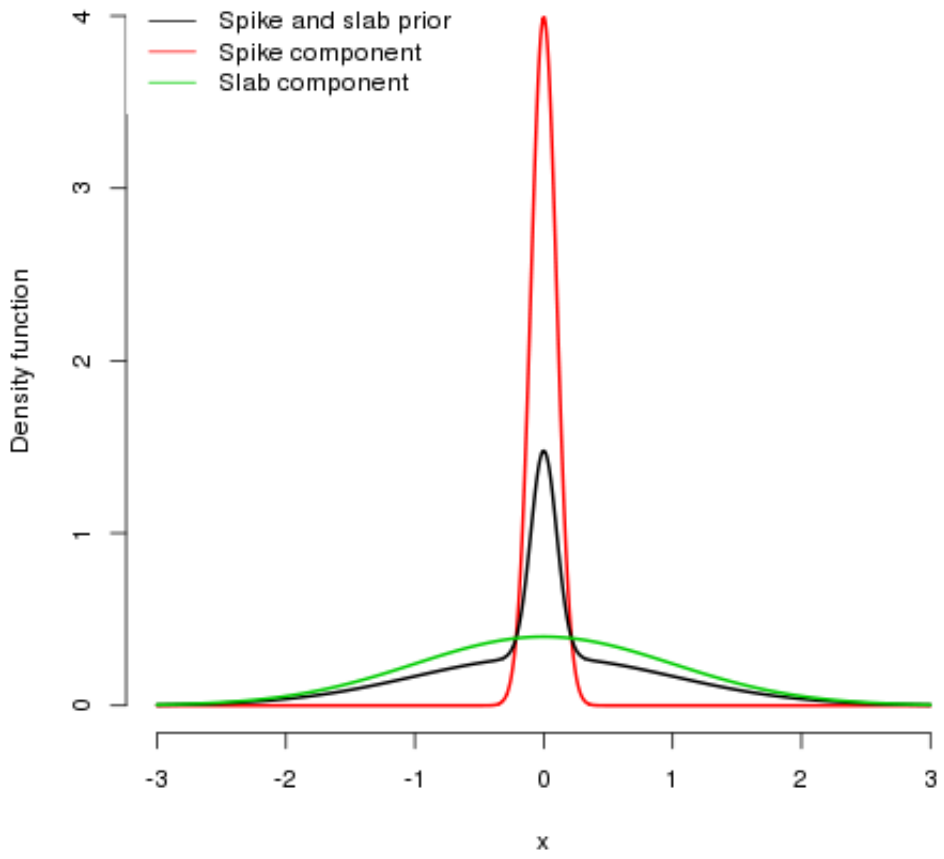

Figure 1.2: Spike slab prior density.

$$\widehat{\boldsymbol{\beta}}^{\text{Elastic net}}(\lambda_1,\lambda_2) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\}. \quad (1.4)$$

Another approach, referred to as the *Group LASSO* [50] may also be considered for the same purpose. The *Group LASSO* uses predefined groups of covariates, to modulate the penalty applied to each regression coefficients according to the group it belongs to.

In terms of optimization problem, it can be written as

$$\widehat{\boldsymbol{\beta}}^{\text{Group LASSO}}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \sum_{k=1}^{g} z_{jk} \omega_k |\beta_j| \right\}, \qquad (1.5)$$

where g is the number of known groups, z_{jk} an indicator variable that equals one if the j-th covariate belongs the k-th group and 0 else; and ω_k stands for a group dependent weight that modulates the penalty in the group k. The *Group Lasso* had a growing influence in Biology where predefined structures like metabolic pathways are easily available [51]. In the usual case when clusters of covariates are unknown, Park and colleagues [32] suggested the use hierarchical clustering based on covariates to build groups of regressors. Their approach was particularly relevant in the case of numerous correlated predictors with weak effects. The optimality properties of their approach were recently studied by Buhlmann and colleagues in [8].

Simultaneaous variable clustering and regression

Correlated covariates may not always yield the same effect on the response. For example, genetic variants closely located to one another are often strongly correlated; however it happens that only one them may associate with an impairment in gene expression. Consequently, performing the clustering of covariates regardless their impact on the response may lead to create clusters with weak relevance. To address this limitation, a seminal article from Bondell and Reich [7] introduced the octogonal shrinkage and clustering algorithm for regression (OSCAR). The OSCAR methodology is a penalized approach that combines the L^1 penalty with the pair-wise L^∞ penalty. Upper-bounding the L^1 norm of the vector of regression coefficients, as LASSO does, forces certain coefficients to equal zero. Similarly, imposing a constraint on the pair-wise L^∞ of regression coefficients, enforces some covariates to have equal coefficients. Covariates with equal regression coefficients naturally cluster together. As a result, the OSCAR methodology may be considered as a supervised clustering approach.

A generalization of the OSCAR methodology has been been recently proposed in [41]. This generalization, namely the Pairwise Absolute Clustering and Sparsity (PACS),

came with a reduction in the computational time required for model inference. The PACS methodology solves the following optimization problem:

$$\widehat{\boldsymbol{\beta}}^{\text{PACS}}(\lambda) = \underset{\boldsymbol{\beta}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \times pen(\boldsymbol{\omega}, \boldsymbol{\beta}) \right\}, \tag{1.6}$$

where $pen(\boldsymbol{\omega}, \boldsymbol{\beta})$ is defined below in Equation (1.7).

$$pen(\boldsymbol{\omega}, \boldsymbol{\beta}) = \sum_{j=1}^{p} \omega_j |\beta_j| + \sum_{1 \leq j < k \leq p} \omega_{jk(-)} |\beta_k - \beta_j| + \sum_{1 \leq j < k \leq p} \omega_{jk(+)} |\beta_k + \beta_j| \tag{1.7}$$

The penalty term in PACS given in Equation (1.7) involves weights (the ω_j's, the $\omega_{jk(-)}$'s and the $\omega_{jk(+)}$'s) that must be supplied by the user. A discussion on how choosing those weights is detailed in [41]. The OSCAR estimator is obtained from PACS by simply imposing the following constraints:

$$\forall j = 1, \ldots, p;\ \omega_j = c \tag{1.8}$$

$$\forall (j,k)\ /\ 1 \leq j < k \leq p;\ \omega_{jk(-)} = \omega_{jk(+)} = \frac{1-c}{2} \tag{1.9}$$

where $c \in [0;1]$. In the aftermath of OSCAR (and PACS), other methodologies aiming at simultaneously performing parameter estimation and clustering were proposed. We can for instance refer to the approaches of Petry [35] and She [42] which also mixed L^1 and pairwise L^∞ penalties or those of Daye [12] and Shen [43] based on alternative penalties.

Variable clustering within linear regression models often translates into multiple covariates having close, if not equal, regression coefficients. The last series of techniques aforementioned, including PACS and the *Cluster Group LASSO* [8], are known to encourage correlated covariates to have similar coefficients. However they do not necessarily encourage covariates having similar association with the response to cluster together. This observation was strongly underlined by Witten and colleagues [48] who introduced the *Cluster Elastic net* (CEN) to circumvent this limitation. The CEN has the advantage of not systematically assign similar coefficients to all correlated covariates but only does when correlated covariates have similar association with the response.

The CEN solves the following optimization problem

$$\underset{C_1,\ldots,C_g,\boldsymbol{\beta}}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \delta \|\boldsymbol{\beta}\|_1 + \frac{\lambda}{2} \sum_{k=1}^{g} \frac{1}{|C_k|} \sum_{j,k \in C_k} \sum_{i=1}^{n} \left(x_{ij}\beta_j - x_{il}\beta_l \right)^2 \right\}, \tag{1.10}$$

which is not convex. In Equation (1.10), $\mathcal{C} = (C_1, \ldots, C_g)$ stands for a partition of the p covariates into g groups and $|C_k|$ represents the size of the k-th group. Since no global optimization of (1.10) can be obtained, the authors proposed to come up with a sub-optimal solution by alternating the optimization over $\boldsymbol{\beta}$ assuming $\mathcal{C}$ is known, then over $\mathcal{C}$ assuming $\boldsymbol{\beta}$ known. The tuning parameters δ, λ and g were proposed to be selected via cross-validation. The latter method was shown to yield good predictive performances both on real simulated data.

Variable clustering in Bayesian linear regression

The field of Bayesian linear regression has been weakly impacted by the advent of methodologies that combine variable selection and variable clustering. However, one contribution is worth mentioning in this review, even if it was primarily developed for Probit regression. This contribution is the *Multiple Bayesian Elastic Net* (MBEN). The latter methodology was introduced in the PhD thesis of H. Yang [1] but still unpublished at the time this review is written. H. Yang and colleagues proposed the following hierarchical prior:

$$\begin{cases} \beta_j | \mu_j, \alpha_j, \lambda, \tau^2 \sim \mathcal{N}\left(\mu_j, \tau^2(\alpha_{k_j} + \lambda)^{-1}\right) \\ \lambda \sim \text{Gamma}(\boldsymbol{r_0}, \boldsymbol{s_0}); \tau^2 \sim \text{Inverse-Gamma}(\boldsymbol{c_0}, \boldsymbol{d_0}) \\ \mu_j | \alpha_j, D, Z, \lambda \sim Z\delta_0(\mu_j) \times \left(\frac{\alpha_j}{\alpha_j + \lambda}\right)^{1/2} + (1 - Z) \times D \\ Z | \pi \sim \text{Bernoulli}(\pi); \pi \sim \text{Beta}(1, \boldsymbol{\alpha_0}); \\ \alpha_j | \gamma \sim \text{Inverse-Gamma}(1, \frac{\gamma}{2}) \\ \gamma | z \sim \text{Gamma}(z\boldsymbol{a_0} + (1-z)\boldsymbol{a_1}, z\boldsymbol{b_0} + (1-z)\boldsymbol{b_1}) \\ D \sim \text{DP}(\boldsymbol{\alpha_0}, D_0) \\ D_0 \equiv \mathcal{N}\left(\mu_j; \boldsymbol{c_1}, \boldsymbol{d_1}\right) \end{cases} \tag{1.11}$$

[1] http://www.stat.duke.edu/people/theses/hy35.pdf

In model (1.11), DP$(\boldsymbol{\alpha_0}, D_0)$ stands for the Dirichlet Process (DP) distribution of concentration parameter $\boldsymbol{\alpha_0}$ and base distribution D_0; and $\delta_0(.)$ denotes the Dirac distribution concentrated on zero.

Model (1.11) assumes the regression coefficient β_j to be drawn from a Gaussian distribution of mean μ_j and variance α_j. μ_j is then assumed to equal 0 with probability π or to be drawn from a Dirichlet Process distribution with probability $(1-\pi)$. Because the Dirichlet Process distribution is almost surely discrete, the (μ_j, α_j)'s are expected to take a finite number L of values. Depending on α_0, L can be taken to be smaller than p. As a consequence, covariates having a large probability to share the same mean and variance, are likely to be clustered together.

Model (1.11) has the appealing property of allowing shrinkage of regression coefficients not only around zero. However, in a fully Bayesian approach with a somewhat complicated prior like (1.11), the choice of the eight hyperparameters might be critical. The authors explained their choice but no guidelines were given for a general context. Moreover, the algorithm proposed for making inference, namely the Gibbs sampler, requires the calculation of inverse of a $p \times p$ matrix. Such calculation is often unaffordable and consequently tend to lessen the interest for the latter approach in high dimensional settings.

MBEN showed however improved predictive performances compared to LASSO, Ridge regression and Elastic net on both simulated and real data of manageable size.

1.2 Contribution of this thesis

The research presented through this thesis is a contribution to the field of variable clustering in high-dimensional linear regression. The primary developments proposed so far in the domain have been mostly based on penalized least squares problems (see Section 1.1.2). We introduce here a new framework, the *CLuster-wise Effect REgression* (CLERE), as a model-based approach for variable clustering in linear regression. This new framework proposes a hierarchical modeling of the regression coefficients as un-

observed random variables distributed according to a mixture of g Gaussians, i.e.

$$\beta_j \sim \sum_{k=1}^{g} \pi_k \mathcal{N}(b_k, \gamma^2). \tag{1.12}$$

In other words, we assume for each regression coefficient β_j the existence of a multinomial distributed unobserved random variable, $\mathbf{z}_j = (z_{j1}, \ldots, z_{jg})$ of parameter $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)'$, such as β_j is drawn from the k-th component of the mixture when $z_{jk} = 1$. Our model can therefore be written as

$$\begin{cases} y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \\ \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \\ \beta_j | \mathbf{z}_j \sim \mathcal{N}\left(\sum_{k=1}^{g} b_k z_{jk}, \gamma^2\right) \\ \mathbf{z}_j = (z_{j1}, \ldots, z_{jg}) \sim \mathcal{M}(\pi_1, \ldots, \pi_g). \end{cases} \tag{1.13}$$

With such a hierarchical definition, model (1.13) can be interpreted as a Bayesian approach. However, to be fully Bayesian, a prior distribution for $\boldsymbol{\theta} = (\beta_0, \boldsymbol{b}, \boldsymbol{\pi}, \sigma^2, \gamma^2)$ would have been necessarily defined. Instead, we proposed to estimate $\boldsymbol{\theta}$ by maximizing the (marginal) log-likelihood, $\log p(\mathbf{y}|\mathbf{X}; \boldsymbol{\theta})$. This partially Bayesian approach is referred to as *Empirical Bayes* (EB) [9].

Our method differs from the Spike and Slab prior presented in Section 1.1.1 in many aspects. First, it does not restrict the number of components to equal 2. Secondly, each component is allowed to have a mean different from zero. Finally, as just explained, our approach is not fully Bayesian since no prior distribution is assumed for $\boldsymbol{\theta} = (\beta_0, \boldsymbol{b}, \boldsymbol{\pi}, \sigma^2, \gamma^2)$. Nevertheless, our method might be considered as an EB generalization of the Spike and Slab methodology. Beyond being an *Empirical Bayes* approach, our model also differs from the *Multiple Bayesian Elastic Net* (MBEN) presented in 1.1.2. Indeed, contrarily to MBEN we propose to select the number of component using information based criteria among a user-specified list of a possible number of components.

Our model offers flexibility as illustrated throughout this manuscript and has a markedly reduced number of parameters: only $2g + 2$ parameters. These two important properties come however with a challenging maximum likelihood estimation. We therefore proposed and implemented in this thesis some practical strategies to make inference in

our model. The present manuscript has two parts that are described below in Sections 1.2.1 and 1.2.2.

1.2.1 Part 1: Introduction to the CLERE methodology

This first part of the thesis contains two chapters. Chapter 2 introduces the CLERE methodology. This chapter aims at presenting the rationale of our approach and the related challenges of the estimation. It also recalls a little of the literature review presented in Sections 1.1.1 and 1.1.2. Moreover, it illustrates through numerical experiments, on simulated and real data, the predictive performances of the CLERE method compared to classical dimension reduction techniques. This chapter has been accepted for publication in the *Journal de la Société Francaise de Statistiques*.

Chapter 3 presents two enhancements of the CLERE model. First, a more computationally efficient algorithm is proposed for parameter estimation. Secondly, the CLERE model is enhanced with the possibility of constraining the first component to have its mean equal to 0, i.e. $b_1 = 0$. This enhancement mainly aimed at facilitating the interpretation of the model. Indeed when b_1 is set to 0, variables assigned to the cluster associated with b_1 might be considered less relevant than other variables. Those two new features were implemented in a C++ program available through the `R` package `clere`. This chapter has been submitted for publication in the *Journal of Statistical Software*.

1.2.2 Part 2: Extensions of the CLERE methodology

The second part of this thesis explores possible extensions of the CLERE methodology through four small chapters. In Chapter 4, we propose an extension of the CLERE methodology to handle binary response data. This extension is proposed under the Generalized Linear Model framework for Probit regression. Extension to logistic regression could only be achieved under approximations since the logistic distribution is not conjugate with Gaussian-related priors. Some of those approximations are presented but no actual implementation was proposed for this model. In turn, extension to Probit regression is done at almost no cost. Numerical experiments are presented to

illustrate the predictive performances of CLERE-probit compared to standard variable selection techniques for binary response regression.

The mixture of Gaussians underlying the CLERE method assumes that all components in the mixture have equal variances (all equal to γ^2). In Chapter 5, we explore through numerical experiments the consequences of relaxing that assumption, i.e. of assuming a different variance per component.

In Chapter 6, we explore the consequences of allowing variable clustering regardless the sign of their associated regression coefficient but only using the magnitude of the latter. The predictive performances of that extension were assessed through numerical experiments on both simulated and real data.

Chapter 7 is a general conclusion of this research and tackles some promising perspectives.

Part I

Introduction to the CLERE methodology

Chapter 2

The CLERE methodology

2.1 Introduction

We consider the standard linear regression model defined as

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i, i = 1, \ldots, n. \tag{2.1}$$

For an individual i, y_i is the observed response, x_{ij} is an observed value for the j-th covariate and ε_i is an error term often assumed to be normally distributed. The ε_i's are also assumed to be independent and identically distributed. β_j is the regression coefficient associated with the j-th covariate. We denote $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ as the vector of regression coefficients.

The dimension p of model (2.1) is tightly related to both its interpretability and ability to yield reliable predictions. It is well known that the more covariates we add to the model the harder becomes its interpretation. Besides, Stein established in [44] that the mean prediction squared error attributable to a linear regression model increases with its dimension. Reducing the model dimension therefore pursues the goal of minimizing prediction error while keeping the model interpretable. This problem, also referred to as the *bias-variance trade-off* [19], becomes increasingly challenging as the set of covariates exceeds the sample size. This high dimensional framework has fueled a number of researches during the last three decades.

Variable selection is one of the most popular approaches for reducing dimensionality. Although it has a direct impact on p, traditional stepwise algorithms for finding the best subset of predictors had a mitigated success because of their heavy computational burden [19]. At a more affordable computational cost, penalized approaches were introduced as an efficient alternative for variable selection. Penalized approaches impose a constraint on $\boldsymbol{\beta}$ that generally depends on a tuning parameter. This parameter can be selected over a grid of values either minimizing the out-of-sample prediction error (cross validation) or using information based criteria like AIC or BIC [53, 38]. Among the most emblematic methods belonging to this second family of approaches we can refer to the least absolute shrinkage and selection operator (LASSO) [45] and the elastic net [52].

Another relevant approach for reducing dimensionality consists of identifying patterns under which covariates can be pooled together. This idea was recently implemented in a gene expression study [32]. In that study, groups of genes were built from hierarchical clustering of gene expression levels. The authors created surrogate covariates by averaging gene expression levels within each group. Those new predictors were afterwards included in a linear regression model, replacing the primary variables. The major limitation in this approach is the independence between the prediction and clustering parts. Consequently, effects of the surrogate covariates can be diluted if they contain primary variables with either no effect or even opposite effects on the response. To sidestep the previous limitation, Bondell and Reich [7] introduced in 2008 the octogonal shrinkage and clustering algorithm for regression (OSCAR). The OSCAR methodology belongs to the family of penalized approaches. It imposes a constraint on $\boldsymbol{\beta}$ that is a weighted combination of the L^1 norm and the pairwise L^∞ norm. Upper-bounding the pairwise L^∞ norm enforces the covariates to have close coefficients. When the constraint is strong enough, closeness translates into equality achieving thus a grouping property. More recently, a generalization to the OSCAR methodology was proposed in [41]. One major advantage of this new approach, namely the pairwise absolute clustering and sparsity (PACS) was the reduced computational cost. In the aftermath of OSCAR and PACS, other methodologies aiming at simultaneously performing parameter estimation and clustering were proposed. We can for instance refer to the approaches of Petry [35] and She [42] which also mixed L^1 and pairwise L^∞ penalties or those of Daye [12] and Shen [43] based on alternative penalties.

In line with the latter works, we introduce the clusterwise effect regression (CLERE), a new methodology aiming at simultaneously performing regression and clustering of covariates. CLERE considers each β_j no longer as a fixed parameter but as an unobserved random variable (Assumption *A1*) following a mixture of Gaussian distributions (Assumption *A2*) with an arbitrary number of components (Assumption *A3*). The means of each component in the mixture are moreover assumed unequal (Assumption *A4*). Under assumptions *A1* and *A2* our approach shows strong similarities with a Bayesian approach for variable selection known as the spike and slab model [30, 23]. Despite those similarities, assumptions *A3* and *A4* drive the main differences between the two methods. Indeed, in spike and slab models the number of components is restricted to two and the means of each component of the mixture are assumed equal to zero. In addition to those two differences, we recall an important issue which is that our primary goal is not variable selection like for spike and slab models but variable clustering. The clustering of the covariates is achieved using the probability of each β_j to be drawn from the same component of the mixture, given the data and the estimated parameters.
The present paper is organized as follows. Section 2.2 presents our model. In Section 2.3, a maximum likelihood strategy is presented to estimate the model parameters as well as a criterion to select the number of latent groups. Section 2.4 presents numerical experiments both on simulated and real data. In this section the predictive performances of our model are compared to standard approaches for dimension reduction in high dimensional linear regression models. The perspectives of this research are discussed in Section 2.5.

2.2 Model definition and notation

2.2.1 Model definition

As aforementioned, the number of predictors may be very large with respect to the number of samples. It is therefore impossible to uniquely estimate each coefficient β_j. However, we may assume the existence of g latent groups of covariates within which the β_j's are sufficiently close to one another that all of them may be summarized by their average. Among possible mathematical translations of the latter assumption, we

propose to consider the β_j's no longer as fixed effect parameters but as unobserved independent random variables following a Gaussian mixture distribution:

$$\beta_j \sim \sum_{k=1}^{g} \pi_k \mathcal{N}\left(b_k, \gamma^2\right). \tag{2.2}$$

In other words, we assume for each β_j the existence of a multinomial distributed random variable, $\mathbf{z} = (z_{j1}, \ldots, z_{jg})$ of parameter $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)'$, such as β_j is drawn from the k-th component of the mixture when $z_{jk} = 1$. Our model can then be written as

$$\begin{cases} y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \\ \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \\ \beta_j | \mathbf{z}_j \sim \mathcal{N}\left(\sum_{k=1}^{g} b_k z_{jk}, \gamma^2\right) \\ \mathbf{z}_j = (z_{j1}, \ldots, z_{jg}) \sim \mathcal{M}\left(\pi_1, \ldots, \pi_g\right). \end{cases} \tag{2.3}$$

Parameter β_0 is associated with a constant variable. Since our primary aim is variable clustering, we did not considered β_0 as random in model (2.3).

2.2.2 Notation

In all subsequent sections of the paper the following notation hold: $\mathbf{y} = (y_1, \ldots, y_n)'$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$, $\mathbf{X} = (x_{ij})$, $\mathbf{Z} = (z_{jk})$, $\mathbf{b} = (b_1 \ldots b_g)'$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)'$.Moreover, $\log p(\mathbf{y}|\mathbf{X}; \boldsymbol{\theta})$ denotes the log-likelihood of model (2.3) assessed for the parameter $\boldsymbol{\theta} = \left(\beta_0, \boldsymbol{b}, \boldsymbol{\pi}, \sigma^2, \gamma^2\right)$ and $\mathcal{Z}$ the set of $p \times g$-matrices defined as

$$\left(z_{jk}\right)_{1 \leq j \leq p, 1 \leq k \leq g} \in \mathcal{Z} \implies \forall j \in \{1, \ldots, p\}, \begin{cases} \exists! k \text{ such as } z_{jk} = 1 \\ \text{if } k' \neq k \text{ then } z_{jk} = 0. \end{cases}$$

2.2.3 Bayes or Empirical Bayes?

With such a hierarchical definition, model (2.3) can be interpreted as a Bayesian approach. However, to be fully Bayesian a prior distribution for $\boldsymbol{\theta} = \left(\beta_0, \boldsymbol{b}, \boldsymbol{\pi}, \sigma^2, \gamma^2\right)$ would have been necessary. Instead, we propose to estimate $\boldsymbol{\theta}$ by maximizing the

(marginal) log-likelihood, $\log p(\mathbf{y}|\mathbf{X};\boldsymbol{\theta})$. This partially Bayesian approach is referred to as *Empirical Bayes* (EB) [9]. Our choice for an EB approach was motivated by the number of parameters we have to estimate. This number, $2(g+1)$, is often small with respect to the sample size n. In this situation, posterior distributions obtained with an EB approach and with a fully Bayesian approach are expected to be close [34].

2.2.4 Degeneracy of the likelihood

To prevent degeneracy of the likelihood, which often occurs in mixture models [4], constraints are generally imposed to the space of hidden variables [36]. In this work the following constraint is proposed:

$$\forall k = 1,\ldots,g \ \sum_{j=1}^{p} z_{jk} \geq 1. \tag{2.4}$$

This constraint basically requires none of the groups to be empty.

2.3 Estimation, prediction, clustering and model selection

2.3.1 Maximum Likelihood Estimation

The log-likelihood $\log p(\mathbf{y}|\mathbf{X};\boldsymbol{\theta})$ is defined as

$$\log p(\mathbf{y}|\mathbf{X};\boldsymbol{\theta}) = \log\left[\sum_{\mathbf{Z}\in\mathcal{Z}} \int_{\mathbb{R}^p} p(\mathbf{y},\boldsymbol{\beta},\mathbf{Z}|\mathbf{X};\boldsymbol{\theta})\mathrm{d}\boldsymbol{\beta}\right]. \tag{2.5}$$

The likelihood cannot be calculated analytically as it involves integration over unobserved data $(\boldsymbol{\beta},\mathbf{Z})$. A direct maximization for estimating $\boldsymbol{\theta}$ is consequently impossible.

The expectation-maximization (EM) algorithm [13] has been introduced to perform MLE in the presence of unobserved data. The EM algorithm is an iterative method, which starts with initial estimates of the parameters and updates these estimates at each iteration until convergence is achieved. We propose in the following subsections

its implementation in the special case of model (2.3).

Initialization

The algorithm is initialized using primary estimates ${\beta_j}^{(0)}$ of each β_j. The latter can be either obtained from univariate regression coefficients or from penalized approaches like the LASSO or the ridge regression. Model (2.2) is then fitted using $\beta^{(0)} = \left(\beta_1^{(0)}, \ldots, \beta_p^{(0)}\right)$ as observed data to produce starting values $\mathbf{b}^{(0)}$, $\boldsymbol{\pi}^{(0)}$ and ${\gamma^2}^{(0)}$ respectively for parameters $\mathbf{b}$, $\boldsymbol{\pi}$ and γ^2. An initial partition $\mathbf{Z}^{(0)} = \left(z_{jk}^{(0)}\right) \in \mathcal{Z}$ is determined as

$$\forall j \in \{1, \ldots, p\},\ z_{jk}^{(0)} = \begin{cases} 1 & \text{if } k = \text{argmin}_{k' \in \{1,\ldots,g\}} \left({\beta_j}^{(0)} - b_{k'}^{(0)}\right)^2 \\ 0 & \text{otherwise.} \end{cases}$$

β_0 and σ^2 are initialized using $\beta^{(0)}$ as following:

$$\beta_0^{(0)} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j^{(0)}x_{ij}\right) \text{ and } {\sigma^2}^{(0)} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \beta_0^{(0)} - \sum_{j=1}^{p}\beta_j^{(0)}x_{ij}\right)^2.$$

The EM algorithm only ensures to converge towards a local maximum of the likelihood. Our approach is therefore potentially subjected to this limitation. Nevertheless, the stochasticity introduced during the *E*-step (see Section 2.3.1) tends to lessen the impact of the starting point. This has already been studied in a general context [10]. Indeed, we illustrate further in Section 2.4.1 that the choice of the starting point is not critical to our method.

(Stochastic) Expectation step

At iteration d of the algorithm, the log-likelihood of the complete data $\log p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(d)})$ has the following expression:

$$\begin{aligned}\log p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(d)}) &= \log p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{X}; \beta_0^{(d)}, \sigma^{2(d)}) + \log p(\boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \mathbf{b}^{(d)}, \boldsymbol{ß}^{(d)}, \gamma^{2(d)}) \\ &= -\frac{n}{2}\log\left(2\pi\sigma^{2(d)}\right) - \frac{1}{2\sigma^{2(d)}}\sum_{i=1}^{n}\left(y_i - \beta_0^{(d)} - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \\ &\quad -\frac{p}{2}\log\left(2\pi\gamma^{2(d)}\right) + \sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}\left(\log\pi_k^{(d)} - \frac{\left(\beta_j - b_k^{(d)}\right)^2}{2\gamma^{2(d)}}\right).\end{aligned}$$

In classical EM algorithm, the E-step requires, at each iteration, the calculation of the expectation of the log-likelihood of the full data $\log p\left(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(d)}\right)$, with respect to the conditional distribution of unobserved data given observed data. This quantity generally denoted as $Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(d)}\right)$, does not have a closed form in model (2.3). We therefore approximate $Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(d)}\right)$ using Monte Carlo simulations. This stochastic version of the EM algorithm was introduced in [46] under the name of Monte Carlo EM (MCEM) algorithm. A Gibbs sampling scheme is proposed to generate draws from the probability distribution $p\left(\boldsymbol{\beta}, \mathbf{Z}|\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}^{(d)}\right)$. In model (2.3), Gibbs sampling requires the definition of the conditional distributions $p\left(\boldsymbol{\beta}|\mathbf{Z}, \mathbf{y}, \mathbf{X}; \boldsymbol{\theta}^{(d)}\right)$ and $p\left(\mathbf{Z}|\boldsymbol{\beta}, \mathbf{y}, \mathbf{X}; \boldsymbol{\theta}^{(d)}\right)$. The latter distributions are given in Equations (2.6) and (2.7). Details about how those distributions were derived are given in Section 2.6.

$$\begin{cases} \boldsymbol{\beta}|\mathbf{Z}, \mathbf{y}; \boldsymbol{\theta}^{(d)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(d)}, \boldsymbol{\Sigma}^{(d)}\right) \\ \boldsymbol{\mu}^{(d)} = \left[\mathbf{X}'\mathbf{X} + \frac{\sigma^{2(d)}}{\gamma^{2(d)}}\mathrm{I}_p\right]^{-1}\mathbf{X}'\left(\mathbf{y} - \beta_0^{(d)}\mathbb{I}_n\right) + \frac{\sigma^{2(d)}}{\gamma^{2(d)}}\left[\mathbf{X}'\mathbf{X} + \frac{\sigma^{2(d)}}{\gamma^{2(d)}}\mathrm{I}_p\right]^{-1}\mathbf{Z}\mathbf{b}^{(d)} \\ \boldsymbol{\Sigma}^{(d)} = \sigma^{2(d)}\left[\mathbf{X}'\mathbf{X} + \frac{\sigma^{2(d)}}{\gamma^{2(d)}}\mathrm{I}_p\right]^{-1} \end{cases} \quad (2.6)$$

and

$$p\left(z_{jk} = 1|\boldsymbol{\beta}; \boldsymbol{\theta}^{(d)}\right) \propto \pi_k^{(d)}\exp\left(-\frac{\left(\beta_j - b_k^{(d)}\right)^2}{2\gamma^{2(d)}}\right). \quad (2.7)$$

Now suppose we have sampled $\left\{\left(\boldsymbol{\beta}^{(1,d)}, \mathbf{Z}^{(1,d)}\right), \ldots, \left(\boldsymbol{\beta}^{(M_d,d)}, \mathbf{Z}^{(M_d,d)}\right)\right\}$ from $p\left(\boldsymbol{\beta}, \mathbf{Z}|\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}^{(d)}\right)$ and verifying the condition (2.4); the approximated E-step can then be written as follows:

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(d)}\right) = \mathbb{E}\left[\log p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(d)})|\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}^{(d)}\right] \approx \frac{1}{M_d}\sum_{m=1}^{M_d} \log p(\mathbf{y}, \boldsymbol{\beta}^{(m,d)}, \mathbf{Z}^{(m,d)}|\mathbf{X}; \boldsymbol{\theta}^{(d)}). \tag{2.8}$$

The computational time and the convergence of the algorithm is governed by the choice of M_d. In [46], the authors suggested using small values for M_d (around 20) when starting the algorithm and increases this value along with the number of iterations. In this paper however M_d was set to a constant large value.

Maximization step

The M-step consists of maximizing $Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(d)}\right)$ with respect to $\boldsymbol{\theta}$. We get the following update equations:

$$\pi_k^{(d+1)} = \frac{1}{M_d p}\sum_{m=1}^{M_d}\sum_{j=1}^{p} z_{jk}^{(m,d)}, \tag{2.9}$$

$$b_k^{(d+1)} = \frac{1}{M_d p \pi_k^{(d+1)}}\sum_{m=1}^{M_d}\sum_{j=1}^{p} z_{jk}^{(m,d)}\beta_j^{(m,d)}, \tag{2.10}$$

$$\gamma^{2(d+1)} = \frac{1}{M_d p}\sum_{m=1}^{M_d}\sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}^{(m,d)}\left(\beta_j^{(m,d)} - b_k^{(d+1)}\right)^2, \tag{2.11}$$

$$\beta_0^{(d+1)} = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - \sum_{j=1}^{p}\left(\frac{1}{M_d}\sum_{m=1}^{M_d}\beta_j^{(m,d)}\right)x_{ij}\right], \tag{2.12}$$

$$\sigma^{2(d+1)} = \frac{1}{nM_d}\sum_{m=1}^{M_d}\sum_{i=1}^{n}\left(y_i - \beta_0^{(d+1)} - \sum_{j=1}^{p}\beta_j^{(m,d)}x_{ij}\right)^2. \tag{2.13}$$

2.3.2 Prediction and Clustering

If $\mathbf{X}^v$ denotes a new design matrix for which we want to predict the response $\mathbf{y}^v$, then we can define the predicted response $\widehat{\mathbf{y}}$ as

$$\widehat{\mathbf{y}} = \mathbf{X}^v \mathbb{E}\left[\beta|\mathbf{y}, \mathbf{X}; \widehat{\boldsymbol{\theta}}\right], \tag{2.14}$$

where $\widehat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of $\boldsymbol{\theta}$. The clustering of the covariates is achieved using the probability of each β_j to be drawn from the same component of the mixture, given the data and the estimated parameters. Therefore the j-th covariate is assigned to the k-th cluster if

$$\forall l = 1, \ldots, g, \mathbb{E}\left[z_{jk}|\mathbf{y}, \mathbf{X}; \widehat{\boldsymbol{\theta}}\right] \geq \mathbb{E}\left[z_{jl}|\mathbf{y}, \mathbf{X}; \widehat{\boldsymbol{\theta}}\right].$$

2.3.3 Model selection

Model (2.3) depends on a tuning parameter g, which is the assumed number of groups of covariates. In few situations, this number can be chosen *a priori*, however in a more general setting a strategy should be proposed to make this choice. The BIC is proposed as a means to select g. This criterion was preferred to other criteria based on estimates of the out-of-sample prediction error like cross-validation (CV) because of its low computational cost. In model (2.3) the number of parameters equals $2(g+1)$. The BIC has therefore the following expression:

$$BIC = -2 \log p(\mathbf{y}|\mathbf{X}; \widehat{\boldsymbol{\theta}}) + 2(g+1) \log(n). \tag{2.15}$$

As the calculation of the likelihood is still intractable, we can derive from Equation (2.5), an approximation of the BIC criterion using Monte Carlo simulations.

2.4 Numerical experiments

In this section, we compare our approach CLERE with standard dimension reduction approaches in terms of prediction error. The methods selected for comparison are the

variable selection using LARS algorithm [15], the ridge regression [21], the elastic net [52], the LASSO [45], PACS [41], the method of Park and colleagues [32] (subsequently denoted AVG) and the spike and slab model [23] (subsequently denoted SS). The first four methods are implemented in freely available R packages `lars` and `glmnet` (for ridge, LASSO and elastic net). Those packages were used with default options. For PACS a R script was released on Bondell's webpage[1]. This R script was however running very slowly. We therefore decided to reimplement it in C++. This led to a 30-fold speed-up in the computational time. Similarly to Bondell's script, our program uses two parameters named `lambda` and `betawt`. In [41], the authors suggest assigning `betawt` with the coefficients obtained from a ridge regression model after the tuning parameter was selected using AIC. In this simulation study we used the same strategy; however the ridge parameter was selected via 5-fold cross validation. 5-fold CV was preferred to AIC since selecting the ridge parameter using AIC always led to estimated coefficients equal to zero. Once `betawt` was selected, `lambda` was chosen via 5-fold cross validation among the following values: 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200 and 500. All other default parameters of their script were unchanged. For the AVG method, we followed the algorithm described in [32] and implemented it in R. We used the R package `spikeslab` to run the spike and slab models. Especially, we used the function `spikeslab` from that package to detect influential variables. The number of iterations used to run the function `spikeslab` was 2000 (1000 discarded). When running CLERE, the number of EM iterations as well as the number M_d of Monte Carlo samples was set to 1000. The number of groups for CLERE was chosen between 1 and 9. In all experiments, CLERE was initialized using the estimated univariate regression coefficients as explained in Section 2.3.1. Our C++ implementations of PACS and CLERE are available on request.

2.4.1 Simulated data

Description

The simulated data are presented under three scenarios. For each scenario, 100 training data sets were simulated from the standard linear regression model (2.1). All training data sets consist of $n = 50$ simulated individuals with $p = 100$ variables. In each sce-

[1] *http://www4.stat.ncsu.edu/bondell/Software/PACS/PACS.R.r*

nario, a validation set consisting of 5000 individuals was used to calculate the scaled mean squared prediction error.

If $\left(\mathbf{y}^t, \mathbf{X}^t\right)$ and $\left(\mathbf{y}^v, \mathbf{X}^v\right)$ are respectively the training and validation data sets, then the scaled mean squared prediction error MSE is calculated as:

$$\text{MSE} = \frac{\|\mathbf{y}^v - \widehat{\mathbf{y}}\left(\mathbf{X}^v, \mathbf{y}^t, \mathbf{X}^t\right)\|_2}{\|\mathbf{y}^v\|_2}, \tag{2.16}$$

where $\widehat{\mathbf{y}}\left(\mathbf{X}^v, \mathbf{y}^t, \mathbf{X}^t\right)$ is the predicted response and $\|.\|_2$ stands for the L^2 norm. For CLERE, predictions are obtained using Equation (2.14). Each of the methods selected for comparison provides a fitted value $\widehat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$. A predicted response under the design $\mathbf{X}^v$ is then calculated as $\mathbf{X}^v\widehat{\boldsymbol{\beta}}$. In all simulations, design matrices $\mathbf{X}^t$ and $\mathbf{X}^v$ were simulated as independently normally distributed:

$$\mathbf{x}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}\right), \tag{2.17}$$

where $\mathbf{R} = \left(r_{jj'}\right)$ is a $p \times p$ matrix defined by $r_{jj'} = 0.5^{|j-j'|}$. In all scenarios, parameters β_0 and σ^2 equal respectively 0 and 100.
The three scenarios are presented below.

1. In scenario 1, the vector $\boldsymbol{\beta}$ of regression coefficients is given by:

$$\boldsymbol{\beta} = (\underbrace{0, \ldots, 0}_{36}, \underbrace{1, \ldots, 1}_{28}, \underbrace{3, \ldots, 3}_{20}, \underbrace{7, \ldots, 7}_{12}, \underbrace{15, \ldots, 15}_{4})'.$$

2. In scenario 2, the vector $\boldsymbol{\beta}$ of regression coefficients is given by:

$$\boldsymbol{\beta} = (\underbrace{0, \ldots, 0}_{36}, \underbrace{4, \ldots, 4}_{28}, \underbrace{24, \ldots, 24}_{20}, \underbrace{124, \ldots, 124}_{12}, \underbrace{624, \ldots, 624}_{4})'.$$

3. In scenario 3, the regression coefficients are chosen uniformally between -10 and +10 :

$$\forall j, \beta_j = -10 + (j-1) \times \frac{20}{99}.$$

Scenarios 1 and 2 were chosen to favor variable selection approaches. In those scenarios indeed 36 out of 100 covariates do not influence the response. Moreover the number of effective variables decreases with their effect size. Scenario 3 was proposed to illustrate the relative predictive performances of CLERE under the assumption that almost all covariates contribute to the response. We also considered three additional scenarios directly deriving from the previous ones. Those scenarios are further denoted as *alternative* scenario 1, 2 and 3. The *alternative* scenario s ($s \in \{1,2,3\}$) is obtained by randomly permuting the regression coefficients in scenario s. These additional scenarios were proposed to explore the performances of our methodology when correlated variables do not necessarily have equal or similar regression coefficients.

Impact of the initialization strategy

We consider in this subsection four initialization schemes based on four initial guesses for the unobserved regression coefficients. In addition to univariate regression, LASSO and ridge regression already mentioned in Section 2.3.1, we also added the elastic net as one possible means to generate initial estimates for the β_j's. We compared the distribution of the maximum likelihood reached and the distribution of prediction error (MSE) for the four initialization strategies using 100 data sets simulated according to scenario 1. As a reference, we also considered the case where the initial guesses were actually the true regression coefficients used to generate the data. Figure 2.1 illustrates the results of that comparison. No significant difference was noticeable between the four initialization strategies. Indeed, none of them seemed to systematically lead to lower or higher likelihood. This is supported by the very large p-value (P = 0.944) obtained after performing a Fisher's test to test for a potential difference between the four strategies. We can therefore argue that initialization is not a critical issue for our method. Similar results have already been observed for stochastic versions of the EM algorithm in [10] from which our implementation partially derives. We also compared the distribution of the MSE for each of the initialization strategies. No difference in terms of MSE was noticeable (P = 0.684).

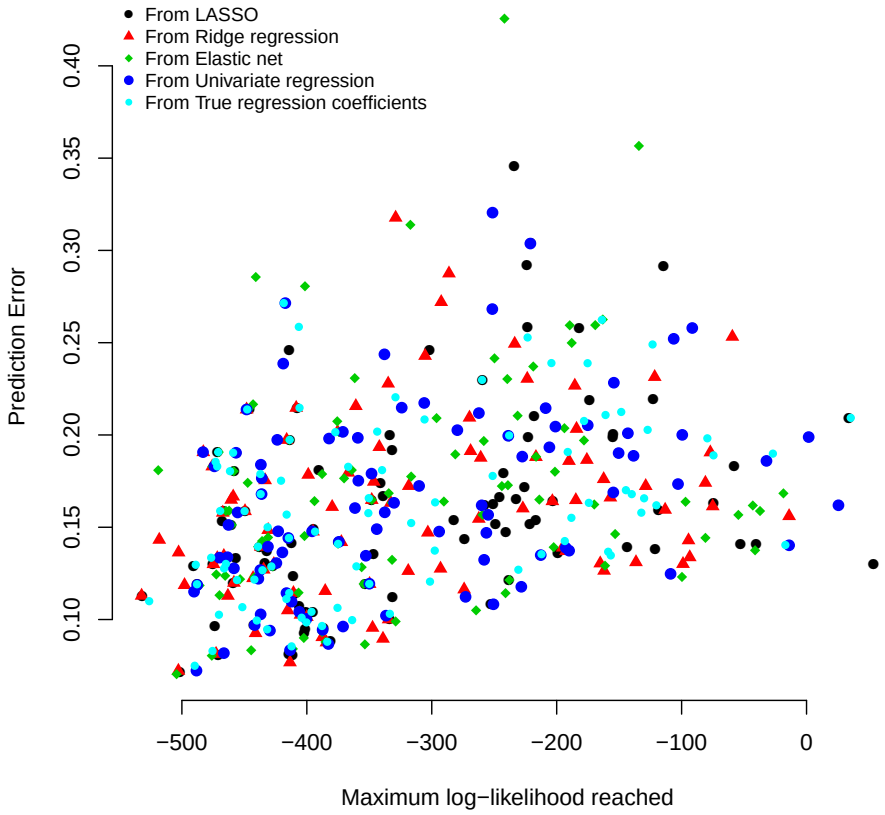

Figure 2.1: Co-distribution of the MSE and the maximum log-likelihood reached for all the initialization strategies. No significant difference is noticeable in the maximum likelihood reached (F-value = 0.189 - P-value = 0.944) nor in the prediction error (F-value = 0.57 - P = 0.684).

Results

Table 2.1 summarizes the MSE calculated under each scenario. Using this measure, CLERE has the best average rank over all scenarios considered. We also considered a measure of model complexity being either the number of unique non-zero parameters or simply the number of parameters for CLERE. Using the latter measure, the present simulation study illustrates that CLERE selects the simplest model in all the scenarios considered.

In scenario 1 and 2, clusters of covariates were simulated. These clusters correspond to covariates having equal regression coefficients. The predictive performances of all the methods were influenced by the separation between the clusters. Indeed, all methods increased their performances along with the clusters separation. This improvement was however much more noticeable for CLERE which outperformed its competitors in scenario 2.
The predictive performances of the methods were also influenced by the correlations between the covariates. This is illustrated by comparing each scenario with its *alternative* counterpart. CLERE robustly showed good performances in all *alternative* scenarios. Especially, it yields the best predictive performance in *alternative* scenarios 1 and 2. In Scenario 3, the regression coefficients were not separated at all. However, CLERE managed to yield competitive performances both under the initial and the *alternative* scenarios.
We also report for CLERE the distribution over 100 simulated data sets of the estimated b_k's. This is shown in Figure 2.2 under scenarios 1, 2 and their *alternative* counterparts. Estimates of the b_k's are known up to a permutation. It was therefore not straightforward to compare estimates from a data set to another. To achieve a global comparison of the estimates across all simulated data sets, we selected, for each data set, the permutation that minimized the bias.

2.4.2 Real data

Description

We used in this section the real data set *mice* from the `spls` R package. This data set consists of $n = 60$ mice for which the expression of 83 gene transcripts from liver tissues was measured and $p = 145$ microsatellite markers were genotyped. For more details about this data set please refer to [11]. One challenging issue of modern Genetics is to bridge gene expression levels with variations in the genomic sequence. Microsatellite markers are such variations. The latter markers are discrete quantitative variables taking values in $\{1, 2, 3\}$, while gene expression levels are real quantitative variables. Instead of considering each transcript as a response, we performed a principal component analysis (PCA) over the gene expression data to come up with a reduced number of outcomes. This PCA did not involve the microsatellite markers. The PCA was per-

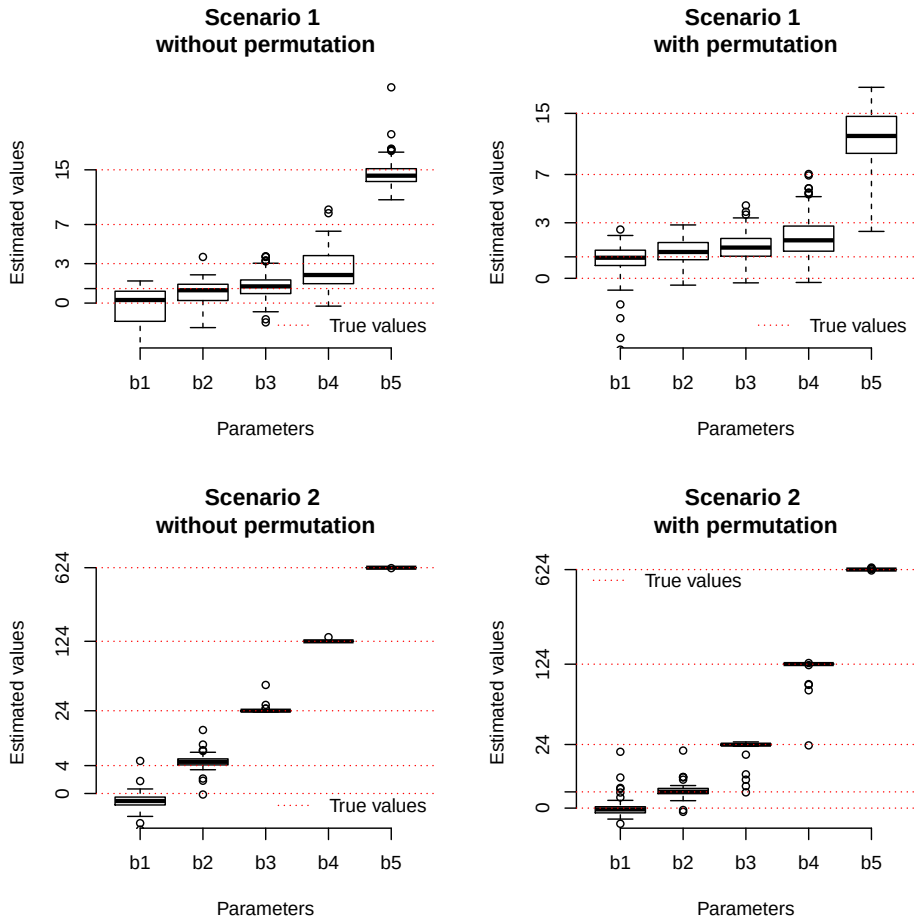

Figure 2.2: Distributions of the maximum likelihood estimates for parameter **b** over 100 simulated data sets under scenarios 1 and 2.

formed using the function `dudi.pca` implemented in the R package `ade4`. The first five principal components (PC) accounted for more than 92% of the total inertia. We then proposed a linear regression model for each of those selected PCs using the microsatellites markers as covariates. The selected PCs are subsequently denoted $PC1, \ldots, PC5$. Since no proper validation data sets were available, all methods were compared in terms of out-of-sample prediction error estimated via 5-fold cross-validation (CV).

Overall results

Table 2.2 summarizes the MSE for each selected PC and each method. Similarly to numerical experiments on simulated data, variable selection using the LARS algorithm yielded very large prediction error for each PC. All other methods had however comparable prediction error. Using the averaged rank as an indicator of overall performance, CLERE was the second best method. The first place was shared by ridge regression, PACS and the Spike and Slab method. CLERE showed the best predictive performances for PC4 and PC5 (no other method was best twice) and was among the most parsimonious methods with the Spike and Slab method and PACS.

		averaged 5-fold CV statistic (Std. Err)	Averaged number of parameters (Std. Dev.)	MSE Rank
*PC*1	LARS	293.57 (125.43)	47 (0)	8
	LASSO	1.26 (0.15)	6.8 (10.55)	4
	Ridge	1.07 (0.04)	145 (0)	3
	Elastic net	2.13 (0.81)	18.6 (21.3)	7
	CLERE	1.31 (0.1)	4 (0)	6
	AVG	1.29 (0.05)	11.4 (9.24)	5
	PACS	**1.03 (0.01)**	0.2 (0.45)	**1**
	SS	1.04 (0.01)	0.2 (0.45)	2
*PC*2	LARS	31.72 (10.76)	47 (0)	8
	LASSO	**0.95 (0.04)**	10.2 (4.71)	**1**
	Ridge	0.98 (0.08)	145 (0)	2
	Elastic net	1.14 (0.13)	43.2 (28.14)	5
	CLERE	1.17 (0.2)	4 (0)	6
	AVG	1.28 (0.14)	21.8 (3.11)	7
	PACS	1.03 (0.05)	5.2 (3.27)	4
	SS	0.99 (0.03)	1 (1.22)	3
*PC*3	LARS	18.74 (3.67)	47 (0)	8

	LASSO	1.66 (0.69)	22.4 (16.61)	6
	Ridge	**0.96 (0.14)**	145 (0)	**1**
	Elastic net	1.68 (0.68)	29.6 (26.49)	7
	CLERE	1.06 (0.2)	4 (0)	3
	AVG	1.61 (0.71)	21.6 (15.19)	5
	PACS	1.17 (0.11)	4.8 (5.07)	4
	SS	1.04 (0.14)	3 (2.24)	2
*PC*4	LARS	30.97 (6.97)	47 (0)	8
	LASSO	1.29 (0.11)	3.8 (4.32)	5
	Ridge	1.16 (0.03)	145 (0)	4
	Elastic net	1.38 (0.1)	12.2 (12.76)	7
	CLERE	**1.09 (0.03)**	4 (0)	**1**
	AVG	1.35 (0.1)	7.2 (4.66)	6
	PACS	1.13 (0.05)	1.6 (1.82)	3
	SS	1.11 (0.04)	0.6 (0.89)	2
*PC*5	LARS	17.26 (8.03)	47 (0)	8
	LASSO	1.07 (0.04)	0 (0)	3
	Ridge	1.07 (0.04)	145 (0)	3
	Elastic net	1.52 (0.49)	10 (21.81)	7
	CLERE	**1.05 (0.002)**	4 (0)	**1**
	AVG	1.16 (0.07)	4.4 (4.1)	6
	PACS	1.07 (0.04)	1.2 (0.45)	3
	SS	1.09 (0.05)	0 (0)	5

Table 2.2: Out-of-sample prediction error estimated using 5-fold CV for each method and each PC for mice data from [11]. The averaged number of fitted parameters, as a measure of model complexity, is also reported. For each scenario in the table we highlighted in bold font the lowest prediction error (equivalent to best MSE rank) or the lower number of parameters.

Focus on $PC1$

We have illustrated above that CLERE is a competitive method for prediction. In this sub-section we now present how CLERE can be used for interpretation purpose. A focus is therefore laid on $PC1$ as a single response variable. The data were no longer partitioned as previously did for cross-validation.

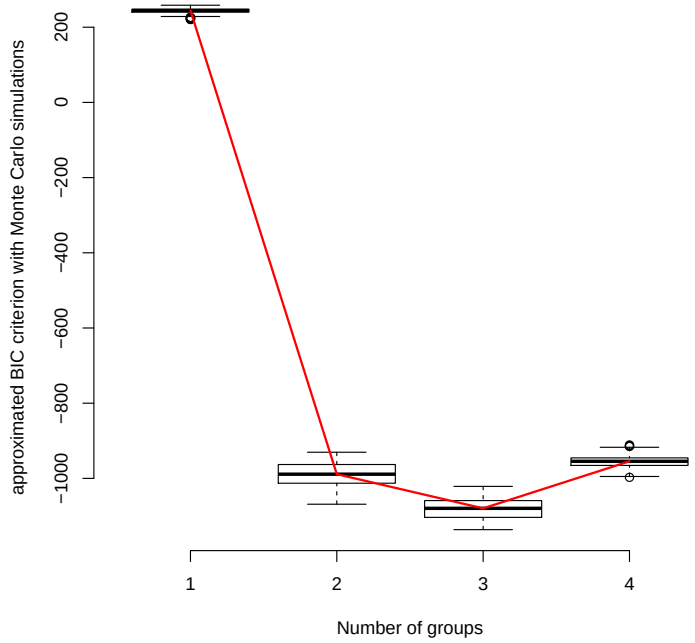

Figure 2.3: Selection procedure for the number of groups. Here $g = 3$ was selected as it minimizes the BIC. The BIC is approximated using Monte Carlo simulations.

Using the whole data set, 3 groups were chosen using the BIC criterion (see Figure 2.3). The estimated parameters are given in Table 2.3. Two groups with moderated positive effects and one group with strong negative effect were identified. In Section 2.3.2, we presented how to make predictions with CLERE using the vector $\mathbb{E}\left[\boldsymbol{\beta}|\mathbf{y},\mathbf{X};\widehat{\boldsymbol{\theta}}\right]$. The latter vector of expectations can be interpreted as a vector of regression coefficients. Consequently, the small estimated value for parameter γ^2 ($\widehat{\gamma^2} = 3.0 \times 10^{-6}$) leads those expectations to be strongly concentrated around the $\widehat{b}_k$'s. CLERE yielded thus a very parsimonious regression model.

The second group, associated with $\widehat{b}_2 = -0.931$, was of interest since it gathers the 11 variables showing the strongest impact on the response according to CLERE. In Table 2.4, we compared for those variables the regression coefficients obtained with LARS, LASSO, ridge regression, elastic net, AVG, PACS and SS. The five methods yielded sign and size consistent regression coefficients for almost all the markers highlighted in Table 2.4. One exception was however noticed for D13Mit16. In addition, CLERE showed that some variables discarded by other methods may still be of interest. Overall this analysis emphasized the ability of CLERE to consistently identify influential covariates using a very parsimonious model. Moreover, this analysis identifies the clusters of markers that may be relevantly investigated for a biological characterization.

2.5 Discussion

We proposed in this paper a new method for simultaneous variable clustering and regression. Our approach showed good predictive performances both on simulated and real data compared to its competitors (see Section 2.4). These good performances were accompanied by a lower complexity in terms of number of fitted parameters. CLERE also brought improvements in terms of interpretability since each fit provides a clustering of the covariates. This work comes in the aftermath of a series of recently published approaches aiming at reducing the dimension in linear regression models by collapsing the covariates into groups. Contrary to those previous works, our approach is not based on penalized least squares problem. However we assumed the existence of a latent structure within the variables that depends only on their unobserved regression coefficients. In such framework, no distributional assumption regarding the covariates is necessary for achieving the clustering. The latent structure is modeled using a Gaussian mixture model whose parameters are estimated via an EM like algorithm. A stochastic version, namely the MCEM, of the latter algorithm was proposed since the E-step was intractable. Even though MCEM has become a standard in many applications, it is noteworthy that its computational cost is not negligible. Indeed, running the estimation with 3 groups on the data set presented in Section 2.4.2 took 30 seconds for CLERE but less than 1 second for the other approaches. Although CLERE seemed to be relatively slow, the estimation time remained manageable. Improvements in speeding up the estimation through parallel computing is a natural perspective of this work,

especially since we are aiming at tackling ultra-high dimensional regression problems in forthcoming research. We proposed in this paper the BIC criterion for choosing the number of latent groups. This criterion was preferred over different existing criteria such as the out-of-sample prediction error because of its small computational cost. Other information-based criteria will be explored in further works.

Variable selection is an appealing extension to our model. In fact, if a constraint is imposed on the parameter space, then CLERE can also be used as a variable selection tool. Such constraint may lead for instance to assume one group k to have its mean b_k and its associated variance equal to zero. This would be a new model which however may be easily derived from the approach presented here. Many applications deal with response variable that may not be continuous. Another promising extension of our model is therefore towards generalized linear models.

2.6 Appendix

2.6.1 Conditional distribution $p\left(\boldsymbol{\beta}|\mathbf{Z},\mathbf{y},\mathbf{X};\boldsymbol{\theta}\right)$

In Section 2.3.1, a Gibbs sampling strategy is proposed to approximate the E-step of our EM like algorithm based on the conditional distribution $p\left(\boldsymbol{\beta}|\mathbf{Z},\mathbf{y},\mathbf{X};\boldsymbol{\theta}\right)$ and $p\left(\mathbf{Z}|\boldsymbol{\beta},\mathbf{y},\mathbf{X};\boldsymbol{\theta}\right)$. We present in this section how we obtained these distributions. Let C denotes the complete log-likelihood:

$$\begin{aligned} C = & -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \\ & + \sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}\left(-\frac{1}{2}\log\left(2\pi\gamma^2\right) - \frac{\left(\beta_j - b_k\right)^2}{2\gamma^2} + \log\pi_k\right) \end{aligned} \tag{2.18}$$

$$\begin{aligned} C &= -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{p}{2}\log\left(2\pi\gamma^2\right) - \frac{1}{2\sigma^2}\left(\mathbf{y}'\mathbf{y} - 2\boldsymbol{\beta}'\mathbf{X}'\mathbf{y} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}\right) \\ &\quad -\frac{1}{2\gamma^2}\left(\boldsymbol{\beta}'\boldsymbol{\beta} - 2\boldsymbol{\beta}'\mathbf{Z}\mathbf{b} + \mathbf{b}'\mathbf{Z}'\mathbf{Z}\mathbf{b}\right) + \sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}\log\pi_k \\ &= K(\mathbf{Z},\sigma^2,\gamma^2,\boldsymbol{\pi}) - \frac{1}{2\sigma^2}\left[\boldsymbol{\beta}'\left(\mathbf{X}'\mathbf{X} + \frac{\sigma^2}{\gamma^2}I\right)\boldsymbol{\beta} - 2\boldsymbol{\beta}'\left(\mathbf{X}'\mathbf{y} + \frac{\sigma^2}{\gamma^2}\mathbf{Z}\mathbf{b}\right)\right], \end{aligned}$$

where $K(\mathbf{Z}, \sigma^2, \gamma^2, \boldsymbol{\pi})$ is defined as

$$K(\mathbf{Z}, \sigma^2, \gamma^2, \boldsymbol{\pi}) = -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{p}{2}\log\left(2\pi\gamma^2\right) - \frac{1}{2\sigma^2}\mathbf{y}'\mathbf{y} + \sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}\log\pi_k.$$

Let $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ respectively be defined as

$$\boldsymbol{\Sigma} = \sigma^2\left(\mathbf{X}'\mathbf{X} + \frac{\sigma^2}{\gamma^2}I\right)^{-1}$$

and

$$\boldsymbol{\mu} = \left(\mathbf{X}'\mathbf{X} + \frac{\sigma^2}{\gamma^2}I\right)^{-1}\left(\mathbf{X}'\mathbf{y} + \frac{\sigma^2}{\gamma^2}\mathbf{Z}\mathbf{b}\right).$$

Then

$$\begin{aligned} C &= K(\mathbf{Z}, \sigma^2, \gamma^2, \boldsymbol{\pi}) - \frac{1}{2}\left[\boldsymbol{\beta}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta} - 2\boldsymbol{\beta}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right] \\ &= H(\mathbf{Z}, \sigma^2, \gamma^2, \boldsymbol{\pi}, \mathbf{b}) - \frac{p}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}|) - \frac{1}{2}\left[(\boldsymbol{\beta} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu})\right], \end{aligned}$$

where $H(\mathbf{Z}, \sigma^2, \gamma^2, \boldsymbol{\pi}, \mathbf{b})$ is defined as

$$\begin{aligned} H(\mathbf{Z}, \sigma^2, \gamma^2, \boldsymbol{\pi}, \mathbf{b}) = &-\frac{n}{2}\log(2\pi\sigma^2) - \frac{p}{2}\log(\gamma^2) - \frac{1}{2\sigma^2}\mathbf{y}'\mathbf{y} + \sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}\log\pi_k \\ &+ \frac{1}{2}\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{1}{2}\log(|\boldsymbol{\Sigma}|). \end{aligned} \tag{2.19}$$

We can identify from Equation (2.19) the density function of a multidimensional normal distribution of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Therefore since $p(\boldsymbol{\beta}|\mathbf{Z}, \mathbf{y}, \mathbf{X}; \boldsymbol{\theta}) \propto p(\boldsymbol{\beta}, \mathbf{Z}, \mathbf{y}|\mathbf{X}; \boldsymbol{\theta})$ we can derive Equation (2.6).

2.6.2 Conditional distribution $p(\mathbf{Z}|\boldsymbol{\beta}, \mathbf{y}, \mathbf{X}; \boldsymbol{\theta})$

If we assume that for all $j \in \{1, \ldots, p\}$, $(z_{j1}, \ldots, z_{j,g})\,|\boldsymbol{\beta}, \mathbf{y}, \mathbf{X}; \boldsymbol{\theta}$ follows a multinomial distribution, then its associated probabilities can be deduced from Equation (2.18). Equation (2.7) derives therefore straightforwardly.

	Without permutation			With permutation		
	100× averaged MSE (Std. Err)	Averaged number of parameters (Std. Dev)	MSE Rank	100× averaged MSE (Std. Err)	Averaged number of parameters (Std. Dev)	MSE Rank
Scenario 1						
LARS	51.1 (1.7)	49 (0)	9	74.1 (2.7)	49 (0)	8
LASSO	15.9 (0.4)	42.4 (4.8)	5	32.1 (0.88)	42.8 (7.3)	4
Ridge	59.7 (0.48)	100 (0)	7	61.6 (0.55)	100 (0)	6
Elastic net	14.3 (0.33)	48.8 (7.1)	3	29 (0.66)	52.3 (9.2)	3
CLERE (g = 5)	15.2 (0.48)	12 (0)	4	**25.3 (0.86)**	12 (0)	**1**
CLERE	16.7 (0.51)	**9.16 (4.8)**	6	25.9 (0.78)	**8.52 (4.8)**	2
AVG	**8.4 (0.33)**	31.5 (6.5)	**1**	36.1 (1.4)	38.2 (8.7)	5
PACS	10.4 (0.28)	35.5 (8.7)	2	74.3 (2.2)	34.1 (15)	9
SS	70.8 (0.7)	87.5 (5.7)	8	73.8 (0.62)	89 (5.5)	7
Scenario 2						
LARS	8.86 (0.68)	49 (0)	7	10.9 (0.56)	49 (0)	6
LASSO	1.15 (0.05)	33.3 (3)	5	3.82 (0.2)	40.5 (3)	3
Ridge	66.4 (0.4)	100 (0)	8	68.6 (0.44)	100 (0)	8
Elastic net	1.23 (0.058)	33.8 (3.1)	6	4.14 (0.22)	40.9 (3)	5
CLERE (g = 5)	0.023 (0.005)	12 (0)	2	0.26 (0.09)	12 (0)	2
CLERE	**0.014 (0.003)**	**15.4 (3)**	**1**	**0.14 (0.07)**	**14.4 (2.9)**	**1**
AVG	0.62 (0.057)	28 (5.4)	3	4.02 (0.19)	40.4 (2.9)	4
PACS	0.817 (0.075)	44.2 (8.6)	4	43.3 (2.9)	41.2 (8.5)	7
SS	98.1 (0.42)	85.4 (8.1)	9	98.9 (0.05)	85.4 (6.8)	9
Scenario 3						
LARS	74.8 (1.9)	49 (0)	8	139 (3.2)	49 (0)	9
LASSO	35.5 (0.75)	46 (6.5)	5	76.4 (1.2)	32.7 (14)	6
Ridge	53 (0.62)	100 (0)	6	73.8 (0.62)	100 (0)	4
Elastic net	24.3 (0.6)	65 (7.7)	4	**61.2 (1.2)**	53.8 (14)	**1**
CLERE (g = 5)	19.4 (0.86)	12 (0)	2	64.3 (2)	12 (0)	2
CLERE	23.8 (1.1)	**9.7 (6.1)**	3	64.7 (2.2)	**8.8 (5.9)**	3
AVG	**8.38 (0.54)**	33.2 (5.9)	**1**	76.1 (1.6)	35.6 (9.5)	5
PACS	70.5 (1.8)	28 (17)	7	94.2 (1.6)	29.7 (18)	8
SS	81.6 (0.47)	89.9 (3.7)	9	86.4 (0.45)	83 (9.5)	7

Table 2.1: Averaged MSE for simulated data under the three scenarios. The average number of non-zero parameters estimated for each method was also reported. When not specified, the number of groups g is chosen using BIC criterion. For each scenario in the table we highlighted in bold font the lowest prediction error (equivalent to best MSE rank) or the lower number of parameters.

$\widehat{\beta}_0$	$\widehat{b}_1$	$\widehat{b}_2$	$\widehat{b}_3$	$\widehat{\pi}_1$	$\widehat{\pi}_2$	$\widehat{\pi}_3$	$\widehat{\gamma^2}$	$\widehat{\sigma^2}$
2.32×10^{-2}	7.87×10^{-2}	-9.32×10^{-1}	7.63×10^{-2}	0.870	0.076	0.054	3.0×10^{-6}	7.35

Table 2.3: Maximum likelihood estimate obtained for CLERE when fitting mice data using $PC1$ as response variable.

Markers	Chromosome	Lars	LASSO	Ridge	Elastic net	CLERE	AVG	PACS	SS
D1Mit87	1	.	.	-0.0265	.	-0.9318	-0.0717	.	.
D3Mit19	3	-0.2347	-0.8962	-0.1940	-0.5670	-0.9316	-0.4253	.	-0.1219
D4Mit149	4	.	.	-0.0855	.	-0.9316	-0.0717	.	-0.2788
D4Mit237	4	-2.7478	-0.8661	-0.1714	-0.4767	-0.9318	-0.0717	.	.
D7Mit56	7	-0.2011	-0.0484	-0.1026	-0.1516	-0.9318	.	.	.
D7Mit76	7	.	-0.0116	-0.1026	-0.1514	-0.9317	.	.	.
D8Mit42	8	0.0119	.	-0.0430	.	-0.9319	.	.	.
D9Mit15	9	-3.1530	-1.6102	-0.2826	-1.0474	-0.9318	-0.4253	.	-0.1887
D13Mit16	13	1.2867	.	0.0530	0.0823	-0.9318	-0.0034	.	.
D15Mit174	15	-1.7012	-0.9335	-0.1149	-0.4312	-0.9319	-0.4253	.	-0.0253
D19Mit34	19	.	.	-0.0449	-0.0303	-0.9317	.	.	.

Table 2.4: Microsatellite markers assigned to the cluster associated with parameter b_2. Regression coefficients for those variables are reported for all compared methods. For CLERE regression coefficients are obtained using $\mathbb{E}\left[\beta|\mathbf{y},\mathbf{X};\widehat{\theta}\right]$. "." means 0.

Chapter 3

Improvements regarding the CLERE methodology

3.1 Introduction

High dimensionality is increasingly ubiquitous in numerous scientific fields including genetics, economics and physics. Reducing the dimensionality is a challenge that most statistical methodologies must meet not only to remain interpretable but also to achieve reliable predictions. In linear regression models, dimension reduction techniques often refer to variable selection. Approaches for variable selection are implemented in publicly available software, that involve the well-known `R` packages `glmnet` and `spikeslab`. The `R` package `glmnet` implements the Elastic net methodology [52], which is a generalization of both the LASSO [45] and the ridge regression (RR) [21]. The `R` package `spikeslab` in turn, implements the Spike and Slab methodology [23], which is a Bayesian approach for variable selection.

Dimension reduction can not however, be restricted to variable selection. Indeed, the field can be extended to include approaches which aim is to create surrogate covariates that summarizes the information carried in initial covariates. Since the emblematic Principal Component Regression (PCR)[25], many of the latter methods spread in the recent literature. As specific examples, we may refer to the OSCAR methodology [7], or the PACS methodology [41] which is a generalization of the latter approach. Those methods mainly proposed variables clustering within a regression model as a

way to reduce the dimensionality. Despite their theoretical and practical appeal, implementations of those methods were often proposed only through `Matlab` or `R` scripts, limiting thus the flexibility and the computational efficiency of their use. The CLusterwise Effect REgression (CLERE) methodology [49], was recently introduced as a novel methodology for simultaneous variables clustering and regression. The CLERE methodology is based on the assumption that each regression coefficient is an unobserved random variable sampled from a mixture of Gaussian distributions with an arbitrary number g of components. In addition, all components in the mixture are assumed to have different means $(b_1, \ldots, b_g)$ and equal variances equal to γ^2.

In this paper, we propose two new features for the CLERE model. First, the stochastic EM (SEM) algorithm is proposed as a more computationally efficient alternative to the Monte Carlo EM (MCEM) algorithm previously introduced in [49]. Secondly, the CLERE model is enhanced with the possibility of constraining the first component to have its mean equal to 0, i.e. $b_1 = 0$. This enhancement mainly aimed at facilitating the interpretation of the model. Indeed when b_1 is set to 0, variables assigned to the cluster associated with b_1 might be considered less relevant than other variables. Those two new features were implemented in a *C++* program available through the `R` package `clere`.

The outline of the present paper is the following. In Section 3.2, the definition of the model is recalled and the strategy to estimate the model parameter is presented. In Section 3.3 are described both the MCEM and SEM algorithms. This section also presents how the number of clusters is chosen and how the constraint on parameter b_1 can be interpreted. Section 3.4 presents the main functionalities of the `R` package `clere`. In Section 3.5, numerical experiments are presented aiming at illustrating the computational gain of the SEM algorithm over our former strategy and the good predictive performances of CLERE compared to standard dimension reduction methods. In that section, we also present two real data analysis performed with the `R` package `clere`,through two datasets. Finally, perspectives and further potential improvements of the package are discussed in Section 3.6.

3.2 Model definition and notation

Our model is defined by the following hierarchical relationships:

$$\begin{cases} y_i \sim \mathcal{N}\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \sigma^2\right) \\ \beta_j | \mathbf{z}_j \sim \mathcal{N}\left(\sum_{k=1}^{g} b_k z_{jk}, \gamma^2\right) \\ \mathbf{z}_j = (z_{j1}, \ldots, z_{jg}) \sim \mathcal{M}\left(\pi_1, \ldots, \pi_g\right). \end{cases} \tag{3.1}$$

For an individual $i = 1, \ldots, n$, y_i is the response and x_{ij} is an observed value for the j-th covariate. β_j is the regression coefficient associated with the j-th covariate ($j = 1, \ldots, p$). Let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$, $\boldsymbol{y} = (y_1, \ldots, y_n)'$, $\mathbf{X} = (x_{ij})$, $\boldsymbol{Z} = (z_{jk})$, $\boldsymbol{b} = (b_1 \ldots b_g)'$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)'$.
Moreover, $\log p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta})$ denotes the log-likelihood of model (3.1) assessed for the parameter $\boldsymbol{\theta} = (\beta_0, \boldsymbol{b}, \boldsymbol{\pi}, \sigma^2, \gamma^2)$. Model (3.1) can be interpretated as a Bayesian approach. However, to be fully Bayesian a prior distribution for parameter $\boldsymbol{\theta}$ would have been necessary. Instead, we proposed to estimate $\boldsymbol{\theta}$ by maximizing the (marginal) log-likelihood, $\log p(\mathbf{y}|\mathbf{X}; \boldsymbol{\theta})$. This partially Bayesian approach is referred to as *Empirical Bayes* (EB) [9]. Let $\mathcal{Z}$ be the set of $p \times g$-matrices partitioning p covariates into g groups. Those matrices are defined as

$$\boldsymbol{Z} = (z_{jk})_{1 \leq j \leq p, 1 \leq k \leq g} \in \mathcal{Z} \Leftrightarrow \forall j = 1, \ldots, p \begin{cases} \exists!\ k \text{ such as } z_{jk} = 1 \\ \text{if } k' \neq k \text{ then } z_{jk} = 0. \end{cases}$$

The log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta})$ is defined as

$$\log p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta}) = \log \left[\sum_{\boldsymbol{Z} \in \mathcal{Z}} \int_{\mathbb{R}^p} p(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\beta} \right].$$

Since it requires integrating over $\mathcal{Z}$ with cardinality g^p, evaluating the likelihood becomes rapidly computationnally unaffordable.
Nonetheless, maximum likelihood estimation is still achievable using the expectation maximization (EM) algorithm [13]. The latter algorithm is an iterative method which starts with an initial estimate of the parameter and updates this estimate until convergence. Each iteration of the algorithm consists of two steps, denoted as the E and the M steps. At each iteration d of the algorithm, the *E step* consists in calculating the

expectation of the log-likelihood of the complete data (observed + unobserved) with respect to $p(\boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}^{(d)})$, the conditional distribution of the unobserved data given the observed data, and the value of the parameter at the current iteration, $\boldsymbol{\theta}^{(d)}$. This expectation, often denoted as $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(d)})$ is then maximized with respect to $\boldsymbol{\theta}$ at the *M step*.

In model (3.1), the *E step* is analytically intractable. A broad literature devoted to intractable *E steps* recommends the use of a stochastic approximation of $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(d)})$ through Monte Carlo (MC) simulations [46], [26]. This approach is referred to as the MCEM algorithm. Besides, mean-field-type approximations are also proposed [17], [28]. Despite their computational appeal, the latter approximations do not generally ensure convergence to the maximum likelihood [18]. Alternatively, the SEM algorithm [10] was introduced as a stochastic version of the EM algorithm. In this algorithm, the *E step* is replaced with a simulation step (*S step*) that consists in generating a complete sample by simulating the unobserved data using $p(\boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}^{(d)})$. After the *S step* follows the *M step* which consists in maximizing $p(\boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Alternating those two steps generate a sequence $\left(\boldsymbol{\theta}^{(d)}\right)$, which is Markov chain whose stationary distribution (when it exists) concentrates around a local maximum of the likelihood.

3.3 Estimation and model selection

3.3.1 Initialization

The two algorithms presented in this section are initialized using a primary estimate ${\beta_j}^{(0)}$ of each β_j. The latter can be chosen either at random, or obtained from univariate regression coefficients or penalized approaches like LASSO and ridge regression. For large SEM or MCEM chains, initialization is not a critical issue. The choice of the initialization strategy is therefore made to speed up the convergence of the chains. A Gaussian mixture model with g component(s) is then fitted using $\boldsymbol{\beta}^{(0)} = \left(\beta_1^{(0)}, \ldots, \beta_p^{(0)}\right)$ as observed data to produce starting values $\mathbf{b}^{(0)}$, $\boldsymbol{\pi}^{(0)}$ and ${\gamma^2}^{(0)}$ respectively for parameters $\mathbf{b}$, $\boldsymbol{\pi}$ and γ^2. Using maximum a posteriori (MAP) clustering, an initial partition

$\mathbf{Z}^{(0)} = \left(z_{jk}^{(0)}\right) \in \mathcal{Z}$ is obtained as

$$\forall j \in \{1,\ldots,p\},\ z_{jk}^{(0)} = \begin{cases} 1 & \text{if } k = \text{argmin}_{k' \in \{1,\ldots,g\}} \left(\beta_j{}^{(0)} - b_{k'}^{(0)}\right)^2 \\ 0 & \text{otherwise.} \end{cases}$$

β_0 and σ^2 are initialized using $\boldsymbol{\beta}^{(0)}$ as follows:

$$\beta_0^{(0)} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j^{(0)}x_{ij}\right) \text{ and } \sigma^{2(0)} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \beta_0^{(0)} - \sum_{j=1}^{p}\beta_j^{(0)}x_{ij}\right)^2.$$

3.3.2 MCEM algorithm

The Stochastic Approximation of the E step

Suppose at iteration d of the algorithm that we have $\left\{\left(\boldsymbol{\beta}^{(1,d)}, \boldsymbol{Z}^{(1,d)}\right), \ldots, \left(\boldsymbol{\beta}^{(M,d)}, \boldsymbol{Z}^{(M,d)}\right)\right\}$, M samples from $p\left(\boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}^{(d)}\right)$. Then the MC approximation of the E-step can be written

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(d)}\right) = \mathbb{E}\left[\log p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(d)})|\mathbf{y}, \mathbf{X}; \boldsymbol{\theta}^{(d)}\right] \approx \frac{1}{M}\sum_{m=1}^{M}\log p(\mathbf{y}, \boldsymbol{\beta}^{(m,d)}, \mathbf{Z}^{(m,d)}|\mathbf{X}; \boldsymbol{\theta}^{(d)}).$$

However, sampling from $p\left(\boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}^{(d)}\right)$ is not straightforward. However, we can use a Gibbs sampling scheme to simulate unobserved data, taking advantage of $p\left(\boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}^{(d)}\right)$ and $p\left(\boldsymbol{Z}|\boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}^{(d)}\right)$ from which it is easy to simulate. Those distributions, respectively Gaussian and multinomial, are described below in Equations (3.2) and (3.3).

$$\begin{cases} \boldsymbol{\beta}|\boldsymbol{Z}, \boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}^{(d)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(d)}, \boldsymbol{\Sigma}^{(d)}\right) \\ \boldsymbol{\mu}^{(d)} = \left[\boldsymbol{X'X} + \frac{\sigma^{2(d)}}{\gamma^{2(d)}}\mathrm{I}_p\right]^{-1}\boldsymbol{X'}\left(\mathbf{y} - \beta_0^{(d)}\mathbf{1}_p\right) + \frac{\sigma^{2(d)}}{\gamma^{2(d)}}\left[\boldsymbol{X'X} + \frac{\sigma^{2(d)}}{\gamma^{2(d)}}\mathrm{I}_p\right]^{-1}\boldsymbol{Zb}^{(d)} \\ \boldsymbol{\Sigma}^{(d)} = \sigma^{2(d)}\left[\boldsymbol{X'X} + \frac{\sigma^{2(d)}}{\gamma^{2(d)}}\mathrm{I}_p\right]^{-1} \end{cases} \quad (3.2)$$

and (note that $p\left(\boldsymbol{Z}|\boldsymbol{\beta},\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}^{(d)}\right)$ does not depend on $\boldsymbol{X}$ nor $\boldsymbol{y}$)

$$p\left(z_{jk}=1|\boldsymbol{\beta};\boldsymbol{\theta}^{(d)}\right)\propto\pi_k^{(d)}\exp\left(-\frac{\left(\beta_j-b_k^{(d)}\right)^2}{2\gamma^{2(d)}}\right). \tag{3.3}$$

In Equation (3.2), I_p and $\mathbf{1}_p$ respectively stands for the identity matrix in dimension p and the vector of $\mathbb{R}^p$ which all coordinates equal 1. To efficiently sample from $p\left(\boldsymbol{\beta}|\boldsymbol{Z},\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}^{(d)}\right)$ a preliminary singular vector decomposition of matrix $\boldsymbol{X}$ is necesary. Once this decomposition is performed the overall complexity of the approximated *E step* is $\mathcal{O}\left[M(p^2+pg)\right]$.

The M step

Using the M draws obtained by Gibbs sampling at iteration d, the *M step* is straightforward as detailed in Equations (3.4) to (3.8). The overall computational complexity of that step is $\mathcal{O}\left(Mpg\right)$.

$$\pi_k^{(d+1)}=\frac{1}{Mp}\sum_{m=1}^{M}\sum_{j=1}^{p}z_{jk}^{(m,d)}, \tag{3.4}$$

$$b_k^{(d+1)}=\frac{1}{Mp\pi_k^{(d+1)}}\sum_{m=1}^{M}\sum_{j=1}^{p}z_{jk}^{(m,d)}\beta_j^{(m,d)}, \tag{3.5}$$

$$\gamma^{2(d+1)}=\frac{1}{Mp}\sum_{m=1}^{M}\sum_{j=1}^{p}\sum_{k=1}^{g}z_{jk}^{(m,d)}\left(\beta_j^{(m,d)}-b_k^{(d+1)}\right)^2, \tag{3.6}$$

$$\beta_0^{(d+1)}=\frac{1}{n}\sum_{i=1}^{n}\left[y_i-\sum_{j=1}^{p}\left(\frac{1}{M}\sum_{m=1}^{M}\beta_j^{(m,d)}\right)x_{ij}\right], \tag{3.7}$$

$$\sigma^{2(d+1)}=\frac{1}{nM}\sum_{m=1}^{M}\sum_{i=1}^{n}\left(y_i-\beta_0^{(d+1)}-\sum_{j=1}^{p}\beta_j^{(m,d)}x_{ij}\right)^2. \tag{3.8}$$

3.3.3 SEM algorithm

In most situations, the SEM algorithm can be considered as a special case of the MCEM algorithm [10], obtained by setting $M = 1$. In model (3.1), such a direct derivation leads to an algorithm which computational complexity is quadratic with respect to p. To reduce that complexity, we propose a SEM algorithm based on the integrated complete data likelihood $p(\boldsymbol{y}, \boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\theta})$ rather than $p(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\theta})$. A closed form of $p(\boldsymbol{y}, \boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\theta})$ is available and given subsequently.

Closed form of the integrated complete data likelihood

Let the SVD decomposition of matrix $\boldsymbol{X}$ be $\boldsymbol{USV'}$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are respectively $n \times n$ and $p \times p$ orthogonal matrices, and $\boldsymbol{S}$ is $n \times p$ rectangular diagonal matrix which diagonal terms are the eigenvalues $\left(\lambda_1^2, \ldots, \lambda_n^2\right)$ of matrix $\boldsymbol{XX'}$. We now define $\boldsymbol{X}^u = \boldsymbol{U'X}$ and $\boldsymbol{y}^u = \boldsymbol{U'y}$. Let $\mathbf{1}_n$ be the vector of $\mathbb{R}^n$ which each coordinate equals 1. Matrix $\boldsymbol{M}$ is defined as the $n \times (g+1)$ matrix which first column equals $\boldsymbol{s}^u = \boldsymbol{U}'\mathbf{1}_n = \left(s_1^u, \ldots, s_n^u\right)'$ and which additional columns are those of matrix $\boldsymbol{X}^u\boldsymbol{Z}$. Let also $\boldsymbol{t} = (\beta_0, \boldsymbol{b}) \in \mathbb{R}^{(g+1)}$ and $\boldsymbol{R}$ be a $n \times n$ diagonal matrix which i-th diagonal term equal $\sigma^2 + \gamma^2\lambda_i^2$. With these notations we can express the complete data likelihood integrated over $\boldsymbol{\beta}$ as

$$\begin{aligned}\log p\left(\boldsymbol{y}, \boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\theta}\right) = &-\frac{n}{2}\log\left(2\pi\right) - \frac{1}{2}\sum_{i=1}^{n}\log\left(\sigma^2 + \gamma^2\lambda_i^2\right) \\ &- \frac{1}{2}\left(\boldsymbol{y}^u - \boldsymbol{Mt}\right)'\boldsymbol{R}^{-1}\left(\boldsymbol{y}^u - \boldsymbol{Mt}\right) \\ &+ \sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}\log\pi_k. \end{aligned} \tag{3.9}$$

Simulation step

To sample from $p\left(\boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}\right)$ we use a Gibbs sampling strategy based on the conditional distributions $p\left(\boldsymbol{z}_j|\boldsymbol{y}, \boldsymbol{Z}^{-j}, \boldsymbol{X}; \boldsymbol{\theta}\right)$, $\boldsymbol{Z}^{-j}$ denoting the set of cluster membership indicators for all covariates but the j-th. Let $\boldsymbol{w}^{-j} = \left(w_1^{-j}, \ldots, w_n^{-j}\right)'$, where $w_i^{-j} = y_i^u - \beta_0 s_i^u - \sum_{l\neq j}\sum_{k=1}^{g} z_{lk}x_{il}^u b_k$. The conditional distribution $p(z_{jk} = 1|\boldsymbol{Z}^{-j}, \boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta})$ can

be written

$$p(z_{jk}=1|\boldsymbol{Z}^{-j},\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}) \propto \pi_k \exp\left[-\frac{b_k^2}{2}\left(\boldsymbol{x}_j^u\right)'\boldsymbol{R}^{-1}\boldsymbol{x}_j^u + b_k\left(\boldsymbol{w}^{-j}\right)'\boldsymbol{R}^{-1}\boldsymbol{x}_j^u\right], \quad (3.10)$$

where $\boldsymbol{x}_j^u$ is the j-th column of $\boldsymbol{X}^u$. In the classical SEM algorithm, convergence to $p\left(\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}\right)$ should be reached before updating $\boldsymbol{\theta}$. However, a valid inference can still be ensured in settings when $\boldsymbol{\theta}$ is updated only after one or few Gibbs iterations. These approaches are referred to as SEM-Gibbs algorithm [6]. The overall computational complexity of the simulation step is $\mathcal{O}\left(npg\right)$, so linear with p and no quadratic as obtained previously with MCEM.

To improve the mixing of the generated Markov chain, we start the simulation step at each iteration by creating a random permutation of $\{1,\ldots,p\}$. Then, according to the order defined by that permutation, we update each z_{jk} using $p(z_{jk}=1|\boldsymbol{Z}^{-j},\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta})$.

Maximization step

$\log p\left(\boldsymbol{y},\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\theta}\right)$ corresponds to the marginal log-likelihood of a linear mixed model [39] which can be written

$$\boldsymbol{y}^u = \boldsymbol{M}\boldsymbol{t} + \boldsymbol{\lambda}\boldsymbol{v} + \boldsymbol{\varepsilon} \quad (3.11)$$

where $\boldsymbol{v}$ is an unobserved random vector such as $\boldsymbol{v} \sim \mathcal{N}\left(0,\gamma^2\mathrm{I}_n\right)$, $\boldsymbol{\varepsilon} \sim \mathcal{N}\left(0,\sigma^2\mathrm{I}_n\right)$ and $\boldsymbol{\lambda} = \mathrm{diag}\left(\lambda_1,\ldots,\lambda_n\right)$. The estimation of the parameters of model (3.11) can be performed using the EM algorithm, as in [39]. We adapt below the EM equations defined in [39], using our notations. At iteration s of the internal EM algorithm, we define $\boldsymbol{R}^{(s)} = \sigma^{2^{(s)}}\boldsymbol{I}_n + \gamma^{2^{(s)}}\boldsymbol{\lambda}'\boldsymbol{\lambda}$. The detailed *internal E* and *M steps* are given below:
Internal E step:

$$
\begin{aligned}
v_{\sigma}^{(s)} &= \mathbb{E}\left[\left(\boldsymbol{y}^u - \boldsymbol{M}\boldsymbol{t}^{(s)} - \boldsymbol{\lambda}\boldsymbol{v}\right)'\left(\boldsymbol{y}^u - \boldsymbol{M}\boldsymbol{t}^{(s)} - \boldsymbol{\lambda}\boldsymbol{v}\right)|\boldsymbol{y}^u\right] \\
&= \sigma^{4(s)}\left(\boldsymbol{y}^u - \boldsymbol{M}\boldsymbol{t}^{(s)}\right)'\boldsymbol{R}^{(s)}\boldsymbol{R}^{(s)}\left(\boldsymbol{y}^u - \boldsymbol{M}\boldsymbol{t}^{(s)}\right) + n \times \sigma^{2(s)} - \sigma^{4(s)}\sum_{i=1}^{n}\frac{1}{\sigma^{2(s)} + \gamma^{2(s)}\lambda_i^2}. \\
v_{\gamma}^{(s)} &= \mathbb{E}\left[\boldsymbol{v}'\boldsymbol{v}|\boldsymbol{y}^u\right] \\
&= \gamma^{4(s)}\left(\boldsymbol{y}^u - \boldsymbol{M}\boldsymbol{t}^{(s)}\right)'\boldsymbol{R}^{(s)}\boldsymbol{\lambda}'\boldsymbol{\lambda}\boldsymbol{R}^{(s)}\left(\boldsymbol{y}^u - \boldsymbol{M}\boldsymbol{t}^{(s)}\right) + n \times \gamma^{2(s)} - \gamma^{4(s)}\sum_{i=1}^{n}\frac{\lambda_i^2}{\sigma^{2(s)} + \gamma^{2(s)}\lambda_i^2}. \\
\boldsymbol{h}^{(s)} &= \mathbb{E}\left[\boldsymbol{y}^u - \boldsymbol{\lambda}\boldsymbol{v}|\boldsymbol{y}^u\right] = \boldsymbol{M}\boldsymbol{t}^{(s)} + \sigma^{2(s)}R^{-1}(s)\left(\boldsymbol{y}^u - \boldsymbol{M}\boldsymbol{t}^{(s)}\right).
\end{aligned}
$$

Internal M step:

$$
\begin{aligned}
&\sigma^{2(s+1)} = v_{\sigma}^{(s)}/n. \\
&\gamma^{2(s+1)} = v_{\gamma}^{(s)}/n. \\
&\boldsymbol{t}^{(s+1)} = \left[\boldsymbol{M}'\boldsymbol{M}\right]^{-1}\boldsymbol{M}'\boldsymbol{h}^{(s)}.
\end{aligned}
$$

Given a non-negative user-specified threshold δ and a maximum number N_{max} of iterations, *Internal E* and *M steps* are alternated until

$$|\log p\left(\boldsymbol{y}, \boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\theta}^{(s)}\right) - \log p\left(\boldsymbol{y}, \boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\theta}^{(s+1)}\right)| < \delta \text{ or } s = N_{max}.$$

The computational complexity of the *M step* is $\mathcal{O}\left(g^3 + ngN_{max}\right)$, thus not involving p.

Attracting and absorbing states

- *Absorbing states*. The SEM algorithm described above defines a Markov chain which stationary distribution is concentrated around values of $\boldsymbol{\theta}$ corresponding to local maxima of the likelihood function. This chain has absorbing states in values of $\boldsymbol{\theta}$ such as $\sigma^2 = 0$ or $\gamma^2 = 0$. In fact, the *internal M step* reveals that updated values for σ^2 and γ^2 are proportional to previous values of those parameters.
- *Attracting states*. We empirically observed that attraction around $\sigma^2 = 0$ was quite frequent when matrix $\boldsymbol{X}$ is centered and $p > n$. To avoid this attraction, we

advocate users of the package not to center the columns when p, the number of variables, is smaller than n, the sample size. A similar behavior was also observed with the MCEM algorithm when $p > n$ and $M < 5$.

3.3.4 Model selection

Once the MLE $\widehat{\boldsymbol{\theta}}$ is calculated (using one or the other algorithm), the maximum log-likelihood and the posterior clustering matrix $\mathbb{E}\left[\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{X};\widehat{\boldsymbol{\theta}}\right]$ are approximated using MC simulations based on Equations (3.9) and (3.10). The approximated maximum log-likelihood $\widehat{l}$, is then utilized to calculate AIC [1] and BIC [38] criteria for model selection. In model (2.3), those criteria can be written as

$$\text{BIC} = -2\widehat{l} + 2(g+1)\log(n) \text{ and } \text{AIC} = -2\widehat{l} + 4(g+1). \tag{3.12}$$

An additional criterion for model selection, namely the ICL criterion [5] is also implemented in the `R` package `clere`. The latter criterion can be written

$$\text{ICL} = -2\widehat{l} + 2(g+1)\log(n) - \sum_{j=1}^{p}\sum_{k=1}^{g}\pi_{jk}\log(\pi_{jk}), \tag{3.13}$$

where $\pi_{jk} = \mathbb{E}\left[z_{jk}|\boldsymbol{y},\boldsymbol{X};\widehat{\boldsymbol{\theta}}\right]$.

3.3.5 Interpretation of the special group of variables associated with $b_1 = 0$

The constraint $b_1 = 0$ is mainly driven by an interpretation purpose. The meaning of this group depends on both the total number g of groups and the estimated value of parameter γ^2. In fact, when $g > 1$ and γ^2 is small, covariates assigned to that group are likely less relevant to explain the response. Determining whether γ^2 is small enough is not straightforward. However, when this property holds, we may expect the groups of covariates to be separated. This would for example translate in the posterior probabilities π_{j1} being larger than 0.7. In addition to the benefit in interpretation, the constraint $b_1 = 0$, reduces the number of parameters to be estimated and consequently the variance of the predictions performed using the model.

3.4 Package functionalities

The R package clere mainly implements a function for parameter estimation and model selection: the function fit.clere(). Four additional functions for graphical representation plot(), summarizing the results summary(), getting the predicted clusters of variables clusters() and making predictions from new design matrices predict() are also implemented in the package.

3.4.1 The main function fit.clere()

The call of the R function fit.clere() is:

```
R> mod <- fit.clere(y, x, g, analysis = "fit", algorithm = "SEM",
+ nItMC = 1, nItEM = 1000, nBurn = 200, dp = 10, nsamp = 2000,
+ maxit = 1000, tol = 1e-6, plotit = FALSE, sparse = FALSE,
+ theta0 = NULL, Z0 = NULL)}
```

Each of the input function arguments is described in Table 3.1. The fit.clere() function returns an R object of class clere. In addition to all input parameters, this object has the other slots detailed in Table 3.2.

Argument name	Description
y	is a vector of response of size n
x	is a $n \times p$ matrix.
g	is either the exact number groups (when analysis = "fit") or the maximum number of groups to be tested (when analysis = "aic", analysis = "bic" or analysis = "icl").
analysis	takes value in {"fit","aic","bic","icl"}. When analysis = "fit", the model is fitted assuming that the exact number of groups is g.

When instead, `analysis = "aic"`, `analysis = "bic"` or `analysis = "icl"`, the model is fitted g times with a number of group(s) between 1 and g. The g models are then compared respectively using AIC, BIC and ICL criterion and the best model is returned.

`algorithm` allows to select between SEM (`algorithm = "SEM"`) and MCEM (`algorithm = "MCEM"`) algorithms to estimate the model parameters.

`nItMC` is the number of complete Gibbs sampling (described in Equation (3.10)) iterations run before simulating a partition $\boldsymbol{Z}$ when running SEM algorithm.

`nItEM` is the number of SEM or MCEM iterations.

`nBurn` is the number of iterations *burn-in* discarded to calculate the MLE in the SEM algorithm. This number is also used in the MCEM algorithm as a number of discarded iterations for the Gibbs sampling required to draw from $p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})$.

`dp` is the length of thinning interval used to break dependence between the sampled partitions. It corresponds to the number of Gibbs iterations to skip between consecutive draws of the chain.

`nsamp` is the number of Gibbs iterations used to approximate the log-likelihood.

`maxit` is the maximum number of *internal EM* algorithm iterations runned at the *M step* of the SEM algorithm.

`tol` is the tolerance parameter required to end the *internal EM* algorithm runned at the *M step* of the SEM algorithm.

`plotit`	equals `FALSE` or `TRUE`. This boolean parameter indicates whether a summary graph should be plotted.
`sparse`	equals `FALSE` or `TRUE`. It indicates whether parameter b_1 is set to 0.
`theta0`	is a user-specified initial guess of the model parameter $\boldsymbol{\theta}$.
`Z0`	is a user-specified initial partition of the variables given as a vector of size p of integers between 1 and g

Table 3.1: Input arguments of the function fit.clere().

3.4.2 Secondary functions summary(), plot(), ggPlot(), clusters() and predict()

Examples of calls for the functions presented in this section are given in Section 3.5.3.
The `summary()` function prints an overview of the estimated parameters and returns the estimated likelihood and information based model selection criteria (AIC, BIC and ICL).
The call of functions `plot()` and `ggPlot()` are similar to the one of function `{summary()`. The latter function function produces graphs such as ones presented in Figure 3.1. The function `ggPlot()` requires a prior installation of the `R` package `ggplot2`. However, there is no dependancies with the latter package since the `R` package `clere` can be installed without `ggplot2`. When `ggplot2` is not installed, the user can still make use of the function `plot()`.
The function `clusters()`, takes one argument of class `clere` and a `threshold` argument. This function assigns each variable to the group which associated posterior probability of membership is larger than the given `threshold`. When `threshold = NULL`, the maximum a posteriori (MAP) strategy is used to infer the clusters.
The `predict()` function has two arguments, being a `clere` and a design matrix $\boldsymbol{X}_{new}$. Using that new design matrix, the `predict()` function returns an approximation of

Argument name	Description
`intercept`	is the estimated value for parameter β_0.
`b`	is the estimated value for parameter $\boldsymbol{b}$. It is a numeric vector of size g.
`pi`	is the estimated value for parameter $\boldsymbol{\pi}$. It is a numeric vector of size g.
`sigma2`	is the estimated value for parameter σ^2.
`gamma2`	is the estimated value for parameter γ^2.
`theta`	is a `nItEM` $\times$ (2g +4) matrix containing values of the model parameters and complete data log-likelihood at each iteration of the SEM/MCEM algorithm.
`likelihood`	is an approximation of the log-likelihood using `nsamp` MC simulations.
`entropy`	is an approximation of the entropy using `nsamp`+ MC simulations.
`P`	is a $p \times g$ matrix approximating $\mathbb{E}\left[\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}\right]$ using `nsamp` MC simulations.
`Bw`	is a $p \times$ `nsamp` matrix which columns are samples from the distribution $p(\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta})$.
`Zw`	is a $p \times$ `nsamp` matrix which columns are samples from the distribution $p(\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta})$. Each column is sampled partition coded a vector of size p containing integers taking values between 0 and $(g-1)$.

Table 3.2: The function fit.clere() function returns an R object of class clere. In addition to all input parameters, this object has the other slots described in this table.

$\mathbb{E}\left[\boldsymbol{X}_{new}\boldsymbol{\beta}|\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}\right]$.

3.5 Numerical experiments

This section presents two sets of numerical experiments. The first set of experiments aims at comparing the MCEM and SEM algorithms in terms of computational time

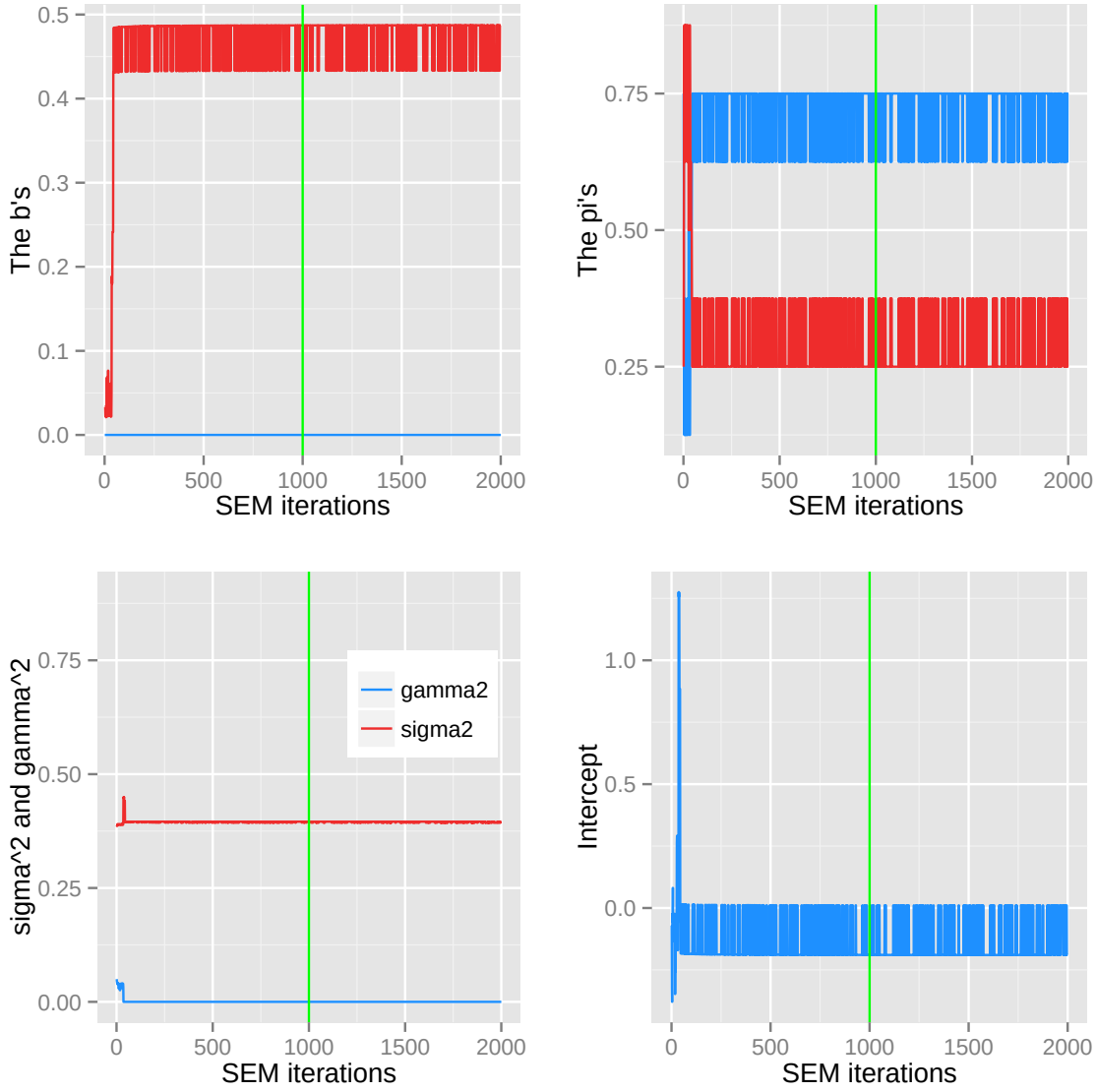

Figure 3.1: Values of the model parameters in view of SEM algorithm iterations. The vertical green line in each of the four plots, represents the number nBurn of iterations discarded before calculating maximum likelihood estimates.

and estimation or prediction accuracy. The second set of experiments aimed at comparing CLERE to standard dimension reduction techniques. The latter comparison is performed on both simulated and real data.

3.5.1 SEM algorithm versus MCEM algorithm

Description of the simulation study

In this section, a comparison between the SEM algorithm and the MCEM algorithm is performed. This comparison is performed using the four following performance indicators:

1. Computational time (CT) to run a pre-defined number of SEM/MCEM iterations. This number was set to 2,000 in this simulation study.

2. Mean squared estimation error (MSEE) defined as

$$MSEE_a = \mathbb{E}\left[(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_a)'(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_a)\right],$$

   where $a \in \{$"SEM","MCEM"$\}$ and $\widehat{\theta}_a$ is an estimated value for parameter $\boldsymbol{\theta}$ obtained with algorithm a. Since $\boldsymbol{\theta}$ is only known up to a permutation of the group labels, we chose the permutation leading to the smallest MSEE value.

3. Mean squared prediction error (MSPE) defined as

$$MSPE_a = \mathbb{E}\left[(\boldsymbol{y}^v - \boldsymbol{X}^v\widehat{\boldsymbol{\theta}}_a)'(\boldsymbol{y}^v - \boldsymbol{X}^v\widehat{\boldsymbol{\theta}}_a)\right],$$

   where $\boldsymbol{y}^v$ and $\boldsymbol{X}^v$ are respectively a vector of responses and a design matrix from a validation dataset.

4. Maximum log-likelihood (ML) reached. This quantity was approximated using 1,000 samples from $p(\boldsymbol{Z}|\boldsymbol{y};\widehat{\boldsymbol{\theta}})$.

Three versions of the MCEM algorithm were proposed for comparison with the SEM algorithm, depending on the number M (or `nsamp`) of Gibbs iterations used to approximate the *E step*. That number was varied between 5, 25 and 125. Those versions were respectively denoted MCEM_5, MCEM_{25} and MCEM_{125}. The comparison was performed using 200 simulated datasets. Each training dataset consisted of $n = 25$ individuals and $p = 50$ variables. Validation datasets used to calculate MSPE consisted of 1,000 individuals each. All covariates were simulated independently according to

the standard Gaussian distribution:

$$\forall (i,j)\ x_{ij} \sim \mathcal{N}(0,1).$$

Both training and validation datasets were simulated according to model (3.1) using $\beta_0 = 0$, $\boldsymbol{b} = (0,3,15)'$, $\boldsymbol{\pi} = (0.64, 0.20, 0.16)'$, $\sigma^2 = 1$ and $\gamma^2 = 0$. This is equivalent to simulate data according to the standard linear regression model defined by:

$$y_i \sim \mathcal{N}\left(\sum_{j=1}^{32} 0 \times x_{ij} + \sum_{j=33}^{42} 3 \times x_{ij} + \sum_{j=43}^{50} 15 \times x_{ij}, 1\right)$$

All algorithms were run using 10 different random starting points. Estimates yielding the largest likelihood were then used for the comparison.

Results of the comparison

Table 3.3 summarizes the results of the comparison between the algorithms. The SEM algorithm ran faster than its competitors in 74.5% of the simulations. The gain in computational time yielded by SEM was between 1.3-fold (when compared to MCEM_5) and 22.2-fold (when compared to MCEM_{125}). This improvement was accompanied with a good accuracy in parameter estimation (second best median MSEE: 0.258; smallest MSEE in 25.5% of the simulations) and a smaller prediction error (smallest median MSPE: 1.237; smallest MSPE in 48.5% of the simulations). Those good performances were mainly explained by the fact that the SEM algorithm most of the time reached a better likelihood than the other algorithms.

3.5.2 Comparison with other methods

Description of the methods

In this section, we compare our model to standard dimension reduction approaches in terms of MSPE. Although a more detailed comparison was proposed in [49], we propose here a quick illustration of the relative predictive performance of our model. The comparison is achieved using data simulated according to the scenario described

Performance indicators	Algorithms	% of times the algorithm was best	Median (Std. Err.)
CT (seconds)	**SEM**	**74.50**	**1.60 (0.23)**
	$MCEM_5$	25.50	2.04 (0.13)
	$MCEM_{25}$	0	7.63 (0.46)
	$MCEM_{125}$	0	35.6 (2.22)
MSEE	**SEM**	25.5	0.258 (0.19)
	$MCEM_5$	**33.0**	1.019 (0.97)
	$MCEM_{25}$	22.5	**0.257 (0.21)**
	$MCEM_{125}$	19.0	0.295 (0.25)
MSPE	**SEM**	**48.5**	**1.237 (0.16)**
	$MCEM_5$	20.5	1.523 (0.49)
	$MCEM_{25}$	19.0	1.258 (0.19)
	$MCEM_{125}$	12.0	1.272 (0.21)
	True parameter	—	1.159 (0.08)
ML	**SEM**	**59.5**	**-78.60 (3.60)**
	$MCEM_5$	10.5	-79.98 (5.78)
	$MCEM_{25}$	18.0	-79.00 (3.84)
	$MCEM_{125}$	12.0	-79.47 (4.20)
	True parameter	—	-77.60 (2.37)

Table 3.3: Performance indicators used to compare SEM and MCEM algorithms. Computational Time (CT) was measured on a Intel(R) Xeon(R) CPU E7- 4870 @ 2.40GHz processor. The best algorithm is defined as the one that either reached the largest log-likelihood (ML) or the lowest CT, Mean Squared Prediction Error (MSPE) and Mean Squared Estimation Error (MSEE). The best algorithm for each criterion is highlighted in bold font.

above in Section 3.5.1. The methods selected for comparison are the ridge regression [21], the elastic net [52], the LASSO [45], PACS [41], the method of Park and colleagues [32] (subsequently denoted AVG) and the spike and slab model [23] (subsequently denoted SS). The first three methods are implemented in freely available `R` package `glmnet` (for ridge, LASSO and elastic net). Those packages were used with default options regarding the choice of tuning parameters.

PACS methodology proposes to estimate the regression coefficients by solving a penal-

ized least squares problem. It imposes a constraint on β that is a weighted combination of the L^1 norm and the pairwise L^∞ norm. Upper-bounding the pairwise L^∞ norm enforces the covariates to have close coefficients. When the constraint is strong enough, closeness translates into equality achieving thus a grouping property. For PACS, no software was available. Only an `R` script was released on Bondell's webpage[1]. Since this `R` script was running very slowly, we decided to reimplement it in C++ and observed a 30-fold speed-up of computational time. Similarly to Bondell's `R` script, our implementation uses two parameters `lambda` and `betawt`. In [41], the authors suggest assigning `betawt` with the coefficients obtained from a ridge regression model after the tuning parameter was selected using AIC. In this simulation study we used the same strategy; however the ridge parameter was selected via 5-fold cross validation. 5-fold CV was preferred to AIC since selecting the ridge parameter using AIC always led to estimated coefficients equal to zero. Once `betawt` was selected, `lambda` was chosen via 5-fold cross validation among the following values: 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200 and 500. All other default parameters of their script were unchanged.

The AVG method is a two-step approach. The first step uses hierarchical clustering of covariates to create surrogate covariates by averaging the variables within each group. Those new predictors are afterwards included in a linear regression model, replacing the primary variables. A variable selection algorithm is then applied to select the most predictive groups of covariates. To implement this method, we followed the algorithm described in [32] and programmed it in R.

The spike and slab model is a Bayesian approach for variable selection. It is based on the assumption that the regression coefficients are distributed according to a mixture of two centered Gaussian distributions with different variances. One component of the mixture (the spike) is chosen to have a small variance, while the other component (the slab) is allowed to have a large variance. Variables assigned to the spike are dropped from the model. We used the R package `spikeslab` to run the spike and slab models. Especially, we used the function `spikeslab` from that package to detect influential variables. The number of iterations used to run the function `spikeslab` was 2,000 (1,000 discarded).

[1] *http : //www4.stat.ncsu.edu/ bondell/Software/PACS/PACS.R.r*

When running `fit.clere()`, the number `nItEM` of SEM iterations was set to 2,000. The number `g` of groups for CLERE was chosen between 1 and 5 using AIC (option `analysis="aic"`). Two versions of CLERE were considered: the one with all parameters estimated and the one with b_1 set to 0. The latter approach is subsequently denoted CLERE_0 (option `sparse=TRUE`).

Results of the comparison

Figure 3.2, summarizes the comparison between the methods. In this simulated scenario, CLERE outperformed the other methods in terms of prediction error. Those good performances were improved when parameter b_1 was set to 0. CLERE was also the most parcimonous approach with an averaged number of estimated parameters equal to 8.5 (6.7 when $b_1 = 0$). The second best approach was PACS which also led to parcimonous models. Variables selection approaches as whole yielded the largest prediction error in this simulation.

3.5.3 Real datasets analysis

Description of the datasets

We used in this section the real datasets `Prostate` and `eyedata` from the `R` packages `lasso2` and `flare` respectively.
The `Prostate` dataset comes from a study that examined the correlation between the level of prostate specific antigen and a number of clinical measures in $n = 97$ men who were about to receive a radical prostatectomy. This dataset was used in multiple publications including [45]. We used the prostate specific antigen (variable `lpsa`) as reponse variable and the $p = 8$ other measurements as covariates.
The `eyedata` dataset is extracted from the published study of [37]. This dataset consists in gene expression levels measured at $p = 200$ probes in $n = 120$ rats. The response variable utilized was the expression of the TRIM32 gene which is a biomarker of the Bardet-Bidel Syndrome (BBS).
Those two datasets was utilized to compare CLERE to the methods described in Section 3.5.2. All the methods were compared in terms of out-of-sample prediction error estimated using cross-validation (CV). 100 CV statistics were calculated by randomly

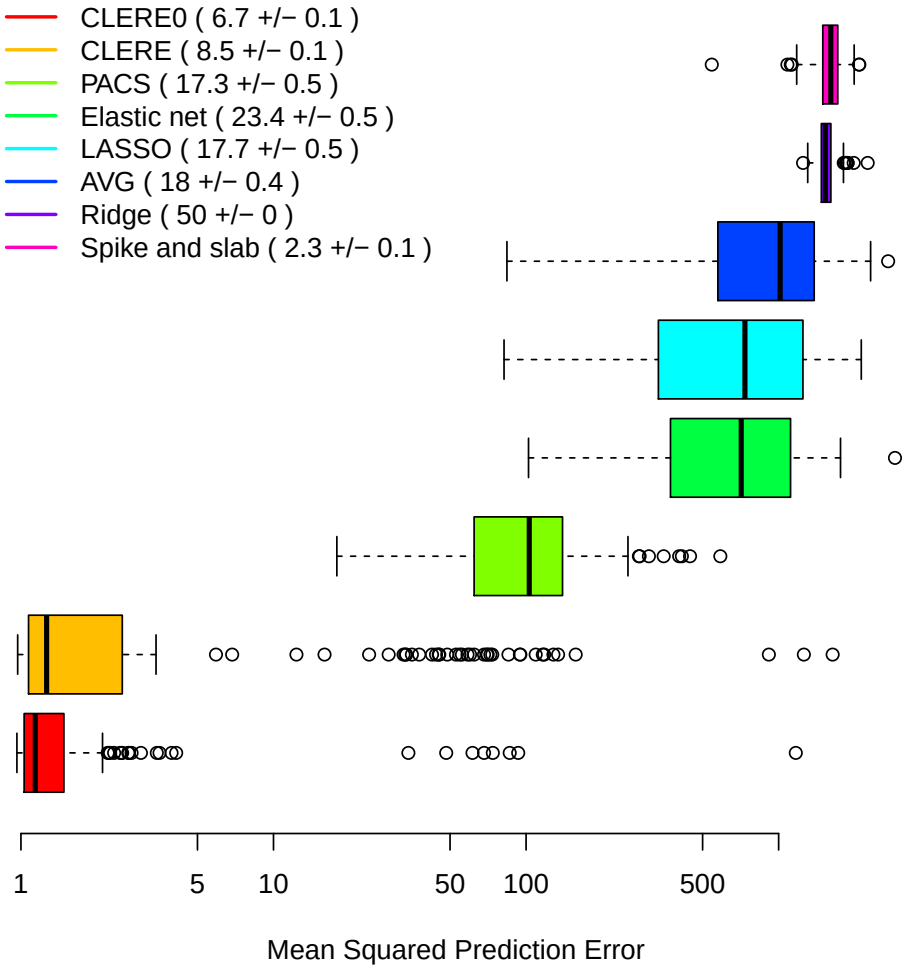

Figure 3.2: Comparison between CLERE and some standard dimension reduction approaches. The number of estimated parameters (+/- standard error) is given with the name of the method to be compared.

splitting each dataset into training (80% of the sample size) and validation (20% of the sample size) sets. Those CV statistics were then averaged and compared across the methods in Table 3.4.

Running the analysis

Before presenting the results of the comparison between CLERE and its competitors, we illustrate the command lines to run the analysis of the `Prostate` dataset. The dataset is loaded by typing:

```
R> library(lasso2)
R> data(Prostate)
R> y <- Prostate[,"lpsa"]
R> x <- as.matrix(Prostate[,-which(colnames(Prostate)=="lpsa")])
```

Possible training (`xt` and `yt`) and validation (`xv` and `yv`) sets are generated as following:

```
R> itraining <- 1:(0.8*nrow(x))
R> xt <- x[ itraining,] ; yt <- y[ itraining]
R> xv <- x[-itraining,] ; yv <- y[-itraining]
```

The `fit.clere()` function is run using AIC criterion to select the number of groups between 1 and 5. To lessen the impact of local minima in the model selection, 20 random starting points are used. This can be implemented as written below

```
R> Mod <- sapply(1:20,function(startingPoint){
+  mod <- fit.clere(y=yt,x=xt,g=5,analysis="aic",
+                   sparse=TRUE,nItEM=2000,nBurn=1000,
+                   nItMC=10,dp=5,nsamp=1000)})
R> AIC   <- sapply(Mod,function(m) summary(m)@AIC)
R> mod   <- Mod[[which.min( AIC )]]
R> summary(mod)
   ------------------------------
   | CLERE | Yengo et al. (2013) |
   ------------------------------

   Model object fitted with g=2 groups of variables.
   The number of groups was selected using AIC criterion among 1 to 5.
```

```
---
 Estimated parameters using SEM algorithm are
 intercept = -0.1395
 b         = 0.0000   0.4737
 pi        = 0.7188   0.2812
 sigma2    = 0.3951
 gamma2    = 4.181e-08

---
 Log-likelihood =  -78.28
 Entropy        =  0.5152
 AIC            =  168.57
 BIC            =  182.63
 ICL            =  183.15
```

```
R> plot(mod)
```

Running the command `ggPlot(mod)` generates the plot given in Figure 3.1. We can also access the cluster membership by running the command `clusters()`. For example, running the command `clusters(mod,threshold=0.7)` yields

```
R> clusters(mod,thresold=0.7)
 lcavol lweight     age    lbph     svi     lcp gleason   pgg45
      2       2       1       1       1       1       1       1
```

In the example above 2 variables, being the cancer volume (`lcavol`) and the prostate weight (`lweight`), were assigned to group 2 ($b_2 = 0.4737$). The other 6 variables were assigned to group 1 ($b_1 = 0$). Posterior probabilities of membership are available through the slot `{P} in object of class \verbclere+`.

```
R> mod@P
        Group 1 Group 2
lcavol    0.000   1.000
lweight   0.000   1.000
```

```
age       1.000   0.000
lbph      1.000   0.000
svi       0.789   0.211
lcp       1.000   0.000
gleason   1.000   0.000
pgg45     1.000   0.000
```

The covariates were respectively assigned to their group with a probability larger than 0.7. Moreover, given that parameter γ^2 had very small value ($\widehat{\gamma^2} = 4.181 \times 10^{-8}$), we can argue that cancer volume and prostate weight are the only relevant explanatory covariates. To assess the prediction error associated with the model we can run the command `predict()` as following:

```
R> error <- mean( (yv - predict(mod,xv))^2 )
R> error
[1] 1.550407
```

Results of the analysis

Table 3.4 summarizes the prediction errors and the number of parameters obtained for all the methods. CLERE_0 had the lowest prediction error in the analysis of the `Prostate` dataset and the second best performance with the `eyedata` dataset. The AVG method was also very competitive compared to variable selection approaches stressing thus the relevance of creating groups of variables to reduce the dimensionality. It is worth noting that in both datasets, imposing the constraint $b_1 = 0$ improved the predictive performance of CLERE.

In the `Prostate` dataset, CLERE robustly identified two groups of variables representing influcial ($b_2 > 0$) and not relevant variables ($b_1 = 0$). In the `eyedata` dataset in turn, AIC led to select only one group of variables. However, this did not lessened the predictive performance of the model since CLERE_0 was second best after AVG, while needing significantly less parameters. PACS low performed in both datasets. The Elastic net showed good predictive performances compared to the variable selection methods like LASSO or Spike and slab model. Ridge regression and Elastic net had comparable results in both datasets.

Dataset	Methods	100×Averaged CV-statistic (Std. Error)	Number of parameters (Std. Error)
`Prostate`	LASSO	59.58 (3.46)	5.75 (0.29)
	RIDGE	57.58 (3.36)	8.00 (0.00)
	Elastic net	57.37 (3.39)	8.00 (0.00)
	CLERE	58.18 (3.13)	6.00 (0.00)
	CLERE$_0$	**55.48 (3.46)**	6.00 (0.00)
	AVG	60.59 (3.58)	6.30 (0.16)
	PACS	67.08 (5.51)	5.15 (0.30)
	Spike and slab	57.76 (3.21)	5.70 (0.28)
`eyedata`	LASSO	0.878 (0.05)	27 (1.69)
	RIDGE	0.854 (0.05)	200 (0.00)
	Elastic net	0.851 (0.05)	200 (0.00)
	CLERE	0.877 (0.06)	4 (0.00)
	CLERE$_0$	0.839 (0.05)	4.12 (0.07)
	AVG	**0.811 (0.06)**	17.2 (0.98)
	PACS	2.019 (0.023)	1.38 (0.07)
	Spike and slab	0.951 (0.07)	11.5 (0.55)

Table 3.4: Real data analysis. Out-of-sample prediction error (averaged CV-statistic) was estimated using cross-validation in 100 splitted datasets. The number of parameters reported for CLERE/CLERE$_0$ was selected using AIC.

3.6 Conclusions and Perspectives

We presented in this paper the `R` package `clere`. This package implements an efficient algorithm for fitting the CLusterwise Effect REgression model. This algorithm, namely the SEM algorithm, was compared to a previously published approach and showed a significant improvement in computational time. The good performances of SEM over MCEM could have been expected regarding the computational complexities of the two algorithms that are $\mathcal{O}\left(npg + g^3 + N_{max}ng\right)$ and $\mathcal{O}\left(M(p^2 + pg)\right)$ respectively. In fact, as long as $p > n$, the SEM algorithm has a lower complexity. However, the computational time to run our SEM algorithm is more variable compared to MCEM as its M step does not have a closed form. We finally advocate the use the MCEM algorithm only when $p < n$.

Another improvement was also proposed to facilitate the interpretation. This improve-

ment was proposed by constraining the model parameter b_1 to equal 0. We illustrated through simulations that such constraint may also lead to a reduced prediction error. We illustrated on a real dataset, how to run an analysis using a detailed `R` script presented in Section 3.5.3. Our model can easily be extended to the analysis of binary responses. This extension will be proposed in forthcoming version of the package.

Part II

Extensions of the CLERE methodology

Chapter 4

Extension to binary response data

Binary response data occur in situations where objects can be classified into two categories. Such data are therefore naturally encountered in econometrics (wealthy households versus poor households) or in medical sciences (diseased versus healthy), as specific examples. The main regression-based approaches to handle these data are the Generalized Linear Models (GLM). In GLMs, the binary response c_i for an individual i ($i = 1, \ldots, n$) is assumed to take values into $\{0, 1\}$ and to follow a Bernoulli distribution of probability p_i. Probability p_i is defined as $p_i = F(\mathbf{x}_i\boldsymbol{\beta})$, where $\boldsymbol{x}_i$ is a vector of p covariates, $\boldsymbol{\beta}$ the vector of associated regression coefficients and F is some non-decreasing link function mapping $\mathbb{R}$ into $]0, 1[$. The choice of F is very critical as it influences the interpretation and the tractability of the inference. Classical choices for F are the so-called *Logit* link function defined as

$$Logit : x \mapsto F(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

and the *Probit* link function defined as

$$Probit : x \mapsto F(x) = \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left[-\frac{t^2}{2}\right] dt,$$

This chapter presents possible extensions of the CLERE methodology to handle binary response data. These extensions are considered under the GLM framework and the two classical link functions (*Logit* and *Probit*) are studied. Extension to logistic regression could only be achieved under approximations, since the logistic distribution is not

conjugate with Gaussian-related priors. Some of those approximations are presented but no actual implementation was proposed for this model. In turn, extension to *Probit* regression is done at almost no cost. The numerical experiments presented throughout this chapter were consequently presented only for *Probit* regression.

4.1 Logistic regression

The CLERE methodology can be seen as an *Empirical Bayes* approach. As a consequence, Bayesian approaches for logistic regression were considered as a reference to extend our model. Bayesian approaches for logistic regression using a Gaussian prior for $\boldsymbol{\beta}$ are confronted with the intractability of the posterior distribution $p(\boldsymbol{\beta}|\mathbf{c}, \mathbf{X}, \boldsymbol{\theta})$ [24, 16]. This intractability translates into inability to analytically integrate the complete data likelihood over $\boldsymbol{\beta}$. As already studied in the context of linear regression, such limitation would strongly impact the computational complexity of the inference.

Most strategies to overcome this limitation are based on analytical approximations of the complete data likelihood $p(\boldsymbol{\beta}, \mathbf{c}, \mathbf{X}, \boldsymbol{\theta})$ or on MCMC methods using samples from $p(\boldsymbol{\beta}|\mathbf{c}, \mathbf{X}, \boldsymbol{\theta})$. Two main analytical approximations are proposed in the literature being the Laplace and the variational approximations. The Laplace approximation proposes a Gaussian approximation of the posterior distribution $p(\boldsymbol{\beta}|\mathbf{c}, \mathbf{X}, \boldsymbol{\theta})$ around its mode with respect to $\boldsymbol{\beta}$. This mode has no closed form and the computational complexity to estimate it is often quadratic w.r.t the number of variables. The variational approximation in turn, leads to analytical calculations and is often preferred as it yields computationally efficient inference. Section 4.1 has two sub-sections. First, Section 4.1.1 presents how a variational approximation can be used for inference in our extended model; then in Section 4.1.2, we detail how we can sample from $p(\boldsymbol{\beta}|\mathbf{c}, \mathbf{X}, \boldsymbol{\theta})$ and thus implement a SEM-Gibbs algorithm for estimating the model parameters.

4.1.1 Variational approximation

Variational lower bound for $p(\mathbf{c}, \mathbf{fi}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})$

We use in this sub-section, the approximate variational distribution proposed in [24]. This approximation rely on the following convex inequality:

$$\sigma(x) \geq \sigma(\zeta) \exp\left[\frac{1}{2}(x-\zeta) - \lambda(\zeta)(x^2-\zeta^2)\right], \text{ where } \lambda(\zeta) = \frac{1}{2\zeta}\left[\sigma(\zeta) - \frac{1}{2}\right]. \quad (4.1)$$

We note $\mathbf{c} = (c_1, \ldots, c_n)$, $\mathbf{X}_a = [\mathbf{1}|\mathbf{X}]$ and $\boldsymbol{\beta}_a = [\beta_0, \boldsymbol{\beta}]$. The complete log-likelihood $\log p(\mathbf{c}, \boldsymbol{\beta}_a, \mathbf{Z}|\mathbf{X}_a; \boldsymbol{\theta})$ is defined below

$$\begin{aligned}\log p(\mathbf{c}, \boldsymbol{\beta}_a, \mathbf{Z}|\mathbf{X}_a; \boldsymbol{\theta}) &= \log(\mathbf{c}|\mathbf{X}_a, \mathbf{Z}, \boldsymbol{\beta}_a; \boldsymbol{\theta}) + \log p(\boldsymbol{\beta}_a|\mathbf{X}_a, \mathbf{Z}; \boldsymbol{\theta}) + \log p(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \{c_i \mathbf{x}_i^a \boldsymbol{\beta}_a + \log \sigma(-\mathbf{x}_i^a \boldsymbol{\beta}_a)\} + \log p(\boldsymbol{\beta}_a|\mathbf{X}_a, \mathbf{Z}; \boldsymbol{\theta}) + \log p(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})\end{aligned}$$

Using the inequality (4.1) we obtain a lower bound for $\log p(\mathbf{c}, \boldsymbol{\beta}_a, \mathbf{Z}|\mathbf{X}_a; \boldsymbol{\theta})$ which is expressed below in Equation (4.2):

$$\begin{aligned}\log p(\mathbf{c}, \boldsymbol{\beta}_a, \mathbf{Z}|\mathbf{X}_a; \boldsymbol{\theta}) \geq \boldsymbol{\beta}_a' \mathbf{X}_a' \mathbf{c} + \sum_{i=1}^{n} \log \sigma(\zeta_i) + \frac{1}{2}\sum_{i=1}^{n} (-\mathbf{x}_i^a \boldsymbol{\beta}_a - \zeta_i) - 2\lambda(\zeta_i)\left[(\mathbf{x}_i^a \boldsymbol{\beta}_a)^2 - \zeta_i^2\right] \\ + \log p(\boldsymbol{\beta}_a|\mathbf{X}_a, \mathbf{Z}; \boldsymbol{\theta}) + \log p(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}) \quad (4.2)\end{aligned}$$

The ζ_i's are variational parameters which helps getting the lower-bound as tight as possible. For the sake of simplicity, we will subsequently consider that all the ζ_i's are equal to ζ. This leads to:

$$\begin{aligned}\log p(\mathbf{c}, \boldsymbol{\beta}_a, \mathbf{Z}|\mathbf{X}_a; \boldsymbol{\theta}) \geq \boldsymbol{\beta}_a' \mathbf{X}_a' \left(\mathbf{c} - \frac{1}{2}\mathbf{1}_n\right) + n \log \sigma(\zeta) - \frac{n\zeta}{2} + n\lambda(\zeta)\zeta^2 - \lambda(\zeta)\boldsymbol{\beta}_a' \mathbf{X}_a' \mathbf{X}_a \boldsymbol{\beta}_a \\ + \log p(\boldsymbol{\beta}_a|\mathbf{X}_a, \mathbf{Z}; \boldsymbol{\theta}) + \log p(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}).\end{aligned}$$

Let $\mathbf{y} = \mathbf{c} - \frac{1}{2}\mathbf{1}_n$. The final expression of the variational lower bound is now given

$$\begin{aligned}\log p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}) &\geq \log q(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}) \\ &= \beta_0 \mathbf{1}_n' \mathbf{y} - n\beta_0^2 \lambda(\zeta) + \boldsymbol{\beta}' \mathbf{X}' \left[\mathbf{y} - 2\beta_0 \lambda(\zeta) \mathbf{1}_n\right] - \lambda(\zeta) \boldsymbol{\beta}' \mathbf{X}' \mathbf{X} \boldsymbol{\beta} \\ &\quad + n \log \sigma(\zeta) - \frac{n\zeta}{2} + n\lambda(\zeta)\zeta^2 \\ &\quad - \frac{p}{2} \log(2\pi\gamma^2) - \frac{1}{2\gamma^2} (\boldsymbol{\beta} - \mathbf{Z}\mathbf{b})' (\boldsymbol{\beta} - \mathbf{Z}\mathbf{b}) + \log p(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}).\end{aligned}$$

Maximization w.r.t. the variational parameters

The function $\zeta \mapsto \log q(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}, \zeta)$ is strictly concave. As a consequence, find its maximum amounts to solve the following equation:

$$\frac{\partial \log q(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}, \zeta)}{\partial \zeta} = 0 \implies \zeta^2 = \beta_0^2 + \frac{1}{n} \left[2\beta_0 \boldsymbol{\beta}' \mathbf{X}' \mathbf{1}_n + \boldsymbol{\beta}' \mathbf{X}' \mathbf{X} \boldsymbol{\beta}\right]. \tag{4.3}$$

Variational SEM-Gibbs based on $q(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})$

We have showed in the previous chapter that the SEM-Gibbs algorithm based on the complete data likelihood integrated over $\boldsymbol{\beta}$ was more computational efficient since it does not require sampling $\boldsymbol{\beta}$. A similar integrated approach could have been considered by integrating the variational lower bound $q(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})$, over $\boldsymbol{\beta}$. However, we decided not to make this choice since the resulting lower bound would accumulate approximation errors and consequently be of lower quality compared to $q(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})$. Therefore, this sub-section only concentrates on the description of the SEM-Gibbs algorithm using $q(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})$.

Simulation step

To simulate $(\boldsymbol{\beta}, \mathbf{Z})$ at iteration d, we use the following conditional distributions:

$$\begin{cases} \boldsymbol{\beta}|\mathbf{Z}, \mathbf{y}; \boldsymbol{\theta}^{(d)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(d)}, \boldsymbol{\Sigma}^{(d)}\right) \\ \boldsymbol{\Sigma}^{(d)} = \left(2\lambda(\zeta^{(d)})\mathbf{X}'\mathbf{X} + \frac{1}{\gamma^{2(d)}}\mathrm{I}_p\right)^{-1} \\ \boldsymbol{\mu}^{(d)} = \Sigma^{(d)}\left(\mathbf{X}'\left[\mathbf{y} - 2\beta_0^{(d)}\lambda(\zeta^{(d)})\mathbf{1}_n\right] + \frac{1}{\gamma^{2(d)}}\mathbf{Z}\mathbf{b}^{(d)}\right) \end{cases} \tag{4.4}$$

and

$$p\left(z_{jk} = 1|\boldsymbol{\beta}; \boldsymbol{\theta}^{(d)}\right) \propto \pi_k^{(d)} \exp\left(-\frac{\left(\beta_j - b_k^{(d)}\right)^2}{2\gamma^{2(d)}}\right). \tag{4.5}$$

These equations are similar to Equations (2.6) and (2.7).

Maximization step

The maximization regarding the variational parameter was already described above. In this sub-section we detail the equations for optimizing $q(\mathbf{y}, \boldsymbol{\beta}, \mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})$ regarding to $\boldsymbol{\theta}$. We get the following update equations:

$$\pi_k^{(d+1)} = \frac{1}{p}\sum_{j=1}^{p} z_{jk}^{(d)}, \tag{4.6}$$

$$b_k^{(d+1)} = \frac{1}{p\pi_k^{(d+1)}}\sum_{j=1}^{p} z_{jk}^{(d)}\beta_j^{(d)}, \tag{4.7}$$

$$\gamma^{2(d+1)} = \frac{1}{p}\sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}^{(d)}\left(\beta_j^{(d)} - b_k^{(d+1)}\right)^2, \tag{4.8}$$

$$\beta_0^{(d+1)} = \frac{1}{2n\lambda(\zeta^{(d)})}\left[\mathbf{y} - 2\lambda(\zeta^{(d)})\mathbf{X}\boldsymbol{\beta}^{(d)}\right]'\mathbf{1}_n. \tag{4.9}$$

4.1.2 MCMC based inference

SEM-Gibbs using samples from $p(\boldsymbol{\beta}|\boldsymbol{c}, \boldsymbol{Z}, \boldsymbol{X}; \boldsymbol{\theta})$

Sampling from $p(\boldsymbol{\beta}|\boldsymbol{c}, \boldsymbol{X}; \boldsymbol{\theta})$ when $\boldsymbol{\beta}$ is given a Gaussian prior has been extensively studied by Holmes and Held in [22]. They proposed an auxiliary variable approach

based on the following hierarchy:

$$\begin{cases} c_i = \mathbf{1}_{\{y_i>0\}} \\ y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \\ \varepsilon_i \sim \mathcal{N}(0, \lambda_i) \\ \lambda_i = (2\phi_i)^2 \\ \phi_i \sim KS \end{cases} \tag{4.10}$$

and derived an efficient Gibbs sampler for $p(\boldsymbol{\beta}|\boldsymbol{c}, \boldsymbol{X}; \boldsymbol{\theta})$. In model (4.10), KS stands for the Kolmogorov-Smirnov distribution [14]. Under a Gaussian prior, Andrews and Mallows [3] showed that the marginal distribution $p(\boldsymbol{\beta}|\boldsymbol{c}, \boldsymbol{X}; \boldsymbol{\theta})$ obtained from (4.10) is equivalent to one obtained from the classical definition of the logistic regression. This definition is recalled below in Equation (4.11)

$$\begin{cases} c_i = \mathbf{1}_{\{y_i>0\}} \\ y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \\ \varepsilon_i \sim \text{Logistic}(0,1) \Leftrightarrow \mathbb{P}(\varepsilon_i \leq t) = \sigma(t) = \frac{1}{1+e^{-t}}. \end{cases} \tag{4.11}$$

Using our notations we can express the posterior distribution $p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{X}; \boldsymbol{\theta})$ as in [22]:

$$\begin{cases} \boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{X}; \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{B}, \boldsymbol{V}) \\ \boldsymbol{B} = \boldsymbol{V}\left[\frac{1}{\gamma^2}\boldsymbol{Z}\boldsymbol{b} + \boldsymbol{X}'\boldsymbol{W}(\boldsymbol{y} - \beta_0 \mathbf{1}_n)\right] \\ \boldsymbol{V} = \left(\gamma^2 \mathrm{I}_p + \boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\right)^{-1} \\ \boldsymbol{W} = \text{diag}\left(\lambda_1^{-1}, \ldots, \lambda_n^{-1}\right) \end{cases} \tag{4.12}$$

To be completed, the Gibbs sampler also requires the specification of $p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{X}; \boldsymbol{\theta})$, $p(\boldsymbol{\lambda}|\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{Z}, \boldsymbol{X}; \boldsymbol{\theta})$ and $p(\boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{X}; \boldsymbol{\theta})$. The first two distributions were detailed in [22] and $p(\boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{X}; \boldsymbol{\theta})$ given in Equation (2.7). Once $\boldsymbol{\beta}$ and $\boldsymbol{y}$ are simulated, update equations for model parameters (2.9) to (2.13) can be applied (with $M = 1$). The computational complexity of the algorithm just described is $\mathcal{O}(n^2p^2g^2)$. To reduce such a heavy complexity, we can integrate out $\boldsymbol{\beta}$ and obtain a model that only involves $\boldsymbol{Z}$, $\boldsymbol{y}$ and $\boldsymbol{\lambda}$ as unobserved variables. A SEM-Gibbs algorithm can also be proposed for this integrated model but at a cost of $\mathcal{O}(n^2pg^2)$. This extension is however left out the

scope of this thesis.

4.2 Probit regression

4.2.1 Model definition and parameter estimation

The standard probit regression model is defined by the following equations

$$\begin{cases} c_i = \mathbf{1}_{\{y_i>0\}} \\ y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \\ \varepsilon_i \overset{iid}{\sim} \mathcal{N}(0,1). \end{cases} \tag{4.13}$$

To extend our methodology to models such as model (4.13), we moreover have to assume that

$$\begin{cases} \beta_j|\mathbf{z}_j \overset{iid}{\sim} \mathcal{N}\left(\sum_{k=1}^{g} b_k z_{jk}, \gamma^2\right) \\ \mathbf{z}_j = \left(z_{j1},\ldots,z_{jg}\right) \overset{iid}{\sim} \mathcal{M}\left(\pi_1,\ldots,\pi_g\right). \end{cases} \tag{4.14}$$

which leads to the following complete data likelihood:

$$\log p(\mathbf{c},\mathbf{y},\boldsymbol{\beta},\mathbf{Z};\boldsymbol{\theta}) = \log p(\mathbf{c}|\mathbf{y}) + \log p(\mathbf{y},\boldsymbol{\beta},\mathbf{Z};\boldsymbol{\theta}).$$

This second term in the equation above has been extensively studied in previous chapters. Therefore when integrating over $\boldsymbol{\beta}$, the complete data likelihood (in log scale) is given as

$$\begin{aligned} \log p\left(\mathbf{c},\mathbf{y},\mathbf{Z}|\mathbf{X};\boldsymbol{\theta}\right) = \log p(\mathbf{c}|\mathbf{y}) &- \frac{n}{2}\log\left(2\pi\right) \\ &- \frac{1}{2}\left(\mathbf{y}-\beta_0\mathbf{1}-\mathbf{XZb}\right)'\left[\gamma^2\mathbf{XX}'+\mathrm{I}_n\right]^{-1}\left(\mathbf{y}-\beta_0\mathbf{1}-\mathbf{XZb}\right) \\ &+ \log p(\mathbf{Z};\boldsymbol{\theta}). \end{aligned} \tag{4.15}$$

Similarly to the linear case, we propose to infer the model parameters using the SEM-Gibbs algorithm described in chapter 3. The difference with the linear case is that

we introduced an additional unobserved variable, namely $\mathbf{y}$, that we will also have to sample. $p(\mathbf{y}|\mathbf{c},\mathbf{Z},\mathbf{X};\boldsymbol{\theta})$ is a multivariate truncated Gaussian distribution. Sampling from a multivariate truncated Gaussian distribution is known to be difficult. However, using conditional distributions a Gibbs sampler can be proposed as explained in [22].

Let $\mathbf{H}$ being defined as

$$\mathbf{H} = \left[\mathrm{I}_n + \gamma^2 \mathbf{X}\mathbf{X}'\right]^{-1} \tag{4.16}$$

and $\boldsymbol{\mu} = \beta_0 \mathbf{1}_n + \mathbf{X}\mathbf{Z}\mathbf{b}$. The conditional distribution $p(y_i|\mathbf{y}^{-i},\mathbf{a})$ is given below

$$y_i|\mathbf{y}^{-i},\boldsymbol{\mu} \sim \begin{cases} \mathcal{N}\left(\mu_i - \frac{1}{H_{ii}}\sum_{i'\neq i}^{n} H_{ii'}(y_{i'}-\mu_{i'}), \frac{1}{H_{ii}}\right)\mathbf{1}_{\{y_i>0\}} & \text{if } c_i = 1 \\ \mathcal{N}\left(\mu_i - \frac{1}{H_{ii}}\sum_{i'\neq i}^{n} H_{ii'}(y_{i'}-\mu_{i'}), \frac{1}{H_{ii}}\right)\mathbf{1}_{\{y_i\leq 0\}} & \text{otherwise.} \end{cases} \tag{4.17}$$

Using the SVD decomposition of matrix $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}'$, we can analytically express matrix $\mathbf{H} = (H_{ik})$:

$$H_{ik} = \gamma^2 \sum_{l=1}^{n} \left\{ \lambda_l^2 \times \frac{u_{kl}u_{il}}{\gamma^2\lambda_l^2 + 1} \right\}, \text{ where } \mathbf{U} = (u_{ik})\,. \tag{4.18}$$

Once $\mathbf{y}$ is generated, the M-step is similar to the one of the linear model described in the previous chapter.

The extension to the probit regression model was implemented in C++ and integrated to the `R` package `clere` (version $\geq$ 1.2). To make inference, one can use the function `fit.clere()` by setting the input parameter `analysis="binomial"`. All other options works similarly.

4.2.2 Numerical experiments

This section presents numerical experiments on simulated and real data. These experiments aims at comparing the our extended CLERE-probit to classical model for high-dimensional binary regression models. The methods selected for comparison are the Lasso logistic regression and the ridge logistic regression.

Simulated data

The comparison is performed in terms of classification error on a simulated validation set of 500 individuals. The sample size in the training sets was varied between 100 and 200. The simulated number of variables is $p = 100$. All covariates were simulated independently according to the standard Gaussian distribution. The latent data y_i was simulated as follows:

$$y_i \overset{iid}{\sim} \mathcal{N}\left(\sum_{j=1}^{40} 0 \times x_{ij} + \sum_{j=41}^{70} -1 \times x_{ij} + \sum_{j=71}^{100} 1 \times x_{ij}, 1\right) \tag{4.19}$$

The CLERE-probit model was fitted using 20 different random starting points.
Figure 4.1 shows the classification errors associated with the three methods compared. When $p < n$, the logistic ridge has the lowest classification error. CLERE-probit with the constraint $b_1 = 0$ is second best method. When $p > n$, CLERE-probit shows the lowest classification error compared to its competitors.

Leukemia dataset

We used in this section the dataset `leukemia` from the `R` package `spikeslab`. This dataset involves gene expression measured in samples from human acute myeloid (coded as 0) and acute lymphoblastic leukemias (coded as 1). 3571 expression values were measured on 72 individuals. We primarily reduced the number of variables to 1412 by only including variables showing a nominal association with the response (p-value<0.05). The same three methods were compared using 3-fold cross-validation. Table 4.1, summarizes the results of that comparison.

All methods were comparable in terms of classification error. However, it is noteworthy that logistic ridge and CLERE-probit showed the lowest classification errors (both yieled an error equal to 0.013).

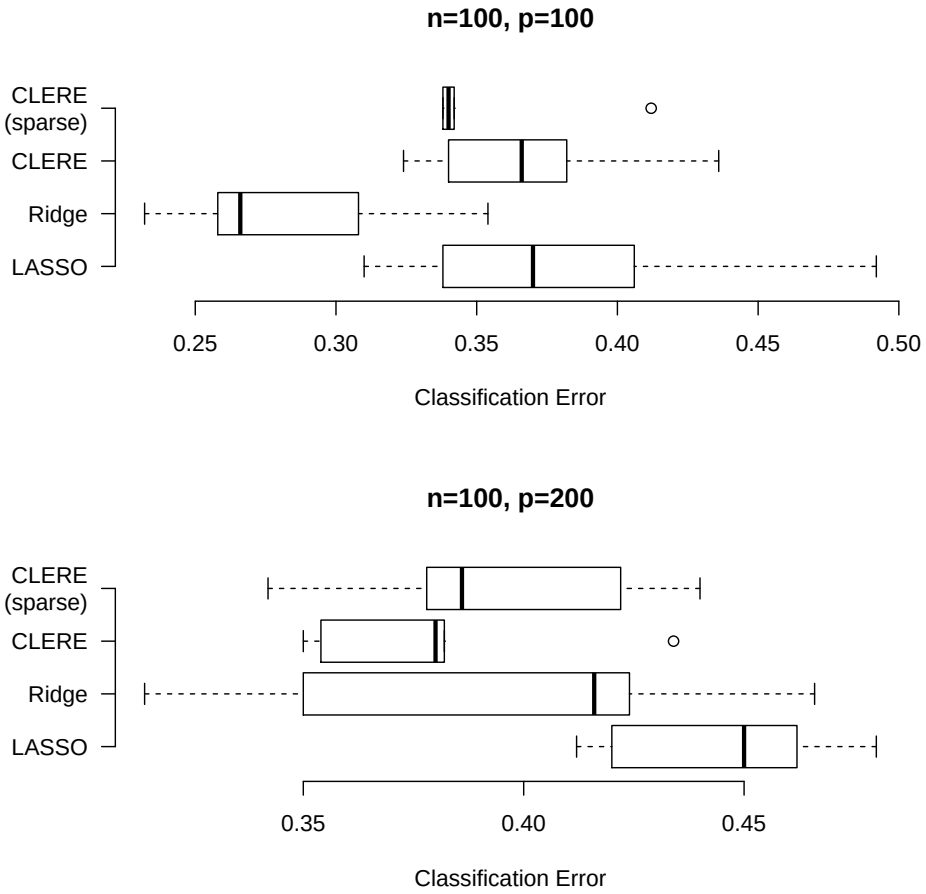

Figure 4.1: Classification errors associated with CLERE-probit, Logistic LASSO and Logistic Ridge. 50 replications were considered.

New York Times stories dataset

We used in this section a dataset from Cosma Shalizi web-page[1]. This dataset consists in 102 stories published in the New York times. 57 of those stories dealt with art and 45 with music. 4432 variables were collected corresponding to words that occurred in at least one story. Each word was then associated with the number of times it occurred in the selected stories. For this analysis, we focused on words which frequency across stories was above 10%. This took the number of variables to 511. We compared the methods in terms of classification error. The latter error was estimated via 3-fold cross-

[1]http://www.stat.cmu.edu/ cshalizi/490/pca/pca-examples.Rdata

Datasets	Methods	Averaged classification Error (Std. Error)	Number of parameters (Std. Error)
`leukemia`	LASSO	0.055 (0.03)	20.0 (1.52)
	RIDGE	0.013 (0.01)	1412 (0.00)
	CLERE-probit	0.013 (0.01)	2.00 (0.00)
	CLERE$_0$-probit	0.069 (0.04)	3.33 (0.66)
`New York Times stories`	LASSO	0.252 (0.01)	27.2 (1.46)
	RIDGE	0.196 (0.01)	511. (0.00)
	CLERE-probit	0.215 (0.01)	2.31 (0.09)
	CLERE$_0$-probit	0.246 (0.01)	4.00 (0.07)

Table 4.1: Out of sample classification error estimated via 3-fold cross-validation. The number of parameters reported for CLERE/CLERE$_0$ was selected using AIC.

validation. Table 4.1, reports the results of that comparison.
The best method was the logistic ridge regression with an averaged classification error of 0.196. The second best was CLERE-probit (averaged classification error = 0.215). CLERE-probit selected always one group. The logistic lasso showed the worst performances.

4.3 Discussion

We presented in this chapter an extension of the CLERE methodology for handling binary response data. This extension was shown to be more challenging than the linear regression case.
One first challenge lies in the choice of the link function. We illustrated in this chapter that although the *logit* link function is often recommended for interpretation purposes, its implementation leads to complex algorithms. Nevertheless, the logistic distribution can be approximated using a Gaussian distribution. Under this approximation, the CLERE-*logit* model can be estimated using an algorithm rather similar to the one proposed in this chapter for the CLERE-*probit* model.
The second challenge relates to the simulation of the auxiliary variable $\boldsymbol{y}$. This necessary step significantly increases the computational complexity of the inference. More efficient approaches have been proposed to directly sample from $p(\boldsymbol{\beta}|\boldsymbol{c}, \boldsymbol{X}; \boldsymbol{\theta})$ without

using an auxiliary variable. The later algorithms introduced by Maruyaama and colleagues [29], could not however apply to the general context of this thesis since they were introduced with the requirement that matrix $\boldsymbol{X}$ have full rank.
Despite the limitations just underlined, the extension of CLERE for binary response data showed very competitive prediction performances compared to known approaches for dimension reduction. As such, the numerical experiments using both simulated and real datasets were really encouraging.
Finally, the extension presented in this chapter can be considered as a first step to generalize the CLERE methodology to multi-class response data and ordinal response data.

Chapter 5

On relaxing the common variance assumption

The CLERE methodology was introduced [49] under the assumption that regression coefficients are random variables drawn from a mixture of g ($g \geq 1$) Gaussian distributions. The components of the mixture were so far assumed to have unequal means $(b_1, \ldots, b_g)$ but equal variances (all equal to γ^2). In this chapter, we propose a new model that relaxes the assumption of equal variances. The rationale for studying this new model is twofold. The first motivation is to improve flexibility with respect to the former definition of our model. Indeed, having this extended alternative model would allow to test whether the common variance assumption is actually supported by the data. In addition, the CLERE method is effective at identifying groups when the effect size associated with the latter are distinct enough. We called this *separation between the clusters* in Chapter 2. As in classical mixture models, this separation is intrinsically related to the variance parameters of each component. We therefore also study this new model as a possible means to improve cluster detection. This chapter first describes how to make inference in this new model. Then, we explore through numerical experiments the consequences of this gain in flexibility over the previously introduced CLERE methodology.

5.1 Model definition

The model is now written:

$$\begin{cases} y_i \sim \mathcal{N}\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \sigma^2\right) \\ \beta_j | \mathbf{z}_j \sim \mathcal{N}\left(\sum_{k=1}^{g} b_k z_{jk}, \sum_{k=1}^{g} \gamma_k^2 z_{jk}\right) \\ \mathbf{z}_j = (z_{j1}, \ldots, z_{jg}) \sim \mathcal{M}(\pi_1, \ldots, \pi_g). \end{cases} \tag{5.1}$$

where γ_k^2 stands for the variance of k-th component of the mixture. Let $\boldsymbol{\Delta}$ be the $p \times p$ diagonal matrix which j-th diagonal term δ_j equals $\delta_j = \sum_{k=1}^{g} z_{jk}\gamma_k^2$. Integrating model (5.1) over $\boldsymbol{\beta}$ leads to the following set of equations

$$\begin{cases} \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ \boldsymbol{\mu} = \beta_0 \mathbf{1}_n + \boldsymbol{X}\boldsymbol{Z}\boldsymbol{b} \\ \boldsymbol{\Sigma} = \sigma^2 \mathrm{I}_n + \boldsymbol{X}\boldsymbol{\Delta}\boldsymbol{X}' \\ \mathbf{z}_j = (z_{j1}, \ldots, z_{jg}) \sim \mathcal{M}(\pi_1, \ldots, \pi_g). \end{cases} \tag{5.2}$$

The complete data log-likelihood integrated over $\boldsymbol{\beta}$ can thus be written

$$\log p\left(\boldsymbol{y}, \boldsymbol{Z} | \boldsymbol{X}; \widetilde{\boldsymbol{\theta}}\right) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})'\,\Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu}) + \sum_{j,k} z_{jk}\log(\pi_k).$$

We use a SEM algorithm [10] for estimating the model parameters as proposed in Chapter 3. This algorithm is a stochastic version of the EM algorithm [13] commonly utilized for maximum likelihood in presence of incomplete (unobserved) data. This algorithm starts with initial values for the model parameters $\widetilde{\boldsymbol{\theta}} = \left(\beta_0, \boldsymbol{b}, \boldsymbol{\pi}, \gamma_1^2, \ldots, \gamma_g^2, \sigma^2\right)' \in \mathbb{R}^{(3g+1)}$ and alternates two steps, namely the *simulation* and the *maximization* steps. The *simulation* step consists here in simulating unobserved partition $\boldsymbol{Z}$ using the conditional distribution $p(\boldsymbol{Z}|\boldsymbol{y}, \boldsymbol{X}; \widetilde{\boldsymbol{\theta}})$, while the maximization performed afterwards, consists in maximizing the complete data likelihood $p\left(\boldsymbol{y}, \boldsymbol{Z} | \boldsymbol{X}; \widetilde{\boldsymbol{\theta}}\right)$ with respect to $\widetilde{\boldsymbol{\theta}}$. *Initialization*, *simulation* and *maximization* steps of the algorithm are described in Section 5.2.

5.2 Parameter estimation

5.2.1 Initialization

The CLERE model (2.3) is a special case of model (5.1). As a consequence, we suggest using estimates obtained under the common variance assumption to initialize the parameters in model (5.1). Such estimates may be obtained after fitting the CLERE model from different (random) starting points.

5.2.2 Simulation step

The simulation step proposed in this subsection uses a Gibbs sampler based on the conditional distribution $p(z_{jk}|\boldsymbol{Z}^{-j}, \boldsymbol{y}, \boldsymbol{X}; \widetilde{\boldsymbol{\theta}})$. One possible way to evaluate this conditional distribution would be to use the Bayes theorem:

$$p(z_{jk}|\boldsymbol{Z}^{-j}, \boldsymbol{y}, \boldsymbol{X}; \widetilde{\boldsymbol{\theta}}) \propto p(z_{jk}, \boldsymbol{Z}^{-j}, \boldsymbol{y}, \boldsymbol{X}; \widetilde{\boldsymbol{\theta}}).$$

However, this naive way requires the calculation of the complete data likelihood which itself necessitates the inversion of the $n \times n$ covariance matrix $\boldsymbol{\Sigma}$. We propose in this subsection a more efficient calculation for these conditional distributions.
The covariance matrix $\boldsymbol{\Sigma}$ can be written as linear combination of the variance parameters:

$$\boldsymbol{\Sigma} = \sigma^2 \mathrm{I}_n + \sum_{j=1}^{p} \sum_{k=1}^{g} z_{jk} \gamma_k^2 \boldsymbol{x}_j \boldsymbol{x}_j',$$

where $\boldsymbol{x}_j$ is the j-th column of matrix $\boldsymbol{X}$. As consequence, assigning variable variable $\boldsymbol{x}_j$ to the group k^{new} modifies $\boldsymbol{\Sigma}$ as following:

$$\boldsymbol{\Sigma}^{new} = \boldsymbol{\Sigma}^{old} + \left(\gamma_{k^{new}}^2 - \gamma_{k^{old}}^2\right) \boldsymbol{x}_j \boldsymbol{x}_j'. \tag{5.3}$$

A similar result can be derived for the mean $\boldsymbol{\mu}$:

$$\boldsymbol{\mu}^{new} = \boldsymbol{\mu}^{old} + \left(b_{k^{new}} - b_{k^{old}}\right) \boldsymbol{x}_j. \tag{5.4}$$

Let $\Delta\gamma^2 = \gamma_{k^{new}}^2 - \gamma_{k^{old}}^2$ and $\Delta b = b_{k^{new}} - b_{k^{old}}$. Using the Sherman-Morrison identity

[40] we can write:

$$\left[\boldsymbol{\Sigma}^{old} + \Delta\gamma^2 \boldsymbol{x}_j \boldsymbol{x}_j'\right]^{-1} = \left(\boldsymbol{\Sigma}^{old}\right)^{-1} - \frac{\Delta\gamma^2}{1 + \Delta\gamma^2 \boldsymbol{x}_j' \left(\boldsymbol{\Sigma}^{old}\right)^{-1} \boldsymbol{x}_j} \left(\boldsymbol{\Sigma}^{old}\right)^{-1} \boldsymbol{x}_j \boldsymbol{x}_j' \left(\boldsymbol{\Sigma}^{old}\right)^{-1} \tag{5.5}$$

and

$$\log |\boldsymbol{\Sigma}^{old} + \Delta\gamma^2 \boldsymbol{x}_j \boldsymbol{x}_j'| = \log |\boldsymbol{\Sigma}^{old}| + \log \left[1 + \Delta\gamma^2 \boldsymbol{x}_j' \left(\boldsymbol{\Sigma}^{old}\right)^{-1} \boldsymbol{x}_j\right]. \tag{5.6}$$

Equations (5.5) and (5.6), can therefore be used to evaluate the conditional distribution at a computational cost of $\mathcal{O}\left(n^2 pg\right)$ instead of $\mathcal{O}\left(n^3 pg\right)$.

5.2.3 Maximization step

Up to a constant, $\log p\left(\boldsymbol{y}, \boldsymbol{Z} | \boldsymbol{X}; \widetilde{\boldsymbol{\theta}}\right)$ can be interpreted as the log-likelihood of the following linear mixed model:

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{t} + \boldsymbol{X}\boldsymbol{\phi} + \boldsymbol{\varepsilon}. \tag{5.7}$$

where $\boldsymbol{M}$ is the $n \times (g+1)$ matrix which first column is made of 1's and which additional columns are those of matrix $\boldsymbol{X}\boldsymbol{Z}$, $\boldsymbol{t} = (\beta_0, \boldsymbol{b})' \in \mathbb{R}^{(g+1)}$, $\boldsymbol{\phi} \sim \mathcal{N}(0, \boldsymbol{\Delta})$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_n)$.

The maximization step can consequently be obtained through an *internal EM algorithm* just as detailed in Chapter 3. However, new *internal E and M* steps have to be derived. Those steps are described further in the chapter.

Complete data likelihood associated with model (5.7)

Let $\boldsymbol{\theta}_0 = \left(\beta_0, b_1, \ldots, b_g, \gamma_1^2, \ldots, \gamma_g^2, \sigma^2\right)' \in \mathbb{R}^{(2g+2)}$. The complete data log-likelihood $\log p(\boldsymbol{y}, \boldsymbol{\phi}; \boldsymbol{\theta}_0)$ associated with model (5.7) is given below

$$\begin{aligned}\log p(\boldsymbol{y}, \boldsymbol{\phi}; \boldsymbol{\theta}_0) &= \log p(\boldsymbol{y}|\boldsymbol{\phi}; \boldsymbol{\theta}_0) + \log p(\boldsymbol{\phi}; \boldsymbol{\theta}_0) \\ &= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t} - \boldsymbol{X}\boldsymbol{\phi}\|_2 - \frac{p}{2}\log(2\pi) \\ &- \frac{1}{2}\sum_{jk} z_{jk}\left[\log(\gamma_k^2) + \frac{\phi_j^2}{\gamma_k^2}\right]. \end{aligned} \tag{5.8}$$

Internal *E step*

At iteration s of the *internal EM algorithm*, the *internal E step* requires the calculation of $\mathbb{E}\left[\phi_j^2|\boldsymbol{y}; \boldsymbol{\theta}^{(s)}\right]$, $\mathbb{E}\left[\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\phi}|\boldsymbol{y}; \boldsymbol{\theta}^{(s)}\right]$ and $\mathbb{E}\left[\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t} - \boldsymbol{X}\boldsymbol{\phi}\|_2|\boldsymbol{y}; \boldsymbol{\theta}^{(s)}\right]$. Those quantities have been detailed in [39] (pages 297 to 301). Using our notations we can express them as following

$$\begin{aligned}\mathbb{E}\left[\phi_j^2|\boldsymbol{y}; \boldsymbol{\theta}_0^{(s)}\right] &= \sum_{k=1}^{g} z_{jk}\left[\gamma_k^{4(s)}\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t}^{(s)}\right)'\boldsymbol{\Sigma}^{-1(s)}\boldsymbol{x}_j\boldsymbol{x}_j'\boldsymbol{\Sigma}^{-1(s)}\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t}^{(s)}\right)\right] \\ &+ \sum_{k=1}^{g} z_{jk}\left[\gamma_k^{2(s)} - \gamma_k^{4(s)}\boldsymbol{x}_j'\boldsymbol{\Sigma}^{-1(s)}\boldsymbol{x}_j\right], \end{aligned} \tag{5.9}$$

$$\mathbb{E}\left[\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\phi}|\boldsymbol{y}; \boldsymbol{\theta}_0^{(s)}\right] = \boldsymbol{M}\boldsymbol{t}^{(s)} + \sigma^{2(s)}\Sigma^{-1(s)}\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t}^{(s)}\right), \tag{5.10}$$

and

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t} - \boldsymbol{X}\boldsymbol{\phi}\|_2|\boldsymbol{y}; \boldsymbol{\theta}_0^{(s)}\right] &= \sigma^{4(s)}\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t}^{(s)}\right)'\boldsymbol{\Sigma}^{-1(s)}\boldsymbol{\Sigma}^{-1(s)}\left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t}^{(s)}\right) \\ &+ n\sigma^{2(s)} - \sigma^{4(s)}\mathrm{Tr}(\boldsymbol{\Sigma}^{-1(s)}). \end{aligned} \tag{5.11}$$

Internal *M step*

The update equations can also be deduced from [39]:

$$\sigma^{2(s+1)} = \frac{1}{n}\mathbb{E}\left[\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t} - \boldsymbol{X}\boldsymbol{\phi}\|_2|\boldsymbol{y}; \boldsymbol{\theta}_0^{(s)}\right]. \tag{5.12}$$

$$\gamma_k^{2(s+1)} = \frac{\sum_{j=1}^{p} z_{jk} \mathbb{E}\left[\phi_j^2 | \boldsymbol{y}; \boldsymbol{\theta}_0^{(s)}\right]}{\sum_{j=1}^{p} z_{jk}}. \tag{5.13}$$

$$\boldsymbol{t}^{(s+1)} = \boldsymbol{t}^{(s)} + \sigma^{2(s)} \left(\boldsymbol{M}'\boldsymbol{M}\right)^{-1} \boldsymbol{M}'\Sigma^{-1(s)} \left(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{t}^{(s)}\right). \tag{5.14}$$

The overall computational complexity for the maximization step is $\mathcal{O}\left(n^2 p g N_{max}\right)$, where N_{max} is the number of EM iterations until convergence.

5.3 Numerical Experiments on simulated data

5.3.1 Description of the scenarios

This section presents numerical experiments on simulated data. These experiments aim at comparing the CLERE method introduced in chapter 2 with the enhancement proposed in this chapter. The enhanced model will subsequently be denoted as CLARA[1]. CLERE$_0$ stands for the CLERE model under the constraint that $b_1 = 0$ (see Chapter3) while CLARA$_0$ represents the enhanced model under the same constraint. CLERE and CLARA were compared in terms of prediction error and classification error (ability to recover the partition of covariates) using 100 pairs of training and validation datasets. We also added the LASSO and Elastic net for relative comparison. With the two latter methods, the classification error was assessed as the ability to identify non-zero covariates. For CLERE, CLERE$_0$, CLARA and CLARA$_0$, the number of groups was selected using the AIC criterion between 1 and 3. Each training dataset had 30 individuals while corresponding validation sets had 500.
Each row $\boldsymbol{x}_i$ in design matrices of both training and validation datasets was simulated according to a multivariate normal distribution:

$$\boldsymbol{x}_i' \in \mathbb{R}^p \sim \mathcal{N}(0, \boldsymbol{R}); \boldsymbol{R} = (r_{jl}) \ / \ r_{jl} = 0.5^{|j-l|}$$

A vector of regression coefficients was simulated for each pair of training and valida-

[1]The name CLARA is not an acronym. It was chosen because of phonetic proximity with the name CLERE.

tion datasets as following:

$$\forall j = 1, \ldots, p \; \beta_j \sim \begin{cases} \mathcal{N}(3,3) \text{ if } \exists h \; / \; j = 3h \\ 0 \text{ else.} \end{cases}$$

For $p = 15$, an example of sampled regression coefficients is given below through an R line command:

```
R> p  <- 15 ; ms <- rep(0,p) ; ms[which(1:p%%3==0)] <- 3
R> beta = rnorm(p,mean=ms,sd=sqrt(ms))
R> beta
 [1] 0.00 0.00 6.695575 0.00 0.00 1.935368 0.00 0.00
 [9] 3.402003 0.00 0.00 4.342324 0.00 0.00 1.450633
```

The variance of the residuals was set as $\sigma^2 = 1$. Three scenarios were considered according to the number p of covariates simulated. Scenario 1 corresponds to $p = 15$, Scenario 2 to $p = 30$ and Scenario 3 to $p = 60$.

5.3.2 Results

Table 5.1 presents the results of the whole simulation study.

Prediction error

In the first two scenarios ($p \leq n$) we observed that the CLARA method had a lower prediction error compared to CLERE. This improvement that was still noticeable under the constraint that parameter $b_1 = 0$, allowed to yield an averaged prediction error below the ones of variable selection techniques LASSO and Elastic net. In the third scenario ($p > n$), all methods yield very comparable results. No improvement of CLARA over CLERE was noticed. Moreover, the lowest prediction error was achieved by the unconstrained CLERE model.

Classification error

In all considered scenarios, the CLARA method showed an improved classification error compared to CLERE. In the scenario 1, this improvement is almost two-fold. We

also observed that, as p increased the difference in classification errors between the two methods tended to decrease.

# of variables (# of non-zero variables)	Methods	Prediction Error (Std. Error)	Classification Error (Std. Error)	Number of parameters (Std. Error)
$p = 15$ (5)	Elastic net	1.681 (0.06)	0.685 (0.02)	10.24 (0.25)
	LASSO	1.653 (0.06)	0.263 (0.01)	9.49 (0.23)
	CLERE	1.846 (0.05)	0.137 (0.01)	7.26 (0.11)
	$CLERE_0$	1.765 (0.05)	0.126 (0.01)	6.26 (0.10)
	CLARA	1.554 (0.05)	0.068 (0.01)	7.54 (0.12)
	$CLARA_0$	**1.525 (0.06)**	**0.061 (0.01)**	6.51 (0.11)
$p = 30$ (10)	Elastic net	3.943 (0.25)	0.667 (0.01)	20.091 (0.31)
	LASSO	**3.911 (0.28)**	0.296 (0.01)	19.465 (0.31)
	CLERE	5.992 (0.50)	0.151 (0.01)	7.313 (0.10)
	$CLERE_0$	6.063 (0.73)	0.146 (0.01)	6.333 (0.10)
	CLARA	7.589 (3.87)	0.107 (0.01)	8.364 (0.15)
	$CLARA_0$	4.068 (0.73)	**0.095 (0.01)**	7.152 (0.15)
$p = 60$ (20)	Elastic net	98.14 (5.14)	0.547 (0.01)	32.711 (0.73)
	LASSO	115.1 (6.24)	0.313 (0.01)	24.402 (0.60)
	CLERE	**96.77 (4.01)**	0.305 (0.01)	4.969 (0.12)
	$CLERE_0$	104.5 (5.04)	0.290 (0.01)	5.536 (0.10)
	CLARA	100.3 (4.30)	**0.271 (0.01)**	7.093 (0.16)
	$CLARA_0$	104.2 (4.83)	0.289 (0.01)	6.619 (0.12)

Table 5.1: Averaged classification error and out-of-sample prediction error over 100 replications on simulated data. The number of parameters reported for CLERE/$CLERE_0$ and CLARA/$CLARA_0$ was selected using AIC. In these experiments, the computational time for running CLARA was 30 times higher than the time required for running CLERE

5.4 Numerical Experiments on real data

5.4.1 Description of the datasets

We used in this section like in Chapter 3 the real datasets `Prostate` and `eyedata` from the `R` packages `lasso2` and `flare` respectively. The `Prostate` dataset comes from a study that examined the correlation between the level of prostate specific antigen and a number of clinical measures in $n = 97$ men who were about to receive a radical prostatectomy. This dataset was used in multiple publications including [45]. We used the prostate specific antigen (variable `lpsa`) as response variable and the $p = 8$ other measurements as covariates.

The `eyedata` dataset is extracted from the published study of Scheetz and colleagues [37]. This dataset consists in gene expression levels measured at $p = 200$ probes in $n = 120$ rats. The response variable utilized was the expression of the TRIM32 gene which is a biomarker of the Bardet-Bidel Syndrome (BBS).

Those two datasets was utilized to compare CLERE, CLARA, LASSO and Elastic net. All the methods were compared in terms of out-of-sample prediction error estimated using cross-validation (CV). 100 CV statistics were calculated by randomly splitting each dataset into training ($n = 30$) and validation (remaining number of samples) sets. The choice of having training set of size 30 is consistent with the scenarios presented in Section 5.3. Those CV statistics were then averaged and compared across the methods in Table 5.2.

5.4.2 Results

Table 5.2 summarizes the prediction errors and the number of parameters obtained for all the methods. CLARA_0 method had the lowest prediction error in both datasets. However, a specific behavior can be noticed in each data set.

In the `Prostate` dataset, $\text{CLERE}/\text{CLERE}_0$ and $\text{CLARA}/\text{CLARA}_0$ selected in average around 2 and 3 groups respectively. When the constraint "$b_1 = 0$" was not active, the increased number of parameters yielded by CLARA over CLERE did not translate into reduced prediction error. However, when this constraint is imposed, CLARA_0 outperformed not only $CLERE_0$ but also the LASSO and the Elastic net.

The behavior of the CLARA method compared to CLERE was somewhat different in

the second dataset. Indeed, in this dataset, the CLARA/CLARA$_0$ method selected one group in 74% of the simulations. As a result, in most situations CLARA and CLERE were the same models. Nevertheless, some differences are noticeable between the two models regarding the prediction errors. These differences are in our sense explained by the fact that CLARA reached higher likelihood compared to CLERE since the former method was initialized from estimates obtained with the latter.
These results as those obtained on simulated data, stress that CLARA and CLERE tend to be equivalent in high dimensional settings.

Dataset	Methods	$100\times$Averaged CV-statistic (Std. Error)	Number of parameters (Std. Error)
`Prostate`	LASSO	70.27 (1.45)	5.16 (0.17)
	Elastic net	69.71 (1.42)	5.88 (0.17)
	CLERE	70.86 (1.91)	5.88 (0.06)
	CLERE$_0$	65.10 (1.73)	5.02 (0.05)
	CLARA	72.24 (1.85)	9.97 (0.03)
	CLARA$_0$	**64.32 (1.47)**	9.00 (0.00)
`eyedata`	LASSO	0.919 (0.02)	11.35 (0.67)
	Elastic net	0.883 (0.02)	63.34 (7.71)
	CLERE	1.055 (0.02)	4.808 (0.09)
	CLERE$_0$	0.937 (0.02)	3.626 (0.09)
	CLARA	1.083 (0.02)	5.000 (0.18)
	CLARA$_0$	**0.835 (0.01)**	3.000 (0.00)

Table 5.2: Real data analysis. Out-of-sample prediction error (averaged CV-statistic) was estimated using cross-validation in 100 splitted datasets. The number of parameters reported for CLERE/CLERE$_0$ and CLARA/CLARA$_0$ was selected using AIC.

5.5 Discussion

Introducing additional flexibility to the CLERE method by relaxing the assumption of equal variances among the components comes at a heavy computational cost (almost n

times). However, we have been able to identify situations were this additional flexibility also yields both improved predictive performances and fewer classification errors. Even if the simulation scenarios were more favorable to the unconstrained model (a.k.a CLARA model), it is worth underlining that those scenarios were also favorable to variable selection approaches like the LASSO and the Elastic net. However, we observed in Scenario 1 for instance, that the CLARA approach improved the CLERE method to the extent of being better than the LASSO and Elastic net regarding both prediction and classification error. The latter improvement underlined that the CLERE method might be considered ,at least when the common variance assumption is relaxed, as an effective variable selection approach.

In high dimension, the difference between CLERE and CLARA was not obvious. In fact, CLERE had even better predictive performances on simulated and real data. This last observation leads to the following conclusion: the heavy computation cost yielded by CLARA in high dimension does not have to be paid since CLERE already performs well. CLERE model is nested in CLARA model. Thus, there might be situations where model selection criteria like AIC or BIC would preferred CLARA model as the best model for describing the data. Any user of those models should therefore be allowed to make such a choice. As a consequence, researches aiming at lowering the computational cost for model inference in CLARA model remains a attractive perspective.

Chapter 6

Magnitude-driven variable clustering

This chapter presents a new extension of the CLERE methodology. This extension aims at increasing the parsimony of the CLERE model by allowing variables to cluster together regardless the sign of their regression coefficients but only using the magnitude of the latter. This can be translated into replacing the assumption (6.1) of the CLERE model

$$\beta_j|\boldsymbol{z}_j \overset{iid}{\sim} \mathcal{N}\left(\sum_{k=1}^{g} z_{jk} b_k, \gamma^2\right), \tag{6.1}$$

with the following new assumption:

$$\beta_j|\boldsymbol{z}_j, \omega_j \overset{iid}{\sim} \mathcal{N}\left(\sum_{k=1}^{g} z_{jk}(2\omega_j - 1) b_k, \gamma^2\right). \tag{6.2}$$

where ω_j is an unobserved Bernoulli distributed random variable. From Equation (6.2) we can see that when $\omega_j = 1$ and $z_{jk} = 1$, β_j is centered around b_k; while when $\omega_j = 0$ and $z_{jk} = 1$, β_j is centered around $-b_k$. The present chapter has three sections. In Section 6.1 we briefly recall the model definition and how to make inference in that new model. Section 6.2 presents few numerical experiments on both simulated and real data. Those experiments aimed at comparing the modified CLERE model, subsequently called CLERE2 with the initial CLERE model in terms of prediction and classification error. The classification error means here the ability of recovering the latent partition of the covariates. Finally, Section 6.3 discusses the strengths and weaknesses

of CLERE2 compared to classical CLERE.

6.1 Model definition and inference

6.1.1 Model definition

Let $\boldsymbol{\theta}_2 = \left(\beta_0, \boldsymbol{b}, \boldsymbol{\pi}, \gamma^2, \sigma^2, \alpha\right)' \in \mathbb{R}^{(2g+4)}$. The CLERE2 model can be written:

$$\begin{cases} y_i|\boldsymbol{\beta}, \boldsymbol{X}; \boldsymbol{\theta}_2 \overset{iid}{\sim} \mathcal{N}\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \sigma^2\right) \\ \beta_j|\mathbf{z}_j, \omega_j; \boldsymbol{\theta}_2 \overset{iid}{\sim} \mathcal{N}\left(\sum_{k=1}^{g} z_{jk}(2\omega_j - 1)b_k, \gamma^2\right) \\ \mathbf{z}_j = \left(z_{j1}, \ldots, z_{jg}\right) \overset{iid}{\sim} \mathcal{M}\left(\pi_1, \ldots, \pi_g\right) \\ \omega_j \overset{iid}{\sim} \mathcal{B}(\alpha); \alpha \in [0; 1]. \end{cases} \tag{6.3}$$

Let $\boldsymbol{\Omega}$ be the $p \times p$ diagonal matrix which j-th diagonal term equals ω_j. Integrating model (6.3) over $\boldsymbol{\beta}$ leads to the following set of equations

$$\begin{cases} \boldsymbol{y}|\boldsymbol{Z}, \boldsymbol{\Omega}, \boldsymbol{X}; \boldsymbol{\theta}_2 \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \\ \boldsymbol{\mu} = \beta_0 \mathbf{1}_n + \boldsymbol{X}\boldsymbol{\Omega}\boldsymbol{Z}\boldsymbol{b} \\ \boldsymbol{\Sigma} = \sigma^2 \mathrm{I}_n + \boldsymbol{X}\boldsymbol{X}' \\ \mathbf{z}_j = \left(z_{j1}, \ldots, z_{jg}\right) \sim \mathcal{M}\left(\pi_1, \ldots, \pi_g\right) \\ \omega_j \overset{iid}{\sim} \mathcal{B}(\alpha); \alpha \in [0; 1]. \end{cases} \tag{6.4}$$

Let the SVD decomposition of matrix $\boldsymbol{X}$ be $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}'$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are respectively $n \times n$ and $p \times p$ orthogonal matrices, and $\boldsymbol{S}$ is $n \times p$ rectangular diagonal matrix which diagonal terms are the eigenvalues $\left(\lambda_1^2, \ldots, \lambda_n^2\right)$ of matrix $\boldsymbol{X}\boldsymbol{X}'$. We now define $\boldsymbol{X}^u = \boldsymbol{U}'\boldsymbol{X}$ and $\boldsymbol{y}^u = \boldsymbol{U}'\boldsymbol{y}$. Let $\mathbf{1}_n$ be the vector of $\mathbb{R}^n$ which each coordinate equals 1. Matrix $\boldsymbol{M}$ is defined as the $n \times (g+1)$ matrix which first column equals $\boldsymbol{s}^u = \boldsymbol{U}'\mathbf{1}_n = \left(s_1^u, \ldots, s_n^u\right)'$ and which additional columns are those of matrix $\boldsymbol{X}^u\boldsymbol{Z}$. Let also $\boldsymbol{t} = (\beta_0, \boldsymbol{b}) \in \mathbb{R}^{(g+1)}$ and $\boldsymbol{R}$ be a $n \times n$ diagonal matrix which i-th diagonal term equal $\sigma^2 + \gamma^2\lambda_i^2$. With these notations we can express the complete data likelihood

integrated over $\boldsymbol{\beta}$ as

$$\begin{aligned}\log p\left(\boldsymbol{y},\boldsymbol{Z},\boldsymbol{\Omega}|\boldsymbol{X};\boldsymbol{\theta}_2\right) = &-\frac{n}{2}\log\left(2\pi\right) - \frac{1}{2}\sum_{i=1}^{n}\log\left(\sigma^2+\gamma^2\lambda_i^2\right)\\ &-\frac{1}{2}\left(\boldsymbol{y}^u-\boldsymbol{M}\boldsymbol{t}\right)'\boldsymbol{R}^{-1}\left(\boldsymbol{y}^u-\boldsymbol{M}\boldsymbol{t}\right)+\sum_{j=1}^{p}\sum_{k=1}^{g}z_{jk}\log\pi_k\\ &+\left(\sum_{j=1}^{p}w_j\right)\times\log\frac{\alpha}{1-\alpha}+p\log(1-\alpha).\end{aligned} \tag{6.5}$$

6.1.2 Parameter estimation

As in Chapter 3 and Chapter 5, we propose to estimate the model parameters using a SEM algorithm [10]. This algorithm is a stochastic version the EM algorithm [13] commonly utilized for maximum likelihood in presence of incomplete (unobserved) data. This algorithm starts with initial values for the model parameters and alternates two steps, namely the *simulation* and the *maximization* steps. The *simulation* step consists here in simulating the unobserved partition $\boldsymbol{Z}$ and the sign $\boldsymbol{\Omega}$ of the regression coefficients using the conditional distribution $p(\boldsymbol{Z},\boldsymbol{\Omega}|\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}_2)$. The *maximization* step is performed afterwards and consists in maximizing the complete data likelihood $p\left(\boldsymbol{y},\boldsymbol{Z},\boldsymbol{\Omega}|\boldsymbol{X};\boldsymbol{\theta}_2\right)$ with respect to $\boldsymbol{\theta}_2$.

Simulating from $p(\boldsymbol{Z},\boldsymbol{\Omega}|\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}_2)$

Let $\boldsymbol{Z}^{-j}$ and Ω^{-j} respectively denote the sets of cluster membership indicators and sign indicators for all covariates but the j-th. We propose to perform the simulation step using a Gibbs sampler based on the conditional distributions $p(\boldsymbol{z}_j|\boldsymbol{Z}^{-j},\boldsymbol{\Omega},\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}_2)$ and $p(\boldsymbol{\omega}_j|\boldsymbol{Z},\boldsymbol{\Omega}^{-j},\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}_2)$ for $j=1,\ldots,p$.
Let $\boldsymbol{w}^{-j}=\left(w_1^{-j},\ldots,w_n^{-j}\right)'$, where $w_i^{-j}=y_i^u-\beta_0 s_i^u-\sum_{l\neq j}\sum_{k=1}^{g}z_{lk}(2\omega_j-1)x_{il}^u b_k$. The conditional distribution $p(z_{jk}=1|\boldsymbol{Z}^{-j},\boldsymbol{\Omega},\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}_2)$ can be written

$$p(z_{jk}=1|\boldsymbol{Z}^{-j},\boldsymbol{\Omega},\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta}_2)\propto\pi_k\exp\left[-\frac{b_k^2}{2}\left(\boldsymbol{x}_j^u\right)'\boldsymbol{R}^{-1}\boldsymbol{x}_j^u+b_k(2\omega_j-1)\left(\boldsymbol{w}^{-j}\right)'\boldsymbol{R}^{-1}\boldsymbol{x}_j^u\right], \tag{6.6}$$

where $\boldsymbol{x}_j^u$ is the j-th column of $\boldsymbol{X}^u$. We can also express the conditional distribution $p(\omega_j = 1|\boldsymbol{\Omega}^{-j}, \boldsymbol{Z}, \boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}_2)$ as follows:

$$logit\left[\mathbb{P}(\omega_j = 1|\boldsymbol{\Omega}^{-j}, \boldsymbol{Z}, \boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}_2)\right] = \log\left(\frac{\alpha}{1-\alpha}\right) + 2\sum_{k=1}^{g} z_{jk} b_k \left(\boldsymbol{w}^{-j}\right)' \boldsymbol{R}^{-1} \boldsymbol{x}_j^u. \quad (6.7)$$

We recall that the *logit* function maps $]0;1[$ to $\mathbb{R}$ and is defined as $logit : x \mapsto \log\left[\frac{x}{1-x}\right]$.

Maximization step

In the context of model (6.3), the *maximization* step for parameters β_0, $\boldsymbol{b}$, $\boldsymbol{\pi}$, σ^2 and γ^2 is identical to the one proposed in Chapter 3 for classical CLERE. The remaining parameter α is updated at iteration d of the algorithm using Equation (6.8):

$$\alpha^{(d+1)} = \frac{1}{p}\sum_{j=1}^{p} \omega_j^{(d)}. \quad (6.8)$$

In Equation (6.8), $\omega_j^{(d)}$ is simulated using $p(\boldsymbol{Z}, \boldsymbol{\Omega}|\boldsymbol{y}, \boldsymbol{X}; \boldsymbol{\theta}_2^{(d)})$ where $\boldsymbol{\theta}_2^{(d)}$ denotes the value of parameter $\boldsymbol{\theta}_2$ at iteration d of the algorithm.

6.2 Numerical experiments

6.2.1 Experiments on simulated data

Description of the simulation study

The experiments presented in this section aim at comparing CLERE and CLERE2 models in terms of prediction error and classification error, i.e. ability to recover the latent partition of covariates. This comparison was performed over 100 pairs of training and validation datasets. CLERE and CLERE2 were compared under four scenarios. In all scenarios the size of training and validation datasets was respectively $n = 50$ and $N = 1000$. Each scenario corresponded to different numbers of covariates to be included in the model. The number of covariates was varied among 25, 50, 75 and 100, respectively for scenario 1, 2, 3 and 4. Table 6.1 details the regression coefficients used to generate simulated data.

Scenario	Regression coefficients
1	$\boldsymbol{\beta} = (\underbrace{-15,\ldots,-15}_{3},\underbrace{-3,\ldots,-3}_{4},\underbrace{0,\ldots,0}_{11},\underbrace{+3,\ldots,+3}_{4},\underbrace{+15,\ldots,+15}_{3})' \in \mathbb{R}^{25}$
2	$\boldsymbol{\beta} = (\underbrace{-15,\ldots,-15}_{6},\underbrace{-3,\ldots,-3}_{8},\underbrace{0,\ldots,0}_{22},\underbrace{+3,\ldots,+3}_{8},\underbrace{+15,\ldots,+15}_{6})' \in \mathbb{R}^{50}$
3	$\boldsymbol{\beta} = (\underbrace{-15,\ldots,-15}_{9},\underbrace{-3,\ldots,-3}_{12},\underbrace{0,\ldots,0}_{33},\underbrace{+3,\ldots,+3}_{12},\underbrace{+15,\ldots,+15}_{9})' \in \mathbb{R}^{75}$
4	$\boldsymbol{\beta} = (\underbrace{-15,\ldots,-15}_{12},\underbrace{-3,\ldots,-3}_{16},\underbrace{0,\ldots,0}_{44},\underbrace{+3,\ldots,+3}_{16},\underbrace{+15,\ldots,+15}_{12})' \in \mathbb{R}^{100}$

Table 6.1: Description of the regression coefficients used in each scenario to generate simulated data for comparing CLERE and CLERE2 models.

All covariates were simulated independently according to the standard Gaussian distribution:

$$\forall (i,j)\ x_{ij} \sim \mathcal{N}(0,1).$$

Training and validation datasets were simulated according to a classical linear regression model with regression coefficients depending on the scenarios. Details about the regression coefficients used in each scenario are given in Table 6.1. For CLERE and CLERE2 models, the number of groups was selected using AIC and each SEM algorithm used to estimate the model parameters was run using 15 different random starting points.

Results of the comparison

Figure 6.1 illustrates the results of the simulation study. The progression from Scenario 1 to Scenario 4 represents an increase in the model dimensionality and complexity. We observed along this progression that the performances of CLERE2 compared to CLERE tended to deteriorate. Indeed, in Scenario 1 and Scenario 2, CLERE2 outperformed CLERE both in terms of prediction and classification error. This marked improvement is accompanied with a reduced model complexity since CLERE2 had in average three less parameters compared to CLERE. Although CLERE2 remained more parsimonious

than CLERE in subsequent scenarios, the difference in terms of prediction error between CLERE and CLERE2 is less noticeable in Scenario 3 and, CLERE2 is even worse than CLERE in Scenario 4. However, CLERE2 still had a lower or equal classification error rate compared to CLERE in Scenario 3 and Scenario 4.

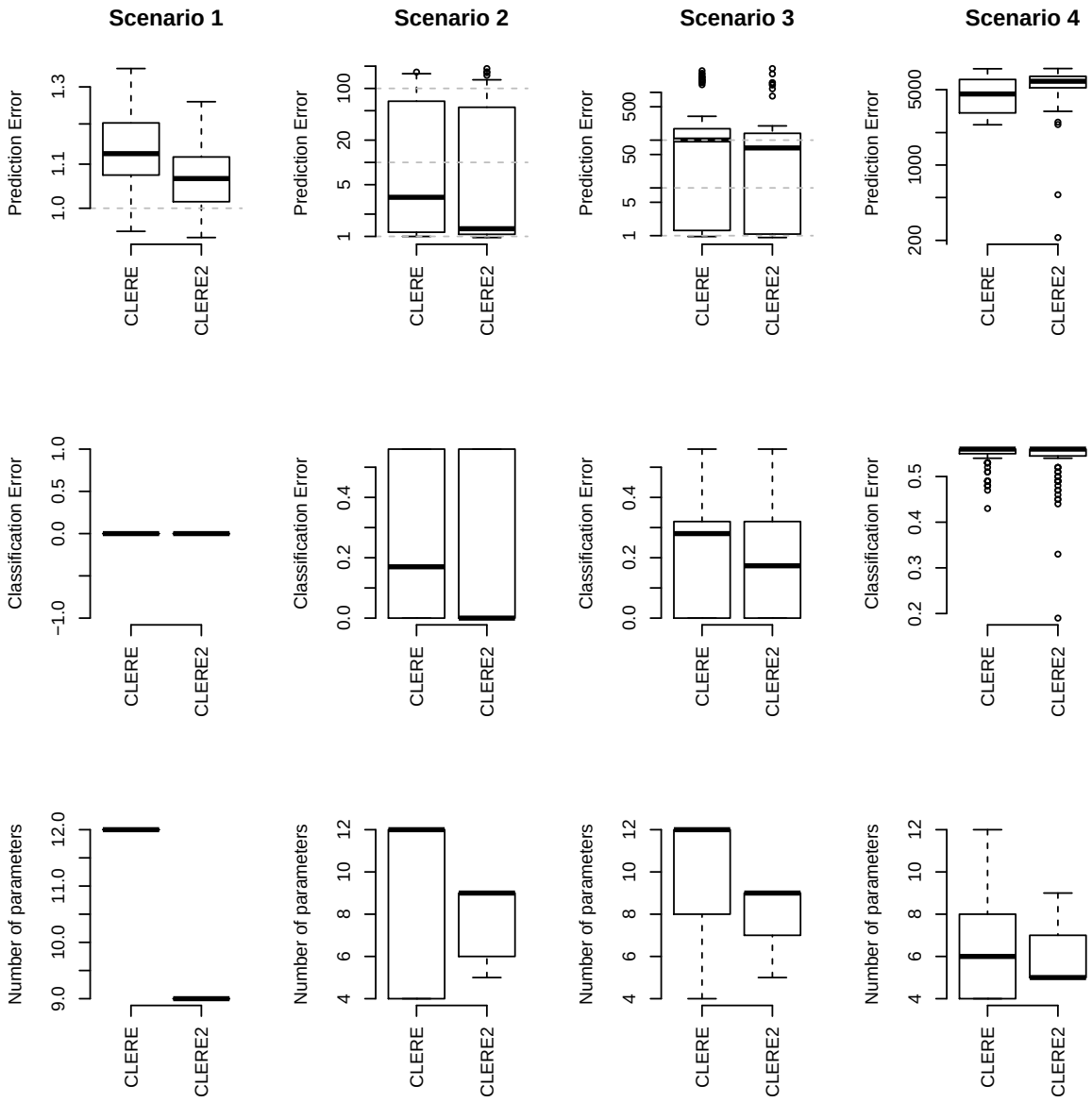

Figure 6.1: Comparison of the distribution of prediction error between CLERE and CLERE2 models across different scenarios.

6.2.2 Experiments on real data

In this section we compared CLERE and CLERE2 models in terms of prediction error on real data. We used for comparison the same datasets described previously in Chapter 3 and 5. We therefore advocate the reader to refer to Section 5.4.1 for more details regarding those datasets. For relative comparison, we also considered the prediction error yielded by the LASSO and Elastic net. Table 6.2 presents the out-of-sample prediction errors obtained for all the methods compared. In the `Prostate` dataset CLERE2 had a larger prediction error compared to CLERE while in the `eyedata` CLERE2 was better. Both model were however quite similar regarding their number of parameters.

Dataset	Methods	100×Averaged CV-statistic (Std. Error)	Number of parameters (Std. Error)
`Prostate`	LASSO	70.27 (1.45)	5.16 (0.17)
	Elastic net	**69.71 (1.42)**	5.88 (0.17)
	CLERE	70.86 (1.91)	5.88 (0.06)
	CLERE2	73.16 (1.85)	5.90 (0.06)
`eyedata`	LASSO	0.919 (0.02)	11.35 (0.67)
	Elastic net	0.883 (0.02)	63.34 (7.71)
	CLERE	1.055 (0.02)	4.808 (0.09)
	CLERE2	**0.853 (0.03)**	4.020 (0.20)

Table 6.2: Real data analysis. Out-of-sample prediction error (averaged CV-statistic) was estimated using cross-validation in 100 splitted datasets. The number of parameters reported for CLERE and CLERE2 was selected using AIC.

6.3 Discussion

We have introduced in this chapter a new model in line with the CLERE methodology. The estimation of the parameters of that new model was performed by applying a minor modification to the estimation strategy already proposed for CLERE.
This new model model was mostly expected to be more parsimonious than classical CLERE. However its performances in terms of prediction and classification errors

needed to be assessed. Through numerical experiments, we have been able to identify situations were CLERE methodology could be improved using that new model. Nevertheless, most of those situations were in low dimensional settings. The latter observation therefore leads us to consider the use of this extended model when the number of covariates is small compared to the sample size or when the Bayesian information criterion leads to select the extended model over classical CLERE.

Chapter 7

Conclusion

This chapter gives a summary of the work presented in this manuscript and highlights further improvements.

7.1 Summary

We proposed in this thesis an original contribution to the field of variable clustering in linear regression through a model-based approach. This contribution was made via a hierarchical modeling of the regression coefficients as random variables drawn from a mixture of Gaussian distributions with equal variances. By transferring the distributional assumption from the covariates to the regression coefficients, our model makes variable clustering easier in cases where the variables do not belong to the same family of distributions (binary and continuous covariates might be clustered together). Parameter estimation in the proposed model was shown to be challenging since the classical EM algorithm could not apply. We first studied an estimation strategy based on a Monte Carlo approximation of the EM algorithm. This strategy also known as the Monte Carlo EM (MCEM) algorithm was however found to be very slow. We then developed a more efficient algorithm for parameter estimation, through the use of the SEM-Gibbs algorithm. Along with this computational improvement, we also enhanced our model to allow variable selection. Given the good predictive performances of the CLERE method compared to standard techniques for dimension reduction, we considered an extension of the latter to binary response data. This extension was studied in the context of Probit regression. This extension also showed competitive predictive

performances. We finally considered two other extensions of our methodology. First, we generalized our model by relaxing the assumption of equal variance for the components in the mixture of Gaussians. The performances of this generalization were compared to those of the initial model under different scenarios on simulated data. Although more computationally intensive, this generalization was shown to improve the classification error associated with the recovery of the latent partition of the covariates. The second extension explored was studied to improve the model parsimony and aimed at performing variable clustering regarless the sign of their effect size but only using their magnitude. We studied the predictive performances of this new model and revealed its limitation in high dimensional settings. This research led to the development of the `R` package `clere` which implements most of the algorithms described in this thesis.

7.2 Discussion

7.2.1 On the quality of the maximum likelihood estimator

We discuss in this section some of the elements that influence the quality of the maximum likelihood estimator (MLE) proposed in the CLERE model.
Let $\boldsymbol{\theta}$ denote the parameter to be estimated in the CLERE model. If $\widehat{\boldsymbol{\theta}}$ is the MLE of parameter $\boldsymbol{\theta}$, then under regularity assumptions $\widehat{\boldsymbol{\theta}}$ verifies:

$$\sqrt{n}\left(\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}\right) \xrightarrow{\mathcal{L}} \mathcal{N}\left(\mathbf{0}; I^{-1}\left(\boldsymbol{\theta};\boldsymbol{y}\right)\right), \tag{7.1}$$

as any maximum likelihood estimator. In Equation (7.1), $I^{-1}\left(\boldsymbol{\theta};\boldsymbol{y}\right)$ stands for the Fisher Information Matrix (FIM) of the CLERE model.
Equation (7.1) translates that when the sample size n is large enough $\widehat{\boldsymbol{\theta}}$ can be assumed normally distributed with mean $\boldsymbol{\theta}$ and covariance matrix $\frac{1}{n}I^{-1}\left(\boldsymbol{\theta};\boldsymbol{y}\right)$.
As a consequence, the variance of $\widehat{\boldsymbol{\theta}}$ is simultaneously influenced by the sample size n and by the *shape* of $I^{-1}\left(\boldsymbol{\theta};\boldsymbol{y}\right)$. The *shape* of a matrix, can be assessed through many different metrics including the matrix dimension or the matrix condition number[1]. The

[1]The condition number of a matrix is defined as the ratio between its largest and its smallest eigenvalue.

dimension of the FIM is known in a general context to deteriorate the variances of the estimated parameters. Likewise, a FIM with a small condition number implies estimates with large variances. What may influence the condition number of the FIM is quite specific to the types of models. In the CLERE model and more generally in mixture models, the condition number of the FIM is affected by the separation between the clusters.

To illustrate this statement, we will consider a simpler version of the CLERE model obtained by assuming two clusters ($g = 2$) and known values for some parameters. We recall below in Equation (7.2), the general form of the CLERE model:

$$\begin{cases} y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \\ \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \\ \beta_j | \mathbf{z}_j \sim \mathcal{N}\left(\sum_{k=1}^{g} b_k z_{jk}, \gamma^2\right) \\ \mathbf{z}_j = (z_{j1}, \ldots, z_{jg}) \sim \mathcal{M}(\pi_1, \ldots, \pi_g). \end{cases} \tag{7.2}$$

The simpler version of the model is obtained by assuming $\beta_0 = 0$, $\sigma^2 = 1$, $\gamma^2 = 0$ and $\pi_1 = \pi_2 = 0.5$. That simpler model can be written:

$$\begin{cases} y_i = b_1 \times \left(\sum_{j=1}^{p} x_{ij}\right) + \sum_{j=1}^{p} x_{ij} z_j (b_2 - b_1) + \varepsilon_i \\ \varepsilon_i \sim \mathcal{N}(0, 1) \\ z_j \sim \mathcal{B}ernoulli(0.5). \end{cases} \tag{7.3}$$

The separation δ in model (7.3) between the clusters can be measured by the difference between b_2 and b_1: $\delta = b_2 - b_1$. If we moreover assume that $b_1 = 0$, then the FIM of model (7.3) is a scalar and still a function of the cluster separation $\delta = b_2$. We propose below in Equation (7.4) an expression of the FIM in model (7.3) under the additional assumption that $b_1 = 0$:

$$I(\delta; \boldsymbol{y}) = \sum_{i=1}^{n} \mathbb{E}_{y_i} \left(\sum_{\boldsymbol{z} \in \mathcal{Z}} \left[a_i(\boldsymbol{z}) - \delta b_i(\boldsymbol{z}) \right]^2 w_i(\boldsymbol{z}; \delta) \right) \tag{7.4}$$

where

$$a_i(\boldsymbol{z}) = \sum_{j=1}^{p} y_i x_{ij} z_j \text{ and } b_i(\boldsymbol{z}) = \sum_{j=1}^{p} \sum_{j'=1}^{p} x_{ij} x_{ij'} z_j z_{j'}, \tag{7.5}$$

$$w_i(\boldsymbol{z};\delta) = \frac{p(y_i,\boldsymbol{z};\delta)}{\sum_{\boldsymbol{z}\in\mathcal{Z}} p(y_i,\boldsymbol{z};\delta)} \tag{7.6}$$

and $p(y_i,\boldsymbol{z};\delta)$ is the likelihood of y_i under model (7.3). We cannot formally establish from Equation (7.4) how $I(\delta;\boldsymbol{y})$ varies with respect to δ. However, we can use simulated data to represent the covariation of latter quantities. Figure 7.1 represents the variation of $I(\delta;\boldsymbol{y})$ along with δ. The values depicted on Figure 7.1 were obtained using simulated data based on the predictors in the `Prostate` dataset (see Chapter 3). 1000 response variable were then simulated according to model (7.3) and averaged to yield the approximated FIM. Figure 7.1 supports the positive influence of the cluster separation on the variance of the estimated parameters. Nevertheless, this remains to be formally demonstrated in further researches.

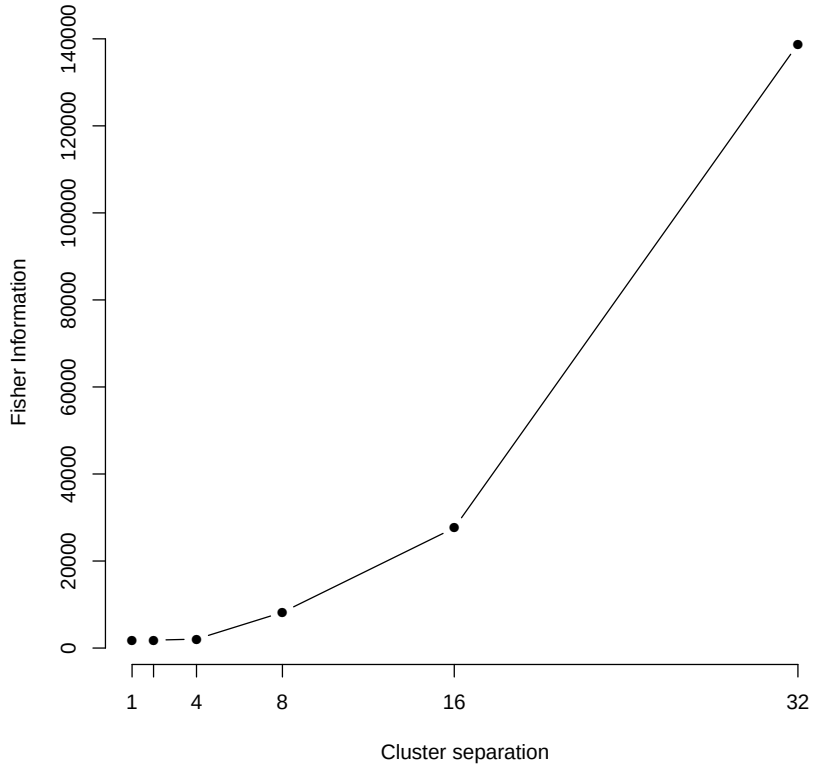

Figure 7.1: Fisher Information Matrix $I(\delta;\boldsymbol{y})$ in view of the cluster separation δ.

7.2.2 On knowledge-based modeling of $p(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\theta})$

The model we proposed in this thesis made the assumption that the distribution of the clusters did not a priori depends on the covariates. This choice translated into specifying $p(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\theta})$ as follows:

$$\log p(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\theta}) = \sum_{j=1}^{p}\sum_{k=1}^{g} z_{jk}\log(\pi_k). \tag{7.7}$$

Making other choices for $p(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\theta})$ is undoubtedly a way of improvement for our model. In fact, as in *group LASSO*, we can incorporate in our model some prior information regarding the covariates, which might be valuable for interpreting the model. Such an improvement may also have computational implications. For instance, if the user has reasons to considered that some partitions are not likely to be observed, then the computational time for evaluating the posterior distribution $p(\boldsymbol{Z}|\boldsymbol{y},\boldsymbol{X};\boldsymbol{\theta})$ will be reduced consequently.

7.2.3 On improved *M step*

In Chapter 3, an analogy was drawn between the *M step* in our model and the maximization of the marginal likelihood in a linear mixed model. This analogy has been valuable for implementing an *internal* EM algorithm in linear complexity with respect to the sample size n. To be fully efficient, this step has however to account for known improvements proposed to perform maximum likelihood estimation in linear mixed models. The optimal strategy as proposed in [27], is often a mix between the EM algorithm and the Newton-raphson algorithm. The first is often used to come up with good starting values and the latter, which converges quicker, is used to end the optimization. Such an improvement will be accounted in forthcoming versions of the `R` package `clere`.

7.2.4 On additional perspectives

In Chapter 5, we showed that relaxing the common variance assumption often leads to improved prediction error and classification error. However, these improvements come at the cost of a heavy computational burden. Reducing the computational com-

plexity of the algorithm for parameter estimation in this context is a major perspective of this research.
We presented in this thesis few applications to real datasets of manageable size. Another very exciting perspective of this work is definitely the application of our model at a larger scale.

Scientific communications

Communications related to this thesis

Conference abstracts

1. L. Yengo, J. Jacques and C. Biernacki, A Block Regression Approach for Simultaneous Variables Clustering and Selection: Application to Genetic Data. *Journées Ouvertes de Biologie, Informatique et Mathématiques*. Paris, (2011).

2. L. Yengo, J. Jacques and C. Biernacki, Classification et Sélection de Variables en Régression. *Journées de Statistiques de la Société Francaise de Statistiques*. Bruxelles, (2012).

Articles in peer-reviewed journals

1. L. Yengo, J. Jacques and C. Biernacki, Variables Clustering in High dimensional Linear Regression Models, *Journal de la Société Francaise de Statistiques*, (2013). In Press.

2. L. Yengo, J. Jacques and C. Biernacki, Variables Clustering in High dimensional Linear Regression: The R package clere, (2014). Submitted to *Journal of Statistical Software*.

R package

L. Yengo, J. Jacques and C. Biernacki, Variables Clustering in High dimensional Linear Regression: The R package clere, (2014). Submitted to *Journal of Statistical Software*.

Other publications in the field of Genetics

1. Coffee and tea consumption, genotype-based CYP1A2 and NAT2 activity and colorectal cancer risk-Results from the EPIC cohort study.Dik VK, Bueno-de-Mesquita HB, Van Oijen MG, Siersema PD, Uiterwaal CS, Van Gils CH, Van Duijnhoven FJ, Cauchi S, Yengo L, *et al.*. Int J Cancer. (2013).

2. GUESS-ing polygenic associations with multiple phenotypes using a GPU-based evolutionary stochastic search algorithm.Bottolo L, Chadeau-Hyam M, Hastie DI, Zeller T, Liquet B, Newcombe P, Yengo L,*et al.*. PLoS Genet. (2013).

3. Parental history of type 2 diabetes, TCF7L2 variant and lower insulin secretion are associated with incident hypertension. Data from the DESIR and RISC cohorts.Bonnet F, Roussel R, Natali A, Cauchi S, Petrie J, Laville M, Yengo L,*et al.* Diabetologia. (2013).

4. Association between large detectable clonal mosaicism and type 2 diabetes with vascular complications.Bonnefond A, Skrobek B, Lobbens S, Eury E, Thuillier D, Cauchi S, Lantieri O, Balkau B, Riboli E, Marre M, Charpentier G, Yengo L, Froguel P. Nat Genet. (2013).

5. Common variants near BDNF and SH2B1 show nominal evidence of association with snacking behavior in European populations. Robiou-du-Pont S, Yengo L, Vaillant E, Lobbens S, Durand E, Horber F, Lantieri O, Marre M, Balkau B, Froguel P, Meyre D. J Mol Med (2013).

6. Analysis of the contribution of FTO, NPC1, ENPP1, NEGR1, GNPDA2 and MC4R genes to obesity in Mexican children.Mejia-Benitez A, Klünder-Klünder M, Yengo L,*et al.*. BMC Med Genet. (2013).

7. Genome-wide association analyses identify 18 new loci associated with serum urate concentrations.Köttgen A, Albrecht E, Teumer A, Vitart V, Krumsiek J, Hundertmark C, Pistis G, Ruggiero D, O'Seaghdha CM, Haller T, Yang Q, Tanaka T, Johnson AD, Kutalik Z, Smith AV, Shi J, Struchalin M, Middelberg RP, Brown MJ, Gaffo AL, Pirastu N, Li G, Hayward C, Zemunik T, Huffman J, Yengo L, *et al.*. Nat Genet. (2013).

8. Large-scale association analyses identify new loci influencing glycemic traits and provide insight into the underlying biological pathways. Scott RA, Lagou V, Welch RP, Wheeler E, Montasser ME, Luan J, Mägi R, Strawbridge RJ, Rehnberg E, Gustafsson S, Kanoni S, Rasmussen-Torvik LJ, Yengo L, *et al.*. Nat Genet. (2012).

9. Large-scale association analysis provides insights into the genetic architecture and pathophysiology of type 2 diabetes. Morris AP, Voight BF, Teslovich TM, Ferreira T, Segrè AV, Steinthorsdottir V, Strawbridge RJ, Khan H, Grallert H, Mahajan A, Prokopenko I, Kang HM, Dina C, Esko T, Fraser RM, Kanoni S, Kumar A, Lagou V, Langenberg C, Luan J, Lindgren CM, Müller-Nurasyid M, Pechlivanis S, Rayner NW, Scott LJ, Wiltshire S, Yengo L,*et al.*. Nat Genet. (2012).

10. Stratifying type 2 diabetes cases by BMI identifies genetic risk variants in LAMA1 and enrichment for risk variants in lean compared to obese cases. Perry JR, Voight BF, Yengo L,*et al.*. PLoS Genet. (2012).

11. Dysfunction of lipid sensor GPR120 leads to obesity in both mouse and human.Ichimura A, Hirasawa A, Poulain-Godefroy O, Bonnefond A, Hara T, Yengo L,*et al.*. Nature. (2012).

12. Rare MTNR1B variants impairing melatonin receptor 1B function contribute to type 2 diabetes. Bonnefond A, Clément N, Fawcett K, Yengo L, *et al.*. Nat Genet. (2012).

13. Low-frequency variants in HMGA1 are not associated with type 2 diabetes risk. Marquez M, Huyvaert M, Perry JR, Pearson RD, Falchi M, Morris AP, Vivequin S, Lobbens S, Yengo L,*et al.*. Diabetes. (2012).

14. Heterozygous mutations causing partial prohormone convertase 1 deficiency contribute to human obesity. Creemers JW, Choquet H, Stijnen P, Vatin V, Pigeyre M, Beckers S, Meulemans S, Than ME, Yengo L, *et al.*. Diabetes. (2012).

15. Disruption of a novel Kruppel-like transcription factor p300-regulated pathway for insulin biosynthesis revealed by studies of the c.-331 INS mutation found in neonatal diabetes mellitus.Bonnefond A, Lomberk G, Buttar N, Busiah K, Vaillant E, Lobbens S, Yengo L, *et al.*. J Biol Chem. (2011).

Bibliography

[1] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[2] E. Amaldi and V. Kann. On the approximability of minimizing non zero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209:237–260, 1998.

[3] D. Andrews and C. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 36:99–102, 1974.

[4] C. Biernacki. Degeneracy in the Maximum Likelihood Estimation of Univariate Gaussian Mixtures for Grouped Data and Behaviour of the EM Algorithm. *Journal of Scandinavian Statistics*, 34:569–586, 2007.

[5] C. Biernacki, G. Celeux, and G. Goavert. Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

[6] C. Biernacki and J. Jacques. A generative model for rank data based on insertion sort algorithm. *Computational Statistics and Data Analysis*, 58:162–176, 2013.

[7] H. D. Bondell and B. J. Reich. Simultaneous Regression Shrinkage, Variable Selection, and Supervised Clustering of Predictors with OSCAR. *Biometrics*, 64:115–123, 2008.

[8] P. Buhlmann, P. Rutimann, S. van de Geer, and C.H. Zhang. Correlated variables in regression: Clustering and sparse estimation. *Journal of Statistical Planning and Inference*, 143:1835–1858, 2013.

[9] G. Casella. An Introduction to Empirical Bayes Data Analysis. *The American Statistician*, 39(2):83–87, 1985.

[10] G. Celeux, D. Chauveau, and J. Diebolt. Some Stochastic versions of the EM Algorithm. *Journal of Statistical Computation and Simulation*, 55:287–314, 1996.

[11] H. Chun and S. Keles. Expression Quantitative Trait Loci Mapping With Multivariate Sparse Partial Least Squares Regression. *Genetics*, 182:79–90, 2009.

[12] Z.J. Daye and X.J. Jeng. Shrinkage and model selection with correlated variables via weighted fusion. *Computational Statistics & Data Analysis*, 53(4):1284–1298, 2009.

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

[14] L. Devroye. *Non-Uniform Random Variate Generation.* New York: Springer. Springer, 1986.

[15] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[16] A. Genkin, D.D. Lewis, and D. Madigan. Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49:291–304, 2007.

[17] G. Govaert and M. Nadif. Block clustering with Bernoulli mixture models: Comparison of different approaches. *Computational Statistics and Data analysis*, 52:3233–3245, 2008.

[18] A. Gunawardana and W. Byrne. Convergence theorems for generalized alternating minimization procedures. *Journal of Machine Learning Research*, 6:2049–2073, 2005.

[19] T. Hastie, R. Tibshirani, and J.H. Friedman. *The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations*. New York: Springer-Verlag, 2001.

[20] R.R. Hocking. The analysis and Selection of variables inlinear regression. *Biometrics*, 32:1–49, 1976.

[21] A. E. Hoerl and W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12:55–67, 1970.

[22] C.C. Holmes and L. Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1):145–168, 2006.

[23] H. Ishwaran and J. Sunil Rao. Spike and slab variable selection: frequentist and Bayesian strategies. *Annals of Statistics*, 33(2):730–773, 2005.

[24] T.S. Jaakkola and M.I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 1999.

[25] I. T. Jolliffe. A Note on the Use of Principal Components in Regression. *Applied Statistics*, 31(3):300+, 1982.

[26] R. A. Levine and G. Casella. Implementations of the Monte Carlo EM Algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

[27] M.J. Lindstrom and D.M. Bates. Newton-Raphson and EM Algorithms for Linear Mixed-Effects Models for Repeated-Measures Data. *Journal of the American Statistical Association*, 83(404):1014–1022, 1988.

[28] M. Mariadassou, S. Robin, and C. Vacher. Uncovering Latent Structure in Valued Graphs: a Variational Approach. *The Annals of Applied Statistics*, 4(2):715–742, 2010.

[29] Y. Maruyama and W.E. Strawderman. A new Monte Carlo sampling in Bayesian probit regression. *arXiv:1202.4339*, 2012.

[30] T.J. Mitchell and J.J. Beauchamp. Bayesian Variable Selection in Linear Regression. *Journal of the American Statistical Association*, 83:1023–1032, 1988.

[31] M. Osborne, B. Presnell, and B. Turlach. A new approach to variable selection in least squares problems. *Journal of Numerical Analysis*, 20(3):389–403, 2000.

[32] M. Y. Park, T. Hastie, and R. Tibshirani. Averaged gene expressions for regression. *Biostatistics*, 8:212–227, 2007.

[33] T. Park and G. Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[34] S. Petrone, J. Rousseau, and C. Scricciolo. Bayes and empirical Bayes: do they merge? *arXiv:1204.1470v1*, 2012.

[35] S. Petry and G. Tutz. Shrinkage and variable selection by polytopes. *Technical report No. 053, Department of Statistics, University of Munich*, 2009.

[36] G. Policello. Conditional Maximum Likelihood Estimation in Gaussian mixtures. In *Statistical Distributions in Scientific Work*, volume 79, pages 111–125. Springer Netherlands, 1981.

[37] T.E. Scheetz. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429, 2006.

[38] G. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6:461–464, 1978.

[39] S.R. Searle, G. Casella, and C.E. McCulloch. *Variance components*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1992.

[40] G. A. F. Seber. *A Matrix Handbook For Statisticians*. Wiley series in probability and mathematical statistics. Wiley, 2007.

[41] D. B. Sharma, H. D. Bondell, and H. H. Zhang. Consistent Group Identification and Variable Selection in Regression with Correlated Predictors. *Journal of Computational and Graphical Statistics. In Press.*, 22(2):319–340, 2013.

[42] Y. She. *Sparse Regression with Exact Clustering*. Stanford University, 2008.

[43] X. Shen and H. Huang. Grouping pursuit in regression. *Journal of American Statistical Association*, 105:727–739, 2010.

[44] C. Stein. Estimation of the Mean of a Multivariate Normal Distribution. *Annals of Statististics*, 9:1135–1151, 1981.

[45] R. Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[46] C. G. Wei and M.A. Tanner. A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 85:699–704, 1990.

[47] F.J. Wenjiang. Penalized Regressions: The Bridge versus the Lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998.

[48] D.M. Witten, A. Shojaie, and F. Zhang. The Cluster Elastic Net for High-Dimensional Regression With Unknown Variable Grouping. *Technometrics. In Press.*, 2013.

[49] L. Yengo, J. Jacques, and C. Biernacki. Variable clustering in high dimensional linear regression models. *Journal de la Societe Francaise de Statistique. In Press.*, 2013.

[50] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[51] H. Zhou, D.H. Alexander, M.E. Sehl, J.S. Sinsheimer, E.M. Sobel, and K. Lange. Common SNPs explain a large proportion of the heritability for human height. In *Pacific Symposium on Biocomputing'11*, volume 42, pages 106–117, 2011.

[52] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[53] H. Zou, T. Hastie, and R. Tibshirani. On the degrees of freedom of the lasso. *Annals of Statistics*, 35(5):2173–2192, 2007.