

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

Clément THÉRY

October 14, 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 la suite, 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.

Enfin, le modèle génératif complet offre la perspective de pouvoir 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 pourrait être appliquée à d'autres contextes tels que le clustering.

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

Le package R intitulé **CorReg**, disponible sur le CRAN¹ sous licence CeCILL², 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-regressions, that simply is a system of linear regressions between the covariates. It points out redundant covariates that can be deleted in a pre-selection step to improve matrix conditioning without significant loss of information and with strong explicative potential because this pre-selection is explained by the structure of sub-regressions, itself easy to interpret.

An algorithm to find the sub-regressions structure inherent to the dataset is provided, based on a full generative model and using Monte-Carlo Markov Chain (MCMC) method. This pretreatment is then applied 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.

In a second part, a plug-in estimator is defined to get back the redundant covariates sequentially. Then all the covariates are used but the sequential approach acts as a protection against correlations.

Finally, the generative model defined here allows, as a perspective, 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.

The industrial motivation of this work defines interpretation as a stronghold at each step. The R package `CorReg`, is on CRAN³ now under CeCILL⁴ 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 led me to be recruited since May.

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. 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 M Θ dal 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 helped me in this last and tough year just by their presence, giving me the courage to go further.

I finally want to thank my family. My wife who had to support most of the charge of the family on 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	9
1.1 Position du problème	9
1.2 Modélisation explicite des corrélations	9
1.3 Modèle marginal	11
1.4 Notion de prétraitement	12
1.5 Estimation de la structure	12
1.6 Relaxation des contraintes et nouveau critère	13
1.7 Résultats	13
1.8 Modèle plug-in pour réduire l'erreur	13
1.9 Valeurs manquantes	14
2 The industrial context	15
2.1 Steelmaking process	15
2.2 Impact of the industrial context	16
3 State of the art in linear regression	17
3.1 Linear regression	17
3.1.1 Industrial context	17
3.1.2 Ordinary least squares and associated problems	18
3.1.3 Ridge regression: a penalized estimator	20
3.2 Variable selection method	21
3.2.1 Least Absolute Shrinkage and Selection Operator (LASSO)	21
3.2.2 Least Angle Regression	22
3.2.3 Elasticnet	24
3.2.4 OSCAR: Octagonal Shrinkage and Clustering Algorithm for Regression .	25
3.2.5 Stepwise	26
3.2.6 Industrial tools	27
3.3 Modeling the parameters	27
3.3.1 CLusterwise Effect REgression (CLERE)	27
3.3.2 Spike and Slab	28
3.4 Non-linear regression methods	29
3.4.1 Sliced Inverse Regression	29
3.4.2 Neural networks	29
3.4.3 Bayesian networks	29
3.4.4 Classification and Regression Trees (CART)	30
3.5 Taking correlations into account	31
3.5.1 Principal Component Regression (PCR)	31
3.5.2 Partial Least Squares Regression (PLS)	32
3.5.3 Simultaneous Equation Model (SEM) and Path Analysis	32
3.5.4 SUR: Seemingly Unrelated Regression	33

3.5.5 Selvarclust: Linear regression within covariates for clustering	33
I Pretreatment for correlations	35
4 Decorrelating covariates by a generative model	36
4.1 Introduction	36
4.2 Explicit modeling of the correlations	37
4.3 A by-product model: marginal regression with decorrelated covariates	38
4.4 Strategy of use: pre-treatment before classical estimation/selection methods . .	39
4.5 Illustration of the tradeoff conveyed by the pre-treatment	40
4.6 MSE comparison on the running example	41
4.7 Connexion with graphs	47
5 Numerical results with a known structure on more complex datasets	49
5.1 The datasets	49
5.2 Results	49
5.2.1 OLS when \mathbf{Y} depends on all variables in \mathbf{X} , true \mathbf{S} known	51
5.2.2 LASSO when \mathbf{Y} depends on all variables in \mathbf{X} , true \mathbf{S} known	52
5.2.3 Elasticnet when \mathbf{Y} depends on all variables in \mathbf{X} , true \mathbf{S} known	53
5.2.4 Stepwise when \mathbf{Y} depends on all variables in \mathbf{X} , true \mathbf{S} known	54
5.2.5 Ridge when \mathbf{Y} depends on all variables in \mathbf{X} , true \mathbf{S} known	55
6 Estimation of the Structure of subregression by MCMC	56
6.1 Choice of model	56
6.1.1 Cross validation	56
6.1.2 Bayesian Information Criterion	56
6.2 Bayesian approach	57
6.3 Sub-regression structure in details	57
6.3.1 Modeling the uncorrelated covariates: a full generative approach on $P(\mathbf{X})$	57
6.3.2 Usage of the structure \mathbf{S} for interpretation	58
6.4 Sub-regression model selection	59
6.4.1 Bayesian criterion for quality	59
6.4.2 Penalization of the integrated likelihood by $\mathbb{P}(\mathbf{S})$	59
6.4.3 Some indicators for proximity	60
6.5 The walk	60
6.5.1 Determinist neighbourhood	60
6.5.2 Transition probabilities	61
6.5.3 Stochastic neighbourhood	61
6.5.4 Active relaxation of constraints	62
6.6 Initialization	65
6.6.1 Correlation-based initialization	65
6.6.2 Multiple intialization	66
6.7 Pruning	67
6.8 Graphical LASSO	67
6.9 CorReg	68
7 Numerical results on simulated datasets	70
7.1 Simulated datasets	70
7.2 Results on $\hat{\mathbf{S}}$	70
7.2.1 Comparison with Selvarclust	71

7.2.2	Computational time	72
7.3	Results on prediction	75
7.3.1	\mathbf{Y} depends on all variables in \mathbf{X}	75
7.3.2	\mathbf{Y} depends only on covariates in \mathbf{X}_f (best case for us)	81
7.3.3	\mathbf{Y} depends only on covariates in \mathbf{X}_r (worst case for us)	87
7.3.4	Robustness with non-linear case	93
8	Numerical results on real datasets	94
8.1	Quality case study	94
8.2	Production case study	96
II	Extensions	98
9	Using coefficients of regression and sub-regression to improve prediction	99
9.1	Introduction	99
9.2	Plug-in model	99
9.3	Interpretation and latent variables	101
9.4	Consistency of the LASSO	101
9.5	Numerical results	102
9.5.1	\mathbf{Y} depends on all variables in \mathbf{X}	102
9.5.2	\mathbf{Y} depends only on covariates in \mathbf{X}_f	109
9.5.3	\mathbf{Y} depends only on covariates in \mathbf{X}_r	109
10	Missing values	120
10.1	State of the art	120
10.2	Estimation of \mathbf{S} with missing values	121
10.2.1	Marginal likelihood	121
10.2.2	Likelihood computation optimized	125
10.2.3	Weighted penalty for BIC	125
10.3	EM Algorithm to estimate $\boldsymbol{\alpha}$ with a given \mathbf{S}	125
10.3.1	Stochastic EM	126
10.3.2	Stochastic imputation by Gibbs sampling	127
10.4	Missing values in the main regression	128
10.5	Numerical results on simulated datasets	128
10.5.1	Estimation of the sub-regression coefficients	128
10.5.2	Multiple imputation	129
10.5.3	Results on the main regression	130
10.6	Missing values on real datasets	131
11	Conclusion and perspectives	133
11.1	Conclusion	133
11.2	Perspectives	133
11.2.1	Non-linear regression	133
11.2.2	Pretreatment not only for regression	133
11.2.3	Improved programming	134
11.2.4	Missing values in classical methods	134
11.2.5	Interpretation improvements	134
References		135

III Appendices	141
12 Identifiability	142
12.1 Definition	142
12.2 Sufficient condition for identifiability	142
13 CorReg: the concept	144

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

$$\mathbf{Y}|\mathbf{X}; \boldsymbol{\beta}, \sigma_Y^2 = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}_Y \quad (1.1)$$

où \mathbf{Y} est la variable réponse de taille $n \times 1$ que l'on souhaite décrire à l'aide de d variables explicatives¹ observées sur n individus et dont les valeurs sont stockées dans la matrice \mathbf{X} de taille $n \times d$. Le vecteur $\boldsymbol{\beta}$ est le vecteur des coefficients de régression qui permet de décrire le lien linéaire entre \mathbf{Y} et \mathbf{X} . Le vecteur $\boldsymbol{\varepsilon}_Y$ est un bruit blanc gaussien $\mathcal{N}(0, \sigma_Y^2 \mathbf{I}_n)$ 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 :

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

Cet estimateur nécessite l'inversion de la matrice $(\mathbf{X}'\mathbf{X})$ 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 :

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

C'est cette situation problématique 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 \mathbf{X} sont presque linéairement dépendantes. Cette quasi dépendance linéaire peut être elle aussi modélisée par régression linéaire. 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 d variables en 2 blocs : les variables réponses (expliquées) et les variables prédictives. Notre modèle repose sur 2 hypothèses fondamentales :

¹En général on ajoute une constante parmi les régresseurs. Par exemple $\mathbf{X}^1 = (1, \dots, 1)'$. Le coefficient associé dans $\boldsymbol{\beta}$ est alors la constante de régression β_1 .

Hypothèse 1 Les corrélations entre covariables viennent de ce que certaines d'entre elles dépendent linéairement d'autres covariables. Plus précisément, il y a $d_r \geq 0$ “sous-regressions”, chaque sous-regression $j = 1, \dots, d_r$ ayant la variable $\mathbf{X}^{J_r^j}$ comme variable réponse ($J_r^j \in \{1, \dots, p\}$ et $J_r^j \neq J_r^{j'}$ pour $j \neq j'$) et ayant les $d_p^j > 0$ variables $\mathbf{X}^{J_p^j}$ comme variables prédictives ($J_p^j \subset \{1, \dots, d\} \setminus J_r^j$ et $d_p^j = |J_p^j|$ le cardinal de J_p^j):

$$\mathbf{X}^{J_r^j} | \mathbf{X}^{J_p^j}; \boldsymbol{\alpha}_j, \sigma_j^2 = \mathbf{X}^{J_p^j} \boldsymbol{\alpha}_j + \boldsymbol{\varepsilon}_j, \quad (1.4)$$

où $\boldsymbol{\alpha}_j \in \mathbb{R}^{d_r^j}$ ($\alpha_j^h \neq 0$ pour tout $j = 1, \dots, d_r$ et $h = 1, \dots, d_p^j$) et $\boldsymbol{\varepsilon}_j \sim \mathcal{N}_n(\mathbf{0}, \sigma_j^2 \mathbf{I})$.

Hypothèse 2 Nous supposons également que les variables réponses et les variables prédictives forment deux blocs disjoints dans \mathbf{X} : pour toute sous-régression $j = 1, \dots, d_r$, $J_p^j \subset J_f$ où $J_r = \{J_r^1, \dots, J_r^{d_r}\}$ est l'ensemble de toutes les variables réponses on a $J_f = \{1, \dots, d\} \setminus J_r$ l'ensemble des variables non expliquées de cardinal $d_f = d - d_r = |J_f|$. Cette seconde hypothèse garantit l'obtention d'un système de sous-régression très simple et sans imbriques ni surtout aucun cycle. Cette hypothèse n'est pas trop restrictive dans la mesure où tout système sans cycle peut (par substitutions) être reformulé sous cette forme simplifiée (avec une variance accrue).

Notations Par la suite nous noterons $\mathbf{J}_r = (J_r^1, \dots, J_r^{d_r})$ le d_r -uplet des variables réponses (à ne pas confondre avec J_r défini plus haut), $\mathbf{J}_p = (J_p^1, \dots, J_p^{d_r})$ le d_r -uplet des prédicteurs de toutes les sous-régressions, $\mathbf{d}_p = (d_p^1, \dots, d_p^{d_r})$ les nombres correspondants de prédicteurs et $\mathbf{S} = (\mathbf{J}_r, \mathbf{J}_p)$ le *model* global (structure de sous-régressions). Pour alléger les notations, on définit alors $\mathbf{X}_r = \mathbf{X}^{J_r}$ la matrice des variables réponses et $\mathbf{X}_f = \mathbf{X}^{J_f}$ la matrice de toutes les autres variables, ainsi considérées comme libres (*free* en anglais). Les valeurs des paramètres sont également concaténées: $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_{d_r})$ est la matrice des coefficients de sous-régression et $\boldsymbol{\sigma}^2 = (\sigma_1^2, \dots, \sigma_{d_r}^2)$ le vecteur des variances associées. Toutes ces notations sont illustrées dans l'exemple ci-après.

$$\mathbf{X}_r | \mathbf{X}_f; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2 = \mathbf{X}_f \boldsymbol{\alpha} + \boldsymbol{\varepsilon} \quad (\text{regression multiple multivariée}) \quad (1.5)$$

La matrice $\boldsymbol{\varepsilon} \in \mathbb{R}^{n \times d_r}$ est la matrice des bruits des sous-régressions composée des colonnes $\boldsymbol{\varepsilon}_j \sim \mathcal{N}(0, \sigma_j^2 \mathbf{I}_n)$ que nous supposons indépendantes entre elles.

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 $d = 5$ variables dont 4 gaussiennes centrées réduites *i.i.d.* $\mathbf{X}_f = (\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^4, \mathbf{X}^5)$ et une variable redondante $\mathbf{X}_r = \mathbf{X}^3 = \mathbf{X}^1 + \mathbf{X}^2 + \boldsymbol{\varepsilon}_1$ avec $\boldsymbol{\varepsilon}_1 \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I}_n)$. Deux régressions principales en \mathbf{Y} ont été testées avec $\boldsymbol{\beta} = (1, 1, 1, 1, 1)'$ and $\sigma_Y \in \{10, 20\}$. Le conditionnement de $(\mathbf{X}' \mathbf{X})$ se déteriorise donc quand σ_1 diminue. Ici $J_f = \{1, 2, 4, 5\}$, $J_r = \{3\}$, $\mathbf{J}_r = (3)$, $\mathbf{d}_p = 2$, $\mathbf{J}_p = (\{1, 2\})$, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1) = ((1, 1)')$, $\mathbf{X}^{J_p^1} = (\mathbf{X}^1, \mathbf{X}^2)$, $\mathbf{S} = ((3), (\{1, 2\}))$.

Remarques

- Les sous-régressions définies en (1.4) sont très simples à comprendre pour l'utilisateur et permettent donc d'avoir un aperçu net des corrélations présentes dans les données étudiées.
- Les hypothèses 1 et 2 , rendent les variables dans \mathbf{X}_f indépendantes.
- \mathbf{S} ne dépend pas de \mathbf{Y} et peut donc être estimé séparément.

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 \mathbf{X}_r sachant la structure de sous-régressions \mathbf{S} . On fait alors un hypothèse supplémentaire :

Hypothèse 3 *On suppose l'indépendance 2 à 2 entre les erreurs de régression $\boldsymbol{\varepsilon}_Y$ et $\boldsymbol{\varepsilon}_j (j = 1, \dots, d_r)$. En particulier on a l'indépendance conditionnelle entre les variables réponses $\{\mathbf{X}^{J_r^j} | \mathbf{X}^{J_p^j}, \mathbf{S}; \boldsymbol{\alpha}_j, \sigma_j^2\}$ définies en (1.4).*

On a donc

$$\mathbb{P}(\mathbf{X}_r | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2) = \prod_{j=1}^{d_r} \mathbb{P}(\mathbf{X}^{J_r^j} | \mathbf{X}^{J_p^j}, \mathbf{S}; \boldsymbol{\alpha}_j, \sigma_j^2) \quad (1.6)$$

$$\mathbb{P}(\mathbf{Y} | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2) = \int_{\mathbb{R}^{d_r}} \mathbb{P}(\mathbf{Y} | \mathbf{X}_r, \mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \sigma_Y^2) \mathbb{P}(\mathbf{X}_r | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2) d\mathbf{X}_r \quad (1.7)$$

$$= \quad (1.8)$$

Ce qui donne

$$\mathbf{Y} | \mathbf{X}, \mathbf{S}; \boldsymbol{\beta}, \sigma_Y^2 = \mathbf{X}_f \boldsymbol{\beta}_f + \mathbf{X}_r \boldsymbol{\beta}_r + \boldsymbol{\varepsilon}_Y \quad (1.9)$$

$$\mathbf{Y} | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2 = \mathbf{X}_f (\boldsymbol{\beta}_f + \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\alpha}_j^*) + \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_Y \quad (1.10)$$

$$= \mathbf{X}_f \boldsymbol{\beta}_f^* + \boldsymbol{\varepsilon}_Y^* \quad (1.11)$$

où $\boldsymbol{\alpha}^* \in \mathbb{R}^{d_f \times d_r}$ est la matrice des coefficients des sous-régressions avec $(\boldsymbol{\alpha}_j^*)_{J_p^j} = \boldsymbol{\alpha}_j$ et $(\boldsymbol{\alpha}_j^*)_{J_f \setminus J_p^j} = \mathbf{0}$, $\boldsymbol{\beta}_r = \boldsymbol{\beta}_{J_r}$, $\boldsymbol{\beta}_f = \boldsymbol{\beta}_{J_f}$, $\boldsymbol{\beta}_f^* = \boldsymbol{\beta}_{J_f}^*$, $\boldsymbol{\beta}^*$ étant l'estimateur de $\boldsymbol{\beta}$ par le modèle marginal avec donc $\boldsymbol{\beta}_r^* = \boldsymbol{\beta}_{J_r}^*$.

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 et plus cette variance est faible. Tout le principe du modèle marginal repose sur le compromis entre l'amélioration du conditionnement de $(\mathbf{X}' \mathbf{X})$ par suppression des variables redondantes et aussi la réduction de la dimension (le modèle marginal ne nécessite que l'inversion de $(\mathbf{X}'_f \mathbf{X}_f)$) face au léger accroissement de la variance issu de la marginalisation. On va donc comparer

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{Y} \text{ au modèle marginal} \quad (1.12)$$

$$\hat{\boldsymbol{\beta}}_f^* = (\mathbf{X}'_f \mathbf{X}_f)^{-1} \mathbf{X}'_f \mathbf{Y} \quad (1.13)$$

Les deux modèles sont de dimension différente, on compare donc leurs erreurs moyennes quadratiques respectives (MSE en anglais pour *Mean Squared Error*):

$$\text{MSE}(\hat{\boldsymbol{\beta}} | \mathbf{X}) = \sigma_Y^2 \text{Tr}(\mathbf{X}' \mathbf{X})^{-1} \quad (1.14)$$

$$\text{MSE}(\hat{\boldsymbol{\beta}}^* | \mathbf{X}) = \left\| \sum_{j=1}^{d_r} \sigma_j \beta_{J_r^j} \right\|_2^2 + \left\| \boldsymbol{\beta}_r \right\|_2^2 + (\sigma_Y^2 + \sum_{j=1}^{d_r} \sigma_j^2 \beta_{J_r^j}^2) \text{Tr}(\mathbf{X}'_f \mathbf{X}_f)^{-1} \quad (1.15)$$

Ces deux équations illustrent bien le compromis biais-variance du modèle marginal. La figure 4.1 compare les erreurs d'estimation obtenues 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 variables 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 \mathbf{Y} donc le modèle final est parsimonieux mais ne fausse pas l'interprétation. L'estimation de β^* peut ensuite se faire en utilisant une quelconque méthode de sélection de variables pour éliminer les variables qui, elles, sont indépendantes de \mathbf{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 pointent 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-régressions 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-régressions. L'idée consiste à voir la structure de sous-régression comme un paramètre binaire, une matrice binaire creuse pour être plus précis. Cette matrice \mathbf{G} de taille $p \times p$ correspond à une matrice d'adjacence qui indique les liaisons entre covariables de la manière suivante : $\mathbf{G}_{i,j} = 1$ si, et seulement si \mathbf{X}^j est expliqué par \mathbf{X}^i .

Chaque étape $(q + 1)$ de l'algorithme propose de garder la structure $\mathbf{G}^{(q)}$ en cours ou bien de bouger vers une structure candidate qui diffère de $\mathbf{G}^{(q)}$ en un unique point. Ainsi, selon les cas, les candidats vont allonger ou réduire des sous-régressions, 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 \mathbf{X} qui complète le modèle de sous-régressions en établissant des modèles de mélanges gaussiens indépendants pour les variables de \mathbf{X}_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 (hypothèse 2). 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 relaxation 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) de la matrice \mathbf{G} :

$$\mathbf{G}_{i,j}^{(q+1)} = 1 - \mathbf{G}_{i,j}^{(q)} \quad (1.16)$$

$$\mathbf{G}_{.,i}^{(q+1)} = \mathbf{G}_{i,j}^{(q)} \mathbf{G}_{.,i}^{(q)} \text{ Si } i \text{ explique } j \text{ alors personne ne peut plus expliquer } i \quad (1.17)$$

$$\mathbf{G}_{j,.}^{(q+1)} = \mathbf{G}_{i,j}^{(q)} \mathbf{G}_{j,.}^{(q)} \text{ Si } i \text{ explique } j \text{ alors } j \text{ ne peut plus expliquer personne} \quad (1.18)$$

Cette méthode de relaxation permet de sortir rapidement des extrema locaux et améliore donc significativement l'efficacité de l'algorithme (Figure 6.10). La méthode est illustrée sur un exemple par les figures 6.1 à 6.6.

Mais il reste un problème. Le nombre de modèles envisageables 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 \mathbf{S} dans une base de données \mathbf{X} , BIC est utilisé comme approximation pour $\mathbb{P}(\mathbf{S}|\mathbf{X}) \propto \mathbb{P}(\mathbf{X}|\mathbf{S})\mathbb{P}(\mathbf{S})$ car $\mathbb{P}(\mathbf{S})$ est considéré comme suivant une loi uniforme. On s'appuie donc sur une loi uniforme hiérarchique $\mathbb{P}_H(\mathbf{S}) = \mathbb{P}_U(\mathbf{J}_p|\mathbf{d}_p, \mathbf{J}_r, d_r)\mathbb{P}_U(\mathbf{d}_p|\mathbf{J}_r, d_r)\mathbb{P}_U(\mathbf{J}_r|d_r)\mathbb{P}_U(d_r)$ pour ajouter une pénalité supplémentaire aux structures complexes (même probabilité globale pour un plus grand nombre de structures donc chaque structure devient moins probable). On note BIC_H ce nouveau critère. 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 à optimiser par ailleurs. Tout reste naturel et intuitif pour une meilleure 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 4.6), 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 Modèle plug-in pour réduire l'erreur

Une première perspective a été développée pour compléter le modèle marginal. Il s'agit de mettre à profit la formulation exacte du modèle marginal pour estimer β_r et ainsi réduire le bruit de la régression marginale puis d'utiliser ce nouvel estimateur pour identifier β_f et ainsi obtenir un nouveau modèle complet s'appuyant sur \mathbf{X} entier mais protégé des corrélations par l'estimation séquentielle.

Après estimation du modèle marginal on s'applique à améliorer l'estimation de β :

- Estimation de ε_Y^* à partir de $\hat{\beta}_f^*$:

$$\hat{\varepsilon}_Y^* = \mathbf{Y} - \mathbf{X}_f \hat{\beta}_f^*. \quad (1.19)$$

- Estimation de ε à partir de $\hat{\alpha}^*$:

$$\hat{\varepsilon} = \mathbf{X}_r - \mathbf{X}_f \hat{\alpha}^*. \quad (1.20)$$

Ces deux nouveaux estimateurs permettent alors d'obtenir un estimateur de β_r meilleur que le marginal $\hat{\beta}_r^* = \mathbf{0}$.

- Amélioration de l'estimation de β_r :

$$\hat{\beta}_r^\varepsilon = (\hat{\varepsilon}' \hat{\varepsilon})^{-1} \hat{\varepsilon}' (\mathbf{Y} - \mathbf{X}_f \hat{\beta}_f^*) \quad (1.21)$$

Cet estimateur est consistant et nous permet de réduire le bruit du modèle marginal, donc d'estimer \mathbf{Y} avec une variance réduite.

$$\hat{\mathbf{Y}}_{\text{plug-in}} = \mathbf{X}_f \hat{\beta}_f^* + \hat{\varepsilon} \hat{\beta}_r^\varepsilon. \quad (1.22)$$

On peut ensuite dans une nouvelle phase d'estimation se rapprocher de β_f .
On a $\beta_f^* = \beta_f + \alpha^* \beta_r$.

- Amélioration de l'estimation de β_f :

$$\hat{\beta}_f^\varepsilon = \hat{\beta}_f^* - \hat{\alpha}^* \hat{\beta}_r^\varepsilon. \quad (1.23)$$

La figure 9.1 montre l'efficacité du modèle plug-in et son champ d'application : les cas avec assez de corrélations pour que les méthodes classiques appliquées à \mathbf{X} soient handicapées mais pas assez de corrélations pour que le retrait des variables redondantes (modèle marginal) se fasse sans perte significative d'information.

1.9 Valeurs manquantes

Une seconde perspective a été entamée concernant les valeurs manquantes. Le fait de disposer d'un modèle génératif complet sur \mathbf{X} avec modélisation explicite des dépendances permet en effet de composer avec les valeurs manquantes. Tout d'abord, l'estimation de α 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 $\hat{\alpha}$.

En pratique, l'étape E n'est pas systématiquement explicite et peut nécessiter de faire appel à une variante de EM : l'algorithme 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 $\hat{\alpha}$ issu du Stochastic EM. 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. Encore une fois, on y gagne en qualité d'interprétation.

Chapter 2

The industrial context

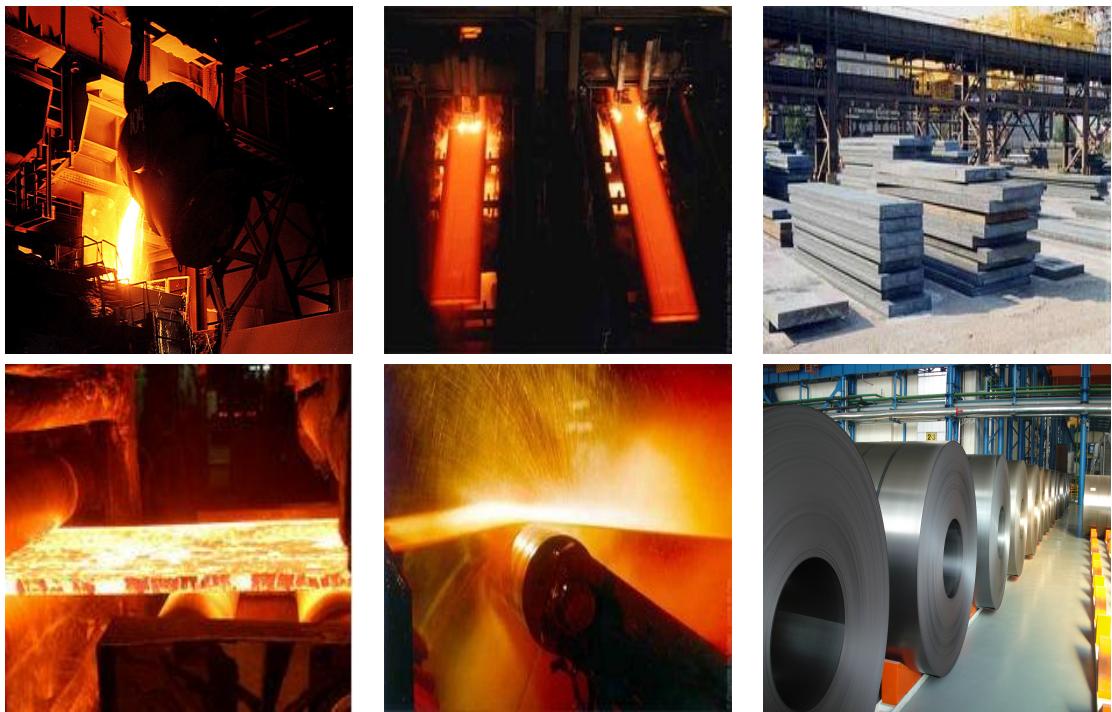
Abstract: This chapter describes the industrial constraints that has led this work. This work takes place in a steel industry context and was funded by ArcelorMittal, the world leading company in steelmaking.

2.1 Steelmaking process

Steelmaking starts from raw materials and aims to give highly specific products used in automotive, beverage cans, *etc.* We first melt down a mix of iron ore and coke to obtain cast iron that is then transformed in steel by addition of pure oxygen to remove carbon. Liquid steel is then refreshed in a mold (continuous casting) to obtain steel slabs (nearly 20 tons each, 23 cm thick).

Cold slabs are then warmed to be pressed in a hot rolling mill to obtain coils (nearly half a millimeter thick or less). This warming phase also allows to adjust mechanical properties of the final product. It is the process that gave its name to the simulated annealing algorithm that allows to escape from local extrema thanks to a parameter called "temperature" in allusion to the steelmaking process.

If the final product requires a thinner strip of 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 ones and sometimes defects are generated by small inclusions 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 continuously improving and we are now able to produce steel that is both thinner and stronger. Steel is 100% recyclable unlike petroleum so we will continue to use it widely in the future. 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. Defects, even generated at the beginning of the process, are often detected in its last part. The steel-making process does include several sub-process. Thus we have many covariates and no a priori on the relevant ones. Moreover, the values of each covariates are linked to 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 :

- Being able to predict the defect and stop the process as early as possible to gain time (and money)
- Being able to find parameters that can be changed and to understand the origin of the defect because the objective is not only to understand but to adapt the problematic part of the process.
- It also must be fast and automatic (without any a priori) to manage totally new phenomenon on new products.

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 and the extreme competition between steelmakers but also because steel making is now well-known and optimized thus new defects only appears on innovative steels with high added value. Any improvement on such crisis can have important impact on market shares and when the customer is impacted, each day won by the automation of the data mining process can lead to substantial savings. So we really need a kind of automatic method, able to manage correlations without any a priori and giving an easily understandable and flexible model.

Chapter 3

State of the art in linear regression

Abstract: Brief state of the art to have a glimpse of existing method we could try to solve our problematic. Most of the tools described here are explained with more details in the book from Hastie, Tibshirani et Friedman : *"The Elements of Statistical Learning: Data Mining, Inference, and Prediction"* accessible on web¹ for free.

3.1 Linear regression

3.1.1 Industrial context

Industrial context necessitates easily understandable models 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 interpretable (no black-box). 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, and is used in nearly all the fields where statistics are made[Montgomery et al., 2012]: Astronomy [Isobe et al., 1990], Sociology [Longford, 2012], and so on. It can be done directly in Microsoft Excel which is well known and often used by engineers to open and analyse most of their datasets. Thus we have chosen to work in 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 good tool for interpretation with the sign of the coefficients indicating whether the associated covariate has a positive or negative impact 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

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



Figure 3.1: An example of simple linear regression

not linear in \mathbf{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: $\|\boldsymbol{\beta}\|_2^2 = \sum_{i=1}^d (\beta_i)^2$, $\|\boldsymbol{\beta}\|_1 = \sum_{i=1}^d |\beta_i|$ and $\|\boldsymbol{\beta}\|_\infty = \max(|\beta_1|, \dots, |\beta_d|)$. Vectors, matrices and tuples are in bold characters.

3.1.2 Ordinary least squares and associated problems

We note the linear regression model:

$$\mathbf{Y} | \mathbf{X}; \boldsymbol{\beta}, \sigma_Y^2 = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}_Y \quad (3.1)$$

where \mathbf{X} is the $n \times d$ matrix of the explicative variables², \mathbf{Y} the $n \times 1$ response vector and $\boldsymbol{\varepsilon}_Y \sim \mathcal{N}(\mathbf{0}, \sigma_Y^2 \mathbf{I}_n)$ the noise of the regression, with \mathbf{I}_n the n -sized identity matrix and $\sigma_Y > 0$. The $d \times 1$ 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):

$$\hat{\boldsymbol{\beta}}_{OLS} = (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{Y} \quad (3.2)$$

with variance matrix

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

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

$$\text{MSE}(\hat{\boldsymbol{\beta}}_{OLS}) = 0 + \sigma_Y^2 \text{Tr}((\mathbf{X}' \mathbf{X})^{-1}) \quad (3.4)$$

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

²Usually a constant is included as one of the regressors. For example we can take $\mathbf{X}^1 = (1, \dots, 1)'$. The corresponding element of $\boldsymbol{\beta}$ is then the intercept β_1 .

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

$$\hat{\boldsymbol{\beta}}_{OLS} = \operatorname{argmin}_{\boldsymbol{\beta}} \left\{ \| \mathbf{Y} - \mathbf{X}\boldsymbol{\beta} \|_2^2 \right\} \quad (3.5)$$

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

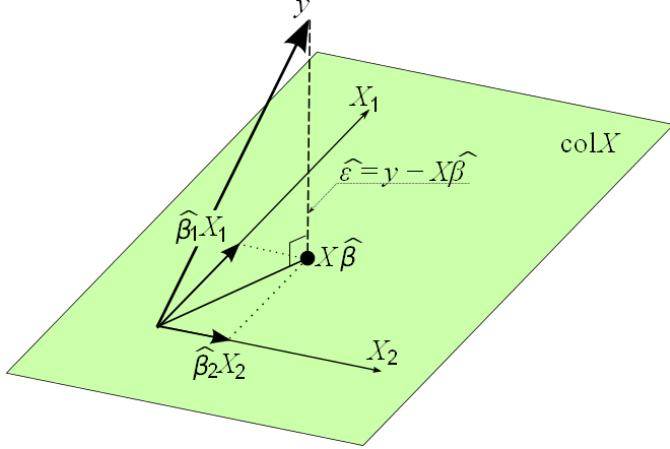


Figure 3.2: Multiple linear regression with Ordinary Least Squares. Public domain image.

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

- The dimension d (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 ($n \geq d$). 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 d . 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 $d = 5$ variables defined by four independent scaled Gaussian $\mathcal{N}(0, 1)$ named $\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^4, \mathbf{X}^5$ and $\mathbf{X}^3 = \mathbf{X}^1 + \mathbf{X}^2 + \boldsymbol{\epsilon}_1$ where $\boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I}_n)$. We also define two *scenarios* for \mathbf{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 $\mathbf{X}'\mathbf{X}$ will become more ill-conditioned as σ_1 gets smaller.

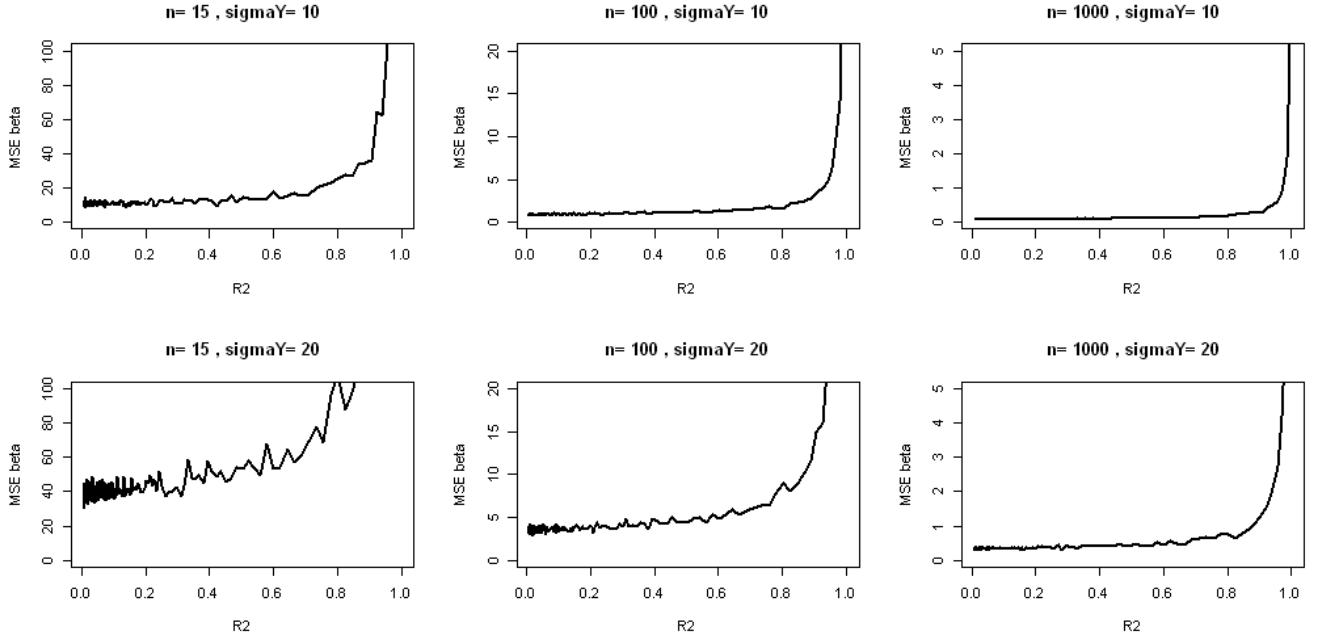


Figure 3.3: Evolution of observed Mean Squared error on $\hat{\beta}_{OLS}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates (running example).

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

Results shown in Figure 3.3 were obtained with usage of *QR* decomposition to inverse matrices, that is less impacted by ill-conditioned matrices [Bulirsch and Stoer, 2002] and used in the `lm` function from R to compute *OLS*. But the correlations issue remains. Our package `CorReg` also uses this decomposition. We show the mean obtained after 100 experiences computed on our running example with validation sample of 1 000 individuals. The MSE does explode with growing values of R^2 . The results confirm that the situation gets better for large values of n but strong correlations can still make the MSE exploding. The variance of $\hat{\beta}$ is proportional to σ_Y^2 so the MSE are bigger when the main regression is weak (as in the real life).

The R^2 stands for:

$$R^2 = 1 - \frac{\text{Var}(\boldsymbol{\varepsilon}_1)}{\text{Var}(\mathbf{X}^3)} \quad (3.6)$$

3.1.3 Ridge regression: a penalized estimator

We have seen that *OLS* is the Best linear Unbiased Estimator for $\hat{\beta}$, 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 $\hat{\beta}$.

Ridge regression [Hoerl and Kennard, 1970, Marquardt and Snee, 1975] proposes a possibly biased estimator for β that can be written in terms of a parametric L_2 penalty:

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

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

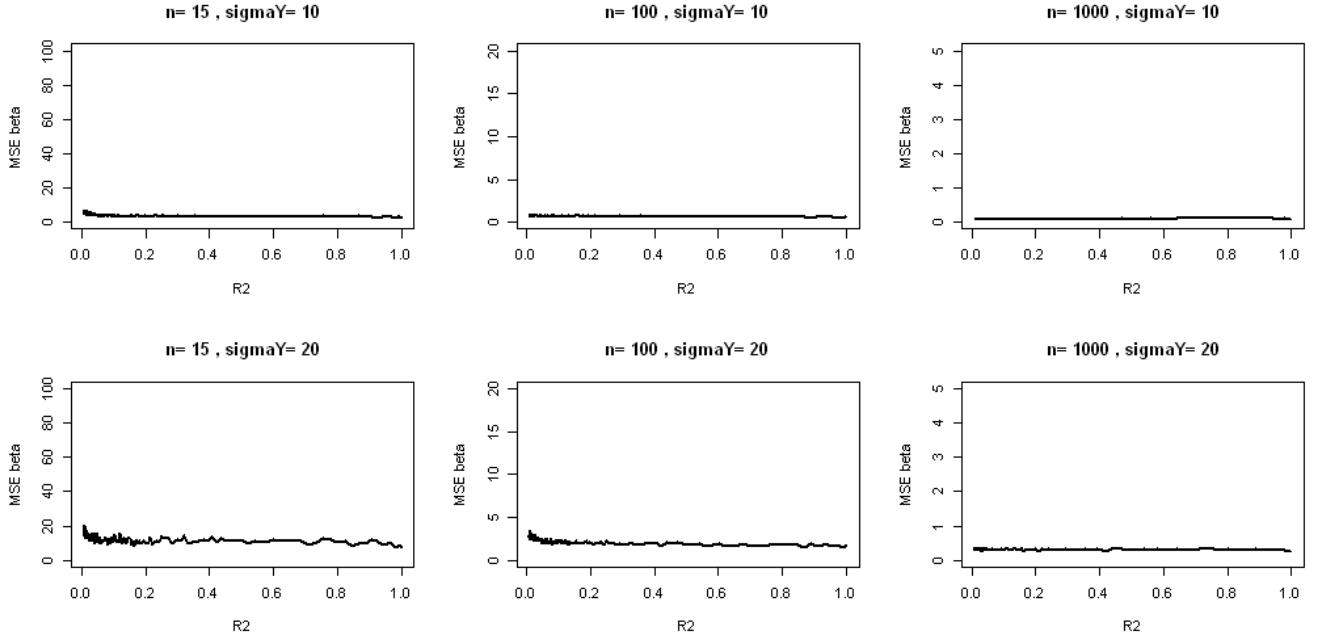


Figure 3.4: Evolution of observed Mean Squared error on $\hat{\beta}_{ridge}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

intensities.

The solution of the ridge regression is given by

$$\hat{\beta} = (\mathbf{X}'\mathbf{X} - \lambda \mathbf{I}_n)^{-1} \mathbf{X}'\mathbf{Y} \quad (3.8)$$

and we see in this equation that a global modification of $\mathbf{X}'\mathbf{X}$ is done for a given λ . Methods do 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, ridge allows to have $n < d$.

Like `OLS`, coefficients tend to 0 but don't reach 0 so it gives difficult interpretations for large values of d . 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. Our industrial context makes necessary to have a variable selection method so we look further.

3.2 Variable selection method

3.2.1 Least Absolute Shrinkage and Selection Operator (LASSO)

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 $\hat{\beta}$ instead of the L_2 penalty of the ridge regression:

$$\hat{\beta} = \operatorname{argmin} \left\{ \| \mathbf{Y} - \mathbf{X}\beta \|_2^2 \right\} \text{ subject to } \| \beta \|_1 \leq \lambda \text{ with } \lambda > 0 \quad (3.9)$$

Figure 3.5 shows 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

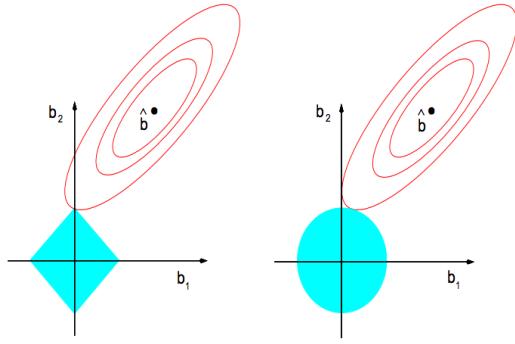


Figure 3.5: Geometric view of the Penalty for the LASSO (left) compared to ridge regression (right) as shown in the book from Hastie [Hastie et al., 2009]

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. It can be used through the `lars` package on CRAN ([Hastie and Efron, 2013]). 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 already selected covariate from an irrelevant covariate. And that is a problem. This consistency issue is illustrated in section 9.4.

Some recent variants of the LASSO do exist for the choice of the penalization coefficient like the adaptive LASSO [Zou, 2006] with the `parcor` package on CRAN ([Kraemer et al., 2009]) or the random LASSO [Wang et al., 2011]. But the consistency issue 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 covariates, thus the penalization is just a mean to achieve selection. But estimation of $\hat{\beta}$ can be improved by a second estimation with OLS based only on selected covariates [Zhang and Shen, 2010].

3.2.2 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 starting 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 3.7). 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 3.8 shows that strong correlations make the MSE explode even with LAR. Results

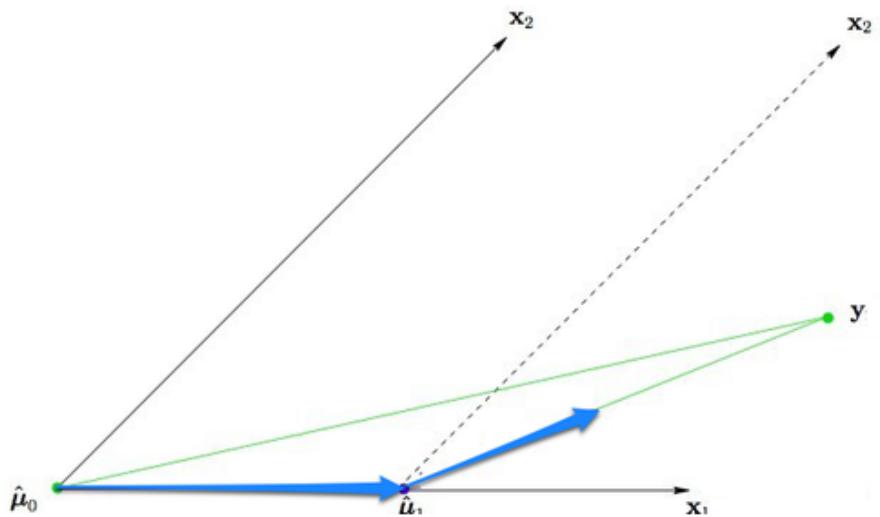


Figure 3.6: The geometry of Least Angle Regression

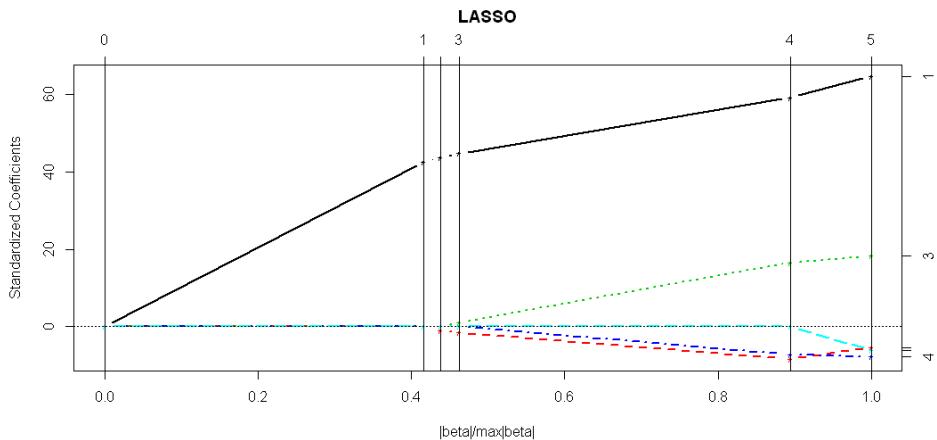


Figure 3.7: The LASSO path computed by lars

obtained with the package `lars` for R, included in `CorReg`.

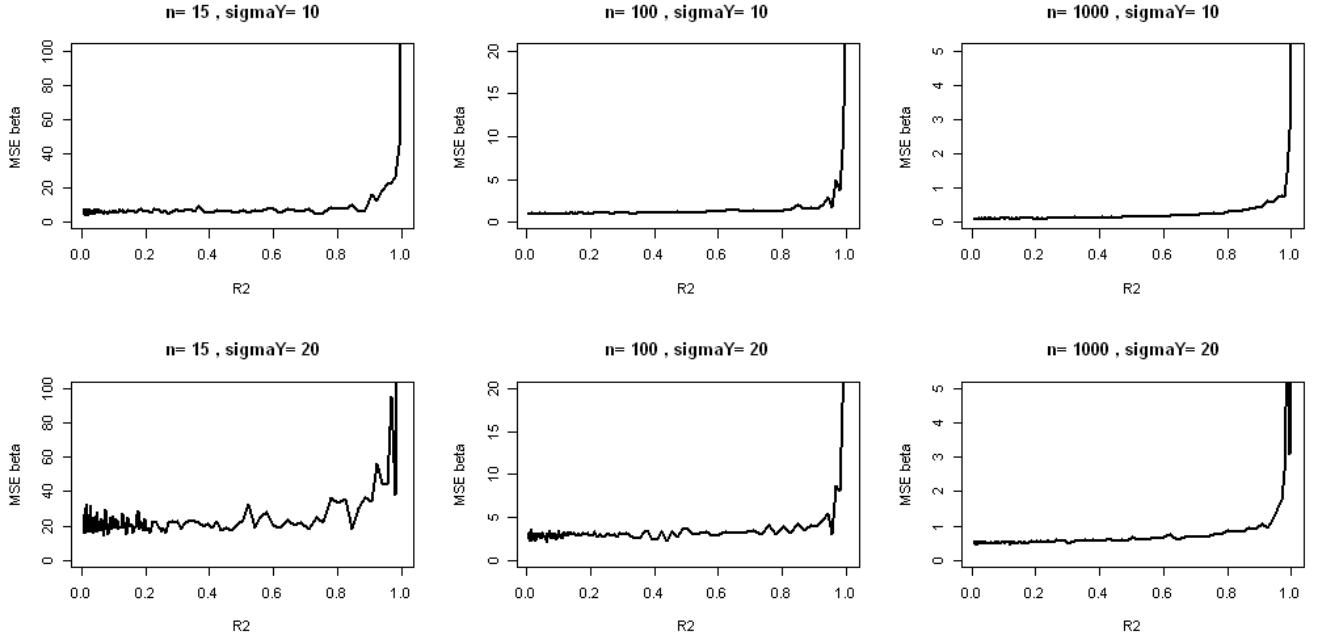


Figure 3.8: Evolution of observed Mean Squared error on $\hat{\beta}_{lar}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

3.2.3 Elasticnet

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

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

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

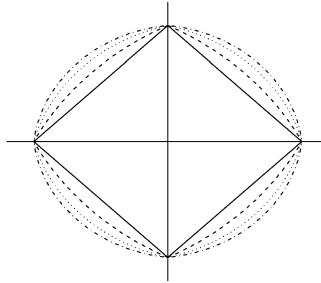


Figure 3.9: Geometric view of the Penalty for elasticnet

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` ([Zou and Hastie, 2012]) for R are given in Figure 3.10.

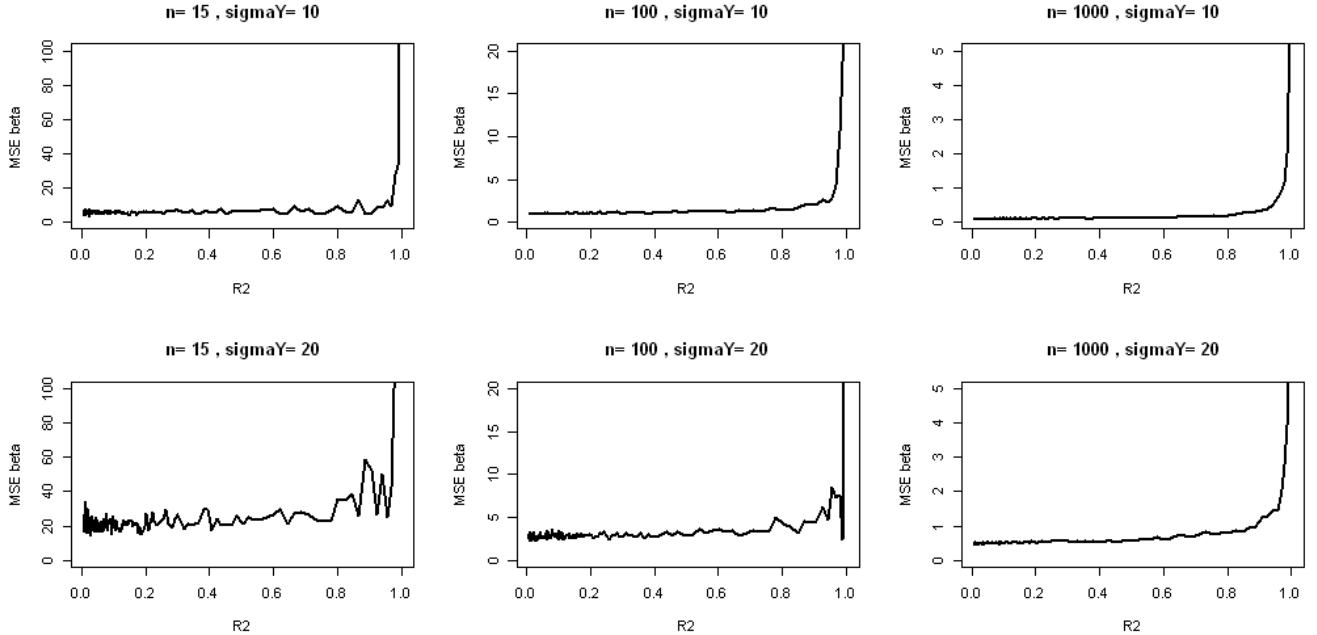


Figure 3.10: Evolution of observed Mean Squared error on $\hat{\beta}_{\text{elasticnet}}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

3.2.4 OSCAR: Octagonal 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). Thus correlations are avoided if correlated covariates are in the same cluster (not extremely flexible). A possible bias is added by the dimension reduction inherent to the coefficients clustering.

$$\hat{\beta} = \operatorname{argmin}_{\beta} \| \mathbf{Y} - \mathbf{X}\beta \|_2^2 \text{ subject to } \sum_{j=1}^d |\beta_j| + c \sum_{j < k} \max(|\beta_j|, |\beta_k|) \leq \lambda \quad (3.11)$$

Moreover OSCAR depends on two tuning parameters: c and 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 .

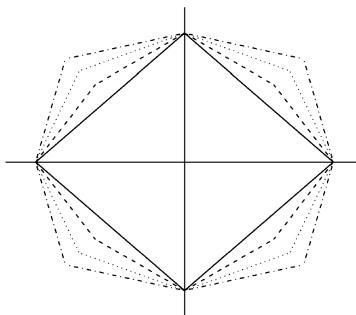


Figure 3.11: Geometric view of the Penalty for OSCAR

Figure 3.11 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 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.

3.2.5 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 a first model (that 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.

- 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. 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 removing 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 step proposes to add a covariate or to delete another so it is not a hierarchical construction any more because successive models are not necessarily nested into each other.

A critical value can be defined to stop the algorithm when improvement becomes too small, in order to avoid over-fitting.

Stepwise regression is subject to over-fitting and the algorithm is in trouble when confronted to correlated covariates [Miller, 2002] giving unstable results, especially for nested strategies, just like regression trees that are unstable because of their discrete nested nature. Figure 3.12 illustrate the consequences of correlations in the dataset.

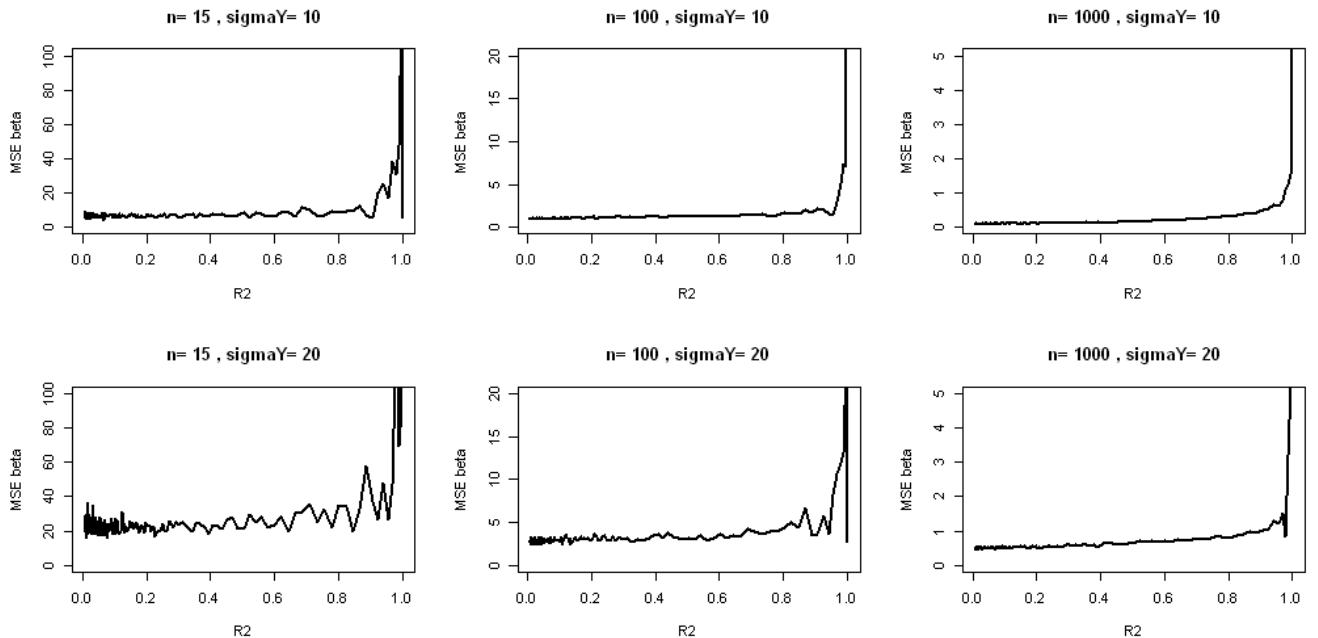


Figure 3.12: Evolution of observed Mean Squared error on $\hat{\beta}_{stepwise}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

3.2.6 Industrial tools

Our goal was also 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 3.13 shows the kind of results proposed by a rule-based software that is sold as a non-statistical tool. 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 $d > 10$ would take years) and is less efficient than regression trees. Moreover, 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.

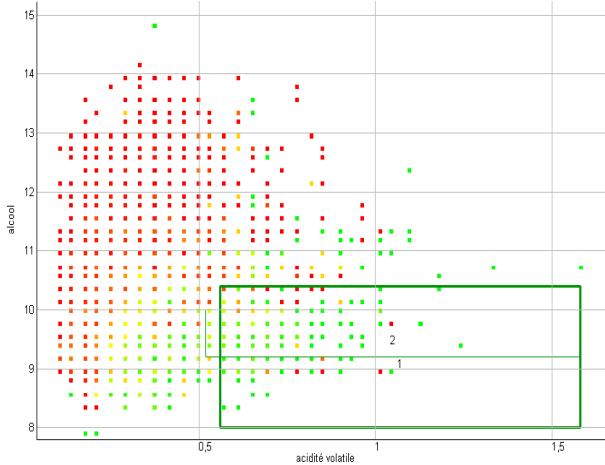
Ergonomy and quality of interpretation are stakes for us to make engineers use efficient methods instead of this kind of stuff.

3.3 Modeling the parameters

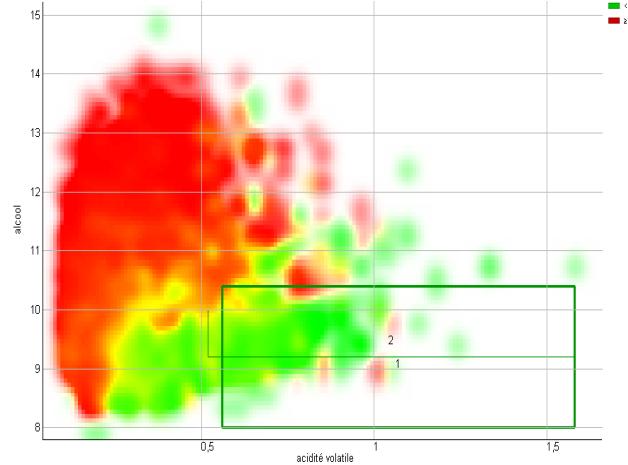
3.3.1 CLusterwise Effect REgression (clere)

The CLusterwise Effect REgression (CLERE [Yengo et al., 2012]) describes the β_j no longer as fixed effect parameters but as unobserved independent random variables with β_j following a Gaussian Mixture distribution, allowing to group them into by their component membership.

The idea is that if the model has a small number of groups of covariates then the mixture will have few enough components to have a number of parameters to estimate significantly lower than d . In such a case, it improves interpretation and ability to yield reliable prediction with a smaller variance on $\hat{\beta}$. A package `clere` for R does exist on CRAN ([Yengo and Canouil, 2014]).



(a) Without blur



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

Figure 3.13: Result on a toy example provided by FLASHPROCESS, similar to decision trees but less efficient and extremely slower. Colors are part of the learning set.

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 $g = d$ gives over-fitting. Here again, it has no specific protection against correlations issues.

3.3.2 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 3.14. It allows to have some coefficients set exactly to zero after some draws. The package **spikeslab** for R is on CRAN.

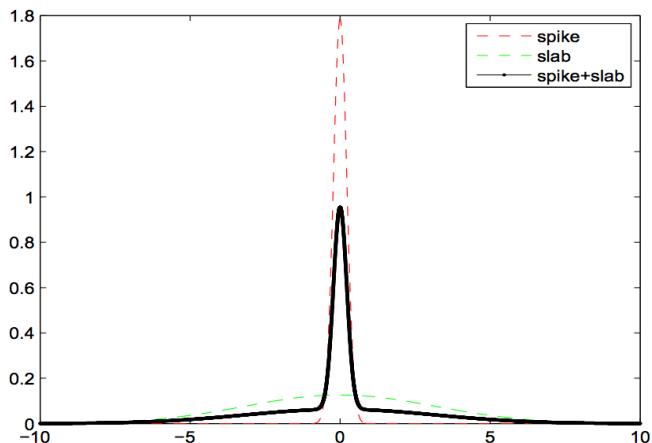


Figure 3.14: The spike and the slab

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.

3.4 Non-linear regression methods

3.4.1 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.

The main idea is to compute the inverse regression $\mathbb{P}(\mathbf{X}|\mathbf{Y})$ instead of $\mathbb{P}(\mathbf{Y}|\mathbf{X})$ to perform a weighted principal component analysis, with which we can identify the effective dimension reducing directions [Li, 1991]. But it requires strong hypotheses on the distribution of \mathbf{X} (elliptic hypothesis), even if it is possible to neglect this hypotheses [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 ([Weisberg, 2002]).

3.4.2 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 cannot be interpreted easily (and cannot be interpreted at all if too complex). So it does not correspond to our needs. Interpretation remains our first goal and prediction comes only in second place.

3.4.3 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 they 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 to 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 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.

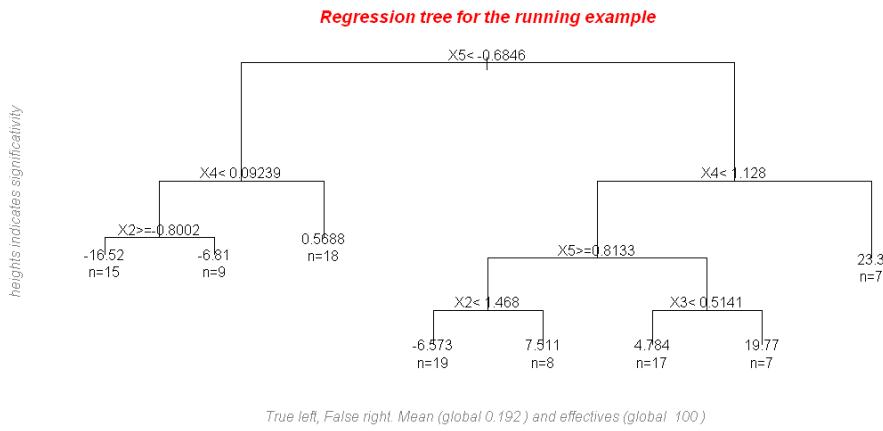


Figure 3.15: Regression tree obtain with the package **CorReg** (graphical layer on top of the **rpart** package) on the running example.

3.4.4 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 are implemented in the package **rpart** for R, on CRAN ([Therneau et al., 2014]). 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 3.16 because a same variable will be used several times and the tree will fail to give a simple interpretation as “ \mathbf{Y} and \mathbf{X}^1 are proportional”. Trivial case : $\mathbf{Y} = \mathbf{X}^1 + \varepsilon_Y$ where $\varepsilon_Y \sim \mathcal{N}(0, \sigma_Y^2 \mathbf{I}_n)$ with $\sigma_Y^2 = 0.5$.

So CART will be used as a complementary tool for datasets with both quantitative and qualitative covariates or when the dependence between \mathbf{Y} and \mathbf{X} is not linear. We will focus our research on linear models with only quantitative variables.

Apart from linear models, 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. More details in the book from Hastie [Hastie et al., 2009].

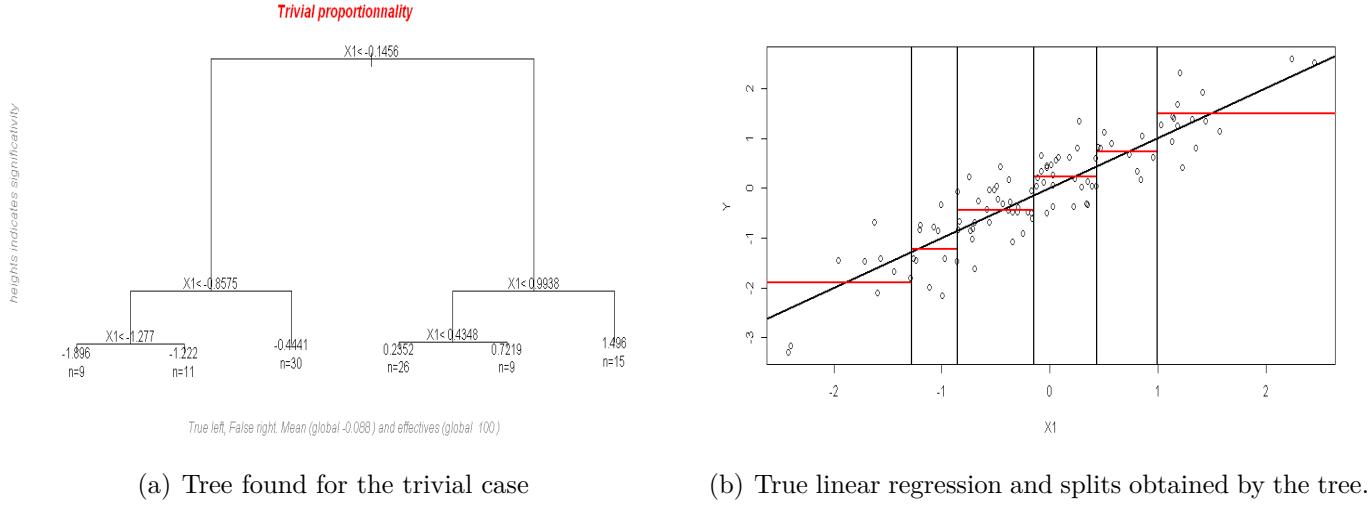


Figure 3.16: Predictive model associated to the tree (red) and true model (black)

3.5 Taking correlations into account

3.5.1 Principal Component Regression (PCR)

Principal Component Regression (PCR)[Jackson, 2005] consists in using the axis from the Principal Component Analysis (PCA) of \mathbf{X} instead of \mathbf{X} itself. Then we have orthogonal covariates. The dataset \mathbf{X} is standardized to have 0 mean and variance 1 for each of the covariates. Dimension reduction is done by keeping only the $M \leq d$ 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 \mathbf{X}^j .

$$\hat{\beta}_M = \mathbf{V}_M \hat{\gamma}_M \in \mathbb{R}^d \text{ where} \quad (3.12)$$

$$\hat{\gamma}_M = (\mathbf{W}'_M \mathbf{W}_M)^{-1} \mathbf{W}'_M \mathbf{Y} \text{ with} \quad (3.13)$$

$$\mathbf{W}_M = \mathbf{X} \mathbf{V}_M \quad (3.14)$$

with \mathbf{V}_M the submatrix of the M first columns of \mathbf{V} d -square matrix of the orthonormal set of right singular vectors of \mathbf{X} defined by:

$$\mathbf{X} = \mathbf{U} \Delta \mathbf{V}' \quad (3.15)$$

with Δ the $d \times d$ diagonal matrix which has on its diagonal the positive eigen values of \mathbf{X} in descending order and \mathbf{U} is the $n \times d$ matrix of the orthonormal set of left singular vectors of \mathbf{X} .

Principal Component Regression requires to choose M the number of axis to keep. Finally, even if dimension reduction is effective when $M < d$ 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. Another problem is that principal components are constructed to explain \mathbf{X} instead of \mathbf{Y} even if there is no reason that relevant variables to explain \mathbf{X} stay relevant to explain \mathbf{Y} .

3.5.2 Partial Least Squares Regression (PLS)

Partial Least Square Regression (PLS)[Abdi, 2003, Geladi and Kowalski, 1986] also relies on a combination of the columns of \mathbf{X} but this combination depends on \mathbf{Y} . The dataset \mathbf{X} is also standardized to have 0 mean and variance 1 for each of the covariates. PLS regression searches for a set of components (called latent vectors) that performs a simultaneous decomposition of \mathbf{X} and \mathbf{Y} with the constraint that these components explain as much as possible of the covariance between \mathbf{X} and \mathbf{Y} . It follows an algorithm that leads to construct successively M orthogonal latent variables that are linear combination of the \mathbf{X}^j :

1. Standardize \mathbf{X} and set $\hat{\mathbf{Y}}^{(0)} = \bar{\mathbf{Y}}\mathbf{1} = (\bar{\mathbf{Y}}, \dots, \bar{\mathbf{Y}})$ and $\mathbf{R}_j^{(0)} = \mathbf{X}^j, j = 1, \dots, d$
2. For $m = 1, \dots, d$ do:
 - (a) $\mathbf{z}_m = \sum_{j=1}^d \hat{\varphi}_{mj} \mathbf{R}_j^{(m-1)}$, where $\hat{\varphi}_{mj} = \langle \mathbf{R}_j^{(m-1)}, \mathbf{Y} \rangle$.
 - (b) $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{Y} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle$.
 - (c) $\hat{\mathbf{Y}}^{(m)} = \hat{\mathbf{Y}}^{(m-1)} + \hat{\theta}_m \mathbf{z}_m$.
 - (d) Orthogonalize each $\mathbf{R}_j^{(m-1)}$ with respect to $\mathbf{z}_m : \mathbf{R}_j^{(m)} = \mathbf{R}_j^{(m-1)} - [\langle \mathbf{z}_m, \mathbf{R}_j^{(m-1)} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle] \mathbf{z}_m, j = 1, \dots, d$.
3. Output the sequence of fitted vectors $\{\hat{\mathbf{Y}}^{(m)}\}_1^d$. Since the $\{\mathbf{z}_l\}_1^M$ are linear in \mathbf{X} , so is $\hat{\mathbf{Y}}^{(M)} = \mathbf{X}\hat{\beta}_{PLS}(M)$ where $\hat{\beta}_{PLS}(M)$ for a given value of M can be recovered from the sequence of PLS transformations.

where $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_i A_i B_i$ is the classical scalar product for two equally sized vectors \mathbf{A} and \mathbf{B} . Like the PCR, PLS regression can give an efficient dimension reduction (with small values of M) but does not really select relevant covariates and interpretation requires to first interpret the latent variables generated so it is not adapted to our needs.

The R package `pls` on CRAN computes both Principal Component Regression and Partial Least Squares Regression ([Mevik et al., 2013]).

3.5.3 Simultaneous Equation Model (sem) and Path Analysis

Applied statistics for non statisticians are well developed in sociology where interpretation stakes are fare beyond prediction. Sociologists use simple models like linear regression (often with $R^2 < 0.2$) and describe complex situations with systems of linear regressions. Such systems are called Structural Equation Model or Simultaneaous Equation Model, better known as SEM [Davidson and MacKinnon, 1993]. Several softwares, from the open-source GRETL [Cottrell and Lucchetti, 2007] to proprietary STATA, does implement the SEM. The systems allow to describe which covariates have an influence on others with an orientation that users can interpret as causality [Pearl, 2000, Pearl, 1998].

SEM are easy to understand for non-statisticians and can be resumed by Directed Acyclic Graphs (DAG) as the Bayesian networks do. But the problem is that the structure of regression between the covariates is defined *a priori*. SEM are used to confirm sociological theories, not to create new theories.

Moreover, estimation of recursive SEM, without instrumental variables is exactly a succession of independent OLS (confirmed with both GRETL and STATA) so the structure is only used for interpretation, not for estimation [Brito and Pearl, 2006]. Last but not least, there is no

specific status for a response variable, each regression has the same status. We want to be able to model complex dependencies within the covariates and use this knowledge to estimate and understand a specific distinct response variable.

3.5.4 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 ([Henningsen and Hamann, 2007]).

3.5.5 Selvarclust: Linear regression within covariates for clustering

`Selvarclust` is a software written in C++ modeling dependencies within a dataset [Maugis et al., 2009] in a Gaussian clustering context³. The idea is to allow covariates to have different roles (S, R, U, W) for the clustering:

- S stands for the subset of the relevant covariates for clustering
- U and W form a partition of the complementary subset of S , with U the subset of irrelevant covariates depending linearly from relevant covariates and W the subset of covariates totally irrelevant and independent from relevant covariates
- R is the subset of S that contains the covariates explaining those in U , with $R \cap U = \emptyset$.

So we have a system of linear regression:

$$\mathbf{X}^U | \mathbf{X}^R; \boldsymbol{\alpha}, \boldsymbol{\Omega} = a + \mathbf{X}^R \boldsymbol{\alpha} + \boldsymbol{\varepsilon} \quad (3.16)$$

where \mathbf{X} is the $n \times d$ dataset as above, \mathbf{X}^R and \mathbf{X}^U are the sub-matrix of \mathbf{X} formed by the covariates in R and U respectively, $\boldsymbol{\alpha}$ is the $\text{Card}(R) \times \text{Card}(U)$ matrix of the regression coefficients, a is the $1 \times \text{Card}(U)$ intercept vector and $\boldsymbol{\varepsilon}$ the $n \times \text{Card}(U)$ matrix of the Gaussian error terms with zero mean and variance $\boldsymbol{\Omega} \in \mathbb{R}^{\text{Card}(U) \times \text{Card}(U)}$.

It leads to decompose the density of \mathbf{X} in three terms $f_{clust}, f_{reg}, f_{indep}$ whose product will give the joint density of \mathbf{X} :

- The relevant variables for clustering of \mathbf{X} are assumed to follow a Gaussian mixture with K components and a form m (see [Biernacki et al., 2006] for more details about the possible forms):

$$f_{clust}(\mathbf{X}^S | K, m, \boldsymbol{\theta}) = \sum_{k=1}^K p_k \Phi(\mathbf{X}^S | \mu_k, \Sigma_k) \quad (3.17)$$

where the parameter vector is $\boldsymbol{\theta} = (p_1, \dots, p_K, \mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K)$, with $\sum_{k=1}^K p_k = 1$, the proportion vector and the variance matrices satisfying the form m .

- The likelihood associated to the linear regression of \mathbf{X}^U on \mathbf{X}^R is then

$$f_{reg}(\mathbf{X}^U | r, a + \mathbf{X}^R \boldsymbol{\alpha}, \boldsymbol{\Omega}) = \prod_{i=1}^n \Phi(\mathbf{X}_i^U | a + \mathbf{X}_i^R \boldsymbol{\alpha}, \boldsymbol{\Omega}) \quad (3.18)$$

where r is the form of $\boldsymbol{\Omega}$, the variance matrix of the residual $\boldsymbol{\varepsilon}$.

³<http://www.math.univ-toulouse.fr/~maugis/SelvarClustHomepage.html>

- Variables independent of the variables that are relevant for clustering are assumed to follow a Gaussian distribution (If they were independent and following a Gaussian mixture they would then be relevant for clustering) with mean vector γ and variance matrix τ with form l :

$$f_{indep}(\mathbf{X}^W | l, \gamma, \tau) = \prod_{i=1}^n \Phi(\mathbf{X}_i^W | \gamma, \tau) \quad (3.19)$$

Selvarclust is one step beyond Simultaneous Equation Modeling with an algorithm to find the structure but:

- It is about clustering and not regression (not the same application field) so we don't have here any response variable neither method to use this structure for estimation of another regression. And the goal is not to find the structure but to find a Gaussian clustering.
- It uses stepwise-like algorithm without protection against correlations [Raftery and Dean, 2006] even if 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 model and a plug-in model. Both algorithm are compared on a dataset from Maugis in section 7.2.1.

Aiming to realize Gaussian clustering, **Selvarclust** does estimate a multivariate Gaussian mixture on \mathbf{X} with dependencies within the covariates. Our algorithm will only aim to find a linear sub-regression structure within \mathbf{X} , relying on some additional hypotheses of independence so the two methods cannot be directly compared. The only common point is the existence of a linear sub-regression model and an algorithm that estimates it. Thus results in section 7.2.1 are just informative.

Part I

Pretreatment for correlations

Chapter 4

Decorrelating covariates by a generative model

Abstract: We give an explicit model for correlation between covariates by a linear regression system. It helps to better understand the dataset and leads to a pretreatment by variable selection. This pretreatment allows to reduce variance of the estimator and then to distinguish irrelevant covariates from redundant covariates.

4.1 Introduction

Most of the above methods do not take explicitly the correlations into account, even if the clustering methods may group the correlated covariates together. The idea of the present thesis is that if we know explicitly the correlations, we could use this knowledge to avoid specific problem it causes. Correlations are thus new information to reduce the variance without adding any bias. Modeling explicitly linear correlation between variables already exists in statistics. In Gaussian model-based clustering, `Selvarclust` [Maugis et al., 2009] considers that some irrelevant covariates for clustering are in linear regression with some relevant ones. We propose to transpose this method for linear regression. More precisely, correlations are modeled through a system of linear sub-regressions between covariates. The set of covariates which are *never* at the place of a response variable in these sub-regressions is finally the greatest set of orthogonal covariates.

Marginalizing over the dependent co-variables leads then to a linear regression (in relation to the initial response variable) with only orthogonal covariates. This marginalization step can be viewed also as a variable selection step but guided only by the correlations between covariates. Advantages of this approach is twofold. First, it improves interpretation through a good readability of dependency between covariates. Second, this marginal model is still a “true” model provided that both the initial regression model and all the sub-regressions are “true”. As a consequence, the associated OLS will preserve an unbiased estimate but with a possibly reduced variance comparing to the OLS with the full regression model. The fact is that the variance decreases depends on the residual variances involved in the sub-regressions: The more the sub-regressions are marked, the less will be the variance of associated OLS. In fact, any other estimation method than OLS can be plugged after the marginalization step. Indeed, it can be viewed as a pretreatment against correlation which can be chained after with dimension reduction methods, without no more suffering from correlations this time.

4.2 Explicit modeling of the correlations

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

We focus now on an original manner to solve the covariates correlation problem. The covariates number problem will be solved at a second stage by standard methods, once only decorrelated covariates will be identified. The proposed method relies on the two following hypotheses.

Hypothesis 1 *In order to take into account the covariates correlation problem, we make the hypothesis correlation between covariates is only the consequence that some covariates linearly depend on some other covariates. More precisely, there are $d_r \geq 0$ such “sub-regressions”, each sub-regression $j = 1, \dots, d_r$ having the covariate $\mathbf{X}^{J_r^j}$ as response variable ($J_r^j \in \{1, \dots, p\}$ and $J_r^j \neq J_r^{j'}$ if $j \neq j'$) and having the $d_p^j > 0$ covariates $\mathbf{X}^{J_p^j}$ as predictor variables ($J_p^j \subset \{1, \dots, d\} \setminus J_r^j$ and $d_p^j = |J_p^j|$ the cardinal of J_p^j):*

$$\mathbf{X}^{J_r^j} | \mathbf{X}^{J_p^j}; \boldsymbol{\alpha}_j, \sigma_j^2 = \mathbf{X}^{J_p^j} \boldsymbol{\alpha}_j + \boldsymbol{\varepsilon}_j, \quad (4.1)$$

where $\boldsymbol{\alpha}_j \in \mathbb{R}^{d_r^j}$ ($\alpha_j^h \neq 0$ for all $j = 1, \dots, d_r$ and $h = 1, \dots, d_p^j$) and $\boldsymbol{\varepsilon}_j \sim \mathcal{N}_n(\mathbf{0}, \sigma_j^2 \mathbf{I})$.

Hypothesis 2 *In addition, we make the complementary hypothesis that the response covariates and the predictor covariates are totally disjoint: for any sub-regression $j = 1, \dots, d_r$, $J_p^j \subset J_f$ where $J_r = \{J_r^1, \dots, J_r^{d_r}\}$ is set of all response covariates and $J_f = \{1, \dots, d\} \setminus J_r$ is the set of all non response covariates of cardinal $d_f = d - d_r = |J_f|$. We call this hypothesis the uncrossing rule.*

This second assumption allows to obtain very simple sub-regressions sequences, discarding hierarchical ones, in particular uninteresting cyclic sub-regressions. However it is not too much restrictive since any hierarchical (but non-cyclic) sequence of sub-regressions can be agglomerated into a non-hierarchical sequence of sub-regressions, even if it may implies to partially loose information through variance increase in the new non-hierarchical sub-regressions. It is made by just successively replacing endogenous covariates by their sub-regression when they are also exogenous in some other sub-regressions.

Further notations In the following, we will note also $\mathbf{J}_r = (J_r^1, \dots, J_r^{d_r})$ the d_r -uple of all the response variable (to be not confused with the corresponding set J_r previously defined), $\mathbf{J}_p = (J_p^1, \dots, J_p^{d_r})$ the d_r -uple of all the predictors for all the sub-regressions, $\mathbf{d}_p = (d_p^1, \dots, d_p^{d_r})$ the associated number of predictors and $\mathbf{S} = (\mathbf{J}_r, \mathbf{J}_p)$ the global model of all the sub-regressions. As more compact notations, we define also $\mathbf{X}_r = \mathbf{X}^{J_r}$ the whole set of response covariates and also $\mathbf{X}_f = \mathbf{X}^{J_f}$ the all other covariates, denominating now as *free* covariates, including those used as predictor covariates in \mathbf{J}_p . An illustration of all these notations is applied on the running example at the end of this section. The parameters are also stacked together: $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_{d_r})$ denotes the global coefficient of sub-regressions and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \dots, \sigma_{d_r}^2)$ denotes the corresponding global variance.

Remarks

- Sub-regressions defined in (4.1) are very easy to understand by any practitioner and, thus, will give a clear view of all the correlations present in the dataset at hand.
- As a consequence of Hypotheses 1 and 2, “free” covariates \mathbf{X}_f are *all* decorrelated.
- We have considered correlations between the covariates of the main regression on \mathbf{Y} , not between the residuals. Thus \mathbf{S} does not depend on \mathbf{Y} and it can be estimated independently as we will see in Chapter 6, even with a larger dataset (if missing values in \mathbf{Y}).
- The model of sub-regressions \mathbf{S} gives a system of linear regressions that can be viewed as a recursive Simultaneous Equation Model (SEM)[Davidson and MacKinnon, 1993, Timm, 2002] or also as a Seemingly Unrelated Regression (SUR) [Zellner, 1962] with \mathbf{X}_r the set of endogenous covariates.
- Each sub-regression may imply a distinct subset of explicative covariates from \mathbf{X}_f .
- Here we suppose the $\boldsymbol{\varepsilon}_j$ independent so we estimate them by separate OLS. But in other cases SUR takes into account correlations between residuals and could be used to estimate the $\boldsymbol{\alpha}^j$.

In the running example: $J_f = \{1, 2, 4, 5\}$, $d_f = 4$ and $\mathbf{X}_f = (\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^4, \mathbf{X}^5)$. We have $d_r = 1$ response covariate $\mathbf{X}^3 | \mathbf{X}^1, \mathbf{X}^2 = \mathbf{X}^1 + \mathbf{X}^2 + \boldsymbol{\varepsilon}_1$ where $\boldsymbol{\varepsilon}_1 \sim \mathcal{N}_n(\mathbf{0}, \sigma_1^2 \mathbf{I})$. Thus, $\boldsymbol{\alpha}_1 = (1, 1)'$, $\mathbf{J}_r = (3)$, $J_r = \{3\}$, $\mathbf{X}_r = (\mathbf{X}^3)$, $\mathbf{d}_p = (2)$, $\mathbf{J}_p = (\{1, 2\})$, $\mathbf{X}^{J_p^1} = (\mathbf{X}^1, \mathbf{X}^2)$ and $\mathbf{S} = ((3), (\{1, 2\}))$.

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

The aim is now to use the model of linear sub-regressions \mathbf{S} (that we assume to be known in this part) between some covariates of \mathbf{X} to obtain a linear regression on \mathbf{Y} relying only on uncorrelated variables \mathbf{X}_f . The way to proceed is to marginalize the joint distribution of $\{(\mathbf{Y}, \mathbf{X}_r) | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2\}$ to obtain the distribution of $\{\mathbf{Y} | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2\}$ depending only on uncorrelated variables \mathbf{X}_f :

$$\mathbb{P}(\mathbf{Y} | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2) = \int_{\mathbb{R}^{d_r}} \mathbb{P}(\mathbf{Y} | \mathbf{X}_f, \mathbf{X}_r, \mathbf{S}; \boldsymbol{\beta}, \sigma_Y^2) \mathbb{P}(\mathbf{X}_r | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2) d\mathbf{X}_r. \quad (4.2)$$

We need the following new hypothesis:

Hypothesis 3 *We assume that all errors $\boldsymbol{\varepsilon}_Y$ and $\boldsymbol{\varepsilon}_j$ ($j = 1, \dots, d_r$) are mutually independent. It implies in particular that conditional response covariates $\{\mathbf{X}^{J_r^j} | \mathbf{X}^{J_p^j}, \mathbf{S}; \boldsymbol{\alpha}_j, \sigma_j^2\}$, with distribution defined in (4.1), are mutually independent:*

$$\mathbb{P}(\mathbf{X}_r | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2) = \prod_{j=1}^{d_r} \mathbb{P}(\mathbf{X}^{J_r^j} | \mathbf{X}^{J_p^j}, \mathbf{S}; \boldsymbol{\alpha}_j, \sigma_j^2). \quad (4.3)$$

Noting $\boldsymbol{\beta}_r = \boldsymbol{\beta}_{J_r}$ and $\boldsymbol{\beta}_f = \boldsymbol{\beta}_{J_f}$ the regression coefficients associated respectively to the responses and to the free covariates, we can rewrite (3.1):

$$\mathbf{Y} | \mathbf{X}, \mathbf{S}; \boldsymbol{\beta}, \sigma_Y^2 = \mathbf{X}_f \boldsymbol{\beta}_f + \mathbf{X}_r \boldsymbol{\beta}_r + \boldsymbol{\varepsilon}_Y. \quad (4.4)$$

Combining now (4.4) with (4.1), (4.2) and (4.3), and also independence between each $\boldsymbol{\varepsilon}_j$ and $\boldsymbol{\varepsilon}_Y$, we obtain the following closed-form for the distribution of $\{\mathbf{Y}|\mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2\}$:

$$\mathbf{Y}|\mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2 = \mathbf{X}_f(\boldsymbol{\beta}_f + \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\alpha}_j^*) + \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_Y \quad (4.5)$$

$$= \mathbf{X}_f \boldsymbol{\beta}_f^* + \boldsymbol{\varepsilon}_Y^*, \quad (4.6)$$

where $\boldsymbol{\alpha}_j^* \in \mathbb{R}^{d_f}$ with $(\boldsymbol{\alpha}_j^*)_{J_p^j} = \boldsymbol{\alpha}_j$ and $(\boldsymbol{\alpha}_j^*)_{J_f \setminus J_p^j} = \mathbf{0}$. We can then define the matrix $\boldsymbol{\alpha}^* \in \mathbb{R}^{(d_f \times d_r)}$ of the coefficient of sub-regression to use more compact notation:

$$\mathbf{X}_r|\mathbf{X}_f, \mathbf{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2 = \mathbf{X}_f \boldsymbol{\alpha}^* + \boldsymbol{\varepsilon} \quad (4.7)$$

Where $\boldsymbol{\varepsilon}$ is the $n \times d_r$ matrix of the $\boldsymbol{\varepsilon}^j$.

Consequently, we have obtained a new regression expression of \mathbf{Y} but relying now *only* on uncorrelated covariates \mathbf{X}_f . This decorrelation process has also acted like a specific variable selection process because $\mathbf{X}_f \subseteq \mathbf{X}$. These two statements are expected to decrease the variance of further estimates of $\boldsymbol{\beta}$.

However, the counterpart is twofold. First, this regression has a higher residual variance than the initial one since it is now $\sigma_Y^{2*} = \sigma_Y^2 + \sum_{j=1}^{d_r} \beta_{J_r^j}^2 \sigma_j^2$ instead of σ_Y^2 . Second, variable selection being equivalent to set $\hat{\boldsymbol{\beta}}_r = \mathbf{0}$, it implies possibly biased estimates of $\boldsymbol{\beta}_r$. As a conclusion, we are faced with a typical *bias-variance trade off*. We will illustrate it in the chapter 5.

In practice, the strategy we propose is to rely estimate of $\hat{\boldsymbol{\beta}}$ upon Equation (4.6). The practitioner can choose any estimate of its choice, like OLS or any variable selection procedure like LASSO. In other words, it is possible to see (4.6) as a kind of *pretreatment* for decorrelating covariates, while assuming nothing on the subsequent estimate process.

In the following, we will denote by CORREG (for *Correlations and Regression*) the new proposed strategy.

Remark: Identifiability of (\mathbf{X}, \mathbf{S}) is not necessary to use a given structure but helps to find it. Moreover, uncrossing rule restricts the size of \mathcal{S}_d and improves identifiability. A sufficient condition for identifiability is to have at least two regressors in each sub-regression (definition of identifiability and proof this criterion in appendix 12). In the followings, true \mathbf{S} is supposed to be identifiable.

Running example: $\mathbf{Y}|\mathbf{X}_f, \mathbf{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2 = 2\mathbf{X}^1 + 2\mathbf{X}^2 + \mathbf{X}^4 + \mathbf{X}^5 + \boldsymbol{\varepsilon}_1 + \boldsymbol{\varepsilon}_Y$

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 $\boldsymbol{\beta}_f^*$, even with variable selection methods like LASSO or a best subset algorithm like stepwise [Seber and Lee, 2012]. However, we always have $\boldsymbol{\beta}_r^* = \mathbf{0}$.

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 \mathbf{S}) and the classical selection step, with different meanings for obtained zeros in $\hat{\boldsymbol{\beta}}_f^*$

(irrelevant covariates) and for $\hat{\beta}_r^* = 0$ (redundant information). Thus we are able to distinguish the reasons that have lead to keep or remove each covariate and consistency issues do not 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 \mathbf{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 hypotheses. We just play on the traditional bias-variance tradeoff.

4.5 Illustration of the tradeoff conveyed by the pre-treatment

We compare the OLS estimator on \mathbf{X} defined in section 3.1.2 with the estimator obtained by the pre-treatment that is \mathbf{X}_f selection.

For the marginal regression model defined in (4.6) we have the OLS unbiased estimator of β^* :

$$\hat{\beta}_f^* = (\mathbf{X}'_f \mathbf{X}_f)^{-1} \mathbf{X}'_f \mathbf{Y} \text{ and } \hat{\beta}_r^* = \mathbf{0} \quad (4.8)$$

We see in (4.5) that it gives an unbiased estimation of \mathbf{Y} and β^* but in terms of β this estimator could be biased:

$$\mathbb{E}(\hat{\beta}_f^*) = \beta_f + \sum_{j=1}^{d_r} \beta_{J_r^j} \alpha_j^* \text{ and } \mathbb{E}(\hat{\beta}_r^*) = \mathbf{0} \quad (4.9)$$

In return, its variance could be reduced compared to this one of $\hat{\beta}$ given in (3.3) as soon as values of σ_j are small enough (it means strong correlations in sub-regressions) as we can see in the following expression

$$\text{Var}(\hat{\beta}_f^*) = (\sigma_Y^2 + \sum_{j=1}^{d_r} \sigma_j^2 \beta_{J_r^j}^2) (\mathbf{X}'_f \mathbf{X}_f)^{-1} \quad \text{and} \quad \text{Var}(\hat{\beta}_r^*) = \mathbf{0}. \quad (4.10)$$

Indeed, no correlations between covariates \mathbf{X}_f imply that the matrix $\mathbf{X}'_f \mathbf{X}_f$ could be sufficiently better conditioned than the matrix $\mathbf{X}' \mathbf{X}$ involved in (3.3) to balance the added variance $\sum_{j=1}^{d_r} \sigma_j^2 \beta_{J_r^j}^2$ in (4.10). This bias-variance trade off can be resumed by the Mean Squared Error (MSE) associated to both estimates:

$$\text{MSE}(\hat{\beta}) = \| \text{Bias} \|_2^2 + \text{Tr}(\text{Var}(\hat{\beta})) \quad (4.11)$$

$$\text{MSE}(\hat{\beta}) = \sigma_Y^2 \text{Tr}((\mathbf{X}' \mathbf{X})^{-1}), \quad (4.12)$$

$$\text{MSE}(\hat{\beta}^*) = \| \sum_{j=1}^{d_r} \sigma_j \beta_{J_r^j} \|_2^2 + \| \beta_r \|_2^2 + (\sigma_Y^2 + \sum_{j=1}^{d_r} \sigma_j^2 \beta_{J_r^j}^2) \text{Tr}((\mathbf{X}'_f \mathbf{X}_f)^{-1}). \quad (4.13)$$

This bias-variance tradeoff is illustrated with the running example.

4.6 MSE comparison on the running example

In this section, all experiences have been made 100 times and we take the mean to obtain smooth curves. So we have generated 100 times \mathbf{X} and \mathbf{Y} .

The MSE on $\hat{\mathbf{Y}}$ is computed on a validation sample with 1 000 individuals.

Observed mse for ols: It is clear in figures 4.1 and 4.2 that the marginal model is more robust than OLS on \mathbf{X} . We see in figures 4.1 and 4.2 that MSE on $\hat{\mathbf{Y}}_{OLS}$ give the same global results as those on $\hat{\beta}_{OLS}$: the marginal model is better for stronger sub-regressions, smaller samples and weaker main regression. But we notice that when the MSE on $\hat{\beta}_{OLS}$ explodes, the MSE on $\hat{\mathbf{Y}}_{OLS}$ 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.

When sub-regression get weaker (R^2 tends to 0) CORREG remains stable until extreme values (sub-regression nearly fully explained by the noise). The marginal should be particularly useful when some covariates are highly correlated, when the sample size is small or when the residual variance of \mathbf{Y} is large. These three classical problems for OLS make CORREG better. 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).

We notice that the curve is not totally smooth for small samples ($n = 15$) because of numerical approximation of the theoretical MSE. It confirms that matricial inversion is not easy with correlated covariates even with a small number of covariates. But it is only the theoretical MSE and we want to know what happens in the real life.

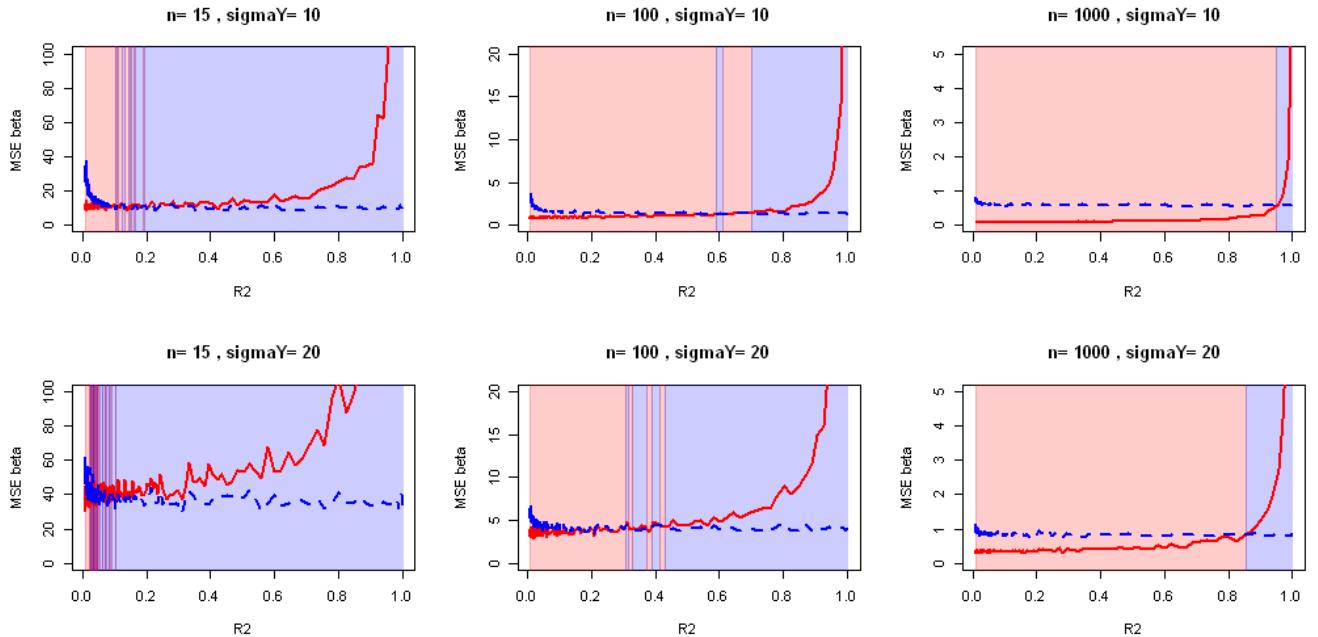


Figure 4.1: Observed MSE on $\hat{\beta}$ of OLS (plain red) and CorReg's marginal model (dotted blue) estimators for varying R^2 of the sub-regression, n and σ_Y . $d = 5$ covariates.

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

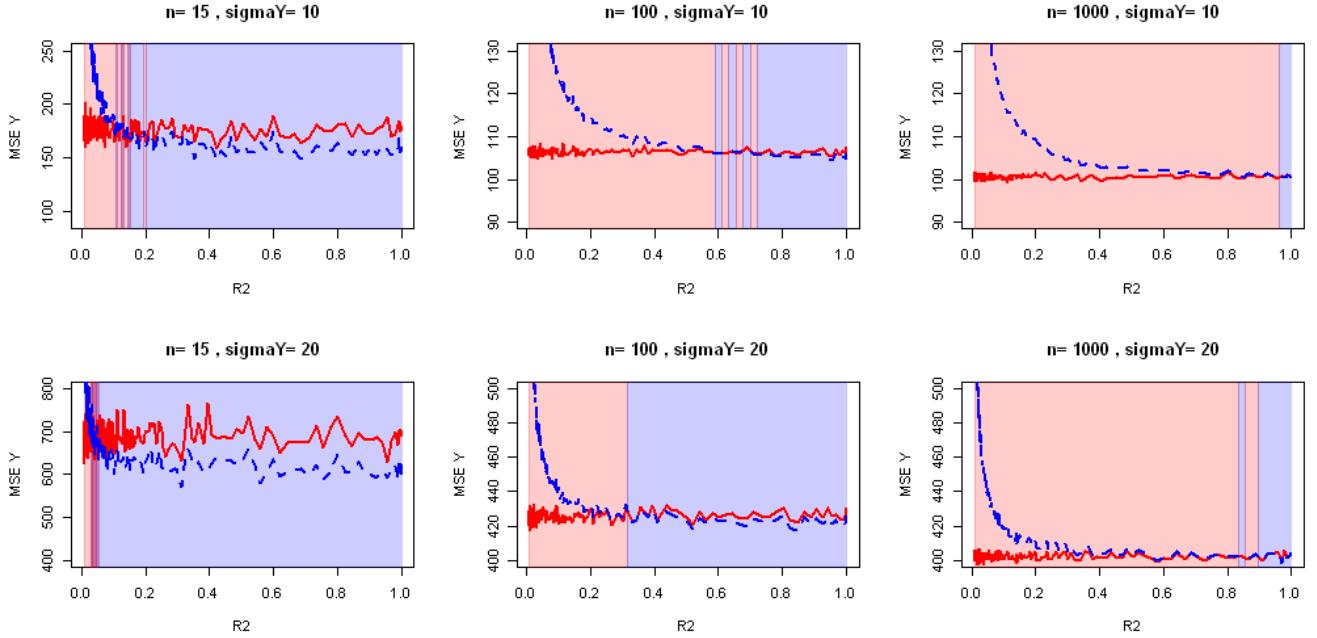


Figure 4.2: Evolution of observed Mean Squared error on \hat{Y}_{OLS} with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

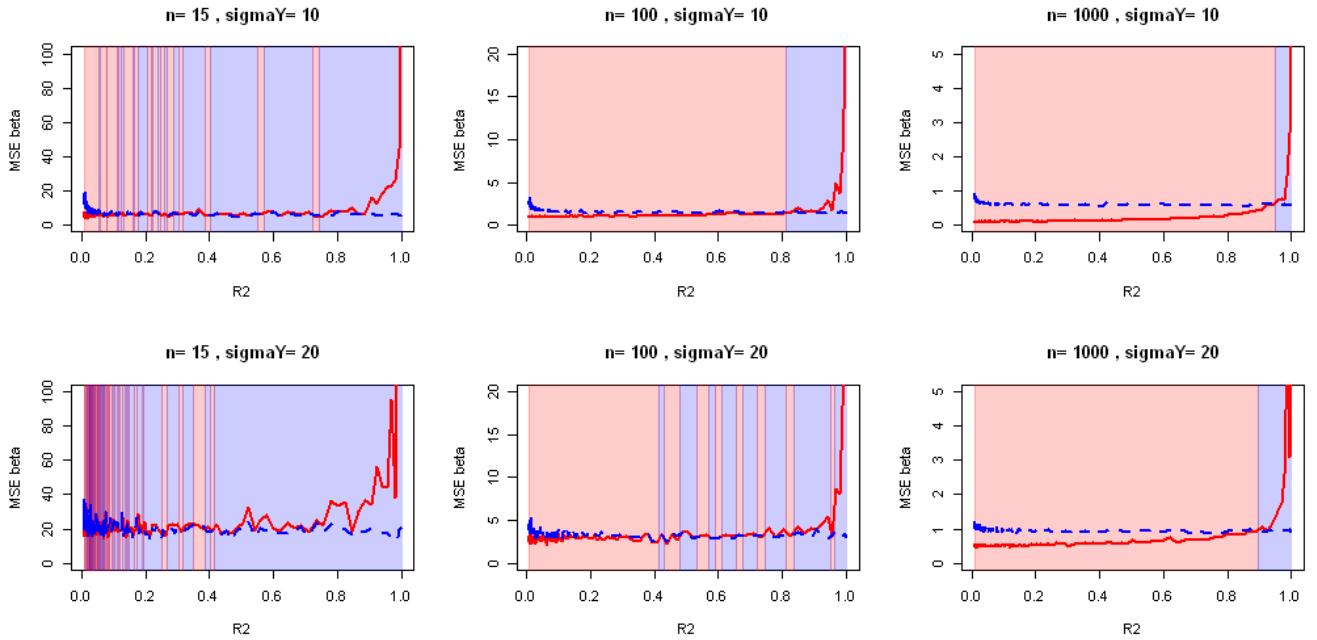


Figure 4.3: Observed MSE on $\hat{\beta}$ of LASSO with LAR on both \mathbf{X} (red) and CorReg's marginal \mathbf{X}^{I_f} (blue) for varying R^2 of the sub-regression, n and σ_Y . $d = 5$ covariates.

Observed mse for variable selection methods: Figure 4.3 shows that variable selection done by the LASSO gives a biased $\hat{\beta}$ by setting some coefficients to 0 but strong correlations makes this bias neutral for prediction (figure 4.4). 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. Elasticnet and stepwise give results quite similar to the LASSO (figures 4.7 to 4.10).

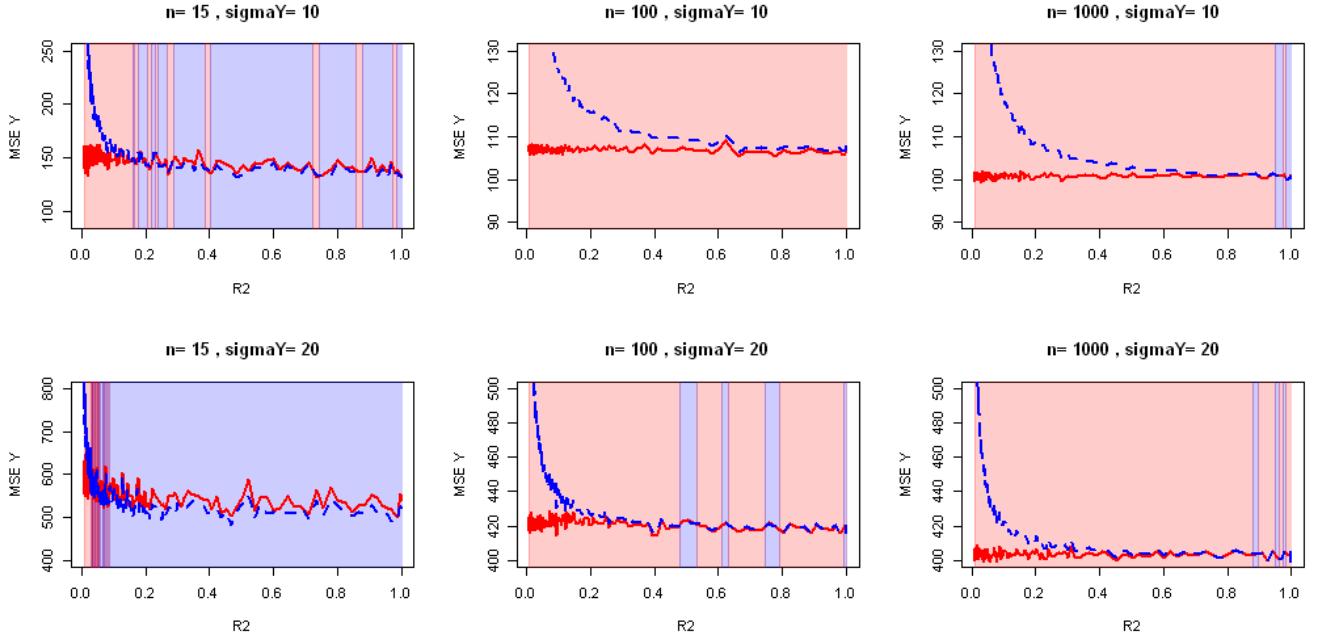


Figure 4.4: Evolution of observed Mean Squared error on \hat{Y}_{LASSO} with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

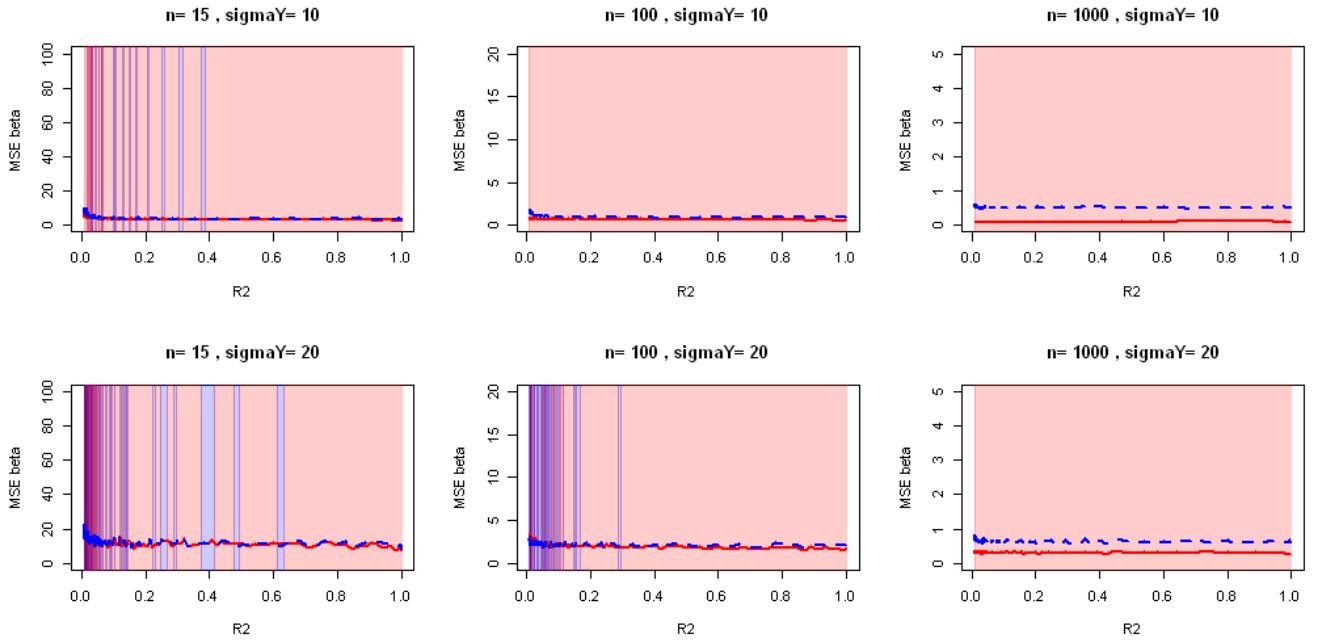


Figure 4.5: Observed MSE on $\hat{\beta}_{ridge}$ on both \mathbf{X} (red) and CorReg's marginal \mathbf{X}_f (blue) for varying R^2 of the sub-regression, n and σ_Y . $d = 5$ covariates.

Observed mse for ridge regression: Here again (figures 4.5 and 4.6), 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 achieve variable selection.



Figure 4.6: Evolution of observed Mean Squared error on \hat{Y}_{ridge} with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

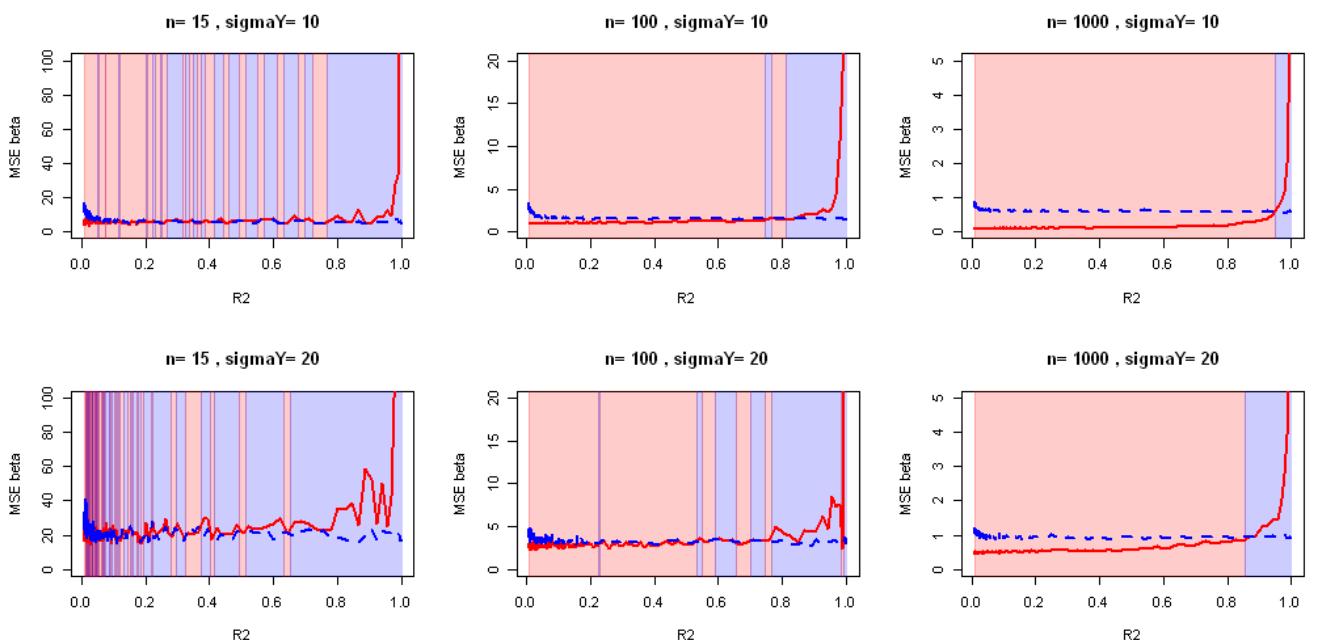


Figure 4.7: Observed MSE on $\hat{\beta}_{elasticnet}$ on both \mathbf{X} (red) and CorReg's marginal \mathbf{X}_f (blue) for varying R^2 of the sub-regression, n and σ_Y . $d = 5$ covariates.



Figure 4.8: Evolution of observed Mean Squared error on $\hat{Y}_{\text{elasticnet}}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

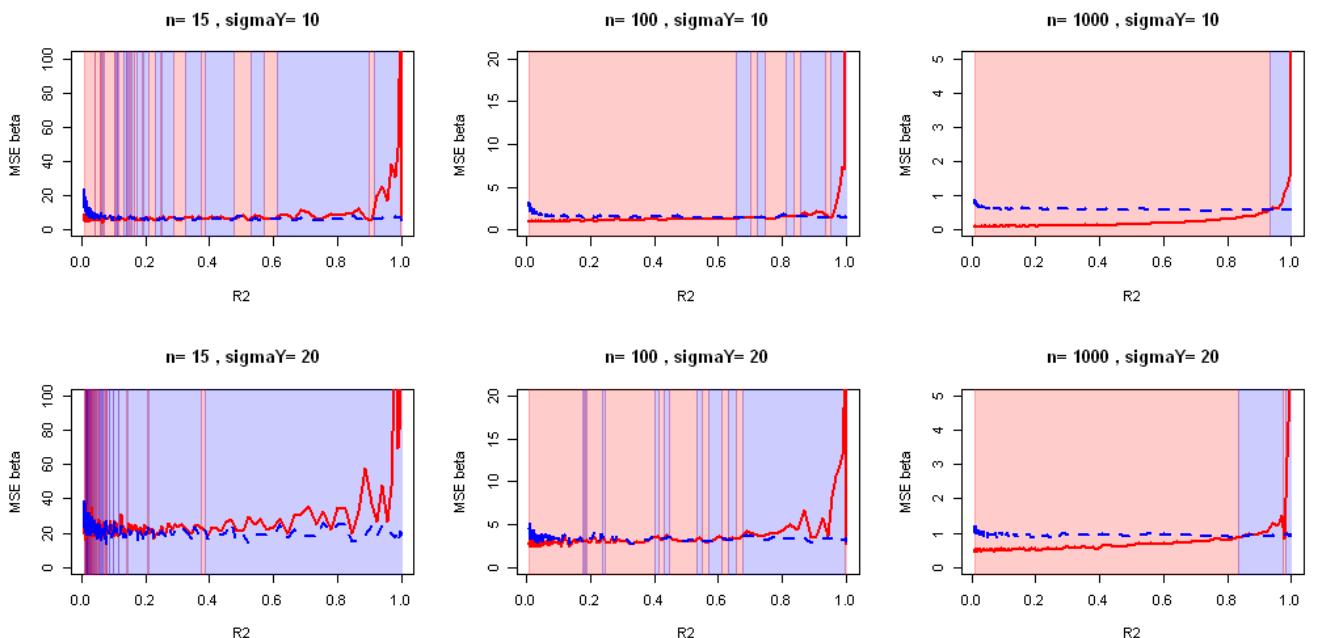


Figure 4.9: Observed MSE on $\hat{\beta}_{\text{stepwise}}$ on both \mathbf{X} (red) and CorReg's marginal \mathbf{X}_f (blue) for varying R^2 of the sub-regression, n and σ_Y . $d = 5$ covariates.

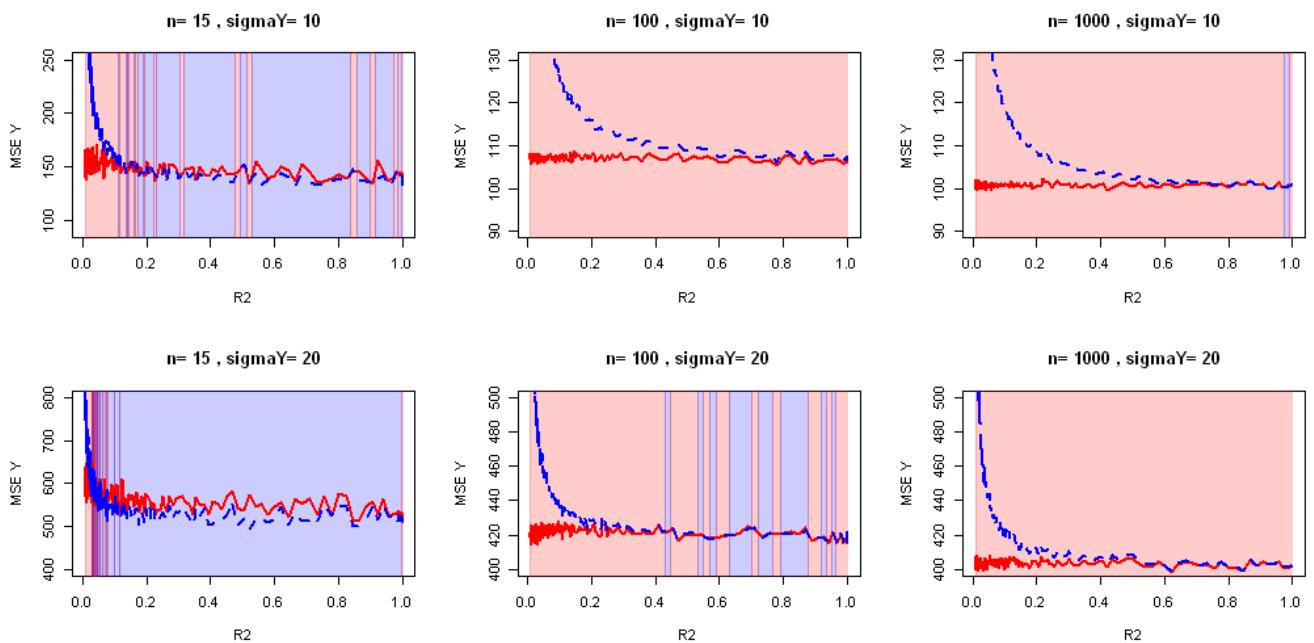


Figure 4.10: Evolution of observed Mean Squared error on $\hat{Y}_{stepwise}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

4.7 Connexion with graphs

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

Graphical representation of \mathbf{S} helps to understand it and can be compared to the Bayesian network representation. It helps to interpret the structure and has also been used to construct the algorithm to find \mathbf{S} (chapter 6). The partition of \mathbf{X} mean that the associated graph is bipartite: vertices follow a partition $(\mathbf{X}_r, \mathbf{X}_f)$ with arcs only going from \mathbf{X}_f to \mathbf{X}_r . We know ([Biggs, 1993]) as a classical result of graph theory that the power of adjacency matrices give the paths in the graph: $\mathbf{G}_{i,j}^k \neq 0$ means that there is at least a path of length k going from \mathbf{X}^i to \mathbf{X}^j . We note \mathcal{S}_d the set of feasible structures (verifying hypothesis 2) with d covariates. Because the graph is bipartite we can deduce that \mathbf{G} is nilpotent: $\mathbf{G}^2 = 0$. And we have the following result:

Theorem: Every binary nilpotent matrix of order 2 can be seen as an adjacency matrix of a structure that respects the uncrossing rule, and *vice versa*. We can deduce that the number of feasible structure with d covariates is the number of d -sized binary nilpotent matrix of order 2.

proof: Every binary matrix can be associated to a structure as an adjacency matrix and every structure can be described by its adjacency matrix. Thus we just have to demonstrate equivalence between uncrossing and nilpotent \mathbf{G} . If there exist a path of length 2 between some vertices i and j (uncrossing rule violated) then $\mathbf{G}_{i,j}^2 \neq 0$ so the matrix is not nilpotent of order 2. So the set of nilpotent matrices is included the set of adjacency matrices associated to \mathcal{S}_d . If \mathbf{G} is nilpotent then there is no path of length 2 in the graph (uncrossing rule verified). So we have the reverse inclusion and then equivalence between the set of adjacency matrices associated to \mathcal{S}_d and the set of nilpotent matrices. \square

We see that \mathbf{G} completely describe \mathbf{S} that is in fact a sparse storage of G . We decompose the structure to enumerate all the feasible structure (and thus all the binary nilpotent matrix of order 2).

The number of possible \mathbf{J}_r for given values of d and d_r is $\binom{d}{d_r} = \frac{d!}{d_r!(d-d_r)!}$ (binomial coefficient).

The number of possible \mathbf{J}_p for given values of d, d_r and \mathbf{J}_r is $(2^{d-d_r} - 1)^{d_r}$.

Thus we have

$$|\mathcal{S}_d| = \sum_{d_r=0}^{d-1} \binom{d}{d_r} (2^{d-d_r} - 1)^{d_r} \quad (4.14)$$

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 d is growing. Next chapter shows results with $d = 40$ and it corresponds to $|\mathcal{S}_{40}| > 7.32 \times 10^{131}$ feasible structures. The function `ProbaZ` of the package `CorReg` gives the number of feasible structures for a given value of d , but R returns $+\infty$ for $d > 62$.

In the running example: $|\mathcal{S}_5| = 841$ and the adjacency matrix is:

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

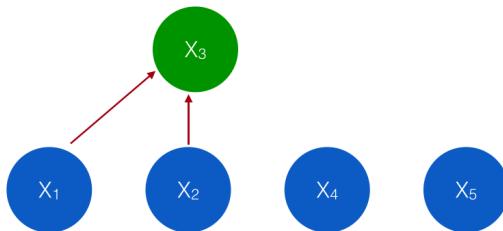


Figure 4.11: The bipartite graph associated to the running example. \mathbf{X}_r is green and \mathbf{X}_f is blue.

Chapter 5

Numerical results with a known structure on more complex datasets

Abstract: First numerical results for a known structure \mathbf{S} (but $\boldsymbol{\alpha}^*$ still has to be estimated). It illustrates efficiency of the variable selection pretreatment made by the marginal model.

5.1 The datasets

We consider regressions on \mathbf{Y} with $d = 40$ covariates and with a R^2 value equal to 0.4. Sub-regressions will have R^2 successively set to $(0.1, 0.3, 0.5, 0.7, 0.99)$. Variables in \mathbf{X}_f arise from a Gaussian mixture model whose the number of components follows a Poisson's law of mean parameter equal to 5. The coefficients of $\boldsymbol{\beta}$ and of the $\boldsymbol{\alpha}_j$'s are independently generated according to the same Poisson distribution but with a uniform random sign. All sub-regressions are of length two ($\forall j = 1, \dots, d_r, d_p^j = 2$ and we have $d_r = 16$ sub-regressions. The datasets are then scaled, to avoid large distortions for variances or for means due to the sub-regressions. Different sample sizes $n \in (30, 50, 100, 400)$ are chosen, thus considering experiments in both situations $n < d$ and $n > d$. Results are based on the true \mathbf{S} used to generate the dataset (function `mixture_generator` in the package `CorReg`).

When $n < d$, a frequently used method is the Moore-Penrose [Katsikis and Pappas, 2008] generalized inverse , thus OLS can obtain some results even with $n < d$. 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 \mathbf{Y} depends on all the covariate and the MSE provided were computed on a validation sample of 1 000 individuals each time. Figures will display both mean and inter-quartile intervals, coloration of the background indicates which curve is lower (*i.e.* better in our case).

5.2 Results

With OLS: Figure 5.1 compares OLS alone to CORREG + OLS. CORREG improves significantly the prediction power of OLS for small values of n and/or heavy sub-regression structures. This advantage then shrinks when n increases because the matrix to invert becomes better-

conditioned and since CORREG does not allow to retrieve that \mathbf{Y} depends on all \mathbf{X} because of the marginalization of some covariates implicated in the sub-regressions. It also illustrates that the model regression in \mathbf{Y} retained by CORREG is more parsimonious.

Variable selection methods: Figure 5.2 shows 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 $n = 400 = 10 \times d$ when used on the whole dataset but keep all the covariates in \mathbf{X}_f when used on the marginal model. The main result here is that the LASSO can be improved by pretreatment selection both with $n < d$ and $n >> d$ with strong correlations so this well known variable selection method really suffers from correlations. Elasticnet and stepwise (Figures 5.3 and 5.4) gives 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 \mathbf{S} and not only variable selection by stepwise like in the method from Maugis [Maugis et al., 2009].

Ridge regression: Figure 5.5 shows that the predictive power of ridge regression is not improved by the marginal model. Ridge regression is protected against correlations but we see that ridge regression applied on \mathbf{X}_f (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 by correlations when variable selection is needed.

Further results are provided with estimated $\tilde{\mathbf{S}}$ (section 7) and then with real industrial datasets (section 8).

5.2.1 OLS when Y depends on all variables in X , true S known

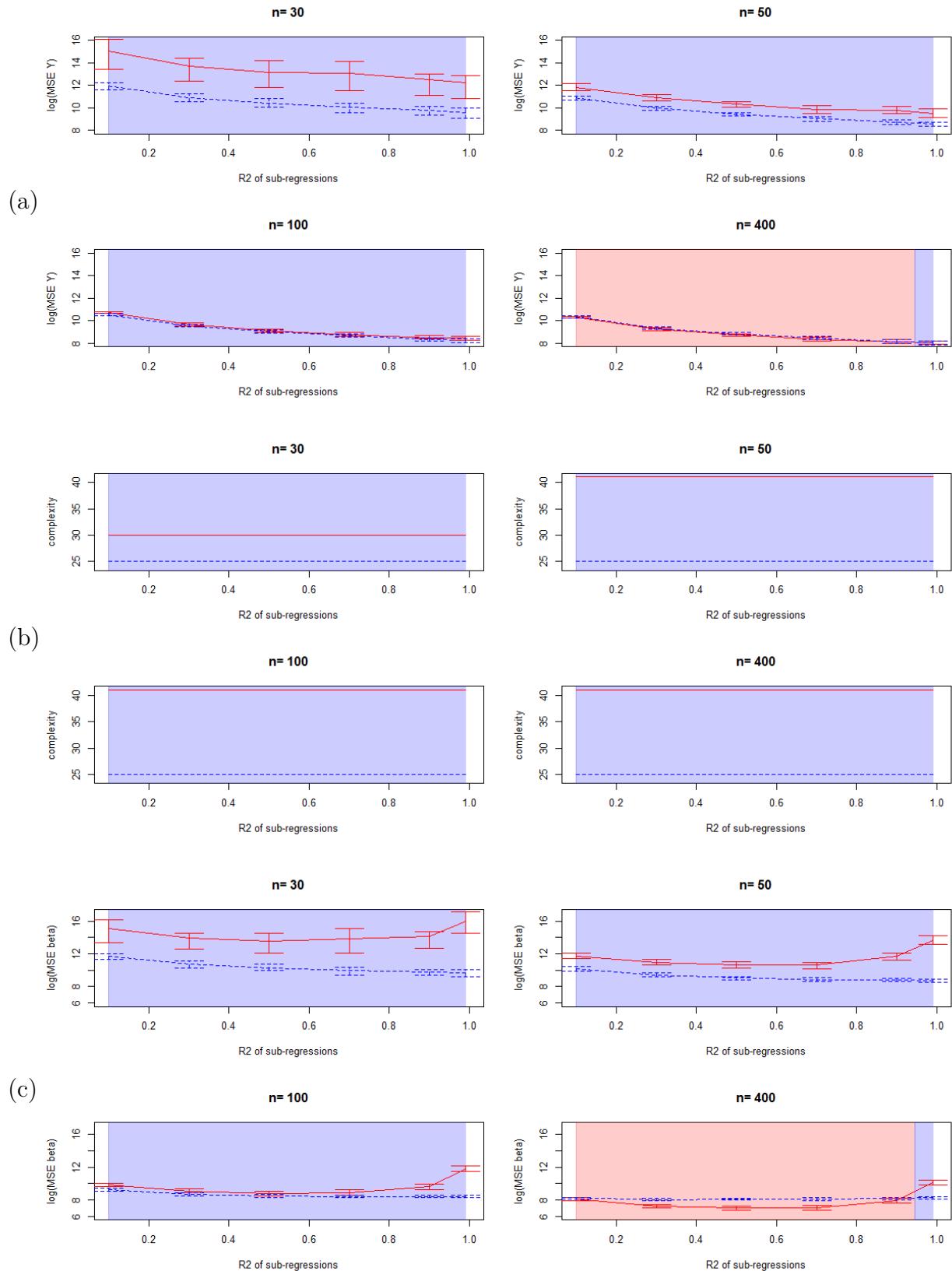


Figure 5.1: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

5.2.2 LASSO when Y depends on all variables in X , true S known

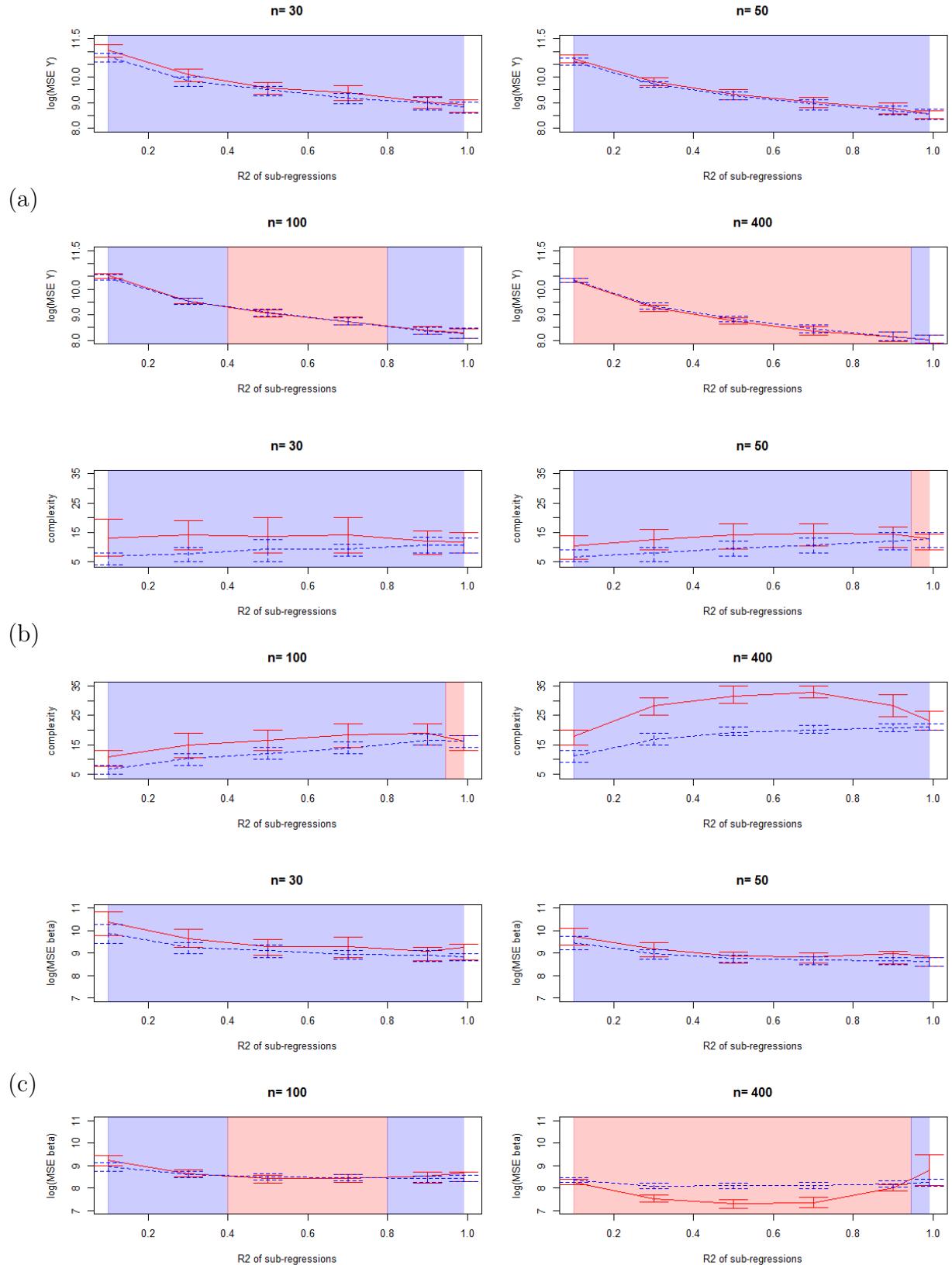


Figure 5.2: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

5.2.3 Elasticnet when Y depends on all variables in X , true S known

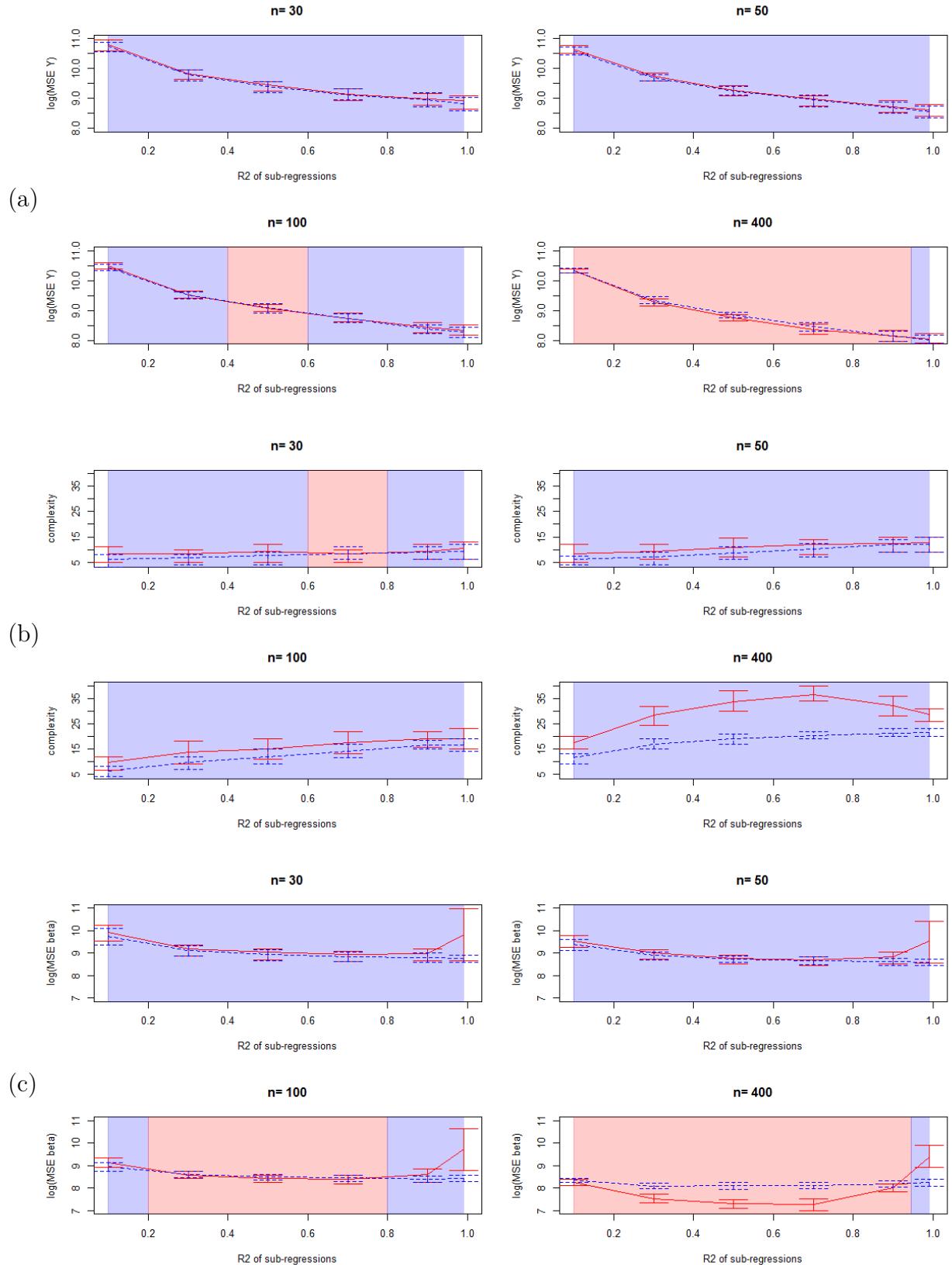


Figure 5.3: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

5.2.4 Stepwise when Y depends on all variables in X , true S known

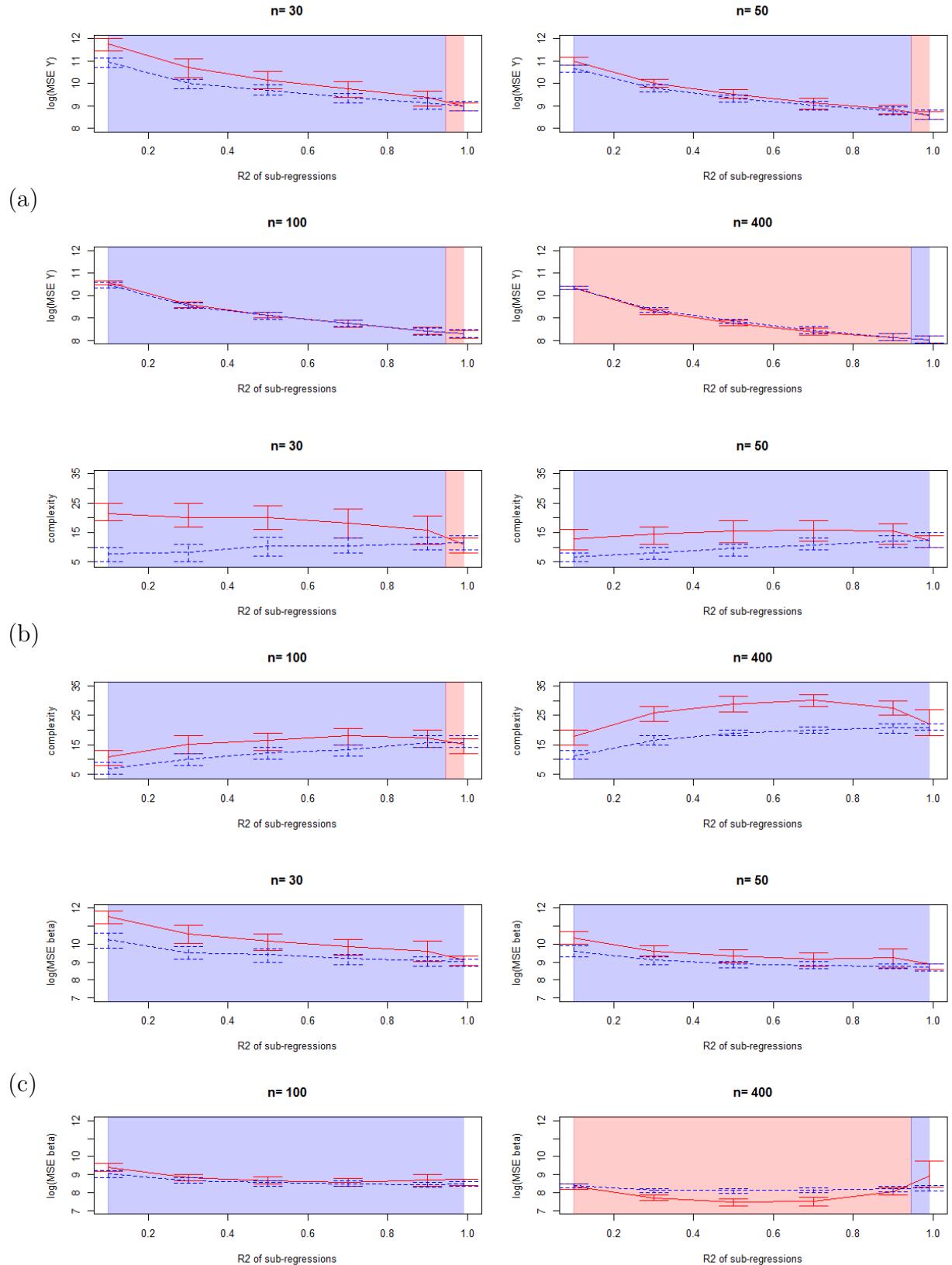


Figure 5.4: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

5.2.5 Ridge when Y depends on all variables in X , true S known

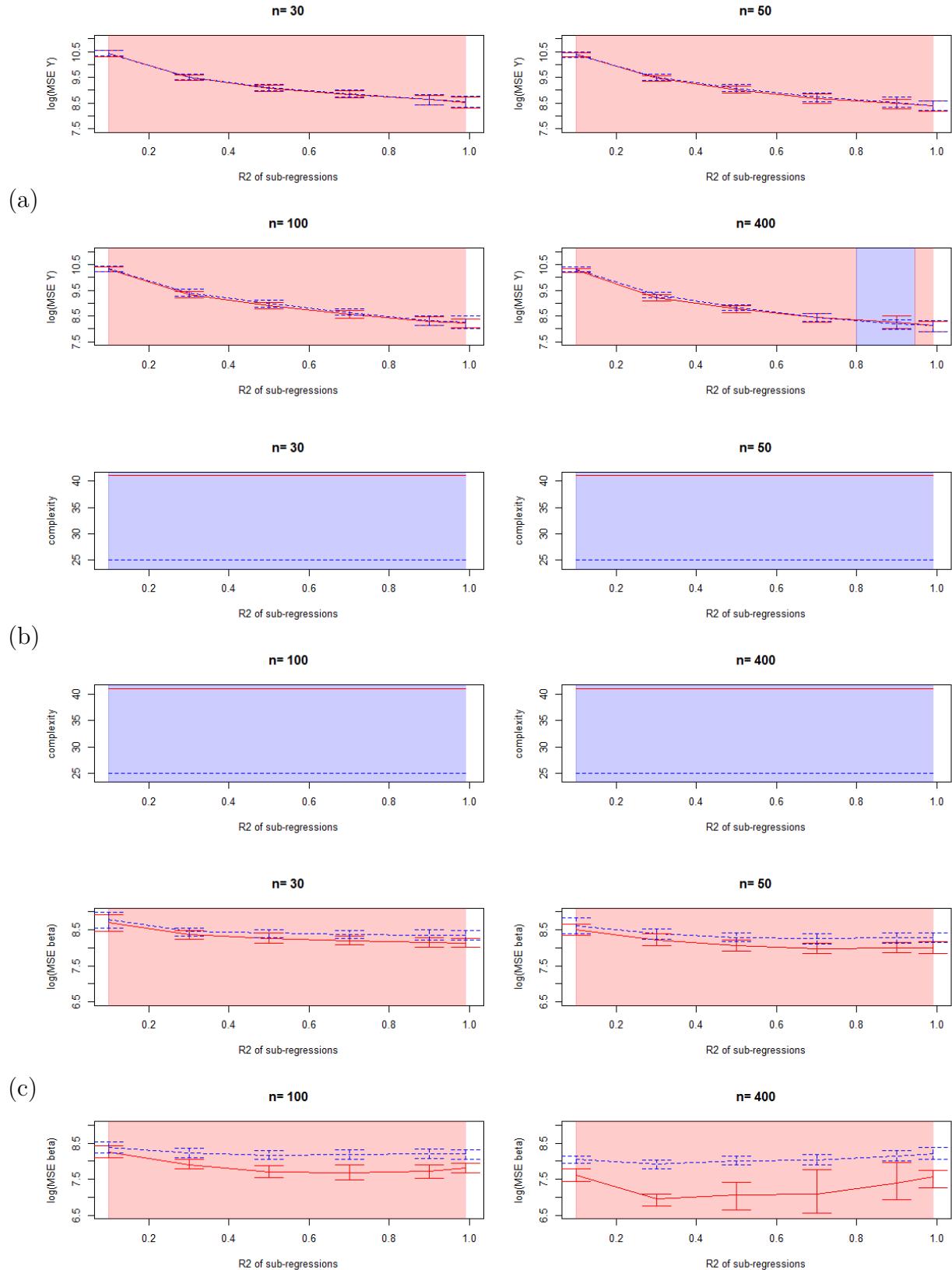


Figure 5.5: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Chapter 6

Estimation of the Structure of subregression by MCMC

Abstract: \mathbf{S} is supposed unknown so we need to find it. An MCMC algorithm is proposed to find \mathbf{S} , relying on a full generative model on \mathbf{X} . A new criterion is also introduced to choose the better \mathbf{S} taking into account the huge number of feasible structures.

In the previous chapters, \mathbf{S} was supposed to be known. But in fact we have to find it so we developed an algorithm to choose the best \mathbf{S} .

6.1 Choice of model

6.1.1 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. 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-samples (k -fold cross-validation) and then each of the k sub-samples is successively used as validation sample for the model learnt 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:

$$\text{MSE}_{\hat{\mathbf{Y}}} = \| \mathbf{Y} - \hat{\mathbf{Y}} \|_2^2 \quad (6.1)$$

6.1.2 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, Schwarz et al., 1978] is a widely used criterion that relies on the likelihood of the dataset knowing the model and 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. BIC is convergent so it asymptotically points out the true model when n rises. The Akaike Information Criterion [Akaike, 1974] 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:

$$\text{BIC} = -2 \ln(L(\hat{\boldsymbol{\theta}}|\mathbf{X})) + |\boldsymbol{\theta}| \ln(n) \quad (6.2)$$

where $L(\boldsymbol{\theta}|\mathbf{X}) = \mathbb{P}(\mathbf{X}|\boldsymbol{\theta})$ is the estimated likelihood of the parameters $\boldsymbol{\theta}$ given the dataset \mathbf{X} , $|\boldsymbol{\theta}|$ is the number of free parameters to estimate and n the number of individuals in the dataset.

6.2 Bayesian approach

Structural equations models are often used in social sciences and economy where a structure is supposed *a priori* but here we want to find it automatically, even if it remains possible to use expert knowledge to complete the structure. 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 \mathbf{S} to avoid cycles.

Cross-validation is very time-consuming and thus not friendly with combinatorial problematics. Moreover, we need a criterion compatible with structures of different sizes (varying d_r) and not related with \mathbf{Y} because the structure is inherent to \mathbf{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]).

6.3 Sub-regression structure in details

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

$$\mathbb{P}(\mathbf{S}|\mathbf{X}) \propto \mathbb{P}(\mathbf{X}|\mathbf{S})\mathbb{P}(\mathbf{S}) = \mathbb{P}(\mathbf{X}_r|\mathbf{X}_f, \mathbf{S})\mathbb{P}(\mathbf{X}_f|\mathbf{S})\mathbb{P}(\mathbf{S}) \quad (6.3)$$

In order to implement this paradigm, we need first to define the three probabilities which are in the right hand of the previous equation.

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

Defining $\mathbb{P}(\mathbf{X}_r|\mathbf{X}_f, \mathbf{S})$ corresponds to the integrated likelihood based on $\mathbb{P}(\mathbf{X}_r|\mathbf{X}_f, \mathbf{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2)$. It can be approximated by a BIC-like approach [Schwarz, 1978]

$$-2 \ln \mathbb{P}(\mathbf{X}_r|\mathbf{X}_f, \mathbf{S}) \approx -2 \ln \mathbb{P}(\mathbf{X}_r|\mathbf{X}_f, \mathbf{S}; \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\sigma}}^2) + (|\boldsymbol{\alpha}| + |\boldsymbol{\sigma}^2|) \ln(n) = \text{BIC}_r(\mathbf{S}), \quad (6.4)$$

where $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\sigma}}^2$ designate respectively the Maximum Likelihood Estimates (MLE) of $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}^2$, and $|\boldsymbol{\psi}|$ designates the number of free continuous parameters associated to the space of any parameter $\boldsymbol{\psi}$.

Defining $\mathbb{P}(\mathbf{X}_f | \mathbf{S})$ It corresponds to the integrated likelihood based on a not yet defined distribution $\mathbb{P}(\mathbf{X}_f | \mathbf{S}; \boldsymbol{\theta})$ on the uncorrelated covariates \mathbf{X}_f and parameterized by $\boldsymbol{\theta}$. In this purpose, we need the following new hypothesis.

Hypothesis 4 All variates \mathbf{X}^j with $j \in J_f$ are mutually independent and arise from the following Gaussian mixture of k_j components

$$\mathbb{P}(\mathbf{X}_f^j | \mathbf{S}; \boldsymbol{\pi}_j, \boldsymbol{\mu}_j, \boldsymbol{\lambda}_j^2) = \sum_{h=1}^{k_j} \pi_{hj} \mathcal{N}_n(\mu_{hj} \cdot (1, \dots, 1)', \lambda_{hj}^2 \mathbf{I}), \quad (6.5)$$

where $\boldsymbol{\pi}_j = (\pi_{1j}, \dots, \pi_{k_j j})$ is the vector of mixing proportions with $\forall 1 \leq h \leq k_j, \pi_{hj} > 0$ and $\sum_{h=1}^{k_j} \pi_{hj} = 1$, $\boldsymbol{\mu}_j = (\mu_{1j}, \dots, \mu_{k_j j})$ is the vector of centers and $\boldsymbol{\lambda}_j^2 = (\lambda_{1j}^2, \dots, \lambda_{k_j j}^2)$ is the vector of variances. We stack together all the mixture parameters in $\boldsymbol{\theta} = (\boldsymbol{\pi}_j, \boldsymbol{\mu}_j, \boldsymbol{\lambda}_j^2; j \in J_f)$. We now have a full generative model on \mathbf{X} .

Noting $\hat{\boldsymbol{\theta}}$ the MLE of $\boldsymbol{\theta}$, the BIC approximation can then be used again:

$$-2 \ln \mathbb{P}(\mathbf{X}_f | \mathbf{S}) \approx -2 \ln \mathbb{P}(\mathbf{X}_f | \mathbf{S}; \hat{\boldsymbol{\theta}}) + |\boldsymbol{\theta}| \ln(n) = \text{BIC}_f(\mathbf{S}). \quad (6.6)$$

Defining $\mathbb{P}(\mathbf{S})$ The most standard choice consists of putting a uniform distribution on the model space \mathcal{S}_d , this choice being noted $\mathbb{P}_U(\mathbf{S}) = |\mathcal{S}_d|^{-1}$, with $|\mathcal{S}_d|$ the space dimension of \mathcal{S}_d as defined in section 4.7.

Remarks

- Hypothesis 4 is the keystone to define a full generative model on the whole covariates \mathbf{X} . On the one hand, the BIC criterion can be applied in this context, avoiding to use a cross-validation criterion which can be much more time-consuming. On the other hand, the great flexibility of Gaussian mixture models [McLachlan and Peel, 2004], provided that the number of components k_j has to be estimated, implies that Hypothesis 4 is particularly weak in fact.
- In practice, Gaussian mixture models are estimated only once for each variable \mathbf{X}^j ($j = 1, \dots, d$). Thus, there is no combinatorial difficulty associated with them. An EM algorithm [Dempster et al., 1977] will be used for estimating the mixture parameters and a classical BIC criterion [Schwarz, 1978] will be used for selecting the different number of components k_j .

6.3.2 Usage of the structure \mathbf{S} for interpretation

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.

6.4 Sub-regression model selection

6.4.1 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 \mathbf{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) that choose a penalty in $\log d$ instead of $\log n$ [Foster and George, 1994] and thus gives more parsimonious models when d 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.

6.4.2 Penalization of the integrated likelihood by $\mathbb{P}(\mathbf{S})$

\mathcal{S}_d being combinatorial, $|\mathcal{S}_d|$ is huge. It has two cumulated consequences: First, the *exact* probability $\mathbb{P}(\mathbf{S}|\mathbf{X})$ may be of the same order of magnitude for a large number of candidates \mathbf{S} , including the best one; Second, the BIC *approximations* of this quantity may introduce additional confusion to wisely distinguish between model probabilities because the number of model compared is not taken into account[Massart and Picard, 2007]. In order to limit this problem, we propose to introduce some information in $\mathbb{P}(\mathbf{S})$ promoting simple models through the following *hierarchical* uniform distribution denoted by $\mathbb{P}_H(\mathbf{S})$:

$$\mathbb{P}_H(\mathbf{S}) = \mathbb{P}_H(\mathbf{J}_r, \mathbf{J}_p) \quad (6.7)$$

$$= \mathbb{P}_H(\mathbf{J}_r, \mathbf{J}_p, d_r, \mathbf{d}_p) \quad (6.8)$$

$$= \mathbb{P}_U(\mathbf{J}_p | \mathbf{d}_p, \mathbf{J}_r, d_r) \times \mathbb{P}_U(\mathbf{d}_p | \mathbf{J}_r, d_r) \times \mathbb{P}_U(\mathbf{J}_r | d_r) \times \mathbb{P}_U(d_r) \quad (6.9)$$

$$= \left[\prod_{j=1}^{d_r} \binom{d - d_r}{d_p^j} \right]^{-1} \times [d - d_r]^{-d_r} \times \left[\binom{d}{d_r} \right]^{-1} \times [d + 1]^{-1}, \quad (6.10)$$

where $\binom{a}{b}$ means the number of b-element subsets of an a-element set and where all probabilities $\mathbb{P}_U(\cdot)$ denote uniform distribution on the related space at hand. $\mathbb{P}_H(\mathbf{S})$ gives decreasing probabilities to more complex models, provided that the following new hypothesis is verified:

Hypothesis 5 We set $d_r < d/2$ and also $d_p^j < d/2$ ($j = 1, \dots, d_r$).

These two thresholds are sufficiently large to be wholly realistic.

Final approximation of $\mathbb{P}(\mathbf{S}|\mathbf{X})$ Merging the previous three expressions, it leads to the following two *global* BIC criteria, to be minimized, denoted by BIC_U or BIC_H , depending on the choice of $\mathbb{P}_U(\mathbf{S})$ or $\mathbb{P}_H(\mathbf{S})$ respectively:

$$\text{BIC}_U(\mathbf{S}) = \text{BIC}_r(\mathbf{S}) + \text{BIC}_f(\mathbf{S}) - 2 \ln \mathbb{P}_U(\mathbf{S}) \quad (6.11)$$

$$\text{BIC}_H(\mathbf{S}) = \text{BIC}_r(\mathbf{S}) + \text{BIC}_f(\mathbf{S}) - 2 \ln \mathbb{P}_H(\mathbf{S}). \quad (6.12)$$

In the following, we will denote by BIC_* any of both BIC_U and BIC_H . Numerical results in Section 7.2 will allow to compare behaviour of both criteria.

Remarks

- As any BIC criterion, the BIC_U and BIC_H criteria are consistent [Lebarbier and Mary-Huard, 2006].
- Even if it favors more parsimonious models, $\mathbb{P}_H(\mathbf{S})$ can be also viewed as a poor informative prior on \mathbf{S} since it is a combination of non informative priors.

6.4.3 Some indicators for proximity

The first criterion is BIC_* which is maximized in the MCMC. But it's value don't have any intrinsic meaning. To show how far the found structure is from the true one in terms of \mathbf{S} we define some indicators to compare the true model \mathbf{S} and the found one $\hat{\mathbf{S}}$. Global indicators :

- $T_r = |\mathcal{J}_r \cap \hat{\mathcal{J}}_r|$ (“True Response”): it corresponds to the number of estimate response covariates in $\hat{\mathbf{S}}$ which are *truly* response covariates in the true model \mathbf{S} .
- $W_r = |\hat{\mathcal{J}}_r| - T_r$ (“Wrong Response”): it corresponds to the number of estimate response covariates in $\hat{\mathbf{S}}$ which are *wrongfully* response covariates in the true model \mathbf{S} .
- $M_r = d_r - T_r$ (“Missing Response”): the number of true response variables not found.
- $\Delta d_r = d_r - \hat{d}_r$: the gap between the number of sub-regression in both model. The sign defines if $\hat{\mathbf{S}}$ is too complex or too simple compared to the true model.
- $\Delta compl = \sum_{j=1}^{d_r} d_p^j$: the difference in complexity between both model.

6.5 The walk

6.5.1 Determinist neighbourhood

We define a *global* neighbourhood space $\mathcal{V}(\mathbf{S})$ of \mathbf{S} composed by the following four *specific* neighbourhood spaces $\mathcal{V}(\mathbf{S}) = \mathcal{V}_{r+}(\mathbf{S}) \cup \mathcal{V}_{r-}(\mathbf{S}) \cup \mathcal{V}_{p+}(\mathbf{S}) \cup \mathcal{V}_{p-}(\mathbf{S})$ described below:

Adding a sub-regression : a new sub-regression with only one predictor covariate is added to \mathbf{S}

$$\mathcal{V}_{r+}(\mathbf{S}) = \left\{ \tilde{\mathbf{S}} : \tilde{\mathbf{S}} \in \mathcal{S}_d, (\tilde{\mathbf{J}}_r, \tilde{\mathbf{J}}_p)^{1, \dots, d_r} = (\mathbf{J}_r, \mathbf{J}_p), \tilde{\mathbf{J}}_r^{d_r+1} \in J_f, \tilde{\mathbf{J}}_p^{d_r+1} = \{j\}, j \in J_f \right\}. \quad (6.13)$$

Removing a sub-regression : a sub-regression is removed from \mathbf{S}

$$\mathcal{V}_{r-}(\mathbf{S}) = \left\{ \tilde{\mathbf{S}} : \tilde{\mathbf{S}} \in \mathcal{S}_d, (\tilde{\mathbf{J}}_r, \tilde{\mathbf{J}}_p) = (\mathbf{J}_r, \mathbf{J}_p)^{\{1, \dots, d_r\} \setminus j}, j \in \{1, \dots, d_r\} \right\}. \quad (6.14)$$

Adding a predictor covariate : a predictor covariate is added to one sub-regression of \mathbf{S}

$$\mathcal{V}_{p+}(\mathbf{S}) = \left\{ \tilde{\mathbf{S}} : \tilde{\mathbf{S}} \in \mathcal{S}_d, \tilde{\mathbf{J}}_r = \mathbf{J}_r, \tilde{J}_p^{\{1, \dots, d_r\} \setminus \{j\}} = J_p^{\{1, \dots, d_r\} \setminus \{j\}}, \tilde{J}_p^j = J_p^j \cup \{h\}, j \in \{1, \dots, d_r\}, h \in J_f \setminus J_p^j \right\}. \quad (6.15)$$

Removing a predictor covariate : a predictor covariate is removed from one sub-regression of \mathbf{S}

$$\mathcal{V}_{p-}(\mathbf{S}) = \left\{ \tilde{\mathbf{S}} : \tilde{\mathbf{S}} \in \mathcal{S}_d, \tilde{\mathbf{J}}_r = \mathbf{J}_r, \tilde{J}_p^{\{1, \dots, d_r\} \setminus \{j\}} = J_p^{\{1, \dots, d_r\} \setminus \{j\}}, \tilde{J}_p^j = J_p^j \setminus \{h\}, j \in \{1, \dots, d_r\}, h \in J_p^j \right\}. \quad (6.16)$$

We just want to find the best model and this neighbourhood is determinist so it does not need to contain the current structure \mathbf{S} .

6.5.2 Transition probabilities

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 $|\mathcal{S}_d|$ rows and columns (combinatory so we just compute the probabilities when we need them). At each step the Markov chain moves with probability:

$$\forall \mathbf{S} \in \mathcal{S}_d, \forall \mathbf{S}^+ \in \mathcal{V}(\mathbf{S}) : \mathcal{P}_{\mathbf{S}, \mathbf{S}^+} = \mathbb{P}(\mathbf{S}^+ | \mathcal{V}(\mathbf{S})) = \frac{\exp(-\text{BIC}_*(\mathbf{S}^+))}{\sum_{\tilde{\mathbf{S}} \in \mathcal{V}(\mathbf{S})} \exp(-\text{BIC}_*(\tilde{\mathbf{S}}))} \quad (6.17)$$

and \mathcal{S}_d 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]. But the walk can also be seen as a Gibbs sampler [Casella and George, 1992] that alternate draws of \mathbf{S} and $\mathcal{V}(\mathbf{S})$ with stationary distribution $\pi \propto \exp(-\text{BIC}_*)$ on the space \mathcal{S}_d .

The output of the algorithm will be the best structure in terms of $P(\mathbf{S} | \mathbf{X})$ 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.

6.5.3 Stochastic neighbourhood

We see that $|\mathcal{V}(\mathbf{S})|$ can be large so the best transition probability may not be far enough from the other to be chosen with a sufficient probability. A classical way to avoid such a problem is to reduce the neighbourhood to a random subset and to allow stationarity (to keep the walk's properties). But then we face a problem: how to choose candidates in a determinist neighbourhood made of several sub-neighbourhood? We would have to index them (time consuming) and then to choose random indexes or to choose a number of candidates to choose from each neighbourhood. One idea would be to choose only one candidate to compare to the current structure, allowing to choose first a sub-neighbourhood and then a candidate. But we observe that each candidate can be described by the way it differs from \mathbf{S} . Then we redefine a neighbourhood based on the modification of the adjacency matrix \mathbf{G} associated to \mathbf{S} .

For each step (q), starting from $\mathbf{S} \in \mathcal{S}_d$ we define a neighbourhood:

$$\mathcal{V}(\mathbf{S}) = \{\mathbf{S}\} \cup \{\mathbf{S}^{(i,j)} \in \mathcal{S}_d | (i, j) \in \mathcal{A}_{(q)}\} \quad (6.18)$$

where $\mathcal{A}_{(q)}$ is a set of couples $(i, j) \in \{1, \dots, d\}^2$ with $i \neq j$ drawn at the step (q) according to a strategy defined below and corresponding to arcs of the graph to modify (add or remove). And we have for $\tilde{\mathbf{S}} = \mathbf{S}^{(i,j)}$:

$$\forall (k, l) \neq (i, j), \quad \tilde{\mathbf{G}}_{k,l} = \mathbf{G}_{k,l} \quad (6.19)$$

$$\tilde{\mathbf{G}}_{i,j} = 1 - \mathbf{G}_{i,j} \quad (6.20)$$

where $\tilde{\mathbf{G}}$ is the adjacency matrix associated to $\tilde{\mathbf{S}}$. Any strategy can be chosen for $\mathcal{A}_{(q)}$, from uniform distribution to specific heuristics.

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.

Here the MCMC is not used for sampling or density estimation. We just want to find the structure with the best value of BIC_* 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.

Strategy to draw $\mathcal{A}_{(q)}$

Many strategies can be imagined. 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 $\mathcal{U}(\{1, \dots, d\})$ and then

$$\mathcal{A}_{(q)}|j = \{(i, j) \in \{1, \dots, d\}^2 : i \neq j\}. \quad (6.21)$$

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 for $\mathcal{A}_{(q)}$ like a fixed number of random couples (i, j) , or the union of the j^{th} line and column of \mathbf{G} . For large values of d , the number of candidate at each step can become critical. Each candidate requires to re-estimate the α^j 's for each modified sub-regression and it requires matricial inversion (estimation by OLS) so computational cost can be high if n is also large.

6.5.4 Active relaxation of constraints

In practice, for some of the $(i, j) \in \mathcal{A}_{(q)}$, we have $\mathbf{S}^{(i,j)} \notin \mathcal{S}_d$ because of the uncrossing rule. 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 this uncrossing rule. Last but not least, even if the walk is regular, local extrema may significantly slow down the research of the optimal structure. Thus we propose a constraints relaxation method by a new definition of $\tilde{\mathbf{S}} = \mathbf{S}^{(i,j)}$ relying on a new definition of $\tilde{\mathbf{G}}$:

Modification of the selected arc (i, j) on the graph: (Figure 6.2)

$$\tilde{\mathbf{G}}_{i,j} = 1 - \mathbf{G}_{i,j} \text{ as usual and} \quad (6.22)$$

$$\forall k \neq i, l \neq j, \quad \tilde{\mathbf{G}}_{k,l} = \mathbf{G}_{k,l} \quad (6.23)$$

Column-wise relaxation : newly predictive covariate cannot be regressed anymore (Figure 6.3):

$$\forall k \in \{1, \dots, d\} \setminus \{i\}, \tilde{\mathbf{G}}_{k,j} = \mathbf{G}_{i,j} \mathbf{G}_{k,j} \quad (6.24)$$

$$(6.25)$$

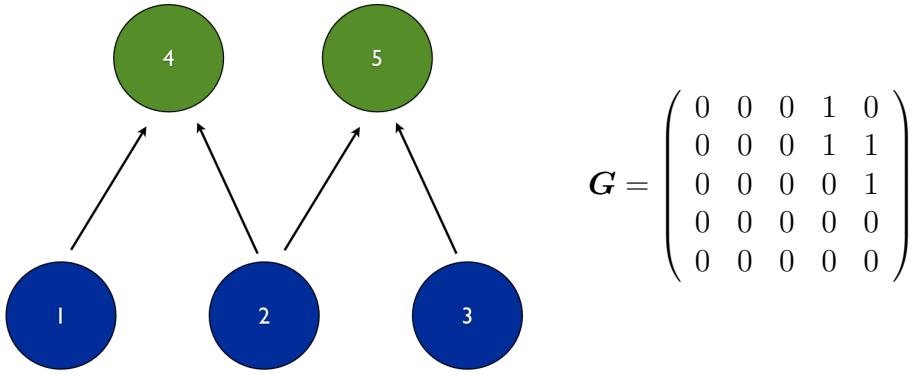


Figure 6.1: We start from a structure $\mathcal{S} = ((4, 5), (\{1, 2\}, \{2, 3\}))$ and its associated matrix G

Row-wise relaxation: newly regressed covariate cannot be predictive anymore (Figure 6.4):

$$\forall l \in \{1, \dots, d\} \setminus \{j\}, \tilde{\mathbf{G}}_{i,l} = \mathbf{G}_{i,j} \mathbf{G}_{i,l} \text{ (row-wise relaxation)} \quad (6.26)$$

It can be seen as forcing the modification by removing what would have made 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 (that was not possible with the determinist neighbourhood) and create a new pairwise sub-regression. It drastically increases the scope of the neighbourhood (Figure 6.6) 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 6.10), 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).

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 7.20).

We give an example to better illustrate how it works in Figures 6.1 to 6.6:

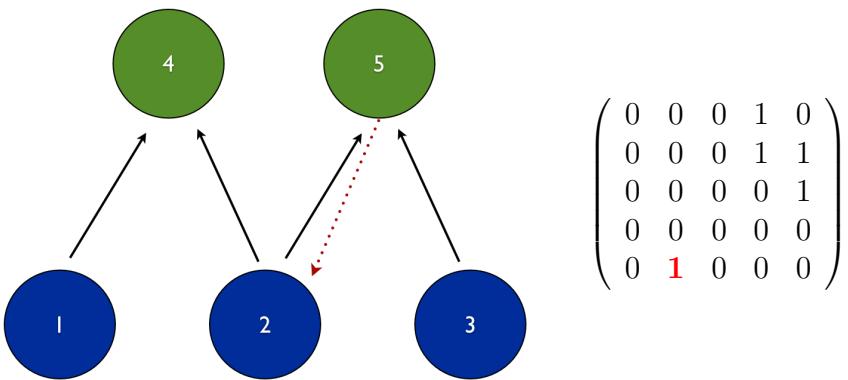


Figure 6.2: We want to define the candidate $\tilde{\mathbf{S}} = \mathbf{S}^{(5,2)}$ and its associated matrix but the structure obtained would not be feasible (breaking the uncrossing rule).

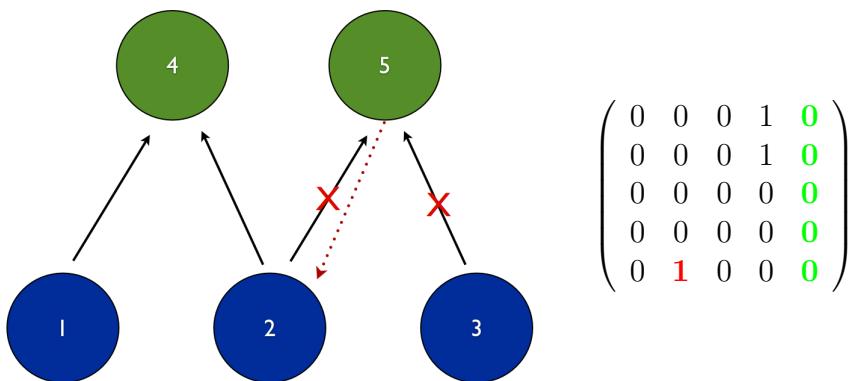


Figure 6.3: Column-wise relaxation: newly predictive covariate cannot be regressed anymore.

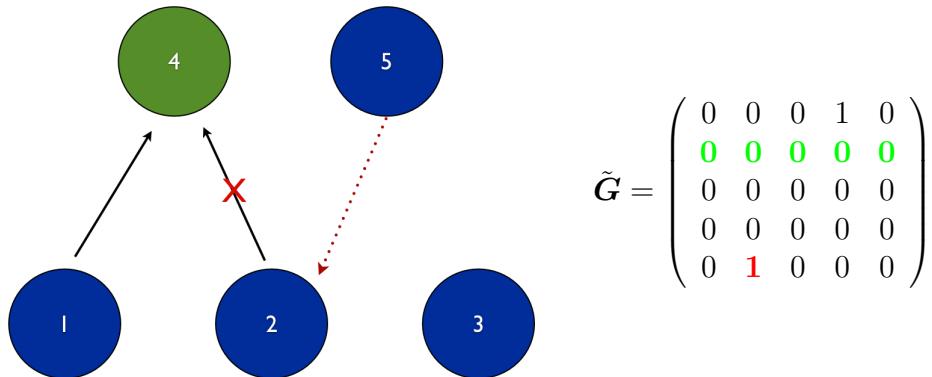


Figure 6.4: Row-wise relaxation: newly predictive covariate cannot be regressed anymore.

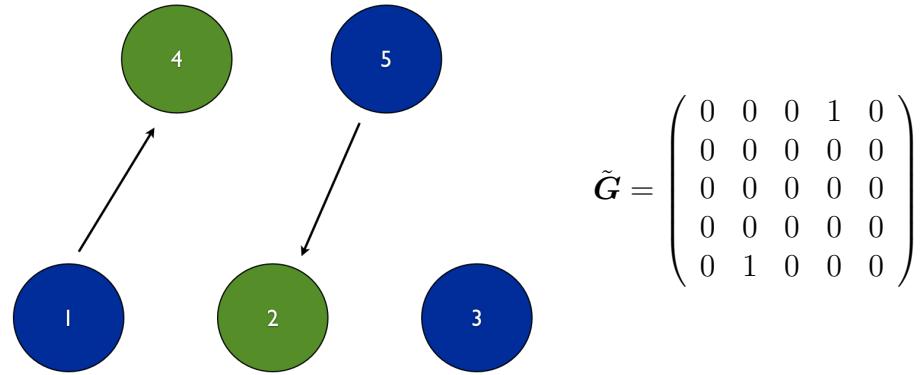


Figure 6.5: We get a feasible candidate $\tilde{\mathbf{S}} = ((2, 4), (\{5\}, \{1\}))$ that does differ from \mathbf{S} in many points.

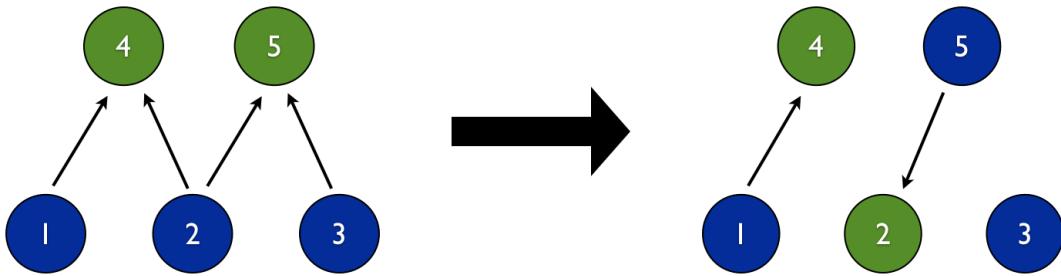


Figure 6.6: All this modifications are made in only one step in the MCMC, meaning an increased scope for the neighbourhoods.

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 set to 1 into \mathbf{G} , 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 ordering of the Bernoulli drawing so we can draw with a random order or by ordering by descending correlations.

We note then that the BIC_* associated to initial model is often worse than the BIC_* of the void structure, so we compare several chains in Figures 6.7 and 6.8. We see that correlation-based initialization quickly beat the void structure. This can be explained by local extrema. Correlations-based initialization gives structures with a smaller "structural distance" to the true model and then the chains are less subject to local extrema.

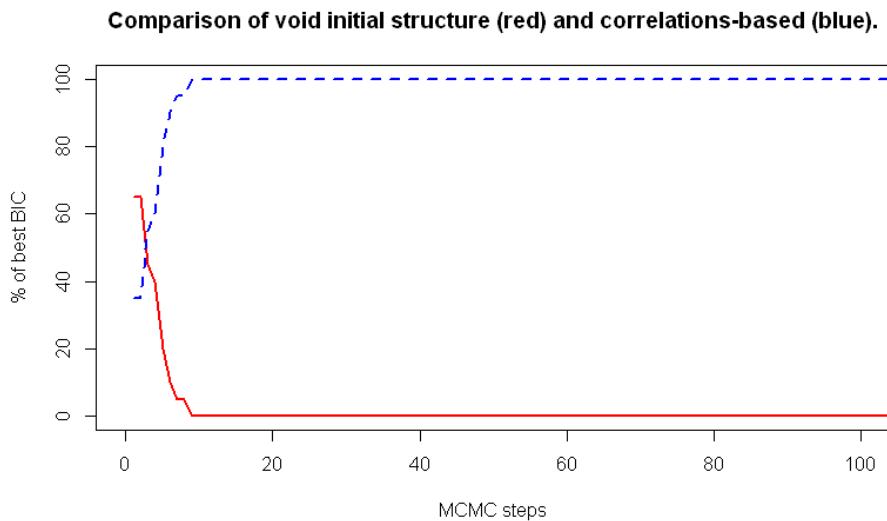


Figure 6.7: Amount of time each method is better for the 100 first steps of the MCMC.

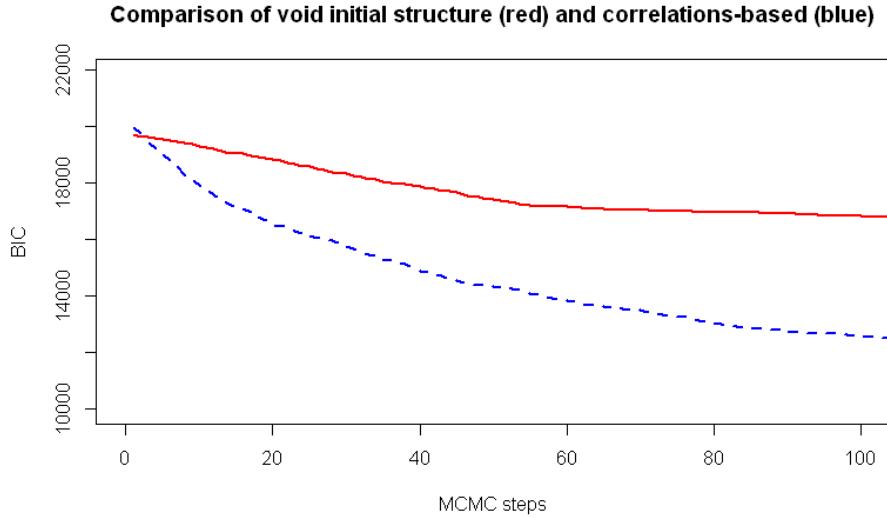


Figure 6.8: Evolution of the BIC (criterion to minimize in the MCMC) for each method.

6.6.2 Multiple initialization

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 6.9 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 at least 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.

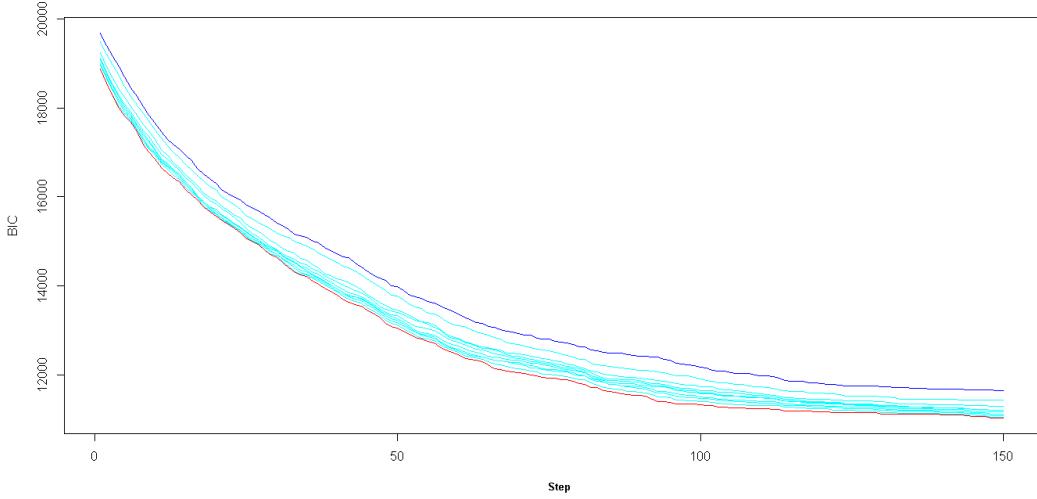


Figure 6.9: Comparison of distinct number of correlation-based initialisations for the MCMC.

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

6.7 Pruning

If the complexity of \mathbf{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 $\mathbf{X}^{J_p^j}$ to estimate the coefficients $\boldsymbol{\alpha}_j$ and obtain some supplementary zeros. Working on $\mathbf{X}^{J_p^j}$ 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 hypotheses

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 in practice (does not run enough steps), 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:

$$\mathcal{A}_{(q)} = \{(i, J_r^j), i \in J_f, j \in \{1, \dots, d_r\} : i \in J_p^j\}. \quad (6.27)$$

It is not based on any arbitrary parameter and change the result only if it finds a better structure in terms of the criterion BIC_* 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 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 \mathbf{X} and we search an oriented graph.

However, we can still use it for initialization, for example by a Hadamard product with $\mathbf{G}^{(0)}$ 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 (especially for an initialisation method) and has no theoretical properties relied to our problematic of minimizing BIC_* , 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 \mathbf{S} as the number of elements in the adjacency matrix, that is the number of links between covariates and is obtained by:

$$\text{Complexity}(\mathbf{S}) = \sum_{j=1}^{d_r} d_p^j. \quad (6.28)$$

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

For Figures 6.10 and 6.11 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.

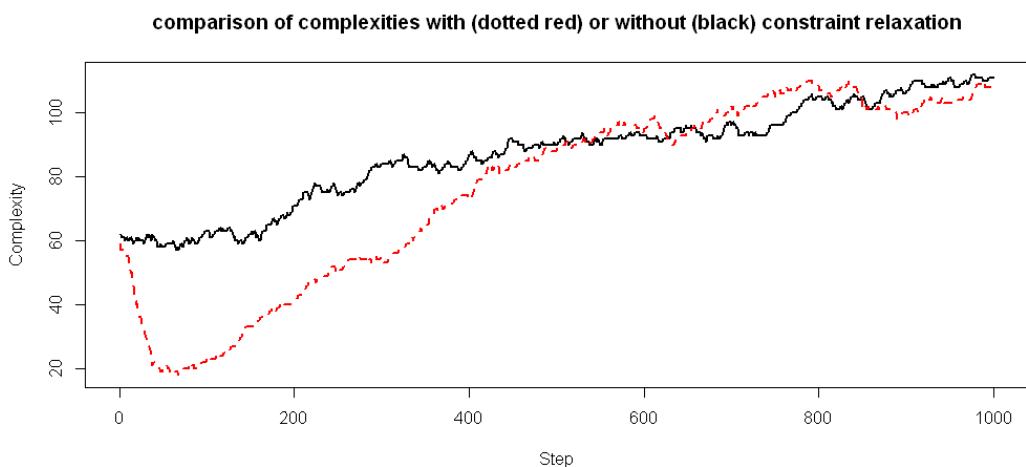


Figure 6.10: Comparison of complexity evolution with or without constraint relaxation.

comparison of BIC with (dotted red) or without (black) constraint relaxation

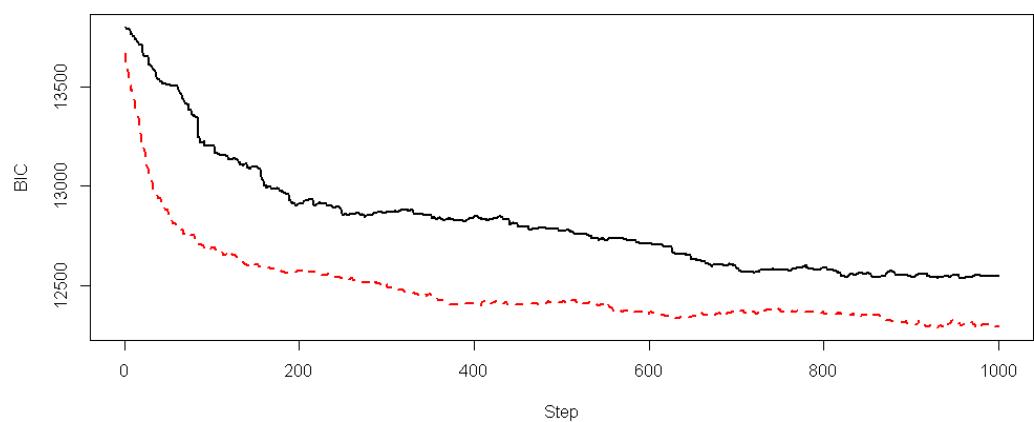


Figure 6.11: Comparison of BIC evolution with or without constraint relaxation.

Chapter 7

Numerical results on simulated datasets

7.1 Simulated 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 package has been tested on the simulated datasets from section 5.1. Section 7.2 shows the results obtained in terms of $\hat{\mathbf{S}}$. 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 $\hat{\mathbf{Y}}$ were computed on a validation sample of 1000 individuals. Several pattern for \mathbf{Y} were tested to evaluate the impact of irrelevant covariates.

We used the package **Rmixmod** from CRAN [Auder et al., 2014] 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 $n < d$, a frequently used method is the Moore-Penrose generalized inverse [Katsikis and Pappas, 2008], thus OLS can obtain some results even with $n < d$. We compare different methods with and without **CorReg** as a pretreatment. All the results are provided by the **CorReg** package. Associated figures will display both mean and inter-quartile intervals.

7.2 Results on $\hat{\mathbf{S}}$

Figure 7.1 illustrates the impact of large samples. For $n \gg d$ the MCMC found 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_U to BIC_H it becomes evident that BIC_H 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_H 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).
- BIC 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.

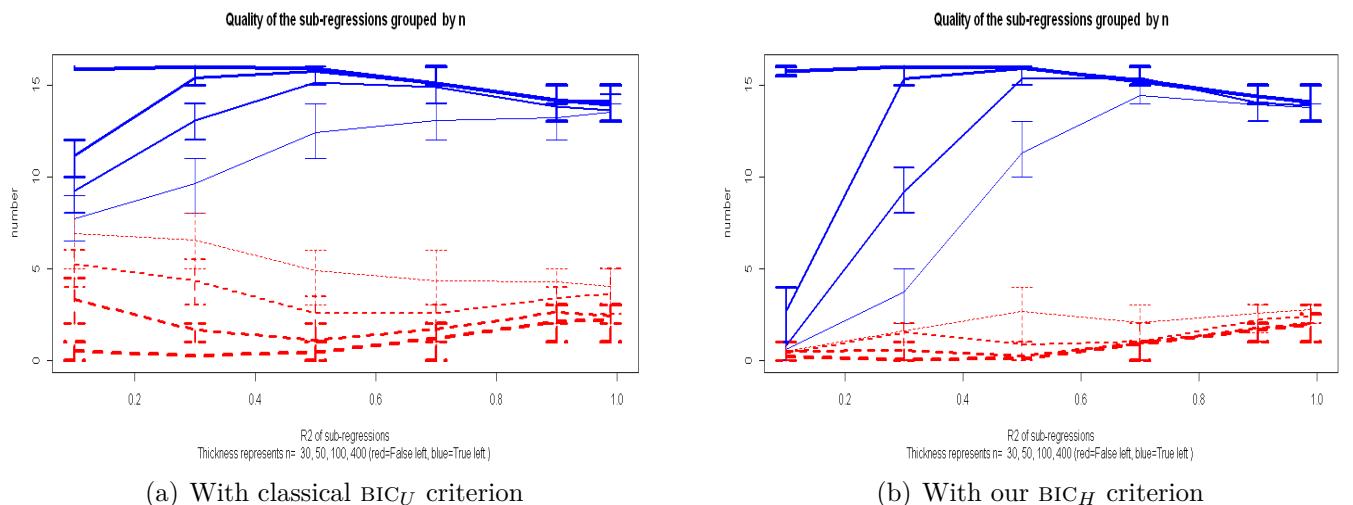


Figure 7.1: Quality of the subregressions found by the MCMC. True left (plain blue) and Wrong left (dotted red) for n varying in $(30, 50, 100, 400)$, the thicker the greater n .

7.2.1 Comparison with Selvarclust

Maugis provides results¹ of SelvarClust on a dataset *Data2.txt* containing 2 000 data points from a mixture of four Gaussian distributions $\mathcal{N}(\mu_k, \mathbf{I}_2)$ with $\mu_1 = (-2, -2)$, $\mu_2 = (-3, -3)$, $\mu_4 = (2, 2)$ and with a proportion vector $\boldsymbol{\pi} = (0.3, 0.2, 0.3, 0.2)$. Variables $\mathbf{X}^1, \mathbf{X}^2$ corresponds to the coordinates. Eight irrelevant variables (for clustering) are appended, simulated according to $\forall 1 \leq i \leq 2\,000$:

$$\mathbf{X}_i^{\{3, \dots, 6\}} | \mathbf{X}_i^{\{1, 2\}}, \boldsymbol{\alpha}^*, \boldsymbol{\Omega} \sim \mathcal{N}(\mathbf{X}_i^{\{1, 2\}} \boldsymbol{\alpha}^*, \boldsymbol{\Omega}) \text{ with } \boldsymbol{\alpha}^* = \begin{pmatrix} 0.5 & 0 & 2 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix} \text{ and } \boldsymbol{\Omega} = \text{diag}(1, 1, 0.5, 0.5)$$

$$\mathbf{X}_i^{\{7, \dots, 10\}} \sim \mathcal{N}(0, \mathbf{I}_4).$$

¹<http://www.math.univ-toulouse.fr/~maugis/SelvarClustHomepage.html>

We compare results obtained on $\hat{\boldsymbol{\alpha}}^*$ by both **CorReg** and **Selvarclust**. Both **SelvarClust** and **CorReg** add an intercept (default parameter) that is like using a constant covariate $\mathbf{X}^0 = \mathbf{1}$ and then adding a first row to $\boldsymbol{\alpha}^*$ containing the intercept of the sub-regressions. This intercept is not part of the selection procedure so it will never have zero value in practice. Hence we compare the second and third lines of the matrices (an horizontal line is drawn to distinguish the intercept from the rest of the matrix).

SelvarClust finds $\hat{\boldsymbol{\alpha}}_{Selvarclust}^* =$

$$\left(\begin{array}{ccccccccc} 0.006052 & -0.025386 & -0.006845 & -0.015952 & 0.003420 & 0.007839 & -0.047422 & -0.005811 \\ \hline \mathbf{0.504791} & -0.002147 & \mathbf{2.007127} & -0.001010 & -0.000105 & 0.022403 & 0.013361 & 0.000083 \\ -0.006709 & \mathbf{1.000927} & 0.007463 & \mathbf{2.997941} & -0.005955 & -0.021958 & -0.010387 & 0.010765 \end{array} \right)$$

CorReg finds:

$$\hat{\boldsymbol{\alpha}}_{CorReg}^* = \left(\begin{array}{cccc} 0.008698209 & -0.02540033 & -0.00978779 & -0.01595849 \\ \hline \mathbf{0.504672402} & 0 & \mathbf{2.00725861} & 0 \\ 0 & \mathbf{1.00088836} & 0 & \mathbf{2.99792339} \end{array} \right)$$

Both software found that $\mathbf{X}^1, \mathbf{X}^2$ are not in $\mathbf{J}_r = (3, \dots, 6)$ but only **CorReg** found the true model with $\hat{\mathbf{J}}_r = \mathbf{J}_r$ and $\hat{\mathbf{J}}_p = \mathbf{J}_p = (\{1\}, \{2\}, \{1\}, \{2\})$. **Selvarclust** finds $\hat{\mathbf{S}} = ((3, \dots, 10), (\{1, 2\}, \dots, \{1, 2\}))$ that is it only finds the partition (J_r, J_f) . Our algorithm gives better results on $\hat{\boldsymbol{\alpha}}^*$ with more parsimonious $\hat{\mathbf{S}}$ (that is the true \mathbf{S}), proving that a new algorithm was needed to estimate the sub-regression structure and that the proposed MCMC is efficient.

CorReg gives better results because it does not suffer from correlations like stepwise does in the algorithm from Maugis and also because the goal is not the same.

Selvarclust aims to find the relevant covariates for clustering $(\mathbf{X}^1, \mathbf{X}^2)$ and achieves this goal. We focus on the explicit structure of regression between the covariate instead of Gaussian clustering. Distinct goals, distinct results even if the sub-regression model is quite the same. Moreover **Selvarclust** does not make the same hypotheses on the dataset and allows dependencies between the regressors. We suppose independence between the regressors and between the conditional distributions so we only estimate the univariate marginal distribution and we make it only once. Then, at each step **Selvarclust** has to estimate the whole distribution. Our algorithm does not allow to find complex distributions with dependencies as **Selvarclust** does. Both models use a sub-regression structure and propose an algorithm that find it, but the comparison stops here. So **CorReg** does not really beats **Selvarclust** because hypotheses and objectives are not the same, so the result cannot really be compared. **Selvarclust** is not a competitor (it could have been but it is not the case) to **CorReg** but a source of inspiration for the concept of sub-regressions within the covariates and the concept of redundant and irrelevant covariates. To obtain better results was just a necessary confirmation of that.

7.2.2 Computational time

In terms of computation time, the most expansive thing in the MCMC is the computation of OLS to estimate $\boldsymbol{\alpha}$ for each candidate, that is mainly successive inversion of the $(\mathbf{X}^{J_p^j})' \mathbf{X}^{J_p^j}$. Figure 7.2 illustrate the evolution of the time needed to achieve 10 times (distinct initializations) 1 000 steps on the datasets described above with the R^2 of the sub-regressions set to 0.7. We see that time increases with n from less than 2 seconds for $n = 30$ to more than 12 seconds for $n = 400$.

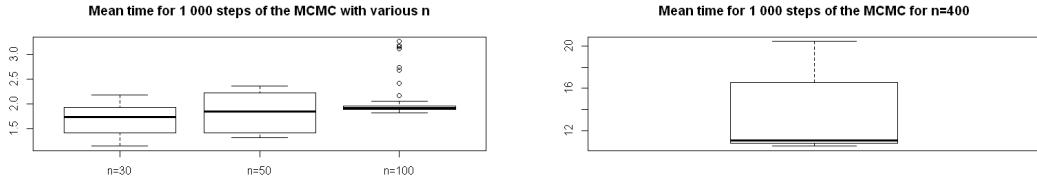


Figure 7.2: Evolution of the mean time for 10 times 1 000 steps of the MCMC, 39 candidates each time (constraint relaxation). $d = 40$ covariates with $d_r = 16$ sub-regressions

The main impact of a change of d will be the expansion of \mathcal{S}_d that will require more iteration/candidates to find the true structure. But the cost of each sub-regression remains the same. Figure 7.3 show the evolution of the time for $n = 30$ and $d_r = 0.4 \times d$ sub-regressions with $R^2 = 0.7$ each time. 40 candidates were tested each time. We first observe that the increase seems slower for small values of d it is because of the amount of time needed for initializations and other annex steps, when d rises they become less significant and it reveals the non-linear time increase. For $d = 1 000$ time raises up to 26 minutes. This time could be reduced by using sparse matrices instead of full binary matrices \mathbf{G} , it would also be more efficient with memory usage. Another way to reduce this time is to compute each initializations in parallel, dividing this time by nearly 10.

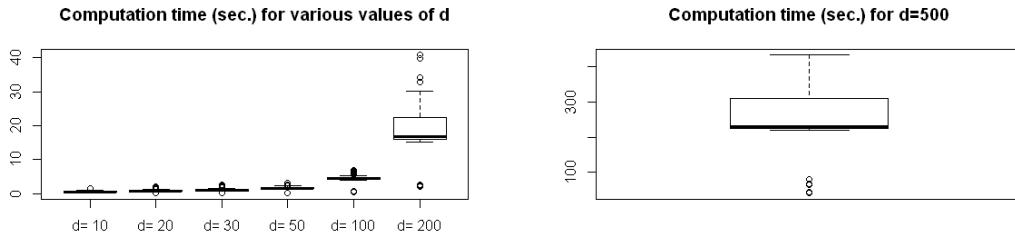


Figure 7.3: Evolution of the mean time for 10 times 1 000 steps of the MCMC, 40 candidates each time (constraint relaxation). $d_r = 0.4 \times d$ sub-regressions, $n = 30$.

Without constraint relaxation (Figure 7.4) the number of candidates evolves at each step, reducing computational cost (but also convergence speed) and we know that only one sub-regression is changed compared to the current so we only have to compute one sub-regression (some candidates requires to compute several sub-regressions with constraint relaxation) reducing again the time of each step. Hence, rejecting candidates is a good way to achieve quickly a fixed number of step, so it could be used as a warming phase before using constraint relaxation to avoid local extrema. All the results were obtained on a laptop with intel(R) core(TM) i5 CPU with 2.4GHz and 2Go of RAM running Windows XP.

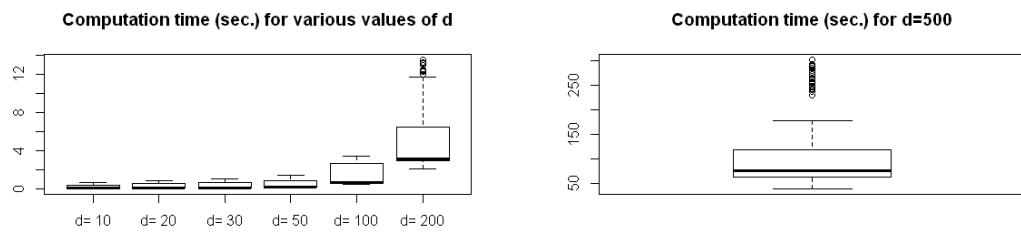


Figure 7.4: Evolution of the mean time for 10 times 1 000 steps of the MCMC, 40 candidates each time (without constraint relaxation). $d_r = 0.4 \times d$ sub-regressions, $n = 30$.

7.3 Results on prediction

7.3.1 \mathbf{Y} depends on all variables in \mathbf{X}

We try the method with a response variable depending on all covariates to compare the results with those from section 5.1 (same \mathbf{X} and \mathbf{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 $\hat{\mathbf{S}}$ are still good so the MCMC is efficient enough to be useful for the study of the response variable \mathbf{Y} . Results are mostly the same than when using the true structure.

Results for OLS (Figure 7.5) are similar to those from section 5.1 excepted for small correlations because the MCMC using BIC_H 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 7.6 to 7.8) where covariates not deleted by the structure are deleted by the variable selection.

Ridge regression results (Figure 7.9) is also very similar to the previous (Figure 5.5).

Having simpler structure is important for interpretation so we keep the choice of using BIC_H , but users of `CorReg` can use classical BIC_U with a single boolean parameter change.

Ordinary Least Squares when Y depends on all variables in X

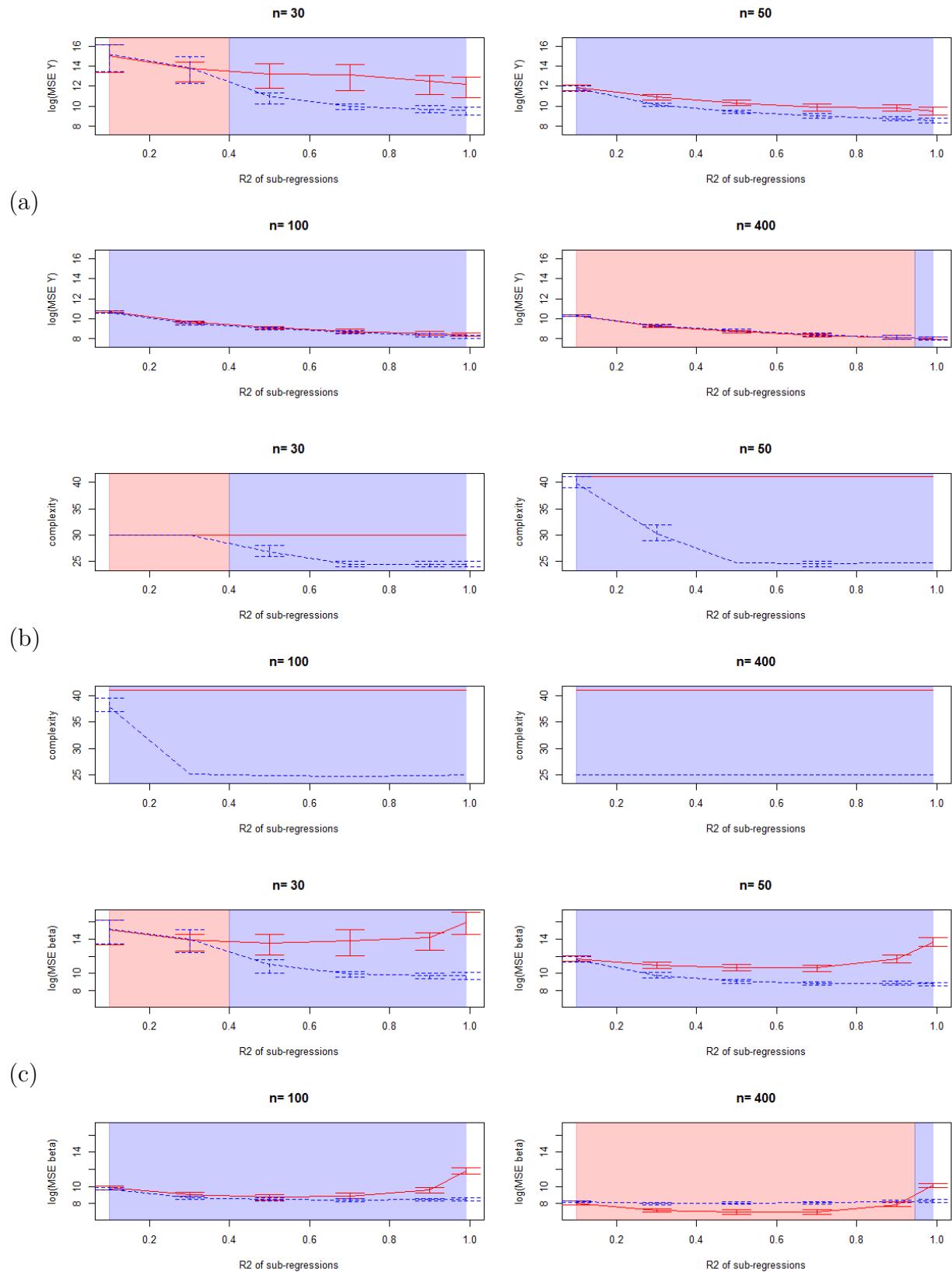


Figure 7.5: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

LASSO when Y depends on all variables in X

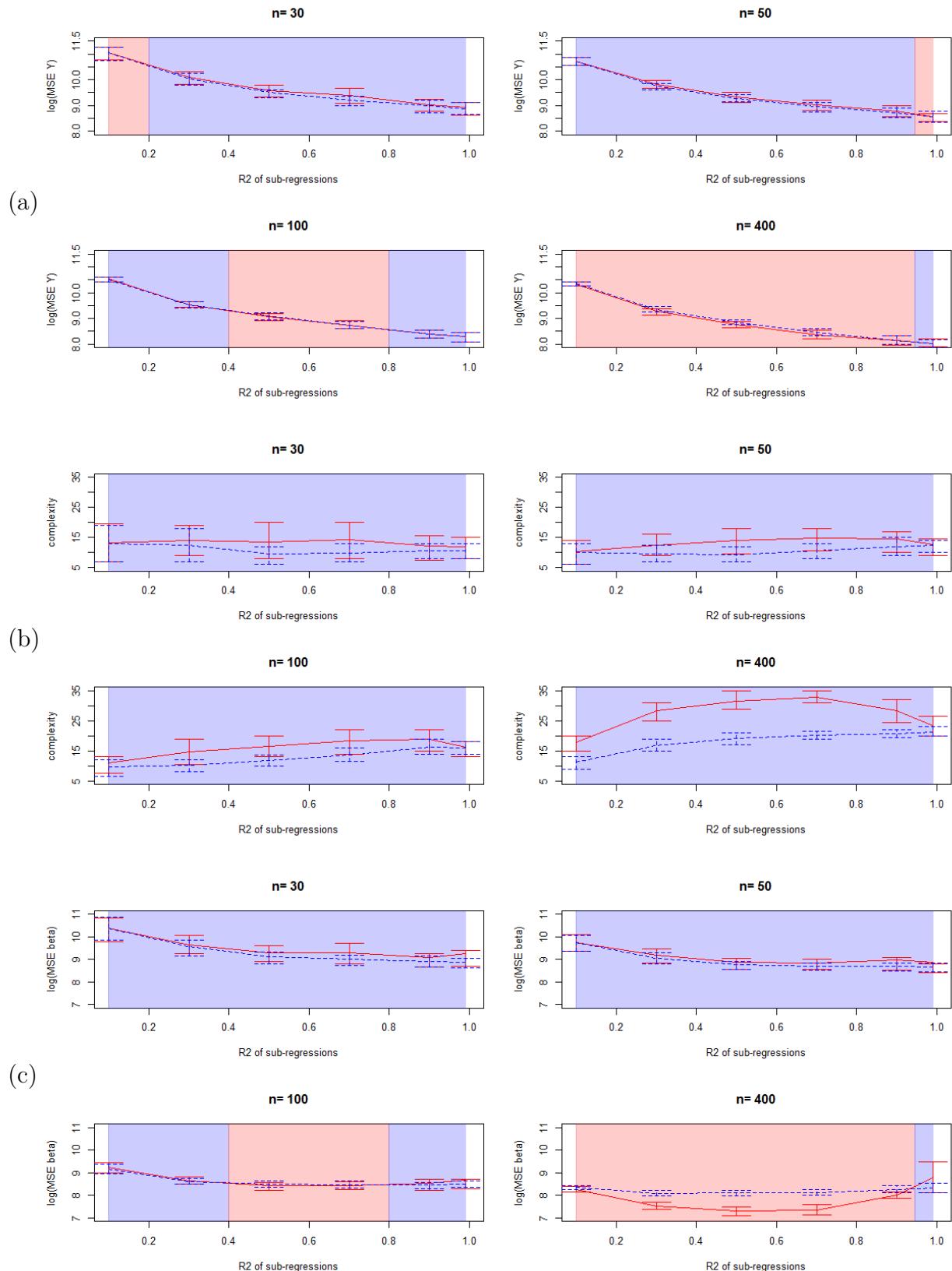


Figure 7.6: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when \mathbf{Y} depends on all variables in \mathbf{X}

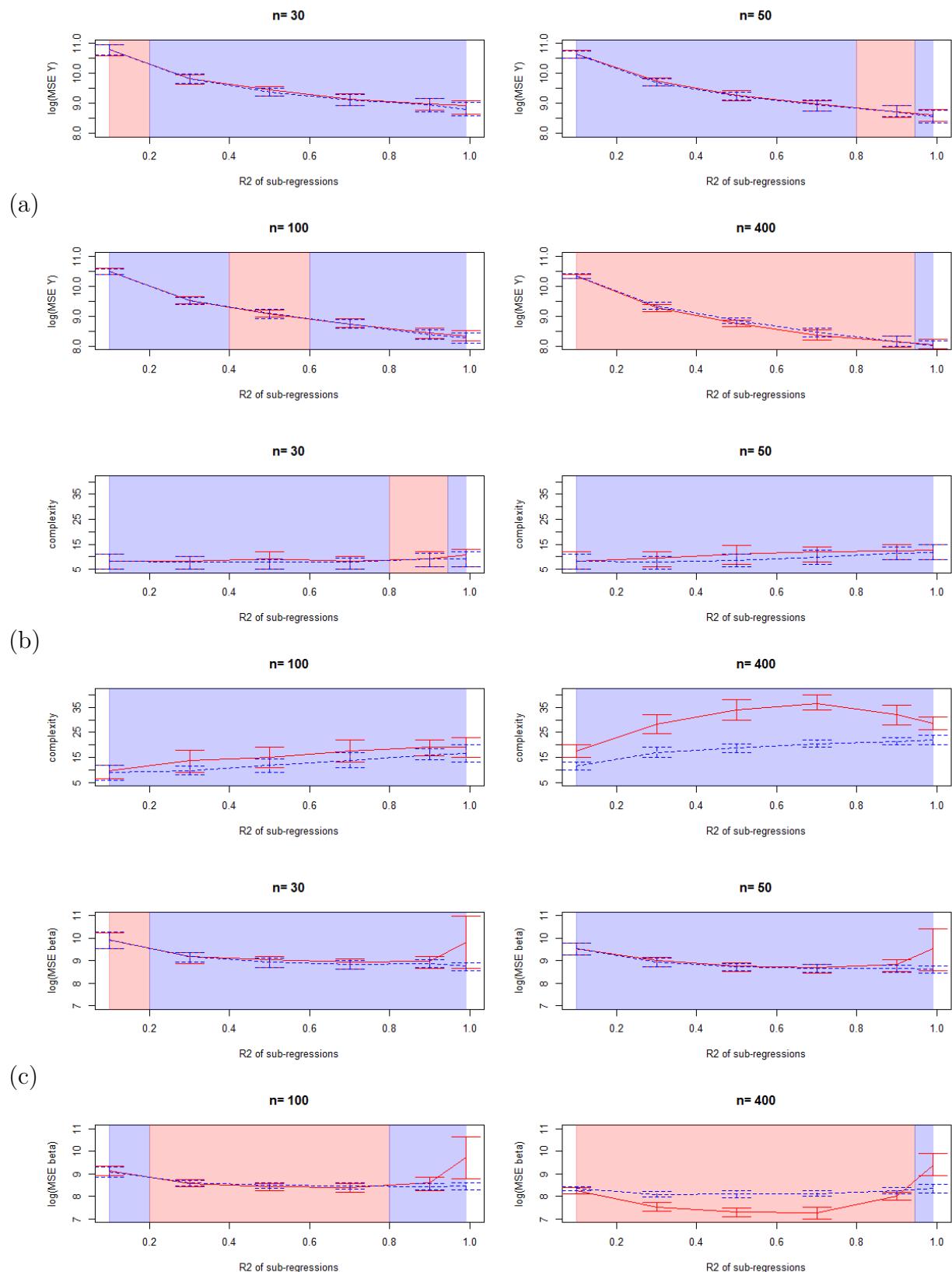


Figure 7.7: Comparison of the MSE on $\hat{\mathbf{Y}}$ (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends on all variables in X

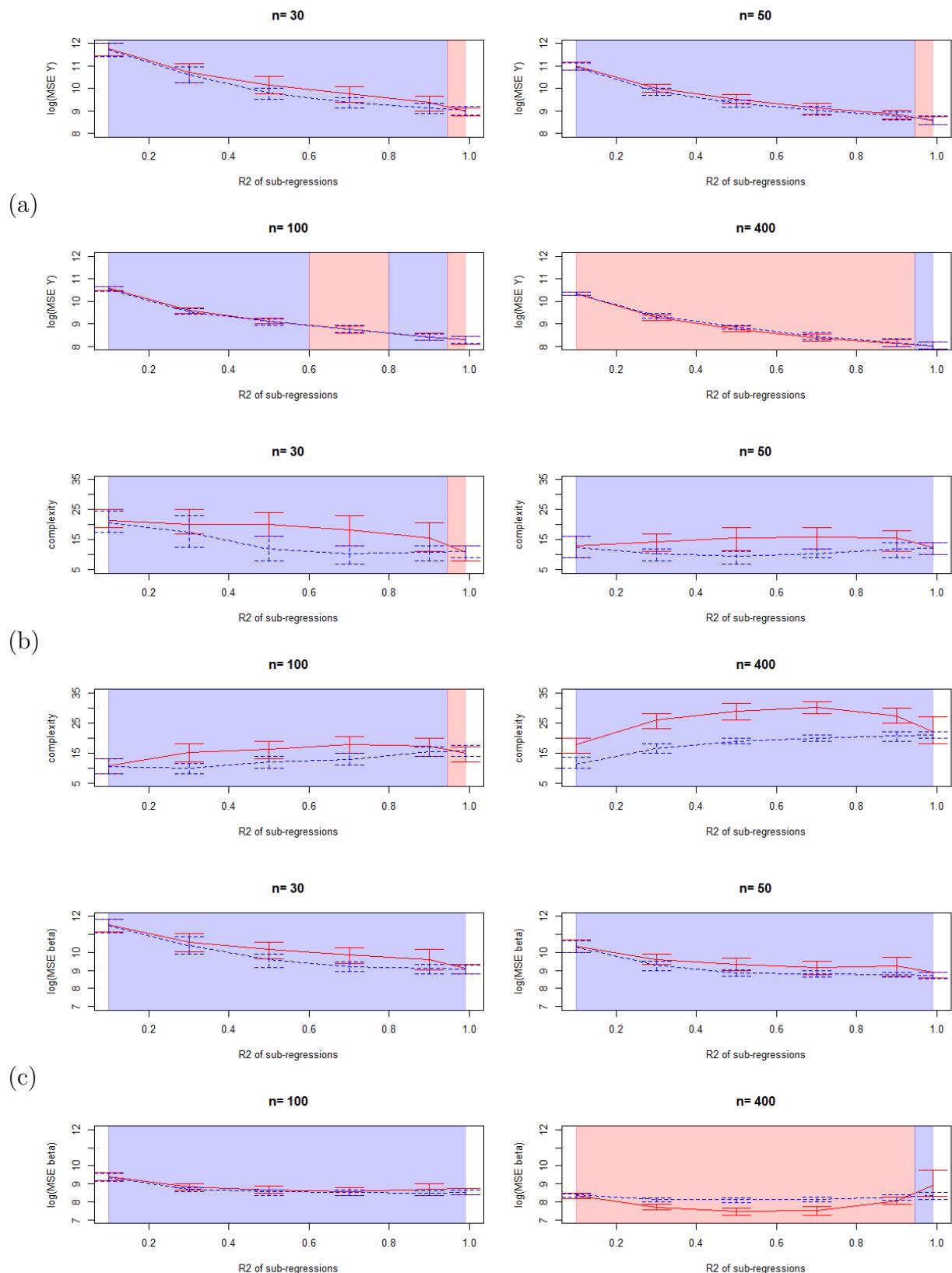


Figure 7.8: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when Y depends on all variables in X

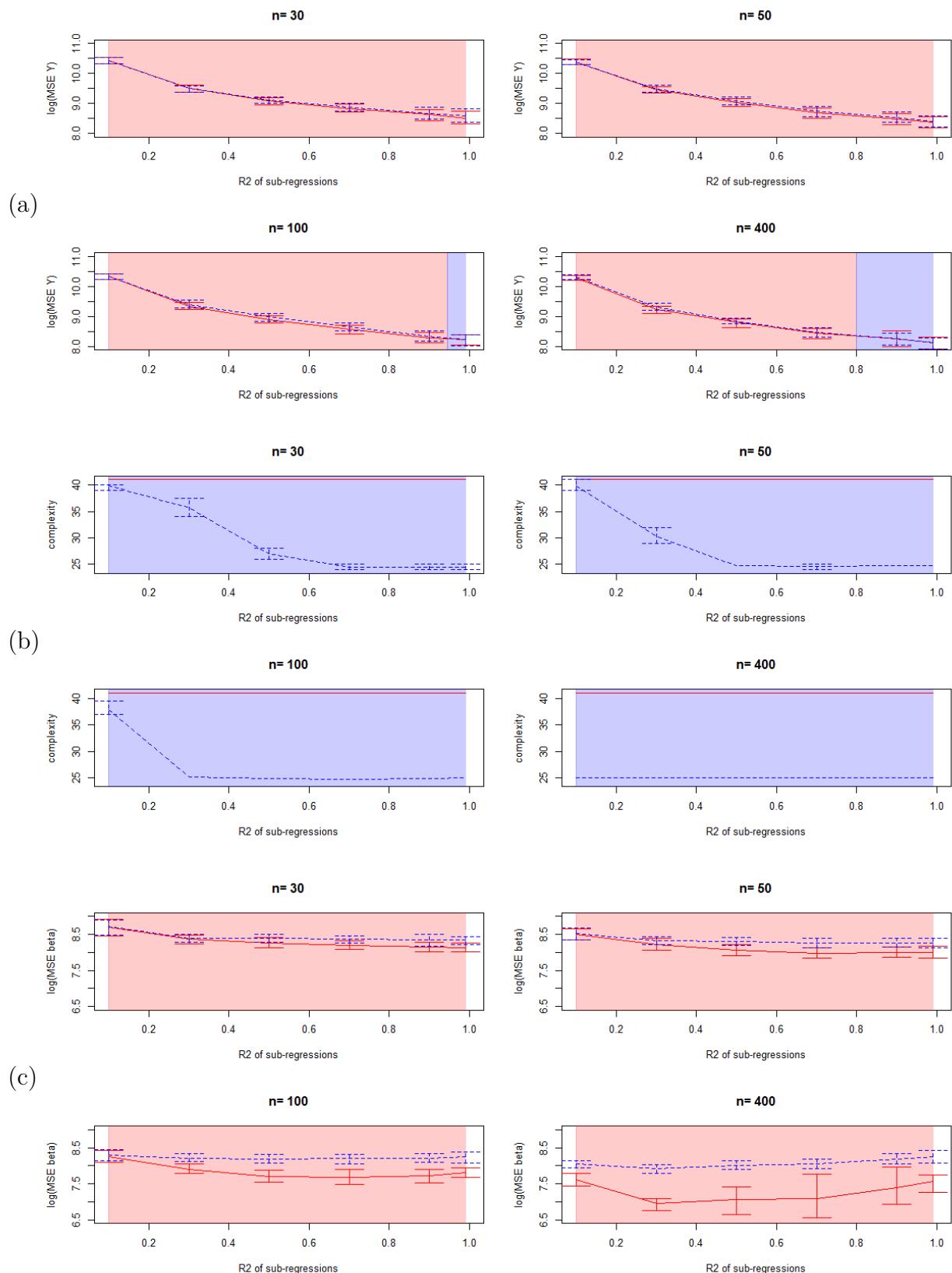


Figure 7.9: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

7.3.2 \mathbf{Y} depends only on covariates in \mathbf{X}_f (best case for us)

We want to see what happens in real life, when some covariates are irrelevant to describe \mathbf{Y} according to the given dataset. We could generate \mathbf{Y} with a random subset of \mathbf{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 \mathbf{X}_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 \mathbf{X} .

We start with \mathbf{Y} depending only on covariates in \mathbf{X}_f . It is the best case for us because our marginal model is then the true model (when the true structure is found) and the complete model will need variable selection to reach the truth. Here \mathbf{Y} depends on the 24 covariates in \mathbf{X}_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.

Ordinary Least Squares: In figure 7.10 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.

Variable selection: When looking at variable selection methods (Figures 7.11 to 7.13)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 \mathbf{Y} depending on the whole dataset \mathbf{X} .

Ridge regression: Ridge regression (figure 7.14) 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 \mathbf{Y} depends on all covariates in \mathbf{X}_f . Our pretreatment is limited in terms of variable selection.

Ordinary Least Squares when Y depends only on covariates in X_f

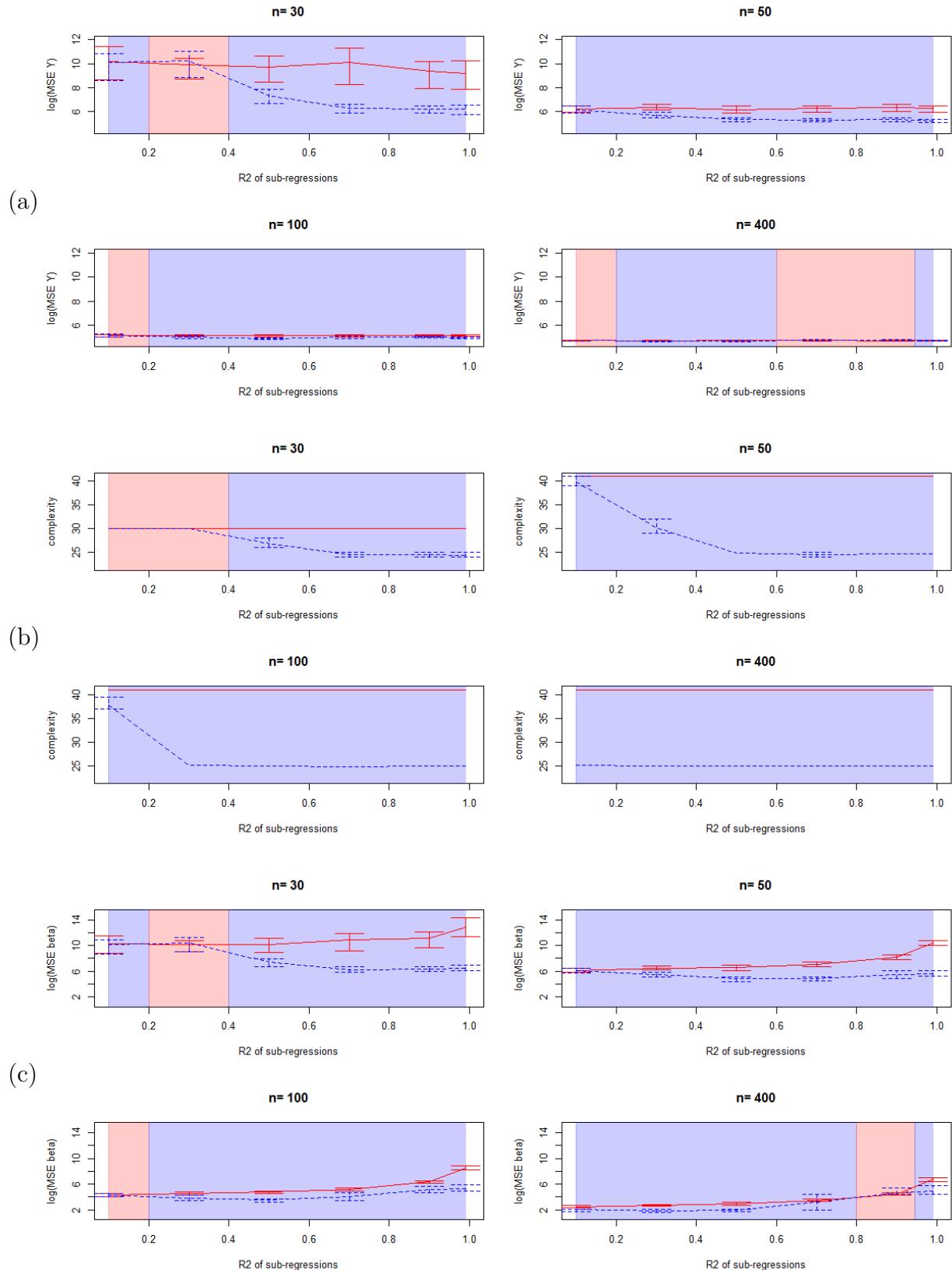


Figure 7.10: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

LASSO when Y depends only on covariates in X_f

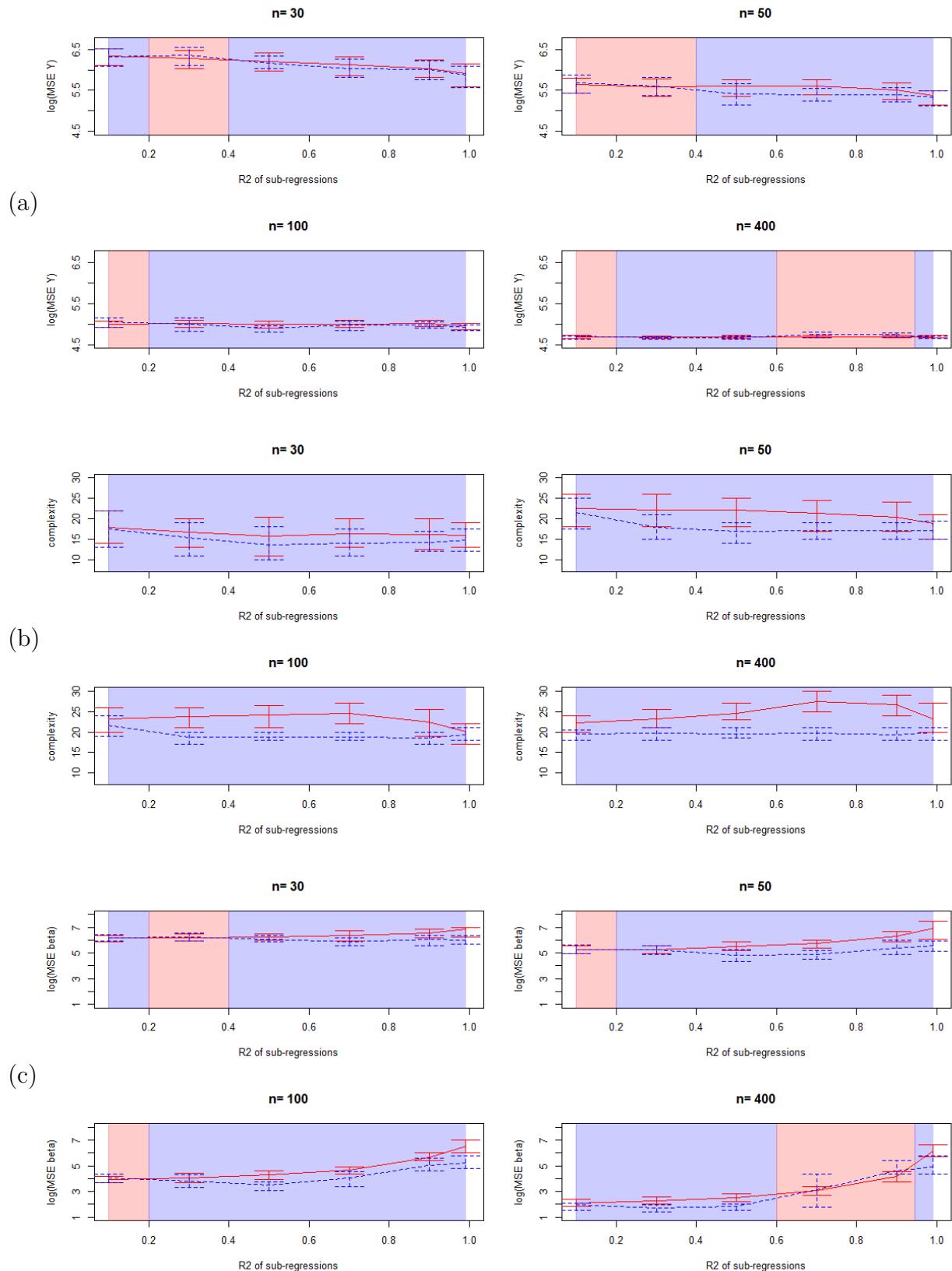


Figure 7.11: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when \mathbf{Y} depends only on covariates in \mathbf{X}_f

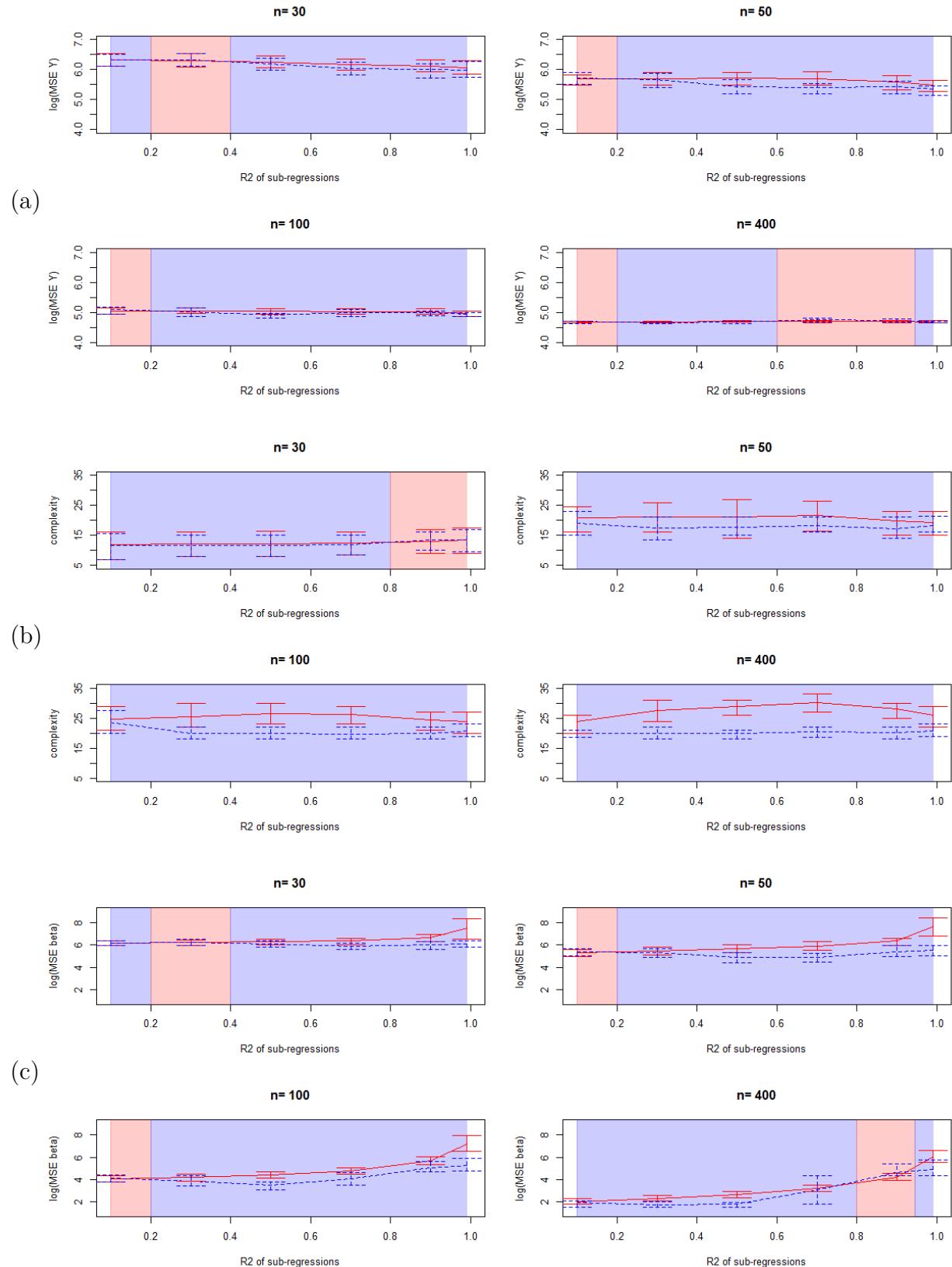


Figure 7.12: Comparison of the MSE on $\hat{\mathbf{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends only on covariates in X_f

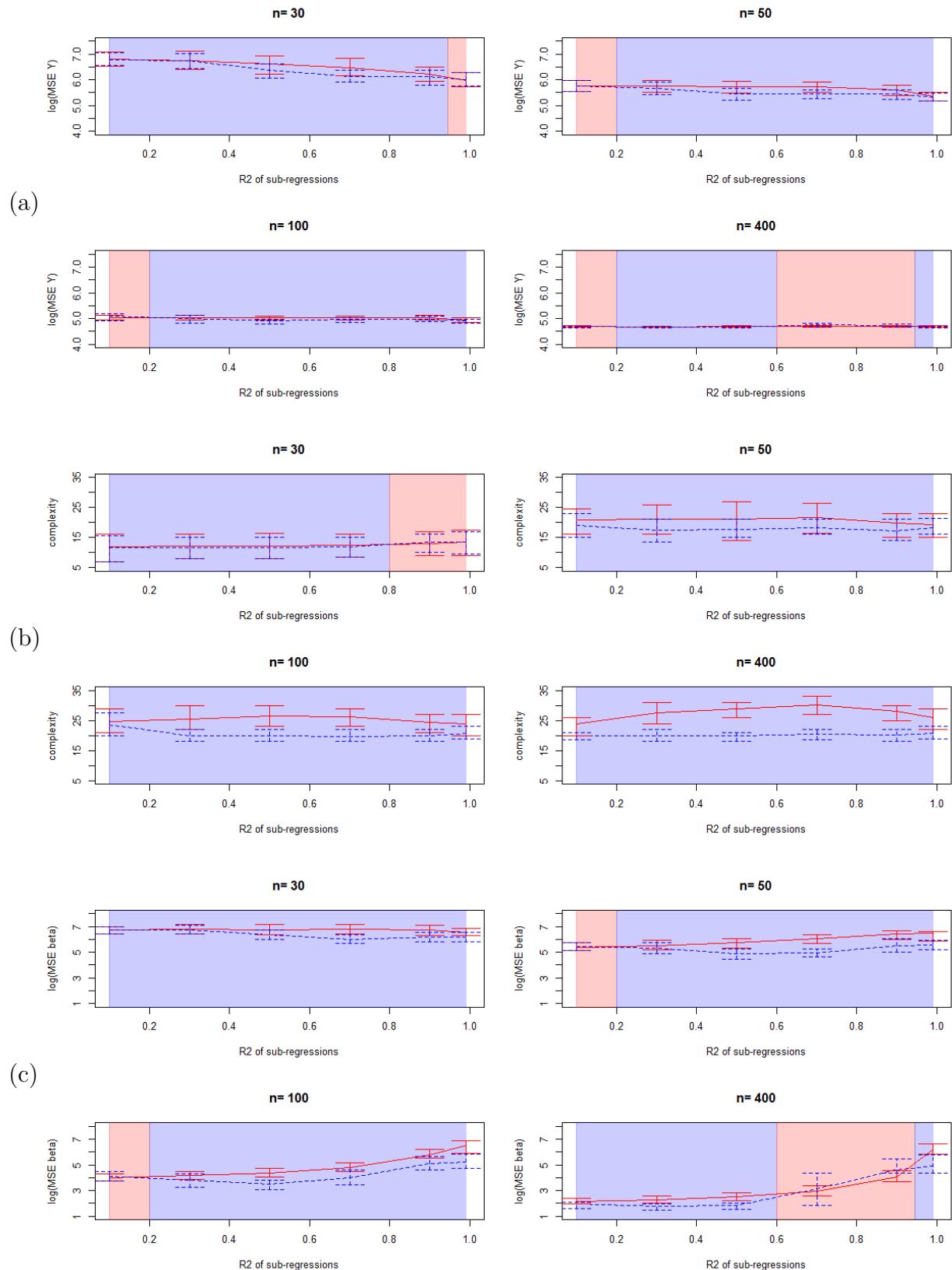


Figure 7.13: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when Y depends only on covariates in X_f

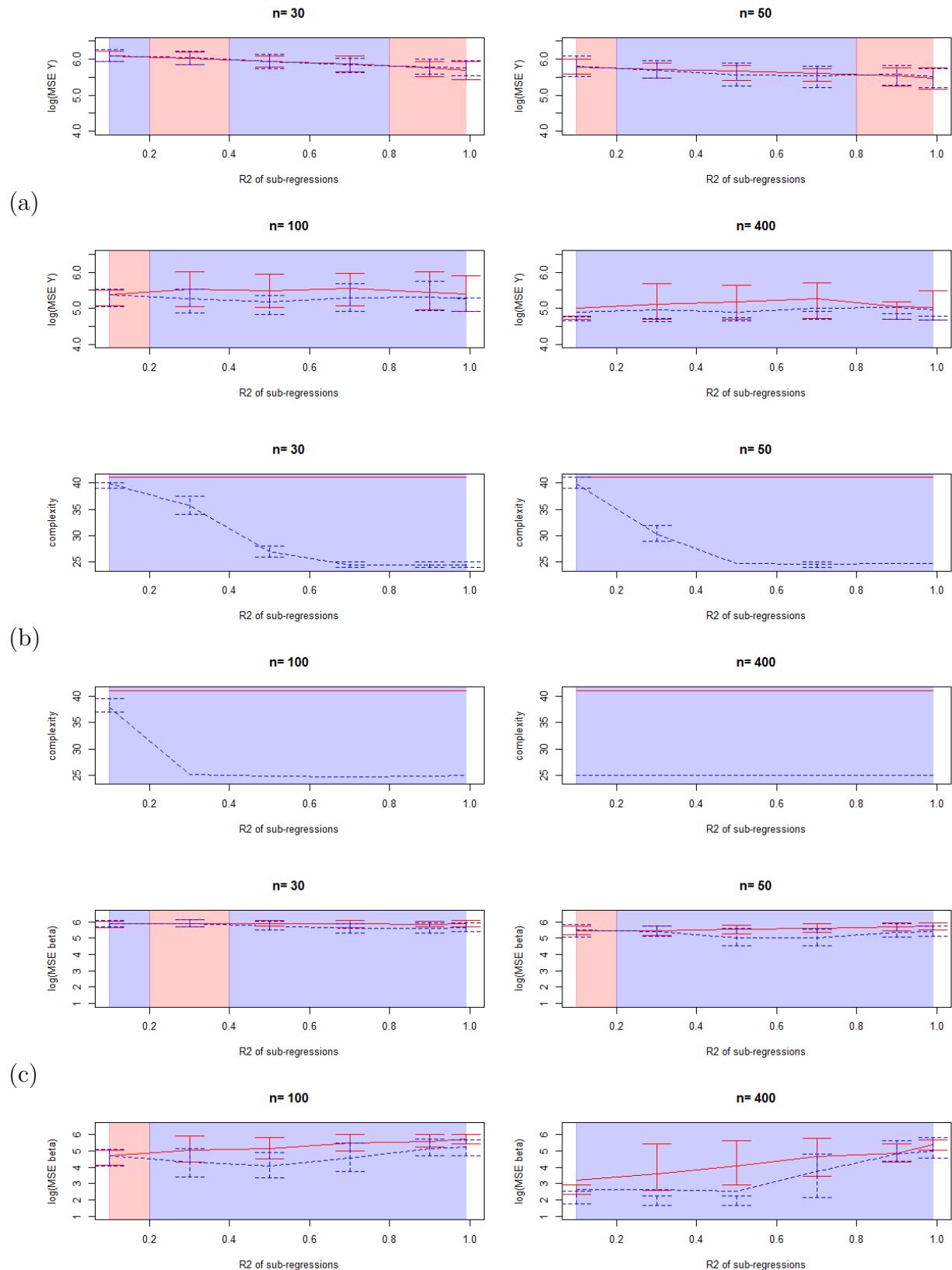


Figure 7.14: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

7.3.3 \mathbf{Y} depends only on covariates in \mathbf{X}_r (worst case for us)

We now try the method with a response depending only on variables in \mathbf{X}_r . The datasets used here were still the same. Depending only on \mathbf{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 4.5.

In such a case, usage of BIC_H instead of BIC_U is more beneficial because each additional sub-regression deletes a covariate in our marginal model and reduces the probability to find the true model.

Ordinary Least Squares: We first look at OLS (Figure 7.15) 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 help whatever the true model is. This is a really encouraging result. Improvement for small correlations but $n < d$ 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 $n < d$ only delete covariates to have $n = d$ (or $d + 1$ when there is an intercept). QR decomposition leads to delete the last covariates in the dataset but in our simulations, covariates in \mathbf{X}_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 $n > d$ 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.

Other methods: Variable selection methods (Figures 7.16 to 7.18) still are impacted by correlations but not enough to be improved by our marginal model. Neither is the ridge regression (Figure 7.19). The results confirm that this is the worst case for our marginal model.

Real datasets will provide \mathbf{Y} depending on a mix of covariates from both \mathbf{X}_f and \mathbf{X}_r so our marginal could help even with variable selection methods or ridge regression. We also recall that the structure \mathbf{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. Thus CorReg also has a psychological impact on a study that should not be overlooked. Once $\hat{\mathbf{S}}$ is found, trying the marginal model has no cost and should be tested.

Ordinary Least Squares when Y depends only on covariates in X_r

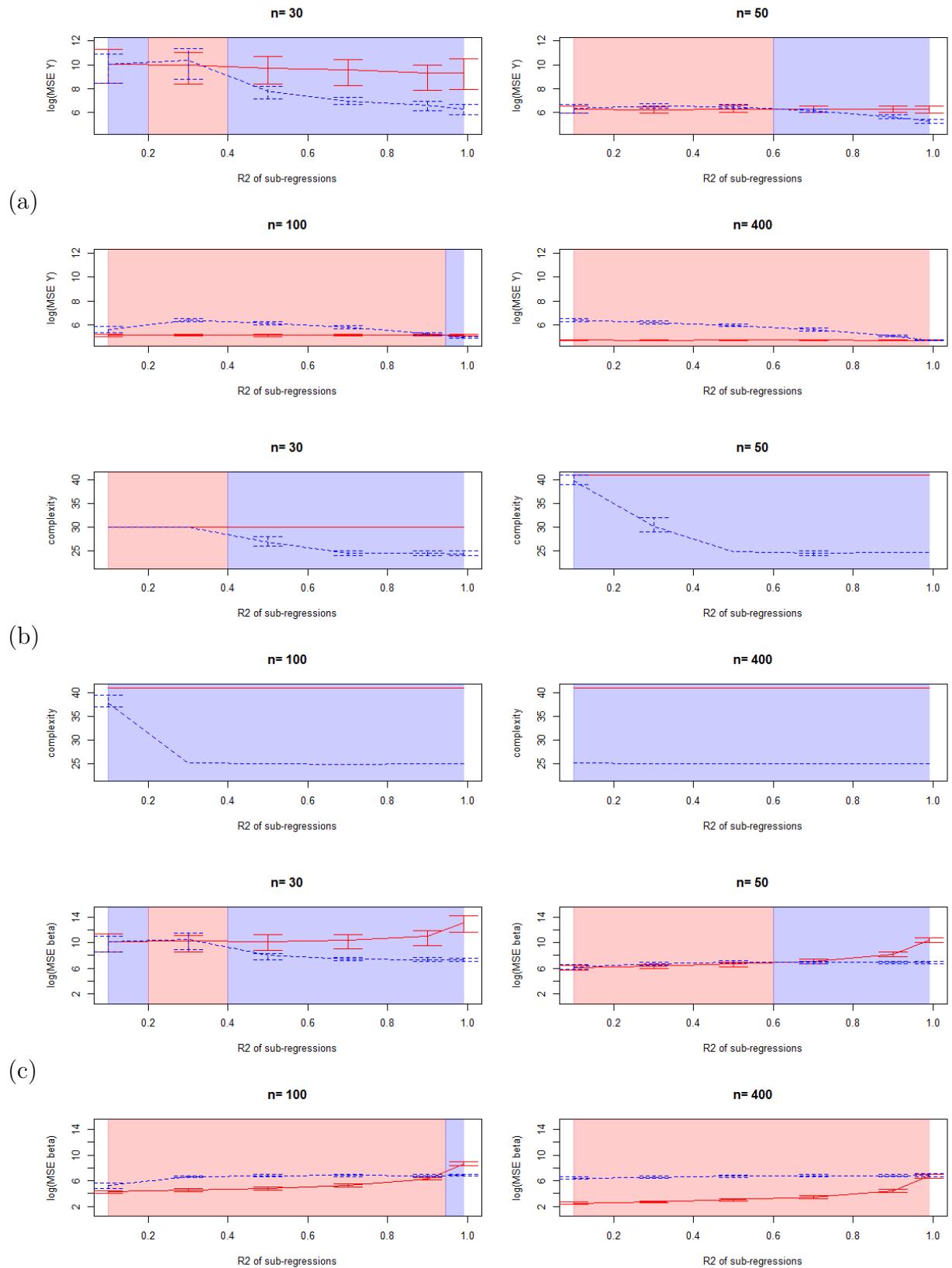


Figure 7.15: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

LASSO when Y depends only on covariates in X_r

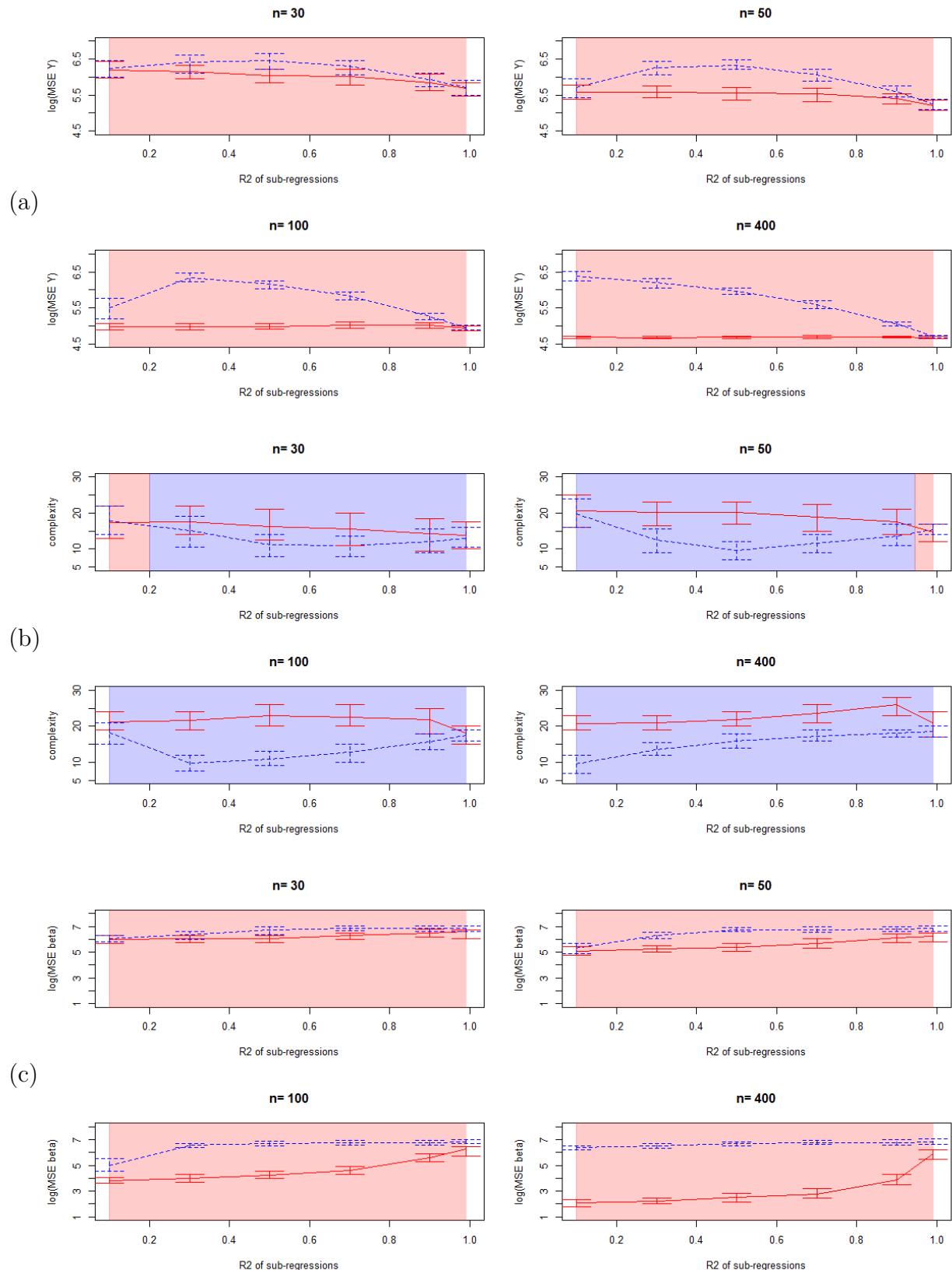


Figure 7.16: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when \mathbf{Y} depends only on covariates in \mathbf{X}_r

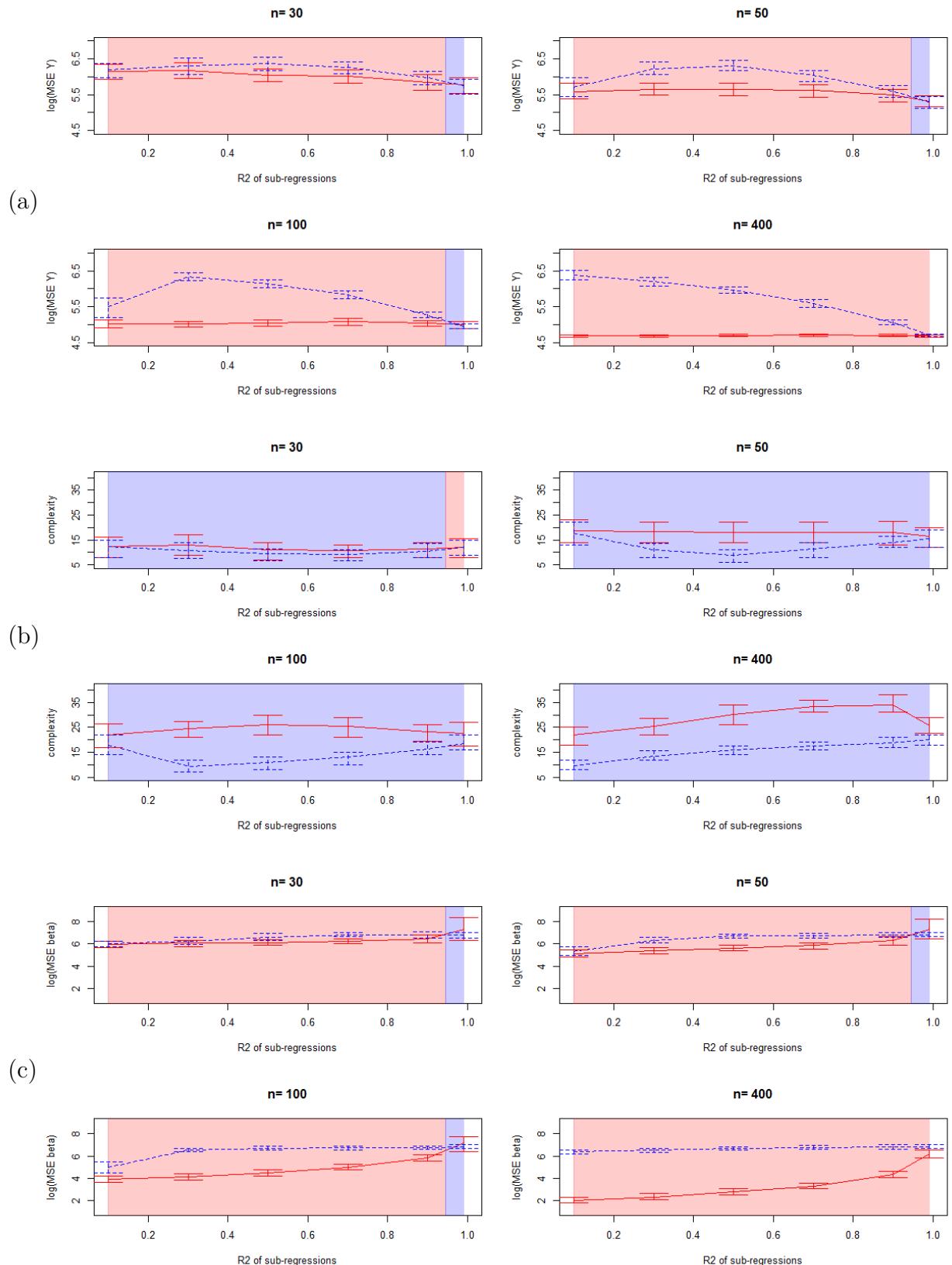


Figure 7.17: Comparison of the MSE on $\hat{\mathbf{Y}}$ (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends only on covariates in X_r

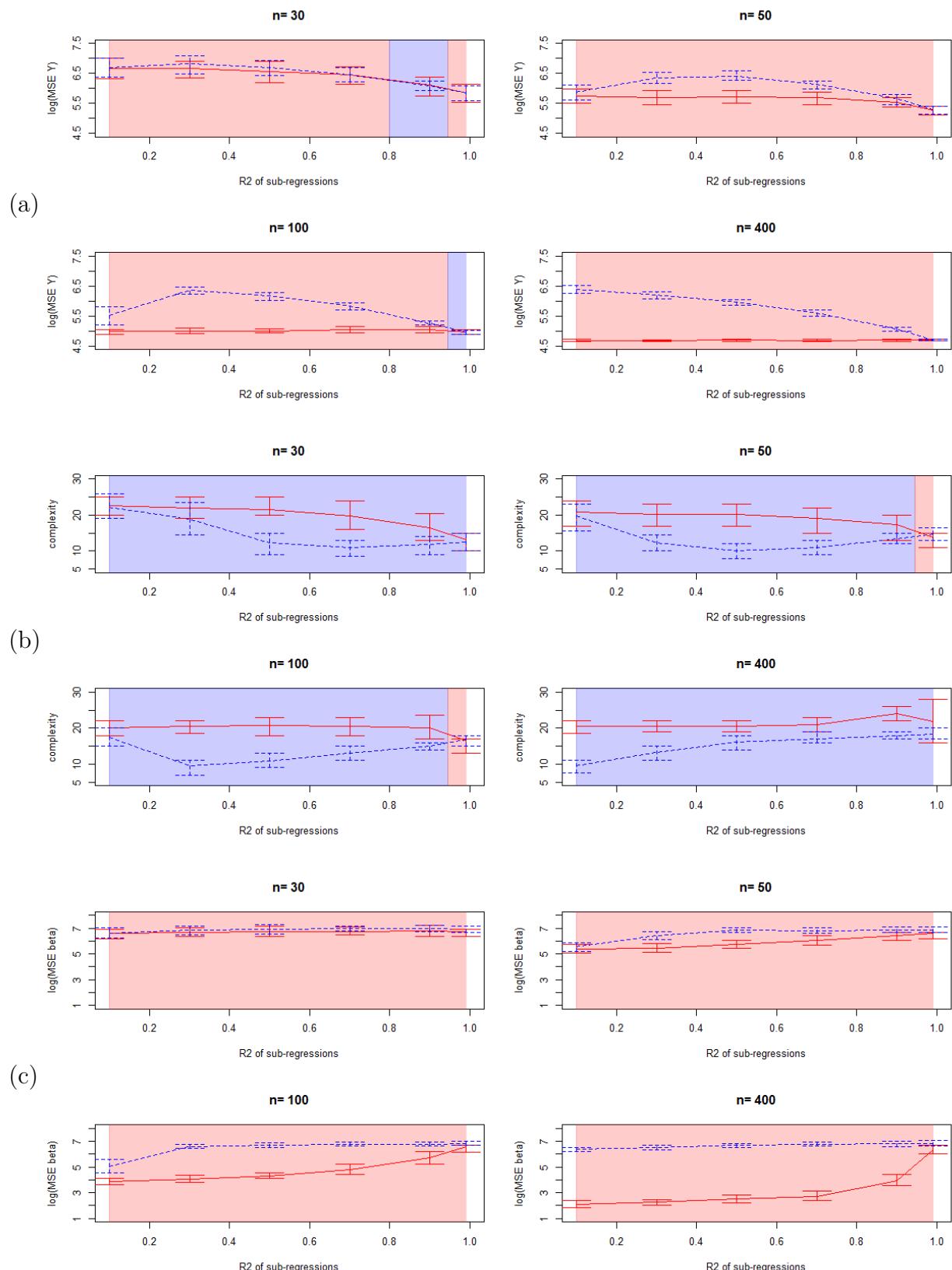


Figure 7.18: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when Y depends only on covariates in X_r

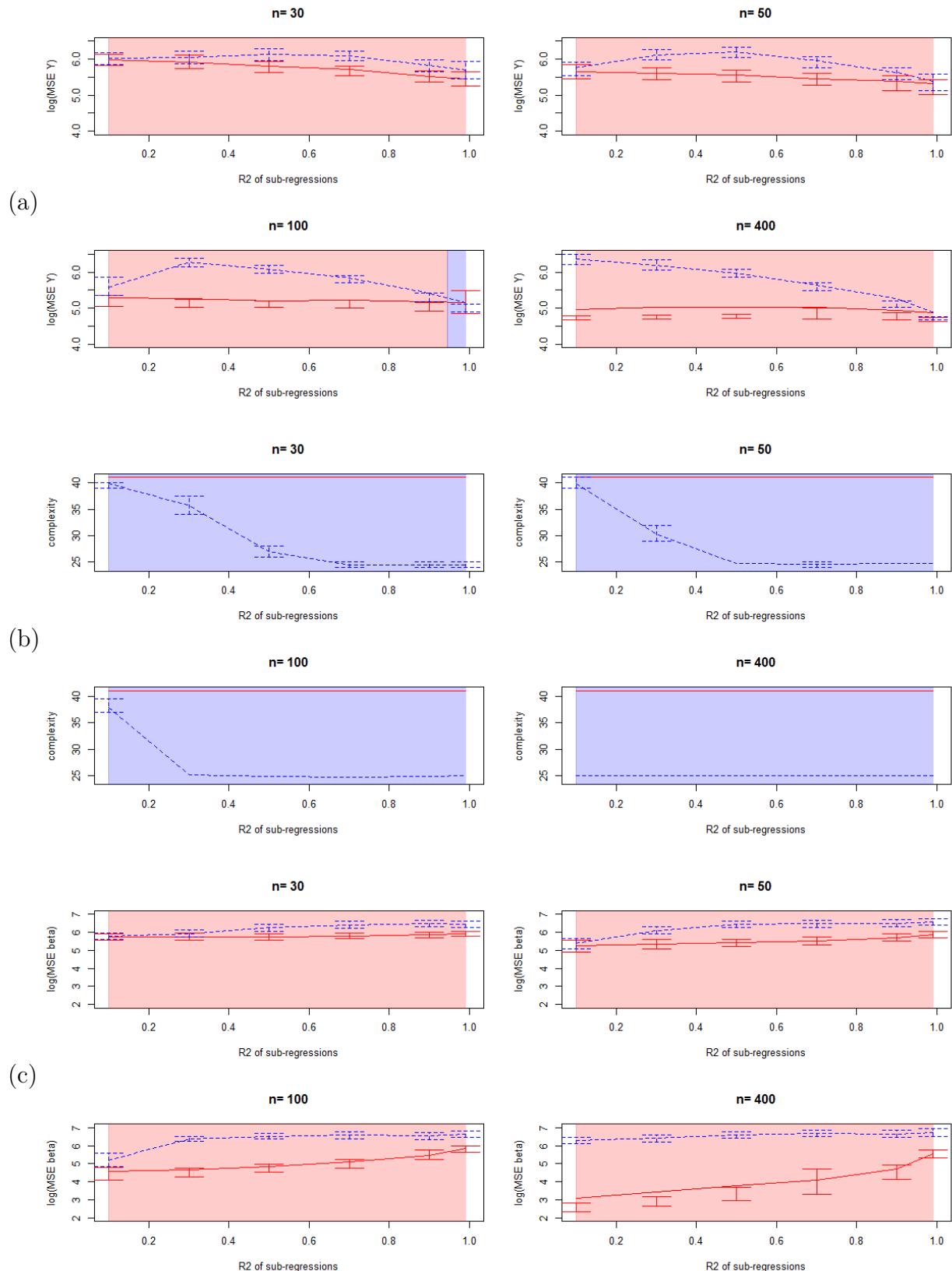


Figure 7.19: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

7.3.4 Robustness with non-linear case

We have generated a non-linear structure to test the robustness of the model. \mathbf{X}_f is a set of 6 independent Gaussian mixtures defined as previously but with random signs for the components means. $\mathbf{X}_r|\mathbf{X}_f; \boldsymbol{\alpha}, \sigma^2 = \mathbf{X}^7|\mathbf{X}^{\{1,2,3\}}; a, \sigma_1^2 = a(\mathbf{X}^1)^2 + \mathbf{X}^2 + \mathbf{X}^3 + \boldsymbol{\varepsilon}_1$. The matrix \mathbf{X} is then scaled before doing

$$\mathbf{Y}|\mathbf{X}; \sigma_Y^2 = \sum_{i=1}^7 \mathbf{X}^i + \boldsymbol{\varepsilon}_Y.$$

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.

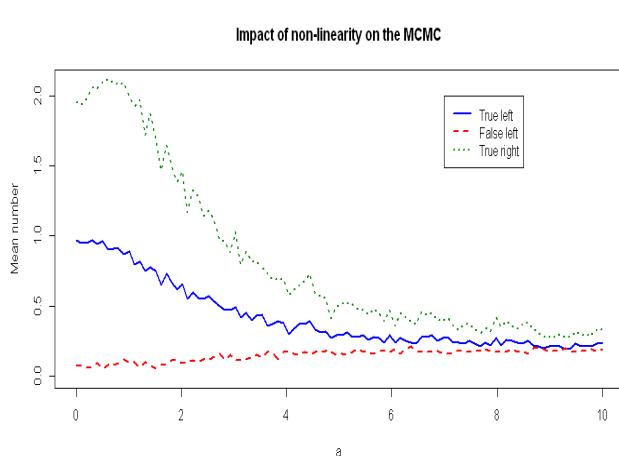


Figure 7.20: Evolution of the quality of $\hat{\mathbf{S}}$ when the paramater a increases

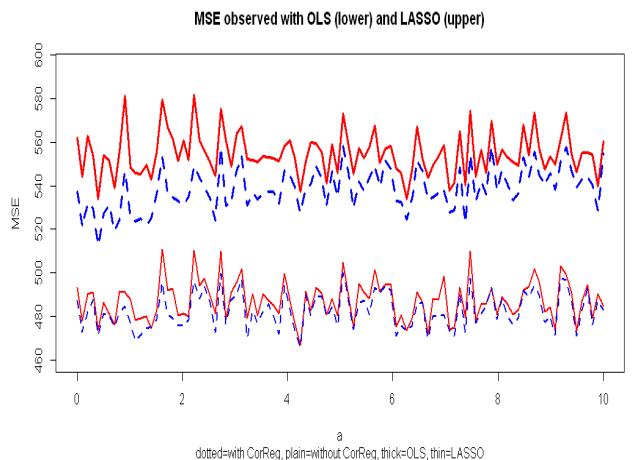


Figure 7.21: MSE on the main regression for OLS(thick) and LASSO (thin) used both with (plain) or without CorReg (dotted).

Figure 7.21 illustrates the advantage of using CorReg even with non-linear structures. Figure 7.20 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,
- $d = 205$ variables from the whole process to explain it.

We get a training set of $n = 3\,000$ products described by these 205 variables from the industrial process and also a validation sample of 847 products.

The objective here is not only to predict non-quality but to understand and then to avoid it. CORREG provides an automatic method without any *a priori* and can be combined with any variable selection methods. So it allows to obtain, in a small amount of time (several hours for this dataset), some indications on the source of the problem, and to use human resources efficiently. When quality crises occur, time is extremely precious so automation is a real stake. The combinatorial aspect of the sub-regression models makes it impossible to do manually.

To illustrate that some industrial variables are naturally highly correlated, we can measure the correlation ρ between some couple of variables. For instance, the width and the weight of a steel slab gives $|\rho| = 0.905$, the temperature before and after some tool gives $|\rho| = 0.983$, the roughness of both faces of the product gives $|\rho| = 0.919$ and a particular mean and a particular max gives $|\rho| = 0.911$. For an overview of correlations, Figure 8.1(a) gives an histogram of ρ where we can see that, however, many other variables are not so highly correlated.

CORREG estimated a structure of $d_r = 76$ sub-regressions with a mean of $\bar{d}_p = 5.17$ predictors. In the resulting uncorrelated covariate set \mathbf{X}_f the number of values $|\rho| > 0.7$ is 79.33% smaller than in \mathbf{X} . Indeed, Figure 8.1(b) displays the histogram of adjusted R^2 value (R_{adj}^2) and we can see that essentially large values of R_{adj}^2 are present. When we have a look at a more detailed level, we can see also that CORREG has been able non only to retrieve the above correlations (the width and the weight of a steel slab, *etc.*) but also to detect more complex structures describing physical models, like the width in function of the mean flow and the mean speed, even if the true physical model is not linear since “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).

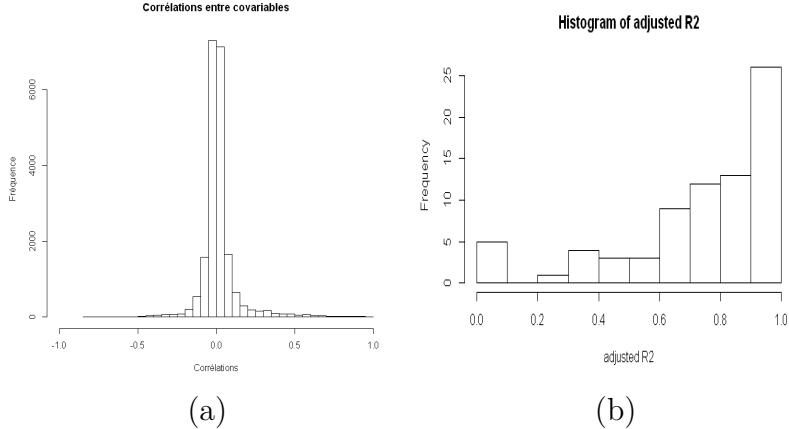


Figure 8.1: Quality case study: (a) Histogram of correlations ρ in \mathbf{X} , (b) histogram of the adjusted R_{adj}^2 for the $d_r = 76$ sub-regressions.

Note that the uncorrelated variables can be very well-modeled by parsimonious Gaussian mixtures as it is illustrated by Figure 8.2(a). In particular, the number of components is quite moderate as seen in Figure 8.2(b).

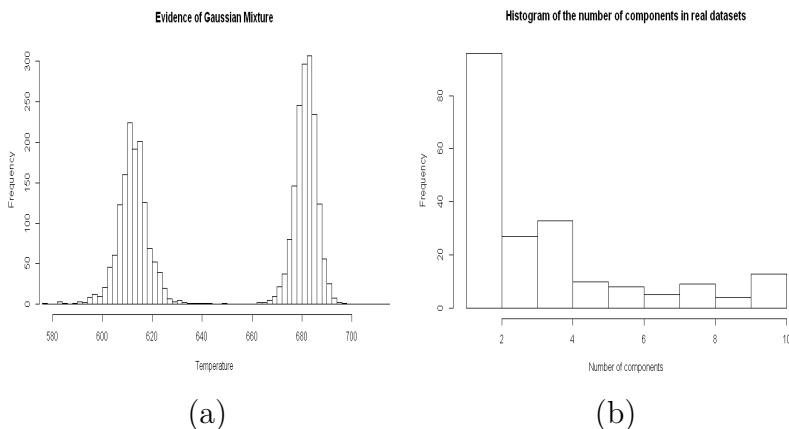


Figure 8.2: Quality case study: (a) Example of a non-Gaussian real variable easily modeled by a Gaussian mixture, (b) distribution of the number of components found for each covariate.

Table 8.1 displays predictive results associated to different estimation methods with and without CORREG. We can see that CORREG improves the results for each method tested in terms of prediction, with generally a more parsimonious regression on \mathbf{Y} . In terms of interpretation, this regression 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 not 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 do not even talk about the impact on the market shares that is even more important.

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: Quality case study: Results obtained on a validation sample ($n = 847$ individuals). In bold, the best MSE value.

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 8.3(a) shows that many variables are highly correlated. CORREG found $d_r = 55$ sub-regressions and corresponding R_{adj}^2 values are displayed in Figure 8.3(b). One of them seems to be weak ($R_{adj}^2 = 0.17$) but it corresponds in fact to a non-linear regression: It points out a link between diameter of a coil and some shape indicator. In this precise case, CORREG found a structure that helped to decorrelate covariates 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.

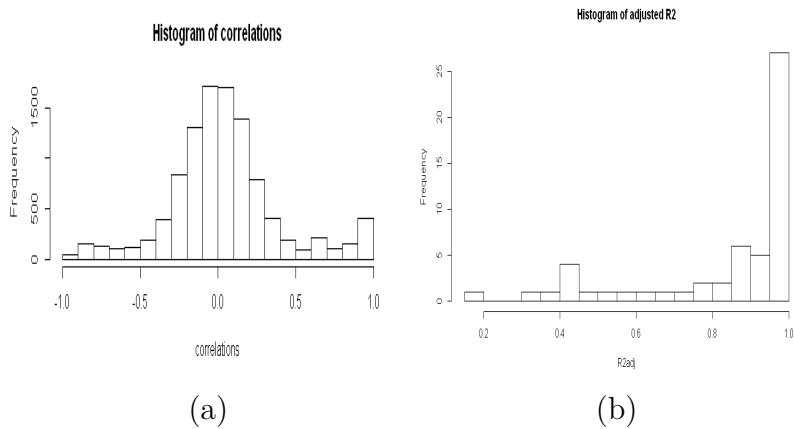


Figure 8.3: Production case study: (a) Histogram of correlations ρ in \mathbf{X} , (b) histogram of the adjusted R_{adj}^2 for the $d_r = 55$ sub-regressions.

As in the previous quality case study, we note that the uncorrelated variables can be very well-modeled by parsimonious Gaussian mixtures as it is illustrated by Figure 8.4(a). In particular, the number of components is really moderate as seen in Figure 8.4(b).

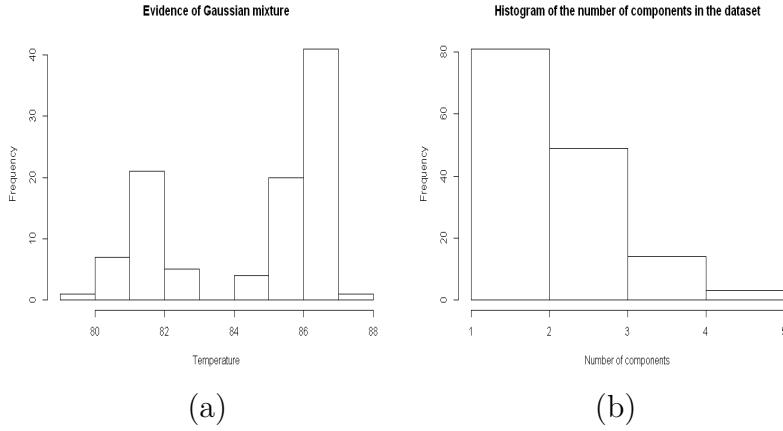


Figure 8.4: Production case study: (a) Example of a non-Gaussian real variable easily modeled by a Gaussian mixture, (b) distribution of the number of components found for each covariate.

Table 8.2 displays predictive results associated to different estimation methods with and without CORREG. Note that MSE is calculated though a leave-one-out method because of the small sample size. We can again see that CORREG globally improves the results for each method tested in terms of prediction, with always a more parsimonious regression on \mathbf{Y} .

Method	Indicator	With CORREG	Without CORREG
OLS	MSE	1.95	51 810
	complexity	91	100
LASSO	MSE	0.106	0.120
	complexity	27	34
elasticnet	MSE	0.140	0.148
	complexity	10	13
ridge	MSE	0.179	0.177
	complexity	91	146

Table 8.2: Production case study: Results obtained with leave-one out cross-validation ($n = 100, d = 145$). MSE is calculated though a leave-one-out method because of the small sample size. In bold, the best MSE value.

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

Part II

Extensions

Chapter 9

Using coefficients of regression and sub-regression to improve prediction

Abstract: We have seen that eviction of redundant covariates improves the results by a good tradeoff 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. So we propose a plug-in model to use the redundant covariates in a second estimation step.

9.1 Introduction

The structure \mathbf{S} of sub-regressions between the covariates has given us the opportunity to introduce a marginal model that in fact removes the response covariates. We have seen that it is an efficient method to decorrelate the covariates and thus to reduce the variance of the estimator, not only on simulated but also on real datasets (Chapter 8). But even if the sub-regressions are strong, we face a loss information that can be damageable as we have seen in section 7.3.3.

We know that using all the covariates simultaneously (classical OLS) gives bad results due to correlations. But we can use them sequentially by using the explicit decomposition of the marginal model that makes appear the coefficient β_r in both the noise and β_f^* . Thus we are able to obtain by plug-in better estimates of β_r and β_f .

9.2 Plug-in model

After the estimation of the marginal model in section 4.3, we have both $\hat{\alpha}^*$ and $\hat{\beta}^*$. Reminding the marginal model

$$\mathbf{Y} | \mathbf{X}_f, \mathbf{S}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2, \sigma_Y^2 = \mathbf{X}_f \underbrace{(\beta_f + \alpha \beta_r)}_{\beta_f^*} + \underbrace{\varepsilon \beta_r + \varepsilon_Y}_{\varepsilon_Y^*}, \quad (9.1)$$

the residual of the marginal model is decomposed as follows:

$$\varepsilon_Y^* = \varepsilon \beta_r + \varepsilon_Y. \quad (9.2)$$

And because we have already estimated the structure and the marginal model we can provide some new estimators to improve estimation of β :

- Estimation of ε_Y^* relying on $\hat{\beta}_f^*$:

$$\hat{\varepsilon}_Y^* = \mathbf{Y} - \mathbf{X}_f \hat{\beta}_f^*. \quad (9.3)$$

- Estimation of $\boldsymbol{\varepsilon}$ relying on $\hat{\boldsymbol{\alpha}}^*$:

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{X}_r - \mathbf{X}_f \hat{\boldsymbol{\alpha}}^*. \quad (9.4)$$

These two new estimators allow us to reduce the noise of the marginal model by linear regression based on equation (9.2) and to obtain an estimator of $\boldsymbol{\beta}_r$ better than $\hat{\boldsymbol{\beta}}_r^* = \mathbf{0}$.

- New estimation of $\boldsymbol{\beta}_r$ to reduce the bias:

$$\hat{\boldsymbol{\beta}}_r^\varepsilon = (\hat{\boldsymbol{\varepsilon}}' \hat{\boldsymbol{\varepsilon}})^{-1} \hat{\boldsymbol{\varepsilon}}' (\mathbf{Y} - \mathbf{X}_f \hat{\boldsymbol{\beta}}_f^*) \quad (9.5)$$

We suppose that $\hat{\boldsymbol{\beta}}_f^*$ and $\hat{\boldsymbol{\alpha}}^*$ are consistent (for example using OLS). The new estimator $\hat{\boldsymbol{\beta}}_r^\varepsilon$ is consistent (OLS and Slutsky's theorem) so we have

$$\mathbb{E}(\hat{\boldsymbol{\beta}}_r^\varepsilon) = \boldsymbol{\beta}_r \quad (9.6)$$

and

$$\text{Var}(\hat{\boldsymbol{\beta}}_r^\varepsilon) = \text{Var}[(\hat{\boldsymbol{\varepsilon}}' \hat{\boldsymbol{\varepsilon}})^{-1} \hat{\boldsymbol{\varepsilon}}' (\mathbf{Y} - \mathbf{X}_f \hat{\boldsymbol{\beta}}_f^*)] \quad (9.7)$$

$$= \text{Var}[(\hat{\boldsymbol{\varepsilon}}' \hat{\boldsymbol{\varepsilon}})^{-1} \hat{\boldsymbol{\varepsilon}}' (\mathbf{Y} - \mathbf{X}_f (\mathbf{X}'_f \mathbf{X}_f)^{-1} \mathbf{X}'_f \mathbf{Y})] \quad (9.8)$$

$$= \text{Var}[(\hat{\boldsymbol{\varepsilon}}' \hat{\boldsymbol{\varepsilon}})^{-1} \hat{\boldsymbol{\varepsilon}}' (\mathbf{I}_n - \underbrace{\mathbf{X}_f (\mathbf{X}'_f \mathbf{X}_f)^{-1} \mathbf{X}'_f}_{\mathbf{H}_f} \mathbf{Y})] \quad (9.9)$$

$$= \sigma_Y^2 (\hat{\boldsymbol{\varepsilon}}' \hat{\boldsymbol{\varepsilon}})^{-1} \hat{\boldsymbol{\varepsilon}}' (\mathbf{I}_n - \mathbf{H}_f) \hat{\boldsymbol{\varepsilon}} (\hat{\boldsymbol{\varepsilon}}' \hat{\boldsymbol{\varepsilon}})^{-1} \quad (9.10)$$

Then we can estimate \mathbf{Y} with a smaller noise:

$$\hat{\mathbf{Y}}_{\text{plug-in}} = \mathbf{X}_f \hat{\boldsymbol{\beta}}_f^* + \hat{\boldsymbol{\varepsilon}} \hat{\boldsymbol{\beta}}_r^\varepsilon. \quad (9.11)$$

This is an asymptotically unbiased estimator of \mathbf{Y} because the marginal model is unbiased and all the estimators are consistent. We can even improve estimation of $\boldsymbol{\beta}_f$ by doing an identification step. We have $\boldsymbol{\beta}_f^* = \boldsymbol{\beta}_f + \boldsymbol{\alpha}^* \boldsymbol{\beta}_r$.

- New estimation of $\boldsymbol{\beta}_f$ to reduce the bias:

$$\hat{\boldsymbol{\beta}}_f^\varepsilon = \hat{\boldsymbol{\beta}}_f^* - \hat{\boldsymbol{\alpha}}^* \hat{\boldsymbol{\beta}}_r^\varepsilon. \quad (9.12)$$

Properties of $\hat{\boldsymbol{\beta}}_f^\varepsilon$ are obtained as above and give:

$$\mathbb{E}(\hat{\boldsymbol{\beta}}_f^\varepsilon) = \boldsymbol{\beta}_f \text{ (still unbiased)} \quad (9.13)$$

$$\text{Var}(\hat{\boldsymbol{\beta}}_f^\varepsilon) = \text{Var}[\hat{\boldsymbol{\beta}}_f^* - \hat{\boldsymbol{\alpha}}^* \hat{\boldsymbol{\beta}}_r^\varepsilon] \quad (9.14)$$

$$= \sigma_Y^2 [(\mathbf{X}'_f \mathbf{X}_f)^{-1} + \hat{\boldsymbol{\alpha}}^* (\hat{\boldsymbol{\varepsilon}}' \hat{\boldsymbol{\varepsilon}})^{-1} \hat{\boldsymbol{\varepsilon}}' (\mathbf{I}_n - \mathbf{H}_f) \hat{\boldsymbol{\varepsilon}} (\hat{\boldsymbol{\varepsilon}}' \hat{\boldsymbol{\varepsilon}})^{-1} (\hat{\boldsymbol{\alpha}}^*)'] \quad (9.15)$$

Variance of $\hat{\boldsymbol{\beta}}_f^\varepsilon$ has supplementary terms but relies on smaller and better conditioned matrices than classical OLS applied on \mathbf{X} so we hope that correlations are strong enough to make the difference. Compared to 4.10 we have a new bias-variance tradeoff with an unbiased estimator.

Figure 9.1 points out the target of this plug-in model: the cases with enough correlations to have problem when using \mathbf{X} but not enough correlations to have truly redundant covariates and to be able to remove some of them (marginal model) without significant information loss. We see that our new model is efficient.

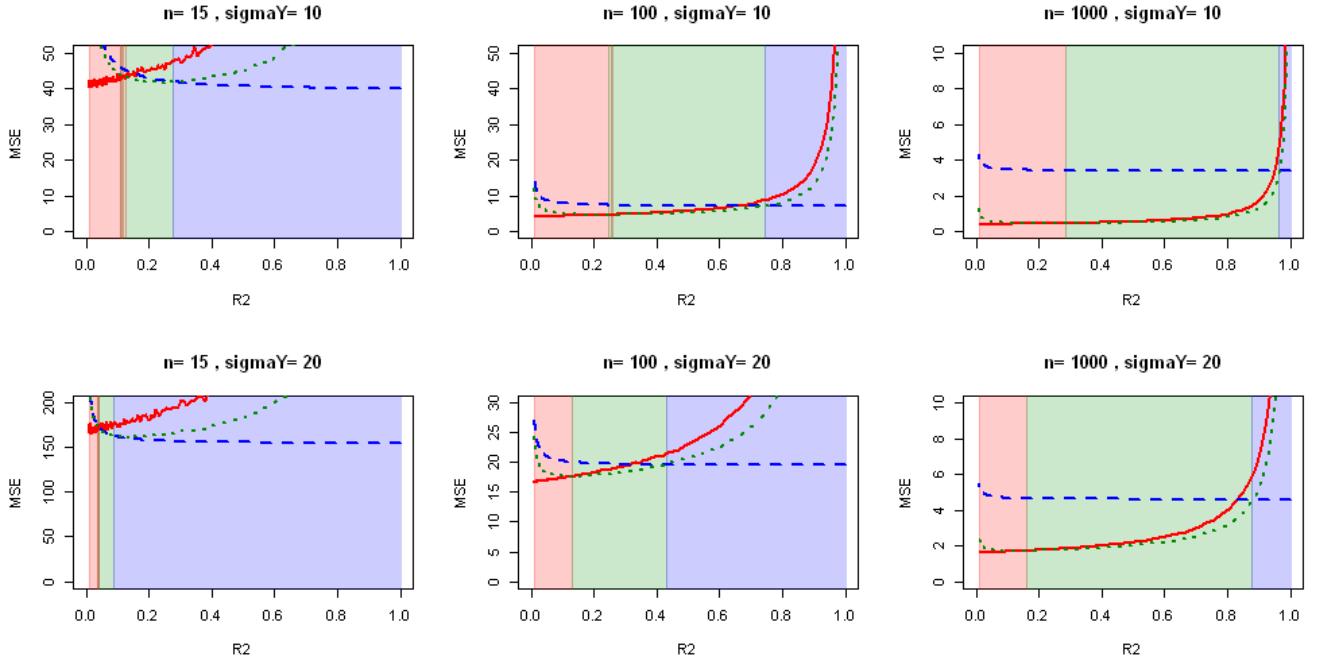


Figure 9.1: MSE on $\hat{\beta}$ of OLS (plain red) and CorReg marginal (blue dashed) and CorReg plug-in (green dotted) estimators for varying R^2 of the sub-regression, n and σ_Y . Results obtained on the running example with $d = 5$ covariates.

9.3 Interpretation and latent variables

$\hat{\beta}_r$ can be interpreted as the proper effect of \mathbf{X}_r on \mathbf{Y} in that it is the effect of the part of \mathbf{X}_r that is independent from other covariates. Then if \mathbf{X}_r is correlated to \mathbf{Y} only through its correlation with \mathbf{X}_f this sequential estimation will point it out and give a parsimonious model ($\hat{\beta}_r = 0$) but the real stake is greater. We can see ε as a latent variable instead of the noise of a sub-regression. This latent variable is known to be independent of \mathbf{X}_f and dependent of \mathbf{X}_r so we can appreciate its meaning and we also know its value by equation (9.4). Thus, the plug-in model can reveal some kinds of latent variables.

Both marginal and plug-in model are easy to compute then we can use for example the marginal model for interpretation (more parsimonious) and the plug-in model for prediction. But we will see that it is not always the better choice because plug-in estimator can always be consistent (each covariate can be used) contrary to the marginal model.

9.4 Consistency of the LASSO

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 $d = 3$ and $n = 1\,000$.

We define $\mathbf{X}^1, \mathbf{X}^2, \varepsilon_Y, \varepsilon_1 i.i.d. \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ and then

$$\mathbf{X}^3 | \mathbf{X}^1, \mathbf{X}^2, \mathbf{S}; \sigma_1^2 = \frac{2}{3}\mathbf{X}^1 + \frac{2}{3}\mathbf{X}^2 + \frac{1}{3}\varepsilon_1 \text{ and} \quad (9.16)$$

$$\mathbf{Y} | \mathbf{X}^1, \mathbf{X}^2, \mathbf{S}; \sigma_Y^2 = 2\mathbf{X}^1 + 3\mathbf{X}^2 + \varepsilon_Y. \quad (9.17)$$

We compare consistencies of complete, marginal and full plug-in model with LASSO (and LAR) for selection. It happens on some tries that our MCMC algorithm don't find the true structure but a permuted one so we both look at the results obtained with the true $\mathbf{S} = ((3), (\{1, 2\}))$ (but $\hat{\boldsymbol{\alpha}}$ is used) and with the structure found by the Markov chain after a few seconds.

True \mathbf{S} was found 987 times on 1 000 tries.

	Classical LASSO	CorReg + marginal LASSO	CorReg + full plug-in LASSO
True \mathbf{S}	1.006784	1.005468	1.006093
$\hat{\mathbf{S}}$	1.006784	1.028154	1.006350

Table 9.1: MSE observed on a validation sample (1 000 individuals)

We observe as we hoped that our marginal model is better when using true \mathbf{S} (coercing real zeros) and that marginal with $\hat{\mathbf{S}}$ is penalised (coercing wrong coefficients to be zeros when true \mathbf{S} is not found). But the main point is that the plug-in model stays better than the classical one with the true \mathbf{S} and corrects enough the marginal model to be better than the classical LASSO when using $\hat{\mathbf{S}}$. Improvements here are very small but regular even on 1 000 tries.

When we look at the consistency (Table 9.2): it is clear that the plug-in model signifi-

	Classical LASSO	CorReg marginal + LASSO	CorReg full plug-in + LASSO
True \mathbf{S}	0	1000	820
$\hat{\mathbf{S}}$	0	987	809

Table 9.2: Number of consistent model found (\mathbf{Y} depending on $\mathbf{X}^1, \mathbf{X}^2$ and only them) on 1 000 tries

cantly improves the consistency of the model estimated on \mathbf{S} when compared to the classical LASSO. Both model have the choice to keep or not each covariate but only the plug-in model is consistent. Model using the true structure can't be improved because the marginal is already consistent so the plug-in is worse or equal to the marginal one in terms of consistency. Classical LASSO is never consistent on this example but we don't only improve this situation, we give consistent models in most of the cases.

9.5 Numerical results

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

9.5.1 \mathbf{Y} depends on all variables in \mathbf{X}

We first try the method with a response depending on all covariates. (The marginal model reduces the dimension and can't give the true model if there is a structure).

Ordinary Least Squares We observe for OLS (Figure 9.2) 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 find 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.

Other methods: Combined with variable selection methods (Figures 9.3 to 9.5) 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 (Figure 9.6) leads to the same conclusion.

Ordinary Least Squares when Y depends on all variables in X

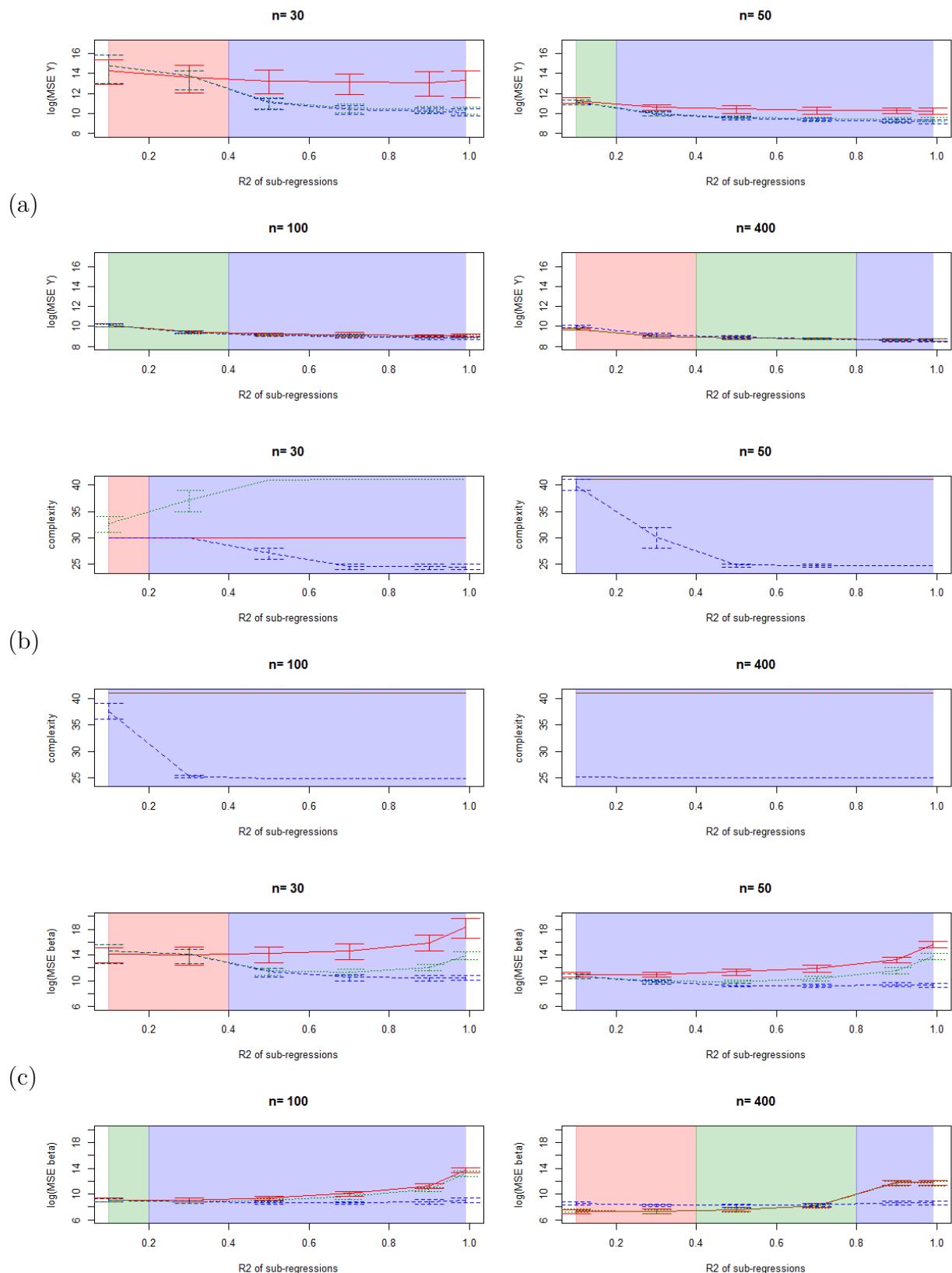


Figure 9.2: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

LASSO when \mathbf{Y} depends on all variables in \mathbf{X}

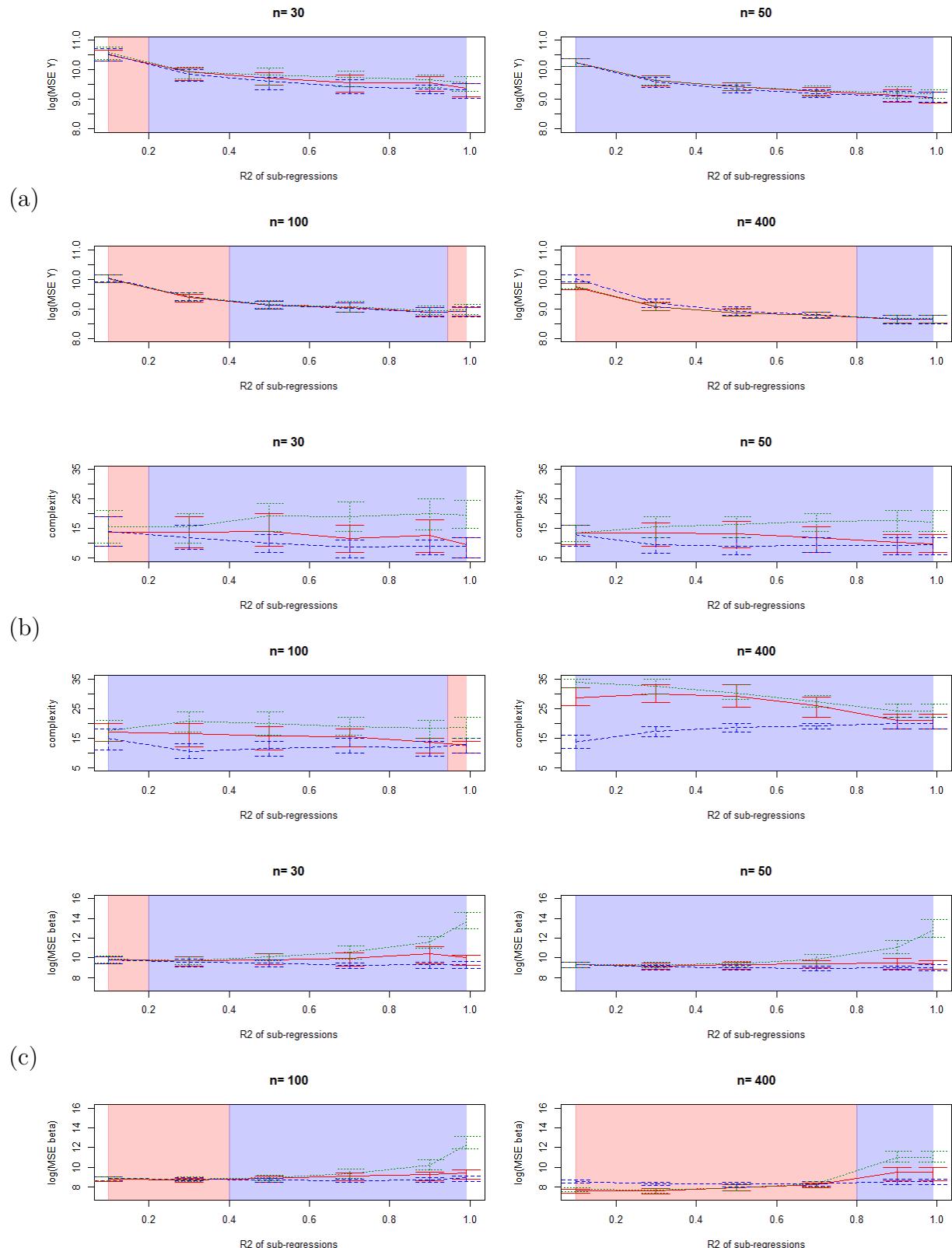


Figure 9.3: Comparison of the MSE on $\hat{\mathbf{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when \mathbf{Y} depends on all variables in \mathbf{X}

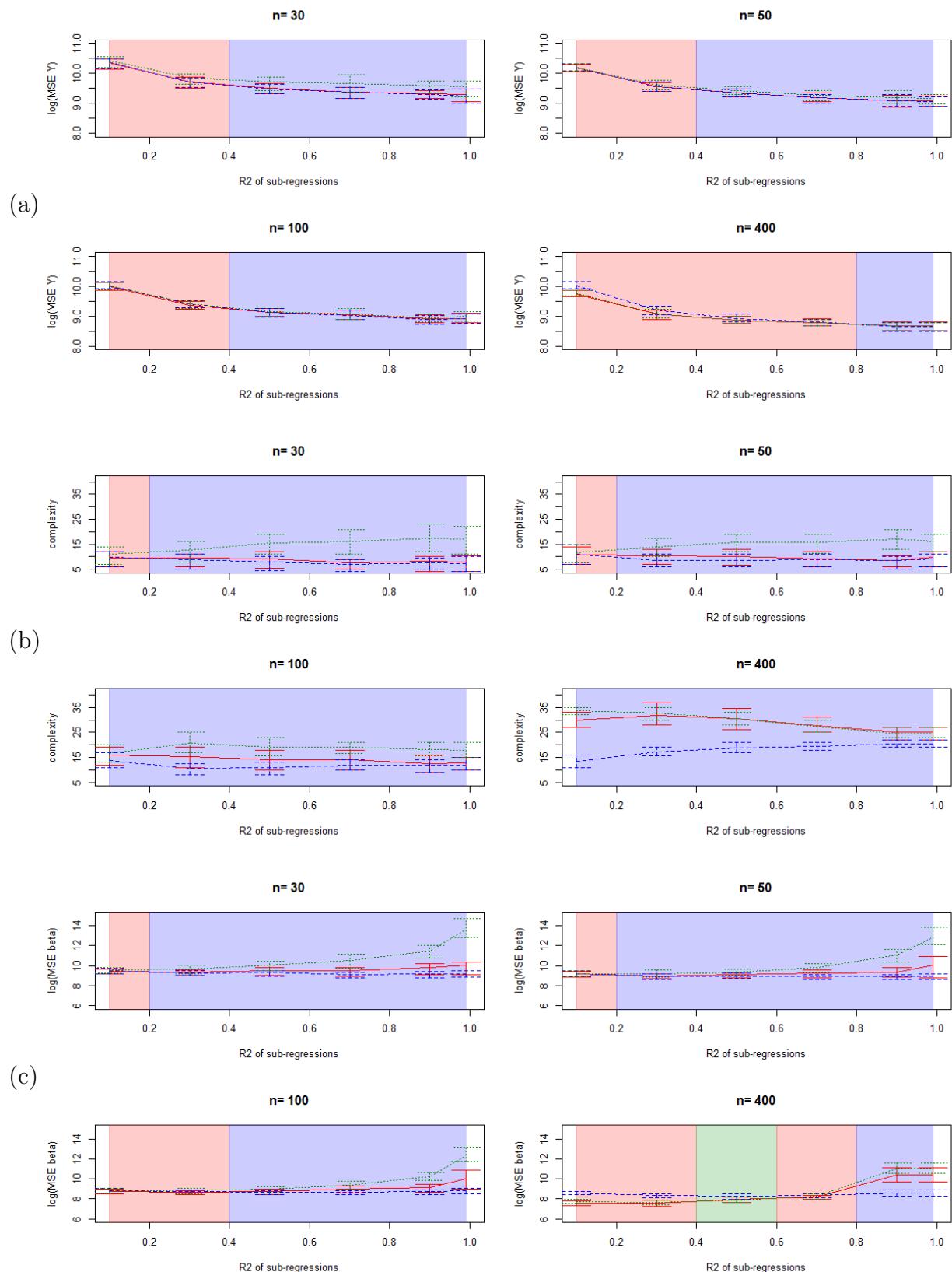


Figure 9.4: Comparison of the MSE on $\hat{\mathbf{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends on all variables in X

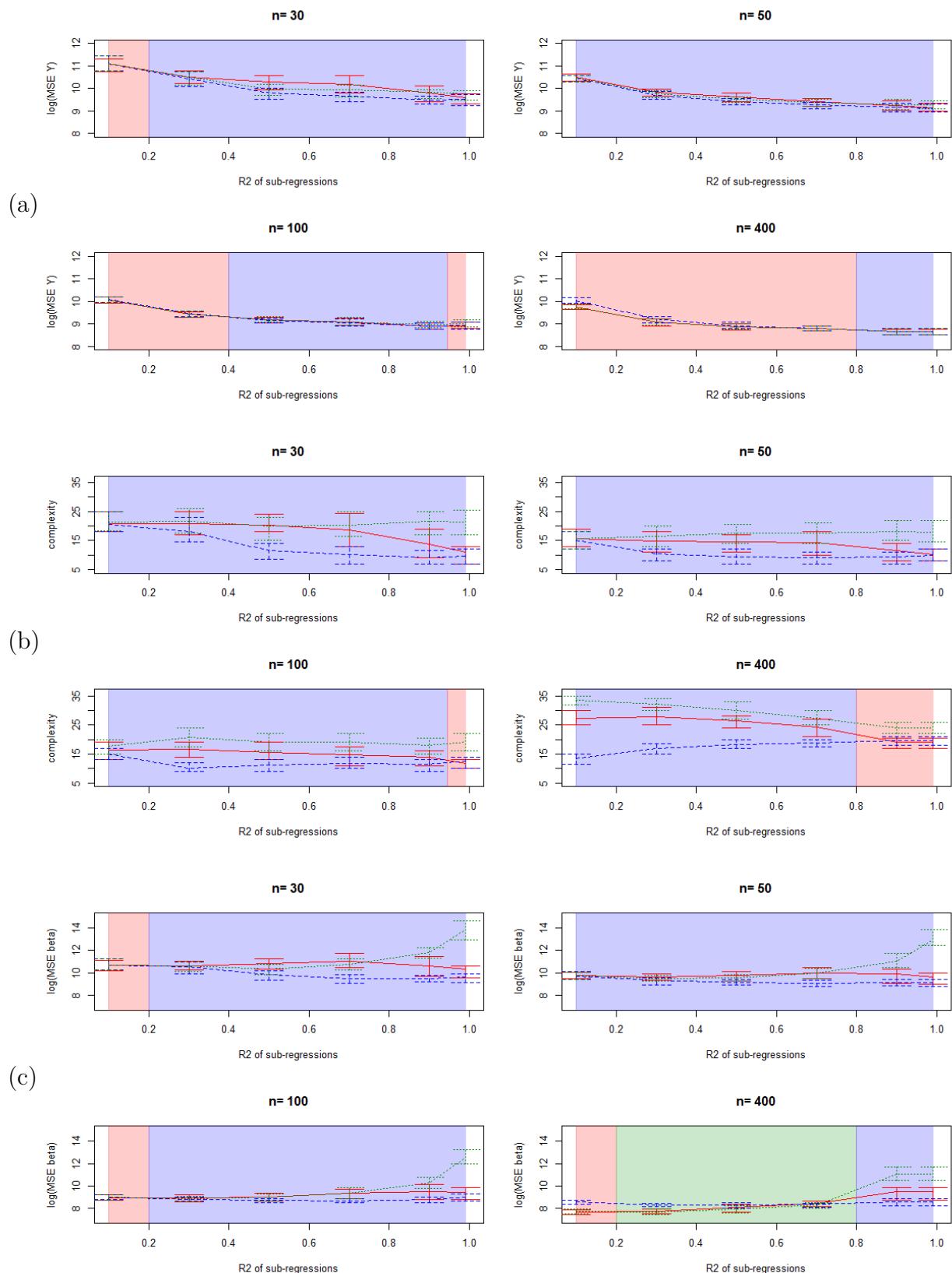


Figure 9.5: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when Y depends on all variables in X

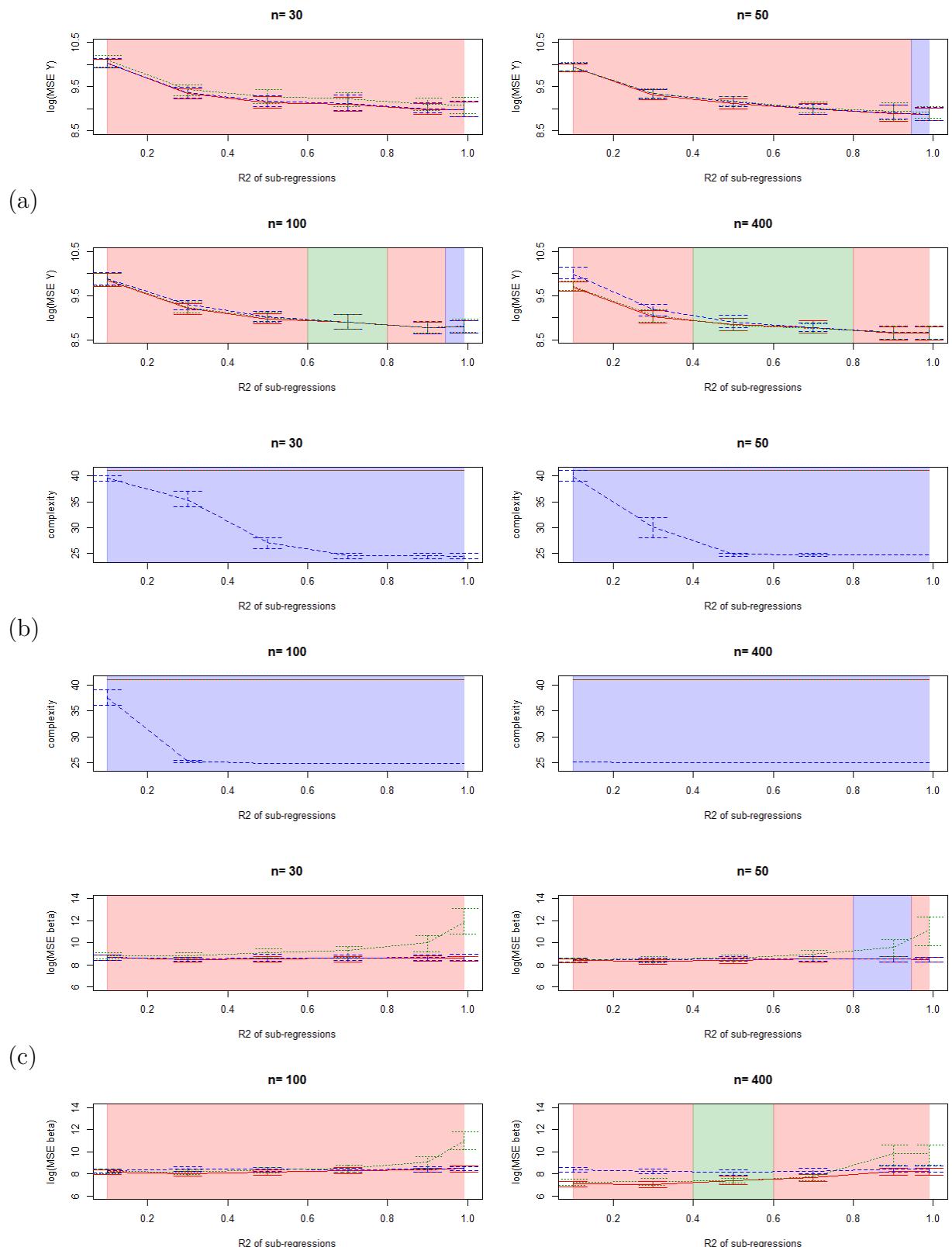


Figure 9.6: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

9.5.2 \mathbf{Y} depends only on covariates in \mathbf{X}_f

We look then at the case where \mathbf{Y} depends only on covariates in \mathbf{X}_f .

Ordinary Least Squares: Figure 9.7 shows that the plug-in model stays better than OLS even when \mathbf{Y} depends only on covariates in \mathbf{X}_f . These good results come from the sequential estimation with the first part of the parameters estimated with a well-conditioned matrix. Hence the variance is reduced even without

Other methods: Variable selection methods (Figures 9.8 to 9.10) are not improved by the plug-in model and the marginal model remains the best. But ridge regression (Figure 9.11) is much similar to OLS and then the plug-in model gives good results, even if the marginal model is sufficient.

9.5.3 \mathbf{Y} depends only on covariates in \mathbf{X}_r

We then try the method with a response depending only on variables in \mathbf{X}_r . Depending only on \mathbf{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 case is the reason why we have developed the plug-in model.

Better but not sufficient: Concerning OLS (Figure 9.12) 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 (Figures 9.13 to 9.15) even if it is not sufficient to improve the complete model. This phenomenon is still observed with the ridge regression (Figure 9.16) with more efficiency but the complete model stays better.

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 $\hat{\mathbf{S}}$ is not exactly the true \mathbf{S} but the partition is good, the marginal model is not impacted whereas the plug-in model uses both $\hat{\beta}_f^*$ and $\hat{\alpha}$ so it does not rely only on one estimator. $\hat{\alpha}$ is another source of bias for finite values of n and it depends itself on $\hat{\mathbf{S}}$. Ordinary Least Squares really are in great trouble when confronted to correlated datasets so the plug-in model improves OLS anyway 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. However, like the marginal model, the plug-in model has a small computational cost compared to the estimation of \mathbf{S} . So we recommend to compute both complete, marginal and plug-in model and to compare the results in a second time.

Ordinary Least Squares when Y depends only on covariates in X_f

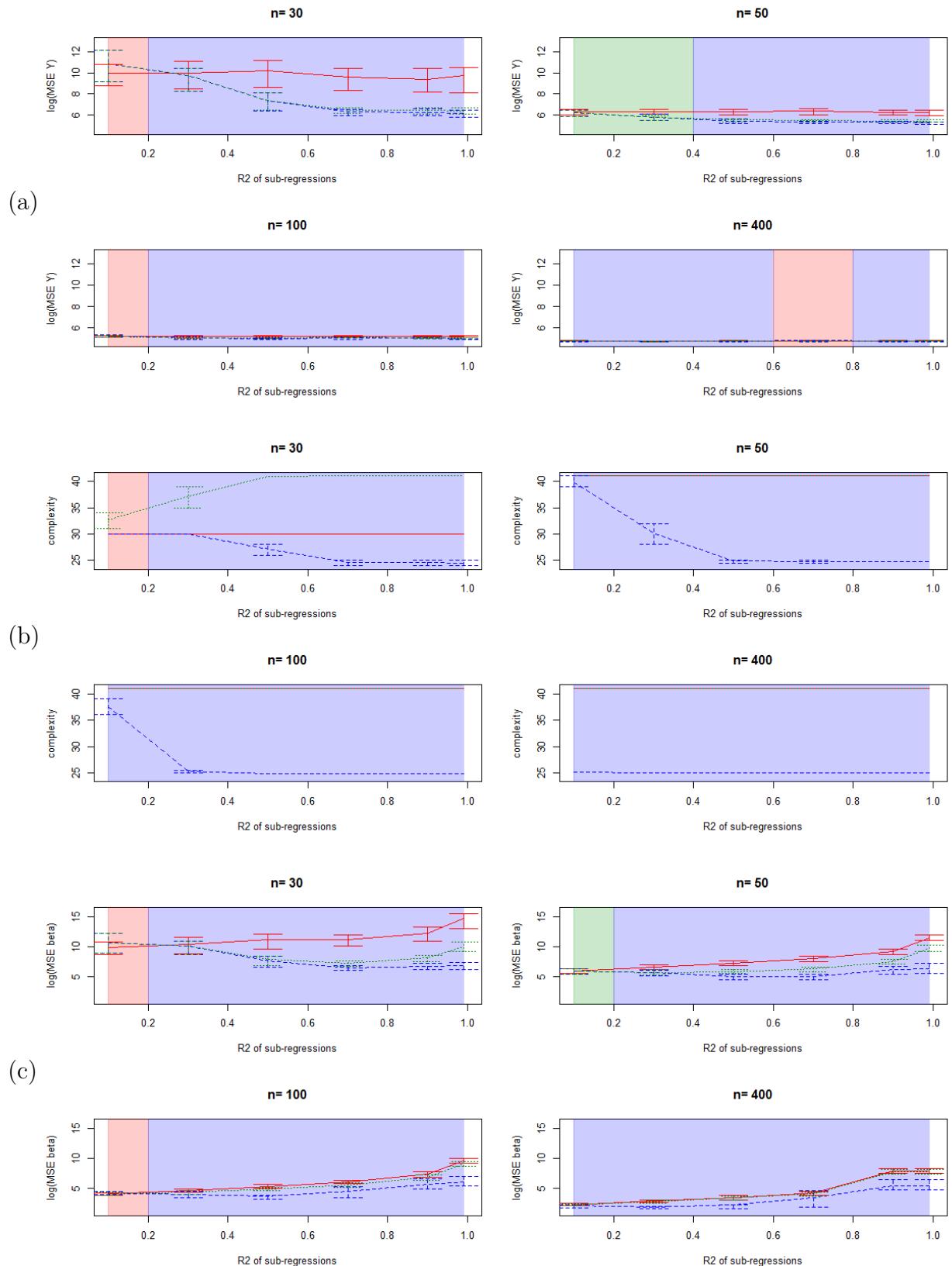


Figure 9.7: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

LASSO when \mathbf{Y} depends only on covariates in \mathbf{X}_f

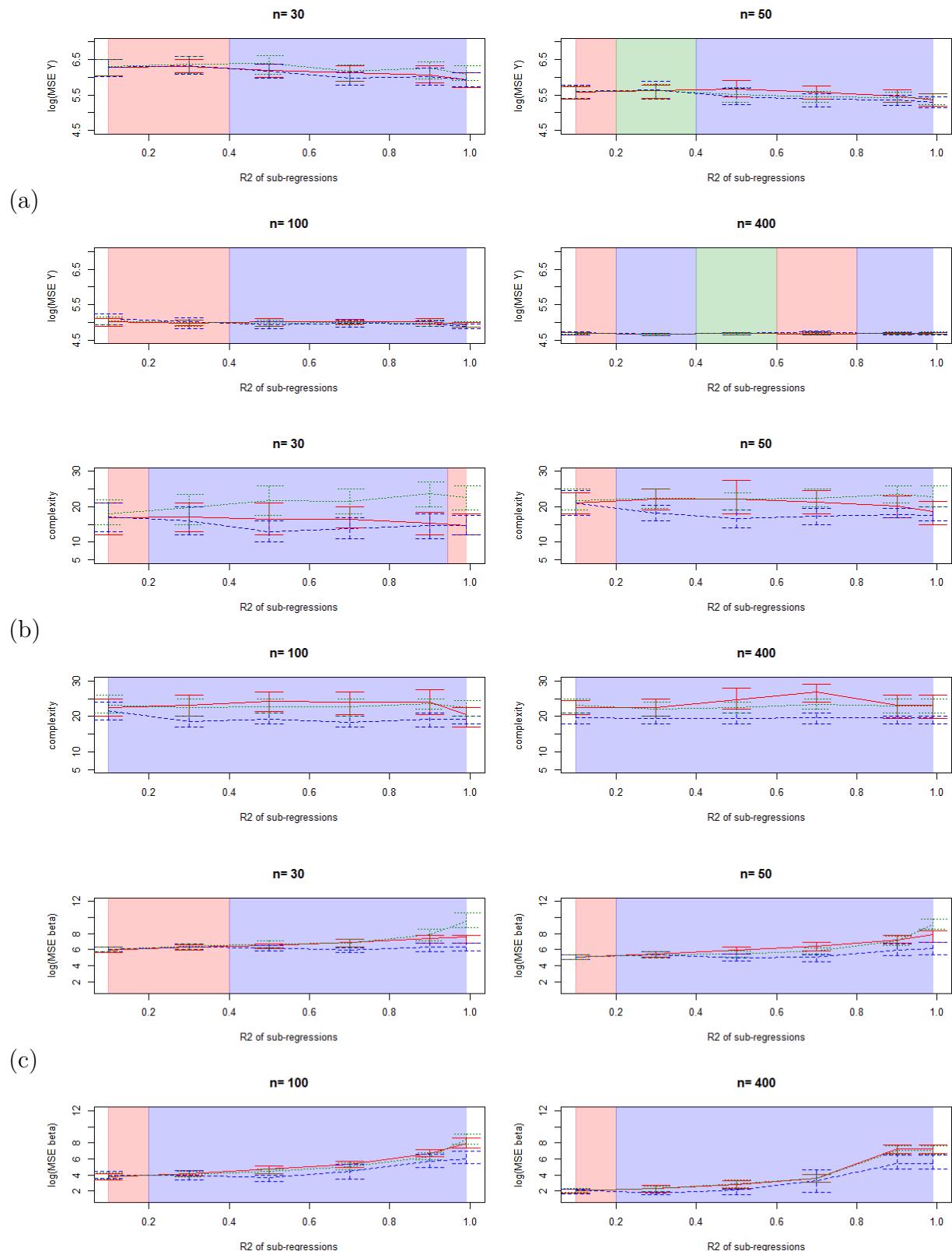


Figure 9.8: Comparison of the MSE on $\hat{\mathbf{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when \mathbf{Y} depends only on covariates in \mathbf{X}_f

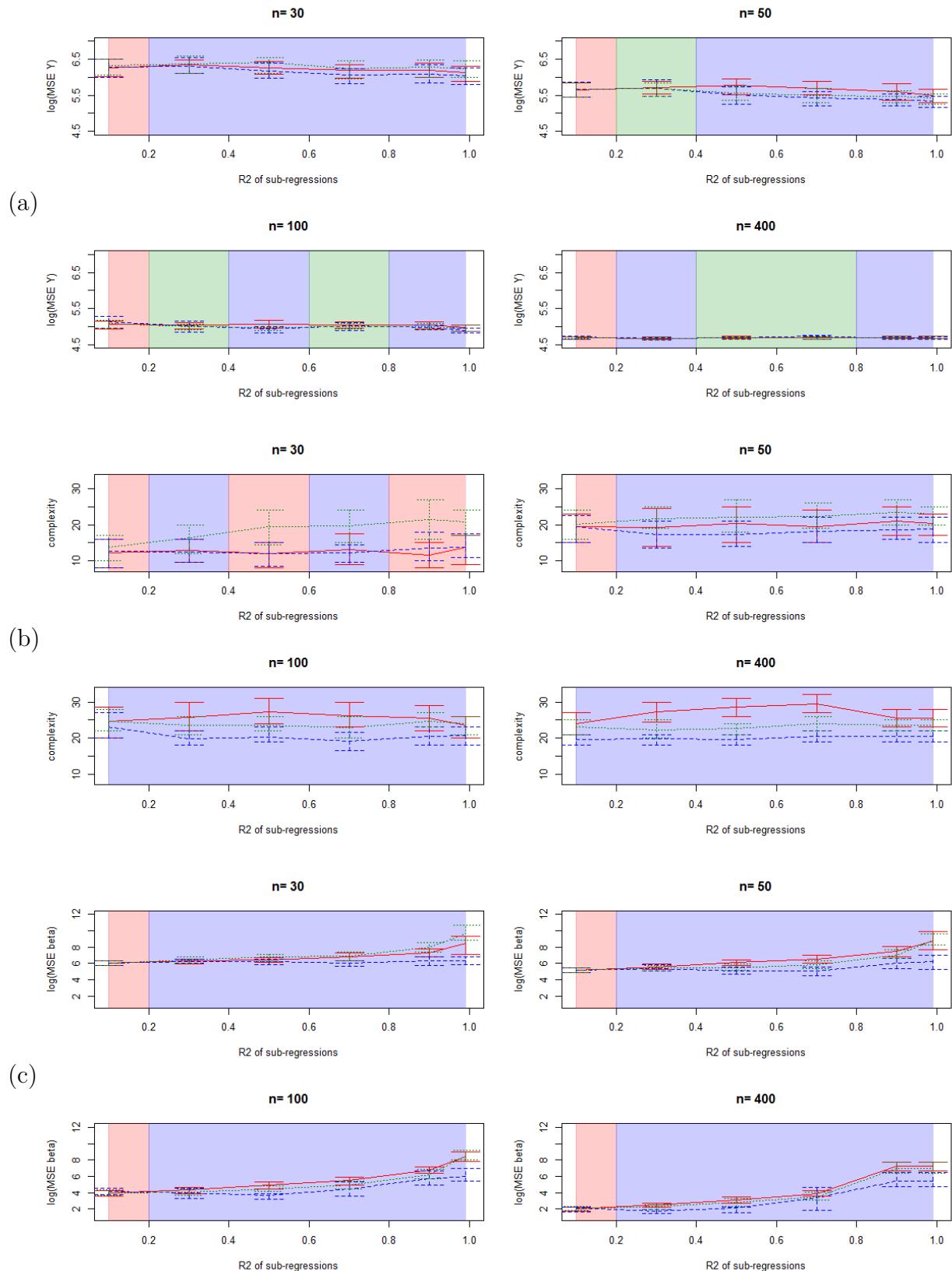


Figure 9.9: Comparison of the MSE on $\hat{\mathbf{Y}}$ (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends only on covariates in X_f

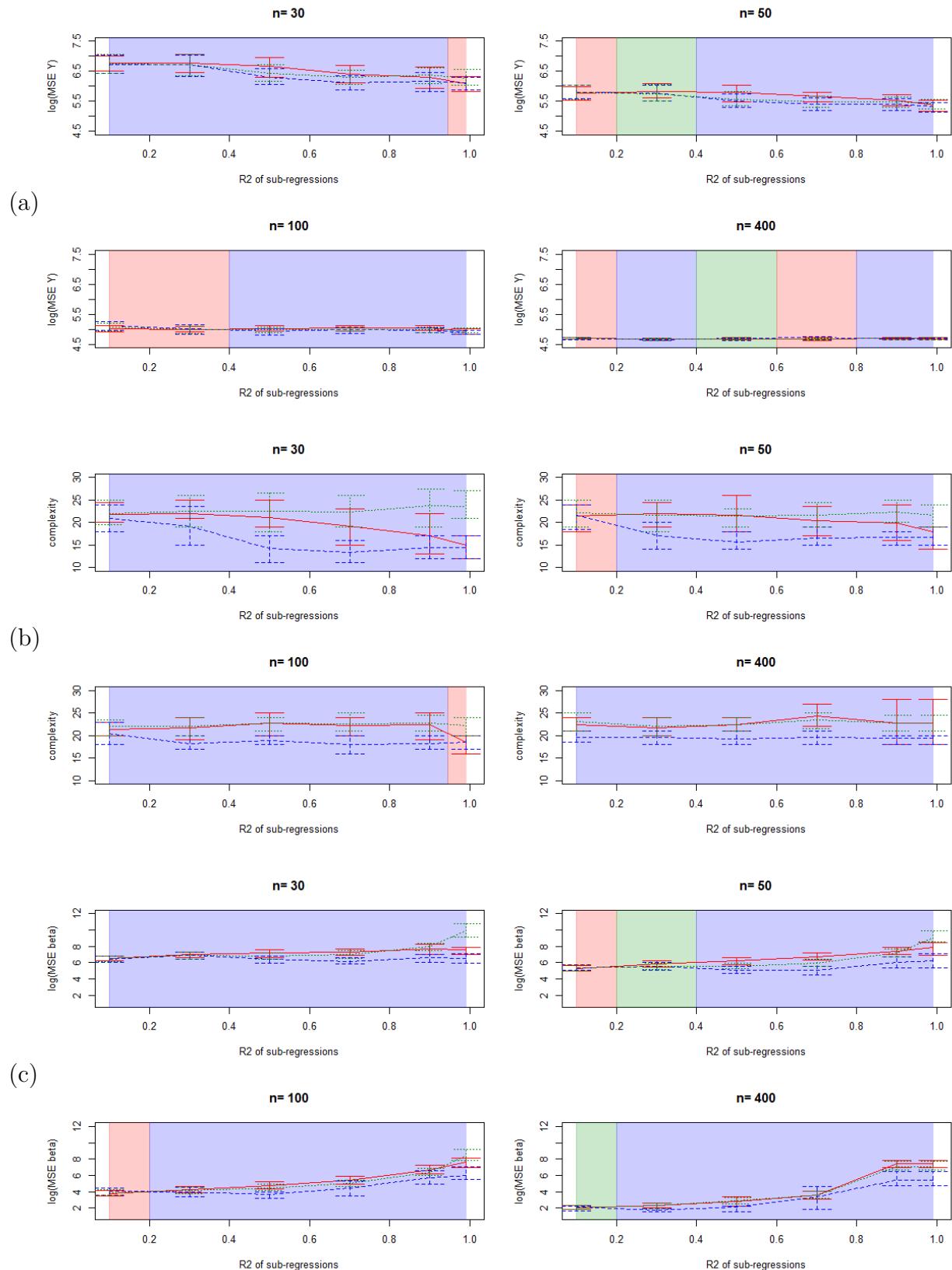


Figure 9.10: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when Y depends only on covariates in X_f

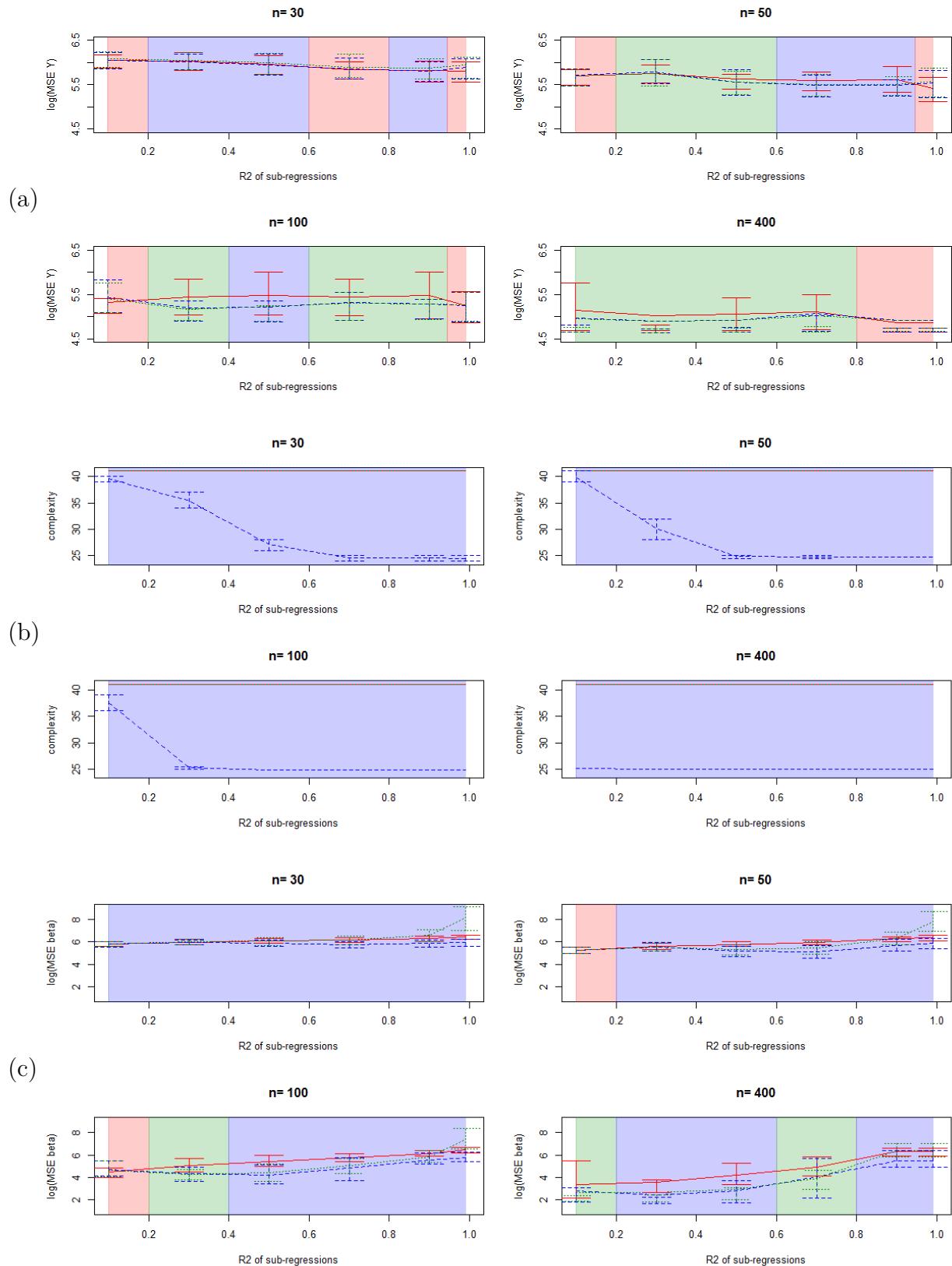


Figure 9.11: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Ordinary Least Squares when Y depends only on covariates in X_r

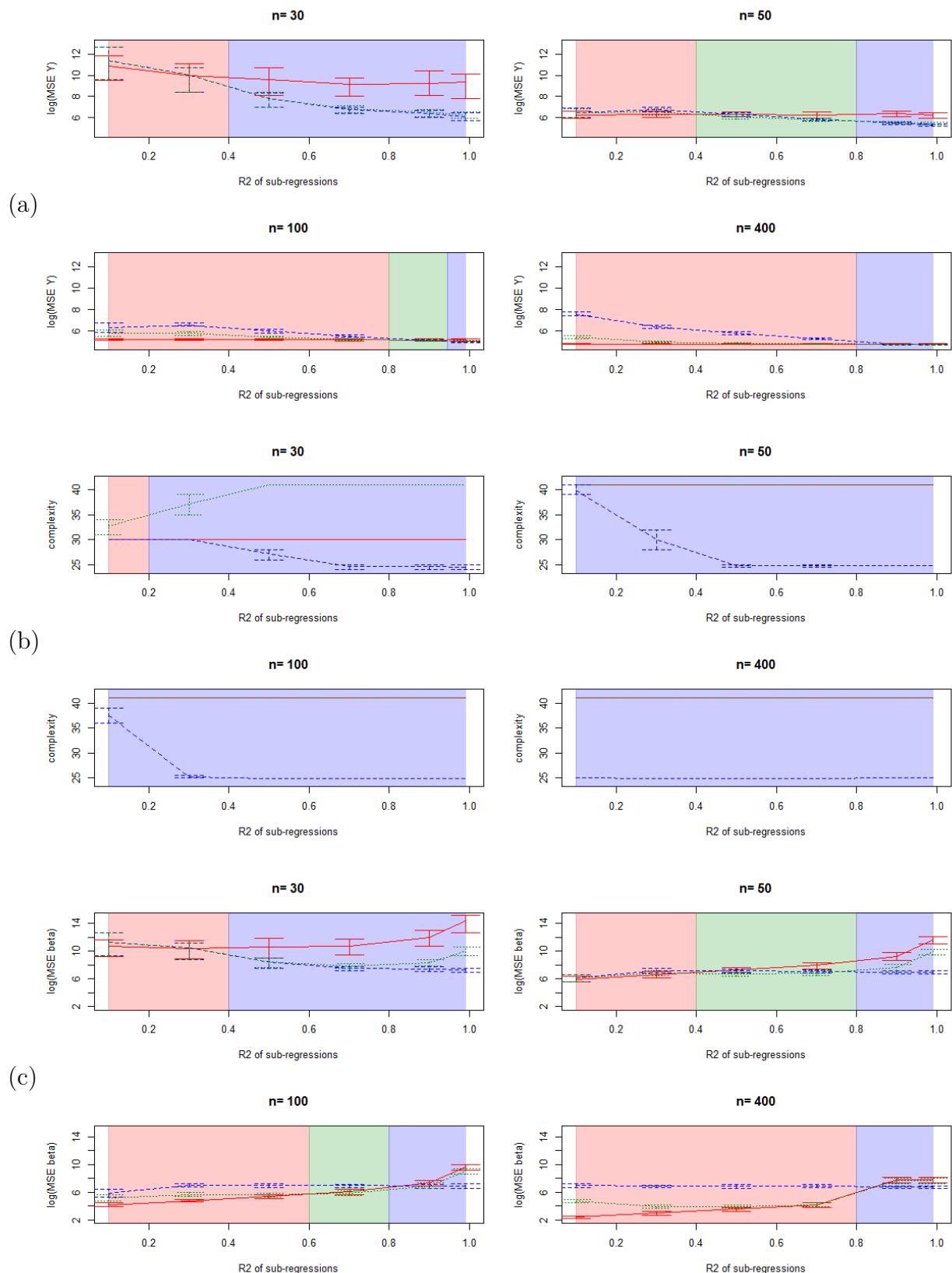


Figure 9.12: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

LASSO when Y depends only on covariates in X_r

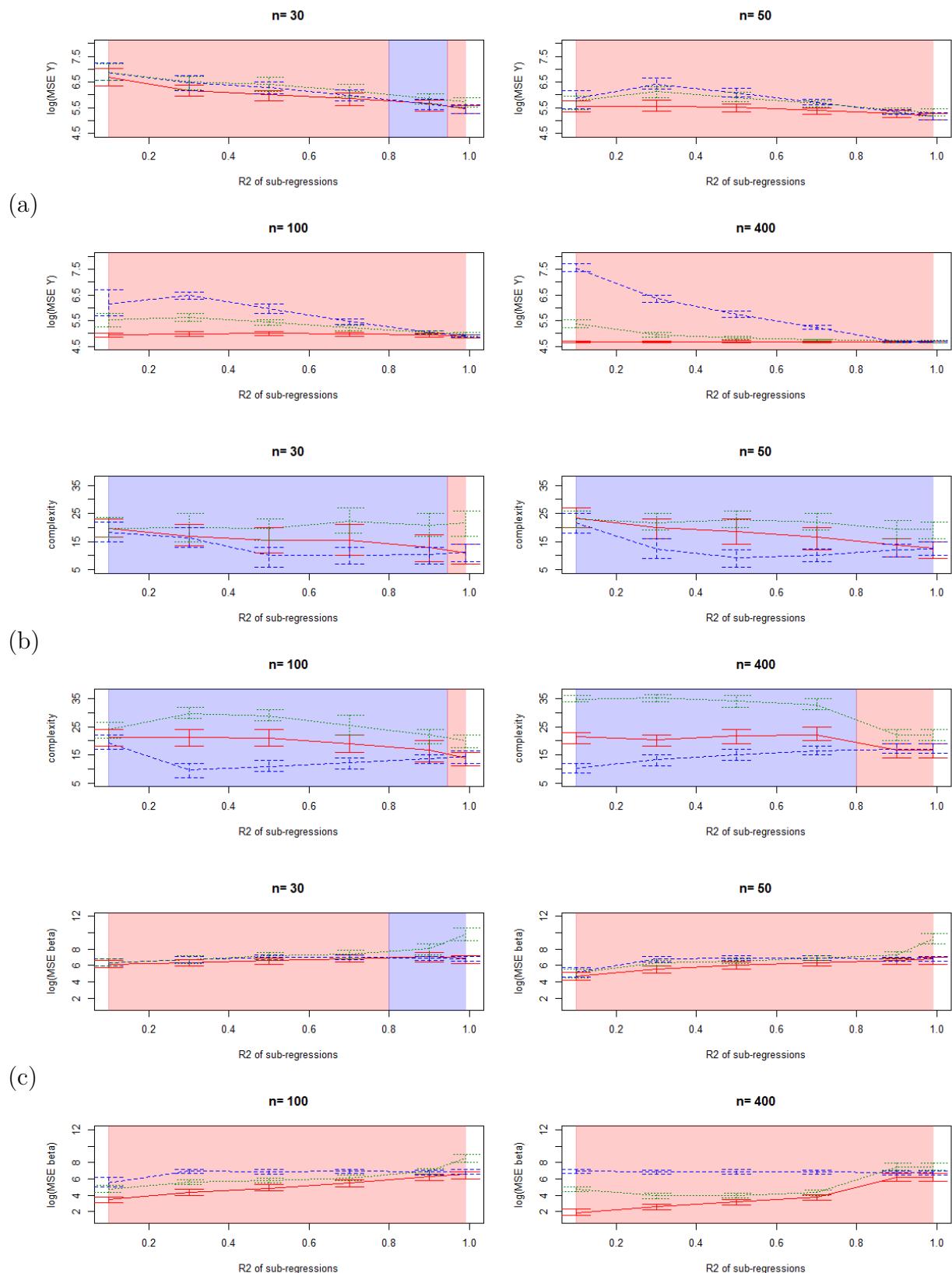


Figure 9.13: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when Y depends only on covariates in X_r

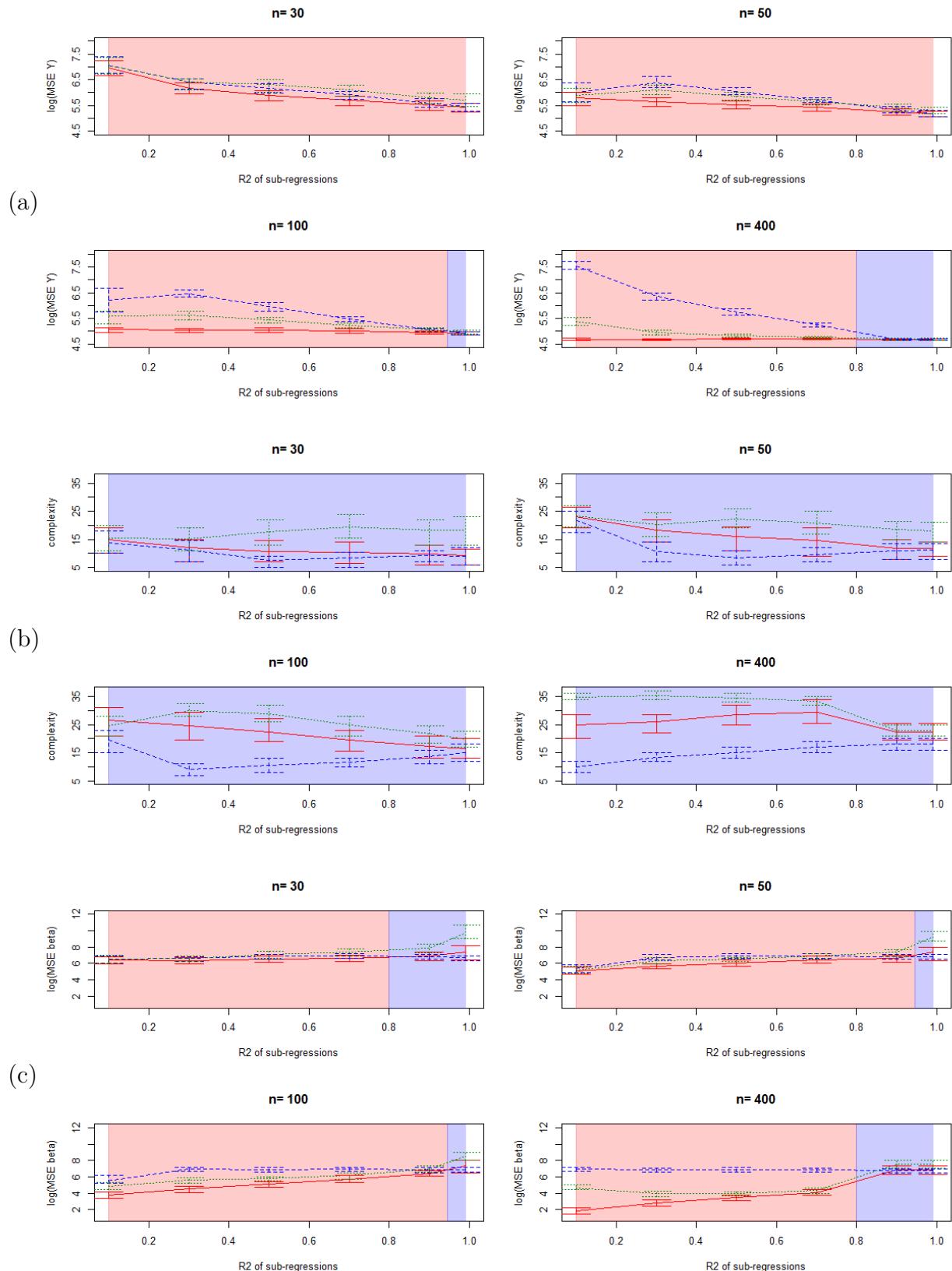


Figure 9.14: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends only on covariates in X_r

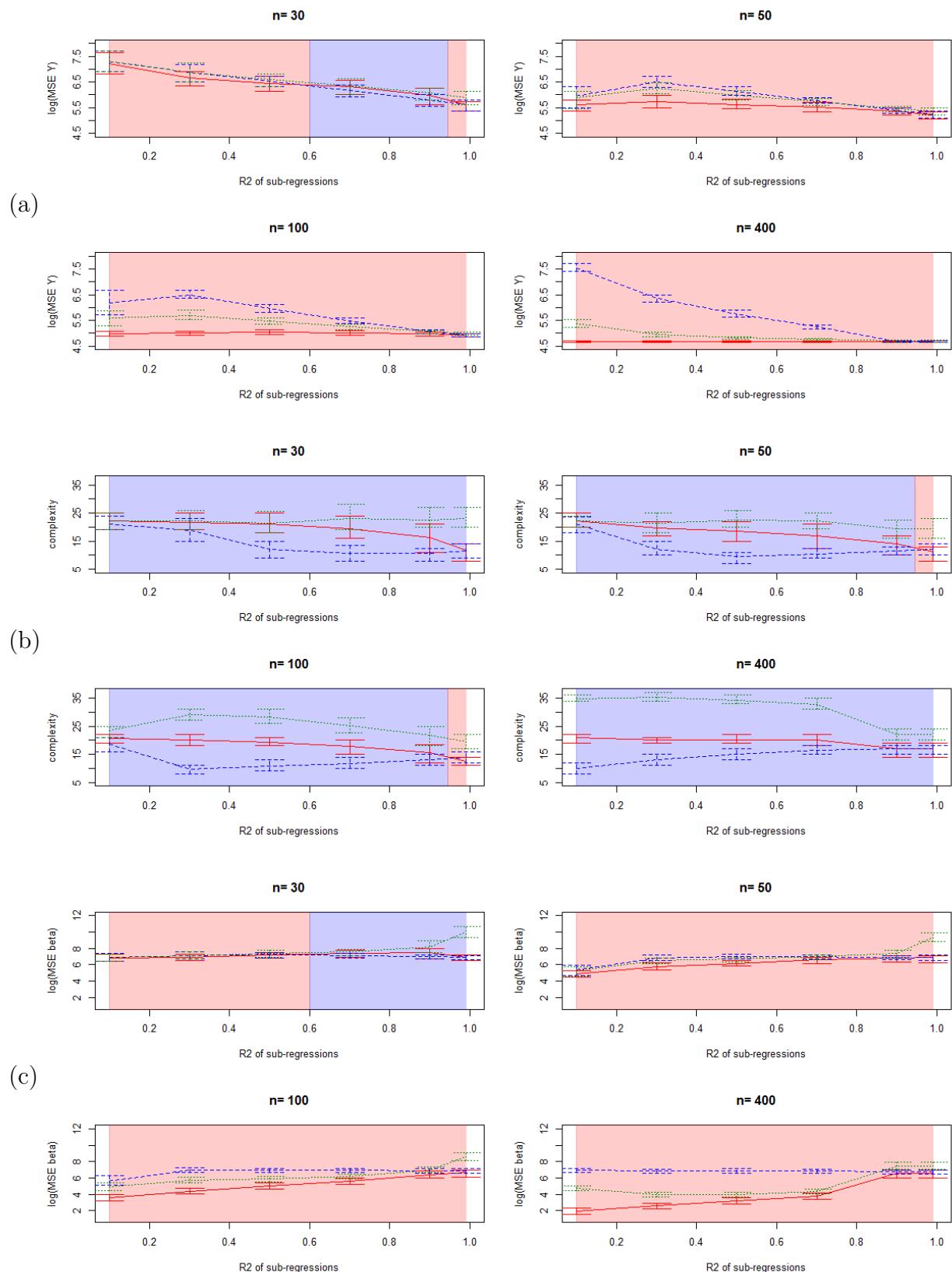


Figure 9.15: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when Y depends only on covariates in X_r

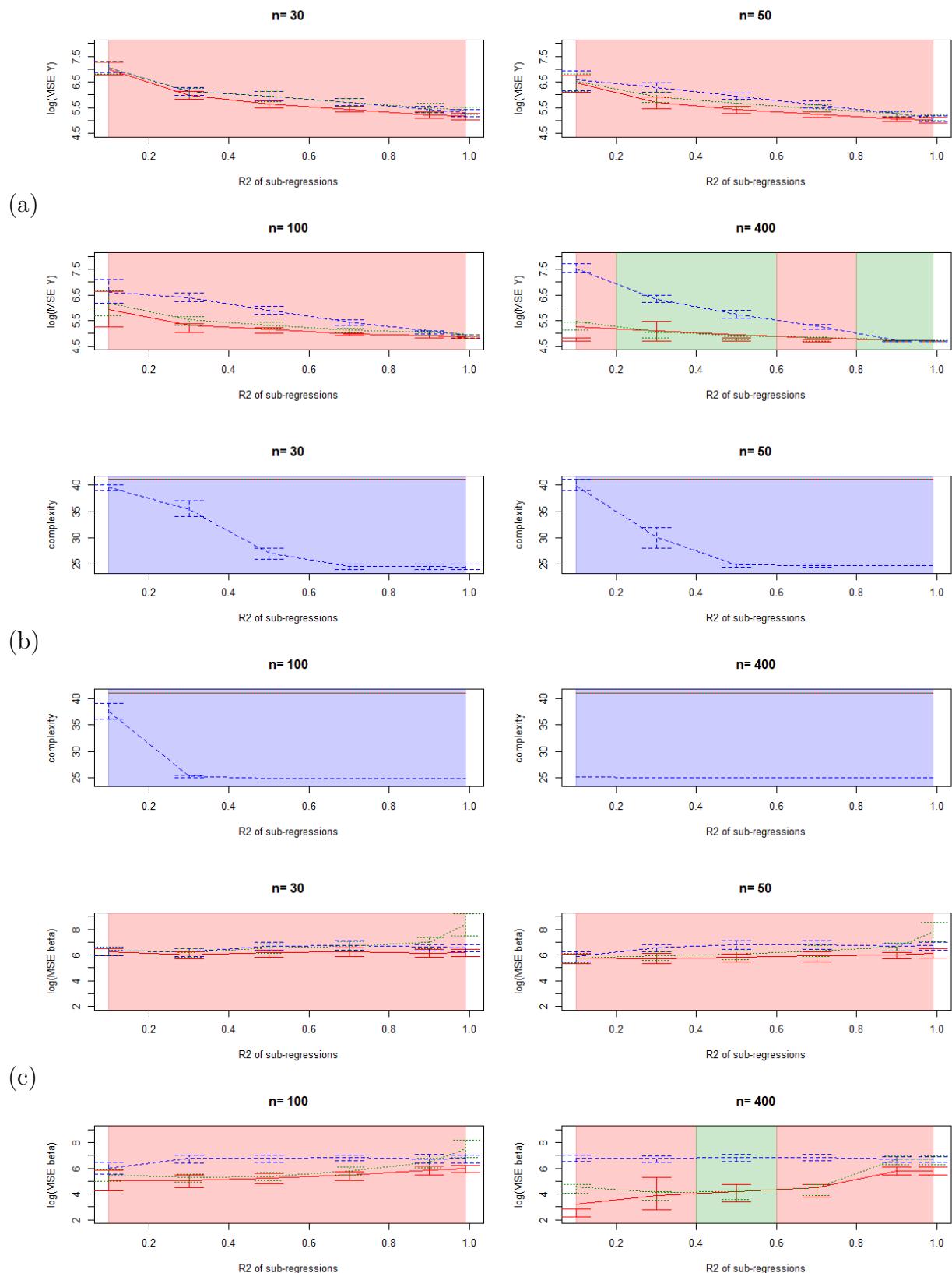


Figure 9.16: Comparison of the MSE on \hat{Y} (a), complexities (b) and MSE on $\hat{\beta}$ (c), red=classical (complete) model, blue=marginal model

Chapter 10

Missing values

Abstract: The full generative model defined on \mathbf{X} with \mathbf{S} gives the conditional distribution of missing values. Thus we are able to make multiple imputations based on the sub-regression structure. The full generative model also helps to manage missing values in the MCMC with the ability to compute BIC_* only on the observed values.

10.1 State of the art

Real datasets often have missing values and it is a very recurrent issue in industry. Here we suppose that missing values are Missing Completely At Random (MCAR) but other missing-data mechanisms do exist for missing values. For example, Missing values can depends on the observed values and then we say that they are Missing At Random (MAR), Missing value can also simply be not missing at random. Many methods does exist to manage such problems in a regression context[Little, 1992]:

1. Complete-Case analysis is a listwise deletion of missing values that may lead to no results if no individual is complete.
2. Available-Case analysis methods use the largest sets of available cases for estimating individual parameters so it does not remove any additional value. But individual estimation of the parameters leads to bad results when the covariates are highly correlated [Haitovsky, 1968].
3. Imputation method are various, from imputation by the mean to conditional imputation based on \mathbf{X} or based on \mathbf{X} and \mathbf{Y} . These methods are often used with weighted least-squares to minimize the influence of imputed values.
4. Methods based on a full generative model, using the joint distribution to estimate the regression parameters or to make multiple imputations.

We have a full generative model on \mathbf{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 \mathbf{X} .

This chapter is a perspective of this thesis that we have already started to work on, but missing values is a wide subject and we only propose some results.

In the followings we note \mathbf{M} the $n \times d$ binary matrix indicating whether a value is missing or not: $\mathbf{M}_{i,j} = 1$ if \mathbf{X}_i^j is missing, 0 else. We also note $x_{i,j}$ the i^{th} individual of the j^{th} covariate. We define $\mathbf{X}_M = (\mathbf{X}_{1,M}, \dots, \mathbf{X}_{n,M})$ the n -uple of the vectors $\mathbf{X}_{i,M} = (x_{i,j})_{j \in \{1, \dots, d\}}$ of the missing values for individual i , $\mathbf{X}_O = (\mathbf{X}_{1,O}, \dots, \mathbf{X}_{n,O})$ the n -uple of the vectors $\mathbf{X}_{i,O} = (x_{i,j})_{j \in \{1, \dots, d\}}$ of the observed values for individual i , $\mathbf{d}_M = (d_{1,M}, \dots, d_{n,M})$ where the $d_{i,M} = |\mathbf{X}_{i,M}|$ are the number of missing values for individual i and $d_M = \sum_{i=1}^n d_{i,M}$ the total number of missing values in \mathbf{X} . We also define the three vectors $\mathbf{X}_{i,O}^{J_r} = (x_{i,l})_{l \in J_r}$, $\mathbf{X}_{i,O}^{J_f} = (x_{i,j})_{j \in J_f}$, and $\mathbf{X}_{i,O}^{J_p} = (x_{i,l})_{l \in J_p}$ to simplify the notations (that are a bit heavy) in the followings.

$\Theta = \{\boldsymbol{\mu}_X, \boldsymbol{\Sigma}_X\}$ stands for the parameters of the Gaussian mixture followed by \mathbf{X} . $\bar{\boldsymbol{\alpha}}$ is the $d \times d$ matrix of the sub-regression coefficients with $\alpha_{i,j}$ the coefficients associated to \mathbf{X}^i in the sub-regression explaining \mathbf{X}^j .

Maugis has developed another software [Maugis-Rabasseau et al., 2012] that manage missing values: **SelvarclustMV**. It relies on the assumption of Missing At Random (MAR) values and on the more constraining hypothesis that regressor covariates are completely observed (with our notations: $\forall j \in \{1, \dots, d_r\}, \forall k \in J_p^j, \forall 1 \leq i \leq n, \mathbf{M}_{i,k} = 0$). The model (Partition, sub-regression parameters and Gaussian mixtures parameters) is then estimated with an Expectation-Maximization algorithm using all the observed values (even incomplete individuals) without any imputation. Numerical results shows that **SelvarclustMV** is highly competitive compared to imputation methods.

Here again the algorithm used in **SelvarclustMV** stands in a Gaussian clustering context and is not exactly adapted to our situation. Estimation of \mathbf{S} and $\boldsymbol{\alpha}^*$ was inconclusive (see section 7.2.1) without missing values so the situation would not be improved with missing values. Moreover, we do not want to make any assumption on the position of the missing values. Thus we investigate the possibility to estimate $\bar{\boldsymbol{\alpha}}$ by Maximum Likelihood as **SelvarclustMV** does but without supposing that any covariate is fully observed.

10.2 Estimation of \mathbf{S} with missing values

10.2.1 Marginal likelihood

The first thing we do with \mathbf{X} in the CORREG process is to estimate \mathbf{S} . It is done by comparison of the BIC that rely on the likelihood. Because covariates in \mathbf{X}_f are orthogonal, complete-case estimation is equivalent to global estimation on the observed values, so we just use **Rmixmod** for each covariate on the observed values to obtain the observed likelihood of each marginal covariate. These likelihood are computed only once before the MCMC starts and are then used when needed to compute the BIC of a given candidate.

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. Each sub-regression is supposed to be parsimonious and might be estimated by a Complete-Case method if the number of missing values is not too high, or any other estimator. We will see later how to use the generative model and maximum likelihood instead. The first challenge is to compute the observed likelihood and then the BIC_* .

We start with the complete likelihood of \mathbf{X} :

$$L(\boldsymbol{\alpha}, \Theta, \mathbf{S}; \mathbf{X}) = \prod_{i=1}^n f(\mathbf{X}_i) = \prod_{i=1}^n \left[f(\mathbf{X}_i^{J_r} | \mathbf{X}_i^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) f(\mathbf{X}_i^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \right] \quad (10.1)$$

$$= \prod_{i=1}^n \left[\prod_{j \in J_r} f(x_{i,j} | \mathbf{X}_i^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \prod_{j \notin J_r} f(x_{i,j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \right] \quad (10.2)$$

$$= \prod_{i=1}^n \left[\prod_{j=1}^{d_r} f(x_{i,J_r^j} | \mathbf{X}_i^{J_p^j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \prod_{j \notin J_r} f(x_{i,j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \right] \quad (10.3)$$

$$(10.4)$$

To compute the BIC_* we use then the log-likelihood:

$$\mathcal{L}(\boldsymbol{\alpha}, \Theta, \mathbf{S}; \mathbf{X}) = \sum_{i=1}^n \left[\sum_{j=1}^{d_r} \log \left(f(x_{i,J_r^j} | \mathbf{X}_i^{J_p^j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \right) + \sum_{j \notin J_r} \log (f(x_{i,j}; \boldsymbol{\alpha}, \Theta, \mathbf{S})) \right] \quad (10.5)$$

When missing values occurs, we restrict the likelihood to the known values by integration on X_M .

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

$$L(\boldsymbol{\alpha}, \Theta, \mathbf{S}; \mathbf{X}_O) = \int_{\mathbf{X}_M} L(\boldsymbol{\alpha}, \Theta, \mathbf{S}; \mathbf{X}) d\mathbf{X}_M = \int_{\mathbb{R}^{d_M}} \sum_{k=1}^K \pi_k \Phi_k(\mathbf{X}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) d\mathbf{X}_M \quad (10.6)$$

$$= \sum_{k=1}^K \pi_k \int_{\mathbb{R}^{d_M}} \Phi_k(\mathbf{X}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) d\mathbf{X}_M \quad (10.7)$$

$$= \sum_{k=1}^K \pi_k \int_{\mathbb{R}^{d_M}} \prod_{i=1}^n \Phi_k(\mathbf{X}_i; \boldsymbol{\alpha}, \Theta, \mathbf{S}) d\mathbf{X}_M \quad (10.8)$$

$$= \sum_{k=1}^K \pi_k \prod_{i=1}^n \int_{\mathbb{R}^{d_{i,M}}} \Phi_k(\mathbf{X}_{i,M}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) d\mathbf{X}_{i,M} \quad (10.9)$$

$$= \sum_{k=1}^K \pi_k \prod_{i=1}^n \Phi_k(\mathbf{X}_{i,O}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \quad (10.10)$$

$$= \sum_{k=1}^K \pi_k \Phi_k(\mathbf{X}_O; \boldsymbol{\alpha}, \Theta, \mathbf{S}) = f(\mathbf{X}_O; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \quad (10.11)$$

To compute this likelihood, we will use the decomposition

$$L(\boldsymbol{\alpha}, \Theta, \mathbf{S}; \mathbf{X}_O) = f(\mathbf{X}_O; \boldsymbol{\alpha}, \Theta, \mathbf{S}) = \prod_{i=1}^n f(\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) f(\mathbf{X}_{i,O}^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \quad (10.12)$$

$$= \prod_{i=1}^n f(\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \prod_{\substack{j \in J_f \\ M_{i,j}=0}} f(x_{i,j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \quad (10.13)$$

with $\forall(i, j)$ with $M_{i,j} = 0$ and $j \notin J_r$:

$$f(x_{i,j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) = \sum_{k=1}^{K_j} \pi_{j,k} \Phi_k(x_{i,j}; \mu_{j,k}, \Sigma_{j,k}) \quad (10.14)$$

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 $\mathbf{M}_{i,j} = 0$ and $j \in \{1, \dots, d_r\}$:

$$f(x_{i,J_r^j} | \mathbf{X}_{i,O}^{J_p^j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) = \sum_{k=1}^{K_{i,J_r^j}} \pi_{i,J_r^j, k} \Phi(x_{i,J_r^j}; \mu_{i,J_r^j, k}, \Sigma_{i,J_r^j, k}) \text{ where} \quad (10.15)$$

$$\boldsymbol{\pi}_{i,J_r^j} = \bigotimes_{\substack{l \in J_p^j \\ \mathbf{M}_{i,l}=1}} \boldsymbol{\pi}_l \text{ and } K_{i,J_r^j} = |\boldsymbol{\pi}_{i,J_r^j}|, \quad (10.16)$$

$$\boldsymbol{\mu}_{i,J_r^j} = \sum_{\substack{l \in J_p^j \\ \mathbf{M}_{i,l}=0}} \bar{\boldsymbol{\alpha}}_{l,J_r^j} x_{i,l} + \bigoplus_{\substack{l \in J_p^j \\ \mathbf{M}_{i,l}=1}} \bar{\boldsymbol{\alpha}}_{l,J_r^j} \boldsymbol{\mu}_l \quad (10.17)$$

$$\boldsymbol{\Sigma}_{i,J_r^j} = \sigma_{J_r^j}^2 + \bigoplus_{\substack{l \in J_p^j \\ \mathbf{M}_{i,l}=1}} \bar{\boldsymbol{\alpha}}_{l,J_r^j}^2 \boldsymbol{\Sigma}_l \quad (10.18)$$

Where \bigotimes is the Kronecker product and and \bigoplus is the Kronecker sum with

$$A \bigoplus B = A \bigotimes \mathbf{I}_b + \mathbf{I}_a \bigotimes B$$

where A and B are two squares matrices of order a and b respectively and $\mathbf{I}_a, \mathbf{I}_b$ are the two associated identity matrices. This could be easily used for imputation of the missing values in \mathbf{X}_r knowing the parameters $\boldsymbol{\alpha}, \Theta$ and \mathbf{S} .

We note that we obtain a Gaussian when there is no missing value in J_p^j . But we see that $f(\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S})$ is not the product of the $f(x_{i,J_r^j} | \mathbf{X}_{i,O}^{J_p^j}; \boldsymbol{\alpha}, \Theta, \mathbf{S})$ if a same missing value occurs in distinct sub-regressions. Thus if each sub-regression is a distinct connex component then we can use (10.15) and we have

$$L(\boldsymbol{\alpha}, \Theta, \mathbf{S}; \mathbf{X}_O) = \prod_{i=1}^n \prod_{\substack{j=1 \\ \mathbf{M}_{i,J_r^j}=0}}^{d_r} f(x_{i,J_r^j} | \mathbf{X}_{i,O}^{J_p^j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \prod_{\substack{j \in J_f \\ \mathbf{M}_{i,j}=0}} f(x_{i,j}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \quad (10.19)$$

In such a case, computation is fast and easy. It is the case supposed in **SelvarclustMV**.

But for the general case we need to manage the dependencies implied by missing values in common covariates in the J_p^j .

$$L(\boldsymbol{\alpha}, \Theta, \mathbf{S}; \mathbf{X}_O) = \prod_{i=1}^n \sum_{k=1}^K \pi_k \Phi_k(\mathbf{X}_{i,O}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \quad (10.20)$$

$$= \prod_{i=1}^n \sum_{k=1}^K \pi_k \Phi_k(\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \Phi_k(\mathbf{X}_{i,O}^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \quad (10.21)$$

$$= \prod_{i=1}^n \sum_{k=1}^K \pi_k \Phi_k(\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) \prod_{\substack{j \in J_f \\ \mathbf{M}_{i,j}=0}} \Phi_k(x_{i,j}; \mu_{j,k}, \Sigma_{j,k}) \quad (10.22)$$

where

$$\boldsymbol{\pi} = (\pi_1, \dots, \pi_K) = \bigotimes_{j \in J_f} \boldsymbol{\pi}_j \text{ (Kronecker product)} \quad (10.23)$$

$$K = |\boldsymbol{\pi}| \quad (10.24)$$

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

$$\boldsymbol{\mu}_{\mathbf{X}^{J_f}} = \prod_{j \in J_f} \boldsymbol{\mu}_j \text{ (Cartesian product) and} \quad (10.25)$$

$$\boldsymbol{\sigma}_{\mathbf{X}^{J_f}} = \prod_{j \in J_f} \boldsymbol{\sigma}_j \text{ (Cartesian product).} \quad (10.26)$$

Then we can describe the parameters as following: $\forall 1 \leq i \leq n, \forall 1 \leq k \leq K$ we have the conditional distribution of the sub-regressions

$$\Phi_k(\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f}; \boldsymbol{\alpha}, \Theta, \mathbf{S}) = \Phi_k(\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f}; \boldsymbol{\mu}_{\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f, k}}, \boldsymbol{\Sigma}_{\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f, k}}) \quad (10.27)$$

where the mean vector is

$$\boldsymbol{\mu}_{\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f, k}} = \boldsymbol{\mu}_{\mathbf{X}_{i,O}^{J_r, k}} + \boldsymbol{\Sigma}_{\mathbf{X}_{i,O}^{J_r}, \mathbf{X}_{i,O}^{J_f, k}} (\boldsymbol{\Sigma}_{\mathbf{X}_{i,O}^{J_f}, \mathbf{X}_{i,O}^{J_f, k}})^{-1} ((\mathbf{X}_{i,O}^{J_f})' - \boldsymbol{\mu}_{\mathbf{X}_{i,O}^{J_f, k}}) \quad (10.28)$$

and the variance is

$$\boldsymbol{\Sigma}_{\mathbf{X}_{i,O}^{J_r} | \mathbf{X}_{i,O}^{J_f, k}} = \boldsymbol{\Sigma}_{\mathbf{X}_{i,O}^{J_r}, \mathbf{X}_{i,O}^{J_r, k}} - \boldsymbol{\Sigma}_{\mathbf{X}_{i,O}^{J_r}, \mathbf{X}_{i,O}^{J_f, k}} (\boldsymbol{\Sigma}_{\mathbf{X}_{i,O}^{J_f}, \mathbf{X}_{i,O}^{J_f, k}})^{-1} \boldsymbol{\Sigma}_{\mathbf{X}_{i,O}^{J_f}, \mathbf{X}_{i,O}^{J_r, k}}. \quad (10.29)$$

We can then describe each part of these formulas for each $1 \leq i \leq n$ and $1 \leq k \leq K$:

The mean vector of observed regressed covariates: $\forall j \in \{1, \dots, d_r\}$ with $\mathbf{M}_{i,J_r^j} = 0$:

$$\boldsymbol{\mu}_{\mathbf{X}_{i,J_r^j, k}} = \sum_{l \in J_p^j} \bar{\boldsymbol{\alpha}}_{l,J_r^j} \mu_{l,k} \quad (10.30)$$

The variance of observed regressed covariates: $\forall j \in \{1, \dots, d_r\}$ with $\mathbf{M}_{i,J_r^j} = 0$

$$\text{Var}_k(x_{i,J_r^j}) = \sigma_j^2 + \sum_{l \in J_p^j} \bar{\boldsymbol{\alpha}}_{l,J_r^j}^2 \sigma_{\mathbf{X}^l, k}^2 \quad (10.31)$$

The variance of observed regressors: $\forall j \in J_f$ with $\mathbf{M}_{i,j} = 0$

$$\text{Var}_k(x_{i,j}) = \sigma_{X^j, k}^2 \quad (10.32)$$

The covariance between two regressed covariates with common regressors:

$\forall j_1 \in \{1, \dots, d_r\}, j_2 \in \{1, \dots, d_r\}, J_p^{j_1} \cap J_p^{j_2} \neq \emptyset$ with $\mathbf{M}_{i,J_r^{j_1}} = \mathbf{M}_{i,J_r^{j_2}} = 0$

$$\text{Cov}_k(x_{i,J_r^{j_1}}, x_{i,J_r^{j_2}}) = \sum_{l \in J_p^{j_1} \cap J_p^{j_2}} \bar{\boldsymbol{\alpha}}_{l,J_r^{j_1}} \bar{\boldsymbol{\alpha}}_{l,J_r^{j_2}} \sigma_{\mathbf{X}^l, k}^2 \quad (10.33)$$

The covariance between two regressed covariates without any common regressors:

$\forall j_1 \in \{1, \dots, d_r\}, j_2 \in \{1, \dots, d_r\}, J_p^{j_1} \cap J_p^{j_2} = \emptyset$ with $\mathbf{M}_{i,J_r^{j_1}} = \mathbf{M}_{i,J_r^{j_2}} = 0$

$$\text{Cov}_k(x_{i,J_r^{j_1}}, x_{i,J_r^{j_2}}) = 0 \quad (10.34)$$

The covariance between two regressors is always zero:

$\forall j_1 \in J_f, j_2 \in J_f$ with $\mathbf{M}_{i,j_1} = \mathbf{M}_{i,j_2} = 0$

$$\text{Cov}_k(x_{i,j_1}, x_{i,j_2}) = 0 \quad (10.35)$$

The covariance between a regressed covariate and one of its regressors:

$\forall j_1 \in \{1, \dots, d_r\}, j_2 \in J_p^{j_1}$ with $\mathbf{M}_{i,J_r^{j_1}} = \mathbf{M}_{i,j_2} = 0$

$$\text{Cov}_k(x_{i,J_r^{j_1}}, x_{i,j_2}) = \bar{\boldsymbol{\alpha}}_{j_2, J_r^{j_1}} \sigma_{X^{j_2}, k}^2 \quad (10.36)$$

The covariance between a regressed covariate and covariate neither regressing it nor regressed is zero by hypothesis 1: $\forall j_1 \in \{1, \dots, d_r\}, j_2 \notin J_p^{j_1} \cup J_r$ with $\mathbf{M}_{i,J_r^{j_1}} = \mathbf{M}_{i,j_2} = 0$

$$\text{Cov}_k(x_{i,J_r^{j_1}}, x_{i,j_2}) = 0 \quad (10.37)$$

Then we are able to maximize this likelihood to find $\boldsymbol{\alpha}$.

10.2.2 Likelihood computation optimized

The main problem with the likelihood in its global form (10.22) 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 Σ can be stored and used back in each iteration for a given structure \mathbf{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 (10.15) elsewhere. We just need to compute the row-sums of the adjacency matrix \mathbf{G} or to search for redundancy in J_f and then if there is no redundancy or if $\forall j$ redundant we have $\sum_{i=1}^n \mathbf{M}_{i,j} = 0$ then we can use the simplified form of the likelihood given in (10.19). 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 evaluate 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 for BIC

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 \mathbf{X}^j with $j \in J_r$, the number of parameters is not the same for each individual depending whether or not $\mathbf{M}_{i,j} = 0$. But the penalty (for 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.

$$k_j = \frac{1}{n} \sum_{i=1}^n k_{i,j} \quad (10.38)$$

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

$$-2 \log \mathbb{P}(\mathbf{X} | \mathbf{S}) \approx BIC = -2\mathcal{L}(\mathbf{X}, \mathbf{S}, \Theta) + |\Theta| \log(n) \quad (10.39)$$

$$= -2\mathcal{L}(\mathbf{X}, \mathbf{S}, \Theta) + \left(\sum_{j=1}^d k_j \right) \log(n) \quad (10.40)$$

Thus if a structure is only touched by one missing value the penalty will be smaller than if the same structure has more missing values implied. Another way would be to use RIC (see [Foster and George, 1994]) so the penalty does not depend on n but is associated with $\log(d)$.

10.3 EM Algorithm to estimate α with a given \mathbf{S}

Sometimes the maximization of a likelihood is too complex to be solved 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 $\boldsymbol{\theta}^{(q)}$.

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(q)}) = E_{Z|\mathbf{X},\boldsymbol{\theta}^{(q)}}[\log L(\boldsymbol{\theta}; \mathbf{X}, Z)] \quad (10.41)$$

- The Maximization step that maximizes:

$$\boldsymbol{\theta}^{(q+1)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(q)}) \quad (10.42)$$

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 set 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].

10.3.1 Stochastic EM

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 (10.5) is not linear so a simple Expectation-Maximization (EM) would be difficult to compute.

Another method would be to estimate the $\boldsymbol{\alpha}_j$ with OLS applied on sub-matrix of $(\mathbf{X}^{J_r^j}, \mathbf{X}^{J_p^j})$ without missing values (complete-case method). Small sub-regressions may increase the probability to find such sub-matrices. Moreover, small sub-regression have only few parameters and can be estimated even with only a small number of individuals.

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 complete-case method (sparse structure) or using imputed values in \mathbf{X} and then OLS. At iteration (h),

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

M step: We estimate

$$\boldsymbol{\alpha}^{(h+1)} = \operatorname{argmax}_{\boldsymbol{\alpha}} E [\mathcal{L}(\mathbf{X}|\boldsymbol{\alpha}, \mathbf{S}, \Theta)] \quad (10.43)$$

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.

10.3.2 Stochastic imputation by Gibbs sampling

Gibbs sampling [Casella and George, 1992] is a special case of Markov Chain Monte Carlo algorithm [Gilks et al., 1996, Chib and Greenberg, 1995, Roberts and Rosenthal, 2001] that allows to sample from a complex d -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 $\mathbf{X}^{(0)}$ and then for each iteration (q) and successively each variable $x_j^{(q+1)}$ to draw, it draws from $\mathbb{P}(x_j|x_1^{(q+1)}, \dots, x_{j-1}^{(q+1)}, x_{j+1}^{(q)}, \dots, x_d^{(q)})$ using the most recent drawn values each time.

Gibbs sampling method can be used to generate the missing values at the SE step. We note Z the set of the $Z_{i,j}$ indicating the component from which $x_{i,j}$ (the i^{th} individual of the j^{th} covariate) is generated.

Initialisation: \mathbf{X}_M are imputed by the marginal means of drawn independently from the univariate distribution estimated for the MCMC (by `Rmixmod` for example). all the $Z_{i,j}$ are randomly set based on the $t_{i,j,k}$ as explained below (we just need those of \mathbf{X}_f).

Iteration: At each iteration of the Gibbs sampler we first make successive imputation on the missing values:

Regressed covariates: $\forall 1 \leq i \leq n, \forall j \in \{1, \dots, d_r\}, \mathbf{M}_{i,J_r^j} = 1$: x_{i,J_r^j} is generated according to

$$\mathbb{P}(x_{i,J_r^j} | \mathbf{X}_i \setminus x_{i,J_r^j}, Z; \boldsymbol{\alpha}^{(h)}, \Theta, \mathbf{S}) = \mathbb{P}(x_{i,J_r^j} | \mathbf{X}_i^{J_r^j}; \boldsymbol{\alpha}_j^{(h)}, \Theta, S) \quad (10.44)$$

$$= \Phi(\mathbf{X}_i^{J_r^j} \boldsymbol{\alpha}_j^{(h)}; \sigma_j^2) \text{ Gaussian} \quad (10.45)$$

And for regressors: $\forall (i, j) \in \{1, \dots, n\} \times J_f, \mathbf{M}_{i,j} = 1$: $x_{i,j}$ is generated according (using Schur complement and Gaussian conditional distribution) to

$$\mathbb{P}(x_{i,j} | \mathbf{X}_i \setminus x_{i,j}, Z; \boldsymbol{\alpha}^{(h)}, \Theta, \mathbf{S}) = \mathbb{P}(x_{i,j} | \mathbf{X}_{i,\bar{j}}, Z; \boldsymbol{\alpha}^{(h)}, \Theta, \mathbf{S}) \quad (10.46)$$

$$= \Phi(\mu_{j|Z_{i,j}} + \Sigma_{j,X_{i,\bar{j}}|Z_i} \Sigma_{X_{i,\bar{j}}|Z_i}^{-1} (X_{i,\bar{j}} - \boldsymbol{\mu}_{X_{i,\bar{j}}|Z_i}); \sigma_{j|Z_{i,j}}^2 - \Sigma_{j,X_{i,\bar{j}}|Z_i} \Sigma_{X_{i,\bar{j}}|Z_i}^{-1} \Sigma'_{j,X_{i,\bar{j}}|Z_i}) \quad (10.47)$$

Where all the values needed here were described above for the likelihood computation with $\mathbf{X}_{i,\bar{j}} = \mathbf{X}_i \setminus x_{i,j}$ and $Z_i = \{Z_{i,j} : j = 1, \dots, d\}$.

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

$$\mathbb{P}(Z_{i,j} | \mathbf{X}, Z \setminus Z_{i,j}; \Theta, \boldsymbol{\alpha}, \mathbf{S}) = P(Z_{i,j} | \mathbf{X}_i, Z \setminus Z_{i,j}; \Theta, \boldsymbol{\alpha}, \mathbf{S}) = \mathcal{M}(t_{i,j,1}, \dots, t_{i,j,K_j}) \quad (10.48)$$

$$\text{where } t_{i,j,k} = \frac{\pi_{j,k} \Phi(x_{i,j}; \mu_{j,k}, \sigma_{j,k}^2)}{\sum_{l=1}^{K_j} \pi_{j,l} \Phi(x_{i,j}; \mu_{j,l}, \sigma_{j,l}^2)} \quad (10.49)$$

We see that $Z_{i,j}$ are not used if there is no missing values in \mathbf{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 Stochastic EM, we

do not need to draw Z .

Instead of using long chain for each Gibbs, we can use small chains because Stochastic EM 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 (Stochastic EM) 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 estimate the likelihood. We can use a Gibbs algorithm to estimate the likelihood:

$$\mathbb{P}(\mathbf{X}_O; \Theta, \mathbf{S}, \boldsymbol{\alpha}) = \sum_{Z \in \mathcal{Z}} \int_{\mathbf{X}_M} \frac{\mathbb{P}(\mathbf{X}_M, Z, \mathbf{X}_O; \Theta, \boldsymbol{\alpha}, \mathbf{S})}{\mathbb{P}(\mathbf{X}_M, Z | \mathbf{X}_O; \Theta, \boldsymbol{\alpha}, \mathbf{S})} \mathbb{P}(\mathbf{X}_M, Z | \mathbf{X}_O; \Theta, \boldsymbol{\alpha}, \mathbf{S}) d\mathbf{X} \quad (10.50)$$

$$\approx \frac{1}{Q} \sum_{q=1}^Q \frac{\mathbb{P}(\mathbf{X}_M^{(q)}, \mathbf{X}_O, Z^{(q)}; \Theta, \boldsymbol{\alpha}, \mathbf{S})}{\mathbb{P}(\mathbf{X}_M^{(q)} | \mathbf{X}_O, Z^{(q)}; \Theta, \boldsymbol{\alpha}, \mathbf{S})} \text{ by the law of large numbers} \quad (10.51)$$

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

$$\mathbb{P}(\mathbf{X}_O; \Theta, \mathbf{S}, \boldsymbol{\alpha}) \approx \frac{1}{Q} \sum_{q=1}^Q \frac{\mathbb{P}(\mathbf{X}_M^{(q)}, \mathbf{X}_O, Z^{(q)}; \Theta, \boldsymbol{\alpha}^{(q)}, \mathbf{S})}{\mathbb{P}(\mathbf{X}_M^{(q)} | \mathbf{X}_O, Z^{(q)}; \Theta, \boldsymbol{\alpha}^{(q)}, \mathbf{S})} \quad (10.52)$$

10.4 Missing values in the main regression

The easier way to manage missing values in the main regression would be to draw missing values and then use classical methods. Imputation could be done with the Stochastic EM described above, with the possibility to repeat the estimation of the coefficients of regression a few times (with distinct imputations) 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 imputation 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.

10.5 Numerical results on simulated datasets

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 Stochastic EM to those obtain by classical OLS after imputation of the missing values by the marginal empirical means. Here $d = 40$ 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 \mathbf{S} . Initial value of $\boldsymbol{\alpha}$ for the Stochastic EM was the result of the method using imputation by the empirical mean. Only 10 iterations for the Stochastic EM after 2 warming steps with only 1 iteration for the Gibbs at each step.

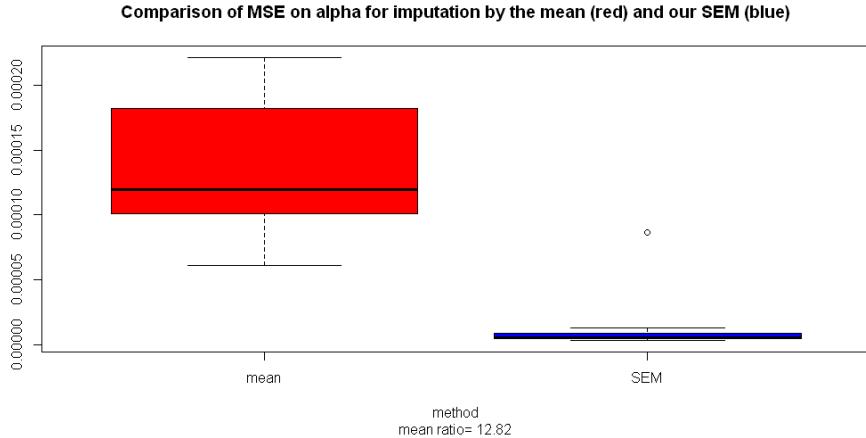


Figure 10.1: MSE on $\boldsymbol{\alpha}$ is significantly lower and with smaller variance with our Stochastic EM than with imputation by the mean.

We see (Figure 10.1) that our Stochastic EM is nearly 13 times more efficient in mean than 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 noise.

10.5.2 Multiple imputation

We have then imputed missing values in \mathbf{X}_r by using the corresponding sub-regressions after $\boldsymbol{\alpha}$ has been estimated by the Stochastic EM. Missing values in \mathbf{X}_f are estimated by the mean of 50 Gibbs iterations after the Stochastic EM and 2 warming steps of the Gibbs. Figure 10.2 shows the significative gain in MSE produced by our method.

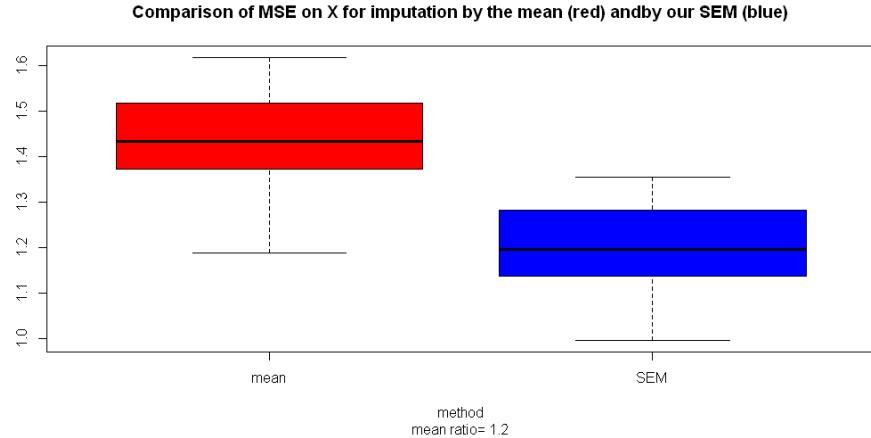


Figure 10.2: MSE on $\hat{\mathbf{X}}$ is significantly lower when using our Stochastic EM than with imputation by the mean.

10.5.3 Results on the main regression

We use the previously imputed $\hat{\mathbf{X}}$ to estimate \mathbf{Y} with $\beta = (1, \dots, 1)$ and $\sigma_Y = 10$.

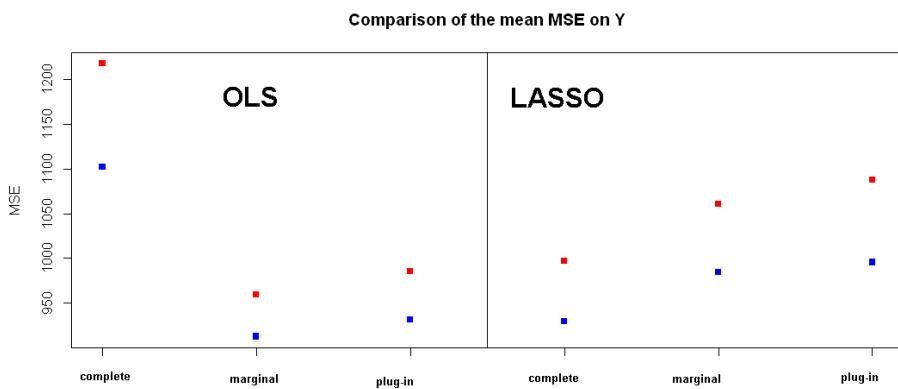


Figure 10.3: MSE on $\hat{\mathbf{Y}}$ are lower when using our Stochastic EM (blue) than with imputation by the mean (red) for the three model (complete, marginal and plug-in) using OLS or LASSO

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 Stochastic EM is more robust, but the Gibbs do not give satisfying results at the moment. 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 α and makes possible to find \mathbf{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 \mathbf{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 \mathbf{S} and use previously computed values of α as initial value for the Stochastic EM.

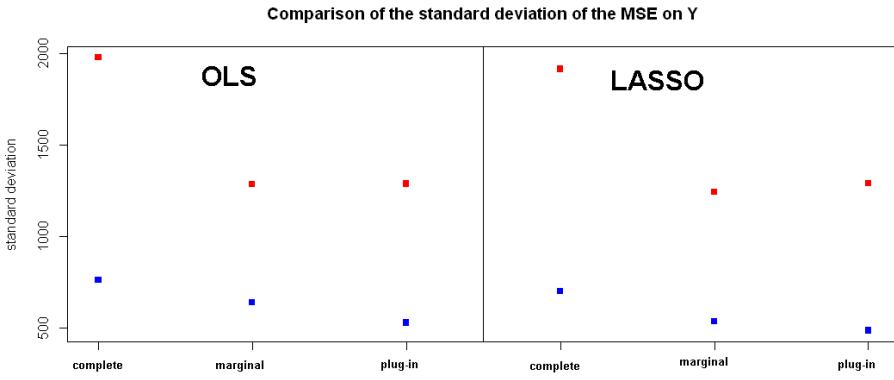


Figure 10.4: our Stochastic EM (blue) provides more robust results than imputation by the mean (red) but the variances are still too wide.

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.

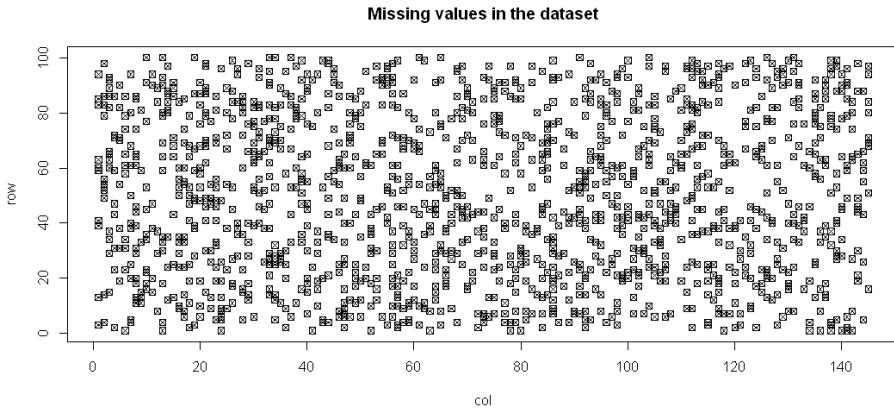


Figure 10.5: Graphical representation of the dataset with 10% of missing values

We see in figure 10.6 that our Stochastic EM 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 first steps of this huge perspective, but these first results are encouraging and make us feel like this orientation is worthy to be followed.

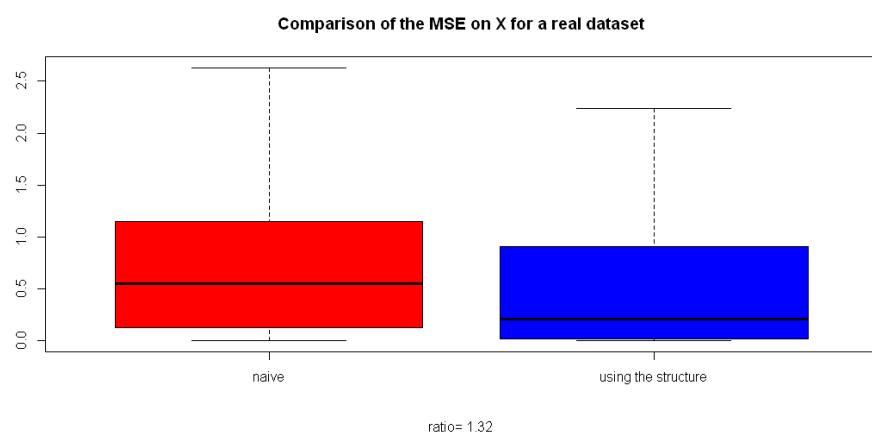


Figure 10.6: MSE on $\hat{\mathbf{X}}$ is 1.32 times lower in mean when using our method (blue) than with imputation by the mean (red).

Chapter 11

Conclusion and perspectives

11.1 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.

11.2 Perspectives

11.2.1 Non-linear regression

Polynomial regression, *etc.* might be improved by a method like this. Logistic regression [Hosmer and Lemeshow, 2000] is in fact a linear regression with a post-treatment on \mathbf{Y} so it is clear that we would obtain same phenomenon. It is an easy generalization of CORREG.

11.2.2 Pretreatment not only for regression

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

11.2.3 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).

11.2.4 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. Missing values were just introduced here and represent a consequent perspective.

11.2.5 Interpretation improvements

Ergonomy of the software should 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 get closer to this goal.

Regression mixture models

When the population studied can be decomposed in several classes of individuals, correlations between the covariates can depend on these classes. One perspective would be to search mixture sub-regressions [De Veaux, 1989] with some parameters and sub-regressions common to several classes and some other with distinct values.

Using \mathbf{Y} to estimate S

Like PLS regression, we could imagine to estimate S using also \mathbf{Y} to find a structure that will give the best results when applied on \mathbf{Y} . But it would probably give distinct structures according to the chosen model (marginal or plug-in). It would also require a criterion that could be the *BIC* of the combination of both S and the chosen regression on \mathbf{Y} or a MSE but with added computational cost if the LASSO (for example) has to be computed for each candidate structure in the MCMC.

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.
- [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.
- [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.
- [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.

- [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.
- [Cottrell and Lucchetti, 2007] Cottrell, A. and Lucchetti, R. (2007). *Gretl Users Guide*.
- [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.
- [Davidson and MacKinnon, 1993] Davidson, R. and MacKinnon, J. (1993). Estimation and inference in econometrics. *Oxford University Press Catalogue*.
- [De Veaux, 1989] De Veaux, R. D. (1989). Mixtures of linear regressions. *Computational Statistics & Data Analysis*, 8(3):227–245.
- [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.
- [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.
- [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.
- [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.
- [Geladi and Kowalski, 1986] Geladi, P. and Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17.
- [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.
- [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.
- [Grinstead and Snell, 1997] Grinstead, C. M. and Snell, J. (1997). *Introduction to probability*. American Mathematical Society.
- [Haitovsky, 1968] Haitovsky, Y. (1968). Missing data in regression analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 67–82.
- [Härdle, 1990] Härdle, W. (1990). *Applied nonparametric regression*, volume 27. Cambridge Univ Press.
- [Hastie and Efron, 2013] Hastie, T. and Efron, B. (2013). *lars: Least Angle Regression, Lasso and Forward Stagewise*. R package version 1.2.
- [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.
- [Henningsen and Hamann, 2007] Henningsen, A. and Hamann, J. D. (2007). systemfit: A package for estimating systems of simultaneous equations in r. *Journal of Statistical Software*, 23(4):1–40.
- [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.
- [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.
- [Kraemer et al., 2009] Kraemer, N., Schaefer, J., and Boulesteix, A.-L. (2009). Regularized estimation of large-scale gene regulatory networks using gaussian graphical models. *BMC Bioinformatics*, 10(384).
- [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.
- [Maugis-Rabasseau et al., 2012] Maugis-Rabasseau, C., Martin-Magniette, M.-L., and Pelletier, S. (2012). Selvarclustmv: Variable selection approach in model-based clustering allowing for missing values. *Journal de la Société Française de Statistique*, 153(2):21–36.
- [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.
- [Mevik et al., 2013] Mevik, B.-H., Wehrens, R., and Liland, K. H. (2013). *pls: Partial Least Squares and Principal Component regression*. R package version 2.4-3.
- [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.
- [R Core Team, 2014] R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- [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.
- [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: présentation générale. *La revue de Modulad*, (22):21–39.
- [Schwarz, 1978] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.
- [Schwarz et al., 1978] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

- [Seber and Lee, 2012] Seber, G. and Lee, A. J. (2012). *Linear regression analysis*, volume 936. John Wiley & Sons.
- [Therneau et al., 2014] Therneau, T., Atkinson, B., and Ripley, B. (2014). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-8.
- [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.
- [Weisberg, 2002] Weisberg, S. (2002). Dimension reduction regression in r. *Journal of Statistical Software*, 7(1):1–22.
- [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.
- [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.
- [Yengo and Canouil, 2014] Yengo, L. and Canouil, M. (2014). *clere: CLERE methodology for simultaneous variables clustering and regression*. R package version 1.1.
- [Yengo et al., 2012] Yengo, L., Jacques, J., Biernacki, C., et al. (2012). Variable clustering in high dimensional linear regression models.
- [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.
- [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.
- [Zou and Hastie, 2012] Zou, H. and Hastie, T. (2012). *elasticnet: Elastic-Net for Sparse Estimation and Sparse PCA*. R package version 1.1.

Part III

Appendices

Chapter 12

Identifiability

12.1 Definition

We call identifiability:

$$\nexists (\mathbf{S}, \tilde{\mathbf{S}}) \in \mathcal{S}_d \times \mathcal{S}_d \text{ with } \mathbf{S} \neq \tilde{\mathbf{S}} \text{ and } \mathbb{P}(\mathbf{X}; \mathbf{S}) = \mathbb{P}(\mathbf{X}; \tilde{\mathbf{S}}) \quad (12.1)$$

To avoid label-switching consideration, we suppose here (without loss of generality) that \mathbf{J}_r is ordered by ascending order of the labels of the covariates. Hence identifiability is paired with the hypotheses we made on \mathcal{S}_d . It is not sufficient to find a structure of linear sub-regression, the structure also has to verify hypotheses 1 to 3 (uncrossing rule + dependencies exhaustively described by the structure and then independence between the conditional response covariates). As a consequence, the covariance between two covariates is not null if and only if these covariates are linked by some sub-regressions.

12.2 Sufficient condition for identifiability

Identifiability criterion: The model \mathbf{S} is identifiable if

$$\forall j \in \{1, \dots, d_r\}, d_p^j > 1. \quad (12.2)$$

That is to have at least two regressors in each sub-regression.

To prove the sufficiency of this condition for identifiability we rely on the following lemma.

Lemma: With \mathbf{X} and \mathbf{S} following hypotheses 1 to 3, covariance between two distinct covariates does differ from 0 in only two cases:

1. One of the two variables is a regressor of the other in a sub-regression

$$j \in \{1, \dots, d_r\}, i \in J_p^j \text{ then } \text{cov}(\mathbf{X}^i, \mathbf{X}^{J_r^j}) \neq 0 \quad (12.3)$$

2. Both variables are regressed by a common covariate in their respective sub-regression:

$$\begin{aligned} \exists k \in J_f, \exists (i, j) \in \{1, \dots, d_r\} \times \{1, \dots, d_r\} \text{ with } i \neq j, \quad k \in J_p^i \text{ and } k \in J_p^j \\ \text{then } \text{cov}(\mathbf{X}^{J_r^i}, \mathbf{X}^{J_r^j}) \neq 0 \end{aligned} \quad (12.4)$$

proof of the lemma: The two cases lead immediately to non-zero covariance so we just look at other combinations of covariates.

- if $\exists(i, j) \in \{1, \dots, d_r\} \times \{1, \dots, d_r\}$, $\text{cov}(\mathbf{X}^{J_r^i}, \mathbf{X}^{J_r^j}) \neq 0$ then hypothesis 2 (uncrossing rule) guarantee that the two covariates are not in a same sub-regression so the covariance must come from the noises of the sub-regression but hypothesis 3 say that they are independent. The only remaining case is then the second case of the lemma: common covariate in the sub-regressions.
- if $(i, j) \in J_f \times J_f$ then $\text{cov}(\mathbf{X}^i, \mathbf{X}^j) = 0$ because covariates in \mathbf{X}_f are orthogonal (by hypotheses 1 and 2).
- if $j \in \{1, \dots, d_r\}, i \in J_f$ and $\text{cov}(\mathbf{X}^{J_r^j}, \mathbf{X}^i) \neq 0$ then $i \in J_p^j$ by hypotheses 1 and 3 and equation 4.3.

□

proof of the identifiability criterion: We suppose that 12.2 is verified and the model is not identifiable:

$$\exists(\mathbf{S}, \tilde{\mathbf{S}}) \in \mathcal{S}_d \times \mathcal{S}_d \text{ with } \mathbf{S} \neq \tilde{\mathbf{S}} \text{ and } \mathbb{P}(\mathbf{X}; \mathbf{S}) = \mathbb{P}(\mathbf{X}; \tilde{\mathbf{S}}) \quad (12.5)$$

$\tilde{\mathbf{S}} = (\tilde{\mathbf{J}}_r, \tilde{\mathbf{J}}_p)$ contains \tilde{d}_r sub-regressions and is characterized by $\tilde{\mathbf{J}}_r = (\tilde{J}_r^1, \dots, \tilde{J}_r^{\tilde{d}_r}), \tilde{\mathbf{J}}_p = (\tilde{J}_p^1, \dots, \tilde{J}_p^{\tilde{d}_r})$.

Because $\mathbf{S} \neq \tilde{\mathbf{S}}$ we have $\mathbf{J}_r \neq \tilde{\mathbf{J}}_r$ or $\mathbf{J}_p \neq \tilde{\mathbf{J}}_p$.

- If $\mathbf{J}_r = \tilde{\mathbf{J}}_r$ and $\mathbf{S} \neq \tilde{\mathbf{S}}$ then one sub-regression as a predictor that stands only in one of the two structures. We suppose (without loss of generality) that $\exists j \in \{1, \dots, d_r\}$ for which $\exists i \in J_p^j$ with $i \notin \tilde{J}_p^j$ so $\text{cov}_{\mathbf{S}}(\mathbf{X}^{J_r^j}, \mathbf{X}^i) \neq 0$ and $\text{cov}_{\tilde{\mathbf{S}}}(\mathbf{X}^{J_r^j}, \mathbf{X}^i) = 0$ (from the lemma) so the two structure don't give the same joint distribution, leading to a contradiction.
- If $\mathbf{J}_r \neq \tilde{\mathbf{J}}_r$ then one of the two models has a sub-regression that is not in the other. We suppose (without loss of generality) that $\exists J_r^j \in J_r$ with $J_r^j \notin \tilde{\mathbf{J}}_r$ then $J_r^j \in J_f$ (recall $J_f = \{1, \dots, d\} \setminus J_r$). We note that $J_r^j \in J_r$ means $\exists k_1 \neq k_2, \{k_1, k_2\} \subset J_r^j \subset J_f$. Then $\text{cov}_{\mathbf{S}}(\mathbf{X}^{J_r^j}, \mathbf{X}^{k_1}) \neq 0$ and $\text{cov}_{\mathbf{S}}(\mathbf{X}^{J_r^j}, \mathbf{X}^{k_2}) \neq 0$ so by the lemma k_1 and k_2 are responses variables in $\tilde{\mathbf{S}}$: $\exists(l_1, l_2) \in \{1, \dots, \tilde{d}_r\} \times \{1, \dots, \tilde{d}_r\}, \tilde{J}_r^{l_1} = k_1, \tilde{J}_r^{l_2} = k_2$ and J_r^j is a regressor of k_1 and k_2 : $J_r^j \in J_p^{l_1}, J_r^j \in J_p^{l_2}$ thus $\text{cov}_{\tilde{\mathbf{S}}}(\mathbf{X}^{k_1}, \mathbf{X}^{k_2}) \neq 0$ that is not possible because $\{k_1, k_2\} \subset J_r^j \subset J_f$ and covariates in \mathbf{X}^{J_f} are orthogonal by hypotheses.

Finally, condition 12.2 is sufficient for identifiability of \mathbf{S} . □

Remark: Because sub-regressions with at least two regressors are identifiable, the only non-identifiable sub-regressions could be those with only one regressor, leading only to pairwise correlations that can be seen directly in the correlation matrix. Such sub-regression can be permuted without any impact on interpretation so such trivial sub-regression are not a problem even if they may occur with real datasets. One more thing: exact sub-regression with at least two sub-regressors are identifiable with our hypotheses.

Chapter 13

CorReg: the concept

The **CorReg** package is already downloadable on 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 by the end of the year 2014, probably using Basic Excel R Toolkit(BERT¹).

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 at Arcelor-Mittal Dunkerque to provide ergonomic and powerful statistical tools to non-statisticians.

¹<https://github.com/StructuredDataLLC/Basic-Excel-R-Toolkit>