Model-based pretreatment for correlated datasets, Application to linear regression and missing values. Real datasets from steel industry

Clément THERY

August 15, 2014

To my sons

Résumé

Les travaux effectués durant cette thèse ont pour but de pouvoir pallier le problème des corrélations au sein des bases de données, particulièrement fréquentes dans le cadre industriel. Une modélisation explicite des corrélations par un système de sous-régressions entre covariables permet de pointer les sources des corrélations et d'isoler certaines variables redondantes.

Il en découle une pré-sélection de variables nettement moins corrélées sans perte significative d'information et avec un fort potentiel explicatif (la pré-selection elle-même est expliquée par la structure de sous-régression qui est simple à comprendre car uniquement constituée de modèles linéaires).

Un algorithme de recherche de structure de sous-régressions est proposé, basé sur un modèle génératif complet sur les données et utilisant une chaîne MCMC (Monte-Carlo Markov Chain). Ce prétraitement est utilisé pour la régression linéaire à des fins illustratives mais ne dépend pas de la variable réponse et peut donc être utilisé de manière générale pour toute problématique de corrélations.

Par suite, le modèle génératif complet peut être utilisé pour gérer d'éventuelles valeurs manquantes dans les données, tant pour la recherche de structure que pour de l'imputation multiple préalable à l'utilisation de méthodes classiques incompatibles avec la présence de valeurs manquantes. Cela permet également d'estimer les valeurs manquantes et à terme de fournir un estimateur de la variance de leur estimation. Encore une fois, la régression linéaire vient illustrer l'apport de la méthode qui reste cependant générique et applicable à d'autres contextes tels que le clustering.

Enfin, un estimateur plug-in pour la régression linéaire est proposé pour ré-injecter les variables redondantes de manière séquentielle et donc utiliser toute l'information sans souffrir des corrélations entre covariables.

Tout au long de ces travaux, l'accent est mis principalement sur l'interprétabilité des résultats en raison du caractère industriel du financement de cette thèse.

Le package R intitulé `CorReg`, disponible sur le CRAN[1] sous licence CeCILL[2], implémente les méthodes développées durant cette thèse.

Mots clés: Prétraitement, Régression, Corrélations, Valeurs manquantes, MCMC, modèle génératif, Critère Bayésien, sélection de variable, méthode séquentielle, graphs.

[1] http://cran.r-project.org
[2] http://www.cecill.info

Abstract

This thesis was motivated by correlation issues in real datasets, in particular industrial datasets. The main idea stands in explicit modeling of the correlations between covariates by a structure of sub-regression, that simply is a system of linear regression between the covariates. It points out redundant covariates that can be deleted in a pre-selection step to improve matrix conditonning without significant loss of information and with strong explicative potential because this pre-selection is explained by the structure of sub-regression, itself easy to interpret.

An algorithm to find the sub-regression structure inherent to the dataset is provided, based on full generative model and using Monte-Carlo Markov Chain (MCMC) method. This pretreatment is then illustrated on linear regression to show its efficiency but does not depend on a response variable and thus can be used in a more general way with any correlated datasets.

The generative model defined here allows to manage missing values both during the MCMC and then for imputation (for example multiple imputation) to be able to use classical methods that are not compatible with missing datasets. Missing values can be imputed with a confidence interval to show estimation accuracy. Once again, linear regression is used to illustrate the benefits of this method but it remains a pretreatment that can be used in other contexts, like clustering and so on.

Finally a plug-in estimator is defined to get back the redundant covariates sequentially. Then all the covariates are used but the sequential approach act as a protection against correlations.

The industrial motivation of this work define interpretation as a stronghold at each step. The R package `CorReg`, is on CRAN[3] now under CeCILL[4] license. It implements the methods created during this thesis.

Keywords: Pretreatment, Regression, Correlations, Missing values, MCMC, generative model, Bayesian Criterion, variable selection, plug-in method,...

[3]http://cran.r-project.org
[4]http://www.cecill.info

Acknowledgments

I want to thank ArcelorMittal for the funding of this thesis, the opportunity to make this thesis with real datasets and the confidence in this work that has lead me to be recruited since may but let me work in Villeneuve d'Ascq this last year to reduce road time.

But this work would not have been possible without the help of Gaétan LORIDANT, my hierarchic superior and friend who has convinced ArcelorMittal to fund this thesis and helped me in this work by a strong moral support and spending a great amount of time with me to find the good direction between academic and industrial needs with some technical help when he could. I would not have made all this work without him.

I also want to thank Christophe BIERNACKI, my academic director who accepted to lead this work even if the subject was not coming from the university and even if I won't pursue a research career. He also has spent a lot of time on this thesis with patience and has trust in the new method enough to share it with others, and it really means a lot to me.

The last year I have worked mostly in the INRIA center with MΘdal team, especially those from the "bureau 106 " who helped me to put `CorReg` on CRAN (in particular Quentin GRIMONPREZ) and those who had already submitted something on CRAN know that it is not always fun to achieve this goal. They also help me in this last and rude year just by their presence, giving me the courage to go further.

I finally want to thank my family. My wife who has to support most of the charge of the family in top of her work and to let me work during the holidays and my three sons, Nathan, Louis and Thibault who have been kind with her even if they rarely saw their father during the week. I love them and am thankful for their comprehension.

Contents

1 Résumé substantiel en français **8**
1.1 Position du problème . . . 8
1.2 Modélisation explicite des corrélations . . . 8
1.3 Modèle marginal . . . 9
1.4 Notion de prétraitement . . . 9
1.5 Estimation de la structure . . . 10
1.6 Relaxation des contraintes et nouveau critère . . . 10
1.7 Résultats . . . 11
1.8 Au-delà du marginal avec le plug-in . . . 11
1.9 Les valeurs manquantes . . . 12

2 The industrial context **13**
2.1 Steelmaking process . . . 13
2.2 Impact of the industrial context . . . 14

3 State of the art **15**
3.1 Linear regression . . . 15
3.1.1 Industrial context . . . 15
3.2 Ordinary least squares and associated problems . . . 16
3.3 Penalized models . . . 19
3.3.1 Ridge regression . . . 19
3.3.2 LASSO: Least Absolute Shrinkage and Selection Operator . . . 20
3.3.3 Least Angle Regression . . . 21
3.3.4 Elasticnet . . . 23
3.3.5 OSCAR: Octogonal Shrinkage and Clustering Algorithm for Regression . . . 24
3.4 Modeling the parameters . . . 25
3.4.1 CLERE: CLusterwise Effect REgression . . . 25
3.4.2 Spike and Slab . . . 25
3.5 Miscellaneous . . . 26
3.5.1 Principal Component Regression and Partial Least Squares Regression . . . 26
3.5.2 Sliced Inverse Regression . . . 26
3.5.3 Classification and Regression Trees (CART) . . . 27
3.5.4 Neural networks . . . 28
3.5.5 Bayesian networks . . . 28
3.6 Choice of model . . . 29
3.6.1 Cross validation . . . 29
3.6.2 Bayesian Information Criterion . . . 29
3.6.3 Stepwise . . . 29
3.7 MCMC . . . 30
3.8 Gibbs . . . 30

3.9 EM . . . 31
3.10 Industrial tools . . . 31
3.11 Multiple Equations . . . 32
3.11.1 Simultaneous Equation Model (SEM) and Path Analysis . . . 32
3.11.2 SUR: Seemingly Unrelated Regression . . . 32
3.11.3 SPRING: Structured selection of Primordial Relationships IN the General linear model . . . 32
3.11.4 Selvarclust: Linear regression within covariates for clustering . . . 32

I Pretreatment for correlations 34

4 Decorrelating covariates by a generative model 35
4.1 Our proposal: modelisation of the correlations . . . 35
4.2 Graph theory . . . 36
4.3 A by-product model: marginal regression with decorrelated covariates . . . 37
4.4 Strategy of use: pre-treatment before classical estimation/selection methods . . . 38

5 Numerical results with a known structure 39
5.1 Illustration of the tradeoff conveyed by the pre-treatment . . . 39
5.2 Observed MSE comparison . . . 40
5.2.1 On the running example . . . 40
5.2.2 On more complex datasets . . . 47

6 Estimation of the Structure of subregression by MCMC 53
6.1 Bayesian approach . . . 53
6.2 Sub-regression structure in details . . . 53
6.2.1 Modeling the uncorrelated covariates: a full generative approach on $P(\boldsymbol{X})$ 53
6.2.2 Identifiability of the structure . . . 54
6.2.3 Impact of the structure itself . . . 55
6.3 Sub-regression model selection . . . 55
6.3.1 Bayesian criterion for quality . . . 55
6.3.2 Penalization of the integrated likelihood by $P(S)$ . . . 55
6.3.3 Some indicators for proximity . . . 56
6.4 Neighbourhood . . . 57
6.4.1 Strategy . . . 57
6.4.2 Active relaxation of the constraints . . . 57
6.5 The walk . . . 58
6.6 Initialization . . . 58
6.6.1 Correlation-based initialization . . . 58
6.6.2 Multiple intialization . . . 59
6.7 Pruning . . . 60
6.8 The Graphical LASSO . . . 61
6.9 CorReg . . . 61

7 Numerical results on simulated datasets 63
7.1 The datasets . . . 63
7.2 Results on $\hat{S}$ . . . 63
7.3 Results on prediction . . . 65
7.3.1 $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$ . . . 65
7.3.2 $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X}^{\boldsymbol{I_f}}$ (best case for us) . . . 71

7.3.3 $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X}^{\boldsymbol{I_r}}$ (worst case for us) . . . 77
7.3.4 Robustness with non-linear case . . . 83

8 Numerical results on real datasets **84**
8.1 Quality case study . . . 84
8.2 Production case study . . . 86

II Further usage of the structure and perspectives **88**

9 Taking back the residuals **89**
9.1 The model . . . 89
9.2 Interpretation and latent variables . . . 90
9.3 Consistency . . . 90
9.4 Numerical results . . . 91
9.4.1 $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$ . . . 91
9.4.2 $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X}^{\boldsymbol{I_f}}$ . . . 97
9.4.3 $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X}^{\boldsymbol{I_r}}$ . . . 97
9.4.4 About the plug-in model . . . 97

10 Missing values **108**
10.1 Introduction . . . 108
10.2 Estimation of the sub-regressions with missing values . . . 108
10.2.1 The integrated likelihood . . . 108
10.2.2 Likelihood computation optimized . . . 111
10.2.3 Weighted penalty . . . 112
10.3 SEM . . . 112
10.3.1 Our implementation of SEM . . . 112
10.3.2 Stochastic imputation by Gibbs sampling . . . 113
10.3.3 Alternative E step . . . 114
10.4 Missing values in the main regression . . . 114
10.5 Numerical results on simulated datastets . . . 115
10.5.1 Estimation of the sub-regression coefficients . . . 115
10.5.2 Imputation by the sub-regression . . . 115
10.5.3 Multiple imputation for the main regression . . . 116
10.6 Missing values on real datasets . . . 117

11 CorReg: the concept **119**

12 Conclusion and perspectives **120**
12.1 Conclusion . . . 120
12.2 Perspective . . . 120
12.2.1 Non-linear regression . . . 120
12.2.2 Pretreatment not only for regression . . . 120
12.2.3 Improved programming . . . 121
12.2.4 Missing values in classical methods . . . 121
12.2.5 Interpretation improvements . . . 121

References **122**

Chapter 1

Résumé substantiel en français

1.1 Position du problème

La régression linéaire est l'outil de modélisation le plus classique et se résume à une équation bien connue :

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}_Y \tag{1.1}$$

où $\boldsymbol{Y}$ est la variable réponde de taille $n \times 1$ que l'on souhaite décrire à l'aide de p variables explicatives observées sur n individus et dont les valeurs sont stockées dans la matrice $\boldsymbol{X}$ de taille $n \times p$. Le vecteur $\boldsymbol{\beta}$ est le vecteur des coefficients de régression qui permet de décrire le lien linéaire entre $\boldsymbol{Y}$ et $\boldsymbol{X}$. Le vecteur $\boldsymbol{\varepsilon}_Y$ est un bruit blanc gaussien $\mathcal{N}(0, \sigma_Y^2 \boldsymbol{I}_n)$ qui représente l'inexactitude du modèle de régression.

On connaît l'estimateur sans biais de variance minimale (ESBVM) de $\boldsymbol{\beta}$ qui est obtenu par Moindres Carrés Ordinaires (MCO) selon la formule :

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y} \tag{1.2}$$

Cet estimateur nécessite l'inversion de la matrice $(\boldsymbol{X}'\boldsymbol{X})$ qui est mal conditionnée si les variables explicatives sont corrélées entre elles. Ce mauvais conditionnement nuit à la qualité de l'estimation et vient impacter la variance de l'estimateur comme le montre la formule :

$$\mathrm{Var}_{\boldsymbol{X}}(\hat{\boldsymbol{\beta}}) = \sigma_Y^2(\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{1.3}$$

C'est cette situation problèmatique que nous nous proposons d'améliorer.

1.2 Modélisation explicite des corrélations

Le mauvais conditionnement de la matrice provient de la quasi-singularité (parfois singularité numérique) de celle-ci quand les colonnes de $\boldsymbol{X}$ sont presque linéairement dépendantes. Cette quasi dépendance linéaire peut être elle aussi modélisée par régression lineaire. On se propose donc de considérer notre problématique comme l'existence d'un modèle de sous-régressions au sein des variables explicatives avec certaines des variables expliquées par d'autres, formant ainsi une partition des p variables en 2 blocs:

$$\boldsymbol{X}^{I_r} = \boldsymbol{X}^{I_f}\boldsymbol{\alpha} + \boldsymbol{\varepsilon} \quad \text{(regression multiple multivariée)} \tag{1.4}$$

Avec $\boldsymbol{X}^{I_r}$ sous-matrice de p_r variables redondantes dans $\boldsymbol{X}$. Nous supposons alors l'indépendance entre les variables non redondantes qui définissent $\boldsymbol{X}^{I_f}$. La matrice $\boldsymbol{\alpha}$ de taille $(p - p_r) \times (p_r)$ a

pour colonnes les vecteurs $\boldsymbol{\alpha}_j$ qui contiennent les coefficients de régression associées à la j^{ieme} colonne de $\boldsymbol{X}$.

La figure 3.3 illustre la déterioration de l'estimation quand les sous-régressions deviennent trop fortes (R^2 proche de 1) pour différentes valeurs de n sur un modèle composé de 5 variables dont 4 gaussiennes centrées réduites *i.i.d.* $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_5$ et une variable redondante $\boldsymbol{x}_3 = \boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{\varepsilon}_3$ avec $\boldsymbol{\varepsilon}_3 \sim \mathcal{N}(\boldsymbol{0}, \sigma_3^2 \boldsymbol{I}_n)$. Deux régressions principales en $\boldsymbol{Y}$ ont été testées avec $\boldsymbol{\beta} = (1, 1, 1, 1, 1)$ and $\sigma_Y \in \{10, 20\}$. Le conditionnement de $(\boldsymbol{X}'\boldsymbol{X})$ se déteriore donc quand σ_3 diminue.

1.3 Modèle marginal

Le fait de modéliser explicitement les corrélations entre les covariables nous permet de réécrire le modèle de régression principal. On peut en effet substituer les variables redondantes par leur sous-régression, ce qui revient à intégrer la régression sur $\boldsymbol{X}^{I_r}$ sachant la structure de sous-régressions.

$$
\begin{aligned}
P(\boldsymbol{Y}|\boldsymbol{X}^{I_f}) &= \int_{\boldsymbol{X}^{I_r}} P(\boldsymbol{Y}|\boldsymbol{X}^{I_r}, \boldsymbol{X}^{I_f}) P(\boldsymbol{X}^{I_r}|\boldsymbol{X}^{I_f}) d\boldsymbol{X} && (1.5)\\
\boldsymbol{Y}_{|\boldsymbol{X}^{I_f}, S} &= \boldsymbol{X}^{I_f}(\boldsymbol{\beta}_{I_f} + \sum_{j \in I_r} \beta_j \boldsymbol{\alpha}_j) + \sum_{j \in I_r} \beta_j \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_Y && (1.6)\\
&= \boldsymbol{X}^{I_f} \boldsymbol{\beta}^*_{I_f} + \boldsymbol{\varepsilon}^*_Y && (1.7)
\end{aligned}
$$

on se retrouve donc avec un modèle marginal plus parsimonieux, sans biais mais avec une variance accrue. Cet accroissement de la variance est proportionnel à $\boldsymbol{\varepsilon}$ qui est la matrice des résidus des sous-régressions. Plus les sous-régressions sont fortes plus cette variance sera donc faible. Tout le principe du modèle marginal repose sur le compromis entre l'amélioration du conditionnement de $(\boldsymbol{X}'\boldsymbol{X})$ par suppression des variables redondantes et aussi la réduction de la dimension (le modèle marginal ne nécessite que l'inversion de $(\boldsymbol{X}^{I'_f}\boldsymbol{X}^{I_f})$) face au léger accroissement de la variance issu de la marginalisation.
On va donc comparer

$$
\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y} \text{ au modèle marginal} && (1.8)\\
\hat{\boldsymbol{\beta}}^* &= (\boldsymbol{X}^{I'_f}\boldsymbol{X}^{I_f})^{-1}\boldsymbol{X}^{I'_f}\boldsymbol{Y} && (1.9)
\end{aligned}
$$

Les deux modèles sont sans biais et de dimension différente, on compare donc leurs erreurs moyennes quadratiques respectives (MSE):

$$
\begin{aligned}
E[MSE(\hat{\boldsymbol{\beta}}|\boldsymbol{X})] &= \sigma_Y^2 \operatorname{Tr}(\boldsymbol{X}'\boldsymbol{X})^{-1} && (1.10)\\
E[MSE(\hat{\boldsymbol{\beta}}^*|\boldsymbol{X}^*)] &= (\sigma_Y^2 + \sum_{j \in I_r} \sigma_j^2 \beta_j^2) \operatorname{Tr}(\boldsymbol{X}^{I'_f}\boldsymbol{X}^{I_f})^{-1} && (1.11)
\end{aligned}
$$

La figure 5.1 compare ces deux erreurs pour différentes valeurs des paramètres, montrant la nette amélioration rendue possible par la marginalisation.

1.4 Notion de prétraitement

Le modèle marginal peut être interprété comme étant un prétraitement par sélection de variable puisqu'on se ramène à un modèle de régression linéaire classique pour lequel n'importe quel estimateur peut être utilisé. Cela fait de ce modèle un outil générique. La préselection permet de cibler des variables qui n'interviendront pas dans le modèle final sans pour autant être

indépendante de $\boldsymbol{Y}$ donc le modèle final est parsimonieux mais ne fausse pas l'interprétation. L'estimation de $\boldsymbol{\beta}^*$ peut ensuite se faire en utilisant une quelconque méthode de sélection de variables pour éliminer les variables qui, elles, sont indépendantes de $\boldsymbol{Y}$.
On obtient donc deux types de 0 : ceux de la marginalisation qui pointent les variables redondantes et ceux de sélection qui viennent dans un second temps et point les variables indépendantes. L'interprétation est donc enrichie par rapport à une méthode de sélection classique qui fournirait le même modèle final. Or, le contexte industriel de ces travaux rend indispensable d'avoir une bonne qualité d'interprétation. L'objectif est donc atteint pour ce point précis. Notre modèle marginal est un outil de décorrélation de variables par préselection.

1.5 Estimation de la structure

La raison d'être de notre modèle marginal est la fragilité des méthodes de régression face à des covariables fortement corrélées. Il serait donc vain d'essayer les sous-régression en estimant les modèles de régression de chaque variable en fonction de toutes les autres. Pour cette raison, nous avons établi un algorithme MCMC pour trouver le meilleur modèle de sous-régression. L'idée consiste à voir la structure de sous-régression comme un paramètre binaire, une matrice binaire creuse pour être plus précis. Cette matrice $\boldsymbol{G}$ de taille $p \times p$ correspond à une matrice d'adjacence qui indique les liaisons entre covariables de la manière suivante : $\boldsymbol{G}_{i,j} = 1$ si, et seulement si $\boldsymbol{X}^j$ est expliqué par $\boldsymbol{X}^i$.

Chaque étape $(q+1)$ de l'algorithme propose de garder la structure $\boldsymbol{G}^{(q)}$ en cours ou bien de bouger vers une structure candidate qui diffère de $\boldsymbol{G}^{(q)}$ en un unique point. Ainsi, selon les cas, les candidats vont allonger ou réduire des sous-régression, les supprimer ou les créer.
Pour pouvoir trancher entre plusieurs candidats, nous avons besoin d'une fonction coût qui soit capable de comparer des modèles avec des nombres distincts de sous-régressions. Nous définissons alors un modèle génératif complet sur $\boldsymbol{X}$ qui complète le modèle de sous-régressions en établissant des modèles de mélanges gaussiens indépendants pour les variables de $\boldsymbol{X}^{I_f}$. Une fois ce modèle génératif établi, nous pouvons utiliser le critère BIC pour comparer les différents modèles et conduire chaque étape de la chaîne MCMC par un tirage aléatoire pondéré par les écarts entre les BIC des différents modèles proposés (dont le modèle en cours). L'algorithme continue ainsi sa marche et fournit à l'utilisateur le modèle rencontré (qu'il ait été choisi ou non) qui a le BIC le plus faible.
La chaîne MCMC est conditionnée par le critère de partitionnement : les variables expliquées ne doivent en expliquer aucune autre ($\boldsymbol{X}^{I_r} \cap \boldsymbol{X}^{I_f} = \emptyset$). Chaque modèle réalisable peut être entièrement construit ou déconstruit pendant la marche aléatoire donc l'algorithme suit une chaîne de Markov régulière. Ainsi, il est certain que, asymptotiquement l'algorithme trouve le modèle ayant le meilleur BIC.

1.6 Relaxation des contraintes et nouveau critère

Pour améliorer la mélangeance de l'algorithme et donc sa vitesse de convergence, on peut jouer avec la contrainte de partitionnement par une méthode de relxation semblable à un recuit simulé. Quand une structure candidate n'est pas réalisable (ne produit pas de partition), on peut la modifier en d'autres endroits pour la rendre réalisable. Il suffit de suivre les formules

suivantes pour une modification en (i, j):

$$\boldsymbol{G}_{i,j}^{(q+1)} = 1 - \boldsymbol{G}_{i,j}^{(q)} \quad (1.12)$$

$$\boldsymbol{G}_{.,i}^{(q+1)} = (\boldsymbol{G}_{i,j}^{(q)})\boldsymbol{G}_{.,i}^{(q)} \text{ Si } i \text{ explique } j \text{ alors personne ne peut plus expliquer } i \quad (1.13)$$

$$\boldsymbol{G}_{j,.}^{(q+1)} = (\boldsymbol{G}_{i,j}^{(q)})\boldsymbol{G}_{j,.}^{(q)} \text{ Si } i \text{ explique } j \text{ alors } j \text{ ne peut plus expliquer personne} \quad (1.14)$$

Cette méthode de relaxation permet de sortir rapidement des extrema locaux et améliore donc significativement l'efficacité de l'algorithme.

Mais il reste un problème. Le nombre de modèles envisageable est considérable et le critère BIC ne tient pas compte de cette quantité, menant à des modèles trop complexes. On lui ajoute donc une pénalité qui tient compte du nombre de modèles réalisables pour pénaliser plus lourdement les modèles complexes. De manière générale quand on estime la vraisemblance d'une structure S dans une base de données $\boldsymbol{X}$, BIC est utilisé comme approximation pour $P(S|\boldsymbol{X}) \propto P(\boldsymbol{X}|S)P(S)$ car $P(S)$ est considéré comme suivant une loi uniforme. On s'appuie donc sur une loi uniforme hiérarchique sur $P(S) = P(I_f|\boldsymbol{p}_f, I_r, p_r)P(\boldsymbol{p}_f|I_r, p_r)P(I_r|p_r)P(p_r)$ pour ajouter une pénalité supplémentaire aux structures complexes (même probabilité globale pour un plus grand nombre de structures donc chaque structure devient moins probable). Avec I_r l'ensemble des variables redondantes, p_r le nombre de sous-régressions, $\boldsymbol{p}_f$ le vecteur des complexités des sous-régressions et enfin I_f l'ensemble des ensembles des variables explicatives dans les sous-régressions. On note BIC_+ ce nouveau critère car il ne modifie pas BIC (on conserve ainsi ses propriétés) mais ajoute simplement une pénalisation.
Ces deux outils viennent améliorer l'efficacité de l'algorithme sans paramètre utilisateur supplémentaire à optimiser. Tout reste naturel et intuitif pour une meilleur automatisation.

1.7 Résultats

La méthode a été testée sur données simulées puis réelles, montrant l'efficacité du modèle marginal s'appuyant sur la vraie structure de sous-régressions (section 5.2), l'efficacité de l'algorithme de recherche de structure (section 7.2), et l'efficacité du modèle marginal s'appuyant sur la structure estimée (section 7.3 et chapitre 8). Le bilan est très positif comme le montrent les graphiques de ces différentes sections.

1.8 Au-delà du marginal avec le plug-in

Une première perspective a été développée pour compléter le modèle marginal. Il s'agit d'un modèle plug-in permettant de ré-injecter les variables redondantes après estimation du modèle marginal pour obtenir un modèle utilisant toutes les variables mais protégé des corrélations par l'estimation séquentielle.

$$\boldsymbol{X}^{I_r} = \boldsymbol{X}^{I_f}\boldsymbol{\alpha} + \boldsymbol{\varepsilon} \quad (1.15)$$

$$\boldsymbol{Y} = \boldsymbol{X}^{I_r}\underbrace{(\boldsymbol{\beta}_{I_r} + \boldsymbol{\alpha}\boldsymbol{\beta}_{I_f})}_{\boldsymbol{\beta}^*} + \boldsymbol{\varepsilon}\boldsymbol{\beta}_{I_r} + \boldsymbol{\varepsilon}_Y \quad (1.16)$$

Ce qui donne donc

$$\boldsymbol{Y} - \boldsymbol{X}^{I_r}\boldsymbol{\beta}^* = \boldsymbol{\varepsilon}\boldsymbol{\beta}_{I_r} + \boldsymbol{\varepsilon}_Y \text{ avec} \quad (1.17)$$

$$\boldsymbol{\varepsilon} = \boldsymbol{X}^{I_r} - \boldsymbol{X}^{I_f}\boldsymbol{\alpha} \quad (1.18)$$

D'où le modèle plugin

$$\underbrace{\boldsymbol{Y} - \boldsymbol{X}^{I_r}\hat{\boldsymbol{\beta}}^*}_{\tilde{\boldsymbol{Y}}} = \underbrace{(\boldsymbol{X}^{I_r} - \boldsymbol{X}^{I_f}\hat{\boldsymbol{\alpha}})}_{\tilde{\boldsymbol{X}}}\boldsymbol{\beta}_{I_r} + \boldsymbol{\varepsilon}_Y \tag{1.19}$$

$$\tag{1.20}$$

qui permet d'estimer dans un second temps $\boldsymbol{\beta}_{I_r}$ par un modèle classique de régression linéaire en s'appuyant sur l'estimation de $\boldsymbol{\beta}^*$ issue du modèle marginal et sur $\hat{\boldsymbol{\alpha}}$. Le modèle plug-in réduit le bruit de la régression

$$\boldsymbol{Y} = \boldsymbol{X}^{I_r}\hat{\boldsymbol{\beta}}^* + \hat{\boldsymbol{\varepsilon}}\hat{\boldsymbol{\beta}}_{I_r} + \boldsymbol{\varepsilon}_Y \tag{1.21}$$

and simple étape d'identification (sans estimation supplémentaire) permet de retouver le vrai modèle:

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{X}^{I_r}(\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\alpha}}\hat{\boldsymbol{\beta}}_{I_r}) + \boldsymbol{X}^{I_r}\hat{\boldsymbol{\beta}}_{I_r} + \boldsymbol{\varepsilon}_Y & (1.22)\\ &= \boldsymbol{X}^{I_f}\hat{\boldsymbol{\beta}}_{I_f} + \boldsymbol{X}^{I_r}\hat{\boldsymbol{\beta}}_{I_r} + \boldsymbol{\varepsilon}_Y & (1.23) \end{aligned}$$

La figure 9.1 montre l'efficacié du modèloe plug-in et son champ d'application : les cas avec assez de correlations pour que les méthodes classiques appliquées à $\boldsymbol{X}$ soient handicapées mais pas assez de corrélations pour que le retrait des variables redondantes se fasse sans perte siginificative d'information.

1.9 Les valeurs manquantes

Une seconde perspective a été entamée concernant les valeurs manquantes. Le fait de disposer d'un modèle génératif complet sur $\boldsymbol{X}$ avec modélisation explicite des dépendances permet en effet de gérer les valeurs manquantes. Tout d'abord, l'estimation de $\boldsymbol{\alpha}$ peut se faire sur les données observées en intégrant sur les données manquantes. On peut alors utiliser un algorithme de type EM (expectation Maximization) pour estimer $\hat{\boldsymbol{\alpha}}$.
En pratique, l'étape E n'est pas systématiquement explicite et peut nécessiter de faire appel à une variante de EM : l'aglorithme SEM (pour Stochastic EM) qui remplace l'étape E par une étape stochastique d'imputation des valeurs manquantes, par exemple en utilisant un échantillonneur de Gibbs.
Cet algorithme de Gibbs peut alors être utilisé pour faire de l'imputation multiple sur les valeurs manquantes en s'appuyant sur le $\hat{\boldsymbol{\alpha}}$ issu du SEM. Comme cette imputation tient compte des corrélations entre les variables, elle est plus précise qu'une simple imputation par la moyenne. Un avantage de l'imputation multiple est que l'on peut avoir une idée de la robustesse des imputations en regardant simplement la variance des valeurs imputées, ce qui donne une information de fiabilité sur les imputations. Encore une fois, on y gagne en qualité d'interprétation.

Chapter 2

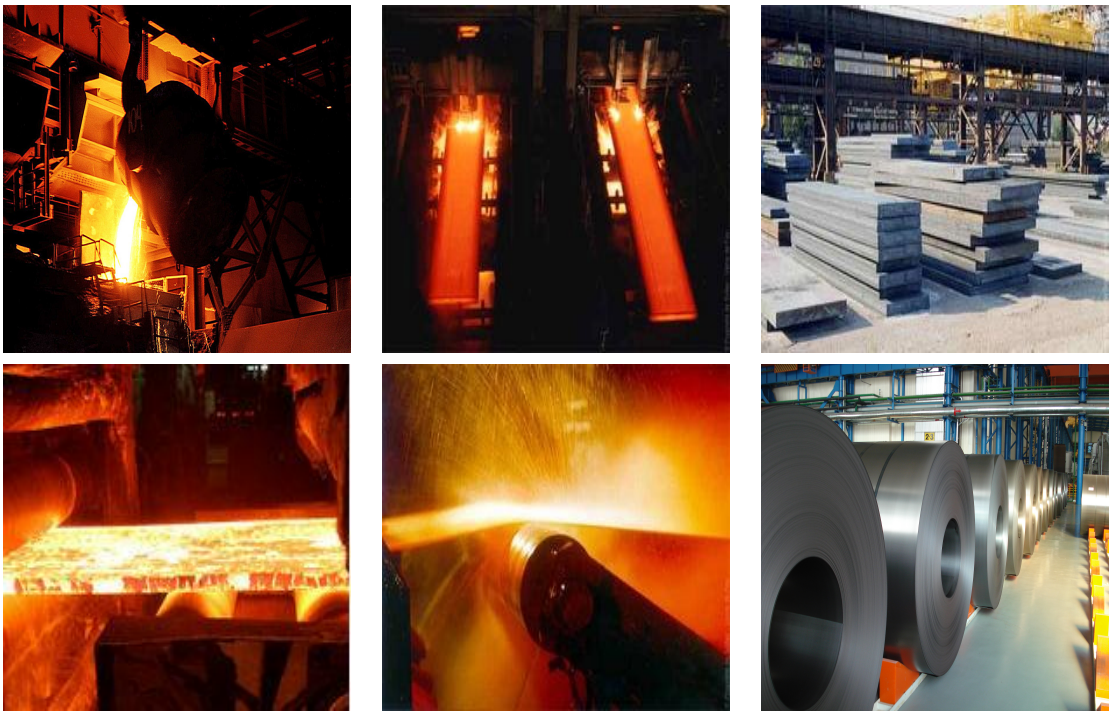
The industrial context

Abstract: Ce chapître explique les contraintes industrielles qui ont orienté les travaux pour répondre aux demandes d'ArcelorMittal qui est le commanditaire de ces travaux de recherche.

2.1 Steelmaking process

This work takes place in a steel industry context, fund by ArcelorMittal, the world leading company in steelmaking. Steelmaking starts from raw materials to give highly specific products found.

We first melt down a mix of iron ore and cock to obtain cast iron that is then transformed in steel by addiction of pure oxygen to reduce the quantity of carbon. Liquid steel is then refreshed in a mold (continuous casting) to obtain steel slabs (nearly 20 tons each). Cold slabs are then warmed to be pressed in a hat rolling mill to obtain coils of steel. If the final product requires a thinner steel, coils pass through a cold rolling mill. Each step of this process involves a whole manufactory and the whole process can take several weeks. The most sensitive products are the thinner and sometimes defects are generated by small inclusion in the steel down to the dozen of microns. So even if quality is evaluated at each step of the process, some defects are only found when the whole process is finished even if the origin comes from the first part of this process. So we have hundreds of parameters to analyse.



Steelmaking is always evolving and we are now able to produce steel that is thinner and stronger at the same time. Steel is completely recyclable so we will always be able to produce it. This quickly evolving industry is associated to a lot of research in metallurgy but also need adapted statistical tools. That is why this thesis has been made.

2.2 Impact of the industrial context

The main objective is to be able to solve quality crisis when they occur. In such a case, a new type of unknown quality issue is observed and we may have no idea of its origin. The defects, even generated at the beginning of the process, are often detected in its last part. The steel-making process includes several sub-process, each implying a whole plant. Thus we have many covariates and no a priori on the relevant ones. Moreover, the values of each covariates essentially depends on the characteristics of the final product, and many physical laws and tuning models are implied in the process. Therefore the covariates are highly correlated. We have several constraints :

- To be able to predict the defect and stop the process as early as possible to gain time (and money)
- To be able to understand the origin of the defect to try to optimize the process
- To be able to find parameters that can be changed because the objective is not only to understand but to correct the problematic part of the process.
- It also must be fast and automatic (without any a priori).

We will see in the state of the art that correlations are a real issue and that the number of variables increases the problem. The stakes are very high because of the high productivity of the steel plants but also because steel making is now well-known and optimized thus new defects only appears on innovative steels with high value. Any improvement on such crisis can have important impact on the market shares and when the customer is implied, each day won by the automation of the data mining process can lead to a gain of hundreds of thousands of euros, sometimes more. So we really need a kind of automatic method, able to manage the correlations without any a priori and giving an easily understandable and flexible model.

Chapter 3

State of the art

Abstract: Rapide aperçu de ce qui existe déjà pour tenter de répondre à notre problématique. La plupart des outils présentés le sont plus en détail dans le livre de Hastie, Tibshirani et Friedman : *"The Elements of Statistical Learning: Data Mining, Inference, and Prediction"* accessible gratuitement sur le web[1].

3.1 Linear regression

3.1.1 Industrial context

Industrial context is often poor in statistical background and the stakes are frequently very high in terms of financial impact. These two points give strong constraints because methods used has to be accessible for non-statistician in a minimum amount of time and results obtained have to be clearly interpreted (no black-box) because if industrial experts don't understand the result, they will not trust it and then they will not use it. So a powerful tool without interpretation becomes kind of useless in such a context.

Every engineer, even non-statistician use frequently linear regression to seek relationship between some covariates. It is easy to understand, fast to do, it can be done directly in Microsoft Excel that remains the most used software in industry and the software used by engineers to open most of the datasets.

Regression appears to be the basis of industrial statistics so we have chosen to work in

[1] `http://web.stanford.edu/~hastie/local.ftp/Springer/OLD/ESLII_print4.pdf`

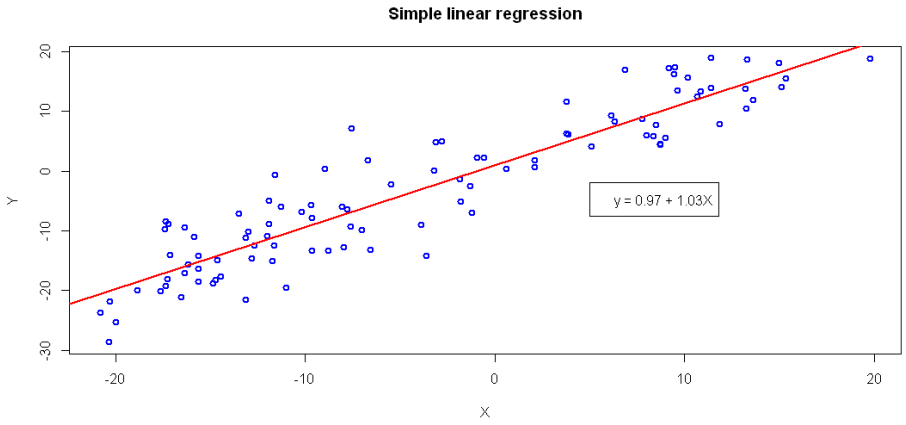


Figure 3.1: An example of simple linear regression

this way. As of 2014 Google Scholar proposes more than 3.8 millions of papers related to regression and many of them were cited several thousands times. It is an old strategy well known and with many derivative (as we will see in the followings) and can be generalized [Kiebel and Holmes, 2003, Wickens, 2004, Nelder and Baker, 1972, McCullagh and Nelder, 1989] . It's simplicity facilitates a wide spread usage in industry and other fields of application. It is also a powerful tool for interpretation allowing to know with precision the positive or negative impact of each covariate on the response variable.

More complex situations can be described by evolved forms of linear regression like the hierarchical linear model [Raudenbush, 2002, Woltman et al., 2012] or multilevel regression [Moerbeek et al., 2003, Maas and Hox, 2004, Hox, 1998] that allows to consider effects of the covariates on nested sub-populations in the dataset. It is like using interactions but with a proper modeling that improves interpretation. It is not really a linear model because it is not linear in $\boldsymbol{X}$ but can be seen as a basis expansion using new covariates (the interactions) composed by the product of some of the original covariates.

Notations: In the following we note classical (respectively L_2, L_1, L_∞) norms: $\| \boldsymbol{\beta} \|_2^2 = \sum_{i=1}^p (\beta_i)^2$, $\| \boldsymbol{\beta} \|_1 = \sum_{i=1}^p |\beta_i|$ and $\| \boldsymbol{\beta} \|_\infty = \max(|\beta_1|, \ldots, |\beta_p|)$. Vectors are in bold characters.

3.2 Ordinary least squares and associated problems

We note the linear regression model:

$$\boldsymbol{Y}_{|\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.1}$$

where $\boldsymbol{X}$ is the $n \times p$ matrix of the explicative variables,$\boldsymbol{Y}$ the $n \times 1$ response vector and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_Y^2 \boldsymbol{I}_n)$ the noise of the regression, with $\boldsymbol{I}_n$ the n-sized identity matrix and $\sigma_Y > 0$. The $p \times 1$ vector $\boldsymbol{\beta}$ is the vector of the coefficients of the regression, that can be estimated by $\hat{\boldsymbol{\beta}}$ with Ordinary Least Squares (OLS):

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1} \boldsymbol{X}'\boldsymbol{Y} \tag{3.2}$$

with variance matrix

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}_{OLS}) = \sigma_Y^2 (\boldsymbol{X}'\boldsymbol{X})^{-1} \tag{3.3}$$

and without any bias [Saporta, 2006, Dodge and Rousson, 2004]. In fact it is the Best Linear Unbiased Estimator (BLUE). The theoretical MSE is given by

$$E[\mathrm{MSE}(\hat{\boldsymbol{\beta}}_{OLS}|\boldsymbol{X})] = 0 + \sigma_Y^2 \,\mathrm{Tr}((\boldsymbol{X}'\boldsymbol{X})^{-1}) \tag{3.4}$$

Equation 3.1 has no intercept but can be generalized by adding to $\boldsymbol{X}$ a first column full of 1. So we don't consider the intercept to simplify notations. In practice, an intercept is added by default.

Ordinary Least Squares find a $p-$dimensional hyperplane that minimizes the distance with each individual $(\boldsymbol{X}_i, Y_i)$. It can be written

$$\hat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}} \left\{ \| \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \|_2^2 \right\} \tag{3.5}$$

So estimation of $\boldsymbol{Y}$ by OLS can be viewed as a projection onto the linear space spanned by the regressors $\boldsymbol{X}$ as shown in figure 3.2.

Estimation of $\boldsymbol{\beta}$ requires the inversion of $\boldsymbol{X}'\boldsymbol{X}$ which will be ill-conditioned or even singular if some covariates depend linearly from each other. For a fixed number n of individuals, conditioning of $\boldsymbol{X}'\boldsymbol{X}$ get worse based on two aspects:

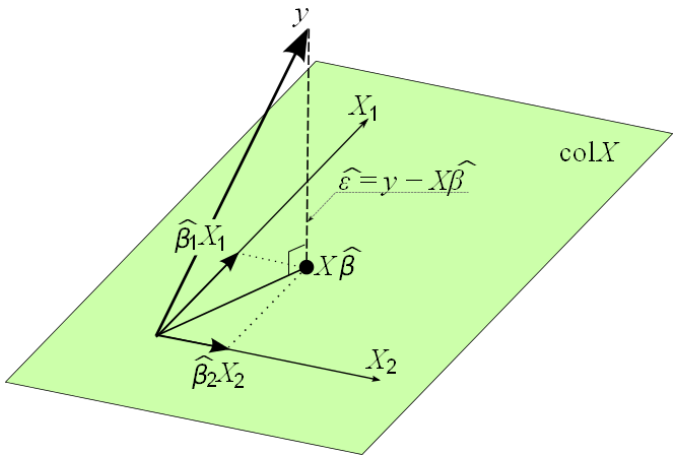


Figure 3.2: Multiple linear regression with Ordinary Least Squares. Public domain image.

- The dimension p (number of covariates) of the model (the more covariates you have the greater variance you get)
- The correlations within the covariates: strongly correlated covariates give bad-conditioning and increase variance of the estimators .

When correlations between covariates are strong, the matrix to invert is ill-conditioned and the variance increases, giving unstable and unusable estimator [Hoerl and Kennard, 1970]. Another problem is that matrix inversion requires to have more individuals than covariates ($n \geq p$). When matrices are not invertible, classical packages like the function lm of R base package [R Core Team, 2014] use the Moore-Penrose pseudoinverse [Penrose, 1955] to generalize OLS.

Last but not least, Ordinary Least Squares is unbiased but if some β_i are null (irrelevant covariates) the corresponding $\hat{\beta}_i$ will only asymptotically tend to 0 so the number of covariates in the estimated model remains p. This is a major issue because we are searching for a statistical tool able to work without a priori on a big dataset containing many irrelevant datasets. Pointing out some relevant covariate and how they impact the response really is the main goal here. We will need a variable selection method one moment or another. It could be as a pre-treatment, during coefficient estimation or by post-treatment.

Running example: we look at a simple case with $p = 5$ variables defined by four independent scaled Gaussian $\mathcal{N}(0, 1)$ named $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_5$ and $\boldsymbol{x}_3 = \boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{\varepsilon}_3$ where $\boldsymbol{\varepsilon}_3 \sim \mathcal{N}(\boldsymbol{0}, \sigma_3^2 \boldsymbol{I}_n)$. We also define two *scenarii* for $\boldsymbol{Y}$ with $\boldsymbol{\beta} = (1, 1, 1, 1, 1)$ and $\sigma_Y \in \{10, 20\}$. So there is no intercept (can be seen as a null intercept). It is clear that $\boldsymbol{X}'\boldsymbol{X}$ will become more ill-conditioned as σ_3 gets smaller.

Figure 3.3 shows the theoretical MSE obtained on $\hat{\boldsymbol{\beta}}$ with OLS. These results are based on equation 3.3, we show the mean obtained after 100 experiences computed on our running example.

The R^2 stands for:

$$R^2 = 1 - \frac{\sum_i (y_i - x_i\beta)^2}{\sum_i (y_i - \bar{y})^2} \tag{3.6}$$

where $\bar{\boldsymbol{Y}} = \frac{1}{n} \sum_{i=1}^{n} y_i$

Many other estimation methods were created to obtain better estimations by playing on the bias/variance tradeoff or by making additionnal hypothesis. To have an easier comparison, we also look at the empiric MSE obtained on $\hat{\boldsymbol{\beta}}$.

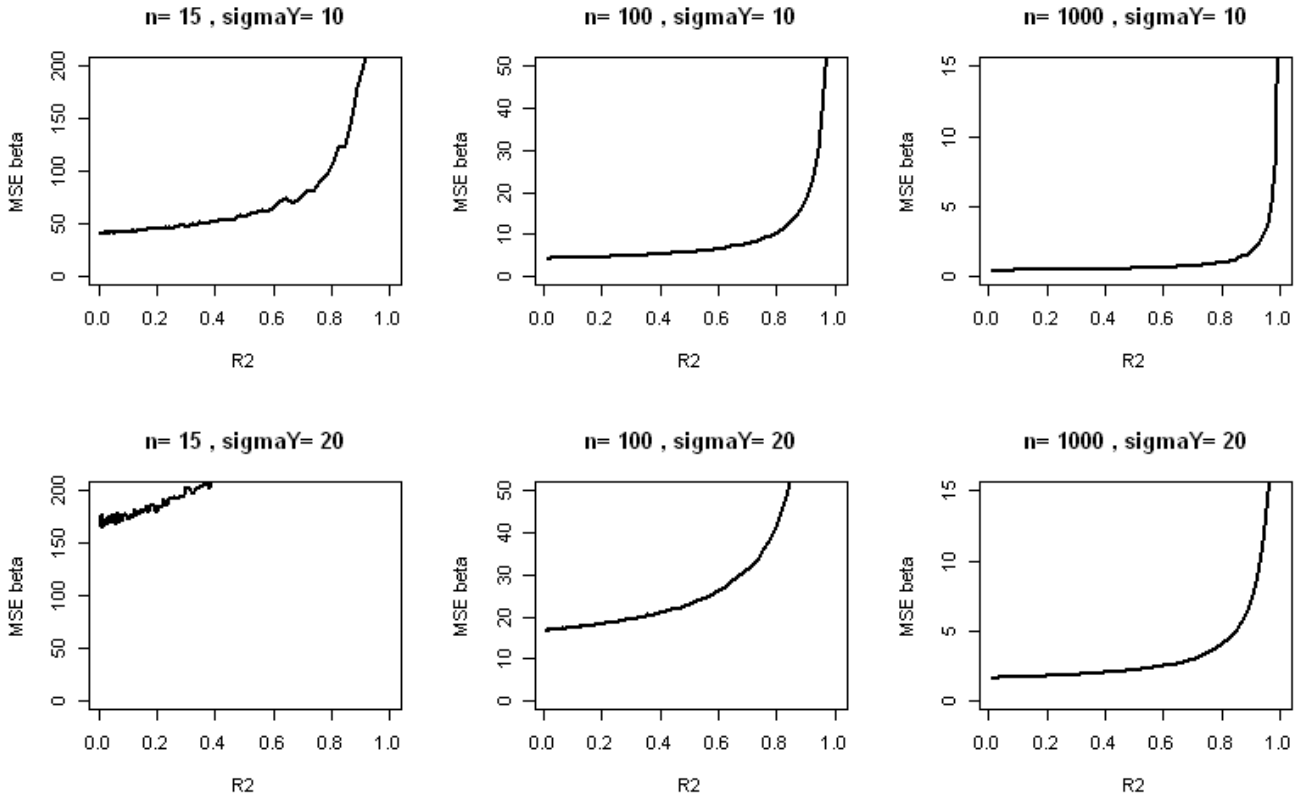


Figure 3.3: Evolution of theoretical Mean Squared error on $\hat{\boldsymbol{\beta}}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

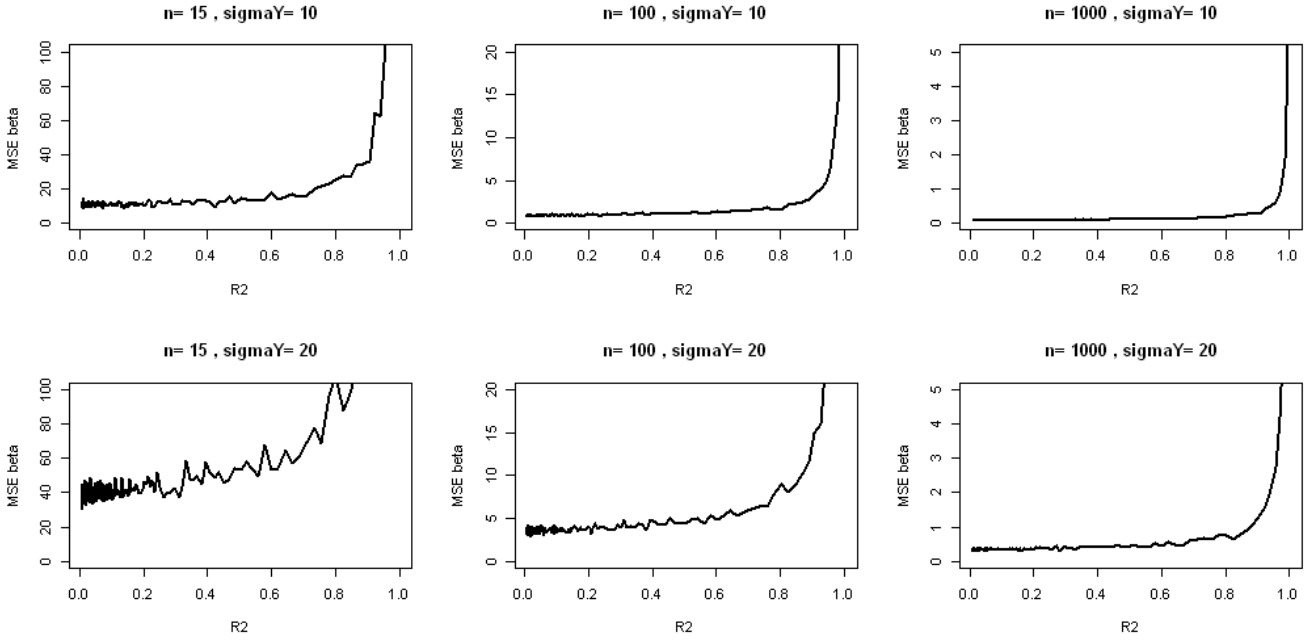


Figure 3.4: Evolution of observed Mean Squared error on $\hat{\boldsymbol{\beta}}_{OLS}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

We first observe that real results are better from expected (Figure 3.4). This comes from usage of QR decomposition to inverse matrices, that are less impacted by ill-conditioned matrices [Bulirsch and Stoer, 2002]. But the correlations issue remains and so do the impact of n and σ_Y. Our package `CorReg` also uses this decomposition.

3.3 Penalized models

We have seen that OLS is the Best linear Unbiased Estimator for $\hat{\boldsymbol{\beta}}$, meaning that it has the minimum variance. But it remains possible to play with the bias/variance tradeoff to reduce the variance by adding some bias. The underlying idea is that a small bias and a small variance could be preferred to a huge variance without bias. Many methods do this by a penalization on $\hat{\boldsymbol{\beta}}$. Some of them propose an effective variable selection.

3.3.1 Ridge regression

Ridge regression [Hoerl and Kennard, 1970, Marquardt and Snee, 1975] proposes a biased estimator for $\boldsymbol{\beta}$ that can be written in terms of a parametric L_2 penalty:

$$\hat{\boldsymbol{\beta}} = \text{argmin}\left\{\| \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \|_2^2\right\} \text{ subject to } \| \boldsymbol{\beta} \|_2^2 \leq \lambda \text{ with } \lambda > 0 \tag{3.7}$$

But this penalty is not guided by the correlations. It introduce an additional parameter λ to choose for the whole dataset whereas correlations may concern only some of the covariates with several intensities.

The solution of the ridge regression is given by

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}'\boldsymbol{X} - \lambda \boldsymbol{I}_n\right)^{-1} \boldsymbol{X}'\boldsymbol{Y} \tag{3.8}$$

and we see in this equation that a global modification of $\boldsymbol{X}'\boldsymbol{X}$ is done for a given λ. Methods does exist to automatically choose a good value for λ [Cule and De Iorio, 2013, Er et al., 2013] and a R package called `ridge` is on CRAN [Cule, 2014]. We have computed the same experiment as in previous figure but with the `ridge` package instead of OLS. It is clear that the ridge regression is efficient in variance reduction (it is what it is built for).

Moreover, like OLS, coefficients tend to 0 but don't reach 0 so it gives difficult interpretations for large values of p. Ridge regression is efficient to improve conditioning of the estimator but gives no clue to the origin of ill-conditioning and keep irrelevant covariates. It remains a good candidate for prediction-only studies. But our industrial context makes necessary to have a variable selection method.

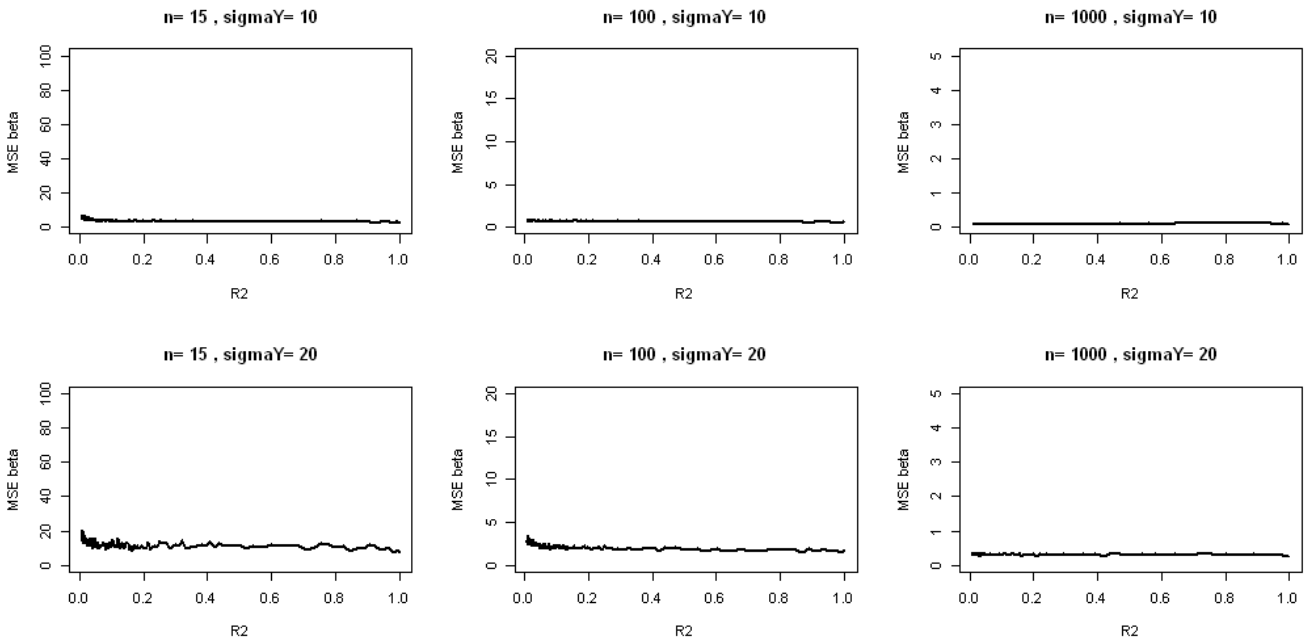


Figure 3.5: Evolution of observed Mean Squared error on $\hat{\boldsymbol{\beta}}_{ridge}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

3.3.2 LASSO: Least Absolute Shrinkage and Selection Operator

The Least Absolute Shrinkage and Selection Operator (LASSO [Tibshirani, 1996, Tibshirani et al.,]) consists in a shrinkage of the regression coefficients based on a λ parametric L_1 penalty to obtain zeros in $\hat{\boldsymbol{\beta}}$ instead of the L_2 penalty of the ridge regression:

$$\hat{\boldsymbol{\beta}} = \text{argmin}\left\{\| \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \|_2^2\right\} \text{ subject to } \| \boldsymbol{\beta} \|_1 \leq \lambda \text{ with } \lambda > 0 \tag{3.9}$$

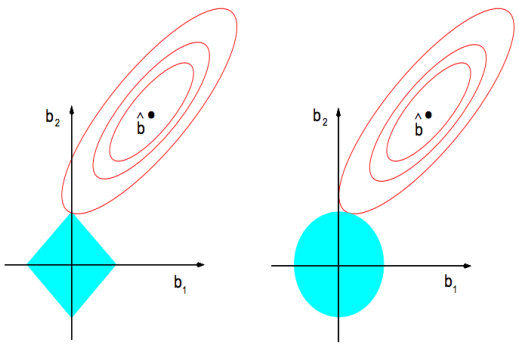


Figure 3.6: Geometric view of the Penalty for the LASSO (left) compared to ridge regression (right) as shown in the book from Hastie [Hastie et al., 2009]

Figure 3.6 show the contour of error (red) and constraint function (blue) for both lasso (left) and ridge regression (right). We see that the optimum will be found on an axis for the lasso because its constraint zone is a polyhedron whose vertices are on the axis but not for the ridge regression. Here the axis stands for the regression coefficients.

Here again we have to choose a value for λ. The Least Angle Regression (LAR [Efron et al., 2004]) algorithm offers a very efficient way to obtain the whole LASSO path. But like the ridge regression, the penalty does not distinguish correlated and independent covariates so there is no guarantee to have less correlated covariates. In practice, we know that the LASSO faces

consistency issues when confronted to correlated covariates [Zhao and Yu, 2006]. When two covariates are correlated, it tends to keep only one of them. For example, if two covariates are equal and have the same effect, the LASSO will keep only one of them. As explained earlier, variable selection is a real stake for us but is necessary to have a good interpretation. The LASSO does not distinguish a covariate not selected because it is totally redundant with another that was selected from an irrelevant covariate. And that is a problem. This consistency issue is illustrated in section 9.3.

Some recent variants of the LASSO do exist for the choice of the penalization coefficient like the adaptive LASSO [Zou, 2006] or the random LASSO [Wang et al., 2011]. But the consistency issues remains because it is still the same model. Only the choices of λ differ.

It is notable that the main goal of the LASSO is to select some covariate, thus the penalization is just a mean to achieve selection. But estimation of $\hat{\boldsymbol{\beta}}$ can be improved by a second estimation with OLS based only on selected covariates [Zhang and Shen, 2010].

3.3.3 Least Angle Regression

The least Angle Regression algorithm solves the LASSO problem. It requires only the same order of magnitude of computational effort as OLS applied to the full set of covariates. The idea is to start with all coefficients to zero and then grow them beginning with the most correlated with the response variable until another variable is equally correlated with the residual. So it is a progressive growth of the coefficient leading to reduce the residual. It finishes with the Ordinary Least Squares solution (on the right in figure 3.8). We then have a list of models with several numbers of non-zero coefficients and can choose between them with cross-validation for example.

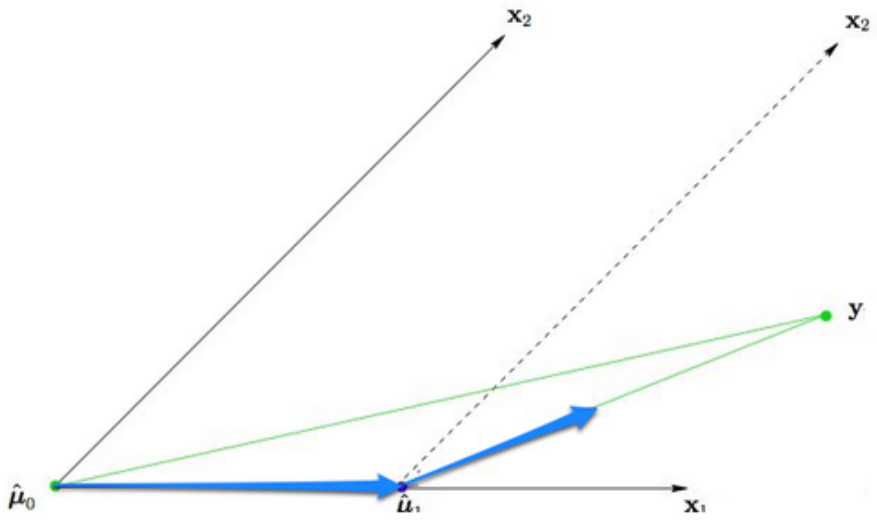


Figure 3.7: The geometry of Least Angle Regression

Results obtained with the package `lars` for R, included in `CorReg`. Figure 3.9 shows that strong correlations make the MSE explode even with lar.

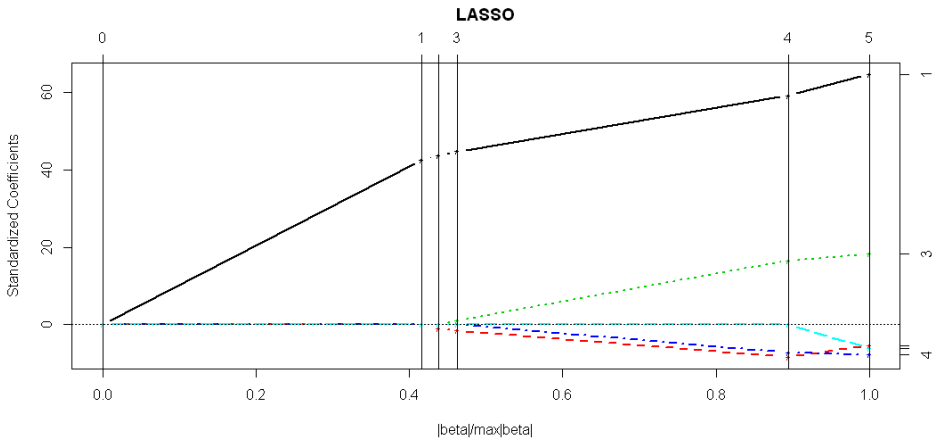


Figure 3.8: The LASSO path computed by lars

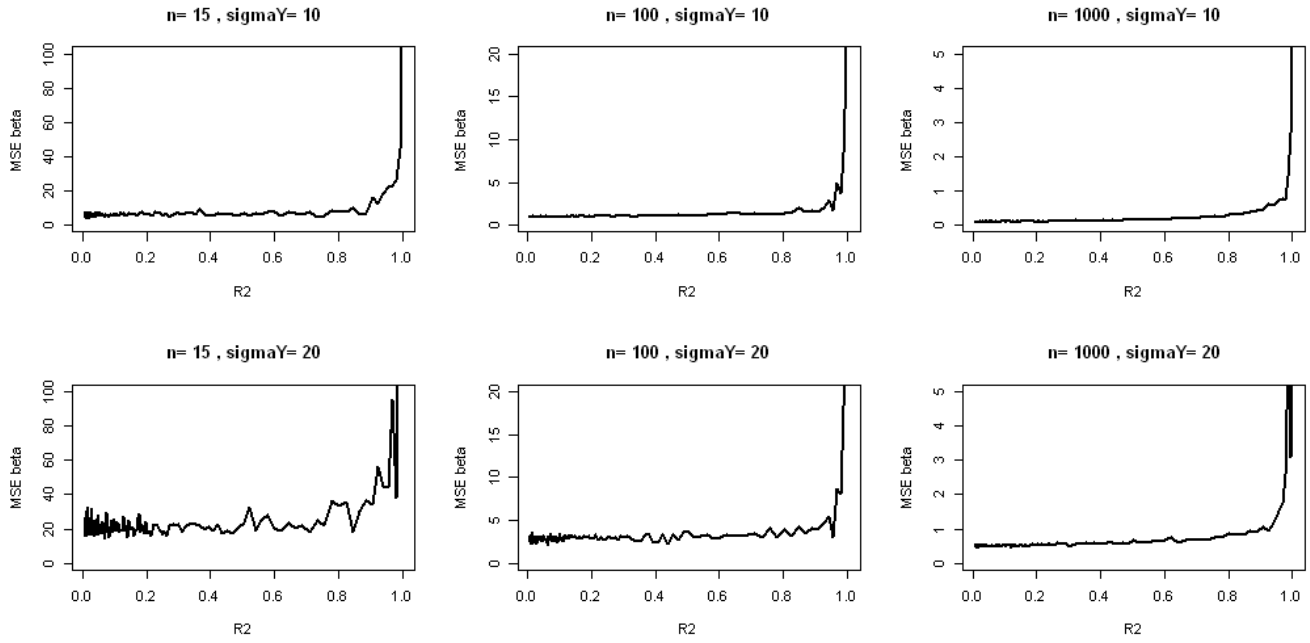


Figure 3.9: Evolution of observed Mean Squared error on $\hat{\beta}_{lar}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

3.3.4 Elasticnet

Elastic net [Zou and Hastie, 2005] is a method developed to be a compromise between Ridge regression and the LASSO by mixing both L_1 and L_2 penalties:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (1+\lambda_2)\,\text{argmin}\left\{\| \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \|_2^2\right\} \text{ subject to} \\ &\quad (1-\alpha) \| \boldsymbol{\beta} \|_1 + \alpha \| \boldsymbol{\beta} \|_2^2 \leq t \text{ for some } t \end{aligned} \tag{3.10}$$

where $\alpha = \frac{\lambda_2}{(\lambda_1+\lambda_2)}$.

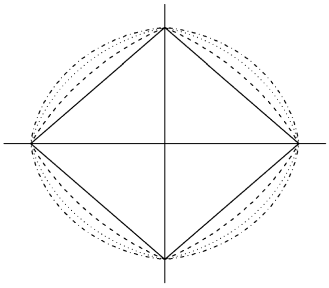


Figure 3.10: Geometric view of the Penalty for elasticnet

But it is based on the grouping effect so correlated covariates get similar coefficients and are selected together whereas LASSO will choose between one of them and will then obtain same predictions with a more parsimonious model. Once again, nothing specifically aims to reduce the correlations.

Results obtained with the package `elasticnet` for R.

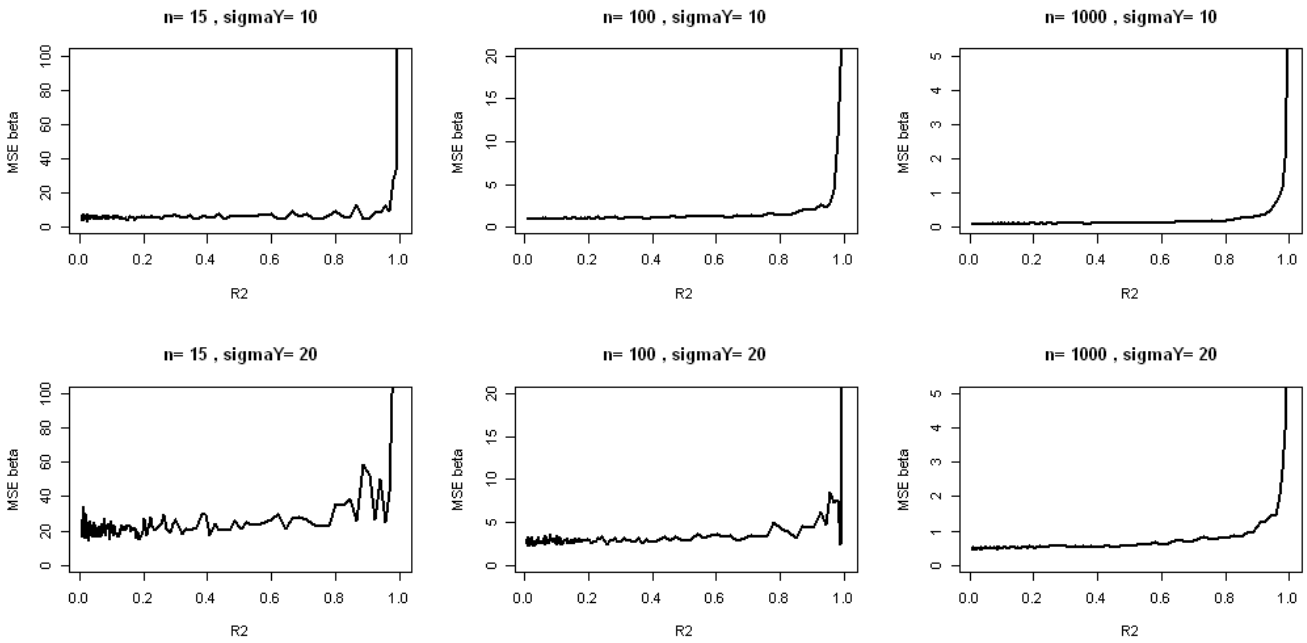


Figure 3.11: Evolution of observed Mean Squared error on $\hat{\boldsymbol{\beta}}_{elasticnet}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

3.3.5 OSCAR: Octogonal Shrinkage and Clustering Algorithm for Regression

Like elasticnet, OSCAR [Bondell and Reich, 2008] uses combination of two norms for its penalty. Here the objective is to group covariates with the same effect (by a pairwise L_∞ norm) and give them exactly the same coefficient (reducing the dimension) with a simultaneous variable selection (implied by the L_1 norm).

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \parallel \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \parallel_2^2 \text{ subject to } \sum_{j=1}^{p} |\beta_j| + c \sum_{j<k} \max(|\beta_j|, |\beta_k|) \leq \lambda \tag{3.11}$$

But OSCAR depends on two tuning parameters: c anf λ. For a fixed c the λ can be found by the LAR algorithm but c still has to be found "by hand" comparing final models for many values of c.

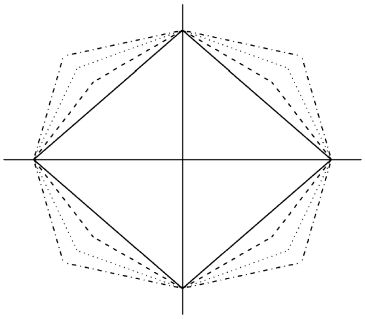


Figure 3.12: Geometric view of the Penalty for OSCAR

Figure 3.12 show the geometric interpretation of the penalty. It follows the same principle as the LASSO with supplementary vertices in the four quarters to obtain equal values for the β_j. So the estimator will give both zero coefficients and equal coefficients that can be

grouped for interpretation and correspond to a dimension reduction. So two covariates with a similar effect may obtain the same estimated coefficient. But correlations are only implicitly taken into account and only pairwise. It lacks of an efficient algorithm (to find c) and need a supplementary study to interpret the groups found.

3.4 Modeling the parameters

3.4.1 CLERE: CLusterwise Effect REgression

The CLusterwise Effect REgression (CLERE [Yengo et al., 2012]) describes the β_j no longer as fixed effect parameters but as unobserved independant random variables with grouped β_j following a Gaussian Mixture distribution.

The idea is to hope that the model have a small number of groups of covariates and that the mixture will have few enough components to have a number of parameters to estimate significantly lower than p. In such a case, it improves interpretation and ability to yeld reliable prediction with a smaller variance on $\hat{\boldsymbol{\beta}}$. A package `clere` for R does exist on CRAN and is included in `CorReg`. But We have to choose the maximum number of components g and have no method to choose this value. Yengo recommends to use $g = 5$ in our case. It could be interpreted as the possibility to have a group of irrelevant covariates and groups with small or big values (both positives or negatives). The package is able to choose automatically the best number of components between 1 and g based on a BIC criterion but setting $g = p$ gives over-fitting.
Here again, it has no specific protection against or specific model for correlations.

3.4.2 Spike and Slab

Spike and Slab variable selection [Ishwaran and Rao, 2005] also relies on Gaussian mixture (the spike and the slab) hypothesis for the β_j and gives a subset of covariates (not grouped) on which to compute OLS but has no specific protection against correlations issues. The β_j are supposed to come from a mixture distribution as shown in figure 3.13. It allows to have some coefficients set exactly to zero after some draws.

The package `spikeslab` for R on CRAN is also included in `CorReg`.

Modeling the parameters implies to have no exact value to give to the coefficient and it is not really user-friendly, especially in our industrial context. However, these two methods can be used for variable selection in the `correg` function using the parameter `select="clere"` or `select="spikeslab"`.

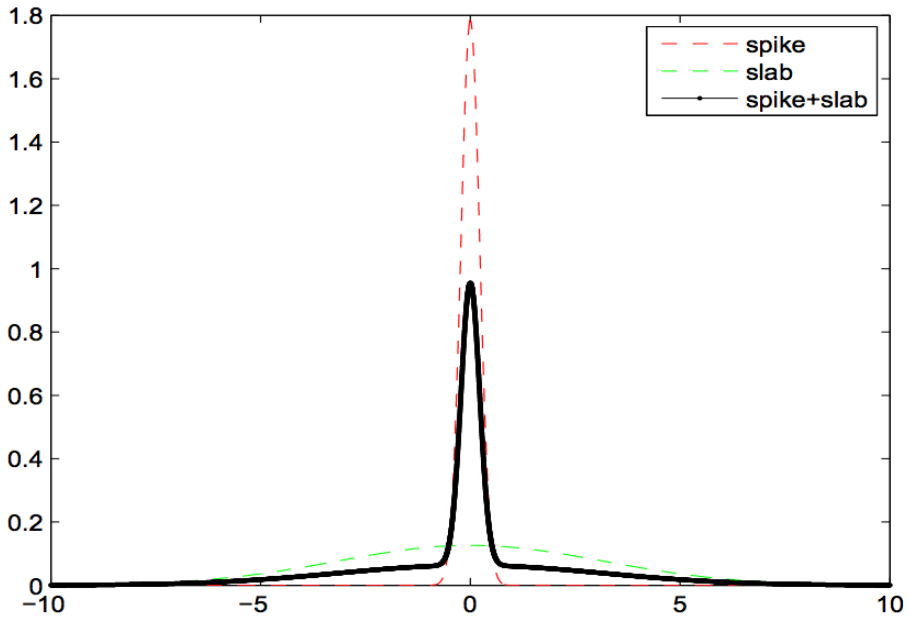


Figure 3.13:

3.5 Miscellaneous

3.5.1 Principal Component Regression and Partial Least Squares Regression

Principal Component Regression [Jackson, 2005] consists in using the axis from the Principal Component Analysis (PCA) of $\boldsymbol{X}$ instead of $\boldsymbol{X}$ itself. Then we have orthogonal covariates. Dimension reduction is done by keeping only the $M \leq p$ first components of the PCA. Because the axis are linear combination of the original covariates we can then express the model in terms of coefficients of the $\boldsymbol{X}^j$.

Principal Component Regression requires to choose M the number of axis to keep. Finally, even if dimension reduction is effective when $M < p$ each axis depends on all original covariates so it does not select any covariates and that is also a problem for interpretation. We have to choose arbitrary how to interpret each axis and how many covariates really explain each of them. So it is not really satisfying in our industrial context. Principal Component method can be seen as a truncation method whereas the ridge regression is a shrinkage method.

Partial Least Square Regression [Abdi, 2003, Geladi and Kowalski, 1986] also relies on a combination of the columns of $\boldsymbol{X}$ but this combination depends on $\boldsymbol{Y}$. It uses scalar products of the covariates with the response variable

It seems to be influenced more by the variance of the covariates than by their correlation with the response variable so results tends to be similar to Principal Component Regression. The R package `pls` on CRAN computes both Principal Component Regression and Partial Least Squares Regression.

3.5.2 Sliced Inverse Regression

Sliced Inverse Regression is a semi-parametric approach that could be seen as easier to interpret than general non-parametric regression [Eubank, 1999, Hardle, 1990] that are too complex to interpret for our industrial context.

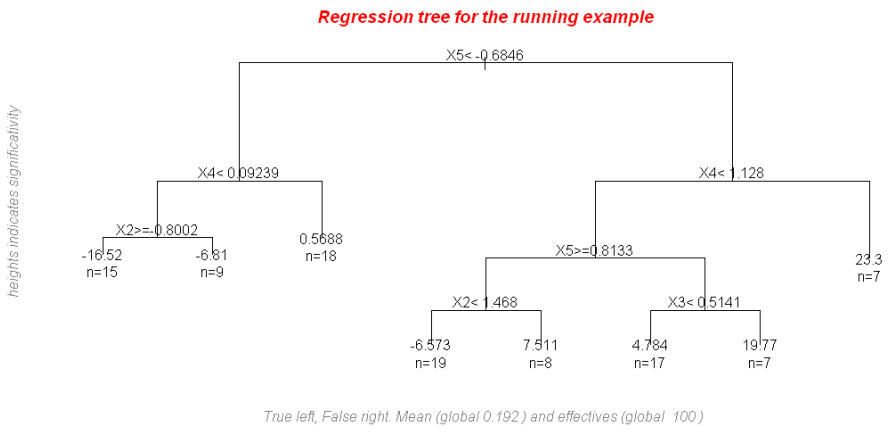


Figure 3.14: Regression tree obtain with the package `CorReg` (graphical layer on top of the `rpart` package) on the running example.

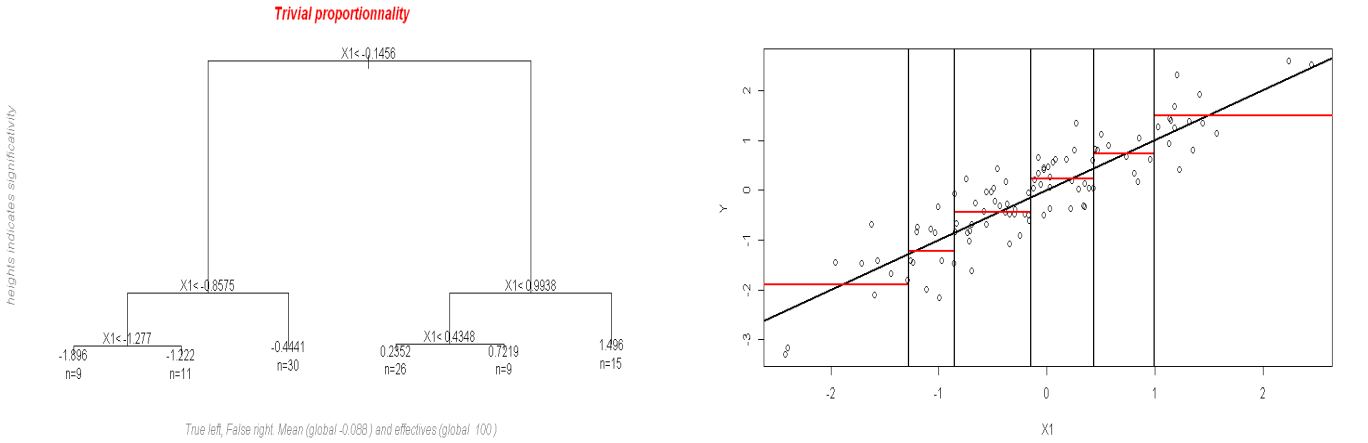
The main idea is to compute the inverse regression $P(\boldsymbol{X}|\boldsymbol{Y})$ instead of $P(\boldsymbol{Y}|\boldsymbol{X})$ to perform a weighted principal component analysis, with which we can identify the effective dimension reducing directions [Li, 1991]. But it needs strong hypothesis on the distribution of $\boldsymbol{X}$ (elliptic hypothesis), even if it is possible to neglect this hypothesis [Saracco et al., 1999] and see what happens. This method has been rejected because it was not sufficient in terms of ease of use (during and after estimation) for non-statisticians due to the semi-parametric aspect and the elliptic hypothesis is compromised on datasets with hundreds of covariates. It is part of the dimension reduction package `dr` on CRAN.

3.5.3 Classification and Regression Trees (CART)

Classification And Regression Trees (CART) [Breiman, 1984] are extremely simple to use and interpret, can work simultaneously with quantitative and qualitative covariates and are very fast to compute. They consist in recursive partitioning of the sample according to binary rules on the covariate (only one at a time) to obtain a hierarchy defined by simple rules and containing pure leaves (same value). It is followed by a pruning method to obtain leaves that are quite homogeneous and described with simple rules.

CART is implemented in the package `rpart` for R. Our `CorReg` package offers a function to compute and plot the tree in one command with a subtitle to explain how to read the tree and global statistics on the dataset. But it is not convenient for linear regression problems as we see in figure 3.15 because a same variable will be used several times and the tree will fail to give a simple interpretation as " $\boldsymbol{Y}$ and $\boldsymbol{X}_1$ are proportional ". Trivial case : $\boldsymbol{Y} = \boldsymbol{X}_1 + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_n)$ with $\sigma = 0.5$.
So CART will be used as a complementary tool for datasets with both quantitative and qualitative covariates or when the dependence between $\boldsymbol{Y}$ and $\boldsymbol{X}$ is not linear. We will focus our research on linear models with quantitative only variables.

More details in the book from Hastie [Hastie et al., 2009]. The main issues are the lack of smoothness (prediction function with jumps) and especially instability because of the hierarchical partitioning. Modifying only one value in the dataset can impact a split and then change the range of possible splits in the resulting sub-samples so if a top split is modified the tree can be widely changed. Random Forests are a way to solve this problem and can be seen as a cross-validation method for regression trees.



(a) Tree found for the trivial case

(b) True linear regression and splits obtained by the tree.

Figure 3.15: Predictive model associated to the tree (red) and true model (black)

3.5.4 Neural networks

Neural networks [Fausett, 1994] are a statistical model designed to mimic the brain and its neuronal organization. It seems to be really powerful but it has a predictive only goal and the model obtained can't be interpreted easily (and can't be interpreted at all if too complex). So it does not correspond to our needs. Interpretation remains our first goal and prediction comes far behind.

3.5.5 Bayesian networks

Bayesian networks [Heckerman et al., 1995, Jensen and Nielsen, 2007, Friedman et al., 2000] model the covariates and their conditional dependencies via a Directed Acyclic Graph (DAG). Such an orientation is very user-friendly because it is similar to the way we imagine causality. But it is only about conditional dependencies. Conditional dependencies allow to use information from independent covariates. The usual example is the case of wet grass in a garden. You don't remember if the sprinkler was on or off, you don't know if it has rain and these two facts are independent. Then you look at the grass in your neighbour's garden and it is not wet ... You will deduce that your sprinkler was on. Such conditionals dependencies are used in chapter 10 when confronted to missing values.
Bayesian networks are quite good in terms of interpretation because of that graphical and oriented representation of conditional probabilities but suffer from great dimension (combinatory issue) and require to transform the dataset arbitrary (discretisation), that imply a loss of information and usage of a priori (that is explicitly not suitable in our industrial context). The choice of the way you discretise the dataset has a great impact on the results and nothing can help if you have no a priori on the result you want to obtain. Computation relies on a table that describes all the possible combinations for each covariate. Hence it is extremely combinatory if the graph has too much edges or is not sparse enough. Moreover, you need to define the graph before computing the bayesian network and without a priori it can be challenging and time consuming.
The concept of representing dependencies with directed acyclic graph is good an we keep it in our model.

3.6 Choice of model

3.6.1 Cross validation

Time reveals quality of a model. To have an idea of the stability of a model, it is recommended to test it on a validation sample. The model parameters are estimated with a learning sample and then the model is evaluated (by its predictive MSE for example) on a validation sample to avoid over-fitting. But it is not always possible to have a validation sample and over-fitting is a real problem. A solution is to use Cross-Validation [Kohavi et al., 1995, Arlot et al., 2010]. It consists in splitting the dataset in k sub-sample (k-fold cross-validation) and then each of the k sub-samples is successively used as validation sample for the model learn with the $k-1$ remaining sub-samples. Each time a quality criterion is computed (predictive MSE or other) and then the mean of this criterion is taken as the global criterion. The global estimator is also the mean of the estimators. The two main issues are:

- How to choose k the number of sub-samples ?
- It can be time consuming as the model is estimated k times.

If $k = n$ we call this method the "leave-one-out" cross-validation. Cross-validation allows to learn the model using all individual exactly once for validation. Cross-validation is computed on each model we want to compare and just allows to avoid over-fitting when computing the comparison criterion. It is often used with the Mean Squared Error (MSE), for example on the prediction:

$$MSE_{\hat{Y}} = \| \boldsymbol{Y} - \hat{\boldsymbol{Y}} \|_2^2 \tag{3.12}$$

3.6.2 Bayesian Information Criterion

Cross-validation depends on a criterion to be used as a method to choose a model. The mean square error is not the only criterion. Probabilist criteria can also be used when we have an hypothesis on the distribution of the model studied. Such criteria can also be used without cross-validation. The Bayesian Information Criterion [Lebarbier and Mary-Huard, 2006] is a widely used criterion that relies on the likelihood of the dataset knowing the model the estimated parameters. The advantage of BIC over simple usage of the likelihood is the penalty added to take into account the numbers of parameters to estimate (complexity is then penalized) and the number of individuals in the dataset. The Akaike Information Criterion **??** known as AIC or the Risk Inflation Criterion [Foster and George, 1994] can also be used. In this work we decided to start with the BIC that is approximated by:

$$BIC = -2\ln(\hat{L}) + k\ln(n) \tag{3.13}$$

where $\hat{L}$ is the estimated likelihood, k the number of free parameters to estimate and n the number of individuals in the dataset.

3.6.3 Stepwise

Stepwise [Seber and Lee, 2012] is an algorithm to choose a subset of covariates to use in the final regression model. It is a variable selection method using OLS for estimation. It is proposed in the R package `stats` with the function `step`. The main idea is to start with first model (than can be either void or using the whole dataset or using any subset of covariates) and then to add and remove covariates step by step to improve the chosen criterion.

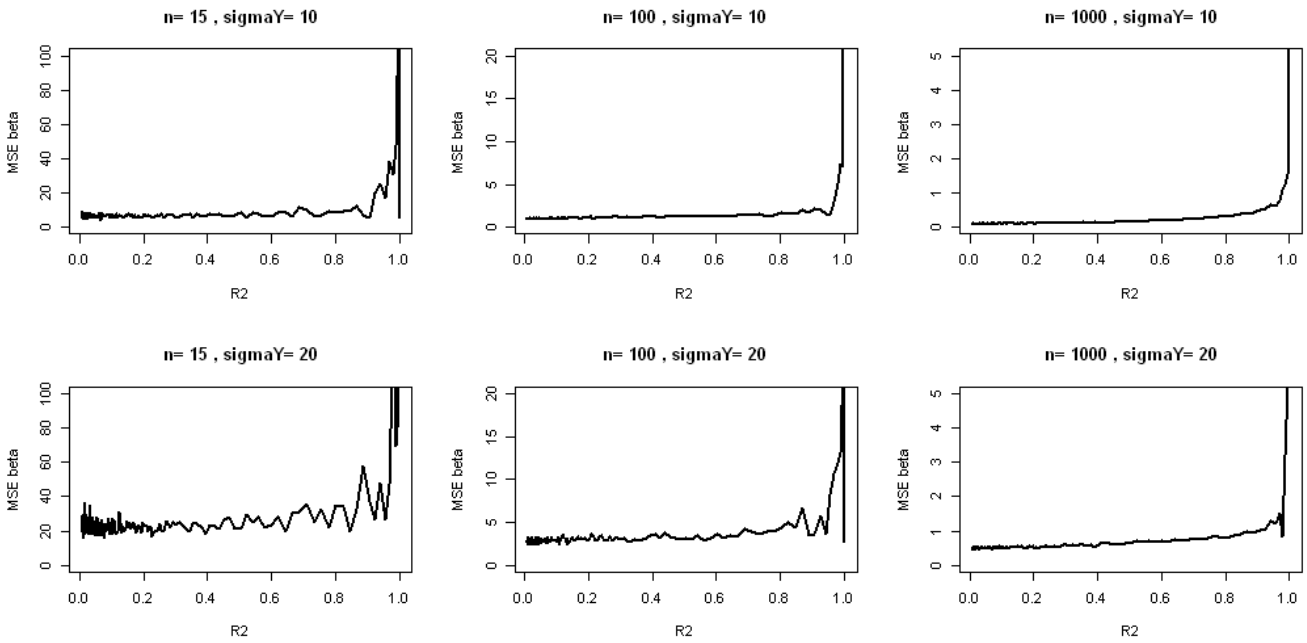


Figure 3.16: Evolution of observed Mean Squared error on $\hat{\beta}_{stepwise}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

- Starting with a void model and having only adding steps is called Forward Selection. Covariates are added by choosing first the one that improves the most the criterion and the algorithm stops when all the covariates are in or when remaining covariates does not improve the model.
- Backward Elimination is the same as Forward selection but starting with the full model and deleting at each step the covariates that improves the most the criterion once deleted.
- Bidirectional elimination is more flexible and allows to start from any model. Each proposes to add a covariate and then to delete another so it is not a hierarchical construction any more because succesives models are not necessarily nested into each other.

A critical value can be defined to stop the algorithm when improvement become too small, to try to avoid over-fitting.
Stepwise regression is subject to overfitting and the algorithm is in trouble when confronted to correlated covariates **??** giving unstable results, especially for nested strategies, just like regression trees that are unstable because of their discrete nested nature. Figure 3.16 illustrate the consequences of correlations in the dataset.

3.7 MCMC

[Gilks et al., 1996, Chib and Greenberg, 1995, Roberts and Rosenthal, 2001]

3.8 Gibbs

Gibbs sampling [Casella and George, 1992] is a special cas of Markov Chain Monte Carlo algorithm that allows to sample from a complex $p-$multivariate distribution when direct sampling is difficult. It is a randomized algorithm so each run may give distinct results. It generates a

Markov Chain that follows the desired distribution with nearby draws. It starts from an initial value $\boldsymbol{X}^{(0)}$ and then for each iteration (q) and successively each variable $x_j^{(q+1)}$ to draw, it draws from $P(x_j|x_1^{(q+1)},\ldots,x_{j-1}^{(q+1)},x_{j+1}^{(q)},\ldots,x_p^{(q)})$ using the most recent drawn values each time.
It will be used in this work for imputation of missing values in chapter 10.

3.9 EM

Sometimes the maximization of a likelihood is too complex to be done analytically. In such a case, we can use Expectation-Maximization algorithms [McLachlan and Krishnan, 2007] that maximize a likelihood by the optimization of its parameters and estimation of latent variables Z. This kind of algorithm allows to manage missing values [Dempster et al., 1977].
EM is an alternate optimization of the parameters $\boldsymbol{\theta}$ and the latent variables Z with two steps at each iteration (q):

- The Expectation step in which the expected value of the loglikelihood is computed for the current estimate of the parameters $\boldsymbol{\theta}^{(q)}$. It finds the best value of Z given $\boldsymbol{\theta}^{(q)}$.

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(q)}) = E_{Z|\boldsymbol{X},\boldsymbol{\theta}^{(q)}}[\log L(\boldsymbol{\theta};\boldsymbol{X},Z)] \tag{3.14}$$

- The Maximization step that maximizes:

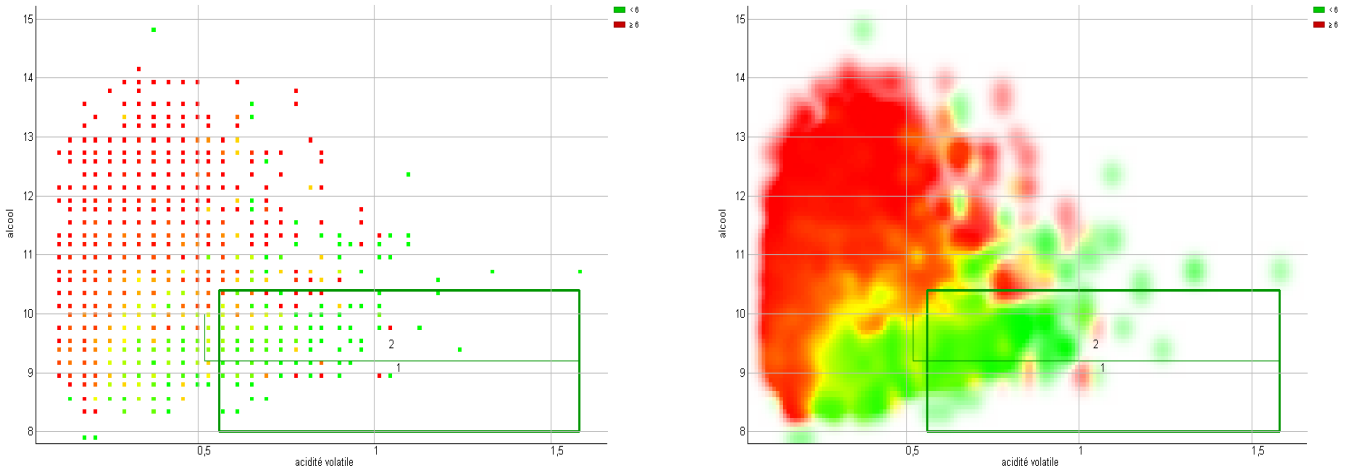
$$\boldsymbol{\theta}^{(q+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(q)}) \tag{3.15}$$

So it alternates between estimation of Z and $\boldsymbol{\theta}$ until convergence. This algorithm can be used to choose the best mixture model associated to the distribution of an observed variable for example. It requires to choose initial values for $\boldsymbol{\theta}^{(0)}$ and faces local extrema problems so it is recommended to make multiple initializations and run of the algorithm.

Some variants were developed like the Stochastic EM [Diebolt and Ip, 1996, Celeux and Diebolt, 1986] or the Classification EM [Celeux and Govaert, 1992].

3.10 Industrial tools

Our engineers are frequently contacted by software resellers. The concerned software are sold as non-statistical methods, based on rules (derivatives of decision trees but without underlying theory). Statistics are often described as too theoretical, without ergonomy and not compatible with the real life. Our goal was to demonstrate that statistics can provide efficient methods for real datasets, easy to use and understand. This was a battle against correlations but also against scepticism.
Figure 3.17 shows the kind of results proposed by rule-based softwares. It is just a partition of the sample by binary rules (like regression trees). Some blur is added to the plot to help interpretation. The algorithm used here is somewhere between exhaustive research and decision trees. It is extremely slow (research with $p > 10$ would take years) and is less efficient than regression trees. Morever, it requires to discretize the response variable to obtain "good" and "bad" values. The green rectangle is very far from the true green zone even for this toy example provided by the reseller.
But some engineers are seduced by this kind of tool. Ergonomy of use and quality of interpretation are stakes for us to make engineers use efficient methods instead of this kind of stuff.



(a) Without blur

(b) With some blur proportional to the density of points

Figure 3.17: Result on a toy example provided by a non-statistical software, similar to decision trees but less efficient and extremely slower. Colors are part of the learning set.

3.11 Multiple Equations

3.11.1 Simultaneous Equation Model (SEM) and Path Analysis

[Davidson and MacKinnon, 1993, Pearl, 2000, Pearl, 1998, Brito and Pearl, 2006]

3.11.2 SUR: Seemingly Unrelated Regression

Seemingly Unrelated Regression [Zellner, 1962] is an estimation method for Multiples equations with correlated error terms. It does not take into account correlations between the covariates but starts to estimate the system jointly instead of independent estimations. This estimation relies on Feasible Generalized Least Squares (FGLS) that depends on the variance-covariance matrix of the error terms. But when the error terms are independent or the subset of covariates are the same it is equivalent to successive independent OLS. The R package `systemfit` on CRAN computes SUR.

3.11.3 SPRING: Structured selection of Primordial Relationships IN the General linear model

[Chiquet J. and S., 2013]

3.11.4 Selvarclust: Linear regression within covariates for clustering

[Maugis et al., 2009] The idea is to allow covariates to have different roles (S, R, U, W) in a clustering context. . But:

- It is about clustering and not regression (not the same application field)
- Using stepwise-like algorithm without protection against correlations [Raftery and Dean, 2006] even it is known to be often unstable [Miller, 2002] in such a context.

In this work we propose to adapt this model for linear regression and to use it as a pretreatment on correlated covariates. We will see that, as a pretreatment it can be used then for a wide range

of statistical tools and not only linear regression. We provide a specific MCMC algorithm to find the structure between the covariates and propose two distinct models to use the structure for prediction: a marginal modal and a plug-in model.

Part I

Pretreatment for correlations

Chapter 4

Decorrelating covariates by a generative model

Abstract: Nous modélisons explicitement les corrélations entre covariables par un système de régressions linéaires entre covariables. Cela permet une meilleure compréhension des données mais aussi une préselection de variables mettant de côté les variables redondantes pour réduire fortement les corrélations tout en ne perdant que peu d'information. La préselection prend un sens particulier grâce à la structure de sous-régression qui permet de distinguer par suite les variables indépendantes de la variable réponse de celles qui sont juste redondantes mais potentiellement liées à la variable réponse.

4.1 Our proposal: modelisation of the correlations

Let $\boldsymbol{X}$ be a $n \times p$ matrix of observed covariates and $\boldsymbol{Y}$ be the $n \times 1$ matrix of the observed response variable. In the following, we note $\boldsymbol{X}^j$ the j^{th} column of $\boldsymbol{X}$ and $\boldsymbol{X}^J$ where $J = \{j_1, \ldots, j_k\}$ the $n \times k$ sub-matrix of $\boldsymbol{X}$ composed by the columns of $\boldsymbol{X}$ whose indices are in the set J.

We make the hypothesis that $\boldsymbol{X}$ can be described by a partition $\boldsymbol{X} = (\boldsymbol{X}^{I_f}, \boldsymbol{X}^{I_r})$ given by an explicit structure S where variables in the $n \times p_r$ sub-matrix $\boldsymbol{X}^{I_r}$ are redundant endogenous covariates resulting from linear sub-regressions based on $\boldsymbol{X}^{I_f}$, the $n \times (p - p_r)$ sub-matrix of free (mutually independent) exogenous covariates. So we model the correlations by $P(\boldsymbol{X}^{I_r} | \boldsymbol{X}^{I_f})$ with $\boldsymbol{X}^{I_f}$ orthogonal covariates.

The structure S of p_r sub-regressions within correlated covariates in $\boldsymbol{X}$ is described by:

$$\boldsymbol{X}^{I_r}_{|\boldsymbol{X}^{I_f},S} \text{ defined by } \forall j \in I_r : \boldsymbol{X}^{j}_{|\boldsymbol{X}^{I_f},S} = \boldsymbol{X}^{I_f}\boldsymbol{\alpha}^j + \boldsymbol{\varepsilon}^j \text{ with } \boldsymbol{\varepsilon}^j \sim \mathcal{N}(\boldsymbol{0}, \sigma_j^2 \boldsymbol{I}_n) \tag{4.1}$$

where $\boldsymbol{\alpha}^j \in \mathcal{R}^{(p-p_r)}$ are the sparse vectors of the regression coefficients between the covariates (each sub-regression freely implies different covariates). We also define $I_f = \{I_f^1, \ldots, I_f^p\}$ the set of the sets of indices of exogenous covariates with

$$\forall j \in I_r, I_f^j = \{i | \boldsymbol{\alpha}_i^j \neq 0\} \tag{4.2}$$

$$\forall j \notin I_r, I_f^j = \emptyset. \tag{4.3}$$

Then we have the explicit structure characterized by $S = \{I_f, I_r, \boldsymbol{p}_f, p_r\}$ where $p_r = |I_r|$, $\boldsymbol{p}_f = (p_f^1, \ldots, p_f^{p_r})$ is the vector of the number of covariates in each sub-regression and $p_f^j = |I_f^j|$, with $|.|$ the cardinal of an ensemble.

The partition of $\boldsymbol{X}$ implies the uncrossing rule $\boldsymbol{X}^{I_r} \cap \boldsymbol{X}^{I_f}$ *i.e.* endogenous variables don't explain other covariates. This hypothesis ensures that S contains no cycle and is straightforward readable (no need to order the sub-regressions). It is not so restrictive because cyclic structures have no sense and any non-cyclic structure can be associated with a structure that verifies the uncrossing constraint by just successively replacing endogenous covariates by their sub-regression when they are also exogenous in some other sub-regressions.

We make the choice to distinguish the response variable from the other endogenous variables (that are on the left of a sub-regression). Thus we have one regression on the response variable ($P(\boldsymbol{Y}|\boldsymbol{X})$) and a system of sub-regressions (without the response variable: $P(\boldsymbol{X}_r|\boldsymbol{X}_f, S)$). Then we consider correlations between the explicative covariates of the main regression, not between the residuals. We see that the S does not depend on $\boldsymbol{Y}$ so it can be learnt independently, even with a larger dataset (if missing values in $\boldsymbol{Y}$).

The structure obtained gives a system of linear regression that can be viewed as a recursive Simultaneous Equation Model (SEM)[Davidson and MacKinnon, 1993] [Timm, 2002].
We note $\boldsymbol{\alpha}$ the $(p-p_r)\times p_r$ matrix of the $\boldsymbol{\alpha}^j$. Here we suppose the $\boldsymbol{\varepsilon}_j$ independent but in other cases SUR (Seemingly Unrelated Regression [Zellner, 1962]) takes into account correlations between residuals SUR (Seemingly Unrelated Regression [Zellner, 1962]) and could be used to estimate $\boldsymbol{\alpha}$.

In the running example: $\boldsymbol{X}_r = \boldsymbol{x}_3$, $\boldsymbol{X}_f = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_5\}$, $p_r = 1$ and $\boldsymbol{\alpha}_3 = (1,1,0,0)'$ and we have $S = (\{\{1,2\}\}, \{3\}, (2), (1))$

4.2 Graph theory

We can model S by a Directed Acyclic Graph (DAG) whose vertices are the p covariates and arcs are the link between them described by the adjacency matrix $\boldsymbol{G}$ [Bondy and Murty, 1976]. This adjacency matrix is a binary $p \times p$ matrix with $\boldsymbol{G}_{i,j} = 1$ if and only if $i \in I_f^j$ that is $\boldsymbol{X}^j$ is explained by $\boldsymbol{X}^i$ and can also be seen as $\boldsymbol{\alpha}_i^j \neq 0$.

Graphical representation of S helps to understand it and can be compared to the bayesian network representation. It helps to interprete the structure has also been used to construct the algorithm to find S (chapter 6).

The partition of $\boldsymbol{X}$ mean that the associated graph is bipartite (vertices follow a partition $(\boldsymbol{X}^{I_r}, \boldsymbol{X}^{I_f})$) with arcs only going from $\boldsymbol{X}^{I_f}$ to $\boldsymbol{X}^{I_r}$.

We know ([Biggs, 1993])as a classical result of graph theory that the power of adjacency matrices give the paths in the graph: $\boldsymbol{G}_{i,j}^k \neq 0$ means that there is a path of length k going from $\boldsymbol{X}^i$ to $\boldsymbol{X}^j$. Because the graph is bipartite and arcs are only going from $\boldsymbol{X}^{I_f}$ to $\boldsymbol{X}^{I_r}$ we can deduce that $\boldsymbol{G}$ is nilpotent: $\boldsymbol{G}^2 = 0$. And we have the following result: every binary nilpotent matrix of order 2 can be seen as an adjacency matrix of a structure that respects the uncrossing rule. proof by contradiction: if there exist a path of length 2 between some vertices i and j then $\boldsymbol{G}_{i,j}^2 \neq 0$ so the matrix is not nilpotent of order 2. We can deduce that the number of feasible structure with p covariates is the number of binary nilpotent matrix of order 2.

We see that $\boldsymbol{G}$ completely describe S and that the sparse storage of G gives I_f which is sufficient to obtain S by doing $\forall 1 \leq j \leq p : p_f^j = |I_f^j|$, $I_r = \{j | p_f^j > 0\}$ and $p_r = |I_r|$. This decomposition helps us to enumerate all the feasible structure (and thus all the binary nilpotent matrix of order 2).
We note $\mathcal{S}_p$ the set of the feasible structure with p covariates. If we consider all the structure

with equiprobability:

$$
\begin{aligned}
S &= \{I_f, I_r, \boldsymbol{p}_f, p_r\} && (4.4)\\
P(S|p_r) &= P(I_f, \boldsymbol{p}_f|I_r, p_r)P(I_r|p_r) && (4.5)\\
&= P(\boldsymbol{p}_f|I_f, I_r, p_r)P(I_f|I_r, p_r)P(I_r|p_r) && (4.6)\\
&= P(I_f|I_r, p_r)P(I_r|p_r) && (4.7)\\
P(I_r|p_r) &= \frac{1}{\binom{p}{p_r}} && (4.8)\\
P(I_f|I_r, p_r) &= \frac{1}{(2^{p-p_r}-1)^{p_r}} && (4.9)\\
|\mathcal{S}_p| &= \sum_{p_r=0}^{p-1} |\mathcal{S}_{p|p_r}| = \sum_{p_r=0}^{p-1} \frac{1}{P(S|p_r)} = \sum_{p_r=0}^{p-1} \binom{p}{p_r}(2^{p-p_r}-1)^{p_r} && (4.10)
\end{aligned}
$$

where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{4.11}$$

is the binomial coefficient.
We have then $|\mathcal{S}_2| = 3$,$|\mathcal{S}_3| = 13$ and $|\mathcal{S}_{10}| > 13.26 \times 10^9$ so the number of feasible structures really explodes when p is growing.

In the running example: $|\mathcal{S}_5| = 841$

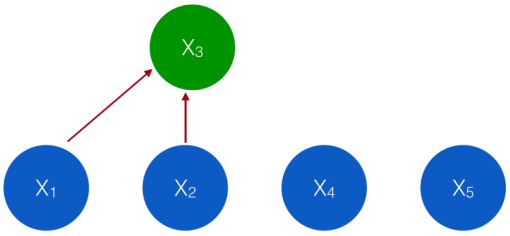


Figure 4.1: The bipartite graph associated to the running example

and the adjacency matrix is:

$$G = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

4.3 A by-product model: marginal regression with decorrelated covariates

Now we know $P(\boldsymbol{X}^{I_r}|\boldsymbol{X}^{I_f}, S)$ by the structure of sub-regressions, we are able to define a marginal regression model $P(\boldsymbol{Y}|\boldsymbol{X}^{I_f}, S)$ based on the reduced set of independent covariates $\hat{\boldsymbol{\beta}}_f$ without significant information loss. We use the information of the correlations structure to rewrite the true model without bias in the marginal space defined by the independent covariates.

Using the partition $\boldsymbol{X} = [\boldsymbol{X}^{I_f}, \boldsymbol{X}^{I_f}]$ we can rewrite (3.1):

$$\boldsymbol{Y}_{|\boldsymbol{X}^{I_f},\boldsymbol{X}^{I_r},S} = \boldsymbol{X}^{I_f}\boldsymbol{\beta}_{I_f} + \boldsymbol{X}^{I_r}\boldsymbol{\beta}_{I_r} + \boldsymbol{\varepsilon}_{\boldsymbol{Y}} \tag{4.12}$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}_{I_f}, \boldsymbol{\beta}_{I_r}) \in \mathcal{R}^p$ is the vector of the regression coefficients associated respectively to $\boldsymbol{X}^{I_f}$ and $\boldsymbol{I}_n$ the identity matrix. We note that (4.1) and (4.12) give also by simple integration on $\boldsymbol{X}^{I_r}$ a marginal regression model on $\boldsymbol{Y}$ *depending only on uncorrelated covariates* $\boldsymbol{X}^{I_f}$:

$$P(\boldsymbol{Y}|\boldsymbol{X}^{I_f}) = \int_{\boldsymbol{X}^{I_r}} P(\boldsymbol{Y}|\boldsymbol{X}^{I_r}, \boldsymbol{X}^{I_f}) P(\boldsymbol{X}^{I_r}|\boldsymbol{X}^{I_f}) d\boldsymbol{X} \tag{4.13}$$

$$\boldsymbol{Y}_{|\boldsymbol{X}^{I_f},S} = \boldsymbol{X}^{I_f}(\boldsymbol{\beta}_{I_f} + \sum_{j\in I_r} \beta_j \boldsymbol{\alpha}_j) + \sum_{j\in I_r} \beta_j \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_Y \tag{4.14}$$

$$= \boldsymbol{X}^{I_f}\boldsymbol{\beta}^*_{I_f} + \boldsymbol{\varepsilon}^*_Y \tag{4.15}$$

This model is still the true model and OLS estimator will still give an unbiased estimator, but its variance will be reduced by both dimension reduction and decorrelation (variables in $\boldsymbol{X}^{I_f}$ are independent so the matrix $\boldsymbol{X}^{I'_f}\boldsymbol{X}^{I_f}$ will be well-conditioned). So the information given by the structure S allows to reduce the variance without adding bias, by simple marginalization. Nevertheless, to be able to compare the bias-variance tradeoff, we can see this model as a variable pre-selection independent of the response in $\boldsymbol{Y}_{|\boldsymbol{X}}$. We note that it is simply a linear regression on some of the original covariates so we only made a pre-treatment on the dataset by selecting $\boldsymbol{X}^{I_f}$ because of the correlations given by S. So we also get the model

$$\boldsymbol{Y}_{|\boldsymbol{X},S} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}^*_Y \text{ where } \boldsymbol{\beta}^* = (\boldsymbol{\beta}^*_{I_f}, \boldsymbol{\beta}^*_{I_r}) \text{ and } \boldsymbol{\beta}^*_{I_r} = \boldsymbol{0} \tag{4.16}$$

for which OLS estimator of the coefficients may be biased.

Running example: $\boldsymbol{Y}_{|\boldsymbol{X}^{I_f}} = 2\boldsymbol{x}_1 + 2\boldsymbol{x}_2 + \boldsymbol{x}_4 + \boldsymbol{x}_5 + \boldsymbol{\varepsilon}_3 + \boldsymbol{\varepsilon}_Y$

4.4 Strategy of use: pre-treatment before classical estimation/selection methods

As a pre-treatment, the model allows usage of any method in a second time to estimate $\boldsymbol{\beta}^*_{I_f}$, even with variable selection methods like LASSO or a best subset algorithm like stepwise [Seber and Lee, 2012]. However, we always have $\boldsymbol{X}^{I_r} = \boldsymbol{0}$

After selection and estimation we will obtain a model with *two steps of variable selection*: the decorrelation step by marginalization(coerced selection associated to redundant information defined in S) and the classical selection step, with different meanings for obtained zeros in $\hat{\boldsymbol{\beta}}^*_{I_f}$ (irrelevant covariates) and for $\hat{\boldsymbol{\beta}}^*_{I_r} = 0$ (redundant information). Thus we are able to distinguish the reasons of selection and consistency issues don't mean interpretation issues any more. So we dodge the drawbacks of both grouping effect and variable selection.

The explicit structure is parsimonious and simply consists in linear regressions and thus is easily understood by non statistician, allowing them to have a better knowledge of the phenomenon inside the dataset and to take better actions. Expert knowledge can even be added to the structure, physical models for example.

Moreover, the uncrossing constraint (partition of $\boldsymbol{X}$) guarantee to keep a simple structure easily interpretable (no cycles and no chain-effect) and straightforward readable.

There is no theoretical guarantee that our model is better. It's just a compromise between numerical issues caused by correlations for estimation and selection versus increased variability due to structural hypothesis. We just play on the traditional bias-variance tradeoff.

Chapter 5

Numerical results with a known structure

Abstract: Premiers résultats numériques pour une structure (hors coefficients de sous-régression) connue. On constate un net apport de la méthode de préselection.

5.1 Illustration of the tradeoff conveyed by the pre-treatment

We compare the OLS estimator on $\boldsymbol{X}$ defined in section 3.2 with the estimator obtained by the pre-treatment that is $\boldsymbol{X}^{I_f}$ selection.

For the marginal regression model defined in (4.15) we have the OLS unbiased estimator of $\boldsymbol{\beta}^*$:

$$\hat{\boldsymbol{\beta}}^*_{I_f} = (\boldsymbol{X}^{I'_f}\boldsymbol{X}^{I_f})^{-1}\boldsymbol{X}^{I'_f}\boldsymbol{Y} \text{ and } \hat{\boldsymbol{\beta}}^*_{I_r} = \mathbf{0} \tag{5.1}$$

We see in (4.14) that it gives an unbiased estimation of $\boldsymbol{Y}$ and $\boldsymbol{\beta}^*$ but in terms of $\boldsymbol{\beta}$ this estimator is biased:

$$\mathrm{E}[\hat{\boldsymbol{\beta}}^*_{I_f}|\boldsymbol{X}^{I_f}] = \boldsymbol{\beta}_{I_f} + \sum_{j\in I_r}\beta_j\boldsymbol{\alpha}_j \text{ and } \mathrm{E}[\hat{\boldsymbol{\beta}}^*_{I_r}|\boldsymbol{X}^{I_f}] = \mathbf{0} \tag{5.2}$$

with variance:

$$\mathrm{Var}[\hat{\boldsymbol{\beta}}^*_{I_f}|\boldsymbol{X}^{I_f}] = (\sigma^2_Y + \sum_{j\in I_r}\sigma^2_j\beta^2_j)(\boldsymbol{X}^{I'_f}\boldsymbol{X}^{I_f})^{-1} \text{ and } \mathrm{Var}[\hat{\boldsymbol{\beta}}^*_{I_r}|\boldsymbol{X}^{I_f}] = \mathbf{0} \tag{5.3}$$

We see that the variance is reduced compared to OLS described in equation (3.3)(no correlations and smaller matrix give better conditioning) for small values of σ_j *i.e.* strong correlations. So we play on the bias-variance tradeoff, reducing the variance by adding a bias.

The theoretical Mean Squared Error (MSE) on $\hat{\boldsymbol{\beta}}$ is:

$$\begin{aligned}
E[\mathrm{MSE}(\hat{\boldsymbol{\beta}}|\boldsymbol{X})] &= \| \text{ Bias } \|^2_2 + \mathrm{Tr}(\mathrm{Var}(\hat{\boldsymbol{\beta}})) && (5.4)\\
E[\mathrm{MSE}(\hat{\boldsymbol{\beta}}_{OLS}|\boldsymbol{X})] &= 0 + \sigma^2_Y\,\mathrm{Tr}((\boldsymbol{X}'\boldsymbol{X})^{-1}) && (5.5)\\
E[\mathrm{MSE}(\hat{\boldsymbol{\beta}}^*_{OLS}|\boldsymbol{X})] &= \| \sum_{j\in I_r}\beta_j\boldsymbol{\alpha}_j \|^2_2 + \| \boldsymbol{\beta}_{I_r} \|^2_2 + (\sigma^2_Y + \sum_{j\in I_r}\sigma^2_j\beta^2_j)\,\mathrm{Tr}((\boldsymbol{X}^{I'_f}\boldsymbol{X}^{I_f})^{-1}) && (5.6)
\end{aligned}$$

To better illustrate the bias-variance tradeoff, we look at the running example. We observe the theoretical Mean Squared Error (MSE) of the estimator of both OLS and CorReg's marginal model for several values of σ_3 (strength of the sub-regression) and n. Figure 5.1 shows the theoretical MSE evolution with the strength of the sub-regression. In this section, all experiences have been made 100 times to obtain smooth curves. So we have generated 100

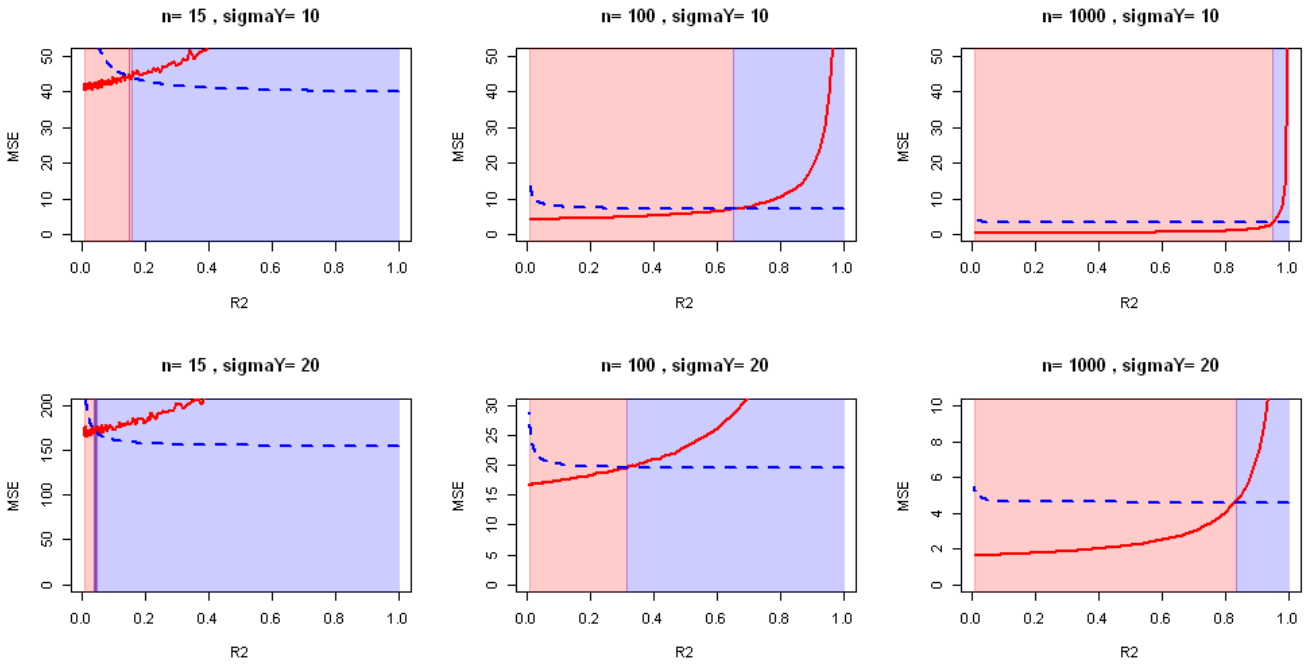


Figure 5.1: Theoretical MSE on $\hat{\boldsymbol{\beta}}$ of OLS (red) and `CorReg`'s marginal model (blue) estimators for varying R^2 of the sub-regression, n and σ_Y.

times $\boldsymbol{X}$ and $\boldsymbol{Y}$. It is clear in Figure 5.1 that the marginal model is more robust than OLS on $\boldsymbol{X}$. And when sub-regression get weaker ($\mathcal{R}^2$ tends to 0) it remains stable until extreme values (sub-regression nearly fully explained by the noise). We also see that the error implied by strong correlations shrinks with the rise of n. We know that σ_Y multiplies $\text{Tr}(\text{Var}(\hat{\boldsymbol{\beta}})) = \text{Tr}(\text{Var}(\hat{\boldsymbol{\beta}}_{I_f}))+ \text{Tr}(\text{Var}(\hat{\boldsymbol{\beta}}_{I_r}))$ for both models but for the marginal model $\text{Tr}(\text{Var}(\hat{\boldsymbol{\beta}}_{I_r})) = 0$. Thus, when σ_Y^2 rises it increases the advantage of `CorReg` versus OLS. It illustrates the importance of dimension reduction when the main model has a strong noise (very usual case on real datasets where true model is not even exactly linear).

But it is only the theoretical MSE and we want to know what happens in the real life.

5.2 Observed MSE comparison

We look at the empirical MSE on both $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{Y}}$. Here again, each configuration is computed 100 times and we take the mean to smooth the curves. The MSE on $\hat{\boldsymbol{Y}}$ is computed on a validation sample with 1 000 individuals. Our marginal model remains better for strong correlations.

We also look at the observed MSE on both $\boldsymbol{\beta}$ and $\boldsymbol{Y}$ for some of the methods depicted above.

5.2.1 On the running example

We see in figures 5.2 and 5.3 that MSE on $\hat{\boldsymbol{Y}}_{OLS}$ give the same results as those on $\hat{\boldsymbol{\beta}}_{OLS}$: the marginal model is better for stronger sub-regressions, smaller samples and weaker main regression. But we notice that when the MSE on $\hat{\boldsymbol{\beta}}_{OLS}$ explodes, the MSE on $\hat{\boldsymbol{Y}}_{OLS}$ does not grow so much. This is a good illustration of the problem generated by the correlations. The model seems to be good in prediction but coefficients are very far from the real value and interpretation can be extremely misleading.

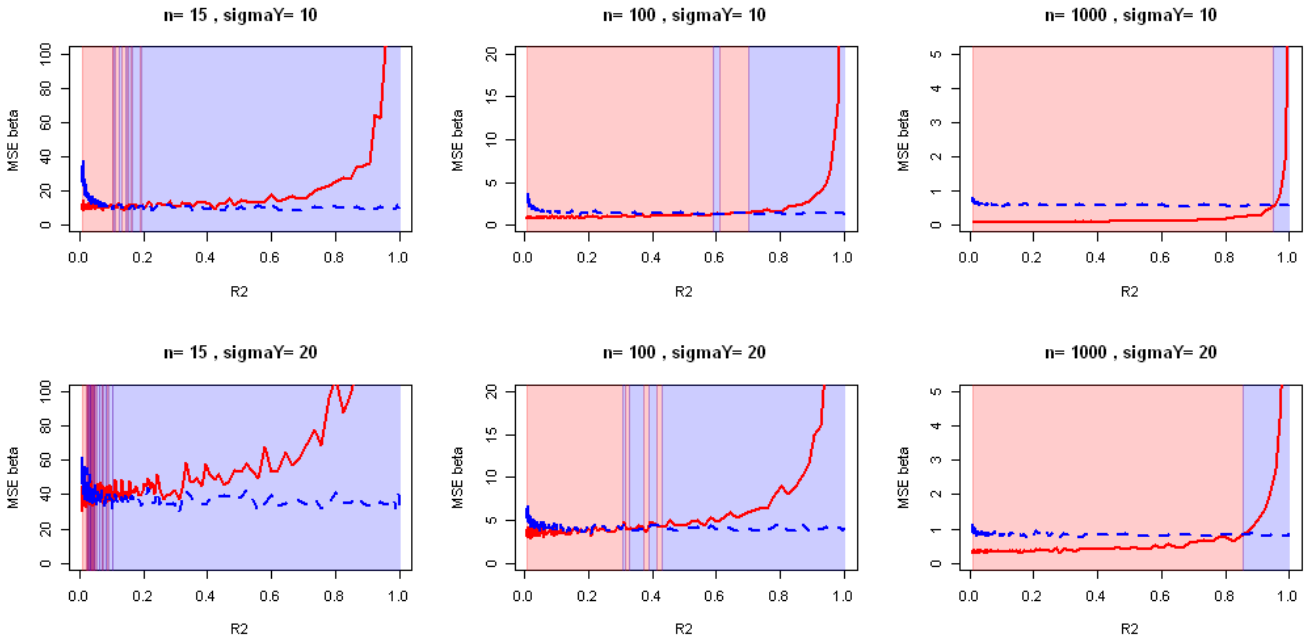


Figure 5.2: Observed MSE on $\hat{\boldsymbol{\beta}}$ of OLS (red) and `CorReg`'s marginal model (blue) estimators for varying R^2 of the sub-regression, n and σ_Y. $p = 5$ covariates.

Once again, usage of QR decomposition to invert matrices does improve the results significantly compared to the theoretical MSE. Mathematics offer a wide range of tools and this is a good example of how linear algebra can be used in statistical fields.

Figure 5.4 shows that variable selection done the LASSO gives a biased $\hat{\boldsymbol{\beta}}$ by setting some coefficients to 0 but strong correlations makes this bias neutral for prediction (figure 5.5). Here the LASSO tends to propose the same model as we do with our marginal model, but without explanation. We will see later in section 7.3 that it is not sufficient in higher dimension.

Here again (figures 5.6 and 5.7), ridge regression provides good results for this running example. But we will see later in section 7.3 that high dimension reduces the efficiency of the ridge regression when some covariates begin to be irrelevant or not enough relevant because ridge regression is not able to select relevant covariates.

Elasticnet and stepwise give results quite similar to the LASSO (figures 5.8 to 5.11).

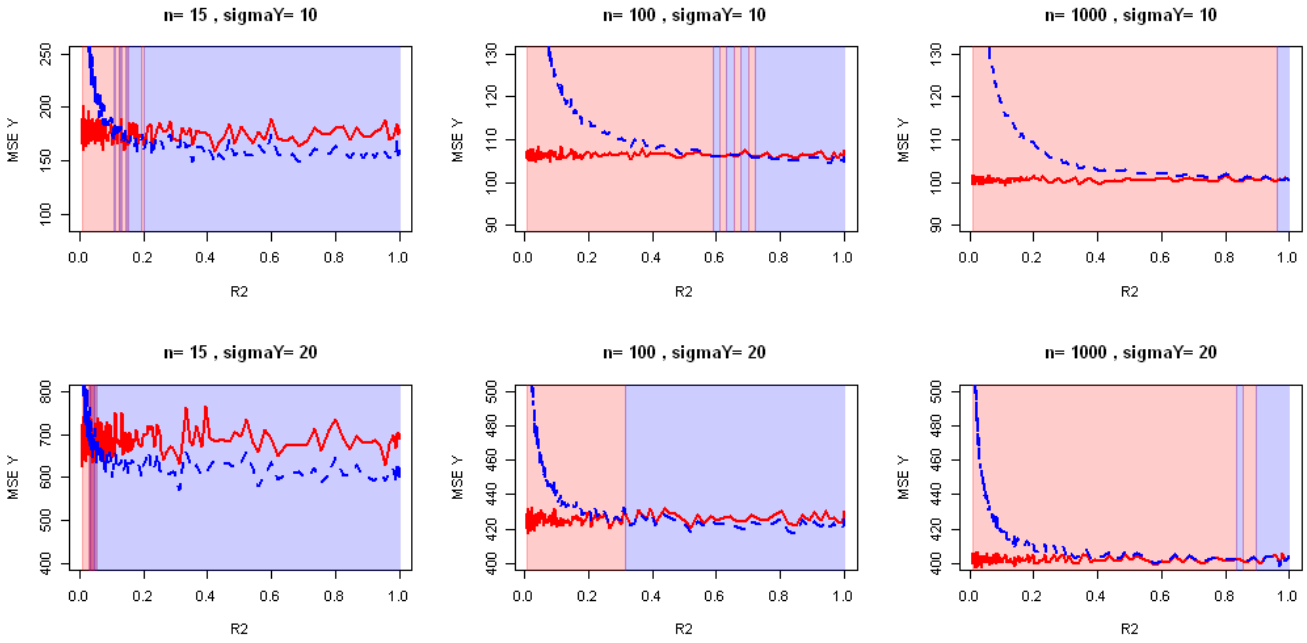


Figure 5.3: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{OLS}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

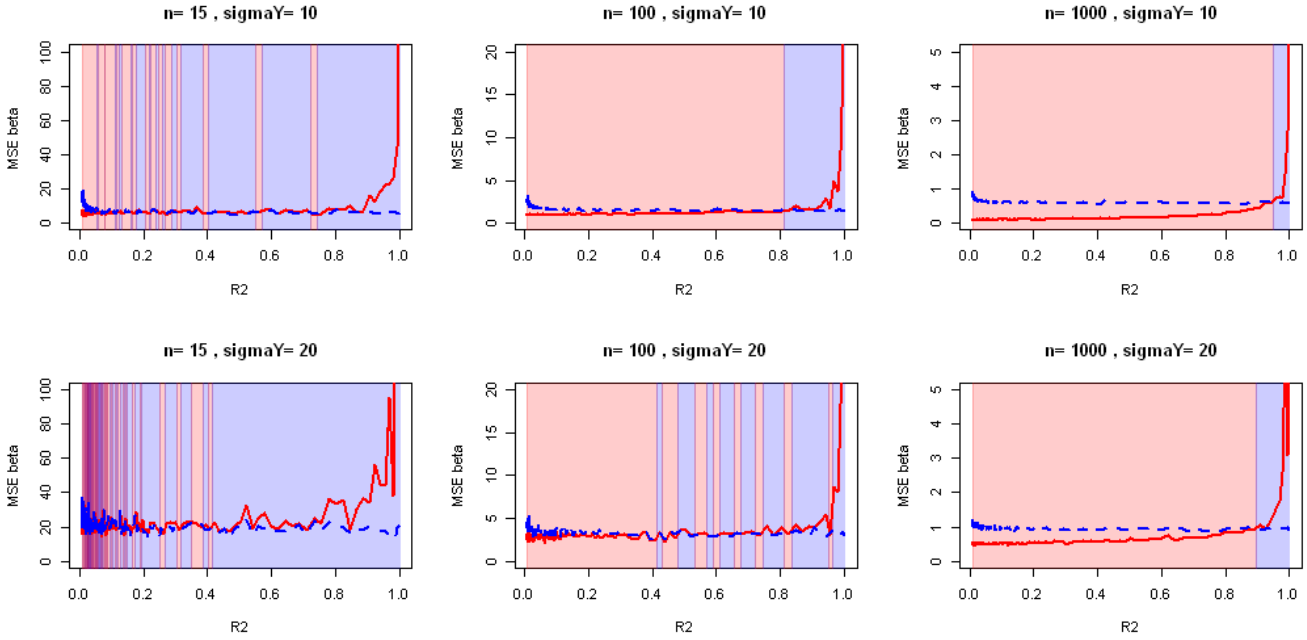


Figure 5.4: Observed MSE on $\hat{\boldsymbol{\beta}}$ of LASSO with LAR on both $\boldsymbol{X}$ (red) and `CorReg`'s marginal $\boldsymbol{X}^{I_f}$ (blue) for varying R^2 of the sub-regression, n and σ_Y. $p = 5$ covariates.

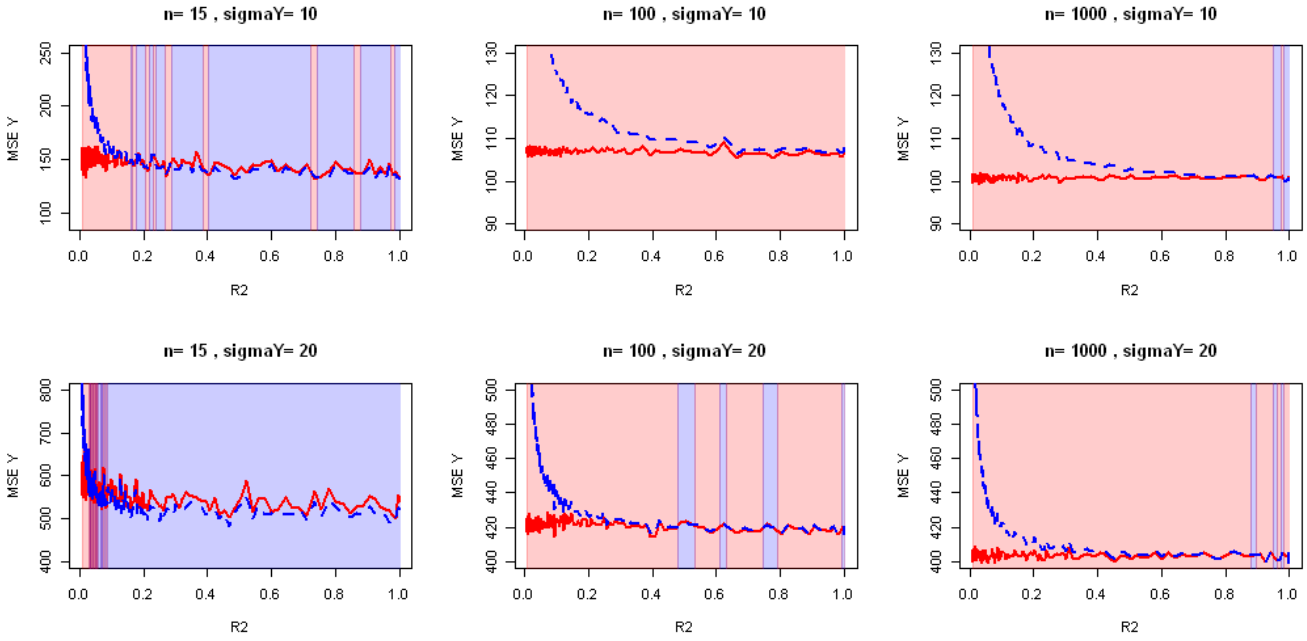


Figure 5.5: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{LASSO}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

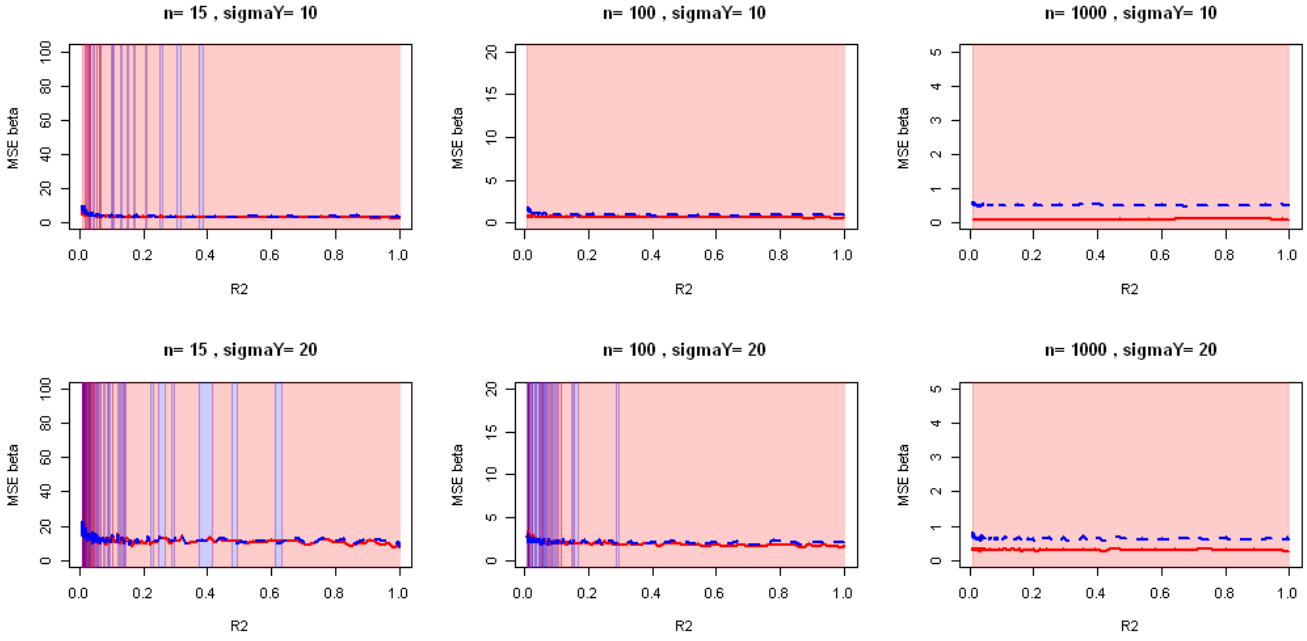


Figure 5.6: Observed MSE on $\hat{\boldsymbol{\beta}}_{ridge}$ on both $\boldsymbol{X}$ (red) and `CorReg`'s marginal $\boldsymbol{X}^{I_f}$ (blue) for varying R^2 of the sub-regression, n and σ_Y. $p = 5$ covariates.

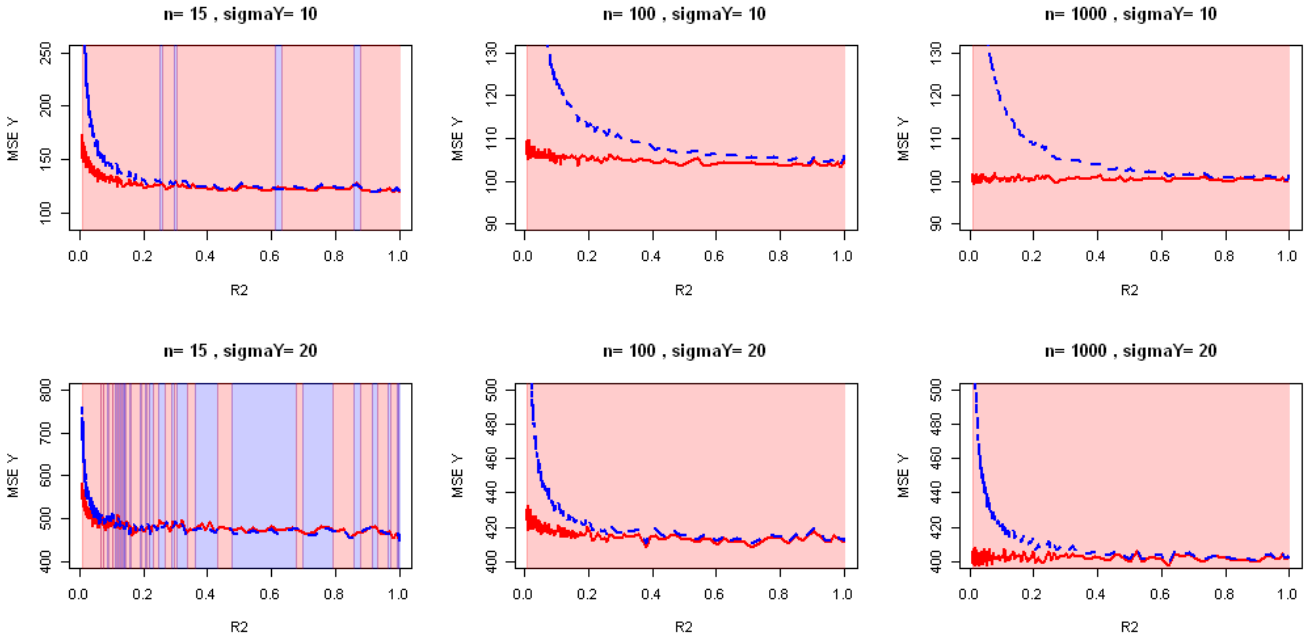


Figure 5.7: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{ridge}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

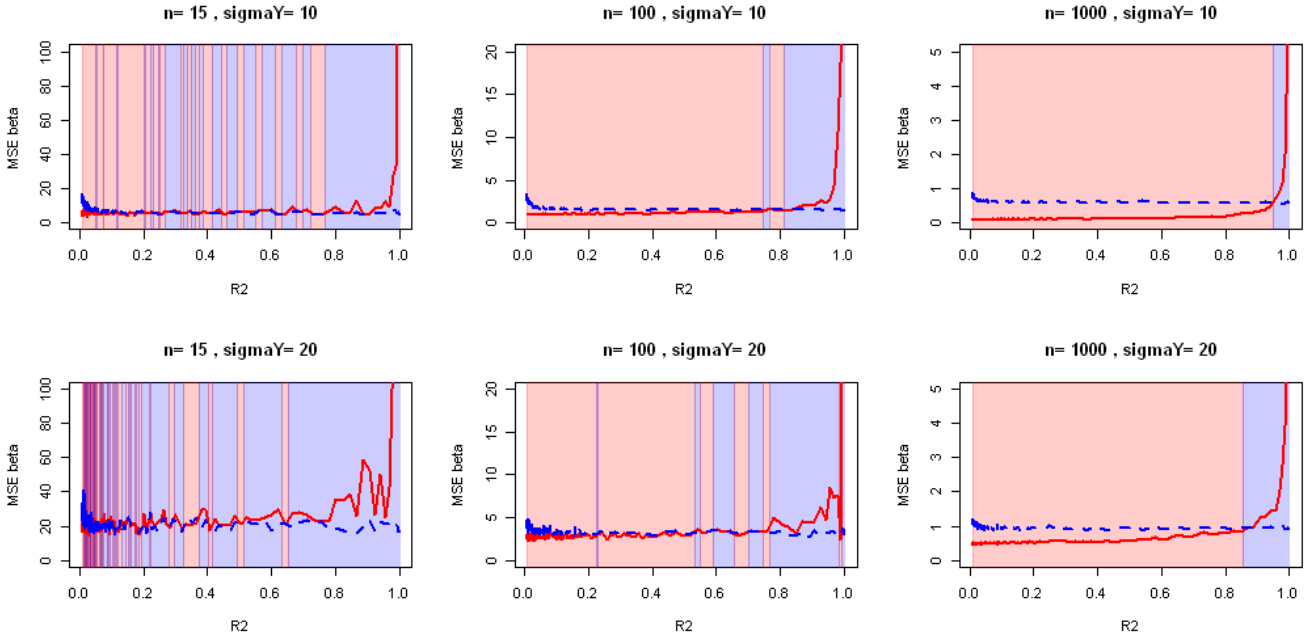


Figure 5.8: Observed MSE on $\hat{\boldsymbol{\beta}}_{elasticnet}$ on both $\boldsymbol{X}$ (red) and `CorReg`'s marginal $\boldsymbol{X}^{I_f}$ (blue) for varying R^2 of the sub-regression, n and σ_Y. $p = 5$ covariates.

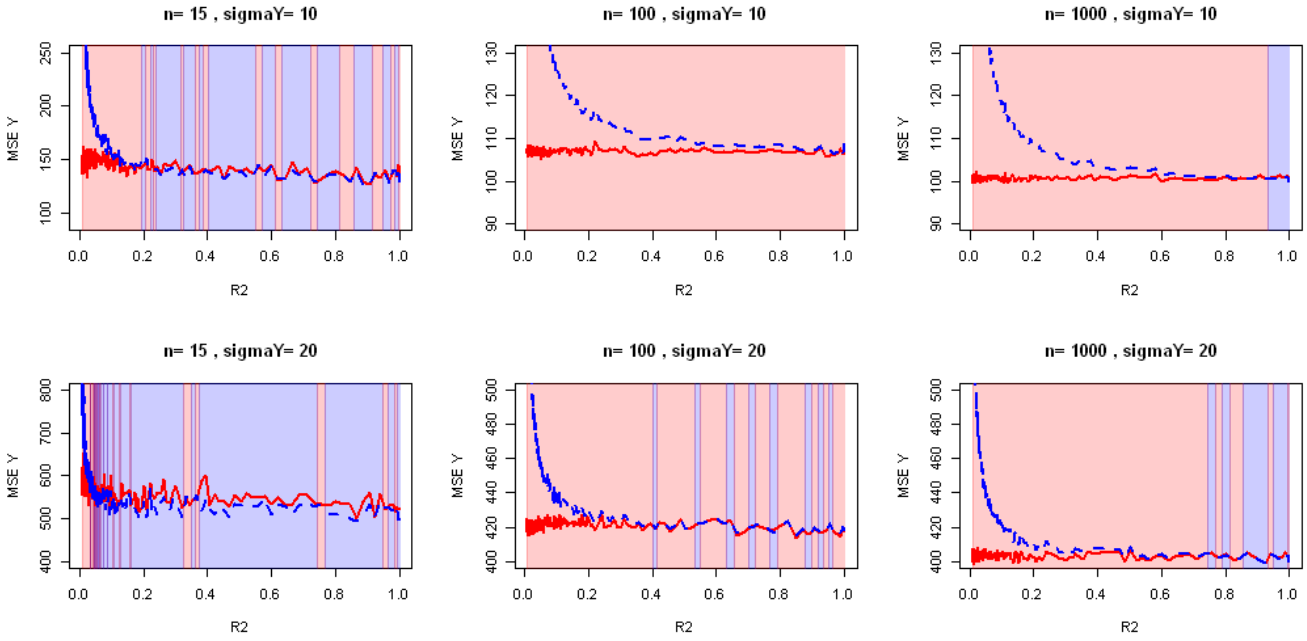


Figure 5.9: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{elasticnet}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

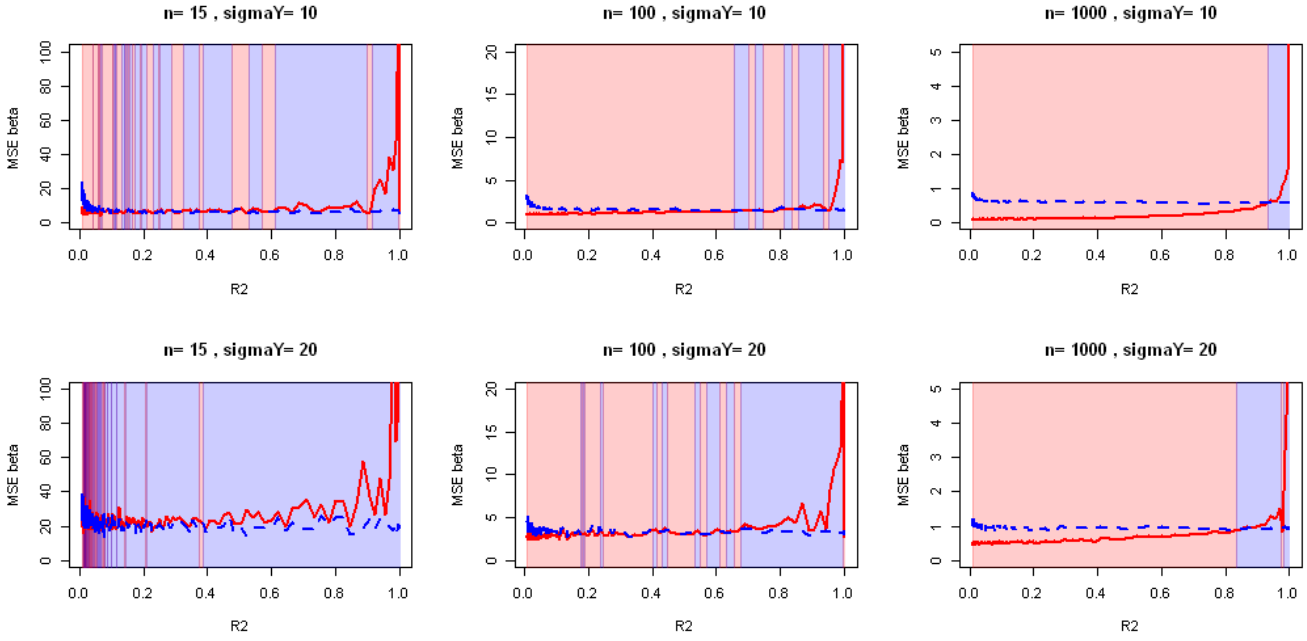


Figure 5.10: Observed MSE on $\hat{\boldsymbol{\beta}}_{stepwise}$ on both $\boldsymbol{X}$ (red) and `CorReg`'s marginal $\boldsymbol{X}^{I_f}$ (blue) for varying R^2 of the sub-regression, n and σ_Y. $p = 5$ covariates.

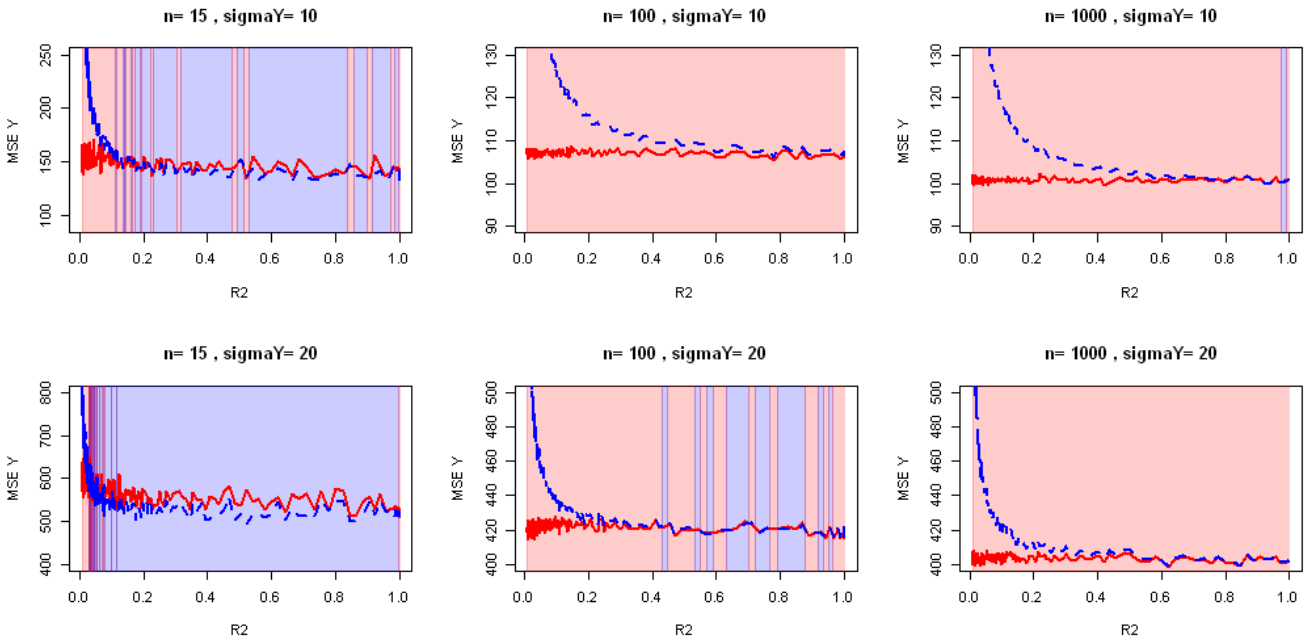


Figure 5.11: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{stepwise}$ with the strength of the correlations for various sample sizes and strength of regression. $p = 5$ covariates.

5.2.2 On more complex datasets

The `CorReg` package has been tested on simulated datasets. For each simulation, $p = 40$, the R^2 of the main regression is 0.4, variables in $\boldsymbol{X}_f$ follow Gaussian mixture models of $\lambda = 5$ classes which means follow Poisson's law of parameter $\lambda = 5$ and which standard deviation is λ. The β_j and the coefficients of the $\boldsymbol{\alpha}_j$ are generated according to the same Poisson law but with a random sign. $\forall j \in I_r, p_1^j = 2$ (sub-regressions of length 2) and we have $p_r = 16$ sub-regressions. The datasets were then scaled so that covariates $\boldsymbol{X}^{I_r}$ don't have a greater variance or mean. Results are based on the true S used to generate the dataset. When $n < p$, a frequently used method is the Moore-Penrose generalized inverse [Katsikis and Pappas, 2008], thus OLS can obtain some results even with $n < p$. When using penalized estimators for selection, a last Ordinary Least Square step is added to improve estimation because penalisation is made to select variables but also shrinks remaining coefficients. This last step allows to keep the benefits of shrinkage (variable selection) without any impact on remaining coefficients (see [Zhang and Shen, 2010]) and is applied for both classical and marginal model. We compare different methods with and without `CorReg` as a pretreatment. All the results are provided by the `CorReg` package. Here $\boldsymbol{Y}$ depends on all the covariate and the MSE provided were computed on a validation sample of 1 000 individuals each time.

Figures 5.12 to 5.14 illustrate what happens for OLS. Our marginal model is better than OLS for $n < p$ and the complete model (that is the true model) only starts to win for values of n many times larger than p and weak correlations. We still have the phenomenon of MSE on $\hat{\boldsymbol{\beta}}$ more impacted by correlations than the MSE on $\hat{\boldsymbol{\beta}}$, and our main goal is interpretation so our marginal model really is efficient in our context.

Figures 5.15 to 5.17 show that even if the LASSO is able to select a subset of covariate and even if we have seen with OLS that taking a subset can give better results, the LASSO does not do so and give more complex models than our marginal model until correlations are extremely strong. We also observe that our marginal model combined with the LASSO has varying complexities so our pretreatment by selection is just a pretreatment and not competitor against the LASSO. Such combination improves the results in a significant way when compared to the LASSO on the complete dataset or OLS on the marginal model. We see that the complexity rises with n but the lasso never keeps all the covariates even with $n = 400 = 10p$ when used on the whole dataset but keep all the covariates in $\boldsymbol{X}^{I_f}$ when used on the marginal model. The main result here is that the LASSO can be improved by pretreatment selection both with $n < p$ and $n >> p$ with strong correlations so this well known variable selection method really suffers from correlations.

Ridge regression (Figures 5.24 to 5.26) is not improved by the marginal model. Ridge regression is protected against correlations but we see that ridge regression applied on $\boldsymbol{X}^{If}$ (even if it is not the true model) give predictions quite similar to those from ridge regression but with less covariates. Ridge regression will only be damaged when variable selection is needed.

Elasticnet and stepwise (Figures 5.18 to 5.23)give results mostly equivalent to the LASSO but stepwise seems to be a bit less efficient (higher MSE values). This last point illustrates why we need a specific algorithm to find the structure S and not only variable selection by stepwise like in the method from Maugis [Maugis et al., 2009].

Further results are provided in sections 7 and 8.

Ordinary Least Squares when Y depends on all variables in X, true S known

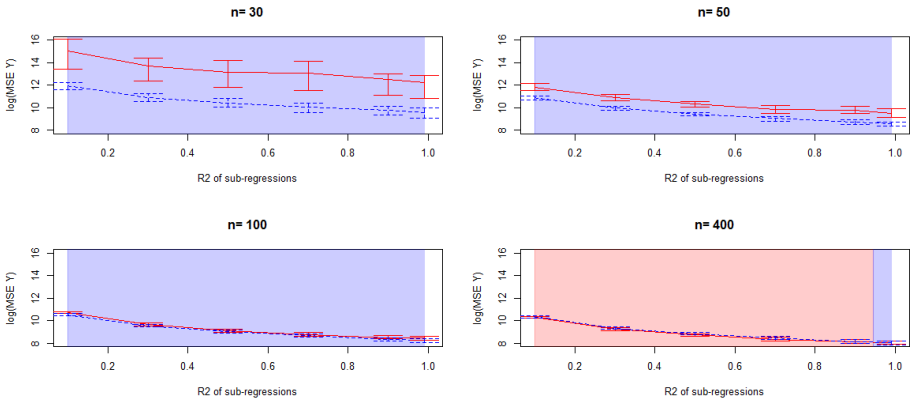


Figure 5.12: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

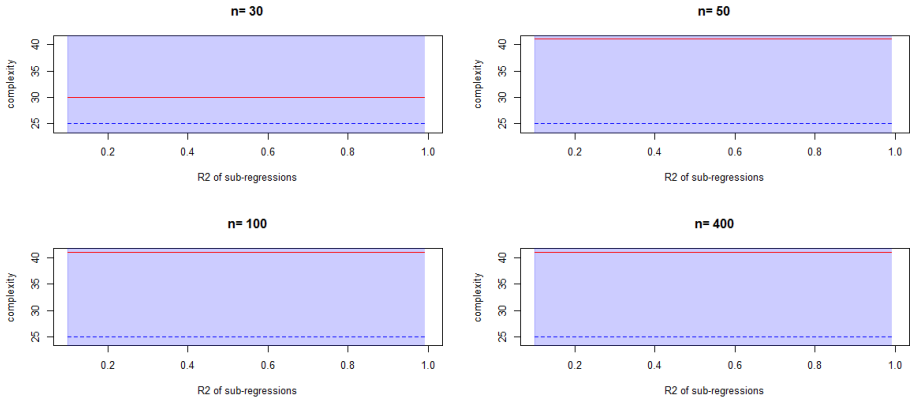


Figure 5.13: Comparison of the complexities, red=classical (complete) model, blue=marginal model

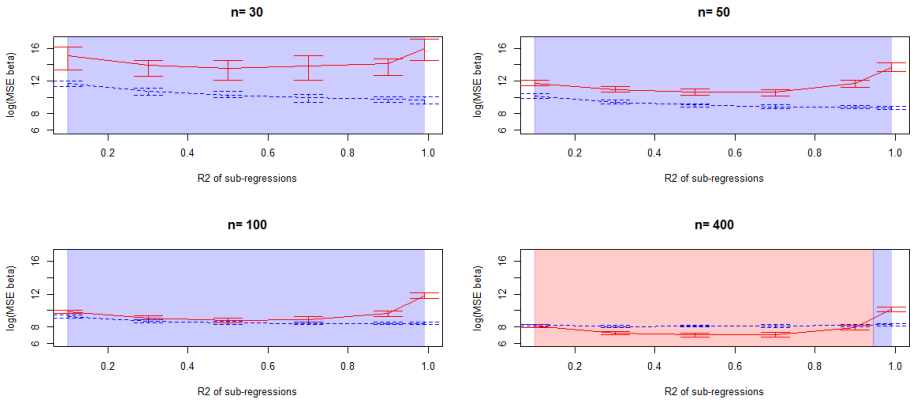


Figure 5.14: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

LASSO when $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$, true S known

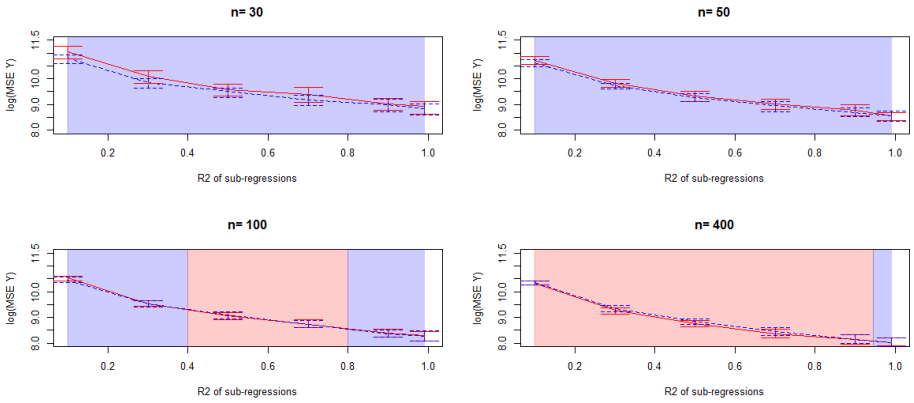


Figure 5.15: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

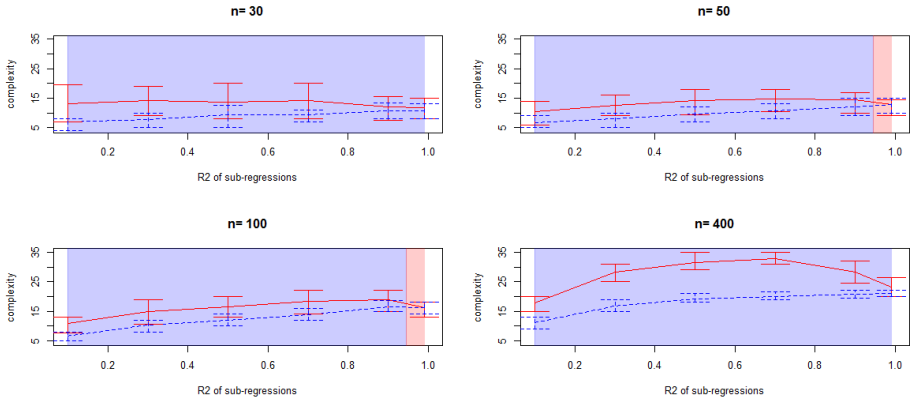


Figure 5.16: Comparison of the complexities, red=classical (complete) model, blue=marginal model

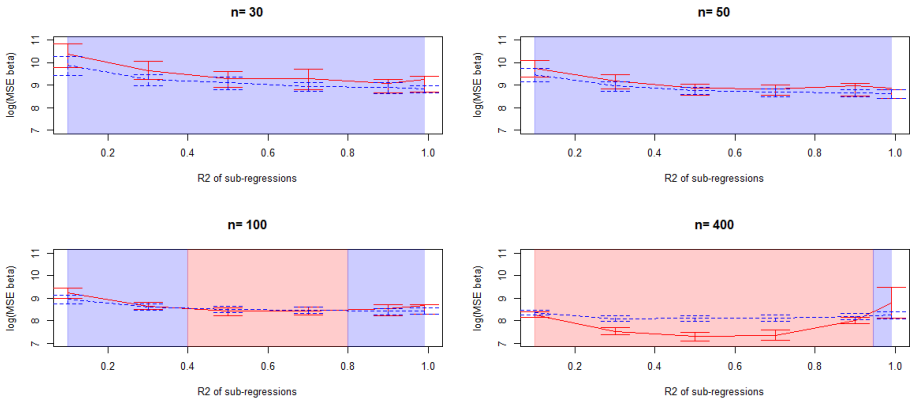


Figure 5.17: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Elasticnet when $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$, true S known

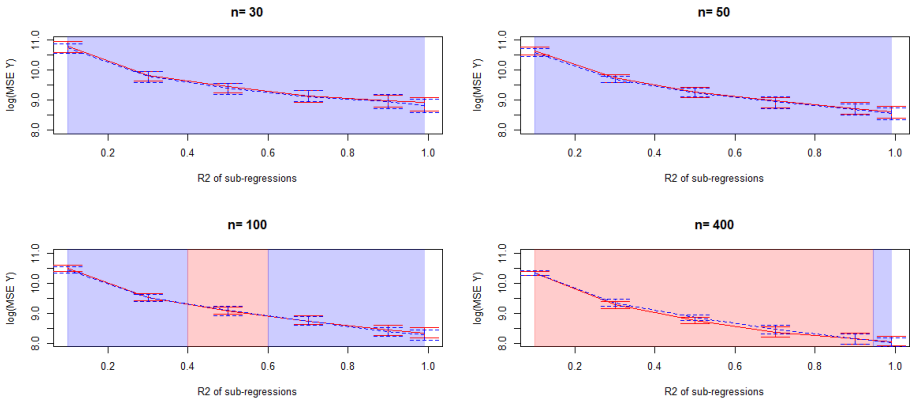


Figure 5.18: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

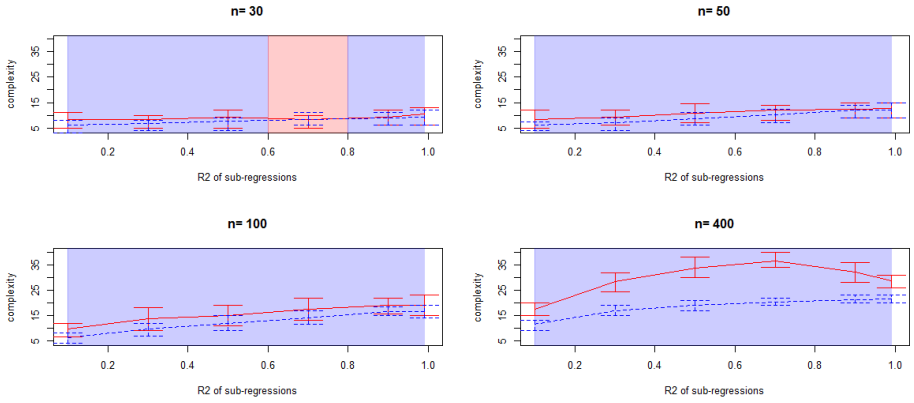


Figure 5.19: Comparison of the complexities, red=classical (complete) model, blue=marginal model

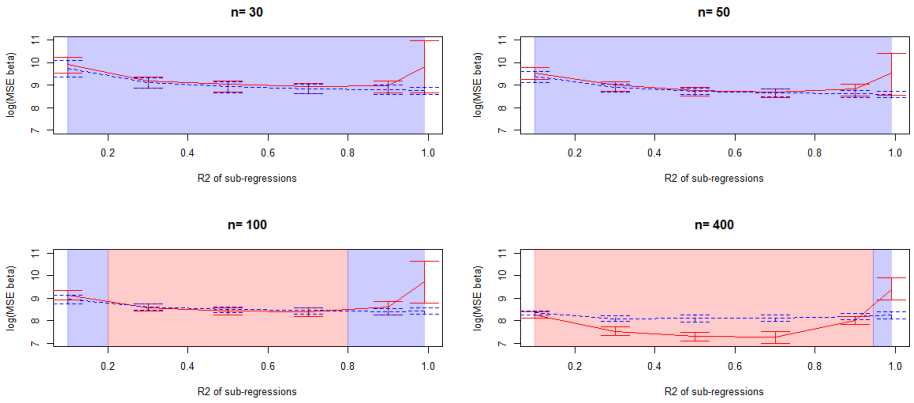


Figure 5.20: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Stepwise when $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$, true S known

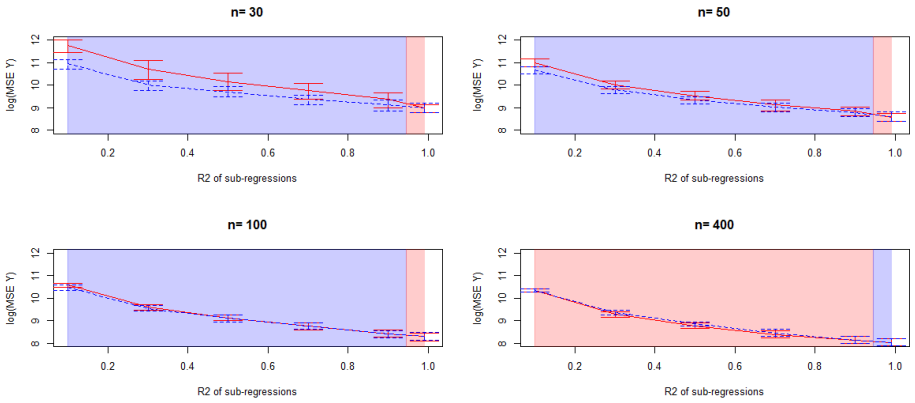


Figure 5.21: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

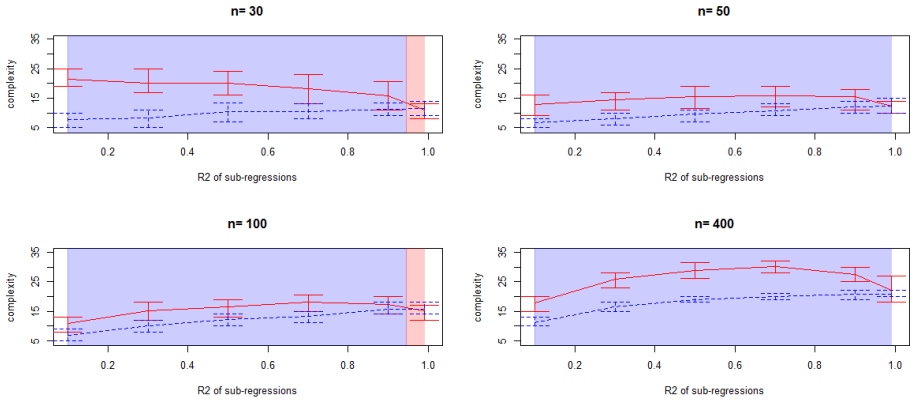


Figure 5.22: Comparison of the complexities, red=classical (complete) model, blue=marginal model

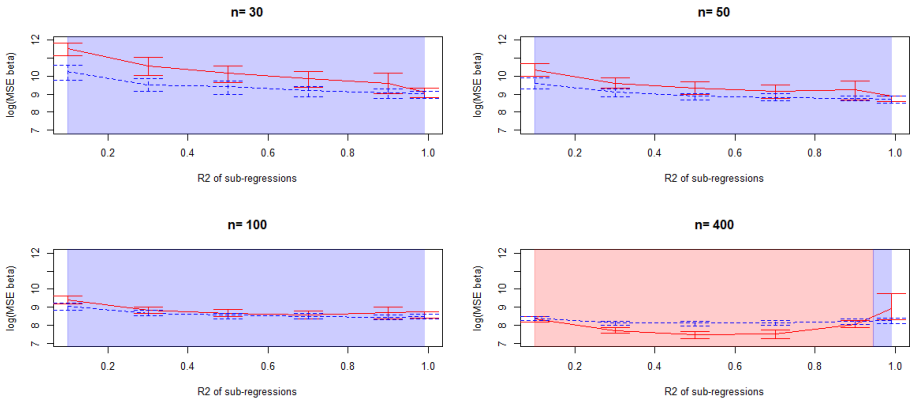


Figure 5.23: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Ridge regression when $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$, true S known

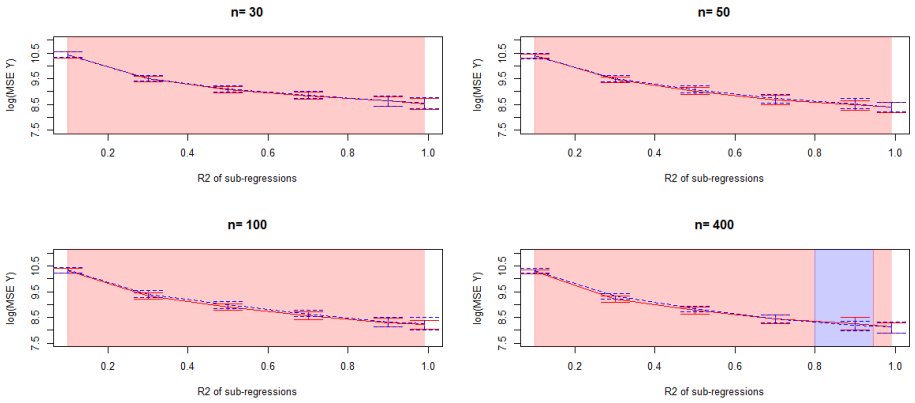


Figure 5.24: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

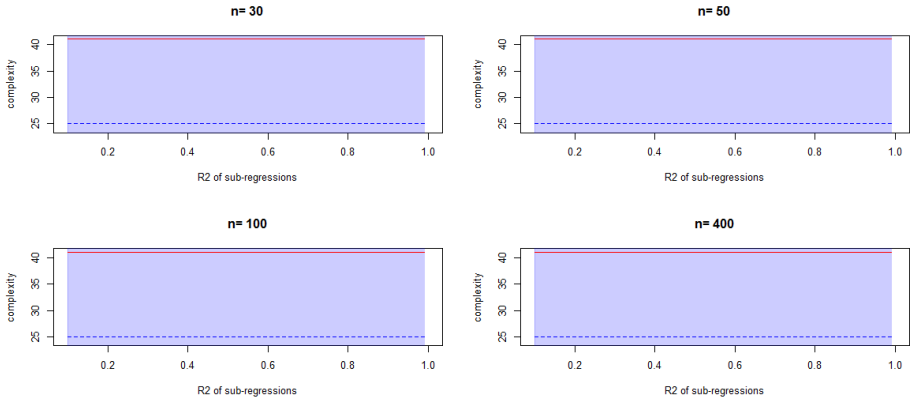


Figure 5.25: Comparison of the complexities, red=classical (complete) model, blue=marginal model

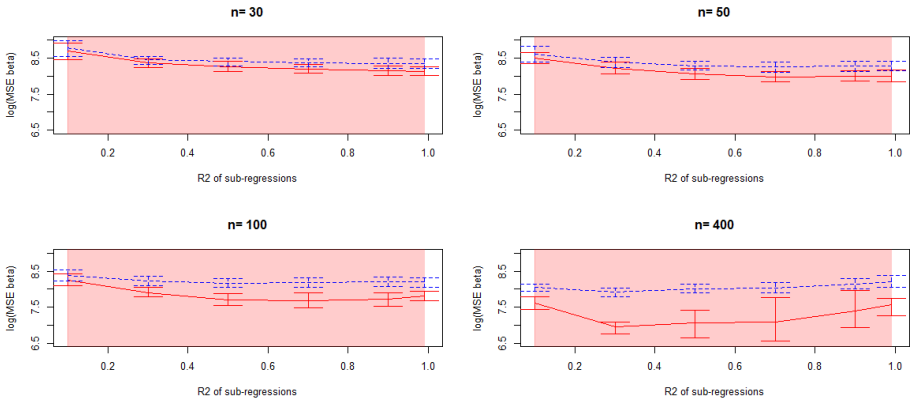


Figure 5.26: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Chapter 6

Estimation of the Structure of subregression by MCMC

Abstract: La structure de sous-régression est supposée inconnue. Il nous faut donc la trouver. Un algorithme de type MCMC est proposé pour résoudre cette problématique. La mise en oeuvre de celui-ci passe par une modélisation complète du jeu de données et nous pousse à introduire un nouveau critère de choix de modèle tenant compte du nombre de modèles testés.

6.1 Bayesian approach

Structural equations models are often used in social sciences and economy where a structure is supposed "by hand" but here we want to find it automatically. Graphical LASSO [Friedman et al., 2008] offers a method to obtain a structure based on the precision matrix (inverse of the variance-covariance matrix), setting some coefficients of the precision matrix to zero (see section 6.8). But the resulting matrix is symmetric and we need an oriented structure for S to avoid cycles.

Cross-validation is very time-consuming and thus not friendly with combinatory problematics. Moreover, we need a criterion compatible with structures of different sizes (varying p_r) and not related with $\boldsymbol{Y}$ because the structure is inherent to $\boldsymbol{X}$ only. Thus it must be a global criterion. Because it is about model selection, we decide to follow a Bayesian approach ([Raftery, 1995], [Andrieu and Doucet, 1999],[Chipman et al., 2001]).

We want to find the most probable structure S knowing the dataset, so we search for the structure that maximizes $P(S|\boldsymbol{X})$ and we have:

$$P(S|\boldsymbol{X}) \quad \propto \quad P(\boldsymbol{X}|S)P(S) = P(\boldsymbol{X}^{I_r}|\boldsymbol{X}^{I_f}, S)P(\boldsymbol{X}^{I_f}|S)P(S) \tag{6.1}$$

So we will try to maximize $\psi(\boldsymbol{X}, S) = P(\boldsymbol{X}|S)P(S)$. It will be done with a Markov chain Monte-Carlo algorithm (MCMC).

6.2 Sub-regression structure in details

6.2.1 Modeling the uncorrelated covariates: a full generative approach on $P(\boldsymbol{X})$

To be able to compare structures with $P(S|\boldsymbol{X})$, we need a full generative model on $\boldsymbol{X}$. Subregressions give $P(\boldsymbol{X}^{I_r}|\boldsymbol{X}^{I_f}, S)$ but $P(\boldsymbol{X}^{I_f}|S)$ is still undefined. We suppose that variables in

$\boldsymbol{X}^{I_f}$ follow Gaussian mixtures of $K_j > 0$ components:

$$\forall \boldsymbol{X}^j \notin \boldsymbol{X}^{I_r} : \boldsymbol{X}^j_{|S} \sim f(\boldsymbol{\theta}_j) = \mathcal{GM}(\boldsymbol{\pi}_j; \boldsymbol{\mu}_j; \boldsymbol{\sigma}^2_j) \text{ with } \boldsymbol{\pi}_j, \boldsymbol{\mu}_j, \boldsymbol{\sigma}^2_j \text{ vectors of size } K_j. \tag{6.2}$$

The great flexibility [McLachlan and Peel, 2004] of such models makes our model more robust. Gaussian case is just a special case ($K_j = 1$) of Gaussian mixture so it is included in our hypothesis.

We now have a full generative model on $\boldsymbol{X}$.

6.2.2 Identifiability of the structure

The model presented above relies on a discrete structure S between the covariates. But to find it we need identifiability property to insure that we will asymptotically find the true model. Identifiability of the structure is asked in following terms: Is it possible to find another structure $\tilde{S}$ of linear regression between the covariates leading to the same joint distribution and marginal distributions?

If there are exact sub-regressions ($\exists j, \sigma^2_j = 0$), the structure won't be identifiable. But it only means that several structure will have the same likelihood and they will have the same interpretation. So it's not really a problem. Moreover, when an exact sub-regression is found, we can delete one of the implied variables without any loss of information and the structure will define a list of variable from which to delete.`CorReg` (Our R package) prints a warning to point out exact regressions when found. In the followings we suppose $\forall j, \sigma^2_j \neq 0$, then $\boldsymbol{X}^{I'_f}\boldsymbol{X}^{I_f}$ and $\boldsymbol{X}'\boldsymbol{X}$ are of full rank (but the later is ill-conditioned for small values of σ^2_j).

Our full generative model is a p-sized Gaussian mixture model of K distinct components and can be seen as a **SR** model defined by Maugis [Maugis et al., 2009]. In this section, S will denote the set of variable as in the paper from Maugis and we call Gaussian mixtures the Gaussian mixtures with at least two distinct components. The equivalence with Maugis's model is defined by: $\boldsymbol{X}^{I_r} = \boldsymbol{y}^{S^c}$ and $\boldsymbol{X}^{I_f} = \boldsymbol{y}^R$. We have supposed independence between variables in $\boldsymbol{X}^{I_f}$ so the identifiability theorem from Maugis tells that our model is identifiable if variables in $\boldsymbol{X}^{I_f}$ are Gaussian mixtures.

We define $\boldsymbol{X}^G \subsetneq \boldsymbol{X}^{I_f}$ containing Gaussian variables and we note the Gaussian mixtures $\boldsymbol{X}^{G^c} \neq \emptyset$ its complement in $\boldsymbol{X}^{I_f}$. We suppose that variables in $\boldsymbol{X}^{I_r}$ are all Gaussian mixtures. It implies that $\forall j \in I_r, \exists i \in I^j_f$ so that $\boldsymbol{X}^i \subset \boldsymbol{X}^{G^c}$ since any linear combination of Gaussian variable would only give a Gaussian (so each sub-regression contain at least one Gaussian mixture as a regressor).
We introduce the matricial notation $\boldsymbol{X}^{I_r} = \boldsymbol{X}^{I_f}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}$ where $\boldsymbol{\alpha}$ is the $(p - p_r) \times p_r$ matrix whose columns are the $\boldsymbol{\alpha}_j$ and $\boldsymbol{\varepsilon}$ is the $n \times p_r$ matrix whose columns are the $\boldsymbol{\varepsilon}_j$

The theorem from Maugis guarantee that a sub-regression between Gaussian mixtures is identifiable in terms of which one is regressed by others.

$$\begin{aligned}
\boldsymbol{X}_{r|\boldsymbol{X}^G, \boldsymbol{X}^{G^c}} &= \boldsymbol{X}^G\boldsymbol{\alpha}_G + \boldsymbol{X}^{G^c}\boldsymbol{\alpha}_{G^c} + \boldsymbol{\varepsilon} && (6.3)\\
\boldsymbol{X}_{r|\boldsymbol{X}^{G^c}} &= \boldsymbol{X}^{G^c}\boldsymbol{\alpha}_{G^c} + \tilde{\boldsymbol{\varepsilon}} \text{ is identifiable where} && (6.4)\\
\tilde{\boldsymbol{\varepsilon}}_j &= \boldsymbol{X}^G\boldsymbol{\alpha}^G_j + \boldsymbol{\varepsilon}_j \text{ is Gaussian.} && (6.5)
\end{aligned}$$

So a sufficient condition for identifiability is to have at least one Gaussian mixture in each sub-regression. It implies then that: $\forall j \in I_r, \boldsymbol{X}^j$ is a Gaussian mixture and $\exists i \in I^j_f, \boldsymbol{X}^i \subset \boldsymbol{X}^G$.

Remark: Identifiability of S is not necessary to use a given structure but helps to find it. In the followings, true S is supposed to be identifiable (at least one Gaussian mixture in each sub-regression).

6.2.3 Impact of the structure itself

Correlations can lead to serious problems in industrial context. Let's imagine a dataset with two highly correlated covariates. Estimators may give them the same coefficient (grouping effect when using elasticnet) or only keep one of them (like the LASSO does).

If only one is kept without explicitly pointing the correlations we will think that the deleted covariate is irrelevant even if it is not and even if it could be more relevant than the kept covariate in terms of possible actions or physical meanings.

On the other hand, keeping both without pointing correlations will lead to modify one of the covariates to impact the value of the response as we want and the other covariate will automatically move so we won't obtain the expected result.

Choosing whereas we have to use or not the marginal model as a pretreatment is independent of the utility of the structure. Knowing explicitly the complex correlations that hold the dataset is a real stake when times come to interpret the model and to decide actions. This is a strength of our method. We don't have to make a choice between grouping effect or variable selection because our explicit structure describes in details the complexity of the situation so we can then act knowing what we do, without having to blindly follow one of the two heuristics.

6.3 Sub-regression model selection

6.3.1 Bayesian criterion for quality

Our full generative generative model allows us to compare structures with criterions like the Bayesian Information Criterion (BIC) which penalize the log-likelihood of the joint law on $\boldsymbol{X}$ according to the complexity of the structure [Lebarbier and Mary-Huard, 2006].

We can also imagine to use other criterions, like the RIC (Risk Inflation Criterion [Foster and George, 19 that choose a penalty in $\log p$ instead of $\log n$ and thus gives more parsimonious models when p is larger than n (high dimension) or any other criterion [George and McCulloch, 1993] thought to be better in a given context. In the followings we use the BIC, that is more classical.

6.3.2 Penalization of the integrated likelihood by $P(S)$

When considering (6.1) we see that uniform law on $P(S)$ gives $\psi(\boldsymbol{X}, S) \propto P(\boldsymbol{X}|S)$ so it is equivalent to a minimization of the BIC. We note $\boldsymbol{\Theta}$ the set of the parameters of the generative model.

$$-2\log P(\boldsymbol{X}|S) \;\approx\; BIC = -2\mathcal{L}(\boldsymbol{X}, S, \boldsymbol{\Theta}) + |\boldsymbol{\Theta}|\log(n) \tag{6.6}$$

But BIC tends to give too complex structures because we test a great range of models and the number of model compared is not taken into account[Massart and Picard, 2007]. Thus we choose to penalise the complexity a bit more. We don't want to modify the BIC to keep its properties.

We have the explicit structure characterized by $S = \{I_f, I_r, p_f, p_r\}$, then we suppose a hierarchical uniform *a priori* distribution $P(S) = P(I_f|\boldsymbol{p}_f, I_r, p_r)P(\boldsymbol{p}_f|I_r, p_r)P(I_r|p_r)P(p_r)$ instead of the simple uniform law on S that is generally used and provides no penalty. It goes against the fact that the number of models with p_r sub-regressions and the number of possible combination for each sub-regression depends on p_r and thus provides distinct penalties according to the complexity. Thus we have :

$$BIC_+(X|S) = BIC(X|S) - \ln(P(S)) \tag{6.7}$$

The hierarchical uniform hypothesis gives:

$$P(S) = P(I_f|\boldsymbol{p}_f, I_r, p_r)P(\boldsymbol{p}_f|I_r, p_r)P(I_r|p_r)P(p_r) \text{ with} \tag{6.8}$$

$$P(p_r) = \frac{1}{p} \tag{6.9}$$

$$P(I_r|p_r) = \frac{1}{\binom{p}{p_r}} \tag{6.10}$$

$$P(\boldsymbol{p}_f|I_r, p_r) = \frac{1}{p_r \times \frac{1}{p-p_r}} = \frac{p-p_r}{p_r} \tag{6.11}$$

$$P(I_f|\boldsymbol{p}_f, I_r, p_r) = \frac{1}{\prod_{j\in I_r}\binom{p-p_r}{p_f^j}} \tag{6.12}$$

instead of $P(S) = \frac{1}{|\mathcal{S}_p|}$ as defined in section 4.2 for the classical uniform hypothesis. It increases penalty on complexity for $p_r \leq \frac{p}{2}$ and $p_f^j \leq \frac{p}{2}$ because probability of a complex model is under-estimated. Hence this constraint on $\hat{p}_r$ and $\hat{p}_f^j$ is given in the research algorithm when the Hierarchical Uniform hypothesis is made instead of Uniform one in numerical experiments (section 7 and 8). BIC_+ does not change BIC but only $P(S)$ so the properties of BIC_+ are the same as classical BIC but we obtain better results when the constraints on the complexity are verified.

6.3.3 Some indicators for proximity

The first criterion is $\psi(\boldsymbol{X}, S)$ which is maximized in the MCMC. But in our case, it is estimated by the likelihood (see (6.1))whose value don't have any intrinsic meaning. To show how far the found structure is from the true one in terms of S we define some indicators to compare the true model S and the found one $\hat{S}$. Global indicators :

- TL (True left) : the number of found dependent variables that really are dependent $TL = |I_r \cap \hat{I}_r|$
- WL (Wrong left) : the number of found dependent variables that are not dependent $WL = |\hat{I}_r| - TL$
- ML (Missing left) : the number of really dependent variables not found $ML = |I_r| - TL$
- Δp_r : the gap between the number of sub-regression in both model: $\Delta p_r = |I_r| - |\hat{I}_r|$. The sign defines if $\hat{S}$ is too complex or too simple compared to the true model
- $\Delta compl$: the difference in complexity between both model: $\Delta compl = \sum_{j\in p_r} p_f^j - \sum_{j\in \hat{p}_r} \hat{p}_f^j$

6.4 Neighbourhood

We note $\mathcal{S}_p$ the ensemble of feasible structures of size p (those uncrossed, *i.e.* with $I_f \cap I_r = \emptyset$). For each step q, starting from $S \in \mathcal{S}_p$ we define a neighbourhood:

$$\mathcal{V}_S \quad = \quad \{S\} \cup \{S^{(i,j)} | (i,j) \in \mathcal{A}_q\} \cap \mathcal{S}_p \tag{6.13}$$

where $\mathcal{A}_q$ is a set of arcs to modify (add or remove) in the associated graph, defined according to a strategy. And we have for given S and (i,j):

- if $i \in I_f^j$ then we remove i from I_f^j (arc removal)
- else we had i in I_f^j

Then coherence between I_f and others parts of S is done by $\forall 1 \leq j \leq p : p_f^j = |I_f^j|$, $I_r = \{j | p_f^j > 0\}$ and $p_r = |I_r|$.

In the adjacency matrix we just do: $G_{i,j} = 1 - G_{i,j}$. The main advantage of such a neighbourhood is that increasing and decreasing complexities are tested at each step without arbitrary ratio. If we just look at the sub-regression system, we have to choose for each sub-regression if we add, remove or keep covariates and we also have to choose if we had or delete some sub-regression. Adjacency matrix makes the neighbourhood extremely natural with just the modification of a value in a binary matrix.

6.4.1 Strategy

Many strategies can be imagined. First, we can decide to keep the local $\hat{S}$ in the neighbourhood or not, that is allowing or not stationarity. Here the MCMC is not used for sampling or density estimation. We just want to find the structure with the best value of $\psi(\boldsymbol{X}, S)$ so it is not an evidence to allow or not stationarity. Our package `CorReg` give the user the choice with stationarity, included in the neighbourhood by default.

The only constraint on $\mathcal{A}_q$ is that $\forall (i,j) \in \mathcal{A}_q, i \neq j$

We propose, for step q to draw j from $\mathcal{U}(\{1, \ldots, p\})$ and then

$$\mathcal{A}_{q|j} = \{(i,j) | i \neq j\} \tag{6.14}$$

Such a strategy can be interpreted as the uniform choice of a sub-regression to modify followed by the proposal of each possible unary change. Our package `CorReg` let the user choose many other strategies like a fixed number of random couples (i,j), or the union of the j^{th} line and column of G.

6.4.2 Active relaxation of the constraints

We have defined the neighbourhood with an intersection with $\mathcal{S}_p$. In practice, for some of the $(i,j) \in \mathcal{A}_q$, we have $S^{(i,j)} \notin \mathcal{S}_p$. Such candidates are basically rejected so the number of candidates is not constant at each step. Moreover, complex structures reduce the size of the potential neighbourhood because of the uncrossing rule. Thus we propose a relaxation method by a new definition of $S^{(i,j)}$:

- if $i \notin I_f^j$ (add):
  - $I_f^j = I_f^j \cup \{i\}$
  - $I_f^i = \emptyset$ (explicative variables can't depend on others : column-wise relaxation)

- $I_f = I_f \setminus \{j\}$ (dependent variables can't explain others : row-wise relaxation)

- else (remove): $I_f^j = I_f^j \setminus \{i\}$

It can be seen as forcing the modification by deleting what would make the structure not feasible. So in one step we can test a model that remove completely a sub-regression, remove the explicative role of a covariate in all sub-regression and create a new pairwise sub-regression. It drastically increases the scope of the neighbourhood and guarantee to always have the same number of candidates during the MCMC. It can be compared to simulated annealing that sometimes proposes exotic candidates to avoid local extrema, but here without any temperature to set. Here again, the neighbourhood remains natural, without arbitrary parameters to tune. Another advantage of the relaxation method is that it reduces complexity very quickly without having to deconstruct a sub-regression (Figure 6.4), so it helps to have simpler models in a small amount of time (asymptotical results are the same because the chain is regular thus ergodic).

6.5 The walk

Once we have a neighbourhood, we have to choose a candidate for the next step. The walk follows a time-homogeneous Markov Chain whose transition matrix $\mathcal{P}$ has $|\mathcal{S}_p|$ rows and columns (combinatory so we just compute the probabilities when we need them). At each step the Markov chain moves with probability:

$$\forall (S, \tilde{S}) \in \mathcal{S}^2 : \mathcal{P}(S, \tilde{S}) \quad = \quad \sum_{j=1}^{p} \mathbf{1}_{\{\tilde{S} \in \mathcal{V}_{S,j}\}} \frac{\exp(-\frac{1}{2}\psi(\boldsymbol{X}, \tilde{S}))}{\sum_{S_l \in \mathcal{V}_{S,j}} \exp(-\frac{1}{2}\psi(\boldsymbol{X}, S_l))} \tag{6.15}$$

And $\mathcal{S}_p$ is a finite state space.

Because the walk follows a regular and thus ergodic markov chain with a finite state space, it has exactly one stationary distribution [Grinstead and Snell, 1997] and the output will be the best structure in terms of $P(S|\boldsymbol{X})$ which weights each candidate. Practically speaking, `CorReg` returns the best structure seen during the walk (even if the corresponding candidate has never been chosen). The package also give the local structure when the walk stops so user can relaunch the algorithm from the same point if he wants to go further. The main criterion to stop the walk is a maximum number of iteration but `CorReg` can also stop the walk after a given number of step on the best model found. Numerical results (Section 4) illustrates the efficiency of the walk when the true model contains structures with various strength (section 7.2) and an example with a non-linear structure (Figure 7.47).

6.6 Initialization

6.6.1 Correlation-based initialization

If we have some knowledge about some sub-regressions (physical models for example) we can add them in the found and/or initial structure. So the model is really expert-friendly. The initial structure can be based on a first warming algorithm taking the correlations into account. Coefficients are randomly placed into I_f, according to a Bernoulli draw weighted by the absolute value of the correlations and with respect to the uncrossing constraint. Uncrossing constraint will not allow some strong correlation to be taken into account according to the order of the Bernoulli drawing so we can draw with a random order or by ordering by descending correlations.

We note that the BIC associated to initial model is often worse than the BIC of the void structure, so we compare several chains in Figures 6.1 and 6.2:

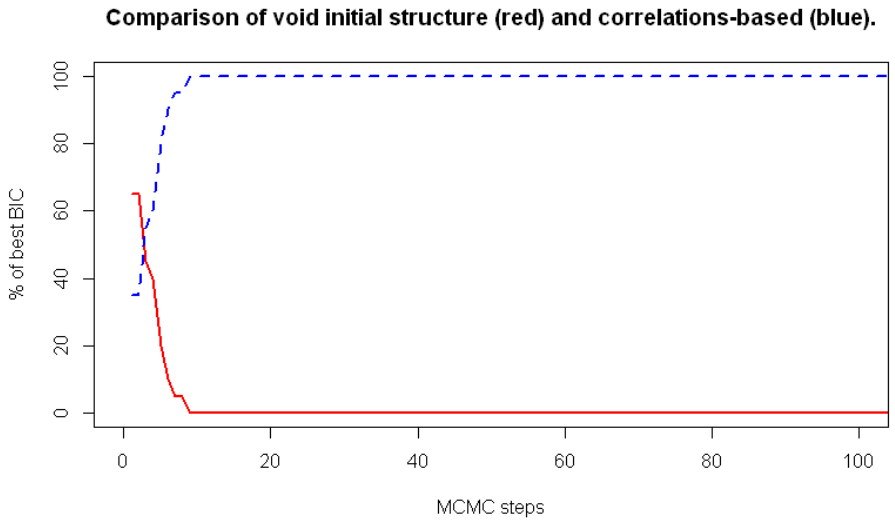


Figure 6.1: Amount of time each method is better for the 100 first steps of the MCMC.

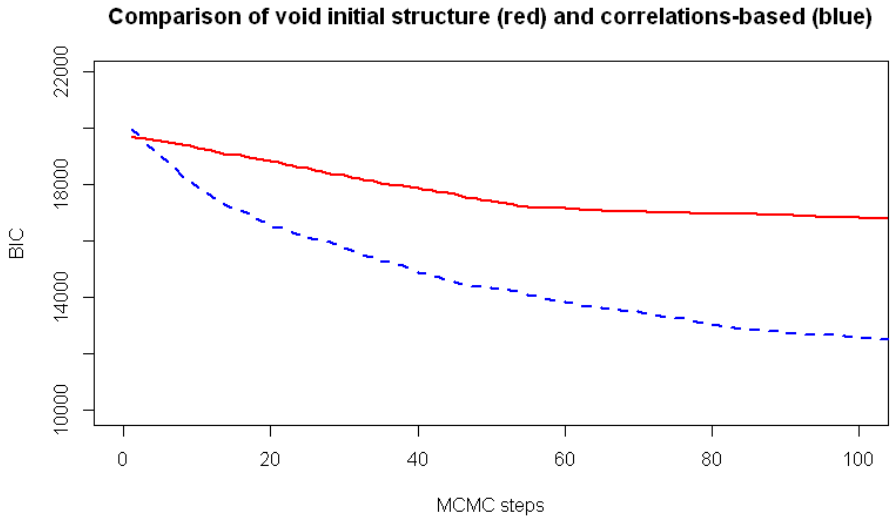


Figure 6.2: Evolution of the BIC (criterion to minimize in the MCMC) for each method.

We see that correlation-based initialization quickly beat the void structure. This can be explained by local extrema.

6.6.2 Multiple intialization

Local extrema are a known issue for most of optimization methods, and one would rather test multiple short chains than lose time in initialisation or long chains [Gilks et al., 1996]. We also compare the results obtained with several number of chains. Figure 6.3 shows the evolution of the BIC of the best chain with a number of chains varying from 1 to 10, so the model with 10 chains contain the others and is almost as good as they are. We see that multiple initialization

is efficient but the gain seems to be logarithmic in the number of tries so it is recommended to use multiple chains but not too much (time consuming). Important remark: multiple chains can be computed in parallel so it is not really time consuming.

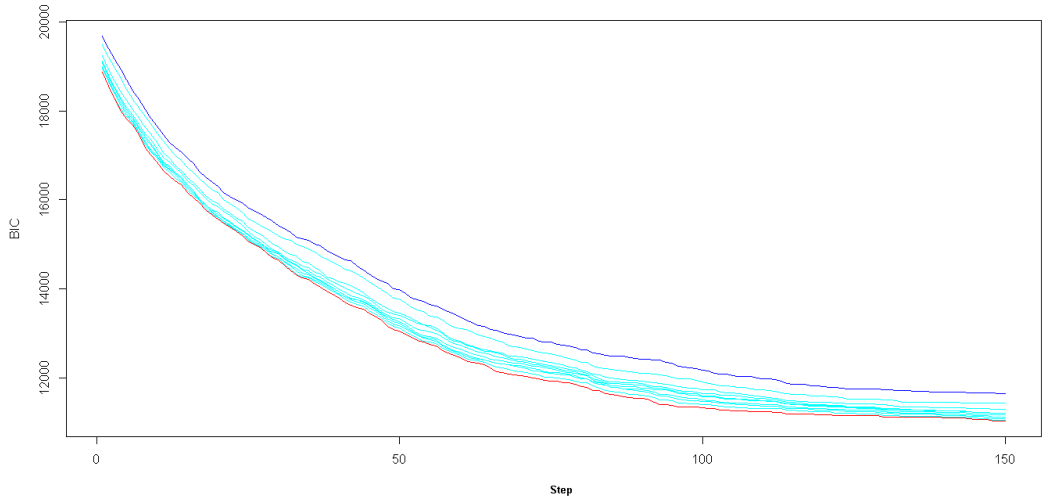


Figure 6.3: Comparison of distinct number of correlation-based initialisations for the MCMC.

In the followings, the chain was launched with twenty initialisations each time, based on the correlation matrix.

6.7 Pruning

If the complexity of S is too high, pruning methods can be used. We note that, for each of the following pruning methods, the final complexity may stay the same (for example if the MCMC had time to find a good model).

Variable selection

We can use variable selection methods like the LASSO on $\boldsymbol{X}^{I_f^j}$ to estimate the coefficients $\boldsymbol{\alpha}_j$ and obtain some supplementary zeros. Working on $\boldsymbol{X}^{I_f^j}$ protects the LASSO against dimension and correlations issues.

R^2 thresholding

We can also define a minimal value for the R^2 of the sub-regression to maintain them in the final structure. But this minimal value would be totally arbitrary and we know that it is frequent to use linear regression with real datasets that only show a R^2 between 0.1 and 0.2. It is particularly true in social sciences.

Test of hypothesis

Another pruning method would be to delete sub-regression that offer a F-statistic under a minimal value.

Additional cleaning steps

Because the walk is not exhaustive, it does make sense to let the walk continue a few steps with neighbourhood containing only suppressions in the structure. Every sub-graph of a bipartite graph is bipartite thus every sub-graph can be reached. It is just an heuristic change in the strategy with:

$$\mathcal{A}_q = \{(i,j)|i \in I_f^j\} \tag{6.16}$$

It is not based on any arbitrary parameter and change the result only if it founds a better structure in terms of the criterion ψ used in the walk. For these reasons, it is our recommended pruning method. The package `CorReg` allows to use this method automatically after the MCMC with the parameter `clean=TRUE` .

6.8 The Graphical LASSO

Graphical LASSO [Friedman et al., 2008] [Witten et al., 2011] [Tibshirani et al.,] [Friedman et al., 2010] is set to give undirectionnal (thus symmetric) graphs by selection in the precision matrix (the inverse of the variance-covariance matrix). It does make sense for exponential family because in these cases, zeros in the precision matrix Σ^{-1} can be interpreted in terms of conditional independence between covariates [Dempster, 1972]. But we have supposed Gaussian mixture on $\boldsymbol{X}$ and we search an oriented graph. However, we can still use it for initialization, for example by a Hadamard product with G_0 the adjacency matrix of the initial structure. We can also try to give the graph a bipartite orientation. We first have to obtain a bipartite graph, that mean to have no even cycles. A particular case would be the minimum spanning tree [Graham and Hell, 1985, Moret and Shapiro, 1991, Gower and Ross, 1969] because trees have no cycles. But it is time consuming and has no theoretical properties relied to our problematic of minimizing ψ, so the idea was left behind after some tries.

6.9 CorReg

The `CorReg` package is now on CRAN and provides many parameters for the walk. If wanted it can return some curves associated to the walk to have an idea of what happens with distinct strategies.

We define the complexity of a structure S as the number of elements in the adjacency matrix, that is the number of links between covariates and is obtained by:

$$\text{Complexity}(S) = \sum_{j \in I_r} p_f^j \tag{6.17}$$

We compare some walks with each time the same dataset and the same seed for the random generator. We have $p = 100$ and $n = 50$.

For Figures 6.4 and 6.5 we start from an arbitrary structure with a complexity of 62. We see that relaxation helps to delete these false sub-regressions and avoid to be stuck in it, improving the BIC much faster. We also observe that final complexities are comparable. Here the MCMC was launched only once (with the totally arbitrary initial structure based on nothing), the true structure had a complexity of 120.

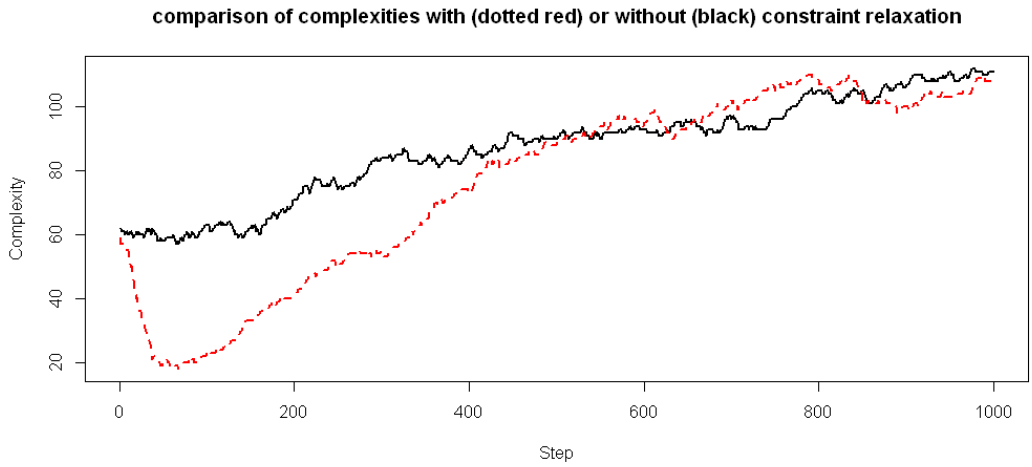


Figure 6.4: Comparison of complexity evolution with or without constraint relaxation.

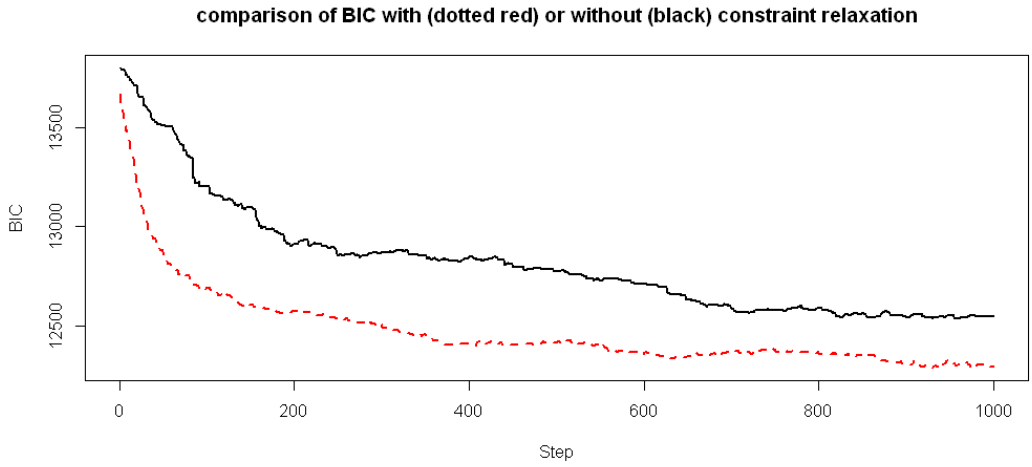


Figure 6.5: Comparison of BIC evolution with or without constraint relaxation.

Chapter 7

Numerical results on simulated datasets

7.1 The datasets

Now we have defined the model and the way to obtain it, we can have a look on some numerical results to see if `CorReg` keeps its promises. The `CorReg` package has been tested on the simulated datasets from section 5.2.2. Section 7.2 shows the results obtained in terms of $\hat{S}$. Section 7.3 shows the results obtained using only `CorReg`, or `CorReg` combined with other methods. The graph in section 7.3 give both mean, first and third quartiles of the chosen indicator. The MSE on $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{Y}}$ were computed on a validation sample of 1000 individuals. Several pattern for $\boldsymbol{Y}$ were tested to evaluate the impact of irrelevant covariates.

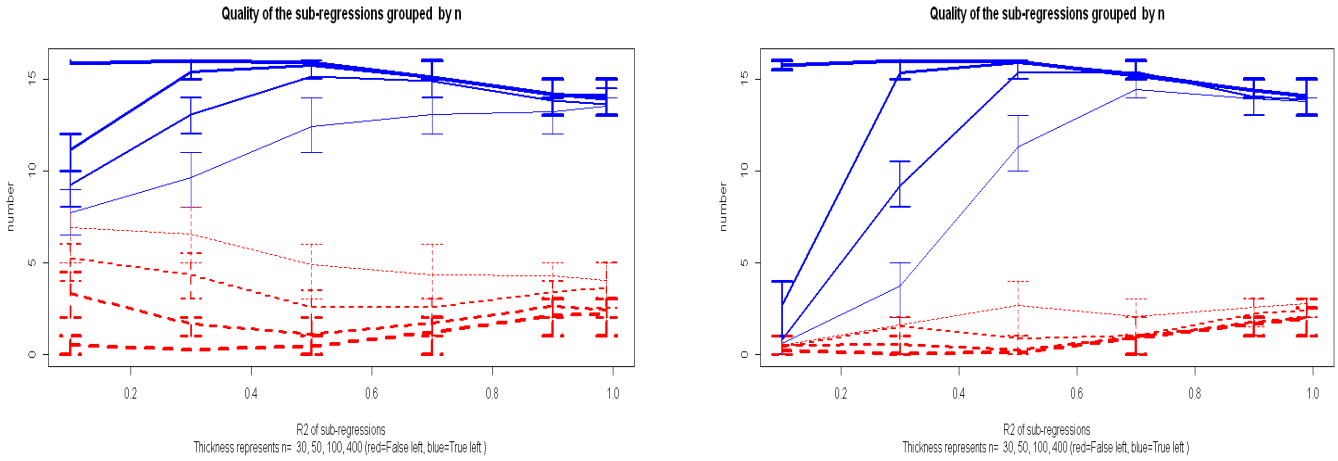
We used Rmixmod to estimate the densities of each covariate. For each configuration, the MCMC walk was launched on 10 initial structures with a maximum of 1 000 steps each time. When $n < p$, a frequently used method is the Moore-Penrose generalized inverse [Katsikis and Pappas, 2008], thus OLS can obtain some results even with $n < p$. We compare different methods with and without `CorReg` as a pretreatment. All the results are provided by the `CorReg` package.

7.2 Results on $\hat{S}$

Figure 7.1 illustrates the impact of large samples. For $n >> p$ the MCMC founds most of the truly redundant covariates and only few wrong redundant covariates. We also observe that strong correlations ($R^2 \geq 0.7$) get more wrong sub-regressions for a same total number of sub-regressions. It comes from induced correlations. If two covariates are explained by the same others, they may have a strong induced pairwise correlation and if the walk tries to combine them in a single sub-regression we can have a local extremum. The walk is ergodic but in a finite number of steps it can keep such a wrong sub-regression, that's why we launch the walk several times with distinct initial structures. Such a wrong sub-regressions is not totally wrong in that it describes real correlations. So interpretation is not compromise and neither is the predictive efficiency as shown in section 7.3.
For smaller values of n we observe that the number of true sub-regressions found increases with their strength (growing R^2).
When comparing BIC to BIC_+ it becomes evident that BIC_+ is less confident to keep sub-regressions (it is what it was made for). Weak sub-regressions are kept only if the sample is large enough to be confident and when the R^2 rises, the number of kept true sub-regressions

grows quickly whereas wrong sub-regressions remain exceptional. Induced pairwise correlations give weaker sub-regressions so the walk is less attracted by them. We can then conclude that BIC_+ does achieve its main purpose that was to reduce the complexity of the structure by keeping only strong sub-regressions. In these simulated datasets, the R^2 were equal for each sub-regression. We can see several reasons to explain why the sub-regression are not all kept or all missing.

- The walk has only walked a finite number of steps so only a subset of all the feasible structures has been tested.
- Some true sub-regressions are polluted by over-fitting and the non-crossing rule can then make other true sub-regressions not compatible (the walk has to clean the previous sub-regression first).
- ψ relies on the likelihood and if marginal laws are well-estimated by Mixmod, the gap between the marginal and dependent likelihood might be small and thus the walk can be slowed whereas we use a finite number of steps.



(a) With classical BIC criterion

(b) With our BIC_+ criterion

Figure 7.1: Quality of the subregressions found by the MCMC. True left (plain blue) and Wrong left (dotted red) for n varying in $(30, 50, 100, 400)$, the thicker the greater n.

7.3 Results on prediction

7.3.1 $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$

We try the method with a response variable depending on all covariates to compare the results with those from section 5.2.2 (same $\boldsymbol{X}$ and $\boldsymbol{Y}$). (`CorReg` reduces the dimension and can't give the true model if there is a structure).

We see that `CorReg` tends to give more parsimonious models and better predictions, even if the true model is not parsimonious. We logically observe that when n rises, all the models get better and the correlations cease to be a problem so the complete model starts to be better (`CorReg` does not allow the true model to be chosen). The main result here is that results based on $\hat{S}$ are still good so the MCMC is efficient enough to be useful for the study of the response variable $\boldsymbol{Y}$.

Results for OLS (Figures 7.2 to 7.4) are similar to those from section 5.2.2 excepted for small correlations because the MCMC using BIC_+ does not find the true structure for small correlations and a void structure gives a marginal model equal to the complete one.
This phenomenon is not observed with variable selection method (Figures 7.5 to 7.13) where covariates not deleted by the structure are deleted by the variable selection.
Ridge regression results (Figures 7.14 to 7.16) are also very similar to the previous (Figures 5.24 to 5.26).

Having simpler structure is important for interpretation so we keep this choice, but user can use classical BIC with single boolean parameter change.

Ordinary Least Squares when Y depends on all variables in X

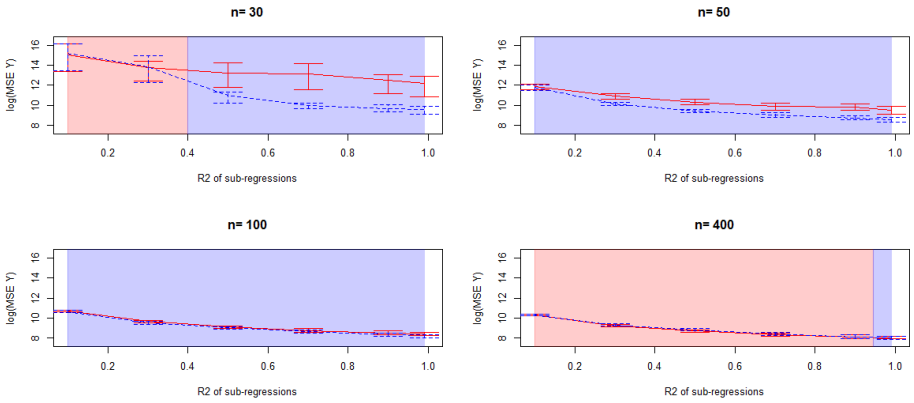


Figure 7.2: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

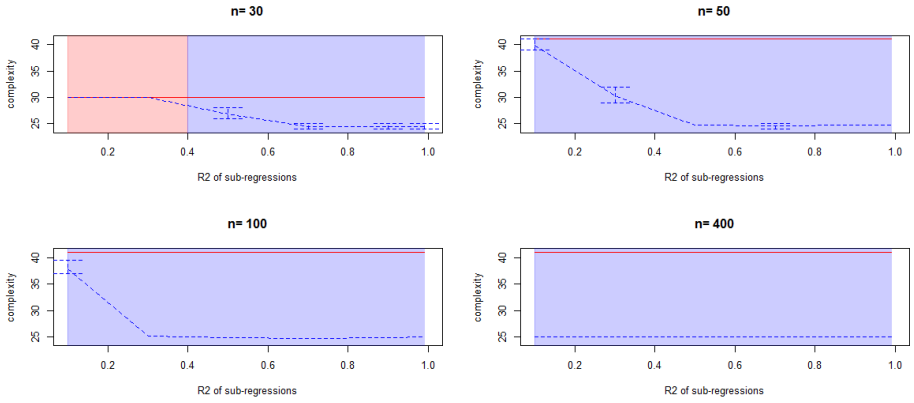


Figure 7.3: Comparison of the complexities, red=classical (complete) model, blue=marginal model

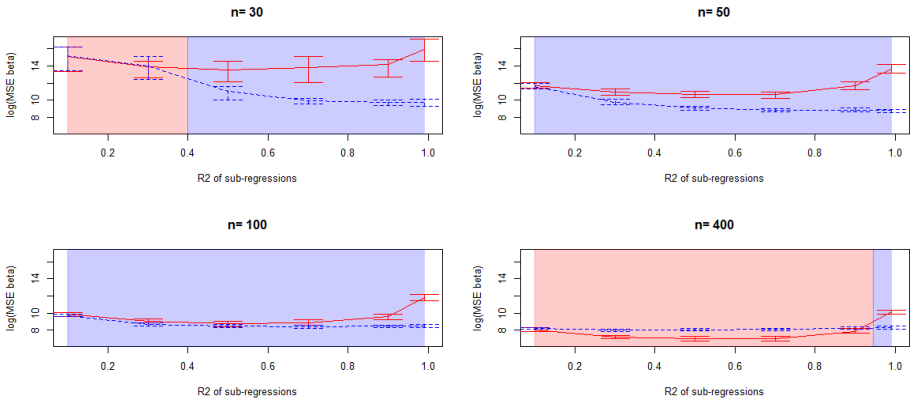


Figure 7.4: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

LASSO when Y depends on all variables in X

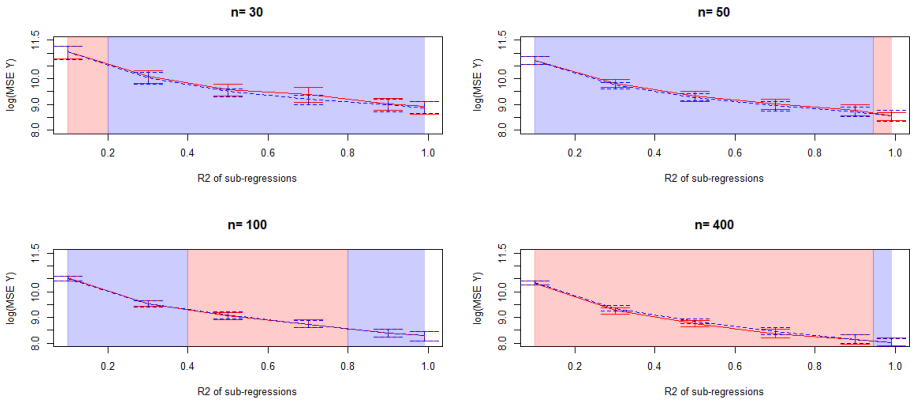


Figure 7.5: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

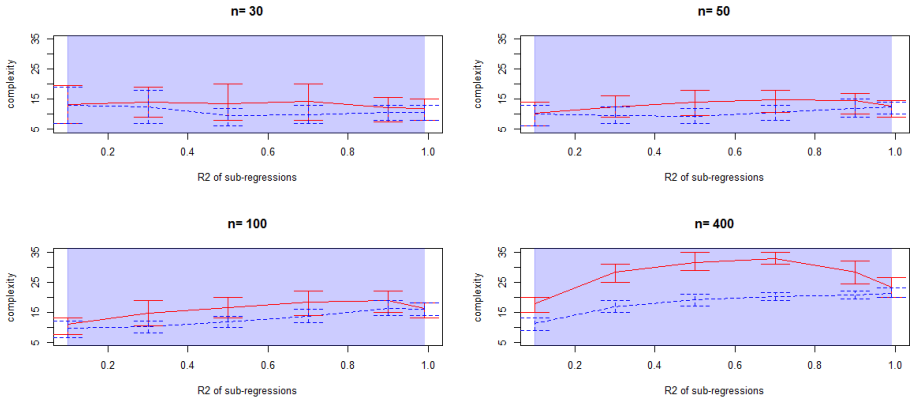


Figure 7.6: Comparison of the complexities, red=classical (complete) model, blue=marginal model

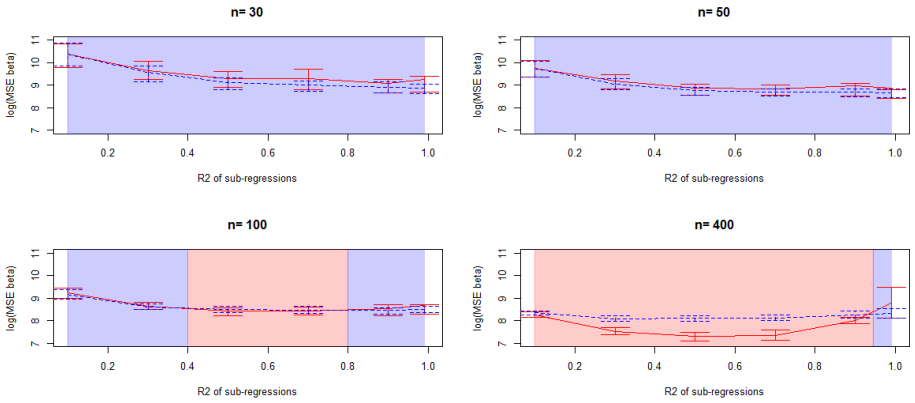


Figure 7.7: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Elasticnet when Y depends on all variables in X

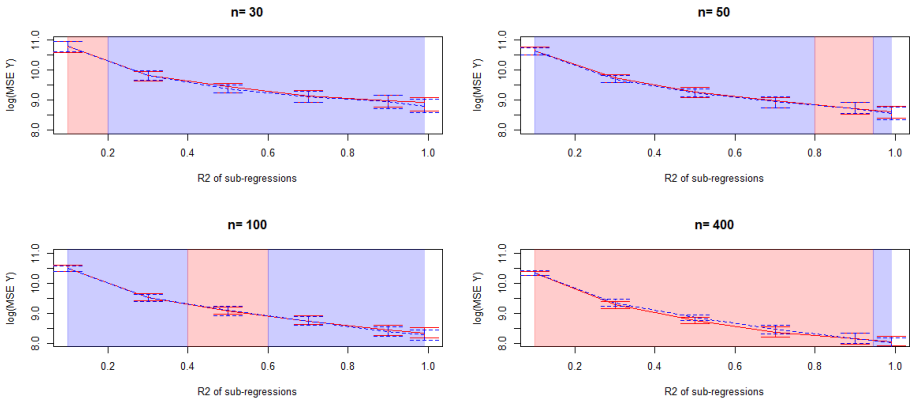


Figure 7.8: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

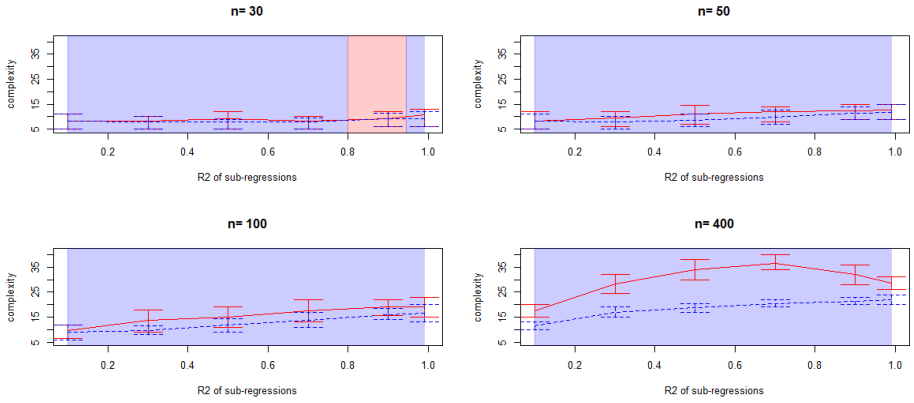


Figure 7.9: Comparison of the complexities, red=classical (complete) model, blue=marginal model

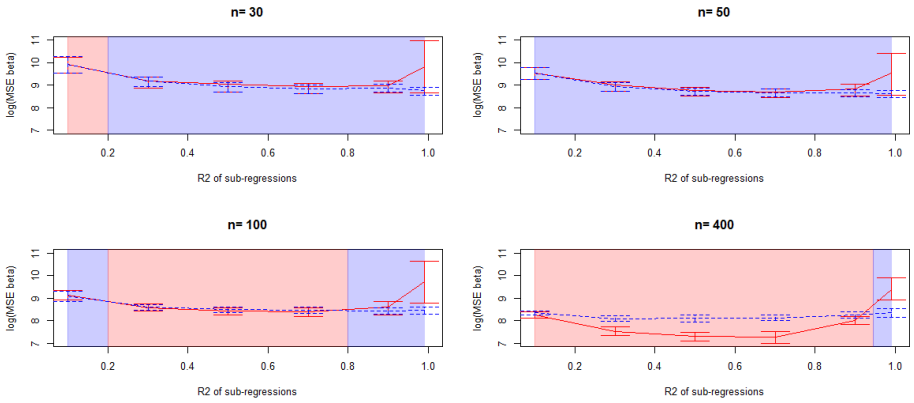


Figure 7.10: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Stepwise when Y depends on all variables in X

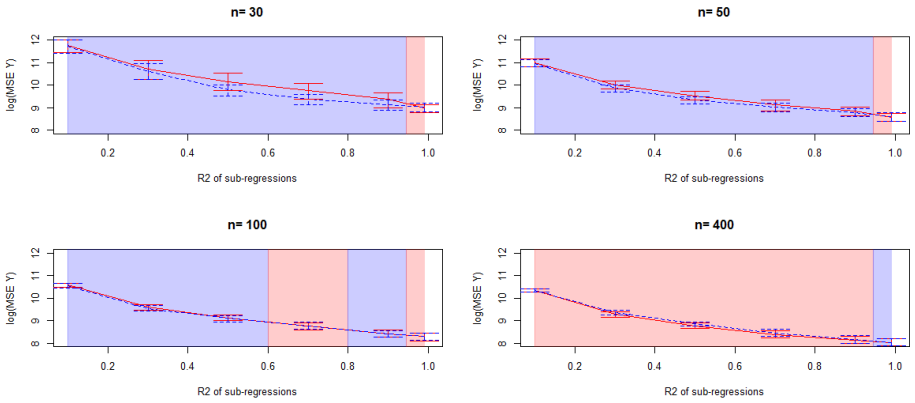


Figure 7.11: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

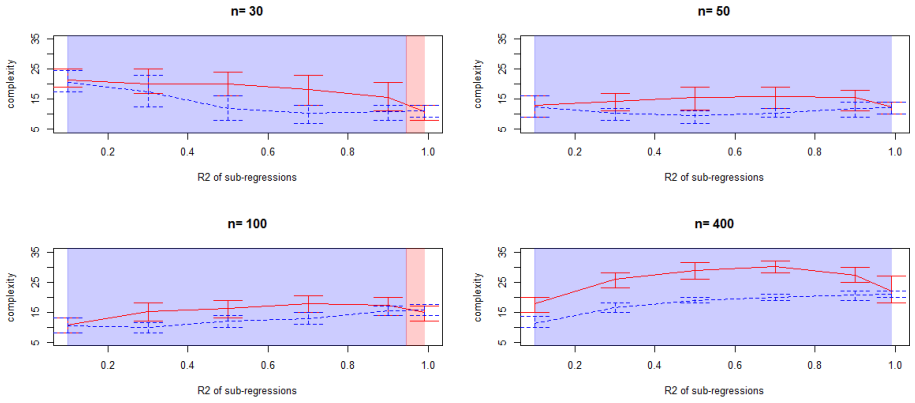


Figure 7.12: Comparison of the complexities, red=classical (complete) model, blue=marginal model

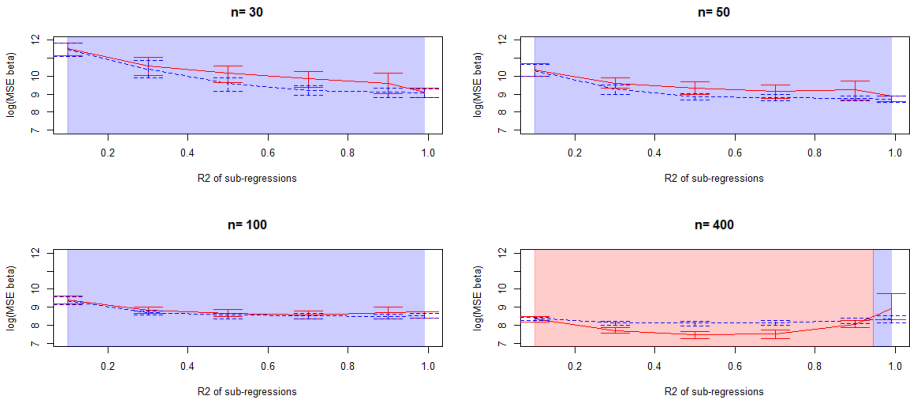


Figure 7.13: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Ridge regression when Y depends on all variables in X

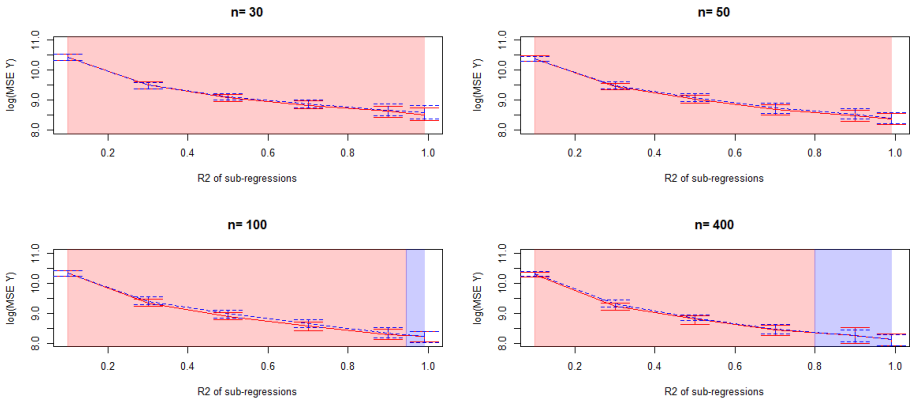


Figure 7.14: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

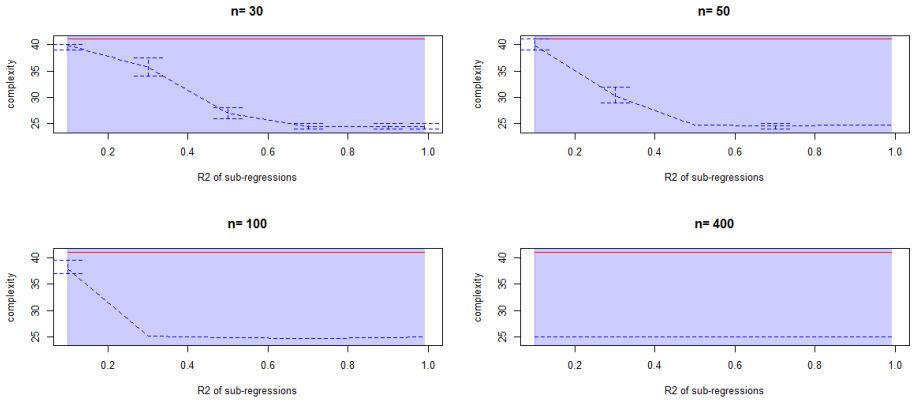


Figure 7.15: Comparison of the complexities, red=classical (complete) model, blue=marginal model

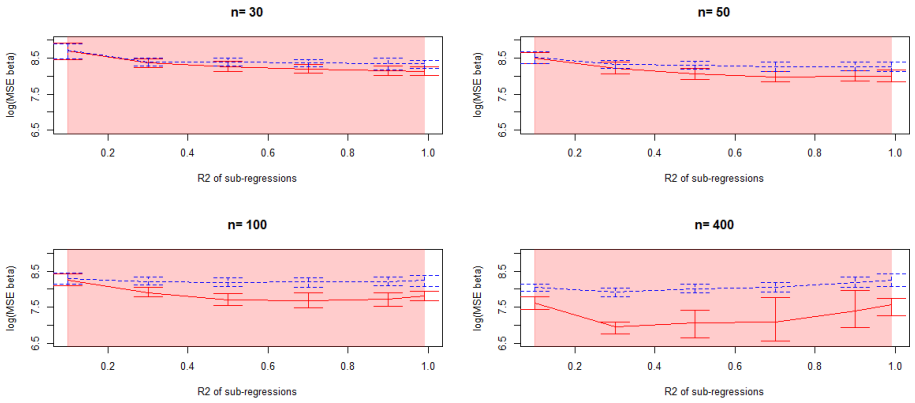


Figure 7.16: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

7.3.2 $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X}^{\boldsymbol{I_f}}$ (best case for us)

We want to see what happens in real life, when some covariates are irrelevant to describe $\boldsymbol{Y}$ according to the given dataset. We could generate $\boldsymbol{Y}$ with a random subset of $\boldsymbol{X}$ but in such a case, it would be impossible to say whether results come from sparsity or from the ratio of covariates in the subset that are in $\boldsymbol{X}^{I_r}$. Moreover, we will study real datasets in the next chapter so the only pattern to test here are those with some irrelevant covariates and relevant covariates only in one part of the partition on $\boldsymbol{X}$.
We start with $\boldsymbol{Y}$ depending only on covariates in $\boldsymbol{X}^{\boldsymbol{I_f}}$. It is the best case for us because our marginal model is then the true model and the complete model will need variable selection to reach the truth. Here $\boldsymbol{Y}$ depends on the 24 covariates in $\boldsymbol{X}^{I_f}$ with an intercept.
Smaller dimension makes the coefficients easier to learn and we observe that MSE are smaller for both model with any method compared to those from section 7.3.1.
For OLS (Figures 7.17 to 7.19) we note the global improvement of the MSE but also a specific improvement for large values of n where our marginal model resists to the complete model. It is logical because the complete model tends to reduce the coefficients associated to irrelevant covariates whereas our marginal delete them.

When looking at variable selection methods we also have this improvement so it confirm the already observed fact that variable selection method are theoretically able to find the true model but efficiency is not really great when confronted to correlated covariates. There is no surprise here after the results for $\boldsymbol{Y}$ depending on the whole dataset $\boldsymbol{X}$.

Ridge regression (figures 7.17 to 7.19) is finally improved here by our pretreatment by selection, like if we had added variable selection feature to the ridge regression. It is the method that provides the best results, but only because $\boldsymbol{Y}$ depends on all covariates in $\boldsymbol{X}^{I_f}$. Our pretreatment is limited in terms of variable selection.

Ordinary Least Squares when Y depends only on covariates in X^{I_f}

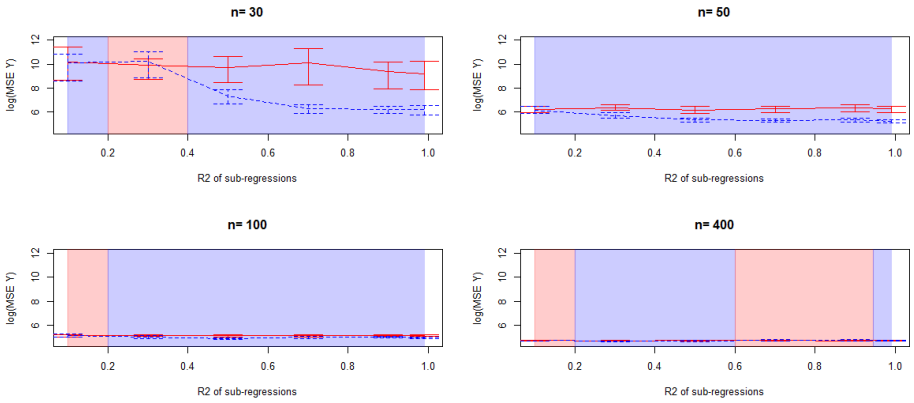


Figure 7.17: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

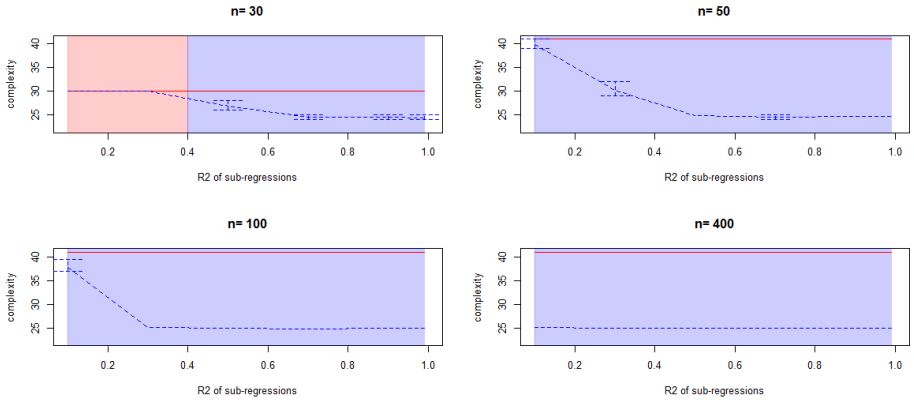


Figure 7.18: Comparison of the complexities, red=classical (complete) model, blue=marginal model

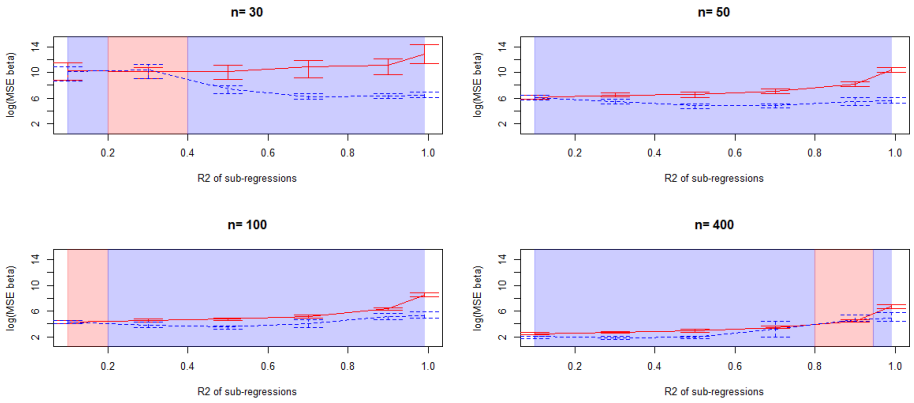


Figure 7.19: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

LASSO when Y depends only on covariates in X^{I_f}

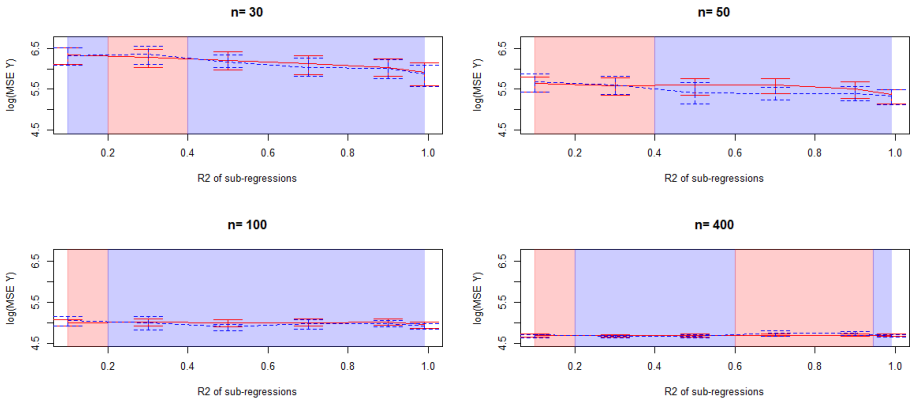


Figure 7.20: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

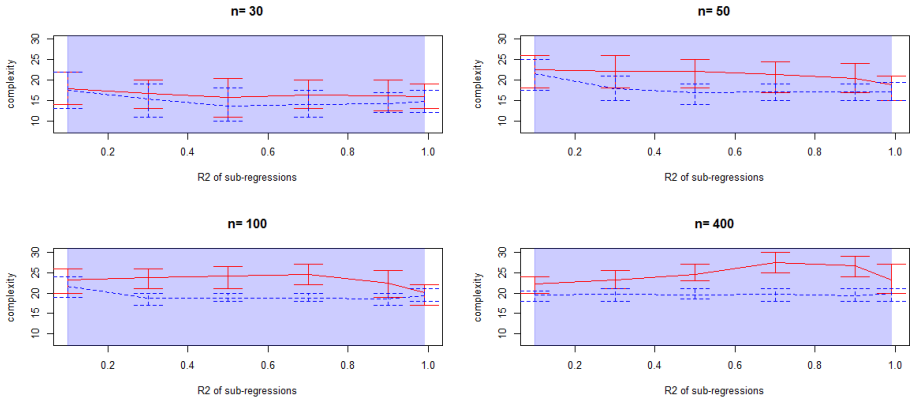


Figure 7.21: Comparison of the complexities, red=classical (complete) model, blue=marginal model

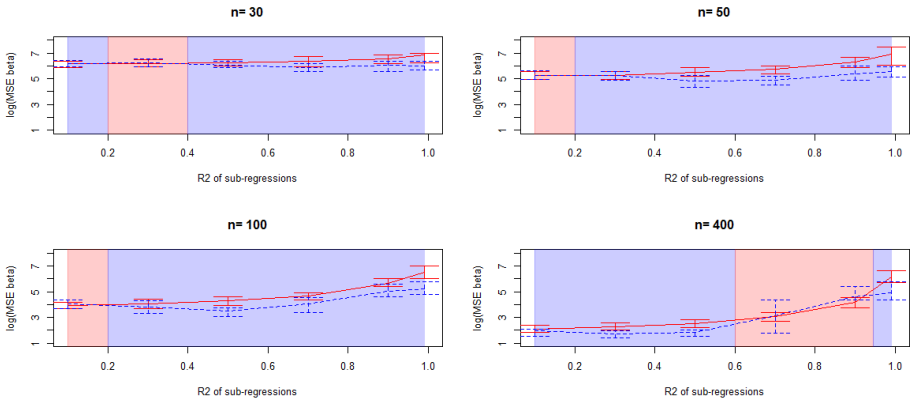


Figure 7.22: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Elasticnet when Y depends only on covariates in X^{I_f}

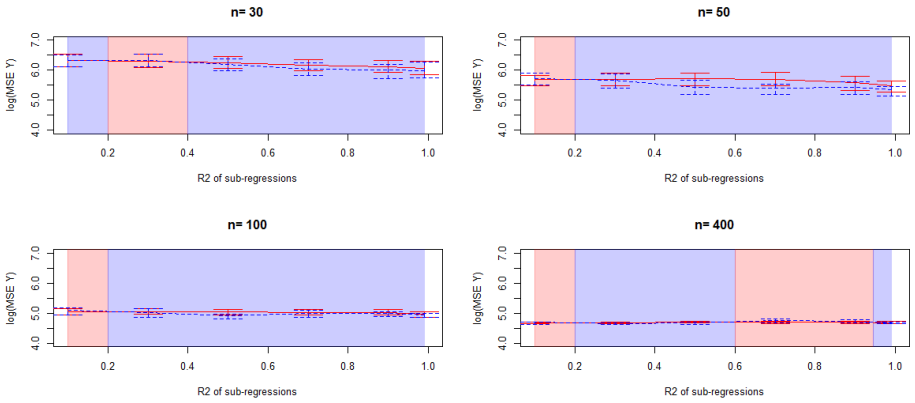


Figure 7.23: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

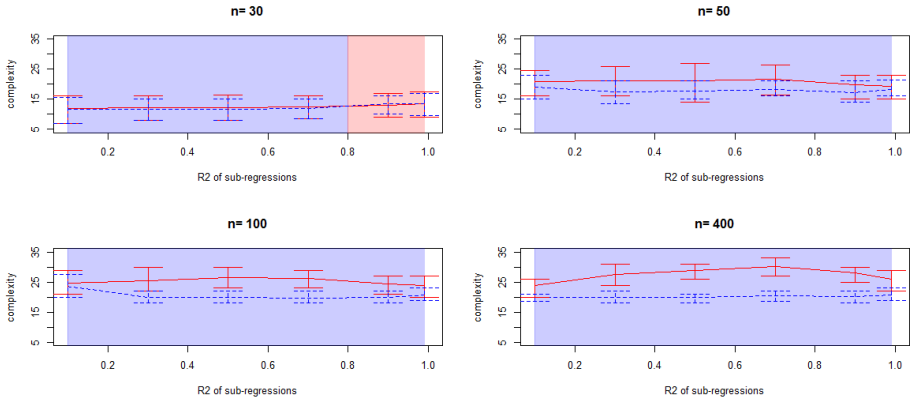


Figure 7.24: Comparison of the complexities, red=classical (complete) model, blue=marginal model

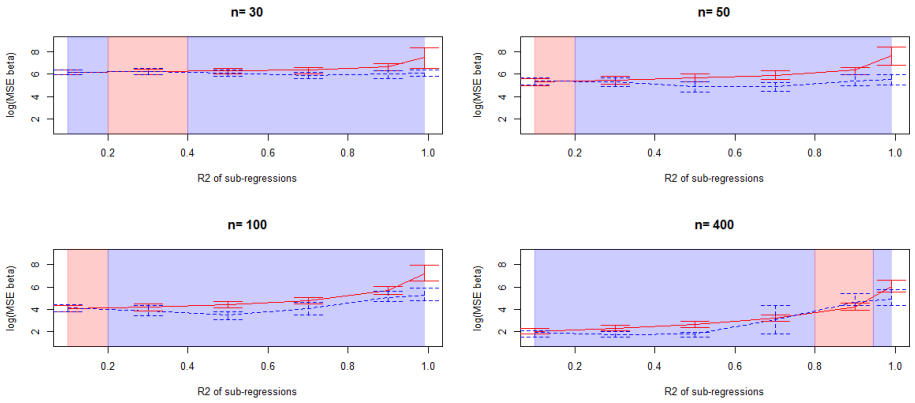


Figure 7.25: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Stepwise when Y depends only on covariates in X^{I_f}

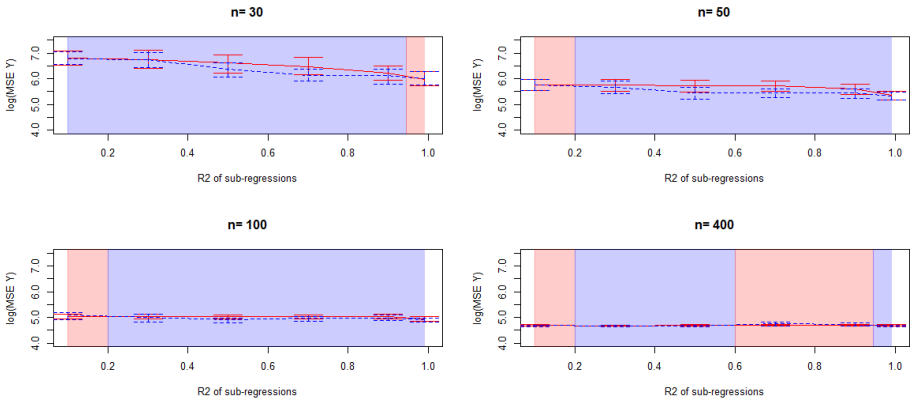


Figure 7.26: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

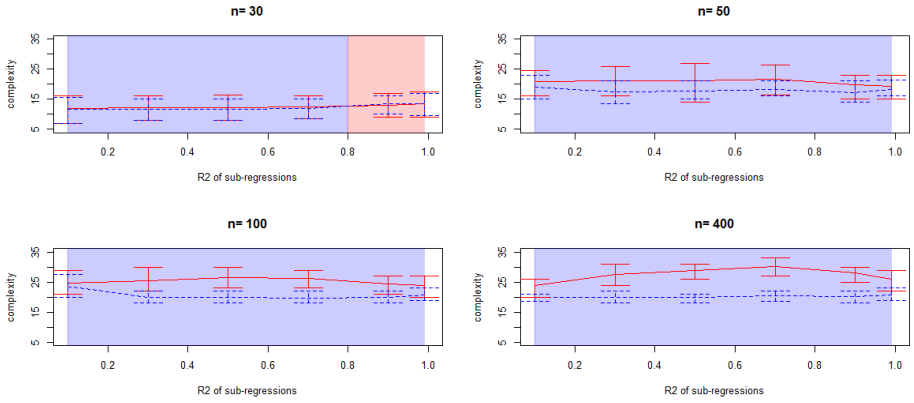


Figure 7.27: Comparison of the complexities, red=classical (complete) model, blue=marginal model

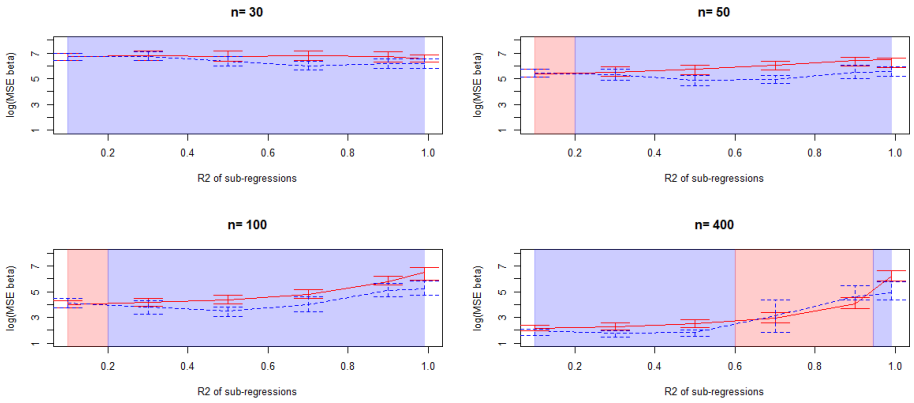


Figure 7.28: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Ridge regression when Y depends only on covariates in X^{I_f}

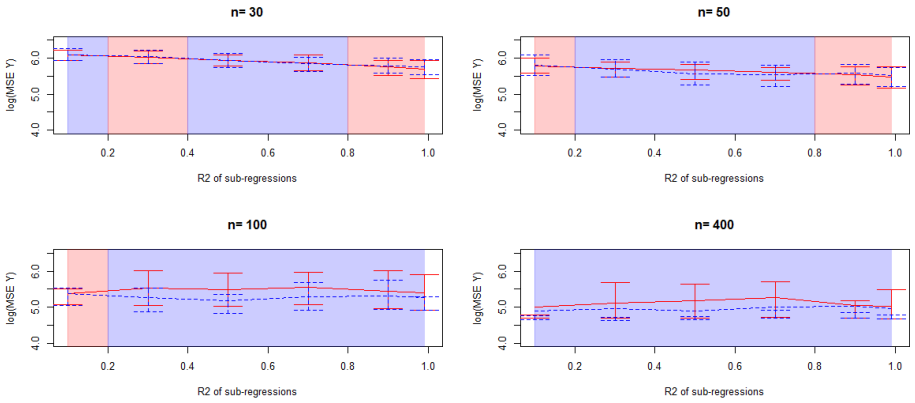


Figure 7.29: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

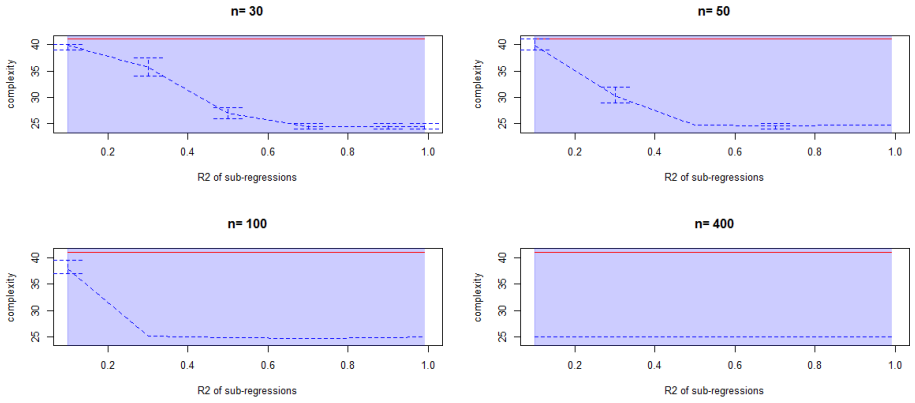


Figure 7.30: Comparison of the complexities, red=classical (complete) model, blue=marginal model

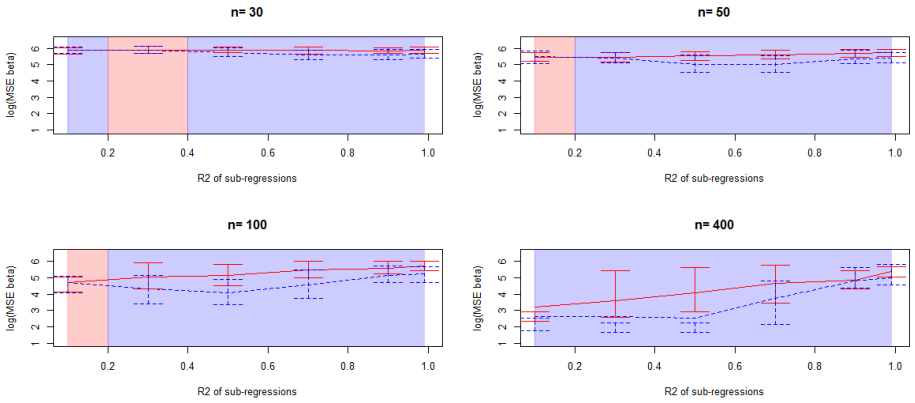


Figure 7.31: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

7.3.3 $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X}^{I_r}$ (worst case for us)

We now try the method with a response depending only on variables in $\boldsymbol{X}^{I_r}$. The datasets used here were still the same. Depending only on $\boldsymbol{X}_r$ implies sparsity and impossibility to obtain the true model when using the true structure. We get unbiased models but with an increase in the variance as described in equation 4.14.
In such a case, usage of BIC_+ instead of BIC is more obvious because each additional sub-regression deletes a covariate in our marginal model and reduces the probability to find the true model.

We first look at OLS (Figures 7.32 to 7.34) and see that we still obtain better results for small values of n or strong correlations. In real studies we will never know the true model but we can be confident that if correlations are strong or if sample is small, using our marginal model can helps whatever the true model is. This is a really powerful result. Improvement for small correlations but $n < p$ comes from dimension reduction. When you don't have enough individual it becomes better to use a small model that does not contain the true one but only covariates correlated to the relevant one instead of trying to work with all the covariates. Let's remember that OLS confronted to $n < p$ only delete covariates to have $n = p$ (or $p + 1$ when there is an intercept). QR decomposition leads to delete the last covariates in the dataset but in our simulations, covariates in $\boldsymbol{X}^{I_r}$ are placed randomly in the dataset so deletion by QR can be seen as random deletion. The gain implied by dimension reduction remains for $n > p$ if correlations are high enough because the matrix to invert is ill-conditioned and OLS needs a lot of individuals to reduce the variance of the estimator. Correlations really put OLS in trouble and our marginal model seems to be a good solution.

Variable selection methods still are impacted by correlations but not enough to be improved by our marginal model. Neither is the ridge regression.
Real datasets will provide $\boldsymbol{Y}$ depending on a mix of covariates from both $\boldsymbol{X}^{I_f}$ and $\boldsymbol{X}^{I_r}$ so our marginal could help. We also have to recall that the structure S is useful by itself to have a better comprehension of the dataset and help the final client to be confident in statistical tools because he sees small models that are known to be true and were found automatically by the method. `CorReg` also has a psychological impact on a study that should not be overlooked. Once $\hat{S}$ is found, trying the marginal model has no cost and should be tested.

Ordinary Least Squares when Y depends only on covariates in X^{I_r}

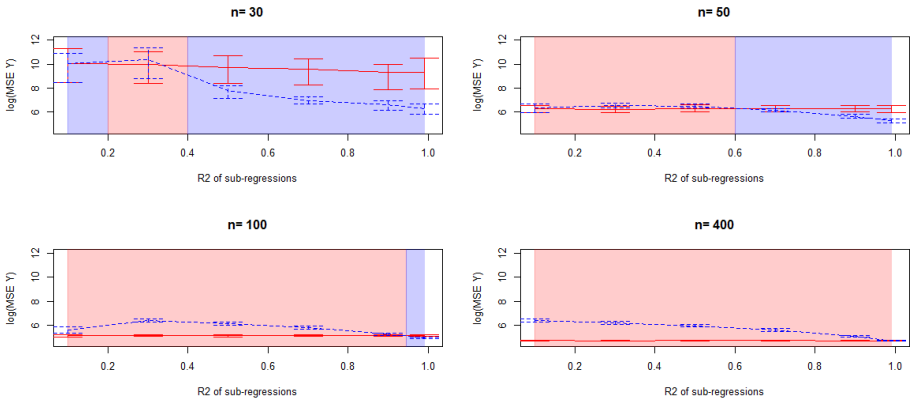


Figure 7.32: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

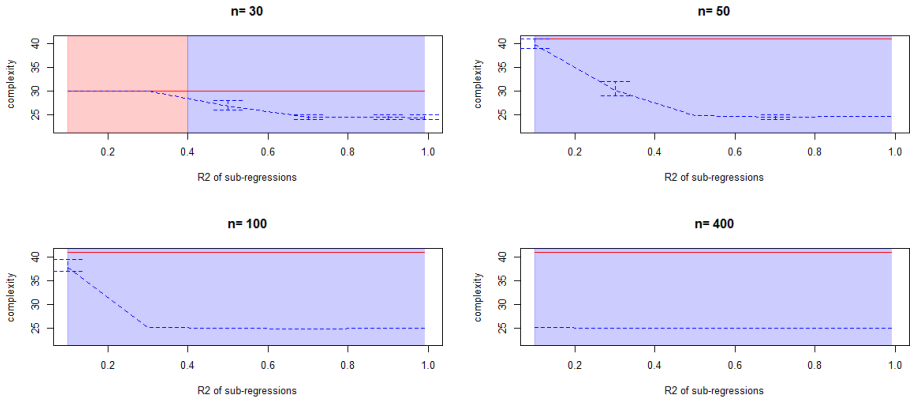


Figure 7.33: Comparison of the complexities, red=classical (complete) model, blue=marginal model

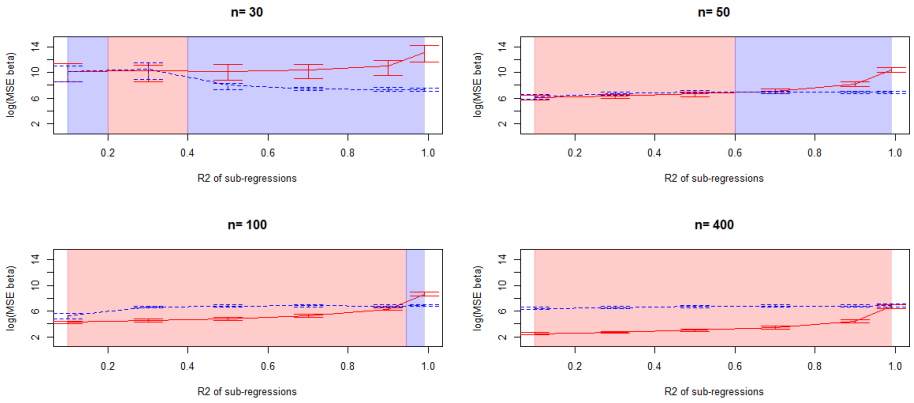


Figure 7.34: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

LASSO when Y depends only on covariates in X^{I_r}

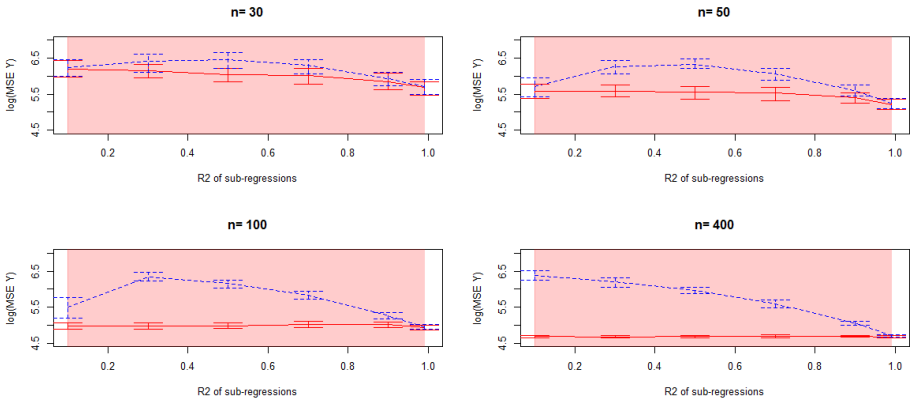


Figure 7.35: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

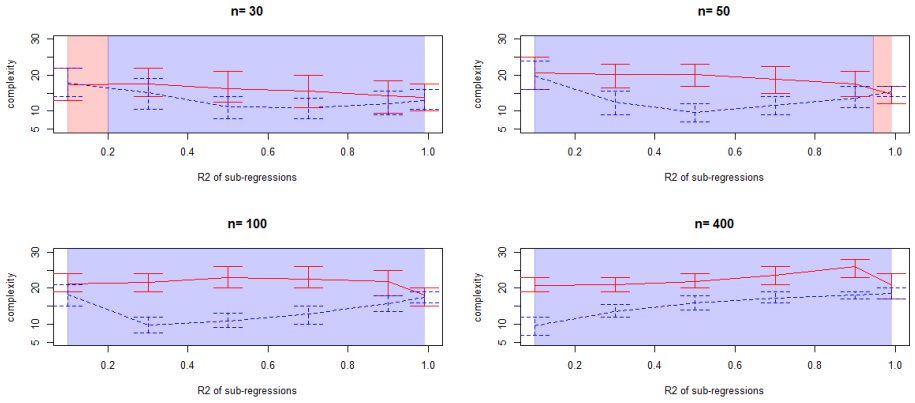


Figure 7.36: Comparison of the complexities, red=classical (complete) model, blue=marginal model

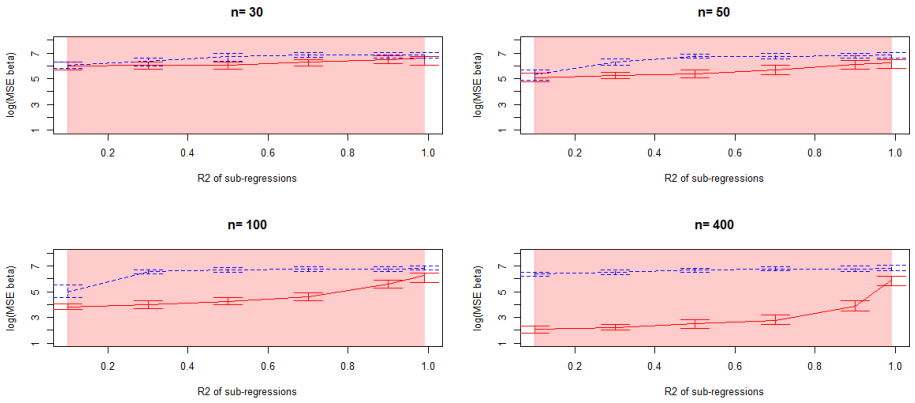


Figure 7.37: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Elasticnet when Y depends only on covariates in X^{I_r}

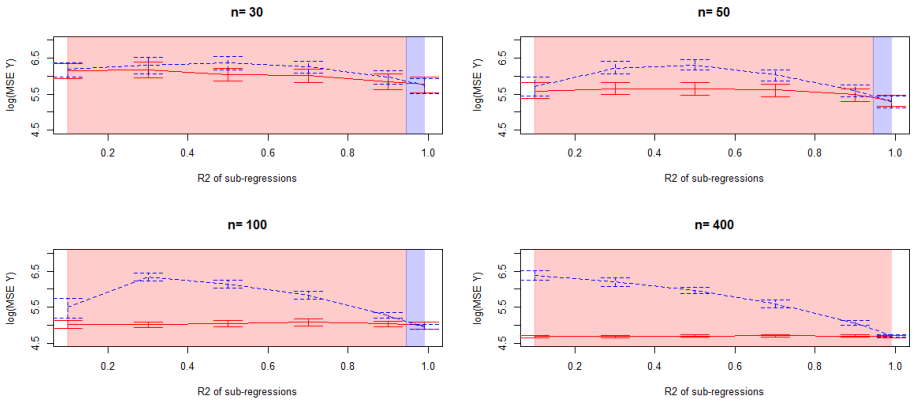


Figure 7.38: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

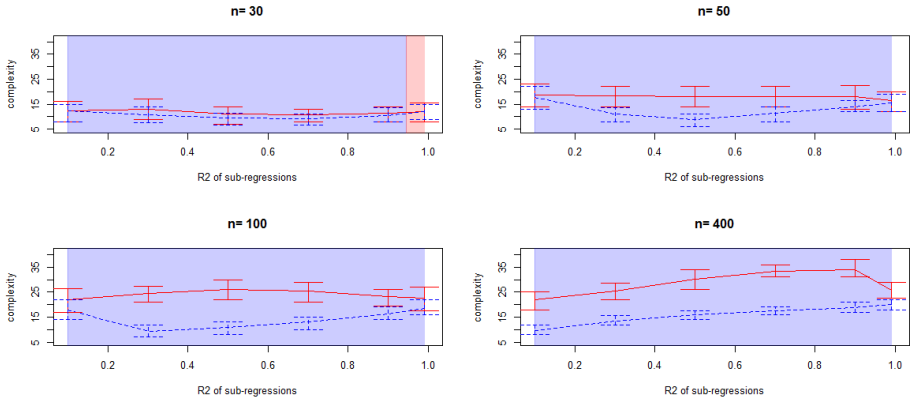


Figure 7.39: Comparison of the complexities, red=classical (complete) model, blue=marginal model

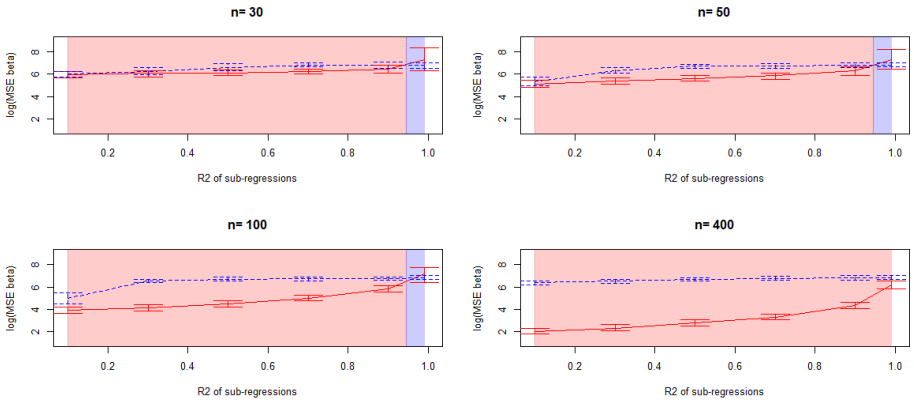


Figure 7.40: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Stepwise when Y depends only on covariates in X^{I_r}

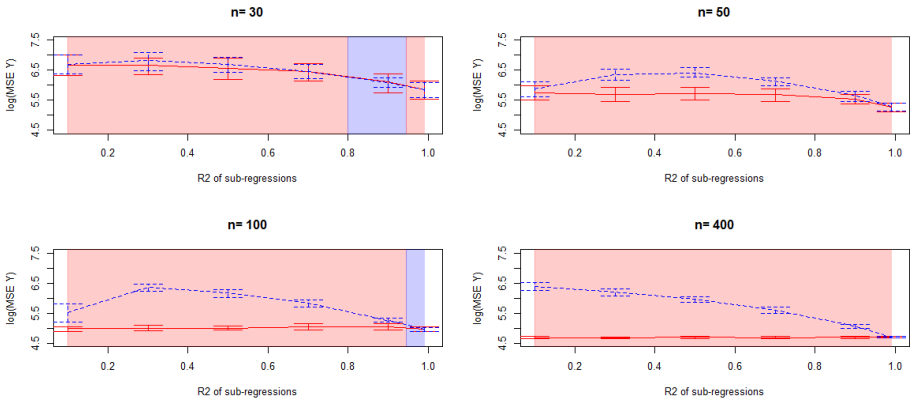


Figure 7.41: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

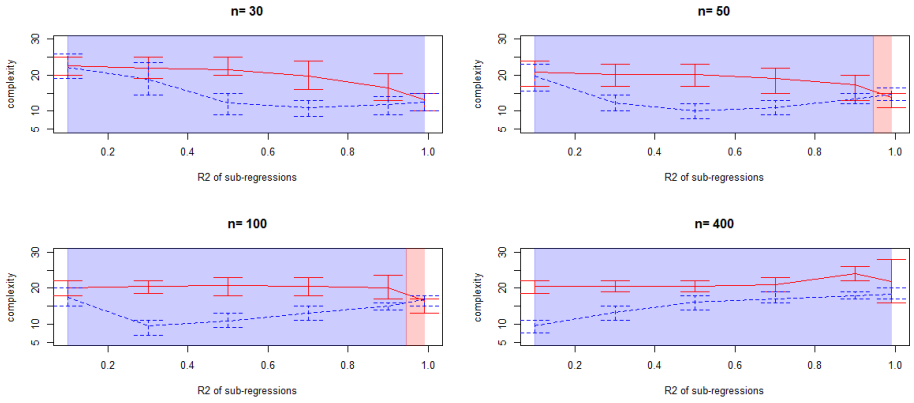


Figure 7.42: Comparison of the complexities, red=classical (complete) model, blue=marginal model

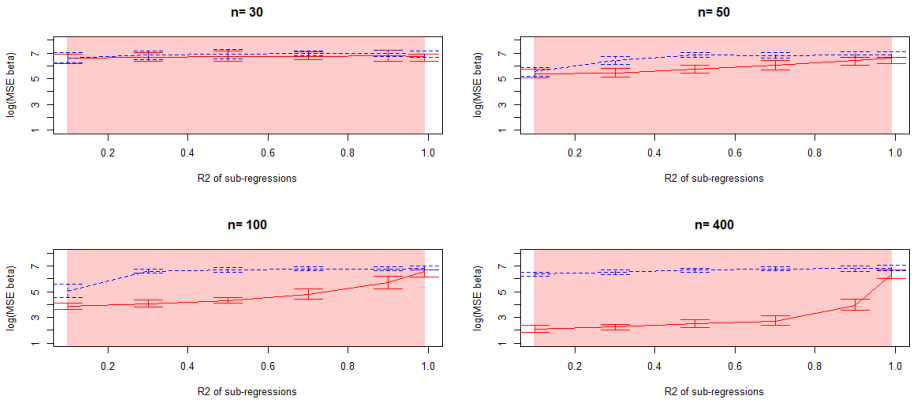


Figure 7.43: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

Ridge regression when Y depends only on covariates in X^{I_r}

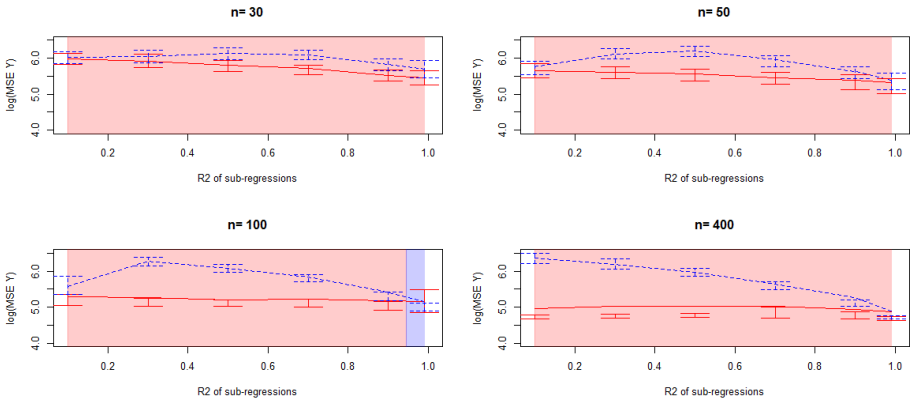


Figure 7.44: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model

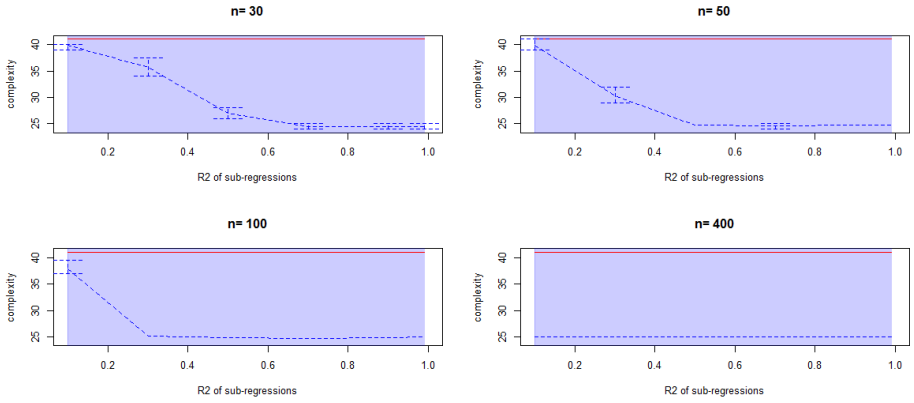


Figure 7.45: Comparison of the complexities, red=classical (complete) model, blue=marginal model

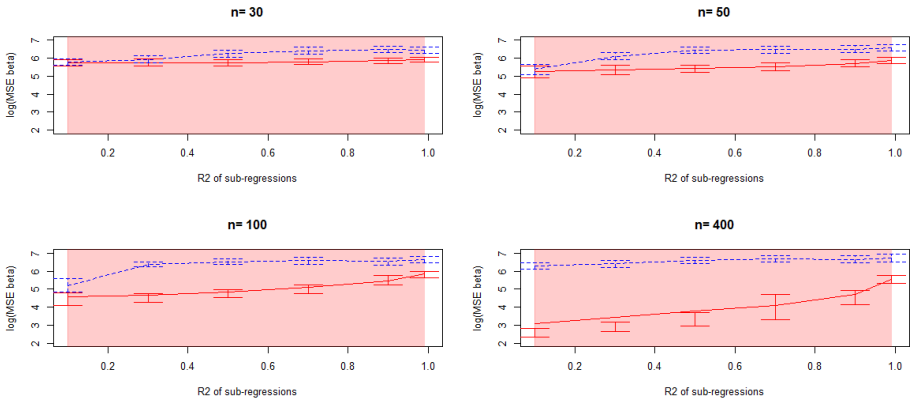


Figure 7.46: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model

7.3.4 Robustness with non-linear case

We have generated a non-linear structure to test the robustess of the model. $\boldsymbol{X}^{I_f}$ is a set of 6 independent Gaussian mixtures defined as previously but with random signs for the components means. $\boldsymbol{X}^{I_r} = \boldsymbol{X}_7 = a\boldsymbol{X}_1^2 + \boldsymbol{X}_2 + \boldsymbol{X}_3 + \varepsilon$. The matrix $\boldsymbol{X}$ is then scaled before doing

$$\boldsymbol{Y} = \sum_{i=1}^{7} \boldsymbol{X}_i + \varepsilon_Y.$$

We let a vary between 0 and 10 to increase progressively the non-linear part of the sub-regression. Once again, simulations has been made 100 times and the MSE were computed with 1 000 individuals validation samples.

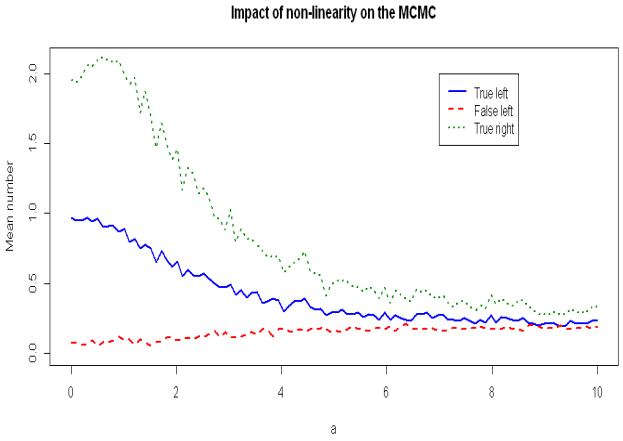


Figure 7.47: Evolution of the quality of $\hat{S}$ when the paramater a increases

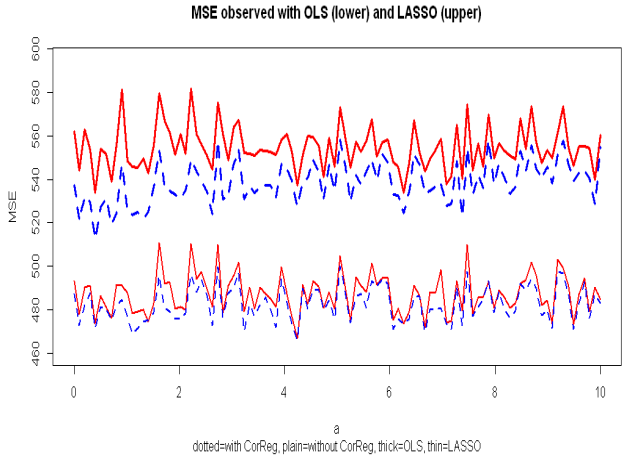


Figure 7.48: MSE on the main regression for OLS(thick) and LASSO (thin) used both with (plain) or without `CorReg` (dotted).

Figure 7.48 illustrates the advantage of using `CorReg` even with non-linear structures. Figure 7.47 shows that the MCMC have more difficulties to find a linear structure as the non-linear part of the sub-regression increases but the model is quite robust (efficient for small values of a).

Chapter 8

Numerical results on real datasets

8.1 Quality case study

This work takes place in steel industry context, with quality oriented objective : to understand and prevent quality problems on finished product, knowing the whole process. The correlations are strong here (many parameters of the whole process without any a priori and highly correlated because of physical laws, process rules, *etc.*).

We have :

- a quality parameter (confidential) as response variable,
- 205 variables from the whole process to explain it.

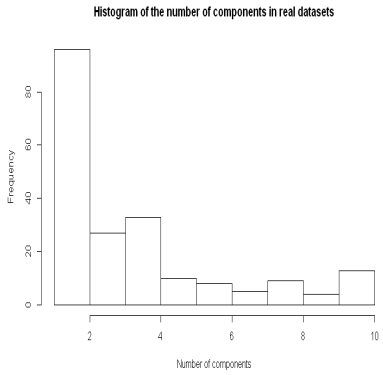


Figure 8.1: Distribution of the number of components found for each covariate.

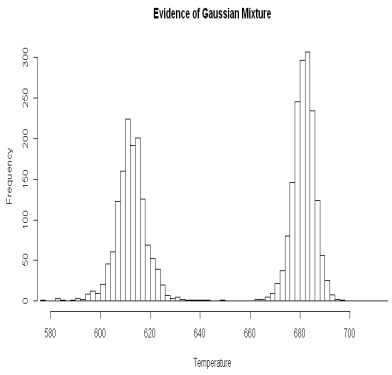


Figure 8.2: Example of non-Gaussian real variable easily modeled by a Gaussian mixture.

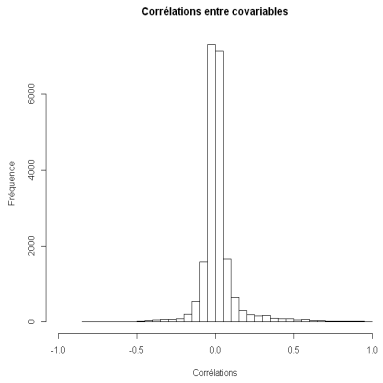


Figure 8.3: Histogram of correlations in $\boldsymbol{X}$.

We get a training set of $n = 3000$ products described by $p = 205$ variables from the industrial process and a validation sample of 847 products. Let's note ρ the absolute value of correlations between two covariates. Industrial variables are naturally highly correlated as the width and the weight of a steel slab ($\rho = 0.905$), the temperature before and after some tool ($\rho = 0.983$), the roughness of both faces of the product ($\rho = 0.919$), a mean and a max ($\rho = 0.911$).

The objective here is not only to predict non-quality but to understand and then avoid it. `CorReg` provides an automatic method without any a priori and is combined with variable selection methods. So it allows to obtain in a small amount of time some indications on the source of the problem, and to use human resources efficiently. When quality crises occurs,

time is extremely precious so automation is a real stake. The combinatorial aspect of the sub-regression models makes it impossible to do manually.

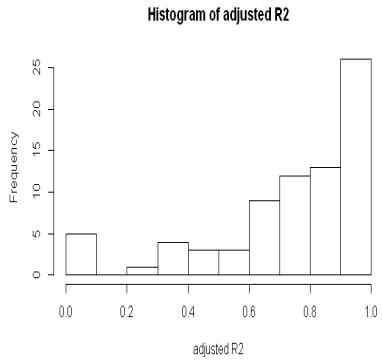


Figure 8.4: R^2_{adj} of the 76 sub-regressions.

`CorReg` found the above correlations but it also found more complex structures describing physical models, like Width = f (Mean flow , Mean speed) even if the true Physcial model is not linear : Width = flow / (speed * thickness) (here thickness is constant). Non-linear regulation models used to optimize the process were also found (but are confidential). These first results are easily understandable and meet metallurgists expertise. Sub-regressions with small values of R^2 are associated with non-linear model (chemical kinetics for example). The algorithm gives a structure of $p_r = 76$ subregressions with a mean of $\bar{\boldsymbol{p}}_f = 5.17$ regressors. In $\boldsymbol{X}^{I_f}$ the number of $\rho > 0.7$ is **79.33%** smaller than in $\boldsymbol{X}$.

It is now time to look at the predictive results (Table 8.1). We see that `CorReg` improves the results for each method tested in terms of prediction. We get parsimonious models automatically in a small amount of time (several hours but able to work during the night or the week-end)

Method	indicator	With `CorReg`	without `CorReg`
OLS	MSE	13.30	14.03
	(complexity)	(130)	(206)
LASSO	MSE	12.77	12.96
	(complexity)	(24)	(21)
Elasticnet	MSE	**12.15**	13.52
	(complexity)	(40)	(78)
Ridge	MSE	12.69	13.09
	(complexity)	(130)	(206)

Table 8.1: Results obtained on a validation sample (847 individuals).

In terms of interpretation, the main regression comes with the family of regression so it gives a better understanding of the consequences of corrective actions on the whole process. It typically permits to determine the *tuning parameters* whereas variable selection alone would point variables we can't directly act on. So it becomes easier to take corrective actions on the process to reach the goal. The stakes are so important that even a little improvement leads to consequent benefits, and we don't even talk about the impact on the market shares that is even more important.

8.2 Production case study

This second example is about a phenomenon that impacts the productivity of a steel plant. We have:

- a (confidential) response variable,
- $p = 145$ variables from the whole process to explain it but only $n = 100$ individuals.
- The stakes : 20% of productivity to gain on a specific product with high added value.

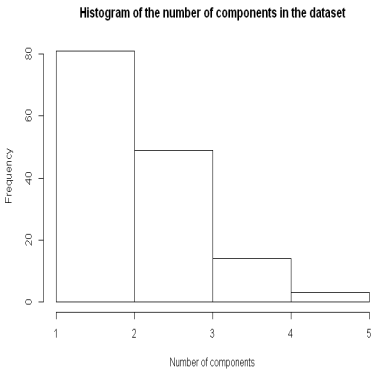


Figure 8.5: Distribution of the number of components found for each covariate.

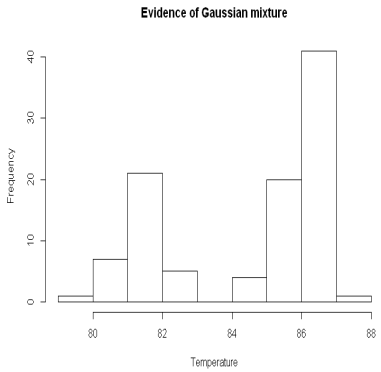


Figure 8.6: Another example of non-Gaussian real variable easily modeled by a Gaussian mixture.

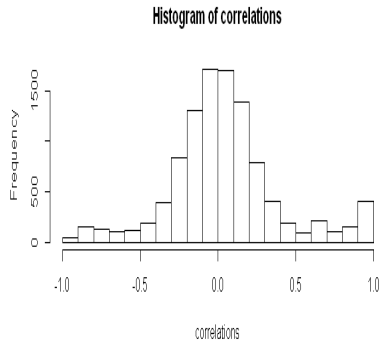


Figure 8.7: Histogram of correlations in $\boldsymbol{X}$.

`CorReg` found 55 sub-regressions as shown in Figure 8.8. One of them seems to be weak $R^2 = 0.17$ but is not linear (points out a link between diameter of a coil and some shape indicator).

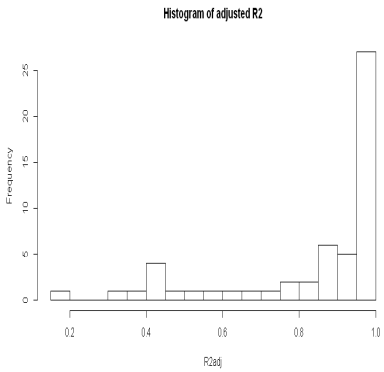


Figure 8.8: R^2_{adj} of the 55 sub-regressions.

The response variable was binary but n was too small compared to p to use logistic regression so we have considered $\boldsymbol{Y}$ as a continuous variable and then made imputation by 1 when $\hat{\boldsymbol{Y}} > 0.5$ and by 0 else.

In this precise case, `CorReg` found a structure that helped to decorrelate covariates in interpretation and to find the relevant part of the process to optimize. This product is made by a long process that requires several steel plants so it was necessary to point out the steel plant where the problem occurred.

Method	indicator	With `CorReg`	without `CorReg`
OLS	well-classified	100	56
	MSE (leave-one-out)	1.95	51 810
	complexity	91	100
LASSO	well-classified	93	93
	MSE (leave-one-out)	0.106	0.120
	complexity	27	34
Elasticnet	well-classified	84	87
	MSE (leave-one-out)	0.140	0.148
	complexity	10	13
Ridge	well-classified	88	85
	MSE (leave-one-out)	0.179	0.177
	complexity	91	146

Table 8.2: Results obtained with leave-one out cross-validation. $n = 100, p = 145$.

Part II

Further usage of the structure and perspectives

Chapter 9

Taking back the residuals

We have seen that eviction of redundant covariates improves the results by a good trade-off between dimension reduction and better conditioning versus keeping all the information. But The fact is that we lost some information and we want to get it back.

9.1 The model

After the estimation of the marginal model, we know both $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}^*$.

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{X}^{I_r}\boldsymbol{\beta}_{I_r} + \boldsymbol{X}^{I_f}\boldsymbol{\beta}_{I_f} + \boldsymbol{\varepsilon}_Y && (9.1)\\ \boldsymbol{X}^{I_r} &= \boldsymbol{X}^{I_f}\boldsymbol{\alpha} + \boldsymbol{\varepsilon} && (9.2)\\ \boldsymbol{Y} &= \boldsymbol{X}^{I_r}\underbrace{(\boldsymbol{\beta}_{I_r} + \boldsymbol{\alpha}\boldsymbol{\beta}_{I_f})}_{\beta^*} + \boldsymbol{\varepsilon}\boldsymbol{\beta}_{I_r} + \boldsymbol{\varepsilon}_Y && (9.3) \end{aligned}$$

Thus we have

$$\begin{aligned} \boldsymbol{Y} - \boldsymbol{X}^{I_r}\boldsymbol{\beta}^* &= \boldsymbol{\varepsilon}\boldsymbol{\beta}_{I_r} + \boldsymbol{\varepsilon}_Y && (9.4)\\ \boldsymbol{\varepsilon} &= \boldsymbol{X}^{I_r} - \boldsymbol{X}^{I_f}\boldsymbol{\alpha} && (9.5) \end{aligned}$$

So we introduce a plug-in model

$$\begin{aligned} \underbrace{\boldsymbol{Y} - \boldsymbol{X}^{I_r}\hat{\boldsymbol{\beta}}^*}_{\tilde{\boldsymbol{Y}}} &= \underbrace{(\boldsymbol{X}^{I_r} - \boldsymbol{X}^{I_f}\hat{\boldsymbol{\alpha}})}_{\tilde{\boldsymbol{X}}}\boldsymbol{\beta}_{I_r} + \boldsymbol{\varepsilon}_Y && (9.6)\\ & && (9.7) \end{aligned}$$

That allows us to estimate $\boldsymbol{\beta}_{I_r}$ with a classical linear model based on previous estimations of $\boldsymbol{\beta}^*$ and $\boldsymbol{\alpha}$. Then we have a model with a smaller noise

$$\boldsymbol{Y} = \boldsymbol{X}^{I_r}\hat{\boldsymbol{\beta}}^* + \hat{\boldsymbol{\varepsilon}}\hat{\boldsymbol{\beta}}_{I_r} + \boldsymbol{\varepsilon}_Y \tag{9.8}$$

and we can even find the original model by doing an identification step:

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{X}^{I_r}(\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\alpha}}\hat{\boldsymbol{\beta}}_{I_r}) + \boldsymbol{X}^{I_r}\hat{\boldsymbol{\beta}}_{I_r} + \boldsymbol{\varepsilon}_Y && (9.9)\\ &= \boldsymbol{X}^{I_f}\hat{\boldsymbol{\beta}}_{I_f} + \boldsymbol{X}^{I_r}\hat{\boldsymbol{\beta}}_{I_r} + \boldsymbol{\varepsilon}_Y && (9.10) \end{aligned}$$

Figure 9.1 shows the target of this plug-in model: the cases with enough correlations to have problem when using $\boldsymbol{X}$ but not enough correlations to have truly redundant covariates and to be able to delete some of them without significant information loss.

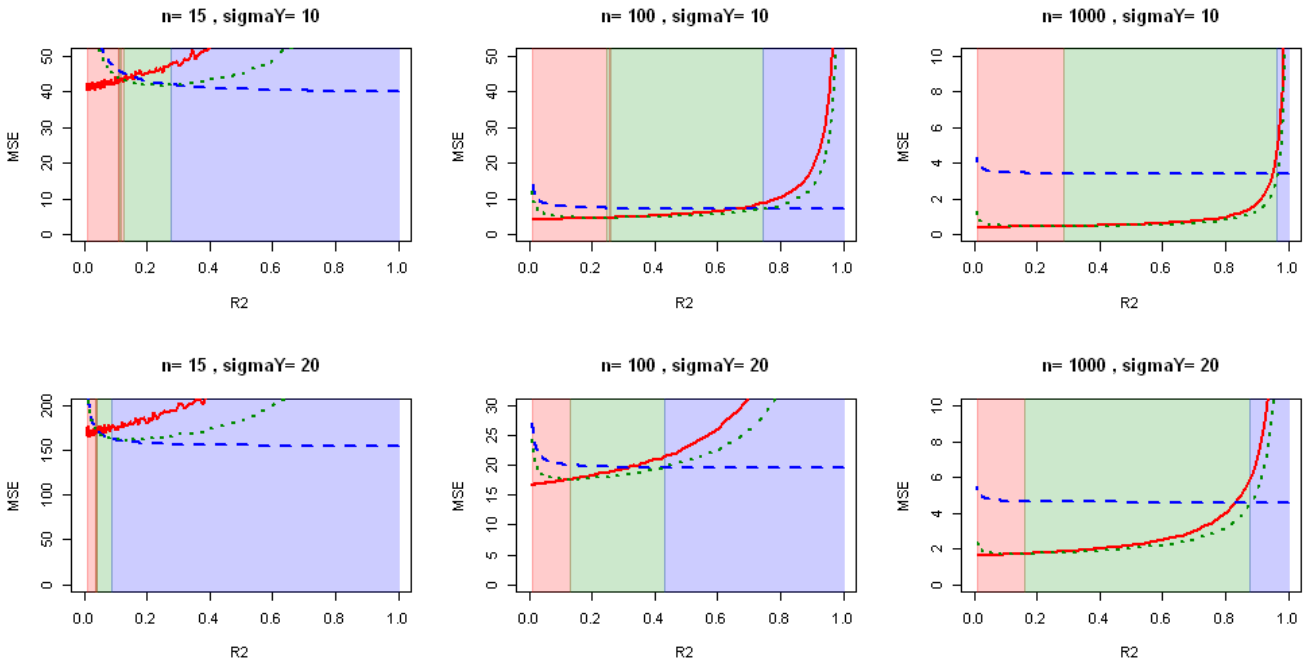


Figure 9.1: MSE of OLS (plain red) and `CorReg` marginal(blue dashed) and `CorReg` plug-in (green dotted) estimators for varying R^2 of the sub-regression, n and σ_Y.

9.2 Interpretation and latent variables

$\hat{\boldsymbol{\beta}}_{I_r}$ can be interpreted as the proper effect of $\boldsymbol{X}^{I_r}$ on $\boldsymbol{Y}$ in that it is the effect of the part of $\boldsymbol{X}^{I_r}$ that is independent from other covariates. Then if $\boldsymbol{X}^{I_r}$ is correlated to $\boldsymbol{Y}$ only through its correlation with $\boldsymbol{X}^{I_f}$ this sequential estimation will point it out and give a parsimonious model ($\hat{\boldsymbol{\beta}}_{I_r} = 0$) but the real stake is greater. We can see $\boldsymbol{\varepsilon}$ as a latent variable instead of the noise of a sub-regression. This latent variable is known to be independent of $\boldsymbol{X}^{I_f}$ and dependent of $\boldsymbol{X}^{I_r}$ so we can appreciate its meaning and we also know its value by $\hat{\boldsymbol{\varepsilon}} = \boldsymbol{X}^{I_r} - \boldsymbol{X}^{I_f}\hat{\boldsymbol{\alpha}}$. Thus, the plug-in model can reveal some kinds of latent variables.

9.3 Consistency

Consistency issues of the LASSO are well known and Zhao [Zhao and Yu, 2006] gives a very simple example to illustrate it. We have taken the same example to show how our method is more consistent. Here $p = 3$ and $n = 1000$.
We define $\boldsymbol{X}^{I_f}, \boldsymbol{X}^{I_r}, \boldsymbol{\varepsilon}_Y, \boldsymbol{\varepsilon}_X i.i.d. \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$ and then
$\boldsymbol{X}_3 = \frac{2}{3}\boldsymbol{X}_1 + \frac{2}{3}\boldsymbol{X}_2 + \frac{1}{3}\boldsymbol{\varepsilon}_X$ and
$\boldsymbol{Y} = 2\boldsymbol{X}_1 + 3\boldsymbol{X}_2 + \boldsymbol{\varepsilon}_Y$.
We compare consistencies of complete, marginal and full plug-in model with LASSO (and LAR) for selection. It happens that our MCMC algorithm don't find the true structure but a permuted one so we also look at the results obtained with the true S (but $\hat{\boldsymbol{\alpha}}$ is used) and with the structure found by the Markov chain after a few seconds.

True S was found 340 times on 1000 tries (model is not identifiable because $\boldsymbol{X}^j$ are all Gaussian).

We observe as we hoped that our marginal model is better when using true S (coercing real zeros) and that marginal with $\hat{S}$ is penalized (coercing wrong coefficients to be zeros when true S is not found). But the main point is that the plug-in model stays better than the classical

	Classical LASSO	`CorReg` + marginal LASSO	`CorReg` + full plug-in LASSO
True S	1.006479	**1.005468**	**1.006093**
$\hat{Z}$	**1.006479**	1.884175	1.006517

Table 9.1: MSE observed on a validation sample (1000 individuals)

one with the true S and corrects enough the marginal model to follow the classical LASSO closely when using $\hat{S}$. And when we look at the consistency :

	Classical LASSO	`CorReg` + marginal LASSO	`CorReg` + full plug-in LASSO
True S	0	1000	830
$\hat{S}$	0	340	**621**

Table 9.2: Number of consistent model found ($\boldsymbol{Y}$ depending on $\boldsymbol{X}_1, \boldsymbol{X}_2$ and only them) on 1000 tries

We also made the same experiment but with $\boldsymbol{X}_1, \boldsymbol{X}_2$ (and consequently $\boldsymbol{X}_3$) following Gaussian mixtures (to improve identifiability) randomly generated by our `CorReg` package for R. True S is now found 714 times on 1000 tries . So it confirms that Gaussian mixture models are easier to identify.

	Classical LASSO	`CorReg` + marginal LASSO	`CorReg` + full plug-in LASSO
True S	1.571029	**1.569559**	**1.570801**
$\hat{S}$	1.005402	1.465768	**1.005066**

Table 9.3: MSE observed on a validation sample (1000 individuals)

And when we look at the consistency :

	Classical LASSO	`CorReg` + marginal LASSO	`CorReg` + full plug-in LASSO
True S	0	1000	789
$\hat{S}$	0	714	**608**

Table 9.4: Number of consistent model found (Y depending on X_1, X_2 and only them) on 1000 tries

9.4 Numerical results

We test the plug-in model with datasets generated the same way as for section 7.3.

9.4.1 $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$

We try the method with a response depending on all covariates (`CorReg` reduces the dimension and can't give the true model if there is a structure). We observe for OLS (Figures 9.2 to 9.4) that the plug-in model gives results similar in efficiency to the marginal model, but remains better than the complete model for smaller correlations even for $n = 400$. We also observe that we can found a model with more than n coefficients when each estimation step computes less than n coefficients. It means that we estimate more coefficients than the classical OLS and keep a smaller variance so the plug-in model can also be an alternative to the complete model. It is interesting to see that OLS combined with the plug-in model is a sort of sequential estimation that allows to estimate more than n coefficients.

Ordinary Least Squares when Y depends on all variables in X

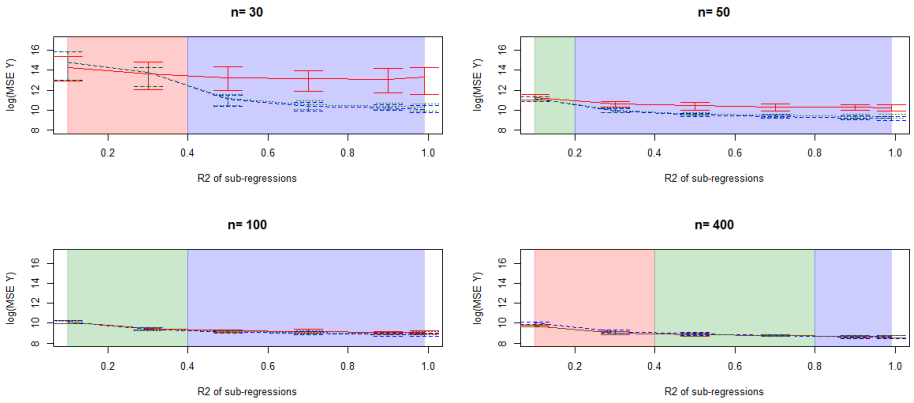


Figure 9.2: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

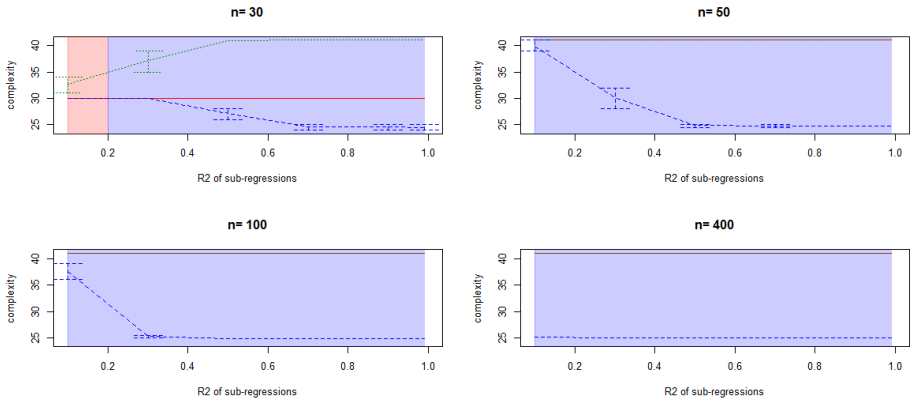


Figure 9.3: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

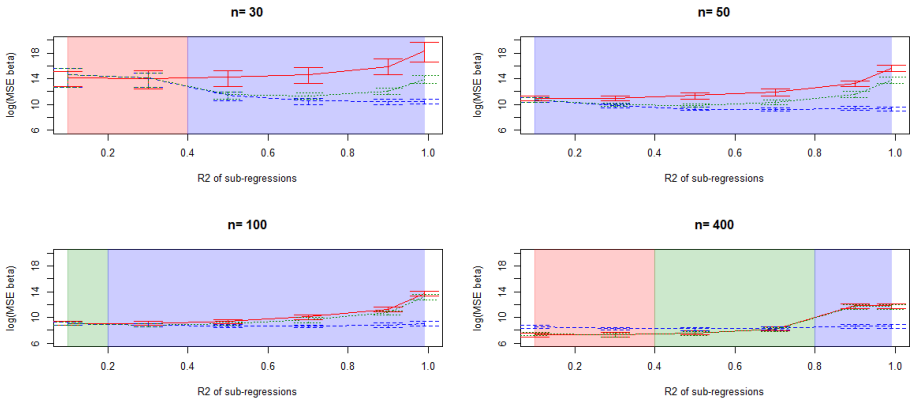


Figure 9.4: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

LASSO when Y depends on all variables in X

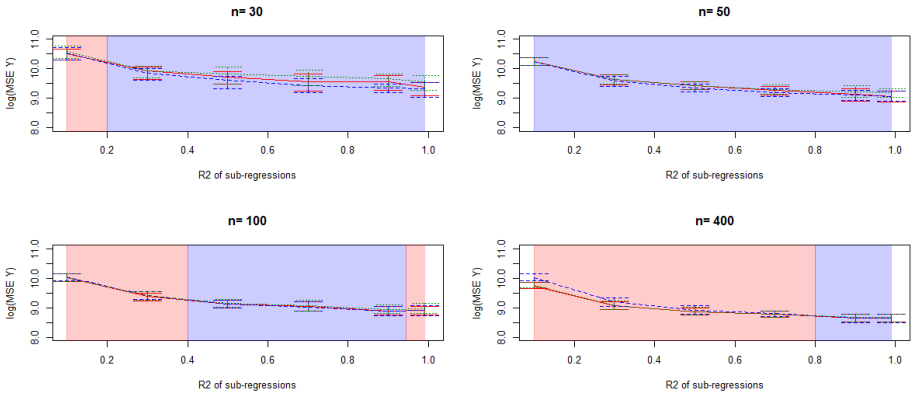


Figure 9.5: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

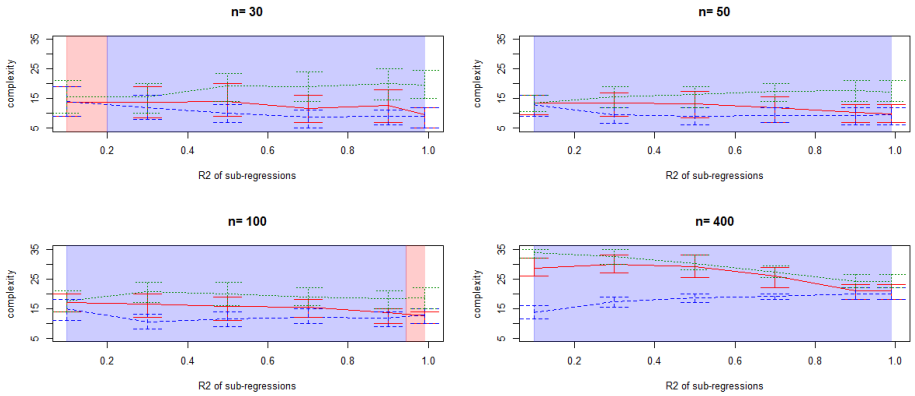


Figure 9.6: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

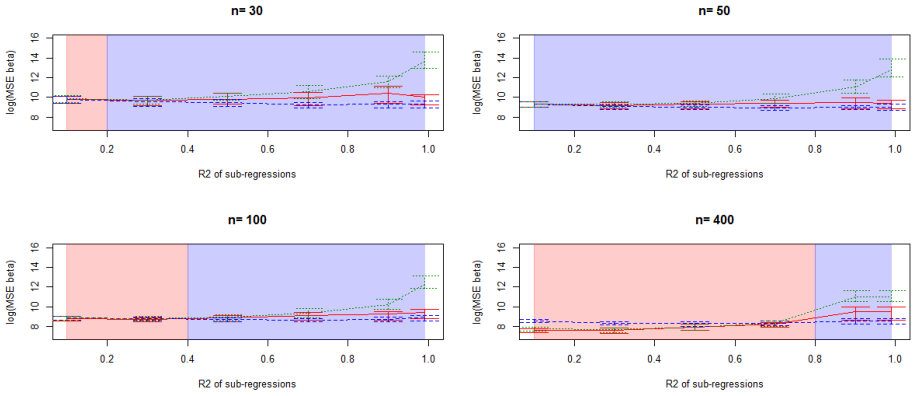


Figure 9.7: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Elasticnet when Y depends on all variables in X

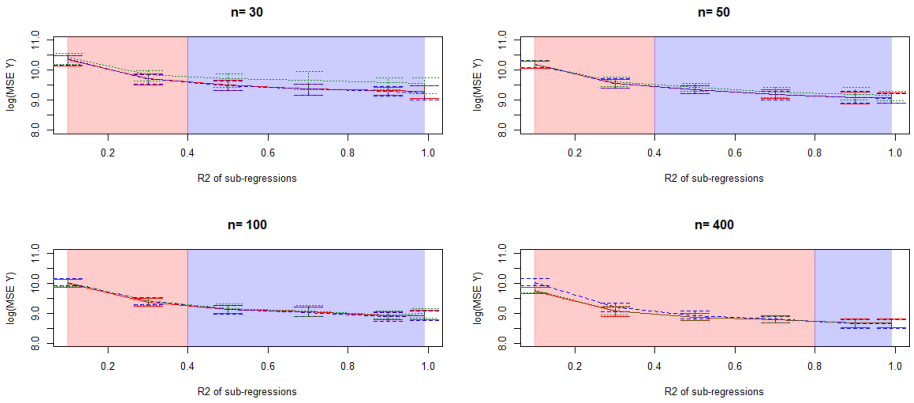


Figure 9.8: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

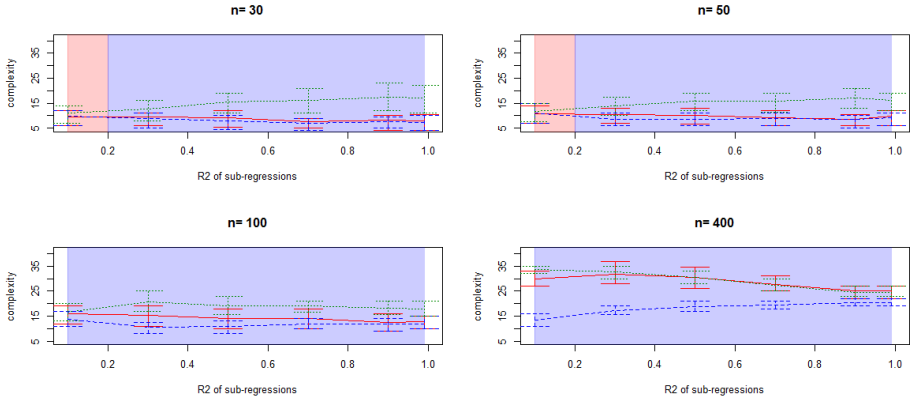


Figure 9.9: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

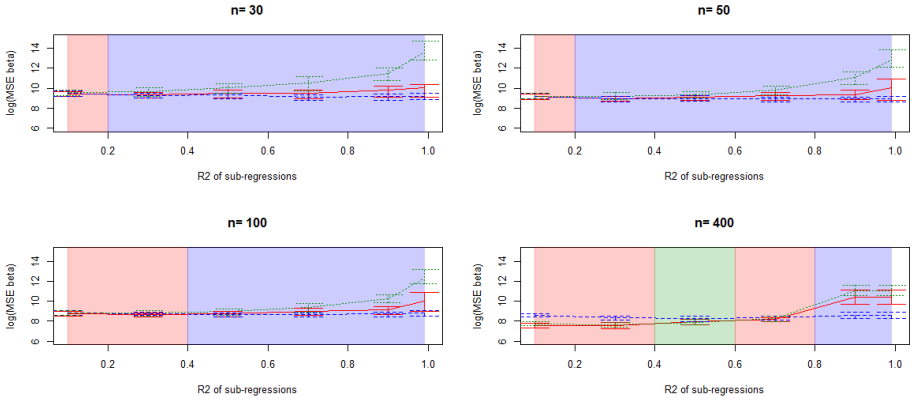


Figure 9.10: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Stepwise when Y depends on all variables in X

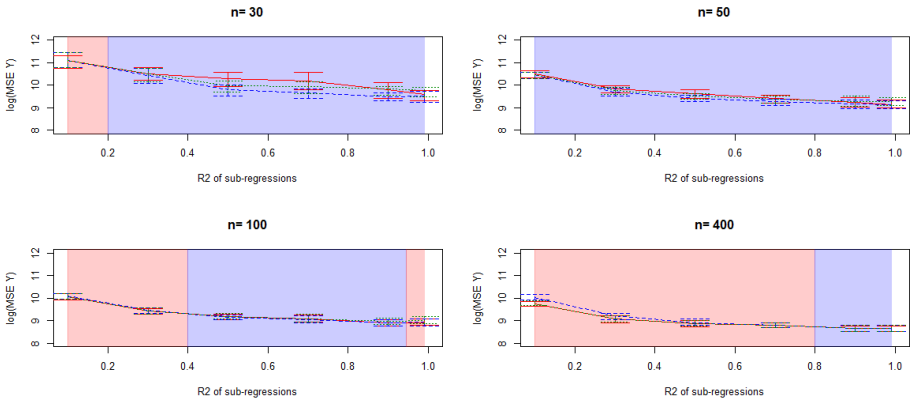


Figure 9.11: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

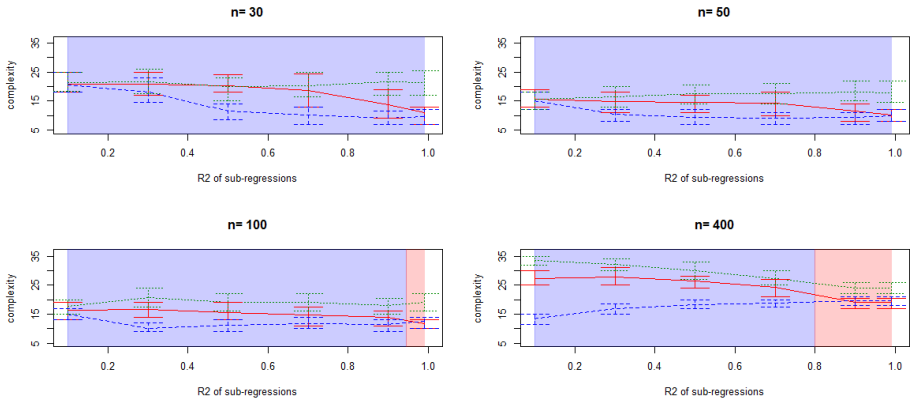


Figure 9.12: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

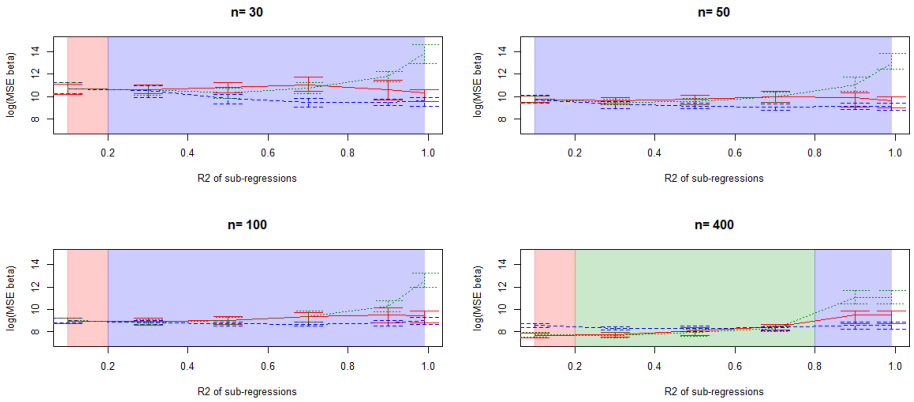


Figure 9.13: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Ridge regression when Y depends on all variables in X

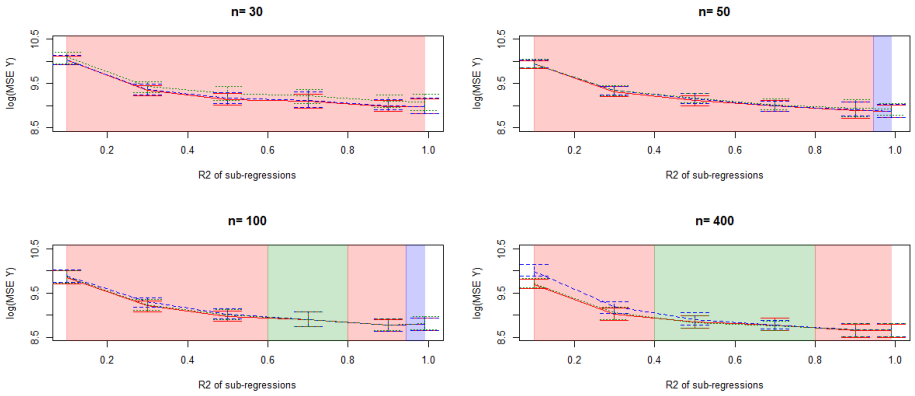


Figure 9.14: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

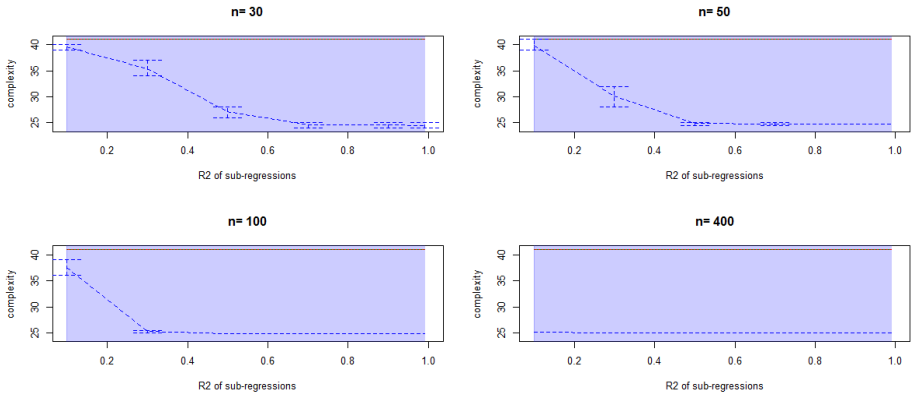


Figure 9.15: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

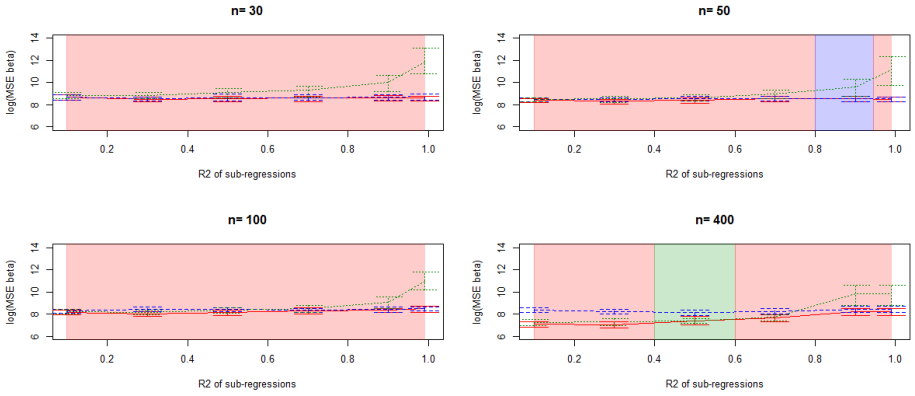


Figure 9.16: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Combined with variable selection methods (Figures 9.5 to 9.13) it does converge to the complete model results for large values of n so it improves the marginal model for weak correlations (it is what it was built for) has no significant interest compared to the complete model. Ridge regression (Figures 9.14 to 9.16) leads to the same conclusion.

9.4.2 Y depends only on covariates in X^{I_f}

This case shows the robustness of selection methods, that rejects covariates in the plug-in model.

9.4.3 Y depends only on covariates in X^{I_r}

We now try the method with a response depending only on variables in $\boldsymbol{X}^{I_r}$. The datasets used here were still the same. Depending only on $\boldsymbol{X}_r$ implies sparsity and impossibility for the marginal model to obtain the true model when using the true structure, so we hope to see an improvement with the plug-in method. This is the reason why we have developed the plug-in model.

For OLS (figures 9.32 to 9.34) we note that the plug-in model improves results of the marginal model for large values of n. This is still the case with variable selection methods even if it is not sufficient to improve the complete model.
This phenomenon is still observed with the ridge regression with more efficiency.

9.4.4 About the plug-in model

The plug-in model sounds good when described theoretically and figure 9.1 makes us hope to obtain good results with it. But the reality is that the plug-in model (by definition) relies on a first estimation (the marginal model) thus it does reach the true model but asymptotically, and with a slow convergence speed. Moreover, if $\hat{S}$ is not exactly the true S but the partition is good, the marginal model is not impacted whereas the plug-in model uses both $\hat{\boldsymbol{\beta}}^*_{I_f}$ and $\hat{\boldsymbol{\alpha}}$ so it is does not rely only on one estimation. $\hat{\boldsymbol{\alpha}}$ is another source of bias for finite values of n and it depends itself on $\hat{S}$. Ordinary Least Squares really are in great trouble when confronted to correlated datasets so the plug-in model improves OLS any way but other methods are a bit less sensitive to correlations so it is difficult to improve them with a plug-in model relying on so many estimators.

Ordinary Least Squares when Y depends only on covariates in X^{I_f}

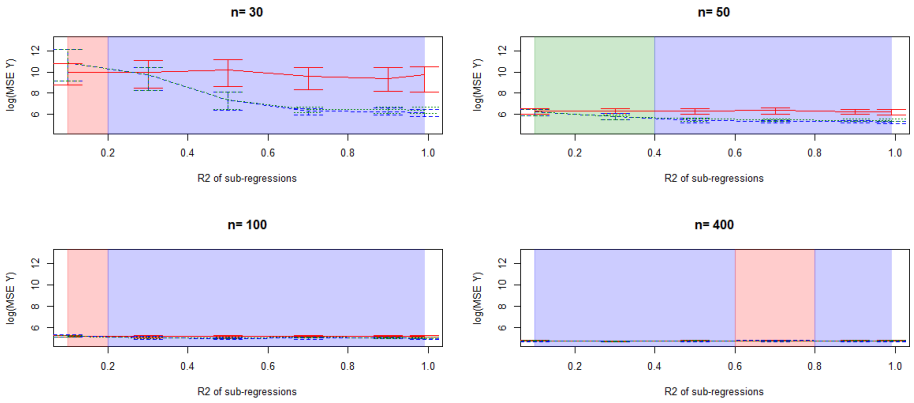


Figure 9.17: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

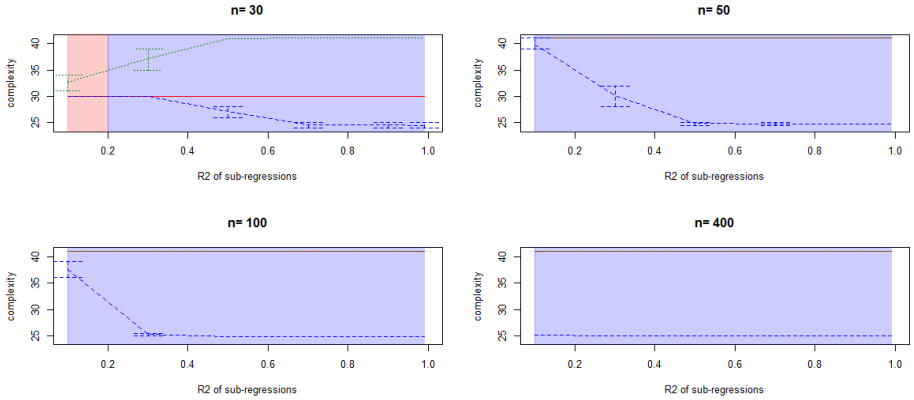


Figure 9.18: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

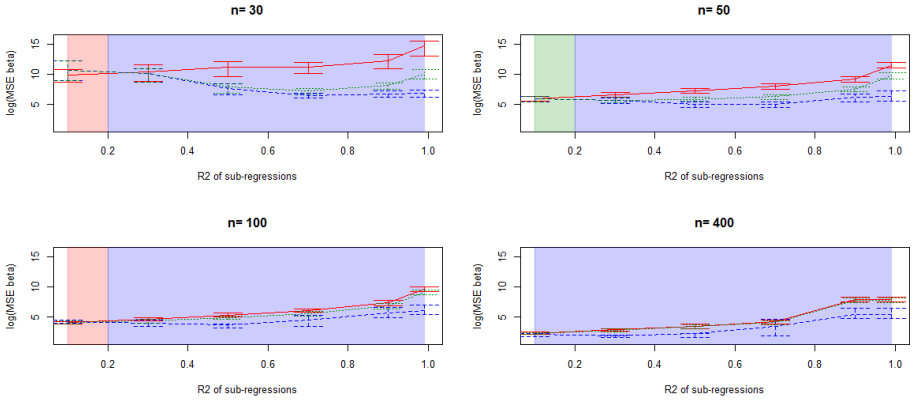


Figure 9.19: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

LASSO when Y depends only on covariates in X^{I_f}

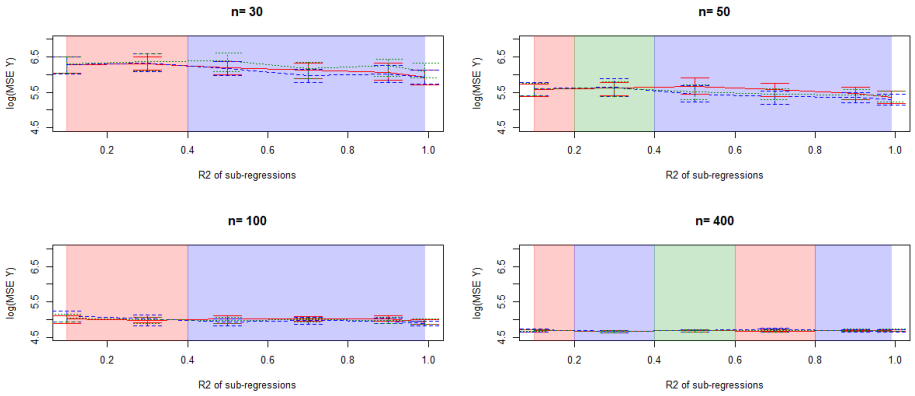


Figure 9.20: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

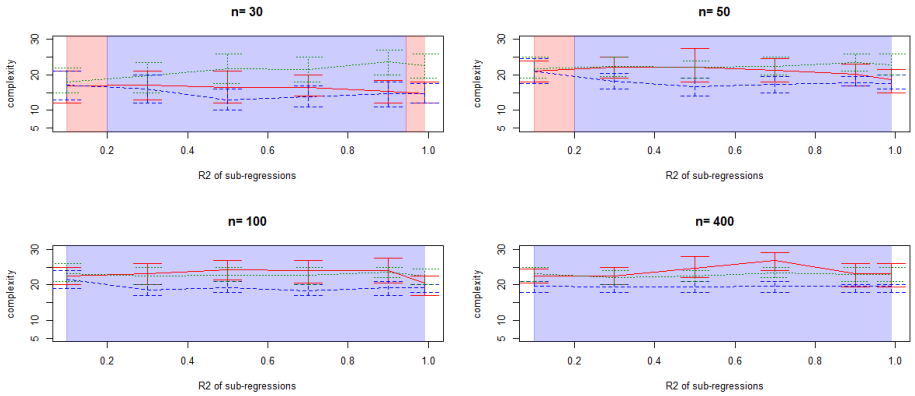


Figure 9.21: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

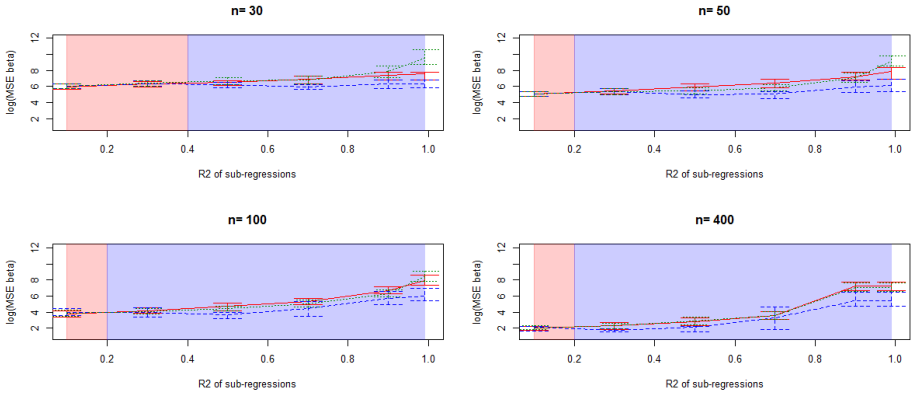


Figure 9.22: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Elasticnet when Y depends only on covariates in X^{I_f}

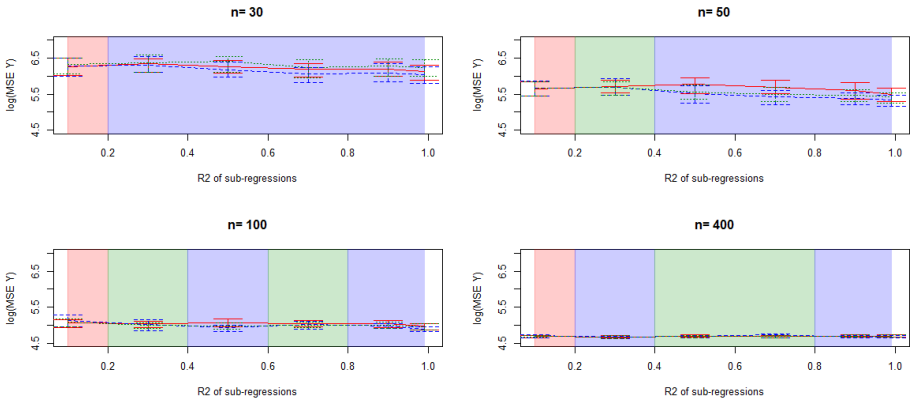


Figure 9.23: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

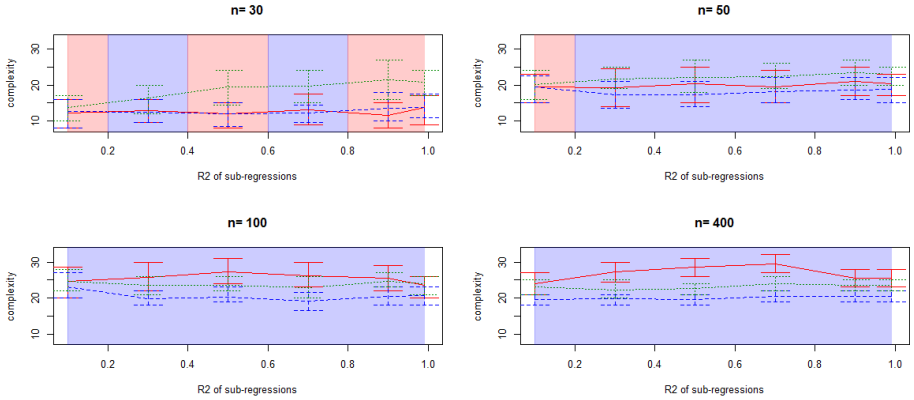


Figure 9.24: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

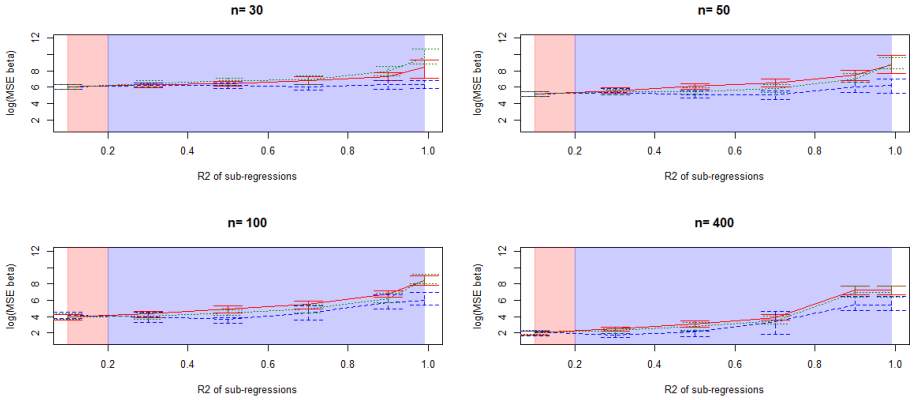


Figure 9.25: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Stepwise when Y depends only on covariates in X^{I_f}

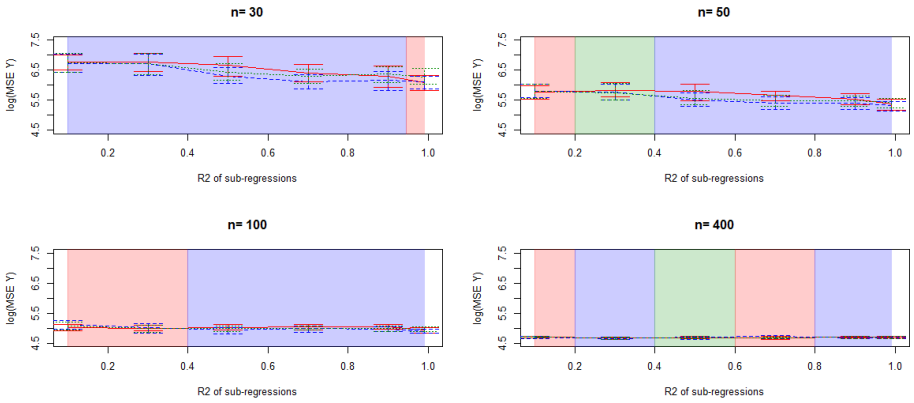


Figure 9.26: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

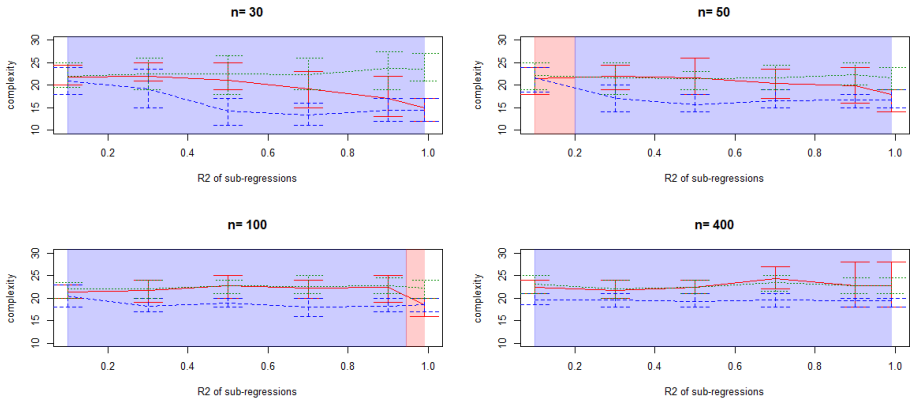


Figure 9.27: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

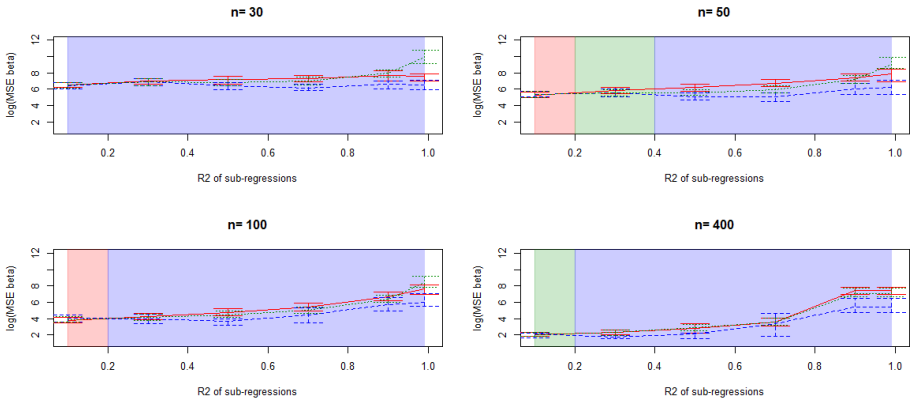


Figure 9.28: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Ridge regression when Y depends only on covariates in X^{I_f}

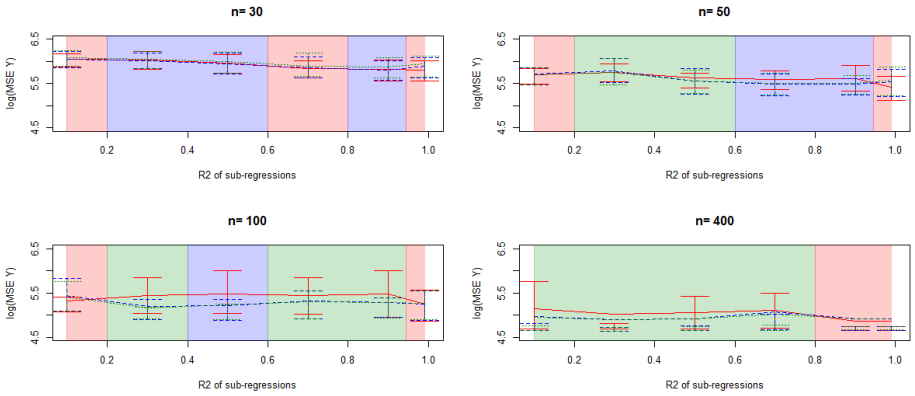


Figure 9.29: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

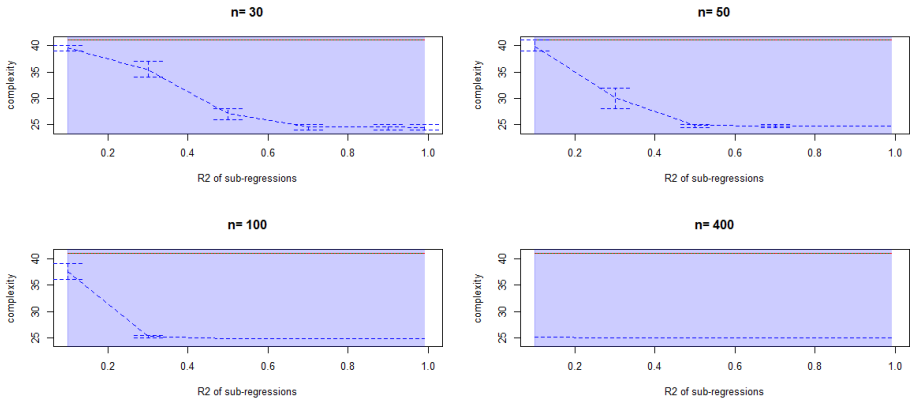


Figure 9.30: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

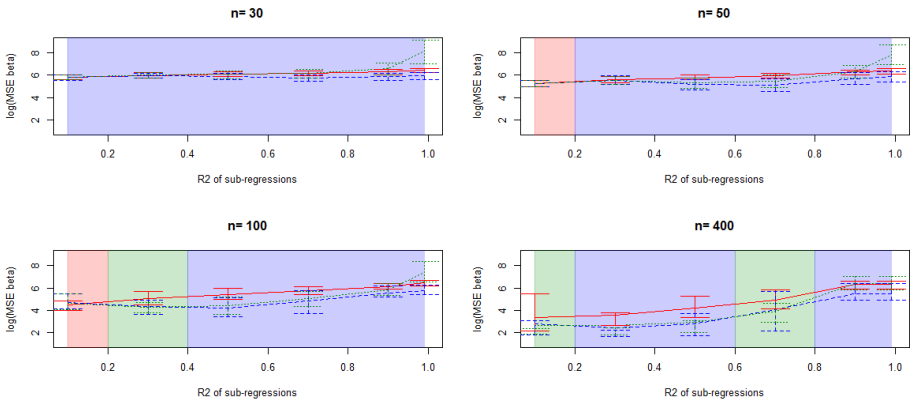


Figure 9.31: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Ordinary Least Squares when Y depends only on covariates in X^{I_r}

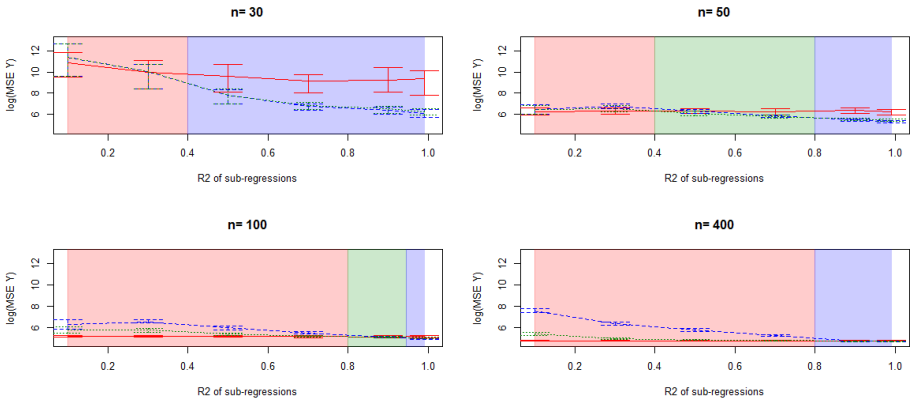


Figure 9.32: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

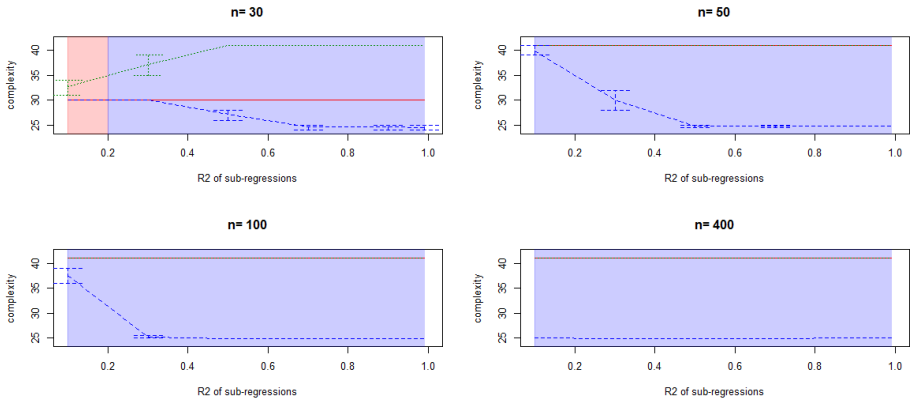


Figure 9.33: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

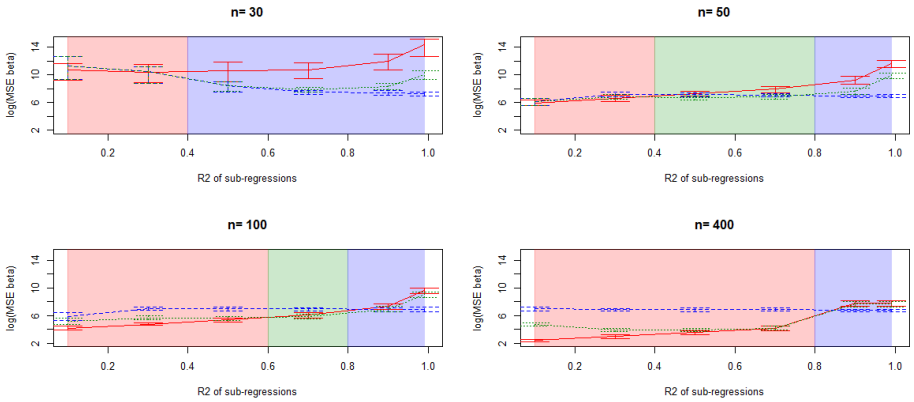


Figure 9.34: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

LASSO when Y depends only on covariates in X^{I_r}

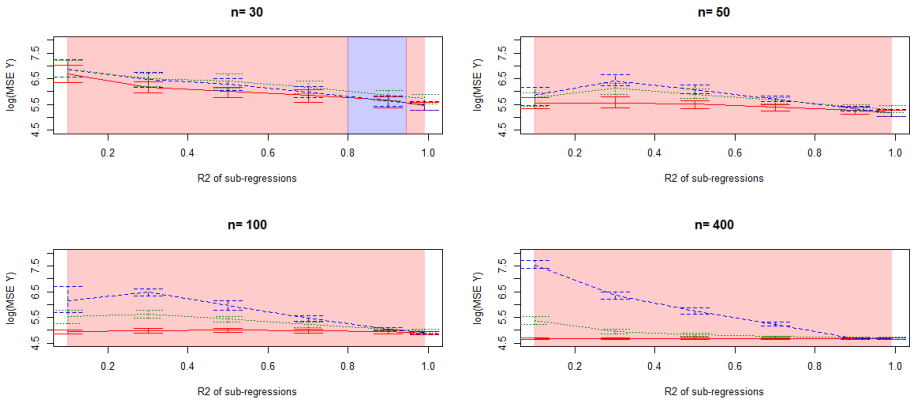


Figure 9.35: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

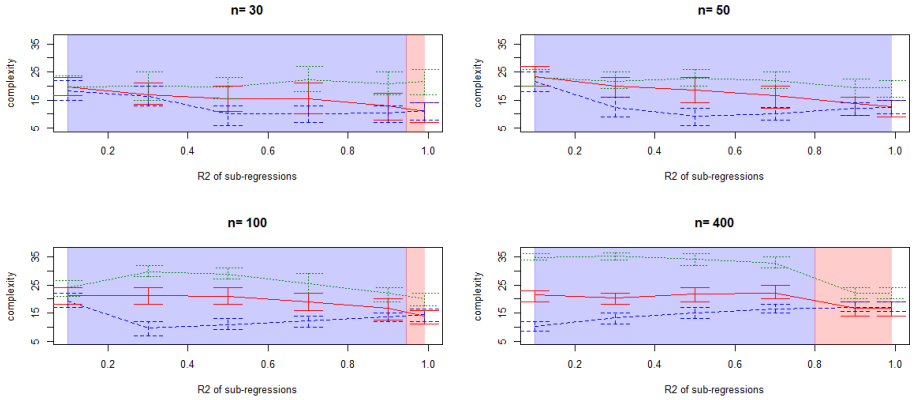


Figure 9.36: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

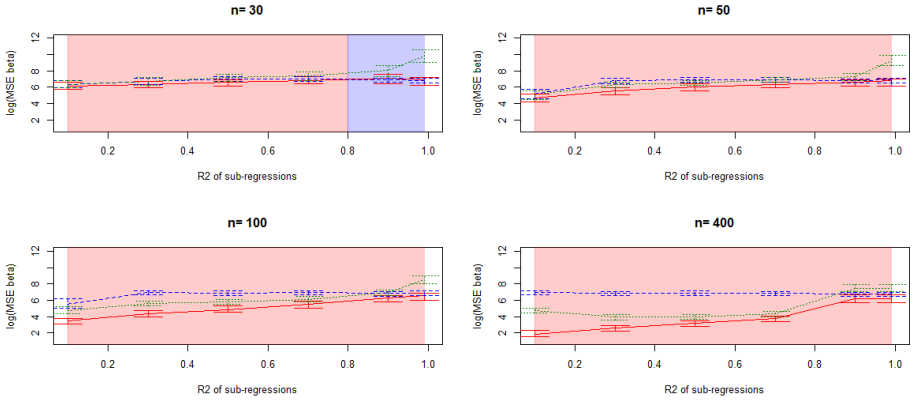


Figure 9.37: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Elasticnet when Y depends only on covariates in X^{I_r}

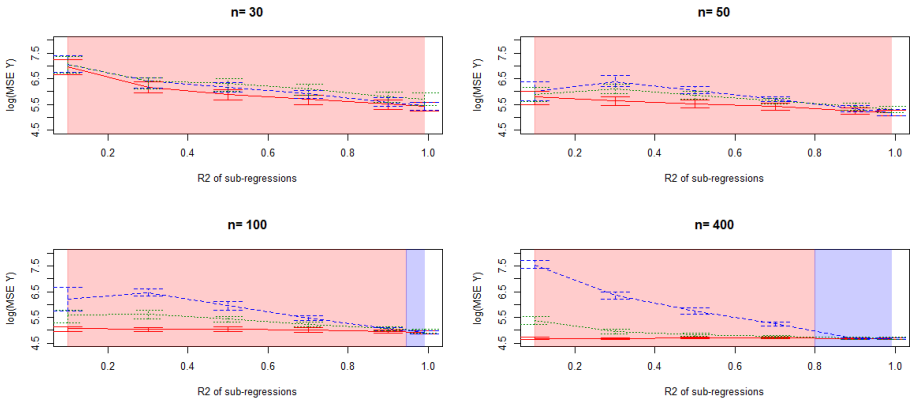


Figure 9.38: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

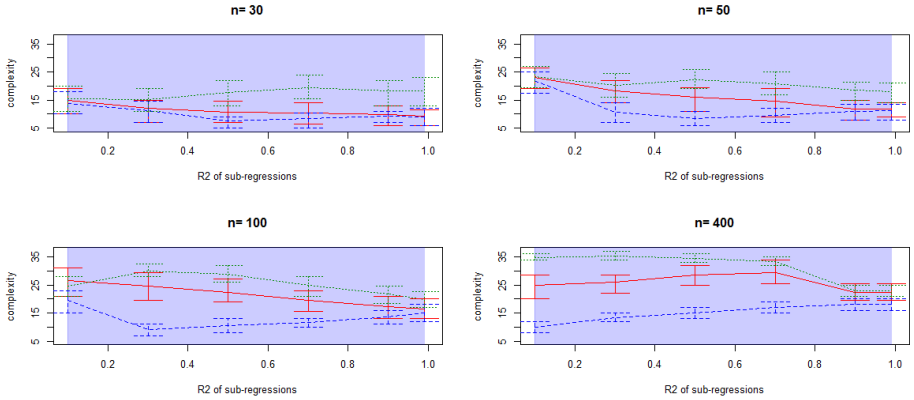


Figure 9.39: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

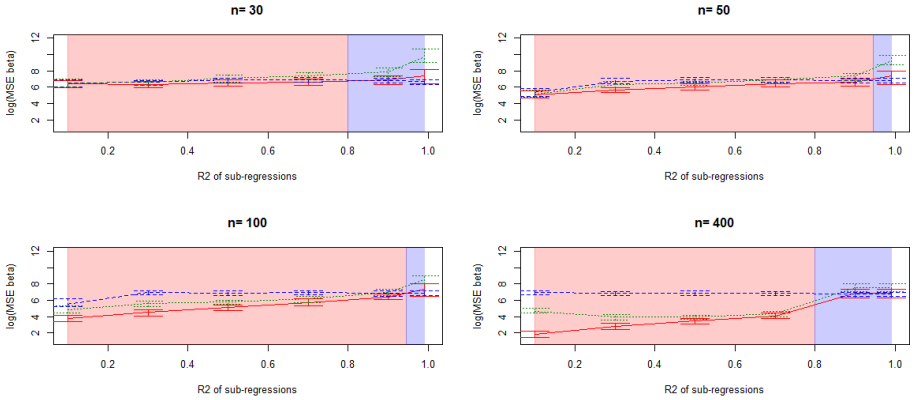


Figure 9.40: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Stepwise when Y depends only on covariates in X^{I_r}

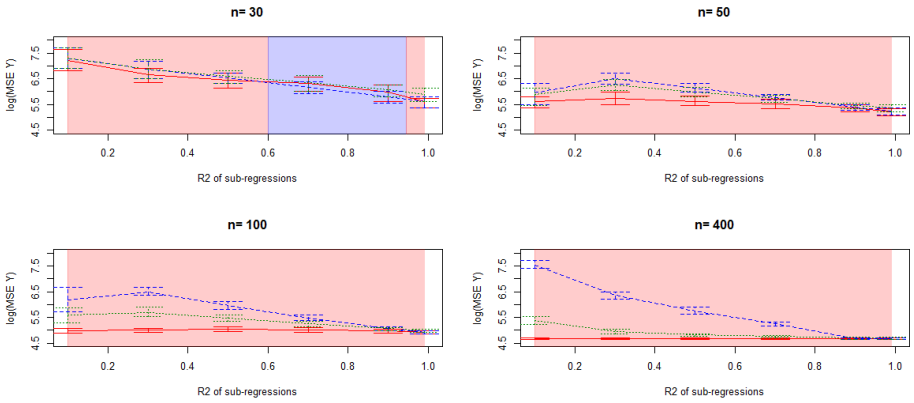


Figure 9.41: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

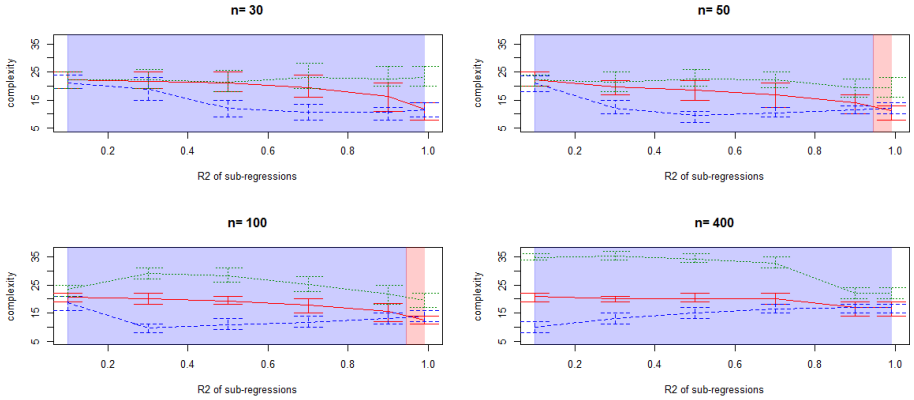


Figure 9.42: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

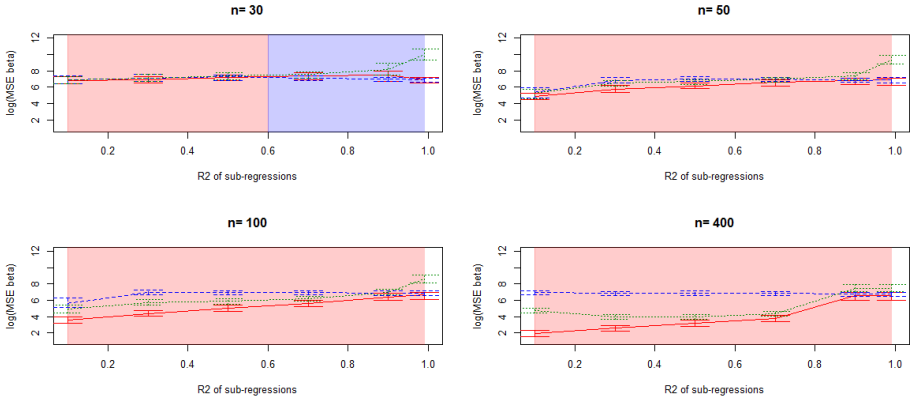


Figure 9.43: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Ridge regression when Y depends only on covariates in X^{I_r}

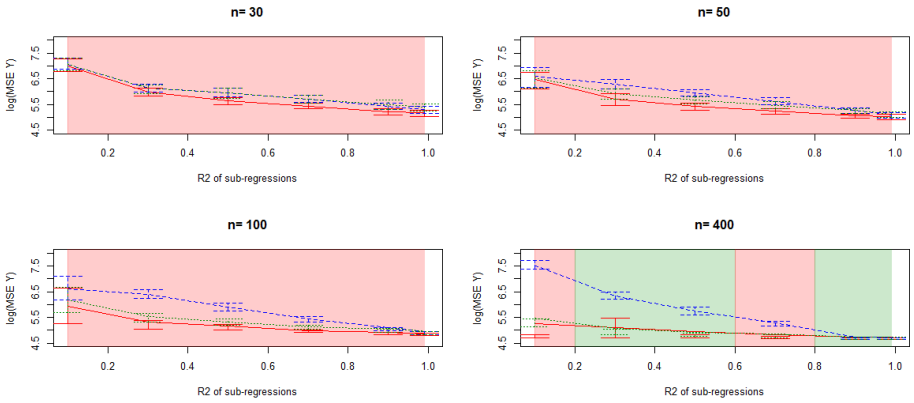


Figure 9.44: Comparison of the MSE on $\hat{\boldsymbol{Y}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

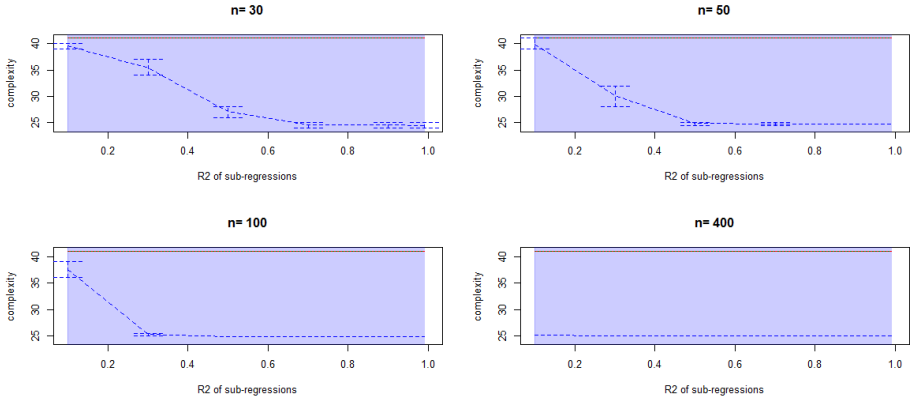


Figure 9.45: Comparison of the complexities, red=classical (complete) model, blue=marginal model, green=plug-in model

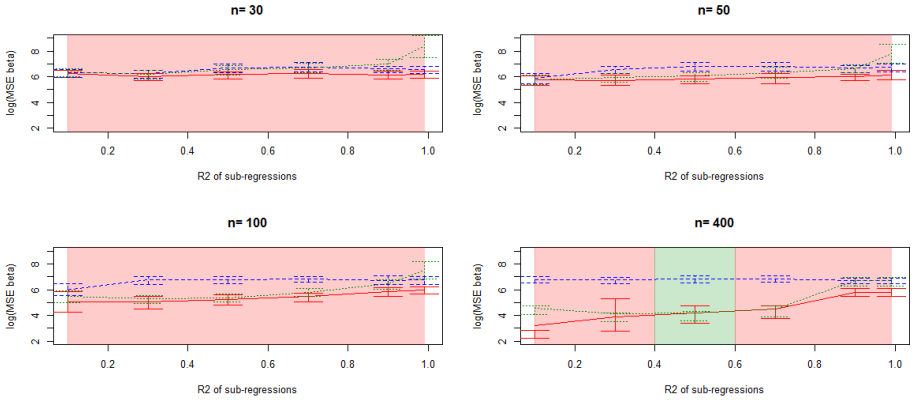


Figure 9.46: Comparison of the MSE on $\hat{\boldsymbol{\beta}}$, red=classical (complete) model, blue=marginal model, green=plug-in model

Chapter 10

Missing values

Résumé: Le modèle génératif complet sur les données nous permet d'obtenir la loi des valeurs manquantes. Mais nous pouvons aller encore plus loin car la modélisation explicite des corrélations nous permet d'obtenir les lois conditionnelles de chaque valeur manquante sachant les valeurs observées. Ce chapitre présente comment nous pouvons par un simple algorithme SEM estimer les paramètres des sous-régressions dans la chaîne MCMC de recherche de structure. Enfin, nous pouvons imputer les valeurs manquantes à l'aide d'un Gibbs qui procède par imputations multiples, fournissant au passage un indicateur de précision sur l'imputation proposée.

10.1 Introduction

Real datasets often have missing values and it is a very recurrent issue in industry. We note $\boldsymbol{M}$ the $n \times p$ binary matrix indicating whereas a value is missing (1) or not (0) in $\boldsymbol{X}$. We note $\boldsymbol{X}_M$ the missing values and $\boldsymbol{X}_O$ the observed values. $\Theta = \{\boldsymbol{\mu}_X, \boldsymbol{\Sigma}_X\}$ stands for the parameters of the Gaussian mixture followed by $\boldsymbol{X}$. $\boldsymbol{\alpha}$ is the matrix of the sub-regression coefficients with $\alpha_{i,j}$ the coefficients associated to $\boldsymbol{X}^i$ in the sub-regression explaining $\boldsymbol{X}^j$.
Here we suppose that missing values are Missing Completely At Random (MCAR). Many methods does exist to manage such problems [Little, 1992] but they make approximation , add noise (imputation methods) or delete information (cutting methods).

We have a full generative model on $\boldsymbol{X}$ with explicit dependencies within the covariates. So when a value is missing, we know its distribution but more than that, we know its conditional distribution based on observed values for the same individual. Thus we are able to make imputation and to describe the missing values with their conditional distribution. This is a positive side-effect of the explicit generative model on $\boldsymbol{X}$.

10.2 Estimation of the sub-regressions with missing values

10.2.1 The integrated likelihood

The first thing we do with $\boldsymbol{X}$ is to estimate S. During the MCMC, for each candidate we have to compute the likelihood of the candidate, depending on $\boldsymbol{\alpha}$ the matrix of the sub-regression

coefficients. We start with the complete likelihood of $\boldsymbol{X}$

$$
\begin{aligned}
L(\boldsymbol{\alpha},\Theta,S;\boldsymbol{X}) &= \prod_{i=1}^{n} f(\boldsymbol{X}_i) = \prod_{i=1}^{n}\left[f(\boldsymbol{X}_i^{I_r}|\boldsymbol{X}_i^{I_f};\boldsymbol{\alpha},\Theta,S)f(\boldsymbol{X}_i^{I_f};\boldsymbol{\alpha},\Theta,S)\right] && (10.1)\\
&= \prod_{i=1}^{n}\left[\prod_{j\in I_r} f(x_{i,j}|\boldsymbol{X}_i^{I_f};\boldsymbol{\alpha},\Theta,S)\prod_{j\notin I_r} f(x_{i,j};\boldsymbol{\alpha},\Theta,S)\right] && (10.2)\\
&= \prod_{i=1}^{n}\left[\prod_{j\in I_r} f(x_{i,j}|\boldsymbol{X}_i^{I_f^j};\boldsymbol{\alpha},\Theta,S)\prod_{j\notin I_r} f(x_{i,j};\boldsymbol{\alpha},\Theta,S)\right] && (10.3)\\
\mathcal{L}(\boldsymbol{\alpha},\Theta,S;\boldsymbol{X}) &= \sum_{i=1}^{n}\left[\sum_{j\in I_r}\log\left(f(x_{i,j}|\boldsymbol{X}_i^{I_f^j};\boldsymbol{\alpha},\Theta,S)\right)+\sum_{j\notin I_r}\log\left(f(x_{i,j};\boldsymbol{\alpha},\Theta,S)\right)\right] && (10.4)
\end{aligned}
$$

In the MCMC we need to compute the likelihood of the dataset knowing the structure. When missing values occurs, we restrict the likelihood to the known values by integration on $\boldsymbol{X}_M$.

We know that $\boldsymbol{X}$ is a Gaussian mixture (*iid* individuals, vectors of orthogonal Gaussian mixtures $\boldsymbol{X}^{I_f}$ and linear combinations of these Gaussian mixtures and some Gaussian for $\boldsymbol{X}^{I_r}$) with K the number of its components.

$$
\begin{aligned}
L(\boldsymbol{\alpha},\Theta,S;\boldsymbol{X}_0) &= \int_{\boldsymbol{X}_M} L(\boldsymbol{\alpha},\Theta,S;\boldsymbol{X})d\boldsymbol{X} = \int_{\boldsymbol{X}_M}\sum_{k=1}^{K}\pi_k\phi_k(\boldsymbol{X};\boldsymbol{\alpha},\Theta,S)d\boldsymbol{X} && (10.5)\\
&= \sum_{k=1}^{K}\pi_k\int_{\boldsymbol{X}_M}\phi_k(\boldsymbol{X};\boldsymbol{\alpha},\Theta,S)d\boldsymbol{X} = \sum_{k=1}^{K}\pi_k\int_{\boldsymbol{X}_M}\prod_{i=1}^{n}\phi_k(\boldsymbol{X}_i;\boldsymbol{\alpha},\Theta,S)d\boldsymbol{X} && (10.6)\\
&= \sum_{k=1}^{K}\pi_k\prod_{i=1}^{n}\int_{\boldsymbol{X}_{i,M}}\phi_k(\boldsymbol{X}_i;\boldsymbol{\alpha},\Theta,S)d\boldsymbol{X}_i = \sum_{k=1}^{K}\pi_k\prod_{i=1}^{n}\phi_k(\boldsymbol{X}_{i,O};\boldsymbol{\alpha},\Theta,S) && (10.7)\\
&= \sum_{k=1}^{K}\pi_k\phi_k(\boldsymbol{X}_O;\boldsymbol{\alpha},\Theta,S) = f(\boldsymbol{X}_O,\boldsymbol{\alpha},\Theta,S) && (10.8)
\end{aligned}
$$

To compute this likelihood, we will use the decomposition

$$
\begin{aligned}
L(\boldsymbol{\alpha},\Theta,S;\boldsymbol{X}_0) &= f(\boldsymbol{X}_O;\boldsymbol{\alpha},\Theta,S) = \prod_{i=1}^{n} f(\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S)f(\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S) && (10.9)\\
&= \prod_{i=1}^{n} f(\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S)\prod_{\substack{j\in I_f\\ M_{i,j}=0}} f(x_{i,j};\boldsymbol{\alpha},\Theta,S) && (10.10)
\end{aligned}
$$

with $\forall(i,j)$ with $\boldsymbol{M}_{i,j}=0$ and $j\notin I_r$:

$$
f(x_{i,j};\boldsymbol{\alpha},\Theta,S) = \sum_{k=1}^{K_j}\pi_{j,k}\Phi_k(x_{i,j};\mu_{j,k},\Sigma_{j,k}) \qquad (10.11)
$$

with $K_j,\pi_{j,k},\mu_{j,k},\Sigma_{j,k}$ and the likelihood estimated once (for example by RMixmod [Auder et al., 2014]) before the MCMC starts.

And, $\forall(i,j)$ with $\boldsymbol{M}_{i,j}=0$ and $j\in I_r$:

$$
\begin{aligned}
f(x_{i,j}|\boldsymbol{X}_{i,O}^{I_f^j};\boldsymbol{\alpha},\Theta,S) &= \sum_{k=1}^{K_{ij}}\pi_{ij,k}\Phi(x_{i,j};\mu_{ij,k},\Sigma_{ij,k}) \text{ where} && (10.12)\\
\boldsymbol{\pi}_{ij} &= \bigotimes_{\substack{l\in I_f^j\\ \boldsymbol{M}_{i,l}=1}}\boldsymbol{\pi}_l \text{ and } K_{ij}=|\boldsymbol{\pi}_{ij}|, && (10.13)\\
\boldsymbol{\mu}_{ij} &= \sum_{\substack{l\in I_f^j\\ \boldsymbol{M}_{i,l}=0}}\alpha_{l,j}x_{i,l}+\bigoplus_{\substack{l\in I_f^j\\ \boldsymbol{M}_{i,l}=1}}\alpha_{l,j}\boldsymbol{\mu}_l && (10.14)\\
\boldsymbol{\Sigma}_{ij} &= \sigma_j^2+\bigoplus_{\substack{l\in I_f^j\\ \boldsymbol{M}_{i,l}=1}}\alpha_{i,l}^2\boldsymbol{\Sigma}_l && (10.15)
\end{aligned}
$$

This could be easily used for imputation of the missing values in $\boldsymbol{X}^{I_r}$ knowing the parameters $\boldsymbol{\alpha},\Theta$ and S. We note that we obtain a Gaussian when there is no missing value in I_f^j. But we see that $f(\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S)$ is not the product of the $f(x_{i,j}|\boldsymbol{X}_{i,O}^{I_f^j};\boldsymbol{\alpha},\Theta,S)$ if a same missing value occurs in distinct sub-regressions. Thus if every sub-regression are distinct connex component then we can use (10.12) and we have

$$
L(\boldsymbol{\alpha},\Theta,S;\boldsymbol{X}_0)=\prod_{i=1}^{n}\prod_{\substack{j\in I_r\\ M_{i,j}=0}}f(x_{i,j}|\boldsymbol{X}_{i,O}^{I_f^j};\boldsymbol{\alpha},\Theta,S)\prod_{\substack{j\in I_f\\ M_{i,j}=0}}f(x_{i,j};\boldsymbol{\alpha},\Theta,S) \tag{10.16}
$$

But for the general case we need to manage the dependencies implied by missing values in common covariates in the I_f^j. We note $f(\boldsymbol{X})=\sum_{k=1}^{K}\pi_k\mathcal{N}(\boldsymbol{\mu}_{X,k};\boldsymbol{\Sigma}_{X,k})$.

$$
\begin{aligned}
L(\boldsymbol{\alpha},\Theta,S;\boldsymbol{X}_0) &= \prod_{i=1}^{n}\sum_{k=1}^{K}\pi_k\Phi_k(\boldsymbol{X}_{i,O};\boldsymbol{\alpha},\Theta,S) && (10.17)\\
&= \prod_{i=1}^{n}\sum_{k=1}^{K}\pi_k\Phi_k(\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S)\Phi_k(\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S) && (10.18)\\
&= \prod_{i=1}^{n}\sum_{k=1}^{K}\pi_k\Phi_k(\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S)\prod_{\substack{j\in I_f\\ M_{i,j}=0}}\Phi_k(x_{i,j};\mu_{j,k},\Sigma_{j,k}) && (10.19)
\end{aligned}
$$

Where

$$
\begin{aligned}
\boldsymbol{\pi} &= \bigotimes_{j\in I_f}\boldsymbol{\pi}_j \text{ (Kronecker product)} && (10.20)\\
K &= |\boldsymbol{\pi}| && (10.21)\\
\boldsymbol{\mu}_{X^{I_f}} &= \prod_{j\in I_f}\boldsymbol{\mu}_j \text{ (Cartesian product)} && (10.22)\\
\boldsymbol{\sigma}_X &= \prod_{j\in I_f}\boldsymbol{\sigma}_j \text{ (Cartesian product)} && (10.23)
\end{aligned}
$$

with $\boldsymbol{\pi}_j,\mu_{j,k},\Sigma_{j,k}$ are estimated once before the MCMC starts (by Mixmod for example).

$\forall 1 \leq i \leq n, \forall 1 \leq k \leq K$ we have

$$\begin{aligned}
\Phi_k(\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S) &= \Phi_k(\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\mu}_{\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f},k},\boldsymbol{\Sigma}_{\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f},k}) && (10.24)\\
P(\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f};\boldsymbol{\alpha},\Theta,S) &= \Phi_k(\boldsymbol{X}_{i,O}^{I_r};\boldsymbol{\mu}_{\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f},k},\boldsymbol{\Sigma}_{\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f},k}) && (10.25)\\
\boldsymbol{\mu}_{\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f},k} &= \boldsymbol{\mu}_{\boldsymbol{X}_{i,O}^{I_r},k} + \boldsymbol{\Sigma}_{X_{i,O}^{I_r},X_{i,O}^{I_f},k}(\boldsymbol{\Sigma}_{X_{i,O}^{I_f},X_{i,O}^{I_f},k})^{-1}({}^t\boldsymbol{X}_{i,O}^{I_f} - \boldsymbol{\mu}_{X_{i,O}^{I_f},k}) && (10.26)\\
\boldsymbol{\Sigma}_{\boldsymbol{X}_{i,O}^{I_r}|\boldsymbol{X}_{i,O}^{I_f},k} &= \boldsymbol{\Sigma}_{X_{i,O}^{I_r},X_{i,O}^{I_r},k} - \boldsymbol{\Sigma}_{X_{i,O}^{I_r},X_{i,O}^{I_f},k}(\boldsymbol{\Sigma}_{X_{i,O}^{I_f},X_{i,O}^{I_f},k})^{-1}\boldsymbol{\Sigma}_{X_{i,O}^{I_f},X_{i,O}^{I_r},k} && (10.27)\\
\forall j \in I_r: \quad \boldsymbol{\mu}_{X_{i,O}^{j}} &= \sum_{l \in I_f^j} \alpha_{l,j}\mu_{l,k} && (10.28)
\end{aligned}$$

$\forall j \in I_r$ with $M_{i,j} = 0$

$$\mathrm{var}_k(x_{i,j}) = \sigma_j^2 + \sum_{l \in I_f^j} \alpha_{l,j}^2 \sigma_{X^l,k}^2 \tag{10.29}$$

$\forall j \notin I_r$ with $M_{i,j} = 0$

$$\mathrm{var}_k(x_{i,j}) = \sigma_{X^j,k}^2 \tag{10.30}$$

$\forall j_1 \in I_r, j_2 \in I_r, I_f^{j_1} \cap I_f^{j_2} \neq \emptyset$ with $M_{i,j_1} = M_{i,j_2} = 0$

$$\mathrm{cov}_k(x_{i,j_1}, x_{i,j_2}) = \sum_{l \in I_f^{j_1} \cap I_f^{j_2}} \alpha_{l,j_1}\alpha_{l,j_2}\,\mathrm{var}_k(x_{i,l}) = \sum_{l \in I_f^{j_1} \cap I_f^{j_2}} \alpha_{l,j_1}\alpha_{l,j_2}\sigma_{X^l,k}^2 \tag{10.31}$$

$\forall j_1 \in I_r, j_2 \in I_r, I_f^{j_1} \cap I_f^{j_2} = \emptyset$ with $M_{i,j_1} = M_{i,j_2} = 0$

$$\mathrm{cov}_k(x_{i,j_1}, x_{i,j_2}) = 0 \tag{10.32}$$

$\forall j_1 \in I_f, j_2 \in I_f$ with $M_{i,j_1} = M_{i,j_2} = 0$

$$\mathrm{cov}_k(x_{i,j_1}, x_{i,j_2}) = 0 \tag{10.33}$$

$\forall j_1 \in I_r, j_2 \in I_f^{j_1}$ with $M_{i,j_1} = M_{i,j_2} = 0$

$$\mathrm{cov}_k(x_{i,j_1}, x_{i,j_2}) = \alpha_{j_2,j_1}\sigma_{X^{j_2},k}^2 \tag{10.34}$$

$\forall j_1 \in I_r, j_2 \notin I_f^{j_1} \cup I_r$ with $M_{i,j_1} = M_{i,j_2} = 0$

$$\mathrm{cov}_k(x_{i,j_1}, x_{i,j_2}) = 0 \tag{10.35}$$

10.2.2 Likelihood computation optimized

The main problem with the likelihood in its global form (10.19) is that the number of components explodes so we can't use it in practice. But in many case, it can be simplified. We see that the 0 in the variance-covariance matrix does not depend on the component k so the structure of sparsity of $\boldsymbol{\Sigma}$ can be stored and used back in each iteration for a given structure S to reduce computing time. Another strictly technical tip would be to use sparse matrix storage to avoid null value storage and useless zero multiplications. Moreover, we can look if there are missing values shared by several sub-regression. Connex component detection could be done to reduce the dimension down to strictly dependent covariates and use equation (10.12) elsewhere. We just need to compute the row-sums of the adjacency matrix G or to search for redundancy in I_f and then if there is no redundancy or if $\forall j$ redundant we have $\sum_{i=1}^n M_{i,j} = 0$ then we can use the simplified form of the likelihood given in (10.16). For faster computation we can

stock the vector of covariates that have missing values. So the true value of the likelihood can be computed efficiently in most of cases but in the MCMC, it remains the possibility to have a structure with explosive likelihood expression when combined with the missing values and we need to compute the likelihood for a great number of candidates. Then it is possible to use directly the simplified form of the likelihood, that can be seen as an approximation of the likelihood, not taking into account some of the dependencies but it would offer no guarantee in terms of efficiency for the MCMC.

10.2.3 Weighted penalty

Now we have defined the way to compute the likelihood, other questions remain : how to define the number of parameters in the structure ? How to take into account missingness (structures relying on highly missing covariates should be penalized) ? We have seen that for a same covariate X^j with $j \in I_r$, the number of parameters is not the same for each individual depending whether or not $M_{i,j} = 0$. But the penalty (for $\psi = BIC$) can't be added at the individual level (because $\log(1) = 0$ so it would be annihilated).

To penalize models that suppose dependencies based only on a few individuals, we propose to use the mean of the complexities obtained for a given covariate.

$$k_j = \frac{1}{n} \sum_{i=1}^{n} k_{i,j} \tag{10.36}$$

where $k_{i,j}$ is the number of parameter to estimate in $P(x_{i,j}|\boldsymbol{X}_i \setminus \boldsymbol{X}_i^j)$.

$$\begin{aligned} -2\log P(\boldsymbol{X}|S) &\approx BIC = -2\mathcal{L}(\boldsymbol{X}, S, \boldsymbol{\Theta}) + |\boldsymbol{\Theta}| \log(n) && (10.37) \\ &= -2\mathcal{L}(\boldsymbol{X}, S, \boldsymbol{\Theta}) + (\sum_{j=1}^{p} k_j) \log(n) && (10.38) \end{aligned}$$

Thus if a structure is only touched by one missing value the penalty will be smaller than another same shaped structure but with more missing values implied. Another way would be to use $\psi = RIC$ (see [Foster and George, 1994]) so the complexity is associated with $\log(p)$ and can be added individually. Or to make a compromise and penalize by $\frac{k_i \log(p)}{\log(n)}$.

10.3 SEM

The integrated likelihood depicted above depends on $\boldsymbol{\alpha}$ which was formerly estimated by OLS when there was no missing values. But when missing values occurs in a sub-regression we need another solution.

We use a Stochastic Expectation Maximization (SEM) algorithm [Celeux and Diebolt, 1986] to estimate $\boldsymbol{\alpha}$ because missing values do not allow to use OLS and the log-likelihood (10.4) is not linear so a simple Expectation-Maximization (EM) would be difficult to compute.

10.3.1 Our implementation of SEM

initialization: We start with some imputation (for example by the mean) for each missing value (done only once for the MCMC). $\boldsymbol{\alpha}^{(0)}$ can be initialized by cutting method (sparse structure) or using imputed values in $\boldsymbol{X}$. At iteration h,

SE step: We generate the missing values according to $P(\boldsymbol{X}_M|\boldsymbol{X}_O; \alpha^{(h)}, \Theta, S)$, that is stochastic imputation.

M step: We estimate

$$\boldsymbol{\alpha}^{(h+1)} = \text{argmax}_{\boldsymbol{\alpha}} E\left[\mathcal{L}(\boldsymbol{X}|\boldsymbol{\alpha}, S, \Theta)\right] \tag{10.39}$$

and we can use the same method as the one for classical case without missing values (OLS, SUR, *etc.*). We continue until convergence ($\| \boldsymbol{\alpha}^{(h+1)} - \boldsymbol{\alpha}^{(h)} \| < tol$ where tol is the tolerance). Then we make m iterations and take $\hat{\boldsymbol{\alpha}}$ as the mean of these m last iterations.

Another (faster but not optimal) way would be to only use the structure for $\boldsymbol{X}^{I_r}$ and use the distribution given by Mixmod for $\boldsymbol{X}^{I_f}$ along the MCMC. The full SEM would then be used only once with the final structure to make imputation in $\boldsymbol{X}$ before using variable selection methods like the LASSO.

10.3.2 Stochastic imputation by Gibbs sampling

We use a Gibbs sampling method to generate the missing values at the SE step. $\boldsymbol{X}$ follows a multivariate Gaussian mixture with K component and we note Z the set of the $Z_{i,j}$ indicating the component from which $x_{i,j}$ is generated.

Initialisation: all the $z_{i,j}$ are set to the first component (such an initialisation does not depend on K) and $\boldsymbol{X}_M$ are imputed by the marginal means.

Iteration: At each iteration of the Gibbs sampler:
$\forall x_{i,j} \in \boldsymbol{X}_M^{I_r}$: $x_{i,j}$ is generated according to

$$\begin{aligned} P(x_{i,j}|\boldsymbol{X}_{i,O}, \boldsymbol{X}_{i,\bar{M}_{i,j}}, Z; \boldsymbol{\alpha}^{(h)}, \Theta, S) &= P(x_{i,j}|\boldsymbol{X}_{i,O}, \boldsymbol{X}_{i,\bar{M}_{i,j}}; \alpha^{(h)}, \Theta, S) & (10.40)\\ &= P(x_{i,j}|\boldsymbol{X}_i^{I_f^j}; \boldsymbol{\alpha}^{(h)}, \Theta, S) = \mathcal{N}(\boldsymbol{X}_i^{I_f^j} \boldsymbol{\alpha}^{(h)}_{I_f^j, j}; \sigma_j^2) & (10.41)\end{aligned}$$

We have $P(\boldsymbol{X}|Z) = \mathcal{N}(\boldsymbol{\mu}_{|Z}, \boldsymbol{\Sigma}_{|Z})$.
$\forall x_{i,j} \in \boldsymbol{X}_M^{I_f}$: $x_{i,j}$ is generated according to

$$P(x_{i,j}|\boldsymbol{X}_{i,O}, \boldsymbol{X}_{i,\bar{M}_{i,j}}, Z; \boldsymbol{\alpha}^{(h)}, \Theta, S) = P(x_{i,j}|\boldsymbol{X}_{i,\bar{j}}, Z_i; \boldsymbol{\alpha}^{(h)}, \Theta, S) \tag{10.42}$$

$$= \mathcal{N}(\mu_{j|Z_{i,j}} + \boldsymbol{\Sigma}_{j,X_{i\bar{j}}|Z_i} \boldsymbol{\Sigma}^{-1}_{X_{i\bar{j}},X_{i\bar{j}}|Z_i} (X_{i\bar{j}} - \boldsymbol{\mu}_{X_{i\bar{j}}|Z_i}); \sigma^2_{j|Z_{i,j}} - \boldsymbol{\Sigma}_{j,X_{i\bar{j}}|Z_i} \boldsymbol{\Sigma}^{-1}_{X_{i\bar{j}},X_{i\bar{j}}|Z_i} \boldsymbol{\Sigma}'_{j,X_{i\bar{j}}|Z_i}) \tag{10.43}$$

Where all the values needed here were described above for the likelihood computation.

Then, $\forall 1 \leq i \leq n, \forall j \in I_f$ we draw new values for $Z_{i,j}$ according to

$$\begin{aligned} P(Z_{i,j}|\boldsymbol{X}, Z_{i,j}^-; \Theta, \boldsymbol{\alpha}, S) &= P(Z_{i,j}|\boldsymbol{X}_i, Z_{i,\bar{j}}; \Theta, \boldsymbol{\alpha}, S) = \mathcal{M}(t_{i,j,1}, \ldots, t_{i,j,K_j}) & (10.44)\\ \text{where } t_{i,j,k} &= \frac{\pi_{j,k} \Phi(x_{i,j}; \mu_{j,k}, \sigma^2_{j,k})}{\sum_{l=1}^{K_j} \pi_{j,l} \Phi(x_{i,j}; \mu_{j,l}, \sigma^2_{j,l})} & (10.45)\end{aligned}$$

We see that $Z_{i,j}$ are not used if there is no missing values in $\boldsymbol{X}_i$ and others are not all needed so we can also optimize computation time by computing only the $Z_{i,j}$ that are needed in the Gibbs. For the last iteration of the Gibbs, in the last iteration of the SEM, we do not need to draw Z.

Instead of using long chain for each Gibbs, we can use small chains because SEM iteration will simulate longer chains so it remains efficient with a smaller computation cost.

Computation cost will be the main purpose here because we need an iterative algorithm (Gibbs sampler) at each iteration of another iterative algorithm (SEM) for each candidate of the MCMC. So alternative method should be preferred for large datasets with many missing values and only a small amount of time.

Because K can be very large we search a way to compute the likelihood. We can use a Gibbs algorithm to estimate the likelihood:

$$P(\boldsymbol{X}_O;\Theta,S,\boldsymbol{\alpha}) = \sum_{Z\in\mathcal{Z}}\int_{\boldsymbol{X}_M}\frac{P(\boldsymbol{X}_M,Z,\boldsymbol{X}_O;\Theta,\boldsymbol{\alpha},S)}{P(\boldsymbol{X}_M,Z|\boldsymbol{X}_O;\Theta,\boldsymbol{\alpha},S)}P(\boldsymbol{X}_M,Z|\boldsymbol{X}_O;\Theta,\boldsymbol{\alpha},S)dX \quad (10.46)$$

$$\approx \frac{1}{Q}\sum_{q=1}^{Q}\frac{P(\boldsymbol{X}_M^{(q)},\boldsymbol{X}_O,Z^{(q)};\Theta,\boldsymbol{\alpha},S)}{P(\boldsymbol{X}_M^{(q)}|\boldsymbol{X}_O,Z^{(q)};\Theta,\boldsymbol{\alpha},S)} \text{ by the law of large numbers} \quad (10.47)$$

where Q is the number of iterations of the Gibbs sampler. But to be faster, we use the previous Gibbs algorithm with:

$$P(\boldsymbol{X}_O;\Theta,S,\boldsymbol{\alpha}) \approx \frac{1}{Q}\sum_{q=1}^{Q}\frac{P(\boldsymbol{X}_M^{(q)},\boldsymbol{X}_O,Z^{(q)};\Theta,\boldsymbol{\alpha}^{(q)},S)}{P(\boldsymbol{X}_M^{(q)}|\boldsymbol{X}_O,Z^{(q)};\Theta,\boldsymbol{\alpha}^{(q)},S)} \quad (10.48)$$

10.3.3 Alternative E step

If we can't (or don't want to) compute the SE step described above, then we can use alternative imputation step for missing data based on $\boldsymbol{\alpha}$ (and keep the alternate optimisation to find the best $\boldsymbol{\alpha}$).

$\forall x_{i,j}\in\boldsymbol{X}_M$ we have:
if $j\in I_r$, Equation(10.12) gives:

$$E[x_{i,j}|\boldsymbol{\alpha}^{(h)},\Theta,\boldsymbol{X}_O,S] = E[\sum_{k=1}^{k_{ij}}\pi_{ij,k}\Phi(x_{i,j}|\mu_{ij,k},\Sigma_{ij,k})|\boldsymbol{\alpha}^{(h)},\Theta,\boldsymbol{X}_O,S] \quad (10.49)$$

Let $r_{i,j}=\{l\in I_r|\boldsymbol{\alpha}_{j,l}\neq 0,\boldsymbol{M}_{i,j}=0\}$ the set of observed covariates for individual i that are explained by $x_{i,j}$ according to S.
If $j\notin I_r$ we can do:

$$E[x_{i,j}|\boldsymbol{\alpha}^{(h)},\Theta,\boldsymbol{X}_O,S] = \frac{1}{|r_{i,j}|}\sum_{k\in r_{i,j}}E_{|\boldsymbol{\alpha}^{(h)},\Theta,\boldsymbol{X}_O,S}\left[\frac{1}{\alpha_{j,k}}\left(x_{i,k}-\varepsilon_k(i)-\sum_{l\in I_f^k}x_{i,l}\alpha_{l,k}\right)\right] \quad (10.50)$$

$$= \frac{1}{|r_{i,j}|}\sum_{k\in r_{i,j}}E_{|\boldsymbol{\alpha}^{(h)},\Theta,\boldsymbol{X}_O,S}\left[\frac{1}{\alpha_{j,k}}\left(x_{i,k}-\sum_{l\in I_f^k}x_{i,l}\alpha_{l,k}\right)\right] \quad (10.51)$$

that is the mean of the expectations of the inverse sub-regressions implying x_i,j with value in $\boldsymbol{X}_i^{I_r}$ not missing.

10.4 Missing values in the main regression

The easier way would be to draw missing values with the SEM described above and then use classical methods on the completed dataset, with the possibility to repeat this procedure a few times and then take the mean. We should for example try multiple draw and LASSO for variable selection like variable selection by random forest. One great advantage of multiple draw procedures is that it gives an idea of the precision of the imputations with the variance of these imputed values among the multiple draws. So we know whether it is reliable or not.

But another way would be to consider classical estimation methods as likelihood optimizer and then adapt them to the integrated likelihood of our model. Thus we can imagine to use LASSO without imputation. But the choice of the penalty using the LAR algorithm need also to adapt the LAR that is based on correlations that are computed on vectors with distinct number of individuals (due to missing values). So it requires more work but could be a good perspective for our method.

10.5 Numerical results on simulated datastets

10.5.1 Estimation of the sub-regression coefficients

We take datasets from the experiments in part I and then we compare the MSE obtained on $\boldsymbol{\alpha}$ with our SEM to those obtain by classical OLS after imputation of the missing values by the marginal empirical means. Here $p = 40$ and $n = 30$, missing values position are generated randomly for each of the 100 datasets to obtain 10% of missing values each time. Thus we have 120 missing values and none of the datasets contain a full individual without missing values. Both methods were tested with the true structure S. Initial value of $\boldsymbol{\alpha}$ for the SEM was the result of the method using imputation by the empirical mean. Only 10 iterations for the SEM after 2 warming steps with only 1 iteration for the Gibbs at each step.

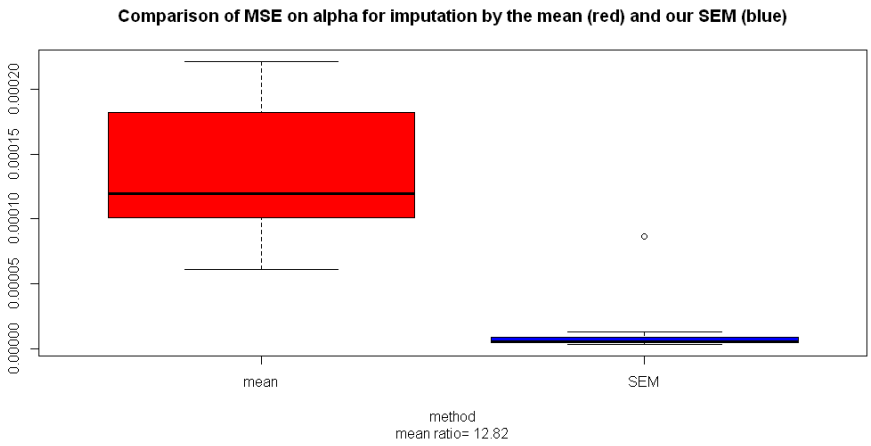


Figure 10.1: MSE on $\boldsymbol{\alpha}$ is significantly lower and more robust with our SEM than with imputation by the mean.

We see (Figure 10.1) that our SEM is nearly 13 times more efficient in mean that estimation based on imputation by the mean. Our results are extremely good because each sub-regression is true and we have 30 individuals (even if missing values kind of reduce this number) to estimate 3 coefficients only each time. Although, using imputed values lead to learn a true regression with a factually incorrect dataset. Thus we should prefer to work without imputing the missing values but using the full generative model and the dependencies it implies. Imputation will always introduce some bias.

10.5.2 Imputation by the sub-regression

We have then imputed missing values in $\boldsymbol{X}^{I_r}$ by using the corresponding sub-regressions after $\boldsymbol{\alpha}$ has been estimated by the SEM. Missing values in $\boldsymbol{X}^{I_f}$ are estimated by the mean of 50 Gibbs iterations after the SEM and 2 warming steps of the Gibbs. Results are shown in figure 10.2.

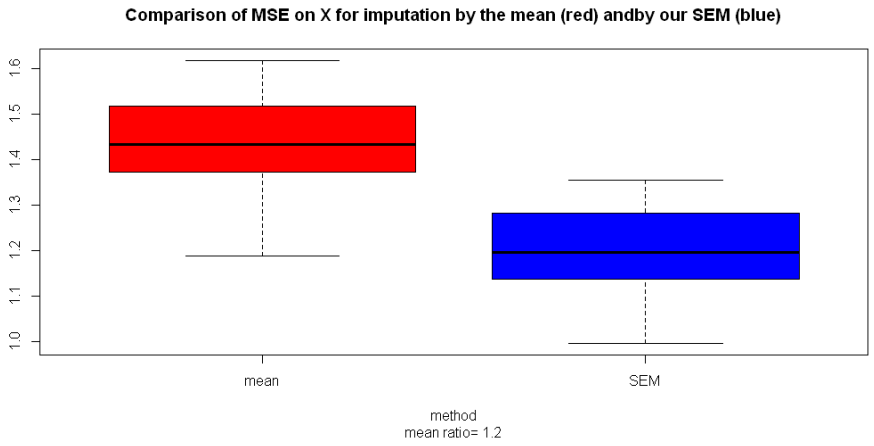


Figure 10.2: MSE on $\hat{\boldsymbol{X}}$ is significantly lower and more robust when using our SEM than with imputation by the mean.

10.5.3 Multiple imputation for the main regression

We use the previously imputed $\boldsymbol{X}$ to estimate $\boldsymbol{Y}$ with $\boldsymbol{\beta} = \mathbf{1}$ and $\sigma_Y = 10$.

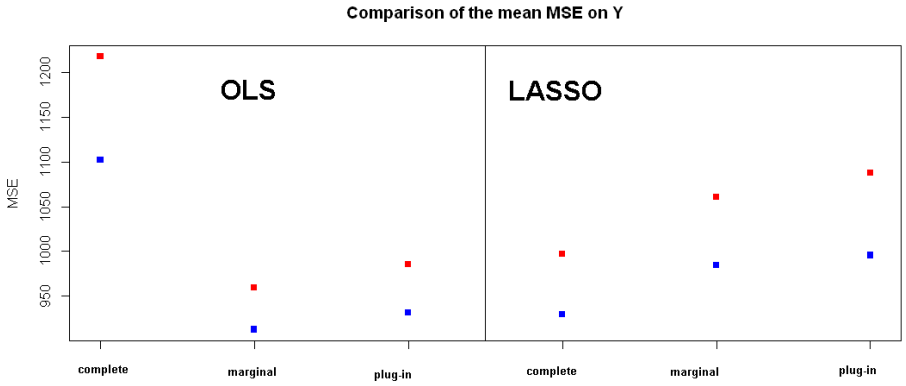


Figure 10.3: MSE on $\hat{\boldsymbol{Y}}$ are lower when using our SEM (blue) than with imputation by the mean (red) for the three model (complete, marginal and plug-in) using OLS or LASSO

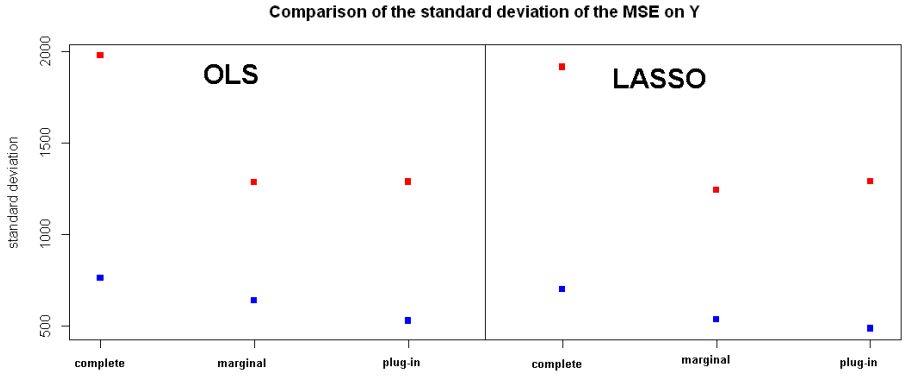


Figure 10.4: our SEM (blue) provides more robust results than imputation by the mean (red) but the variances are still too wide.

We obtain on a validation sample of 1000 individuals a predictive MSE smaller in mean with our method (Figure 10.3). But the variance are too important to really conclude (Figure 10.4). We can say that imputation by SEM is more robust, but the Gibbs do not give satisfying results. Maybe the increase of the number of steps allowed by a code optimization would help to improve these results. For now, we can just say that our generative model significantly improves estimation of $\boldsymbol{\alpha}$ and make possible to find S based on a dataset with missing values.

One big advantage with our regression model is that it does not depend on the response variable $\boldsymbol{Y}$ so the structure can be learnt independently. Thus we can imagine to obtain big samples to learn the structure without being annoyed by the missing values. Then when a response variable is chosen, we can keep the same S and use previously computed values of $\boldsymbol{\alpha}$ as initial value for the SEM.

10.6 Missing values on real datasets

To be able to evaluate the results on a real dataset, we have deleted some values in the production dataset from section 8.2 to obtain 10% of missing values. Figure 10.5 shows the pattern of the missing values (MCAR). It confirms that 10% of missing values is sufficient to have no complete line or column in the dataset.

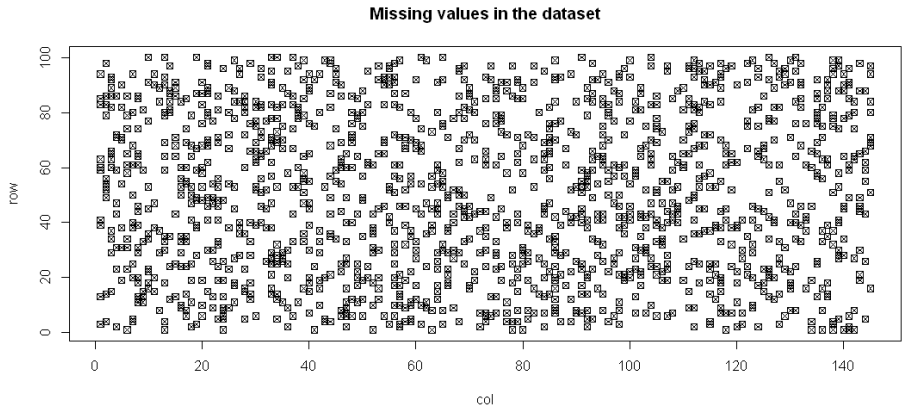


Figure 10.5: Graphical representation of the dataset with 10% of missing values

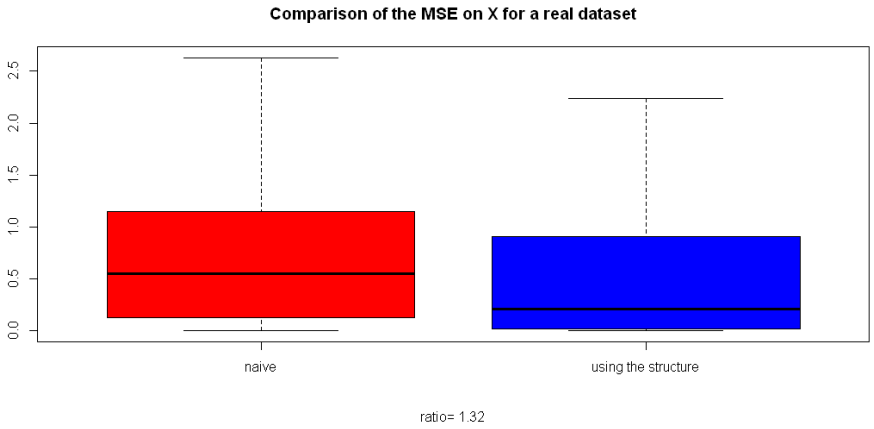


Figure 10.6: MSE on $\hat{\boldsymbol{X}}$ is 1.32 times lower in mean when using our method (blue) than with imputation by the mean (red).

We see in figure 10.6 that our SEM gives a smaller MSE with a smaller variance than imputation by the mean. There is still a lot of work to do in the field of missing values and we have only walked the firsts steps of this huge perspective, but these first results are encouraging and make us feel like this orientation is worthy to be followed.

Chapter 11

CorReg: the concept

The `CorReg` package is already downloadable on the CRAN under CeCILL Licensing. This package permits to generate datasets according to our generative model, to estimate the structure (C++ code) of regression within a given dataset and to estimate both explicative and predictive model with many regression tools (OLS,stepwise,LASSO,elasticnet,clere,spike and slab, adaptive lasso and every models in the LARS package). So every simulation presented above can be done with `CorReg`. `CorReg` also provides tools to interpret found structures and visualize the dataset (missing values and correlations).
The objective of `CorReg` is also to bring recent statistical tools to engineers. Thus it will be made available in Microsoft Excel for the end of the year 2014, probably using Basic Excel R Toolkit(BERT[1]).
It also provides some small scripts put in functions to obtain graphical representations and basic statistics with legends for non-statistician with only one command line (or macro button in Excel).
One example of graphical tool is the `matplot_zone` function that allows to compare several curves according to a given function (input parameter) and was widely used to compare the MSE and complexities in this document. Another example is the `recursive_tree` function to plot classification and regression trees with basic statistics and legend but also to successively compute trees removing some correlated covariates or covariates that cannot be changed in the process to see if they are replaced by others more useful (this recursive aspect has given its name to the function).
More features will be added as statistics will continue to be taught to engineers to provide ergonomic and powerful statistical tools to non-statisticians.

[1] https://github.com/StructuredDataLLC/Basic-Excel-R-Toolkit

Chapter 12

Conclusion and perspectives

12.1 Conclusion

It is well-known that no model is the better in every situation. Here we propose two additional models (marginal and plug-in) but the best idea is to compare the full, marginal and plug-in and then choose the best for the study concerned. Our goal was not to replace any model but to enlarge the scope of statisticians in the real life. It is important to note that our model can be useful for interpretation even if the full model is chosen for interpretation, because we explicitly describe the correlations between the covariates. Moreover, it is only a pretreatment so it could easily be used with future statistical tools.
Our model is easy to understand and to use. Usage of linear regression to model the correlations definitely separates us from "black boxes" so users are confident in what they do. The well-known and trivial sub-regression found comfort users in that if a structure does exist, `CorReg` will find it so when a new sub-regression, or a new main regression is given they are more likely to look further and try it. The automated aspect shows the power of statistics without a priori so users begin to understand that statistics are not only descriptive or predictive but based on *a priori* models. This method seems to have a positive impact on the way users looks at the statistics (according to them).
It is good to see that sequential methods (plug-in model) and automation can produce good results. Probabilistic models are efficient even without human expertise and let the experts improve the results by adding their expertise in the model (coercing some sub-regression for example). So we hope that statistics will continue to be a central tool for engineers.

12.2 Perspective

12.2.1 Non-linear regression

Polynomial regression, logistic regression [Hosmer and Lemeshow, 2000], *etc.* might be improved by a method like this.

12.2.2 Pretreatment not only for regression

Classification and Regression Tree, and any other method could benefit of the variable selection pretreatment implied by our marginal model.

12.2.3 Improved programming

Even if it is written in C++, the algorithm could be optimized by a better usage of sparse matrices, memory usage optimization, and other small things that could reduce computational cost to be faster and allow to work with larger datasets (already works with thousands of covariates).

12.2.4 Missing values in classical methods

The full generative approach could be used to manage missing values without imputation for many classical methods. It can notably be used for clustering and not only in response variable prediction context.

12.2.5 Interpretation improvements

Ergonomy of the software could be improved to better fit industrial needs. This work is in progress and further work will be provided just after the redaction of this thesis to reach this goal.

Bibliography

[Abdi, 2003] Abdi, H. (2003). Partial least squares regression (pls-regression).

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

[Amemiya, 1985] Amemiya, T. (1985). *Advanced econometrics*. Harvard university press.

[Anderson, 1984] Anderson, T. W. (1984). Estimating linear statistical relationships. *The Annals of Statistics*, pages 1–45.

[Andrieu and Doucet, 1999] Andrieu, C. and Doucet, A. (1999). Joint bayesian model selection and estimation of noisy sinusoids via reversible jump mcmc. *Signal Processing, IEEE Transactions on*, 47(10):2667–2676.

[Arlot et al., 2010] Arlot, S., Celisse, A., et al. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79.

[Auder et al., 2014] Auder, B., Lebret, R., Iovleff, S., and Langrognet, F. (2014). *Rmixmod: An interface for MIXMOD*. R package version 2.0.2.

[Biernacki et al., 2006] Biernacki, C., Celeux, G., Govaert, G., and Langrognet, F. (2006). Model-based cluster and discriminant analysis with the mixmod software. *Computational Statistics & Data Analysis*, 51(2):587–600.

[Biggs, 1993] Biggs, N. (1993). *Algebraic graph theory*. Cambridge University Press.

[Bondell and Reich, 2008] Bondell, H. and Reich, B. (2008). Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123.

[Bondy and Murty, 1976] Bondy, J. and Murty, U. (1976). *Graph theory with applications*, volume 290. Macmillan London.

[Breiman, 1984] Breiman, L. (1984). *Classification and regression trees*. Chapman & Hall/CRC.

[Breiman and Friedman, 1997] Breiman, L. and Friedman, J. (1997). Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(1):3–54.

[Brito and Pearl, 2006] Brito, C. and Pearl, J. (2006). Graphical condition for identification in recursive sem.

[Bulirsch and Stoer, 2002] Bulirsch, R. and Stoer, J. (2002). *Introduction to numerical analysis*. Springer Heidelberg.

[Cameron and Trivedi, 2005] Cameron, A. C. and Trivedi, P. K. (2005). *Microeconometrics: methods and applications*. Cambridge university press.

[Casella and George, 1992] Casella, G. and George, E. (1992). Explaining the gibbs sampler. *American Statistician*, pages 167–174.

[Celeux and Diebolt, 1986] Celeux, G. and Diebolt, J. (1986). L'algorithme sem: un algorithme d'apprentissage probabiliste: pour la reconnaissance de mélange de densités. *Revue de statistique appliquée*, 34(2):35–52.

[Celeux and Govaert, 1992] Celeux, G. and Govaert, G. (1992). A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3):315–332.

[Chauveau, 1995] Chauveau, D. (1995). A stochastic {EM} algorithm for mixtures with censored data. *Journal of Statistical Planning and Inference*, 46(1):1 – 25.

[Chib and Greenberg, 1995] Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *American Statistician*, pages 327–335.

[Chipman et al., 2001] Chipman, H., George, E., McCulloch, R., Clyde, M., Foster, D., and Stine, R. (2001). The practical implementation of bayesian model selection. *Lecture Notes-Monograph Series*, pages 65–134.

[Chiquet J. and S., 2013] Chiquet J., M.-H. T. and S., R. (2013). Multi-trait genomic selection via multivariate regression with structured regularization. *Proceedings of MLCB/NIPS'13 workshop*.

[Cule, 2014] Cule, E. (2014). *ridge: Ridge Regression with automatic selection of the penalty parameter*. R package version 2.1-3.

[Cule and De Iorio, 2013] Cule, E. and De Iorio, M. (2013). Ridge regression in prediction problems: automatic choice of the ridge parameter. *Genetic epidemiology*, 37(7):704–714.

[d'Aspremont et al., 2008] d'Aspremont, A., Banerjee, O., and El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66.

[Davidson and MacKinnon, 1993] Davidson, R. and MacKinnon, J. (1993). Estimation and inference in econometrics. *Oxford University Press Catalogue*.

[Dempster, 1972] Dempster, A. (1972). Covariance selection. *Biometrics*, pages 157–175.

[Dempster et al., 1977] Dempster, A., Laird, N., Rubin, D., et al. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38.

[Diebolt and Ip, 1996] Diebolt, J. and Ip, E. (1996). A stochastic em algorithm for approximating the maximum likelihood estimate. *Markov chain Monte Carlo in practice*.

[Dodge and Rousson, 2004] Dodge, Y. and Rousson, V. (2004). Analyse de régression appliquée: manuel et exercices corrigés (coll. eco sup,). *Recherche*, 67:02.

[Edwards, 1984] Edwards, A. W. (1984). *Likelihood T*. CUP Archive.

[Efron, 1979] Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The annals of Statistics*, pages 1–26.

[Efron et al., 2004] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

[Efron and Tibshirani, 1994] Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*, volume 57. CRC press.

[Er et al., 2013] Er, M. J., Shao, Z., and Wang, N. (2013). A systematic method to guide the choice of ridge parameter in ridge extreme learning machine. In *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, pages 852–857. IEEE.

[Eubank, 1999] Eubank, R. L. (1999). *Nonparametric regression and spline smoothing*. CRC press.

[Fan and Li, 2001] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

[Fausett, 1994] Fausett, L. (1994). Fundamentals of neural networks: architectures, algorithms, and applications. *Englewood Cliffs*.

[Foster and George, 1994] Foster, D. and George, E. (1994). The risk inflation criterion for multiple regression. *The Annals of Statistics*, pages 1947–1975.

[Friedman et al., 2008] Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

[Friedman et al., 2010] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Applications of the lasso and grouped lasso to the estimation of sparse graphical models. Technical report, Technical report, Stanford University.

[Friedman et al., 2000] Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620.

[Gareth M.J. and R., 2013] Gareth M.J., C. P. and R., P. (2013). The constrained lasso. *URL: http://www-bcf.usc.edu/˜gareth/research/CLassoFinal.pdf*.

[Geladi and Kowalski, 1986] Geladi, P. and Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17.

[George, 2000] George, E. (2000). The variable selection problem. *Journal of the American Statistical Association*, 95(452):1304–1308.

[George and McCulloch, 1993] George, E. and McCulloch, R. (1993). Variable selection via gibbs sampling. *Journal of the American Statistical Association*, pages 881–889.

[Gilks et al., 1996] Gilks, W., Richardson, S., and Spiegelhalter, D. (1996). *Markov chain Monte Carlo in practice*, volume 2. CRC press.

[Goldberger, 1984] Goldberger, A. S. (1984). Reverse regression and salary discrimination. *Journal of Human Resources*, pages 293–318.

[Gower and Ross, 1969] Gower, J. and Ross, G. (1969). Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18(1):54–64.

[Graham and Hell, 1985] Graham, R. and Hell, P. (1985). On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57.

[Greene, 1984] Greene, W. H. (1984). Reverse regression: The algebra of discrimination. *Journal of Business and Economic Statistics*, 2(2):pp. 117–120.

[Grinstead and Snell, 1997] Grinstead, C. M. and Snell, J. (1997). *Introduction to probability*. American Mathematical Society.

[Hardle, 1990] Hardle, W. (1990). *Applied nonparametric regression*, volume 27. Cambridge Univ Press.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer.

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243.

[Hoerl and Kennard, 1970] Hoerl, A. and Kennard, R. (1970). Ridge regression: applications to nonorthogonal problems. *Technometrics*, pages 69–82.

[Hosmer and Lemeshow, 2000] Hosmer, D. and Lemeshow, S. (2000). *Applied logistic regression*, volume 354. Wiley-Interscience.

[Hox, 1998] Hox, J. (1998). Multilevel modeling: When and why. In *Classification, data analysis, and data highways*, pages 147–154. Springer.

[Ishwaran and Rao, 2005] Ishwaran, H. and Rao, J. (2005). Spike and slab variable selection: frequentist and bayesian strategies. *Annals of Statistics*, pages 730–773.

[Isobe et al., 1990] Isobe, T., Feigelson, E., Akritas, M., and Babu, G. (1990). Linear regression in astronomy. *The astrophysical journal*, 364:104–113.

[Jackson, 2005] Jackson, J. E. (2005). *A user's guide to principal components*, volume 587. John Wiley & Sons.

[Jensen and Nielsen, 2007] Jensen, F. and Nielsen, T. (2007). *Bayesian networks and decision graphs*. Springer Verlag.

[Karalic, 1992] Karalic, A. (1992). Linear regression in regression tree leaves. In *In Proceedings of ECAI-92*, pages 440–441. John Wiley and Sons.

[Katsikis and Pappas, 2008] Katsikis, V. and Pappas, D. (2008). Fast computing of the moore-penrose inverse matrix. *Electronic Journal of Linear Algebra*, 17(1):637–650.

[Kiebel and Holmes, 2003] Kiebel, S. and Holmes, A. (2003). The general linear model. *Human brain function*, 2:725–760.

[Kohavi et al., 1995] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145.

[Le Cessie and Van Houwelingen, 1992] Le Cessie, S. and Van Houwelingen, J. (1992). Ridge estimators in logistic regression. *Applied statistics*, pages 191–201.

[Leamer, 1978] Leamer, E. E. (1978). Least-squares versus instrumental variables estimation in a simple errors in variables model. *Econometrica: Journal of the Econometric Society*, pages 961–968.

[Lebarbier and Mary-Huard, 2006] Lebarbier, É. and Mary-Huard, T. (2006). Une introduction au critère bic: fondements théoriques et interprétation. *Journal de la SFdS*, 147(1):39–57.

[Li, 1991] Li, K. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, pages 316–327.

[Little, 1992] Little, R. (1992). Regression with missing x's: a review. *Journal of the American Statistical Association*, 87(420):1227–1237.

[Longford, 2012] Longford, N. (2012). A revision of school effectiveness analysis. *Journal of Educational and Behavioral Statistics*, 37(1):157–179.

[Maas and Hox, 2004] Maas, C. J. and Hox, J. J. (2004). Robustness issues in multilevel regression analysis. *Statistica Neerlandica*, 58(2):127–137.

[Marquardt and Snee, 1975] Marquardt, D. and Snee, R. (1975). Ridge regression in practice. *American Statistician*, pages 3–20.

[Massart and Picard, 2007] Massart, P. and Picard, J. (2007). *Concentration inequalities and model selection*, volume 1896. Springer.

[Maugis et al., 2009] Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009). Variable selection in model-based clustering: A general variable role modeling. *Computational Statistics & Data Analysis*, 53(11):3872–3882.

[McCullagh and Nelder, 1989] McCullagh, P. and Nelder, J. A. (1989). Generalized linear models.

[McLachlan and Krishnan, 2007] McLachlan, G. and Krishnan, T. (2007). *The EM algorithm and extensions*, volume 382. John Wiley & Sons.

[McLachlan and Peel, 2004] McLachlan, G. and Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.

[Miller, 2002] Miller, A. (2002). *Subset selection in regression*. CRC Press.

[Moerbeek et al., 2003] Moerbeek, M., van Breukelen, G. J., and Berger, M. P. (2003). A comparison between traditional methods and multilevel regression for the analysis of multicenter intervention studies. *Journal of clinical epidemiology*, 56(4):341–350.

[Montgomery et al., 2012] Montgomery, D., Peck, E., and Vining, G. (2012). *Introduction to linear regression analysis*, volume 821. John Wiley & Sons.

[Moret and Shapiro, 1991] Moret, B. and Shapiro, H. (1991). An empirical analysis of algorithms for constructing a minimum spanning tree. *Algorithms and Data Structures*, pages 400–411.

[Nelder and Baker, 1972] Nelder, J. A. and Baker, R. (1972). *Generalized linear models*. Wiley Online Library.

[Pearl, 1998] Pearl, J. (1998). Graphs, causality, and structural equation models. *Sociological Methods & Research*, 27(2):226–284.

[Pearl, 2000] Pearl, J. (2000). *Causality: models, reasoning, and inference*, volume 47. Cambridge Univ Press.

[Penrose, 1955] Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51:406–413.

[Pierro and Wei, 2000] Pierro, A. R. D. and Wei, M. (2000). Some new properties of the equality constrained and weighted least squares problem. *Linear Algebra and its Applications*, 320:145 – 165.

[Quinlan, 1986] Quinlan, J. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

[R Core Team, 2014] R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[Radchenko and James, 2008] Radchenko, P. and James, G. (2008). Variable inclusion and shrinkage algorithms. *Journal of the American Statistical Association*, 103(483).

[Raftery, 1995] Raftery, A. (1995). Bayesian model selection in social research. *Sociological methodology*, 25:111–164.

[Raftery and Dean, 2006] Raftery, A. E. and Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178.

[Raudenbush, 2002] Raudenbush, S. (2002). *Hierarchical linear models : applications and data analysis methods*. Sage Publications, Thousand Oaks.

[Reiersol, 1950] Reiersol, O. (1950). Identifiability of a linear relation between variables which are subject to error. *Econometrica*, 18(S 375).

[Roberts and Rosenthal, 2001] Roberts, G. and Rosenthal, J. (2001). Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, 16(4):351–367.

[Saporta, 2006] Saporta, G. (2006). *Probabilités, analyse des données et statistique*. Editions Technip.

[Saracco et al., 1999] Saracco, J., Larramendy, I., and Aragon, Y. (1999). La regression inverse par tranches ou méthode sir: presentation générale. *La revue de Modulad*, (22):21–39.

[Seber and Lee, 2012] Seber, G. and Lee, A. J. (2012). *Linear regression analysis*, volume 936. John Wiley & Sons.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

[Tibshirani et al.,] Tibshirani, R., Hoefling, G., Wang, P., and Witten, D. The lasso: some novel algorithms and applications.

[Timm, 2002] Timm, N. (2002). *Applied multivariate analysis*. Springer Verlag.

[Wang et al., 2011] Wang, S., Nan, B., Rosset, S., and Zhu, J. (2011). Random lasso. *The annals of applied statistics*, 5(1):468.

[Wickens, 2004] Wickens, T. (2004). The general linear model. *Institute for Pure and Applied Mathematics. URL: http://www. ipam. ucla. edu/publications/mbi2004/mbi2004_5017. pdf.*

[Witten et al., 2011] Witten, D., Friedman, J., and Simon, N. (2011). New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900.

[Witten and Tibshirani, 2009] Witten, D. and Tibshirani, R. (2009). Covariance-regularized regression and classification for high dimensional problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):615–636.

[Woltman et al., 2012] Woltman, H., Feldstain, A., MacKay, J. C., and Rocchi, M. (2012). An introduction to hierarchical linear modeling. *Tutorials in Quantitative Methods for Psychology*, 8(1):52–69.

[Xu et al., 2013] Xu, H., Eis, D., and Ramadge, P. (2013). The generalized lasso is reducible to a subspace constrained lasso. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3268–3272. IEEE.

[Yengo et al., 2012] Yengo, L., Jacques, J., Biernacki, C., et al. (2012). Variable clustering in high dimensional linear regression models.

[Yuan et al., 2007] Yuan, M., Ekici, A., Lu, Z., and Monteiro, R. (2007). Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):329–346.

[Zellner, 1962] Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American statistical Association*, pages 348–368.

[Zhang and Shen, 2010] Zhang, Y. and Shen, X. (2010). Model selection procedure for high-dimensional data. *Statistical Analysis and Data Mining*, 3(5):350–358.

[Zhao and Yu, 2006] Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563.

[Zhou and Lange, 2013] Zhou, H. and Lange, K. (2013). A path algorithm for constrained estimation. *Journal of Computational and Graphical Statistics*, 22(2):261–283.

[Zou, 2006] Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

[Zou and Hastie, 2005] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.