Model-based covariable decorrelation in linear regression. Application to missing data and to steel industry.

Clément THÉRY

January 15, 2015

To my sons

Résumé

Les travaux effectués durant cette thèse ont pour but de 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 comme une présélection des variables explicatives à des fins illustratives mais ne dépend pas de la variable réponse. Il 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 réinjecter l'information résiduelle contenue dans les variables redondantes de manière séquentielle. On utilise ainsi 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. Cela permet la recherche de structure malgré l'absence de certaines données. Mais un autre débouché est l'imputation multiple des données manquantes, préalable à l'utilisation de méthodes classiques incompatibles avec la présence de valeurs manquantes. De plus, l'imputation multiple des valeurs manquantes permet d'obtenir un estimateur de la variance des valeurs imputées. 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 CRAN[1] sous licence CeCILL[2], implémente les méthodes développées durant cette thèse.

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

[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 pre-treatment 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). Then we are 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 pre-treatment 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 CRAN[3] now under CeCILL[4] license. It implements the methods created during this thesis.

Keywords: Pre-treatment, 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

Acknowledgements

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 MΘdal team, especially those from the "bureau 106 " who helped me to put `CorReg` on CRAN (in particular Quentin GRIMONPREZ) and those who had already submitted something on CRAN know that it is not always fun to achieve this goal. They also 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

Chapter 1

Introduction

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

1.1 industrial motivation

1.1.1 Steel making process

Steel making 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 cock 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 mould (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 millimetre 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 steel making process.

If the final product requires a thinner strip of steel, coils pass through a cold rolling mill. Figure 1.1 provides some illustrations of the steel making process. Each step of this process involves a whole manufacture 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.

Steel making 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 needs adapted statistical tools. That is why this thesis has been made.

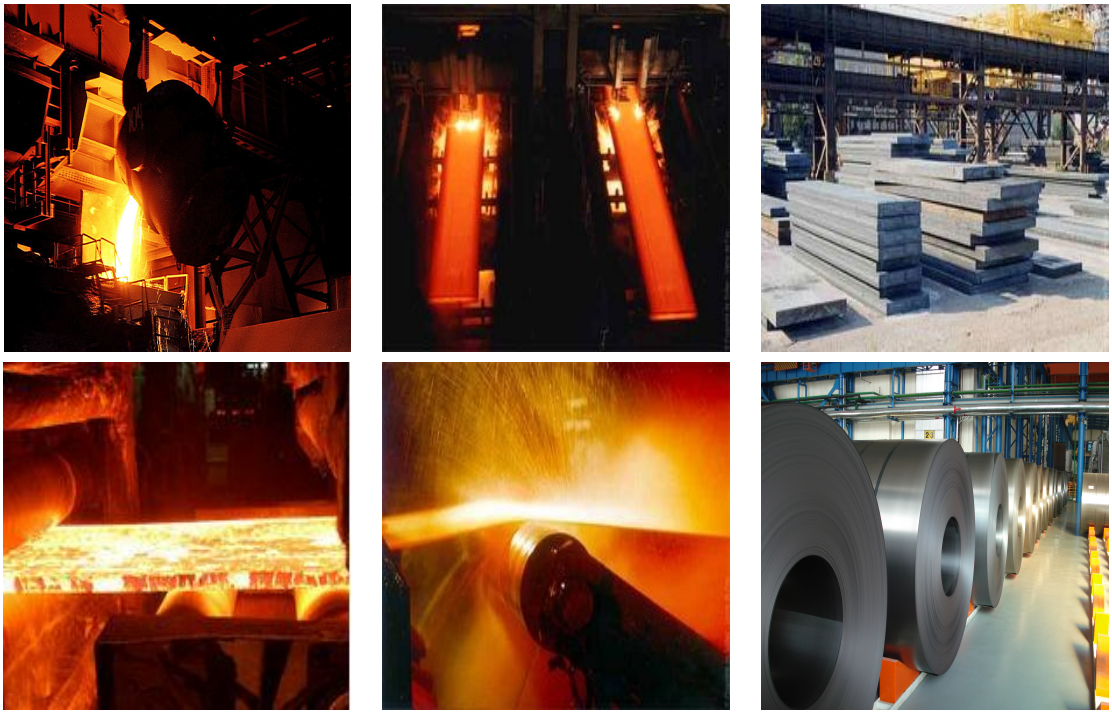


Figure 1.1: Quick overview of the steel making process, from hot liquid metal to coils.

1.1.2 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 includes several sub-process. Thus we have many covariates and no a priori on the relevant ones. Moreover, the values of each covariate 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.

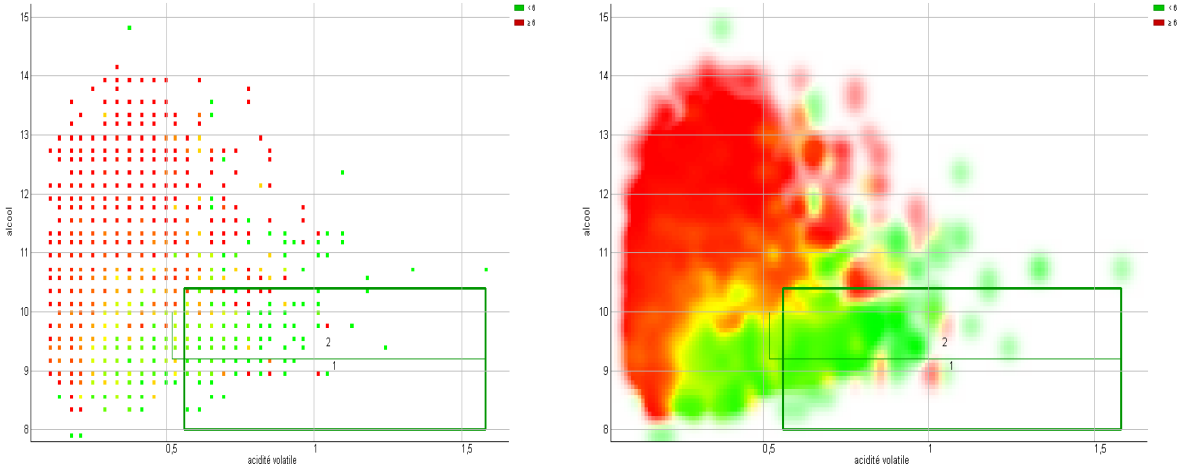
We will see in the state of the art that correlations are a real issue and that the number of variables increases the problem. The stakes are very high because of the high productivity of the steel plants and the extreme competition between steel makers but also because steel making is now well-known 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.

1.1.3 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 1.2 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 in Section 3.1.3). 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. Morever, it requires to discretize the response variable to obtain "good" and "bad" values. The green rectangle is very far from the true green zone even for this toy example provided by the reseller.

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



(a) Without blur

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

Figure 1.2: 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.

Conclusion: 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 used methods 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.
Then we need a tool that is both easy to use and to understand, giving priority to interpretation more than prediction. The tool will have to work without a priori and to be able to select relevant covariates.

1.2 Mathematical motivation

Every engineer, even non-statistician uses frequently linear regression to seek relationship between some covariates. It is easy to understand, fast to do, and is used in nearly all the

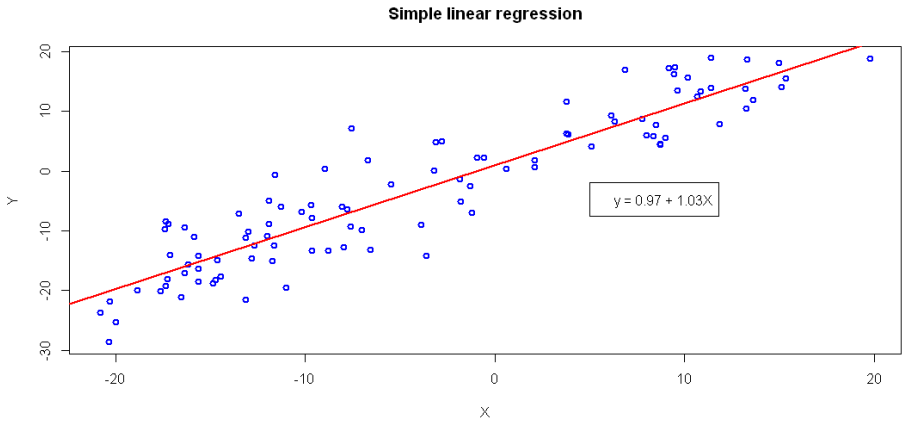


Figure 1.3: An example of simple linear regression

fields where statistics are used [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. Linear regression is an old strategy well known and with many derivatives (as we will see in the following) and can be generalized (see [Kiebel and Holmes, 2003, Wickens, 2004, Nelder and Baker, 1972] and also [McCullagh and Nelder, 1989]). Its 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 not linear in $\boldsymbol{X}$ but can be seen as a basis expansion using new covariates (the interactions) composed of the product of some of the original covariates.

But linear regression is in trouble when correlations between the covariates are strong, as we will see later (Chapter 3). So we have decided to investigate some ways to overcome correlations issues in linear regression, keeping in mind industrial constraints about variable selection and interpretability.

1.3 Outline of the manuscript

After a substantial abstract in french, we start with a brief state of the art

Chapter 2

Résumé substantiel en français

2.1 Position du problème

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

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

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

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

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

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

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

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

2.2 Modélisation explicite des corrélations

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

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

Hypothèse 1 *Les corrélations entre covariables viennent uniquement 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* $j = 1, \ldots, d_r$ *ayant la variable* $\boldsymbol{X}^{J_r^j}$ *comme variable réponse (*$J_r^j \in \{1, \ldots, d\}$ *et* $J_r^j \neq J_r^{j'}$ *pour* $j \neq j'$*) et ayant les* $d_p^j > 0$ *variables* $\boldsymbol{X}^{J_p^j}$ *comme variables predictives (*$J_p^j \subset \{1, \ldots, d\} \backslash J_r^j$ *et* $d_p^j = |J_p^j|$ *le cardinal de* J_p^j*) :*

$$\boldsymbol{X}^{J_r^j} = \boldsymbol{X}^{J_p^j} \boldsymbol{\alpha}_j + \boldsymbol{\varepsilon}_j, \tag{2.1}$$

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

Hypothèse 2 *Nous supposons également que les variables réponses et les variables prédictives forment deux blocs disjoints dans* $\boldsymbol{X}$ *: pour toute sous-régression* $j = 1, \ldots, d_r$, $J_p^j \subset J_f$ *où* $J_r = \{J_r^1, \ldots, J_r^{d_r}\}$ *est l'ensemble de toutes les variables réponses avec* $J_f = \{1, \ldots, d\} \backslash J_r$ *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 imbrications 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 $\boldsymbol{J}_r = (J_r^1, \ldots, J_r^{d_r})$ le d_r-uplet des variables réponses (à ne pas confondre avec J_r défini plus haut), $\boldsymbol{J}_p = (J_p^1, \ldots, J_p^{d_r})$ le d_r-uplet des prédicteurs de toutes les sous-régressions, $\boldsymbol{d}_p = (d_p^1, \ldots, d_p^{d_r})$ les nombres correspondants de predicteurs et $\boldsymbol{S} = (\boldsymbol{J}_r, \boldsymbol{J}_p)$ le *model* global (structure de sous-régressions). Pour alléger les notations, on définit alors $\boldsymbol{X}_r = \boldsymbol{X}^{J_r}$ la matrice des variables réponses et $\boldsymbol{X}_f = \boldsymbol{X}^{J_f}$ la matrice de *toutes* les autres variables, ainsi considérées comme libres (*free* en anglais). Les valeurs des paramètres sont également concaténées : $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{d_r})$ est le d_r-uplet des vecteurs des coefficients de sous-régression et $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_{d_r}^2)$ le vecteur des variances associées :

$$\boldsymbol{X}_r = \boldsymbol{X}_f \boldsymbol{\alpha}^* + \boldsymbol{\varepsilon} \quad \text{(regression multiple multivariée)}$$

où la matrice $\boldsymbol{\varepsilon} \in \mathbb{R}^{n \times d_r}$ est la matrice des bruits des sous-régressions composée des colonnes $\boldsymbol{\varepsilon}_j \sim \mathcal{N}(0, \sigma_j^2 \boldsymbol{I}_n)$ que nous supposons indépendantes entre elles et $\boldsymbol{\alpha}^* \in \mathbb{R}^{d_f \times d_r}$ est la matrice des coefficients des sous-régressions avec $(\boldsymbol{\alpha}_j^*)_{J_p^j} = \boldsymbol{\alpha}_j$ et $(\boldsymbol{\alpha}_j^*)_{J_f \backslash J_p^j} = \boldsymbol{0}$. Ces notations sont illustrées dans l'exemple ci-après.

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

$$R^2 = 1 - \frac{\mathrm{Var}(\boldsymbol{\varepsilon}_1)}{\mathrm{Var}(\boldsymbol{X}^3)} \tag{2.2}$$

La figure 3.5 page 24 illustre la détérioration de l'estimation quand les sous-régressions deviennent trop fortes (R^2 proche de 1) pour différentes valeurs de n sur nos données d'exemple.

Remarques

- Les sous-régressions définies en (2.1) sont très simples à comprendre pour l'utilisateur et permettent donc d'avoir un aperçu net des corrélations présentes dans les données étudiées.
- $\boldsymbol{S}$ ne dépend pas de $\boldsymbol{Y}$ et peut donc être estimé séparément.

2.3 Modèle marginal

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

Hypothèse 3 *On suppose l'indépendance 2 à 2 entre les erreurs de régression $\boldsymbol{\varepsilon}_Y$ et les $\boldsymbol{\varepsilon}_j$, avec $j \in \{1, \dots, d_r\}$. En particulier on a l'indépendance conditionnelle entre les variables réponses : $\{\boldsymbol{X}^{J_r^j} | \boldsymbol{X}^{J_p^j}, \boldsymbol{S}; \boldsymbol{\alpha}_j, \sigma_j^2\}$ définies par l'équation (2.1).*

On a donc

$$\begin{aligned}
\mathbb{P}(\boldsymbol{X}_r | \boldsymbol{X}_f, \boldsymbol{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2) &= \prod_{j=1}^{d_r} \mathbb{P}(\boldsymbol{X}^{J_r^j} | \boldsymbol{X}^{J_p^j}, \boldsymbol{S}; \boldsymbol{\alpha}_j, \sigma_j^2) \\
\mathbb{P}(\boldsymbol{Y} | \boldsymbol{X}_f, \boldsymbol{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2) &= \int_{\mathbb{R}^{d_r}} \mathbb{P}(\boldsymbol{Y} | \boldsymbol{X}_r, \boldsymbol{X}_f, \boldsymbol{S}; \boldsymbol{\beta}, \sigma_Y^2) \mathbb{P}(\boldsymbol{X}_r | \boldsymbol{X}_f, \boldsymbol{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2) d\boldsymbol{X}_r,
\end{aligned}$$

ce qui donne

$$\begin{aligned}
\boldsymbol{Y} &= \boldsymbol{X}_f \boldsymbol{\beta}_f + \boldsymbol{X}_r \boldsymbol{\beta}_r + \boldsymbol{\varepsilon}_Y \\
\boldsymbol{Y} &= \boldsymbol{X}_f (\boldsymbol{\beta}_f + \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\alpha}_j^*) + \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_Y \qquad (2.3) \\
&= \boldsymbol{X}_f \boldsymbol{\beta}_f^* + \boldsymbol{\varepsilon}_Y^*
\end{aligned}$$

où $\boldsymbol{\beta}_r = \boldsymbol{\beta}_{J_r}, \boldsymbol{\beta}_f = \boldsymbol{\beta}_{J_f}$.

On se retrouve donc avec un modèle marginal plus parsimonieux, sans biais sur $\boldsymbol{Y}$ (vrai modèle) mais avec une variance potentiellement 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 $(\boldsymbol{X}'\boldsymbol{X})$ par suppression des variables redondantes et aussi la réduction de la dimension (le modèle marginal ne nécessite que l'inversion de $(\boldsymbol{X}'_f \boldsymbol{X}_f)$) face au léger accroissement de la variance issu de la marginalisation. On va donc comparer

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

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

$$\begin{aligned}
\text{MSE}(\hat{\boldsymbol{\beta}} | \boldsymbol{X}) &= \sigma_Y^2 \operatorname{Tr}(\boldsymbol{X}'\boldsymbol{X})^{-1} \\
\text{MSE}(\hat{\boldsymbol{\beta}}^* | \boldsymbol{X}) &= \| \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\alpha}_j^* \|_2^2 + \| \boldsymbol{\beta}_r \|_2^2 + (\sigma_Y^2 + \sum_{j=1}^{d_r} \sigma_j^2 \beta_{J_r^j}^2) \operatorname{Tr}(\boldsymbol{X}'_f \boldsymbol{X}_f)^{-1}
\end{aligned}$$

Ces deux équations illustrent bien le compromis biais-variance du modèle marginal. La figure 4.2 page 47 compare les erreurs d'estimation obtenues pour différentes valeurs des paramètres, montrant la nette amélioration rendue possible par la marginalisation. Quand n tend vers l'infini les traces tendent vers 0 et donc le modèle complet devient meilleur (car sans biais). Mais pour n fixé, quand les σ_j^2 tendent vers 0 la trace dans le MSE du modèle complet explose (les sous-régressions tendent à devenir exactes) et la variance du modèle marginal diminue donc l'explosion du MSE du modèle complet finit par dépasser le biais du modèle marginal et le MSE du modèle marginal devient le plus faible. On remarque enfin que quand $\boldsymbol{\beta}_r$ est nul, le modèle marginal est le vrai modèle et devient donc meilleur que le modèle complet (qui estime en vain $\boldsymbol{\beta}_r$, même s'il est sans biais).

Remarque

- Les hypothèses 1 à 3, impliquent que les variables dans $\boldsymbol{X}_f$ *ne sont pas corrélées* (covariance nulle : voir Lemme en section A.2).

2.4 Notion de prétraitement

Le modèle marginal peut être vu comme un pari: on décide (pour s'affranchir des problèmes liés aux corrélations) de ne retenir que $\boldsymbol{X}_f$ pour la prédiction. Il s'agit d'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 indépendantes de $\boldsymbol{Y}$. Le modèle final est donc parsimonieux mais ne fausse pas l'interprétation. L'estimation de $\boldsymbol{\beta}^*$ peut ensuite se faire en utilisant une quelconque méthode de sélection de variables pour éliminer les variables qui, elles, ne sont pas pertinentes pour expliquer $\boldsymbol{Y}$.

On obtient donc deux types de 0 : ceux de la marginalisation qui pointent les variables redondantes et ceux de sélection qui viennent dans un second temps et 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.

2.5 Estimation de la structure

La raison d'être de notre modèle marginal est la fragilité des méthodes de régression face à des covariables fortement corrélées. Il serait donc vain d'essayer les sous-régression en estimant les modèles de régression de chaque variable en fonction de toutes les autres car les corrélations nuisent à l'efficacité de ces modèles. 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 $\boldsymbol{G}$ de taille $d \times d$ correspond à une matrice d'adjacence qui indique les liaisons entre covariables de la manière suivante : $\boldsymbol{G}_{i,j} = 1$ si, et seulement si $\boldsymbol{X}^j$ est expliqué par $\boldsymbol{X}^i$.

Chaque étape $(q+1)$ de l'algorithme propose de garder la structure $\boldsymbol{G}^{(q)}$ en cours ou bien de bouger vers une structure candidate qui diffère de $\boldsymbol{G}^{(q)}$ en un unique point. Ainsi,

selon les cas, les candidats vont allonger ou réduire des sous-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 $\boldsymbol{X}$ qui complète le modèle de sous-régressions en établissant des modèles de mélanges gaussiens indépendants pour les variables de $\boldsymbol{X}_f$. 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 meilleur BIC.

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 [Grinstead and Snell, 1997]. Ainsi il est certain que, asymptotiquement (en nombre d'étapes), l'algorithme trouve le modèle ayant le meilleur BIC.

2.6 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 $\boldsymbol{G}$:

1. Modification (suppression/création) de l'arc (i, j) :

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

2. Si la variable $\boldsymbol{X}^i$ devient un prédicteur elle ne peut plus être une variable réponse :

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

3. Si la variable $\boldsymbol{X}^j$ devient une variable réponse elle ne peut plus être un prédicteur :

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

où $\boldsymbol{G}_{i,j}^{(q+1)}$ est la valeur de la matrice $\boldsymbol{G}^{(q+1)}$ ligne i et colonne j, $\boldsymbol{G}_{.,i}^{(q+1)}$ est la $i^{\text{ième}}$ colonne de $\boldsymbol{G}^{(q+1)}$ et $\boldsymbol{G}_{j,.}^{(q+1)}$ la $j^{\text{ième}}$ ligne de $\boldsymbol{G}^{(q+1)}$. Cette méthode de relaxation permet de sortir rapidement des extrema locaux et améliore donc significativement l'efficacité de l'algorithme (Figure 5.10 page 73). La méthode est illustrée sur un exemple par les figures 5.1 à 5.6 (pages 67 à 68).

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 $\boldsymbol{S}$ dans une base de données $\boldsymbol{X}$, BIC

est utilisé comme approximation pour $\mathbb{P}(\boldsymbol{S}|\boldsymbol{X}) \propto \mathbb{P}(\boldsymbol{X}|\boldsymbol{S})\mathbb{P}(\boldsymbol{S})$ car $\mathbb{P}(\boldsymbol{S})$ est considéré comme suivant une loi uniforme. Ici on s'appuie sur une loi uniforme hiérarchique $\mathbb{P}_H(\boldsymbol{S}) = \mathbb{P}_U(\boldsymbol{J}_p|\boldsymbol{d}_p, \boldsymbol{J}_r, d_r)\mathbb{P}_U(\boldsymbol{d}_p|\boldsymbol{J}_r, d_r)\mathbb{P}_U(\boldsymbol{J}_r|d_r)\mathbb{P}_U(d_r)$ 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. Ce critère pénalise plus lourdement les modèles ayant de nombreux paramètres mais conserve les propriétés asymptotiques de BIC.

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.

2.7 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 4.7), l'efficacité de l'algorithme de recherche de structure (section 6.2), et l'efficacité du modèle marginal s'appuyant sur la structure estimée (section 6.3 et chapitre 7). Le bilan est très positif comme le montrent les graphiques de ces différentes sections.

2.8 Modèle plug-in sur les résidus du modèle marginal

Une méthode séquentielle par plug-in a été développée pour tenter d'améliorer le modèle marginal. Il s'agit de mettre à profit la formulation exacte du modèle marginal pour estimer $\boldsymbol{\beta}_r$ et ainsi réduire le bruit de la régression marginale puis d'utiliser ce nouvel estimateur pour identifier $\boldsymbol{\beta}_f$ et ainsi obtenir un nouveau modèle complet s'appuyant sur $\boldsymbol{X}$ entier mais protégé des corrélations par l'estimation séquentielle.

Après estimation du modèle marginal on s'applique à essayer d'améliorer l'estimation de $\boldsymbol{\beta}$:

- Estimation de $\boldsymbol{\varepsilon}_Y^*$ à partir de $\hat{\boldsymbol{\beta}}_f^*$:

$$\hat{\boldsymbol{\varepsilon}}_Y^* = \boldsymbol{Y} - \boldsymbol{X}_f\hat{\boldsymbol{\beta}}_f^*.$$

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

$$\hat{\boldsymbol{\varepsilon}} = \boldsymbol{X}_r - \boldsymbol{X}_f\hat{\boldsymbol{\alpha}}^*. \tag{2.4}$$

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

- Estimation de $\boldsymbol{\beta}_r$ basée sur la définition de $\boldsymbol{\varepsilon}_Y^* = \boldsymbol{\varepsilon}\boldsymbol{\beta}_r + \boldsymbol{\varepsilon}_Y$ (équation (2.3)) :

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

Cet estimateur nous permet de réduire le bruit du modèle marginal, afin d'essayer d'estimer $\boldsymbol{Y}$ plus précisément

$$\hat{\boldsymbol{Y}}_{plug-in} = \boldsymbol{X}_f\hat{\boldsymbol{\beta}}_f^* + \hat{\boldsymbol{\varepsilon}}\hat{\boldsymbol{\beta}}_r^{\varepsilon}.$$

On peut ensuite dans une phase d'identification obtenir un estimateur de $\boldsymbol{\beta}_f$.
On a $\boldsymbol{\beta}_f^* = \boldsymbol{\beta}_f + \boldsymbol{\alpha}^*\boldsymbol{\beta}_r$.

- Estimateur de $\boldsymbol{\beta}_f$ par identification :

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

La figure 8.1 (page 109) montre l'efficacité du modèle plug-in et son champ d'application recommandé : les cas avec assez de correlations pour que les méthodes classiques appliquées à $\boldsymbol{X}$ soient handicapées mais pas assez de corrélations pour que le retrait des variables redondantes (modèle marginal) se fasse sans perte siginificative d'information.

2.9 Valeurs manquantes

Un coproduit du modèle de sous-régression concerne les valeurs manquantes. Le fait de disposer d'un modèle génératif complet sur $\boldsymbol{X}$ avec modélisation explicite des dépendances permet en effet de composer avec les valeurs manquantes en utilisant les lois conditionnelles. Tout d'abord, l'estimation de $\boldsymbol{\alpha}$ peut se faire sur les données observées en intégrant sur les données manquantes. On peut alors utiliser un algorithme de type EM (Expectation Maximization) pour estimer $\hat{\boldsymbol{\alpha}}$.

En pratique, on fait 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 $\hat{\boldsymbol{\alpha}}$ 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. Autrement dit, le modèle génératif sur $\boldsymbol{X}$ donne la loi conditionnelle des valeurs manquantes sachant les valeurs observées, ce qui permet d'imputer les valeurs manquantes en connaissant la variance associée à ces imputations.

Chapter 3

State of the art in linear regression

Abstract: Brief state of the art to have a glimpse of regression methods 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 web[1] for free. The main goal of this chapter is to provide a quick overview of linear regression and the reasons why a new method was needed in our industrial context. It is not a book or a course about regression. References are provided for all presented methods.

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 $\| \boldsymbol{\beta} \|_\infty = \max(|\beta_1|, \ldots, |\beta_d|)$. Vectors, matrices and tuples are in bold characters.

3.1 Regression

3.1.1 General purpose

3.1.2 Linear models

3.1.3 Non-linear models

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. The usual example is the case of wet grass in a garden. You do not remember if the sprinkler was on or off and you do not know if it has rain.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 9 when confronted to missing values.

Figure 3.1.3 illustrates a simpler case with dependency between the sprinkler activation and rain. It also shows probability tables associated to the Bayesian network. 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

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

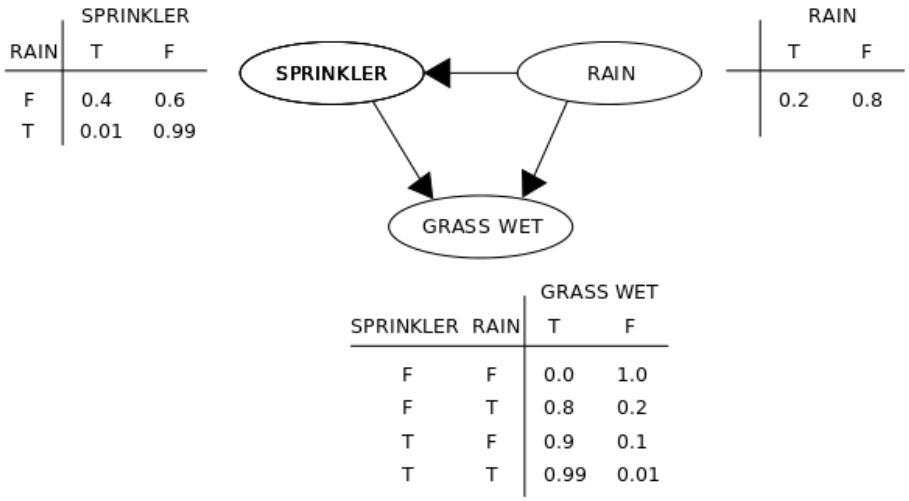


Figure 3.1: Simple Bayesian network and its probability tables. Public domain image.

(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. Thus we will keep the ease of interpretation.

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 3.3 because a same variable will be used several times and the tree will fail to give a simple interpretation as "$\boldsymbol{Y}$ and $\boldsymbol{X}^1$ are proportional". Trivial case: $\boldsymbol{Y} = \boldsymbol{X}^1 + \boldsymbol{\varepsilon}_Y$ where $\boldsymbol{\varepsilon}_Y \sim \mathcal{N}(0, \sigma_Y^2 \boldsymbol{I}_n)$ 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 $\boldsymbol{Y}$ and $\boldsymbol{X}$ 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

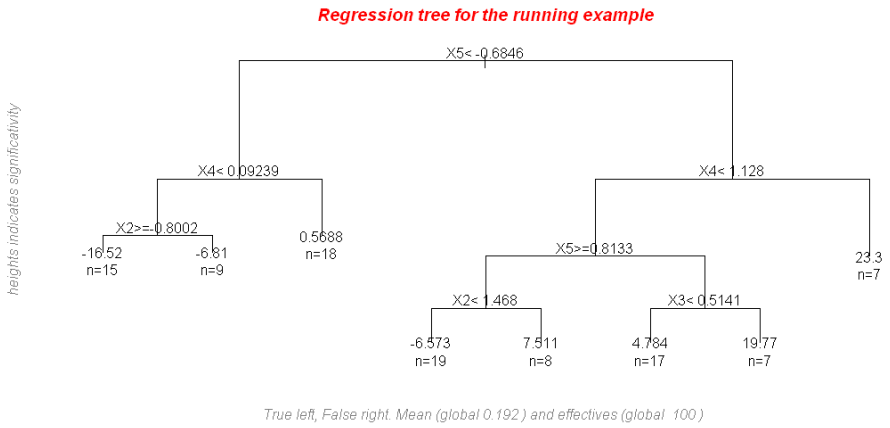
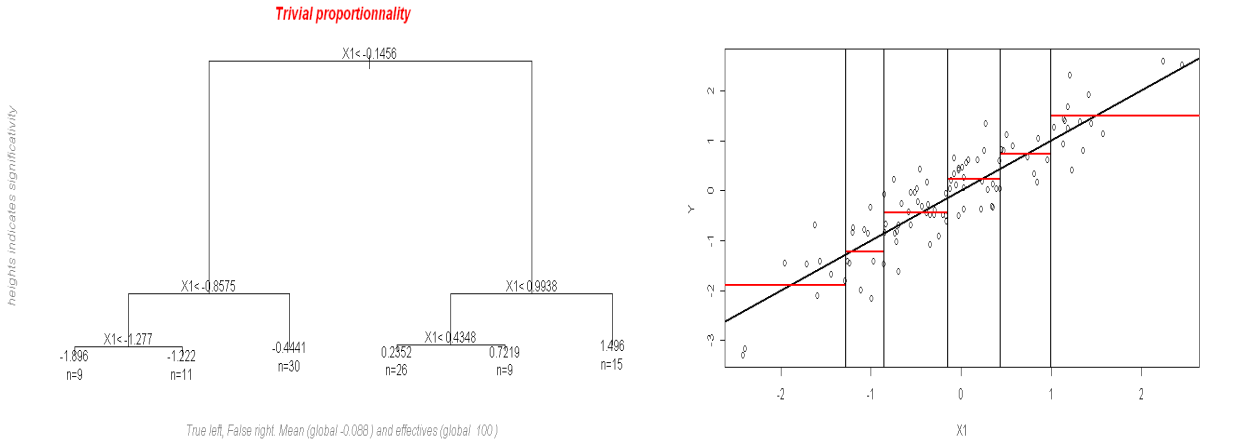


Figure 3.2: Regression tree obtain with the package `CorReg` (graphical layer on top of the `rpart` package) on the running example.



(a) Tree found for the trivial case

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

Figure 3.3: Predictive model associated to the tree (red) and true model (black)

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].

3.2 Parameter estimation

3.2.1 Maximum likelihood and related methods

We note the linear regression model:

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

where $\boldsymbol{X}$ is the $n \times d$ matrix of the explicative variables, $\boldsymbol{Y}$ the $n \times 1$ response vector and $\boldsymbol{\varepsilon}_Y \sim \mathcal{N}(\boldsymbol{0}, \sigma_Y^2 \boldsymbol{I}_n)$ the noise of the regression, with $\boldsymbol{I}_n$ the n-sized identity matrix and $\sigma_Y > 0$. The $d \times 1$ vector $\boldsymbol{\beta}$ is the vector of the coefficients of the regression. Thus we suppose that $\boldsymbol{Y}$ linearly depends on $\boldsymbol{X}$ and that the residuals are Gaussian and *i.i.d.* We also suppose that $\boldsymbol{X}$ has full column rank d. $\boldsymbol{\beta}$ can be estimated by $\hat{\boldsymbol{\beta}}$

with Ordinary Least Squares (OLS), that is the unbiased maximum likelihood estimator [Saporta, 2006, Dodge and Rousson, 2004]:

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

with variance matrix

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

In fact it is the Best Linear Unbiased Estimator (BLUE). The theoretical MSE is given by

$$\mathrm{MSE}(\hat{\boldsymbol{\beta}}_{OLS}) = \sigma_Y^2\,\mathrm{Tr}((\boldsymbol{X}'\boldsymbol{X})^{-1}).$$

Equation (3.1) has no intercept but usually a constant is included as one of the regressors. For example we can take $\boldsymbol{X}^1 = (1,\ldots,1)'$. The corresponding element of $\boldsymbol{\beta}$ is then the intercept β_1. In the following we do not consider the intercept to simplify notations. In practice, an intercept is added by default.

Estimation of $\boldsymbol{Y}$ by OLS can be viewed as a projection onto the linear space spanned by the regressors $\boldsymbol{X}$ that minimizes the distance with each individual $(\boldsymbol{X}_i, Y_i)$ as shown in figure 3.4. It can be written

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

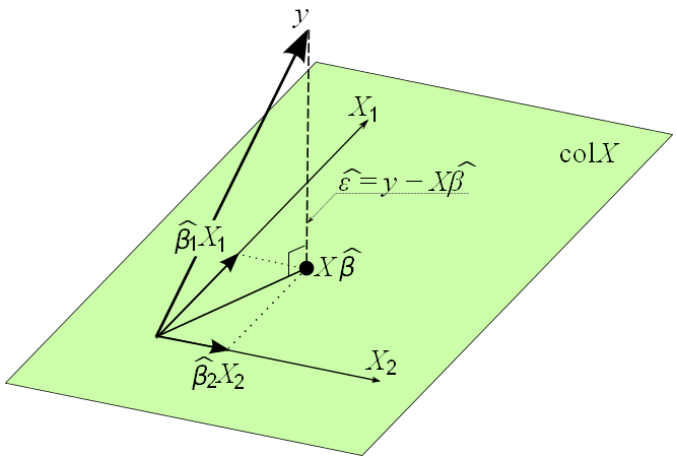


Figure 3.4: Multiple linear regression with Ordinary Least Squares seen as a projection on the $d-$dimensional hyperplane spanned by the regressors $\boldsymbol{X}$. Public domain image.

Estimation of $\boldsymbol{\beta}$ requires the inversion of $\boldsymbol{X}'\boldsymbol{X}$ which will be ill-conditioned or even singular if some covariates depend linearly from each other. For a given number n of individuals, conditioning of $\boldsymbol{X}'\boldsymbol{X}$ 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 $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ is ill-conditioned and the variance of $\hat{\boldsymbol{\beta}}_{OLS}$ increases (equation (3.3)), giving unstable and unusable estimator [Hoerl and Kennard, 1970]. Another problem is that matrix inversion requires

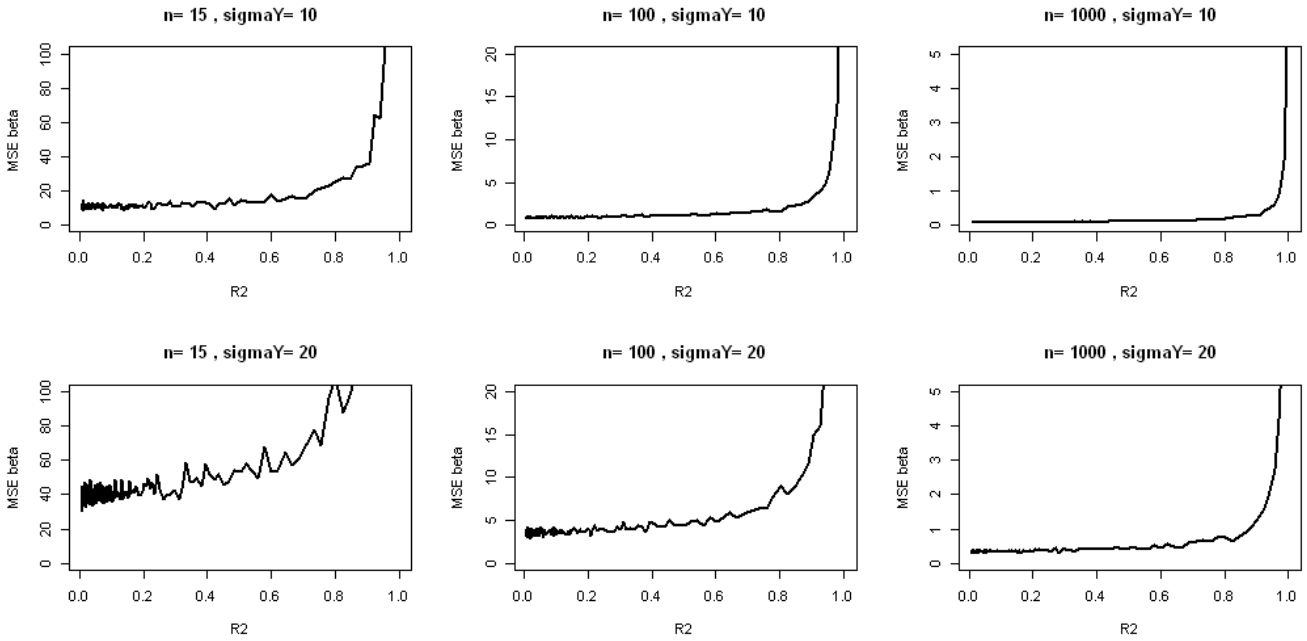


Figure 3.5: Evolution of observed Mean Squared error on $\hat{\boldsymbol{\beta}}_{OLS}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates (running example).

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 covariates. Pointing out some relevant covariates and how they really impact the response is the main goal here. We will need a variable selection method one moment or another. It could be as a pre-treatment (to run a first tool to select relevant covariates and then estimate the values of the non-zero coefficients) , during coefficient estimation (some estimators can lead to exact zeros in $\hat{\boldsymbol{\beta}}$) or by post-treatment (by thresholding with tests of hypothesis, *etc.*).

Running example: We look at a simple case with $d = 5$ variables defined by four independent scaled Gaussian $\mathcal{N}(0, 1)$ named $\boldsymbol{X}^1, \boldsymbol{X}^2, \boldsymbol{X}^4, \boldsymbol{X}^5$ and $\boldsymbol{X}^3 = \boldsymbol{X}^1 + \boldsymbol{X}^2 + \boldsymbol{\varepsilon}_1$ where $\boldsymbol{\varepsilon}_1 \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \boldsymbol{I}_n)$. We also define two *scenarii* for $\boldsymbol{Y}$ with $\boldsymbol{\beta} = (1, 1, 1, 1, 1)'$ and $\sigma_Y \in \{10, 20\}$. So there is no intercept (can be seen as a null intercept). It is clear that $\boldsymbol{X}'\boldsymbol{X}$ will become more ill-conditioned as σ_1 gets smaller. In the following, the R^2 stands for the coefficient of determination which is here:

$$R^2 = 1 - \frac{\mathrm{Var}(\boldsymbol{\varepsilon}_1)}{\mathrm{Var}(\boldsymbol{X}^3)} \tag{3.4}$$

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

Results shown in Figure 3.5 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 $\hat{\boldsymbol{\beta}}$ is proportional to σ_Y^2 so the MSE are bigger when the main regression is weak (as in the real life).

3.2.2 Ridge regression: a penalized estimator

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

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

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\beta} \left\{ \| \boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} \|_2^2 \right\} \text{ subject to } \| \boldsymbol{\beta} \|_2^2 \leq \eta \text{ with } \eta > 0 \tag{3.5}$$

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

The solution of the ridge regression is given by

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}'\boldsymbol{X} - \lambda \boldsymbol{I}_n\right)^{-1} \boldsymbol{X}'\boldsymbol{Y} \tag{3.6}$$

and we see in this equation that a global modification of $\boldsymbol{X}'\boldsymbol{X}$ (on its diagonal) is done for a given λ to improve its conditioning. 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 do not 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.

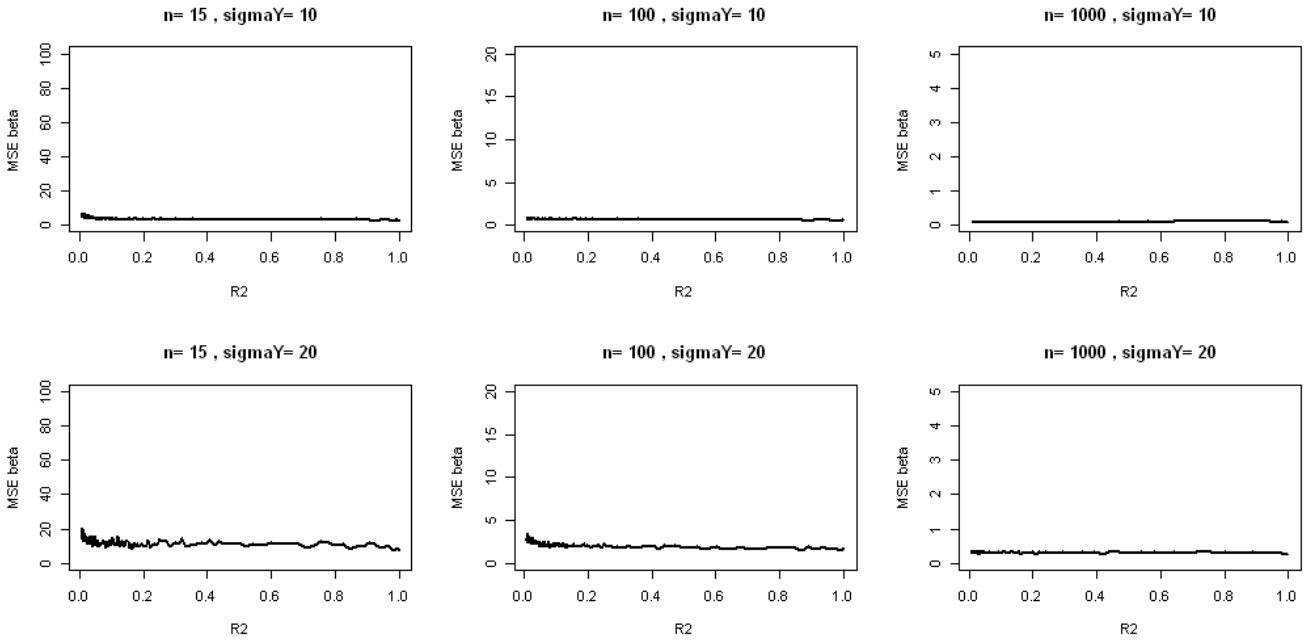


Figure 3.6: Evolution of observed Mean Squared error on $\hat{\boldsymbol{\beta}}_{ridge}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

3.3 Variable selection methods

3.3.1 Least Absolute Shrinkage and Selection Operator (LASSO)

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

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

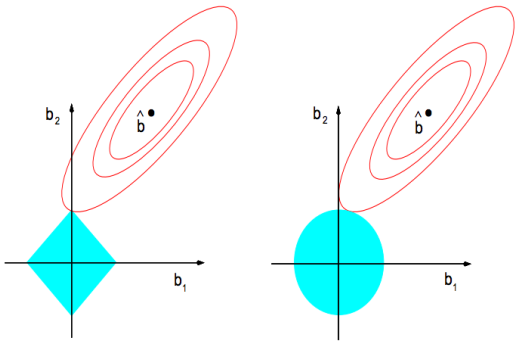


Figure 3.7: Geometric view of the Penalty for the LASSO (left) compared to ridge regression (right) as shown in the book from Hastie [Hastie et al., 2009]

Figure 3.7 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 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 8.3 and compared to our models.

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 that can be found 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 $\hat{\boldsymbol{\beta}}$ can be improved by a second estimation with OLS based only on selected covariates [Zhang and Shen, 2010].

3.3.2 Least Angle Regression (LAR)

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 to 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 3.9). We then have a list of models with several numbers of non-zero coefficients and can choose between them with cross-validation for example.

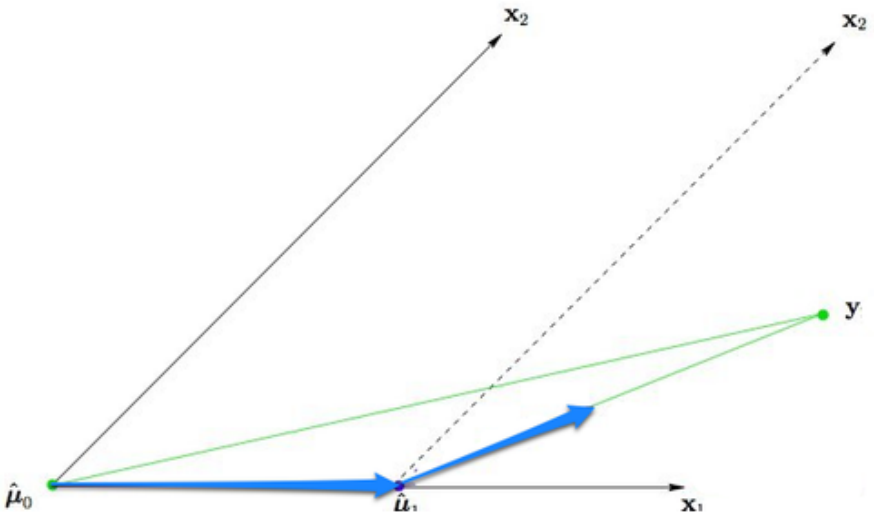


Figure 3.8: The geometry of Least Angle Regression

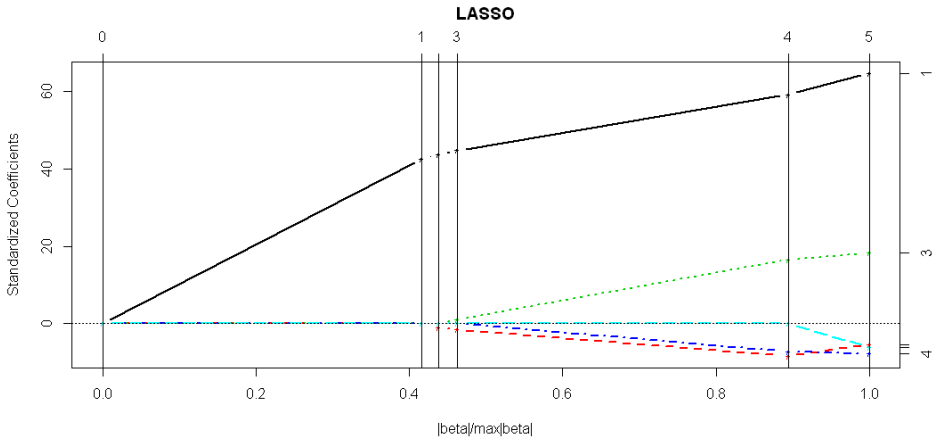


Figure 3.9: The LASSO path computed by lars

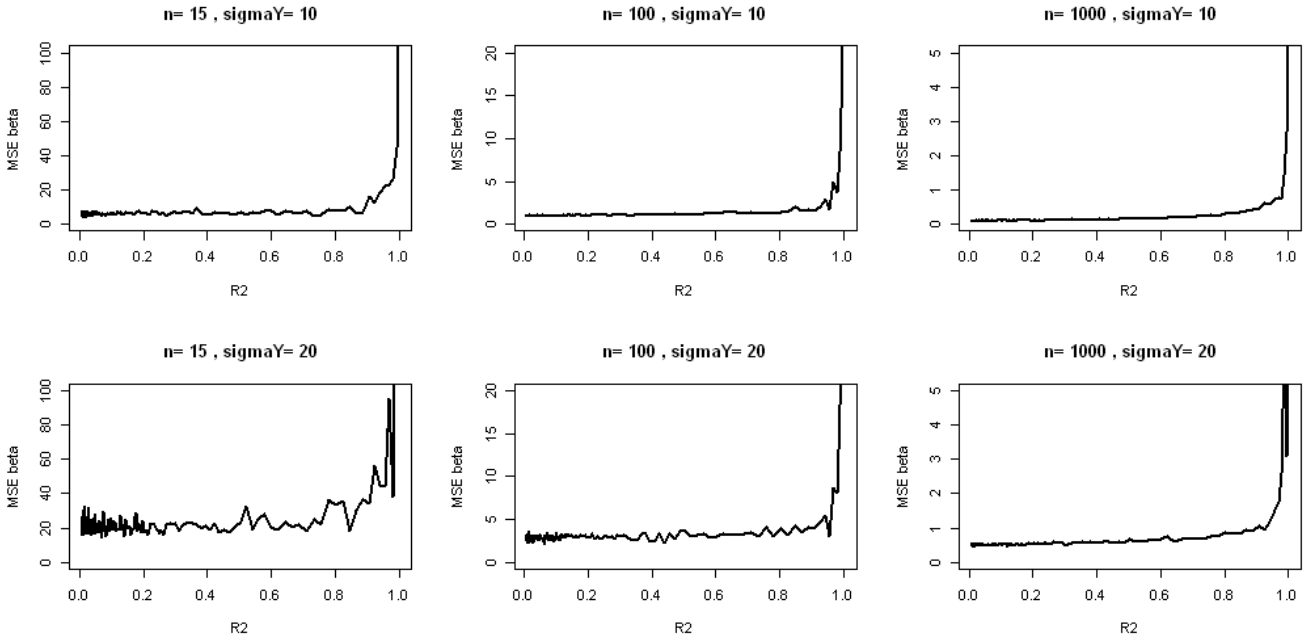


Figure 3.10: Evolution of observed Mean Squared error on $\hat{\boldsymbol{\beta}}_{lar}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

Figure 3.10 shows that strong correlations make the MSE explode even with LAR. Results obtained with the package `lars` for R, included in `CorReg`.

3.3.3 Elasticnet

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

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

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

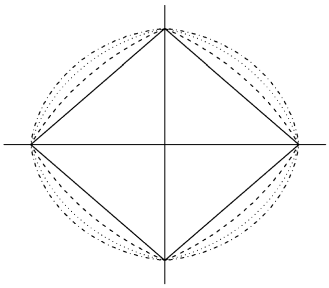


Figure 3.11: Geometric view of the penalty for elasticnet

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 3.12.

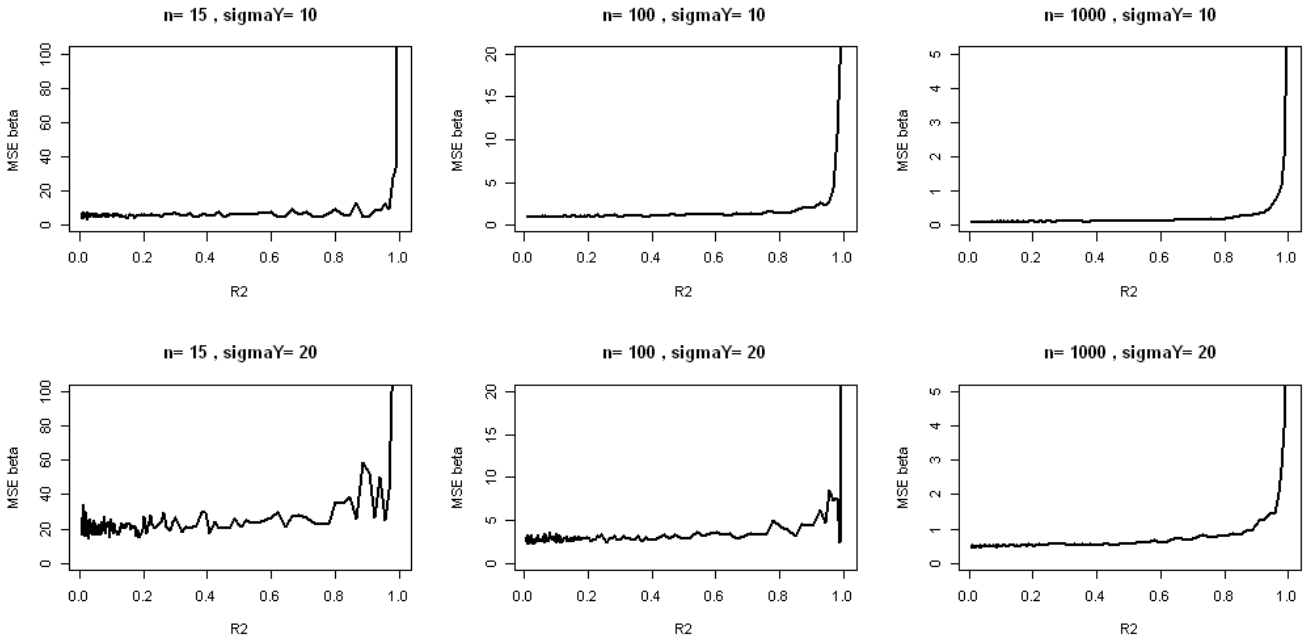


Figure 3.12: Evolution of observed Mean Squared error on $\hat{\boldsymbol{\beta}}_{elasticnet}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

3.3.4 Octagonal Shrinkage and Clustering Algorithm for Regression (OSCAR)

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.

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

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.

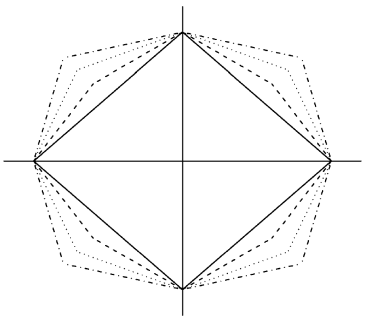


Figure 3.13: Geometric view of the Penalty for OSCAR

Figure 3.13 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.

3.3.5 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. The criterion to optimize can be adjusted R-square, Akaike information criterion, Bayesian information criterion,*etc.*

- 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 3.14 illustrate the consequences of correlations in the dataset.

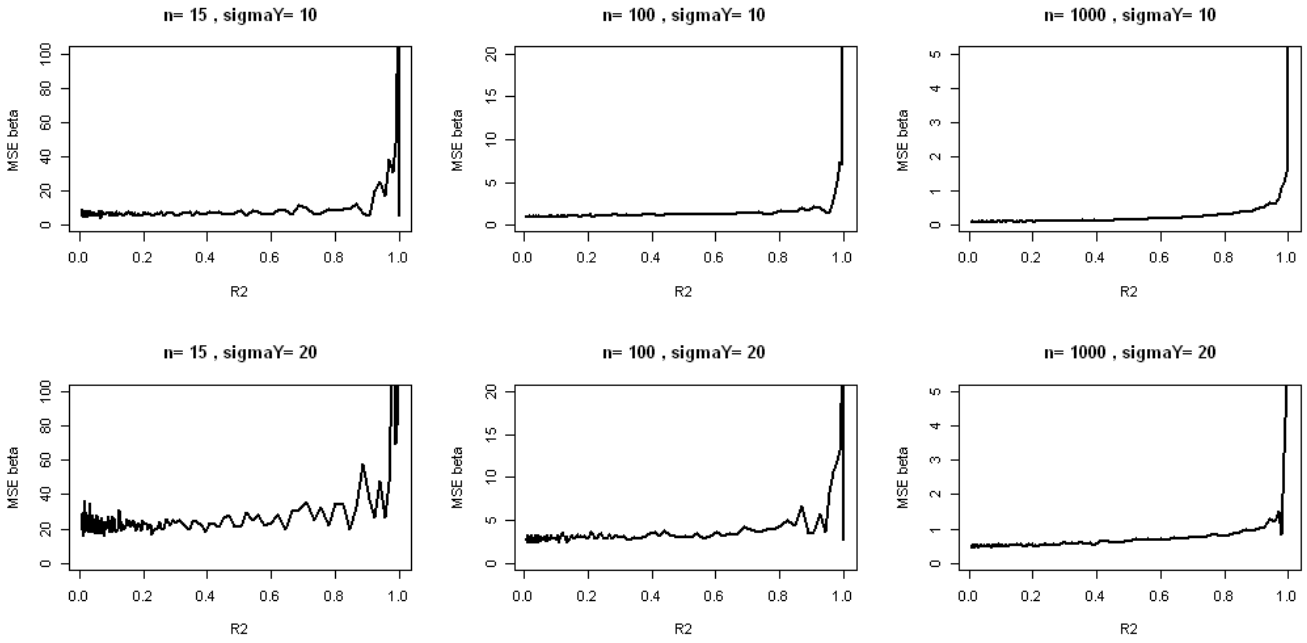


Figure 3.14: Evolution of observed Mean Squared error on $\hat{\boldsymbol{\beta}}_{stepwise}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

3.4 Modeling the parameters

3.4.1 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 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 yeld reliable prediction with a smaller variance on $\hat{\boldsymbol{\beta}}$. A package `clere` for R does exist on CRAN ([Yengo and Canouil, 2014]).

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 or specific model for correlations.

3.4.2 Spike and Slab

Spike and Slab variable selection [Ishwaran and Rao, 2005] also relies on Gaussian mixture (the spike and the slab) hypothesis for the β_j and gives a subset of covariates (not grouped) on which to compute OLS but has no specific protection against correlations issues. The β_j are supposed to come from a mixture distribution as shown in Figure 3.15. It allows to have some coefficients set exactly to zero after some draws. The package

`spikeslab` for R is on CRAN.

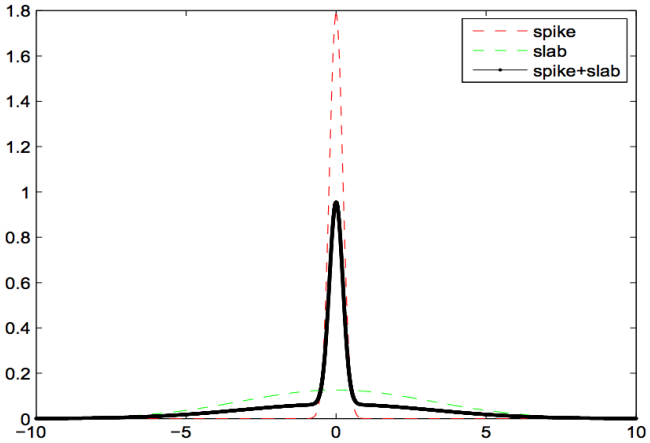


Figure 3.15: The spike and the slab

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.

3.5 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 $\boldsymbol{X}$ instead of $\boldsymbol{X}$ itself. Then we have orthogonal covariates. The dataset $\boldsymbol{X}$ 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 $\boldsymbol{X}^j$.

$$\begin{aligned} \hat{\boldsymbol{\beta}}_M &= \boldsymbol{V}_M \hat{\boldsymbol{\gamma}}_M \in \mathbb{R}^d \text{ where} \\ \hat{\boldsymbol{\gamma}}_M &= (\boldsymbol{W}'_M \boldsymbol{W}_M)^{-1} \boldsymbol{W}'_M \boldsymbol{Y} \text{ with} \\ \boldsymbol{W}_M &= \boldsymbol{X}\boldsymbol{V}_M \end{aligned}$$

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Delta}\boldsymbol{V}'$$

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

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 $\boldsymbol{X}$ instead of $\boldsymbol{Y}$ even if there is no reason that relevant variables to explain $\boldsymbol{X}$ stay relevant to explain $\boldsymbol{Y}$. So a method that would also use $\boldsymbol{Y}$ may give better results in terms of prediction.

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 $\boldsymbol{X}$ but this combination depends on $\boldsymbol{Y}$. The dataset $\boldsymbol{X}$ 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 $\boldsymbol{X}$ and $\boldsymbol{Y}$ with the constraint that linear combination of these components explain as much as possible of the variance of $\boldsymbol{Y}$. It follows an algorithm that leads to construct successively M orthogonal latent variables that are linear combination of the $\boldsymbol{X}^j$:

1. Standardize $\boldsymbol{X}$ and set $\hat{\boldsymbol{Y}}^{(0)} = (\bar{\boldsymbol{Y}}, \ldots, \bar{\boldsymbol{Y}})$ and $\boldsymbol{R}_j^{(0)} = \boldsymbol{X}^j, j = 1, \ldots, d$
2. For $m = 1, \ldots, d$ do:
   (a) $\boldsymbol{z}_m = \sum_{j=1}^{d} \hat{\varphi}_{mj} \boldsymbol{R}_j^{(m-1)}$, where $\hat{\varphi}_{mj} = \langle \boldsymbol{R}_j^{(m-1)}, \boldsymbol{Y} \rangle$.
   (b) $\hat{\theta}_m = \langle \boldsymbol{z}_m, \boldsymbol{Y} \rangle / \langle \boldsymbol{z}_m, \boldsymbol{z}_m \rangle$.
   (c) $\hat{\boldsymbol{Y}}^{(m)} = \hat{\boldsymbol{Y}}^{(m-1)} + \hat{\theta}_m \boldsymbol{z}_m$.
   (d) Orthogonalize each $\boldsymbol{R}_j^{(m-1)}$ with respect to

$$\boldsymbol{z}_m : \boldsymbol{R}_j^{(m)} = \boldsymbol{R}_j^{(m-1)} - \left[\langle \boldsymbol{z}_m, \boldsymbol{R}_j^{(m-1)} \rangle / \langle \boldsymbol{z}_m, \boldsymbol{z}_m \rangle \right] \boldsymbol{z}_m, j = 1, \ldots, d$$

   .
3. Output the sequence of fitted vectors $\{\hat{\boldsymbol{Y}}^{(m)}\}_1^d$. Since the $\{\boldsymbol{z}_l\}_1^M$ are linear in $\boldsymbol{X}$, so is $\hat{\boldsymbol{Y}}^{(M)} = \boldsymbol{X}\hat{\boldsymbol{\beta}}_{PLS}(M)$ where $\hat{\boldsymbol{\beta}}_{PLS}(M)$ for a given value of M can be recovered from the sequence of PLS transformations.

where $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_i A_i B_i$ is the classical scalar product for two equally sized vectors $\boldsymbol{A}$ and $\boldsymbol{B}$.
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 often 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 Seemingly Unrelated Regression (SUR)

Seemingly Unrelated Regression (SUR [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 a system of M regressions jointly instead of independent estimations. The M regressions are:

$$\boldsymbol{Y}_i = \boldsymbol{X}_i\boldsymbol{\beta}_i + \boldsymbol{\varepsilon}_i \ , \quad j = 1, \ldots, M,$$

where $\boldsymbol{Y}_i$ is the $n \times 1$ vector of the response variable of the i^{th} equation, $\boldsymbol{X}_i$ is the $n \times d_i$ matrix of the observations of the d_i explanatory variables for the i^{th} equation, $\boldsymbol{\beta}_i$ is the $d_i \times 1$ vector of the coefficients associated with them and $\boldsymbol{\varepsilon}_i$ is the $n \times 1$ vector of the disturbances with variance σ_i^2. Defining:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Y}_2 \\ \vdots \\ \boldsymbol{Y}_M \end{bmatrix}, \boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{X}_2 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{X}_M \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_M \end{bmatrix}, \boldsymbol{\varepsilon} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \vdots \\ \boldsymbol{\varepsilon}_M \end{bmatrix},$$

the model is then expressed:

$$\boldsymbol{Y} \ = \ \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

This estimation relies on Feasible Generalized Least Squares (FGLS) that depends on $\boldsymbol{\Sigma}$, the $M \times M$ variance-covariance matrix of the error terms:

$$\hat{\boldsymbol{\beta}}_{FGLS} = [\boldsymbol{X}'(\hat{\boldsymbol{\Sigma}} \otimes \boldsymbol{I}_n)\boldsymbol{X}]^{-1}\boldsymbol{X}'(\hat{\boldsymbol{\Sigma}} \otimes \boldsymbol{I}_n)\boldsymbol{Y},$$

where $\hat{\boldsymbol{\Sigma}}$ is a consistent estimator of $\boldsymbol{\Sigma}$, $\boldsymbol{I}_n$ is the $n \times n$ identity matrix and $\otimes$ is the Kronecker product. 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 context[2]. In other words, the population

[2] `http://www.math.univ-toulouse.fr/~maugis/SelvarClustHomepage.html`

is supposed to come from several sub-populations (called the clusters), each following a Gaussian multivariate distribution with d dimensions. The idea is then 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:

$$\boldsymbol{X}^U = a + \boldsymbol{X}^R \boldsymbol{\alpha} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{X}$ is the $n \times d$ dataset as above, $\boldsymbol{X}^R$ and $\boldsymbol{X}^U$ are the sub-matrix of $\boldsymbol{X}$ 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 $\boldsymbol{X}$ in three terms $f_{clust}, f_{reg}, f_{indep}$ whose product will give the joint density of $\boldsymbol{X}$:

- The relevant variables for clustering of $\boldsymbol{X}$ 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):

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

  where the parameter vector is $\boldsymbol{\theta} = (p_1, \ldots, p_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k)$, with $\sum_{k=1}^{K} p_k = 1$, the proportion vector and the variance matrices satisfying the form m. Φ is the Gaussian density function (and will always be in the following).

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

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

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

- 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:

  $$f_{indep}(\boldsymbol{X}^W; l, \gamma, \tau) = \prod_{i=1}^{n} \Phi(\boldsymbol{X}_i^W; \gamma, \tau)$$

`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 do not 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 [Raftery and Dean, 2006] without protection against correlations even if it is known to be often unstable [Miller, 2002] in such a context.

In this work we propose to build a similar model of explicit dependencies between the covariates and to use it as a pre-treatment for linear regression on correlated covariates. We will see that, as a pre-treatment, 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 6.2.1.

Aiming to realize Gaussian clustering, `Selvarclust` does estimate a multivariate Gaussian mixture on $\boldsymbol{X}$ with dependencies within the covariates. Our algorithm will only aim to find a linear sub-regression structure within $\boldsymbol{X}$, 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 6.2.1 are just informative.

3.6 Conclusion

In linear regression there is a lack of methods that both select relevant covariates and manage correlations with strong interpretability. However, some methods already give some partial solutions, sometimes in the field of interpretation by variable selection and/or by creation of groups of covariates, sometimes in the field of prediction by conditioning improvements. We will try to act on both fields (with priority given to interpretation) and then to allow usage of other methods on top of the model that we will propose in the following.

Part I

Model for regression with correlation-free covariates

Chapter 4

Structure of inter-covariates regressions

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 pre-treatment by variable selection. This pre-treatment 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 pre-treatment 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 $\boldsymbol{X} = (\boldsymbol{X}^1, \ldots, \boldsymbol{X}^d)$ be a $n \times d$ matrix of observed covariates and $\boldsymbol{Y}$ be the $n \times 1$ matrix of the observed response variable. In the following, we note $\boldsymbol{X}^j$ the j^{th} column of $\boldsymbol{X}$ and $\boldsymbol{X}^J$ where $J = \{j_1, \ldots, j_k\}$ the $n \times k$ sub-matrix of $\boldsymbol{X}$ composed of the columns of $\boldsymbol{X}$ 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 $j = 1, \ldots, d_r$ having the covariate $\boldsymbol{X}^{J_r^j}$ as response variable ($J_r^j \in \{1, \ldots, p\}$ and $J_r^j \neq J_r^{j'}$ if $j \neq j'$) and having the $d_p^j > 0$ covariates $\boldsymbol{X}^{J_p^j}$ as predictor variables ($J_p^j \subset \{1, \ldots, d\} \backslash J_r^j$ and $d_p^j = |J_p^j|$ the cardinal of J_p^j):*

$$\boldsymbol{X}^{J_r^j} = \boldsymbol{X}^{J_p^j} \boldsymbol{\alpha}_j + \boldsymbol{\varepsilon}_j, \tag{4.1}$$

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

Hypothesis 2 *In addition, we make the complementary hypothesis that the response covariates and the predictor covariates are totally disjoint: for any sub-regression $j = 1, \ldots, d_r$, $J_p^j \subset J_f$ where $J_r = \{J_r^1, \ldots, J_r^{d_r}\}$ is set of all response covariates and $J_f = \{1, \ldots, d\} \backslash J_r$ 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 lose 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 $\boldsymbol{J}_r = (J_r^1, \ldots, J_r^{d_r})$ the d_r-uple of all the response variables (not to be confused with the corresponding set J_r previously defined), $\boldsymbol{J}_p = (J_p^1, \ldots, J_p^{d_r})$ the d_r-uple of all the predictors for all the sub-regressions, $\boldsymbol{d}_p = (d_p^1, \ldots, d_p^{d_r})$ the associated number of predictors and $\boldsymbol{S} = (\boldsymbol{J}_r, \boldsymbol{J}_p)$ the global *model* of all the sub-regressions. As more compact notations, we define also $\boldsymbol{X}_r = \boldsymbol{X}^{J_r}$ the whole set of response covariates and also $\boldsymbol{X}_f = \boldsymbol{X}^{J_f}$ the *all* other covariates, denominating now as *free* covariates, including those used as predictor covariates in $\boldsymbol{J}_p$. An illustration of all these notations is applied on the running example at the end of this section. The parameters are also stacked together: $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{d_r})$ denotes the global coefficient of sub-regressions and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_{d_r}^2)$ 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.

- We have considered correlations between the covariates of the main regression on $\boldsymbol{Y}$, not between the residuals. Thus $\boldsymbol{S}$ does not depend on $\boldsymbol{Y}$ and it can be estimated independently as we will see in Chapter 5, even with a larger dataset (if missing values in $\boldsymbol{Y}$).
- The model of sub-regressions $\boldsymbol{S}$ gives a system of linear regressions that can be viewed ([Davidson and MacKinnon, 1993, Timm, 2002]) as a recursive Simultaneous Equation Model (SEM) or also as a Seemingly Unrelated Regression (SUR) [Zellner, 1962] with $\boldsymbol{X}_r$ the set of endogenous covariates.
- Each sub-regression may imply a distinct subset of explicative covariates from $\boldsymbol{X}_f$.
- 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 $\boldsymbol{X}_f = (\boldsymbol{X}^1, \boldsymbol{X}^2, \boldsymbol{X}^4, \boldsymbol{X}^5)$. We have $d_r = 1$ response covariate $\boldsymbol{X}^3 = \boldsymbol{X}^1 + \boldsymbol{X}^2 + \boldsymbol{\varepsilon}_1$ where $\boldsymbol{\varepsilon}_1 \sim \mathcal{N}_n(\boldsymbol{0}, \sigma_1^2 \boldsymbol{I})$. Thus, $\boldsymbol{\alpha}_1 = (1, 1)'$, $\boldsymbol{J}_r = (3)$, $J_r = \{3\}$, $\boldsymbol{X}_r = (\boldsymbol{X}^3)$, $\boldsymbol{d}_p = (2)$, $\boldsymbol{J}_p = (\{1, 2\})$, $\boldsymbol{X}^{J_p^1} = (\boldsymbol{X}^1, \boldsymbol{X}^2)$ and $\boldsymbol{S} = ((3), (\{1, 2\}))$.

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

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

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

We need the following new hypothesis:

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

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

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):

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

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 $\{\boldsymbol{Y}|\boldsymbol{X}_f, \boldsymbol{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2\}$:

$$\begin{aligned} \boldsymbol{Y}|\boldsymbol{X}_f, \boldsymbol{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2 &= \boldsymbol{X}_f(\boldsymbol{\beta}_f + \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\alpha}_j^*) + \sum_{j=1}^{d_r} \beta_{J_r^j} \boldsymbol{\varepsilon}_j + \boldsymbol{\varepsilon}_Y \quad (4.5) \\ &= \boldsymbol{X}_f \boldsymbol{\beta}_f^* + \boldsymbol{\varepsilon}_Y^*, \quad (4.6) \end{aligned}$$

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 notations:

$$\begin{aligned} \boldsymbol{X}_r|\boldsymbol{X}_f, \boldsymbol{S}; \boldsymbol{\alpha}, \boldsymbol{\sigma}^2 &= \boldsymbol{X}_f\boldsymbol{\alpha}^* + \boldsymbol{\varepsilon} \\ \boldsymbol{Y}|\boldsymbol{X}_f, \boldsymbol{S}; \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma_Y^2, \boldsymbol{\sigma}^2 &= \boldsymbol{X}_f(\boldsymbol{\beta}_f + \boldsymbol{\alpha}^*\boldsymbol{\beta}_r) + \boldsymbol{\varepsilon}\boldsymbol{\beta}_r + \boldsymbol{\varepsilon}_Y \end{aligned} \tag{4.7}$$

Where $\boldsymbol{\varepsilon}$ is the $n \times d_r$ matrix whose columns are the $\boldsymbol{\varepsilon}_j$, the noises of the sub-regressions.

Consequently, we have obtained a new regression expression of $\boldsymbol{Y}$ but relying now *only* on uncorrelated covariates $\boldsymbol{X}_f$. This decorrelation process has also acted like a specific variable selection process because $\boldsymbol{X}_f \subseteq \boldsymbol{X}$. 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 (or equal) residual variance than the initial one since it is now $\sigma_Y^{2*} = \sigma_Y^2 + \sum_{j=1}^{d_r} \beta^2_{J_r^j}\sigma_j^2$ 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 4.8.

In practice, the strategy we propose is to rely estimate of $\hat{\boldsymbol{\beta}}$ upon the reduced model given in 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 *pre-treatment* by pre-selection to decorrelate 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.

Remarks:

- Identifiability of $(\boldsymbol{X}, \boldsymbol{S})$ 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 of this criterion in appendix A). In the following, true $\boldsymbol{S}$ is supposed to be identifiable.
- As a consequence of Hypotheses 1 to 3 , "free" covariates $\boldsymbol{X}_f$ are *all* decorrelated (see the Lemma in appendix A).

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

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 suppose $\boldsymbol{\beta}_r^* = \mathbf{0}$.

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 $\boldsymbol{S}$) and the classical selection step, with different meanings for obtained zeros in $\hat{\boldsymbol{\beta}}^*_f$ (irrelevant covariates) and for $\hat{\boldsymbol{\beta}}^*_r = 0$ (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 $\boldsymbol{X}$) guarantee to keep a simple structure easily interpretable (no cycles and no chain-effect) and straightforward readable.

There is no theoretical guarantee that our model is better. It is 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 trade-off.

4.5 Illustration of the trade-off conveyed by the pre-treatment

We compare the OLS estimator on $\boldsymbol{X}$ defined in section 3.2.1 with the estimator obtained by the pre-treatment that is $\boldsymbol{X}_f$ selection.

For the marginal regression model defined in (4.6) we have the OLS unbiased estimator of $\boldsymbol{\beta}^*$:

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

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

$$\mathbb{E}(\hat{\boldsymbol{\beta}}^*_f) = \boldsymbol{\beta}_f + \sum_{j=1}^{d_r} \beta_{J^j_r}\boldsymbol{\alpha}^*_j \text{ and } \mathbb{E}(\hat{\boldsymbol{\beta}}^*_r) = \mathbf{0}$$

In return, its variance could be reduced compared to this one of $\hat{\boldsymbol{\beta}}$ 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

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}^*_f) = (\sigma^2_Y + \sum_{j=1}^{d_r} \sigma^2_j\beta^2_{J^j_r})(\boldsymbol{X}'_f\boldsymbol{X}_f)^{-1} \quad \text{and} \quad \mathrm{Var}(\hat{\boldsymbol{\beta}}^*_r) = \mathbf{0}. \tag{4.8}$$

Indeed, no correlations between covariates $\boldsymbol{X}_f$ imply that the matrix $\boldsymbol{X}'_f\boldsymbol{X}_f$ could be sufficiently better conditioned than the matrix $\boldsymbol{X}'\boldsymbol{X}$ involved in (3.3) to balance the added variance $\sum_{j=1}^{d_r} \sigma^2_j\beta^2_{J^j_r}$ in (4.8). This bias-variance trade off can be resumed by the Mean Squared Error (MSE) associated to both estimates:

$$\begin{aligned}
\mathrm{MSE}(\hat{\boldsymbol{\beta}}) &= \| \text{ Bias } \|^2_2 + \mathrm{Tr}(\mathrm{Var}(\hat{\boldsymbol{\beta}})) \\
&= \sigma^2_Y \,\mathrm{Tr}((\boldsymbol{X}'\boldsymbol{X})^{-1}), \\
\mathrm{MSE}(\hat{\boldsymbol{\beta}}^*) &= \| \sum_{j=1}^{d_r} \beta_{J^j_r}\boldsymbol{\alpha}^*_j \|^2_2 + \| \boldsymbol{\beta}_r \|^2_2 + (\sigma^2_Y + \sum_{j=1}^{d_r} \sigma^2_j\beta^2_{J^j_r})\,\mathrm{Tr}((\boldsymbol{X}'_f\boldsymbol{X}_f)^{-1}).
\end{aligned}$$

When n rises to $+\infty$, variances of both estimator tends to 0 but the bias of the reduced model remains so the complete model would be better asymptotically in n. But for a fixed value of n, when the σ_j's tend to 0 then the variance of the complete model tends to explode whereas the variance of the reduced model and its bias remain stable (the factor before the inverse even shrinks to σ_Y^2). So the reduced model would be better for strong linear relationship between the covariates. We finally observe for the reduced model that when $\boldsymbol{\beta}_r = 0$ there is no bias, the number of parameters to estimate is smaller than for the complete model and the matrix to invert is well-conditioned. It is the true model estimated with uncorrelated covariates and "knowing" $\boldsymbol{\beta}_r = 0$ so it will give better results.

We also get an estimation of the residual $\boldsymbol{\varepsilon}_Y^*$ relying on $\hat{\boldsymbol{\beta}}_f^*$:

$$\hat{\boldsymbol{\varepsilon}}_Y^* = \boldsymbol{Y} - \boldsymbol{X}_f \hat{\boldsymbol{\beta}}_f^* \tag{4.9}$$

that will be used in Chapter 8. And then:

$$\hat{\boldsymbol{Y}}_{marginal} = \boldsymbol{X}_f \hat{\boldsymbol{\beta}}_f^*.$$

The bias-variance trade-off conveyed by the pre-treatment is then illustrated with the running example (Section 4.7).

4.6 Connexion with graphs

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

Motivations: Graphical representation of $\boldsymbol{S}$ helps to understand it and can be compared to the Bayesian network or Simultaneous Equation Modeling representation. It helps to interpret the structure and has also been used to construct the algorithm to find $\boldsymbol{S}$ (Chapter 5). Another advantage of the modelization by a DAG is that it helps to obtain the cardinal of $\mathcal{S}_d$, the set of the feasible structures for d covariates.

The partition of $\boldsymbol{X}$ means that the associated graph is bipartite: vertices follow a partition $(\boldsymbol{X}_r, \boldsymbol{X}_f)$ with directed edges only going from $\boldsymbol{X}_f$ to $\boldsymbol{X}_r$. We know ([Biggs, 1993]) as a classical result of graph theory that the power of adjacency matrices give the paths in the graph: $\boldsymbol{G}_{i,j}^k \neq 0$ means that there is at least a path of length k going from $\boldsymbol{X}^i$ to $\boldsymbol{X}^j$. 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 $\boldsymbol{G}$ is nilpotent: $\boldsymbol{G}^2 = 0$. 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*. Then the number of feasible structures with d covariates is the number of d-sized binary nilpotent matrices of order 2:

- $\forall \tilde{\boldsymbol{G}} \in \mathcal{M}_{b,d}, \tilde{\boldsymbol{G}}^2 = 0 \Rightarrow \tilde{\boldsymbol{S}} \in \mathcal{S}_d$,
- $\forall \tilde{\boldsymbol{S}} \in \mathcal{S}_d, \tilde{\boldsymbol{G}}^2 = 0$,

where $\tilde{\boldsymbol{G}}$ is the $d \times d$ adjacency matrix associated to the structure $\tilde{\boldsymbol{S}}$ and $\mathcal{M}_{b,d}$ is the set of $d \times d$ binary matrices (filled only with 1 and 0).

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 $\boldsymbol{G}$. If there exists a path of length 2 between some vertices i and j (uncrossing rule violated) then $\boldsymbol{G}^2_{i,j} \neq 0$ so the matrix is not nilpotent of order 2. So the set of nilpotent matrices includes the set of adjacency matrices associated to $\mathcal{S}_d$. If $\boldsymbol{G}$ 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. □

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

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

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

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:

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

4.7 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 $\boldsymbol{X}$ and $\boldsymbol{Y}$ from the running example.

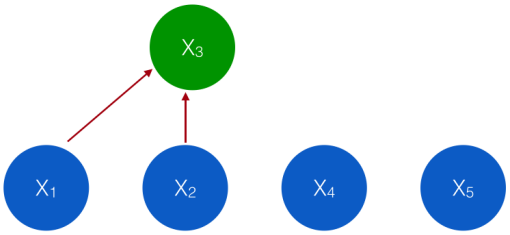


Figure 4.1: The bipartite graph associated to the running example. $\boldsymbol{X}_r$ is green and $\boldsymbol{X}_f$ is blue.

The MSE on $\hat{\boldsymbol{Y}}$ is computed on a validation sample with 1 000 individuals.
Dataset generation, results and curves come from our `CorReg` package.

Observed MSE for OLS: It is clear in figures 4.2 and 4.3 that the marginal model is more robust than OLS on $\boldsymbol{X}$. Colored areas indicate which curve has the minimum value for faster comparison of the curves. This kind of plot will be widely used in this document. Here blue areas stand for the marginal model so our marginal model is better (in terms of MSE) when the background is blue. We see in figures 4.2 and 4.3 that MSE on $\hat{\boldsymbol{Y}}_{OLS}$ give the same global results as those on $\hat{\boldsymbol{\beta}}_{OLS}$: the marginal model is better for stronger sub-regressions, smaller samples and weaker main regression. But we notice that when the MSE on $\hat{\boldsymbol{\beta}}_{OLS}$ explodes, the MSE on $\hat{\boldsymbol{Y}}_{OLS}$ does not grow so much. This is a good illustration of the problem generated by the correlations. The model seems to be good in prediction but coefficients are very far from the real value and interpretation can be extremely misleading.

When sub-regression get weaker (R^2 tends to 0) CorReg (our reduced model) 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 $\boldsymbol{Y}$ 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.

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

Observed MSE for variable selection methods: Figure 4.4 shows that variable selection done by the LASSO gives a biased $\hat{\boldsymbol{\beta}}$ by setting some coefficients to 0 but strong correlations makes this bias neutral for prediction (figure 4.5). Here the LASSO tends to propose the same model as we do with our marginal model, but without explanation. We will see later in section 6.3 that it is not sufficient in higher dimension. Elasticnet and stepwise give results quite similar to the LASSO (figures 4.6 to 4.9).

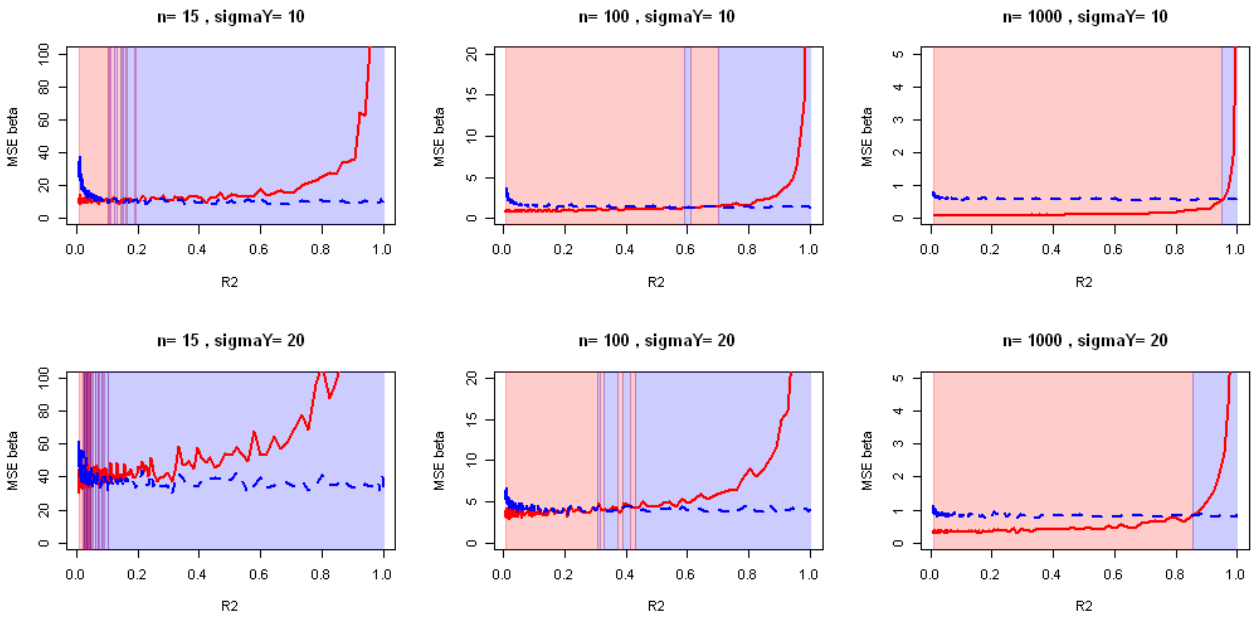


Figure 4.2: Observed MSE on $\hat{\boldsymbol{\beta}}$ of OLS (plain red) and `CorReg`'s marginal model (dashed blue) estimators for varying R^2 of the sub-regression, n and σ_Y. $d = 5$ covariates.

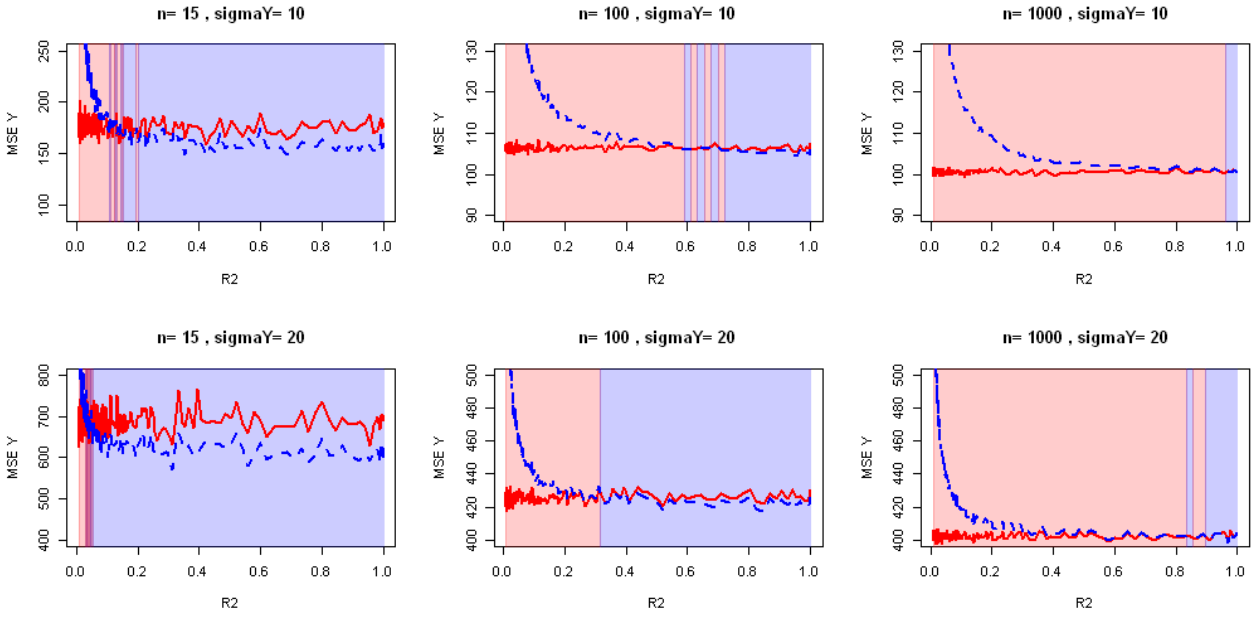


Figure 4.3: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{OLS}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

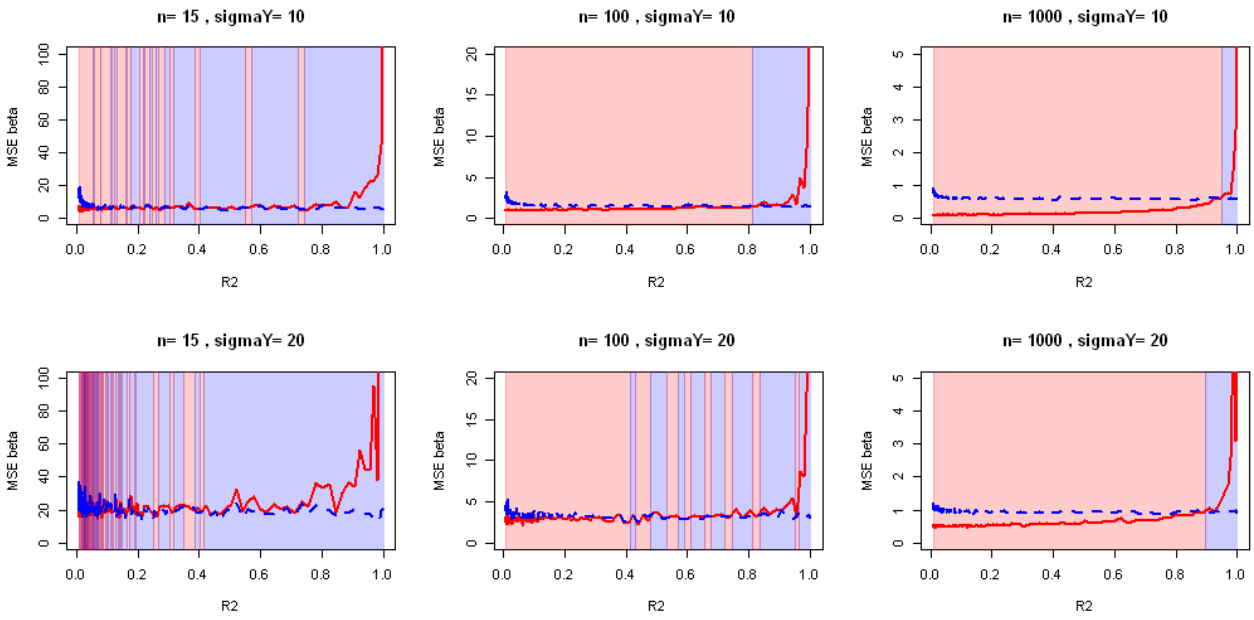


Figure 4.4: Observed MSE on $\hat{\boldsymbol{\beta}}$ of LASSO with LAR on both $\boldsymbol{X}$ (red) and `CorReg`'s marginal $\boldsymbol{X}^{I_f}$ (blue) for varying R^2 of the sub-regression, n and σ_Y. $d = 5$ covariates.

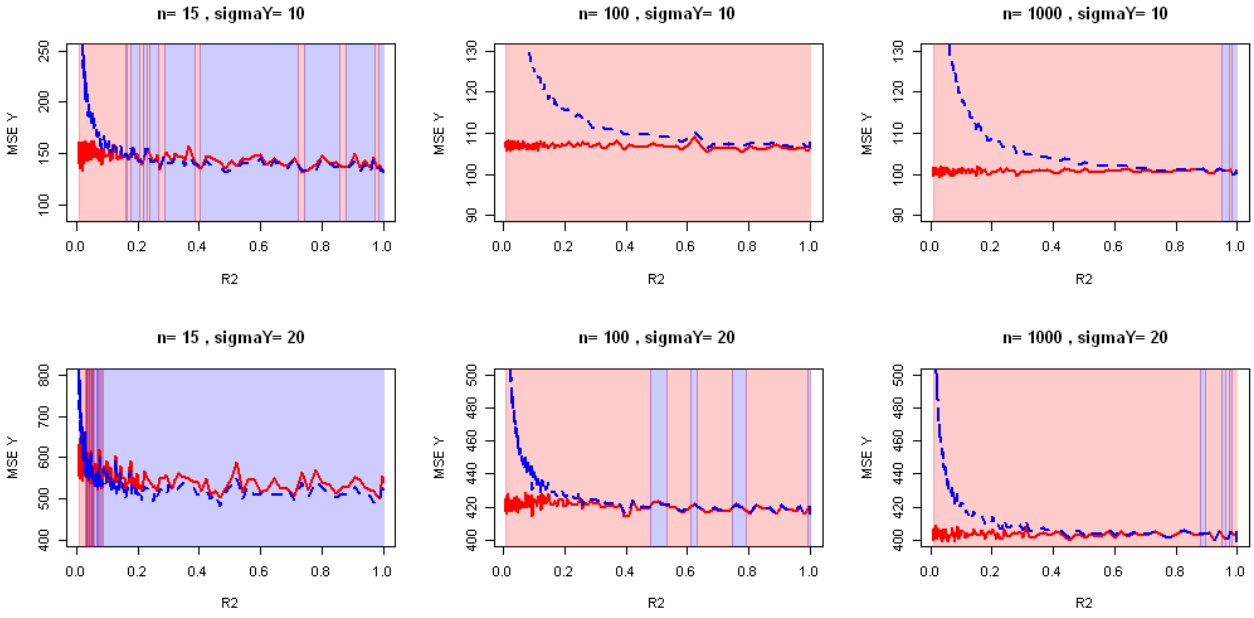


Figure 4.5: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{LASSO}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

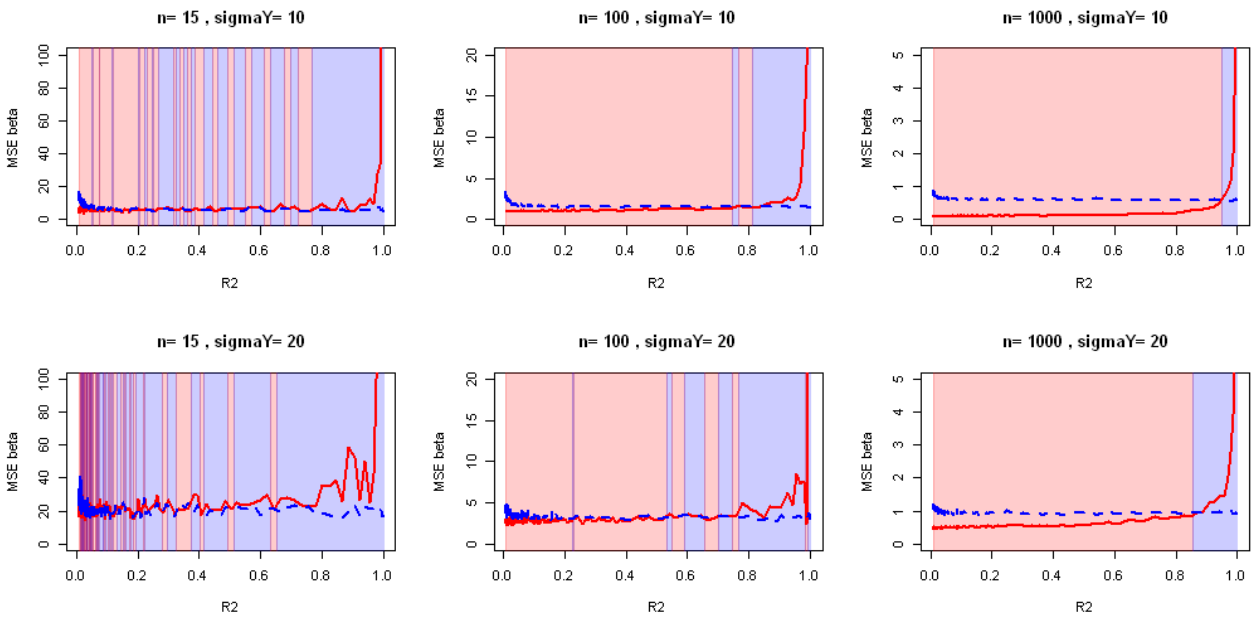


Figure 4.6: Observed MSE on $\hat{\boldsymbol{\beta}}_{elasticnet}$ on both $\boldsymbol{X}$ (red) and `CorReg`'s marginal $\boldsymbol{X}_f$ (blue) for varying R^2 of the sub-regression, n and σ_Y. $d = 5$ covariates.

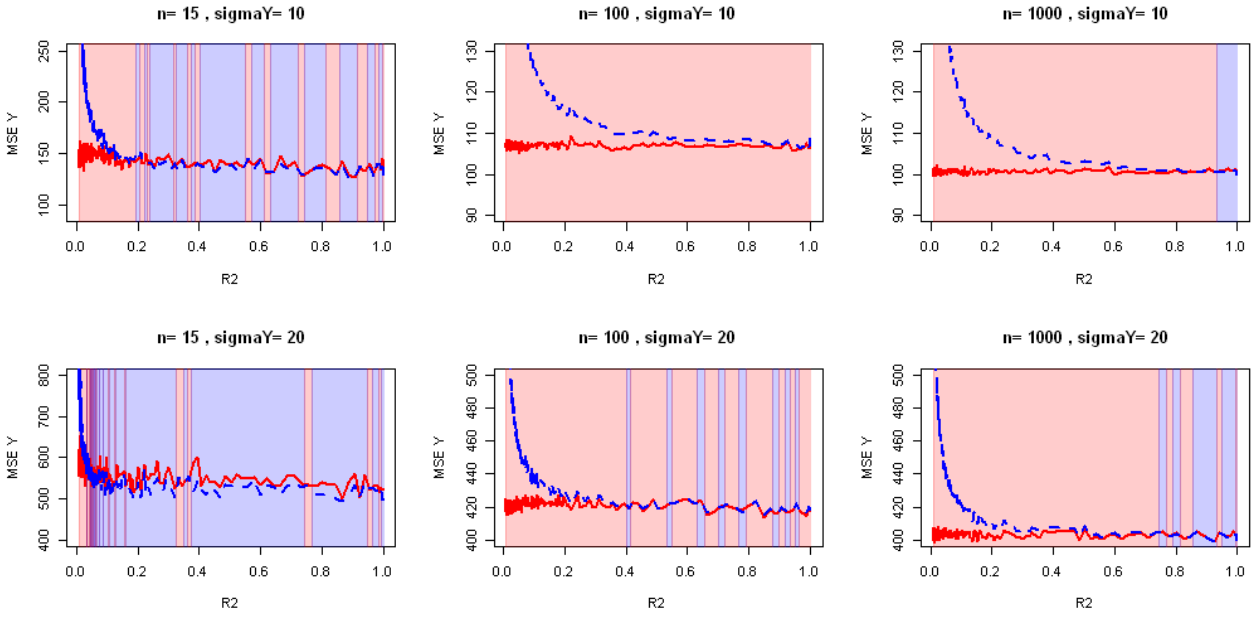


Figure 4.7: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{elasticnet}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

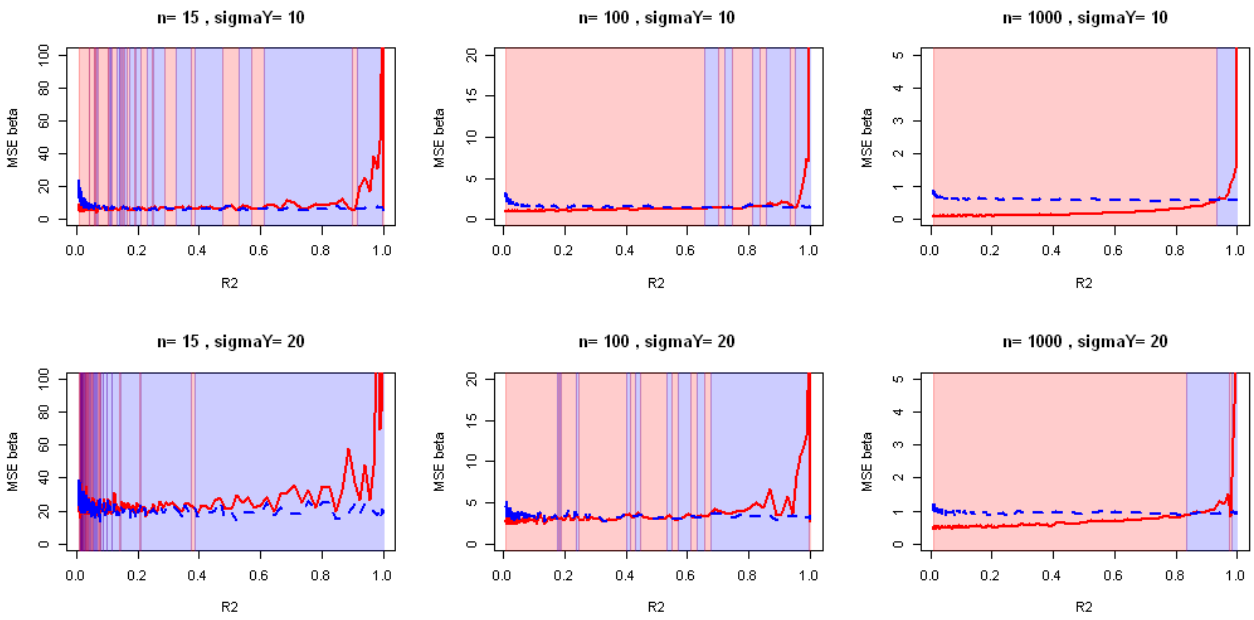


Figure 4.8: Observed MSE on $\hat{\boldsymbol{\beta}}_{stepwise}$ on both $\boldsymbol{X}$ (red) and `CorReg`'s marginal $\boldsymbol{X}_f$ (blue) for varying R^2 of the sub-regression, n and σ_Y. $d = 5$ covariates.

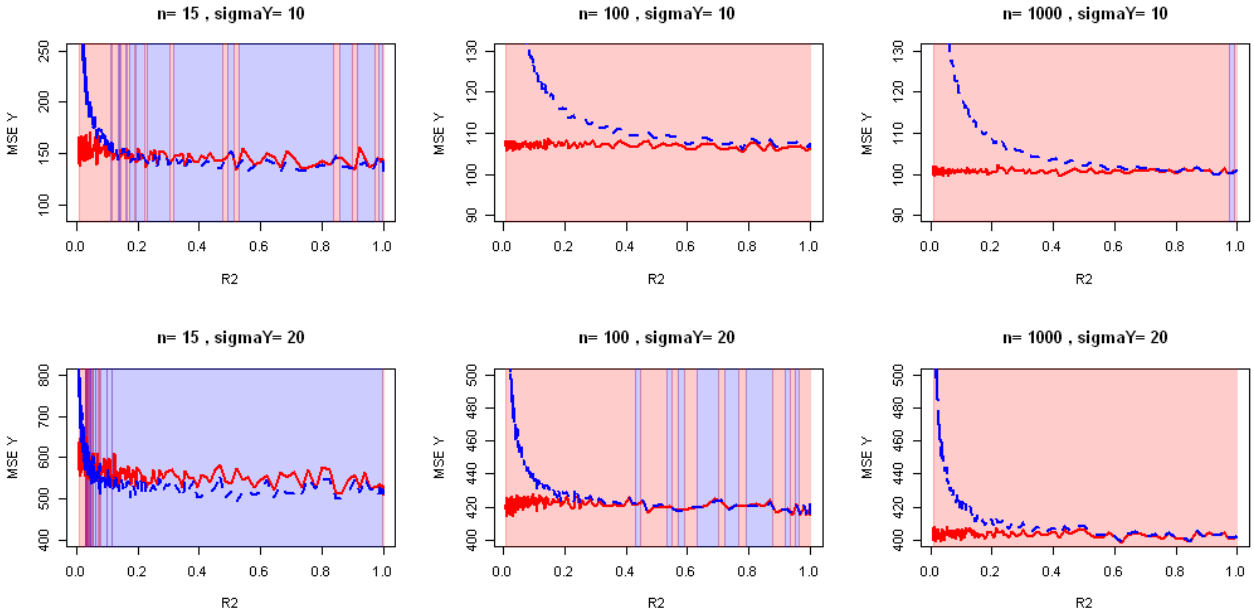


Figure 4.9: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{stepwise}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

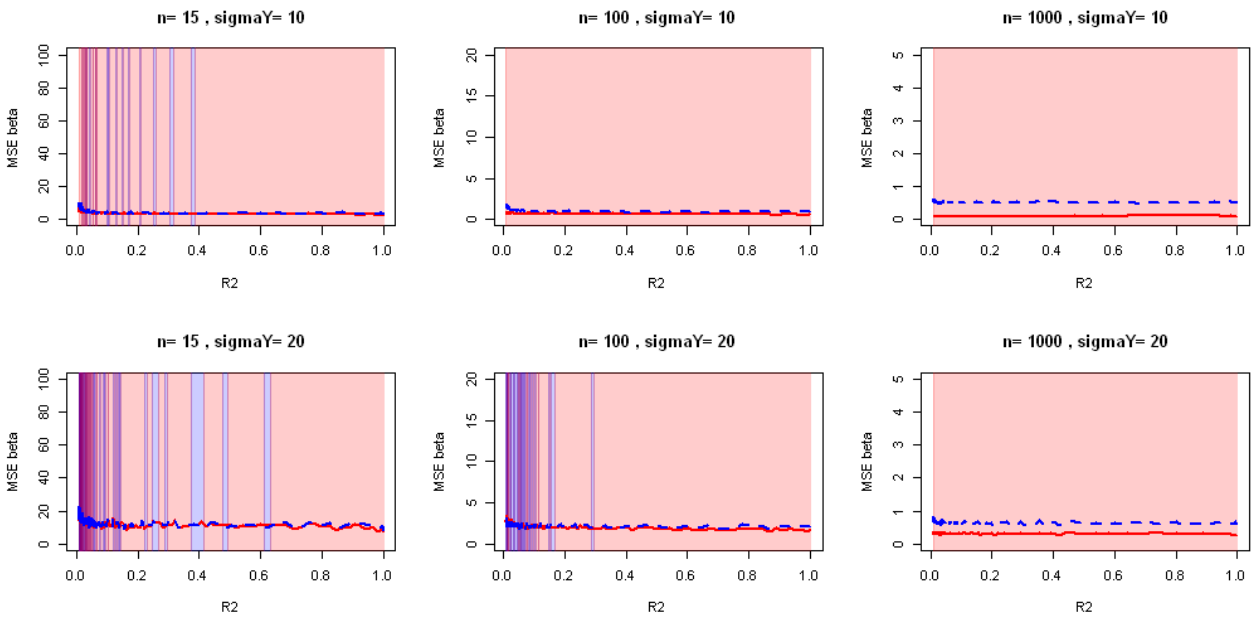


Figure 4.10: Observed MSE on $\hat{\boldsymbol{\beta}}_{ridge}$ on both $\boldsymbol{X}$ (red) and `CorReg`'s marginal $\boldsymbol{X}_f$ (blue) for varying R^2 of the sub-regression, n and σ_Y. $d = 5$ covariates.

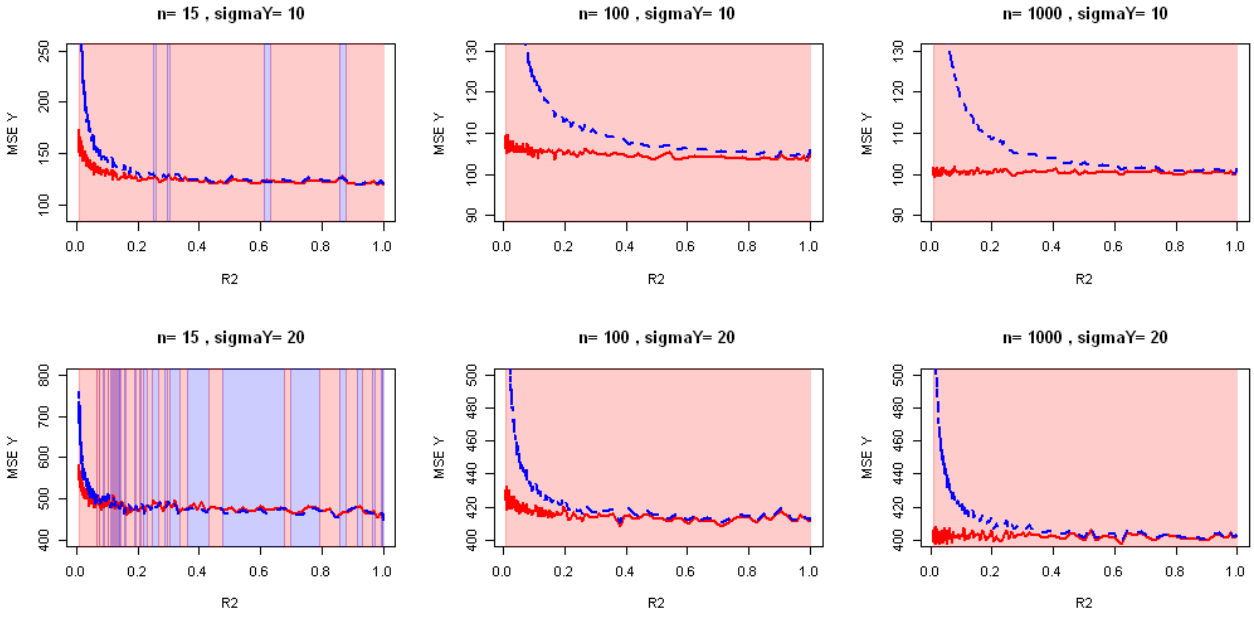


Figure 4.11: Evolution of observed Mean Squared error on $\hat{\boldsymbol{Y}}_{ridge}$ with the strength of the correlations for various sample sizes and strength of regression. $d = 5$ covariates.

Observed MSE for ridge regression: Here again (figures 4.10 and 4.11), ridge regression provides good results for this running example. But we will see later in section 6.3 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.

4.8 Numerical results with a known structure on more complex datasets

Here are the first numerical results obtained for a known structure $\boldsymbol{S}$ on simulated datasets. It illustrates efficiency of the variable selection pre-treatment made by the marginal model.

4.8.1 The datasets

We consider regressions on $\boldsymbol{Y}$ with $d = 40$ covariates and with a R^2 value equal to 0.4. Sub-regressions will have R^2 successively set to $(0.1, 0.3, 0.5, 0.7, 0.99)$. Each Variable in $\boldsymbol{X}_f$ arise from a Gaussian mixture (real covariates won't be Gaussian) 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 ($\forall j = 1, \ldots, d_r, d_p^j = 2$ 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 $\boldsymbol{S}$ used to generate the dataset (function `mixture_genererator` 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 estimation methods on both the complete and the reduced model. All the results are provided by the `CorReg` package. Here $\boldsymbol{Y}$ depends on all the covariate and the MSE provided were computed on a validation sample of 1 000 individuals each time. Figures will display both mean and inter-quartile intervals, coloration of the background indicates which curve is lower (*i.e.* better in our case).

4.8.2 Results when the response depends on all the covariates, true structure known

As previously explained, we have added colored backgrounds for faster comparison of the curves. The background takes the same color as the curve that is the lowest. So we want to obtain as much blue areas as possible,meaning our marginal model is better (in terms of MSE or in terms of parsimony).

With OLS: Figure 4.12 compares OLS on $\boldsymbol{X}$ to OLS on $\boldsymbol{X}_f$ that is our reduced model. CorReg's pre-selection 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 $\boldsymbol{Y}$ depends on all $\boldsymbol{X}$ because of the marginalization of some covariates implicated in the sub-regressions. It also illustrates that the regression in $\boldsymbol{Y}$ retained by CorReg is more parsimonious.

Variable selection methods: Figure 4.13 shows that even if the LASSO is able to select a subset of covariates 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 pre-treatment by selection is just a pre-treatment 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 $\boldsymbol{X}_f$ when used on the marginal model. The main result here is that the LASSO can be improved by pre-treatment 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 4.14 and 4.15)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 $\boldsymbol{S}$ and not only variable selection by stepwise like in the method from Maugis [Maugis et al., 2009].

Ridge regression: Figure 4.16 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 $\boldsymbol{X}_f$ (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.

Using OLS estimation

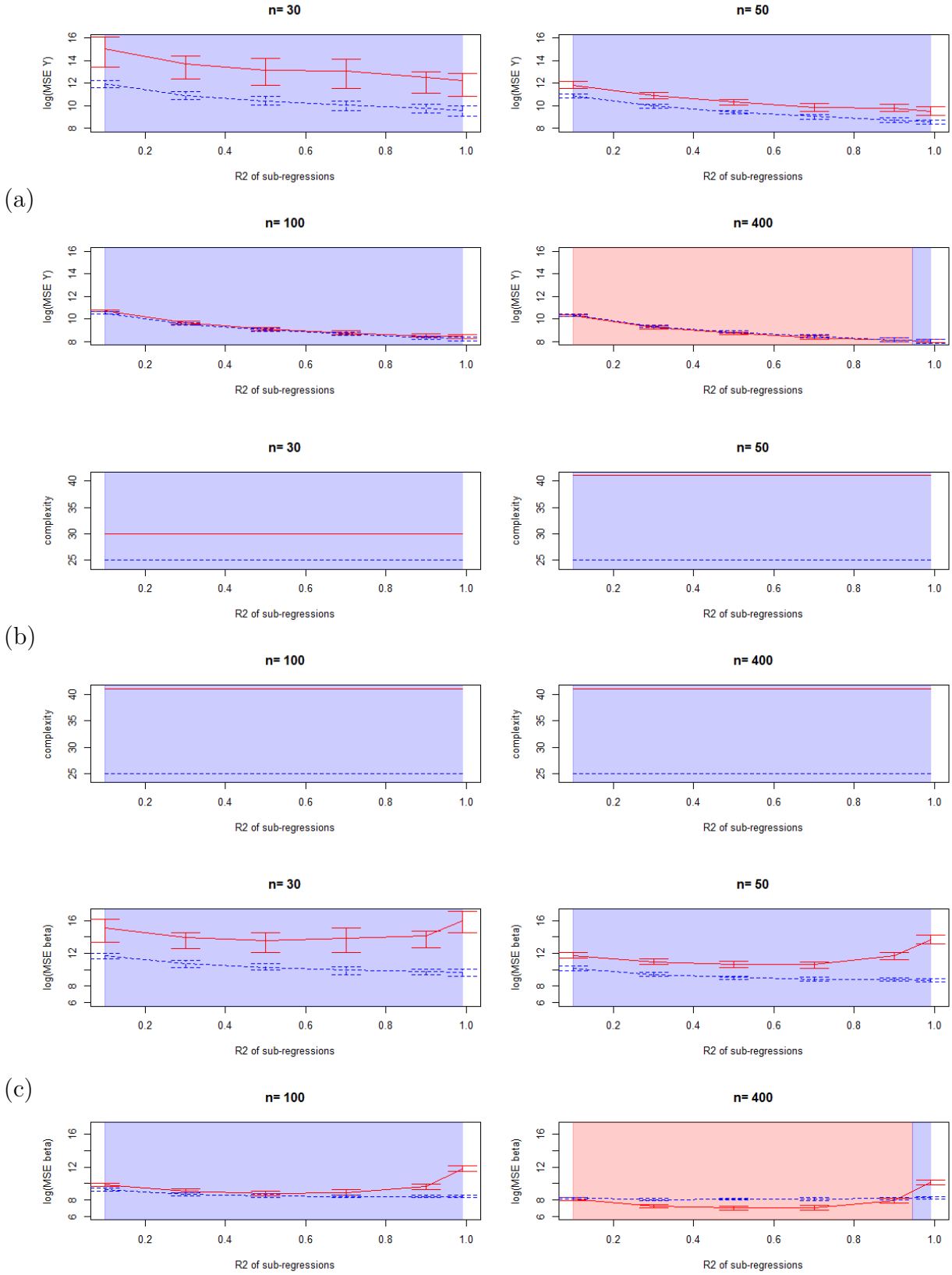


Figure 4.12: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model

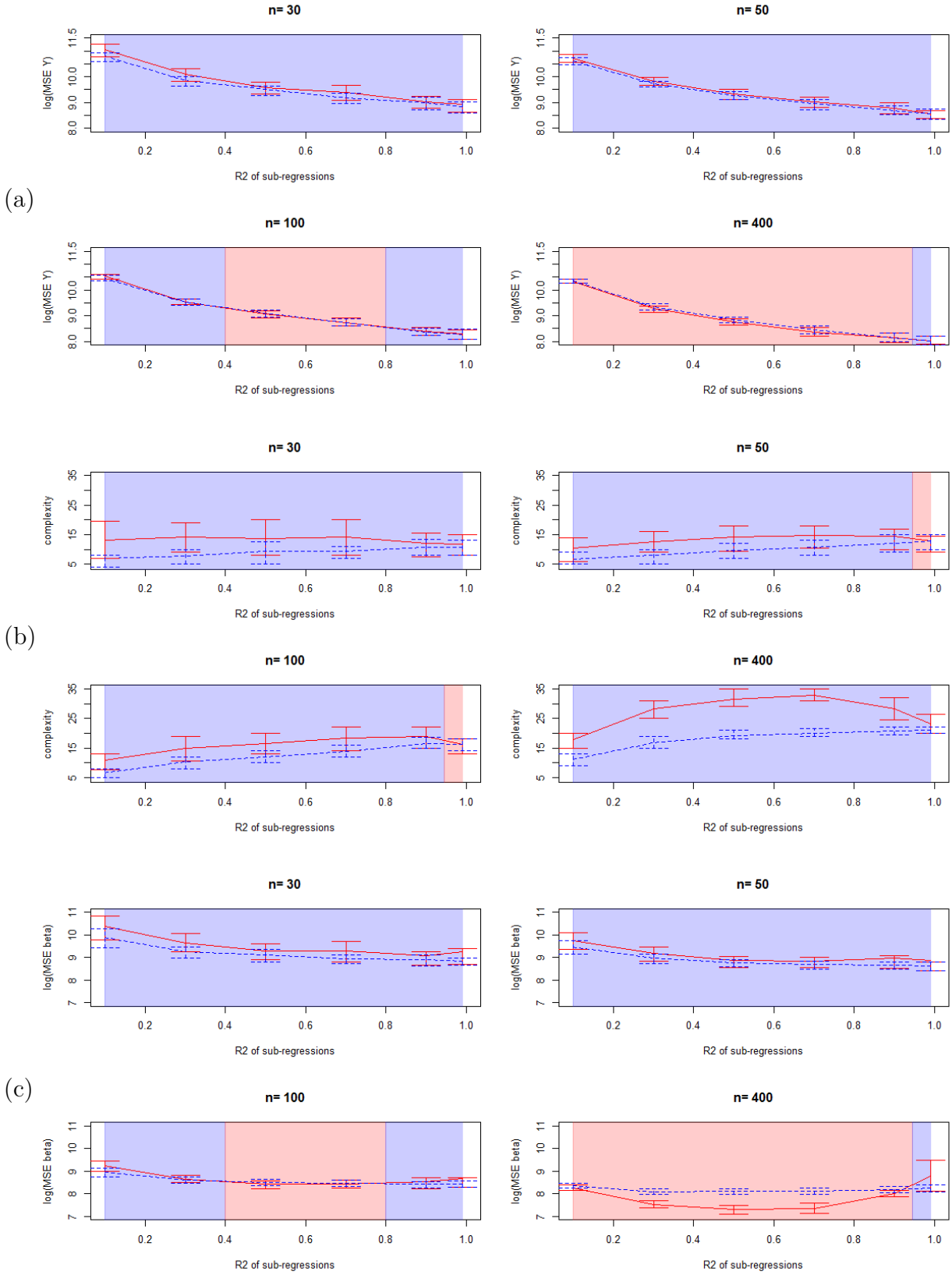


Figure 4.13: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model

Using Elasticnet estimation

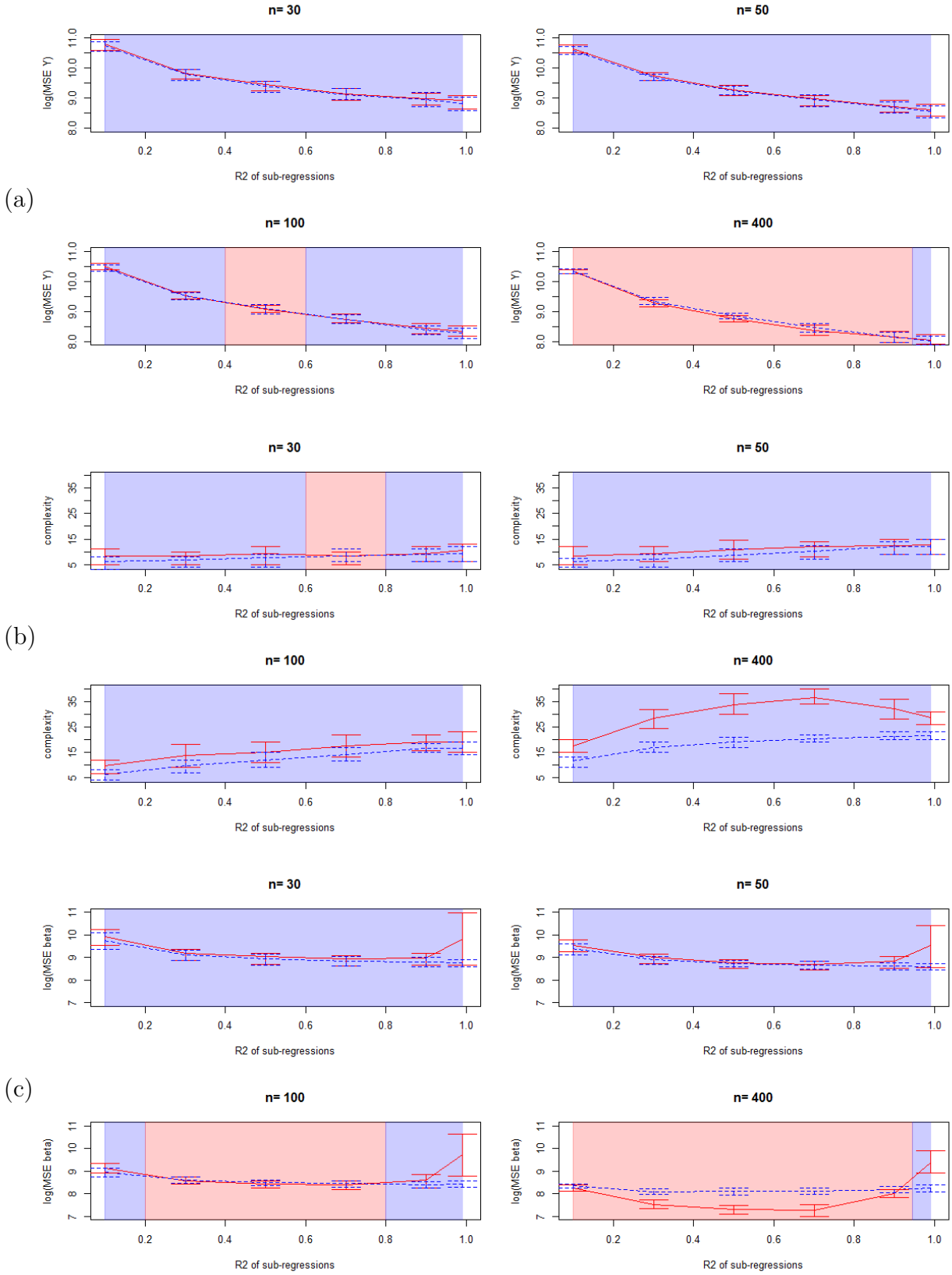


Figure 4.14: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model

Using Stepwise estimation

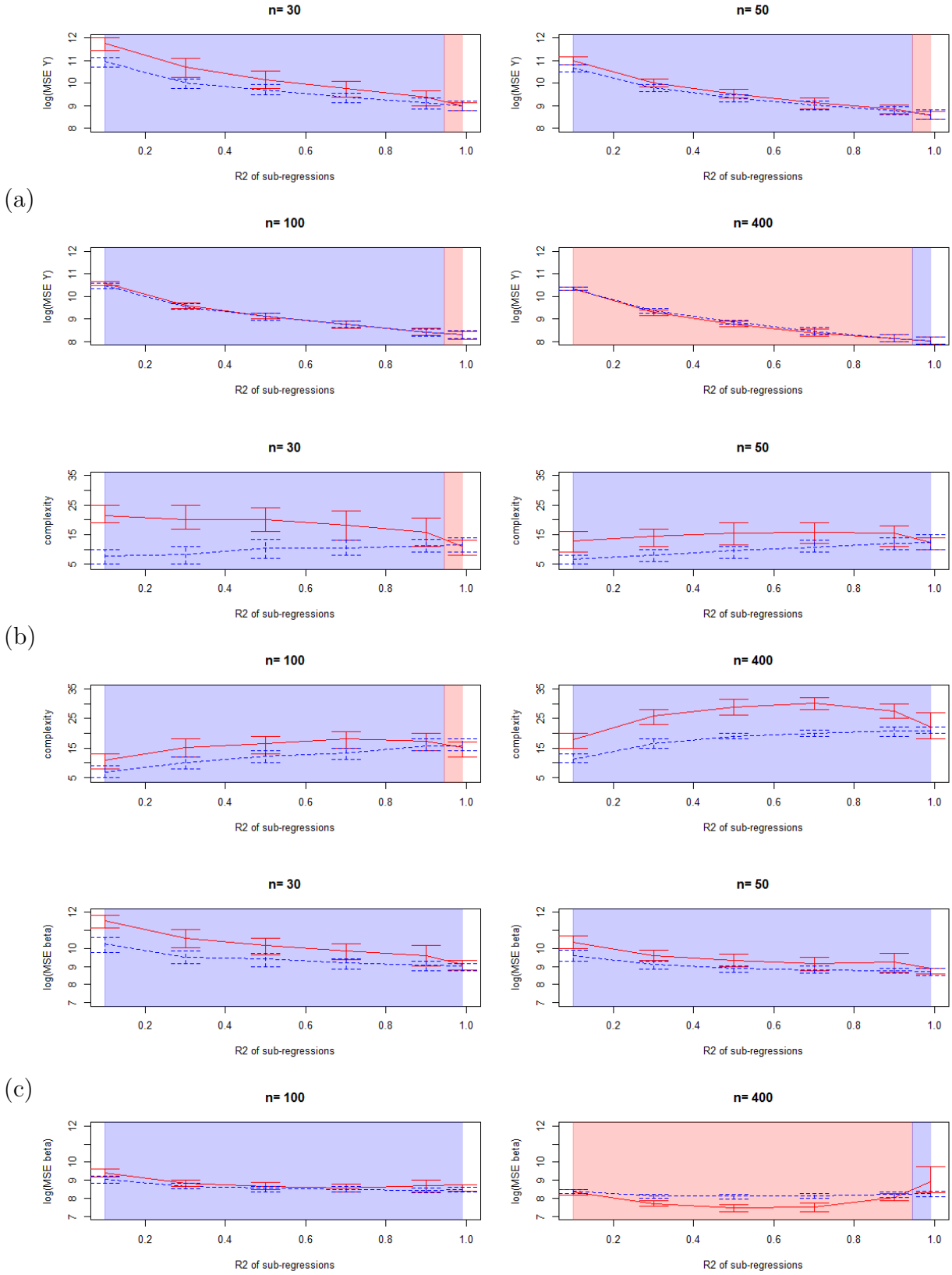


Figure 4.15: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model

Using Ridge estimation

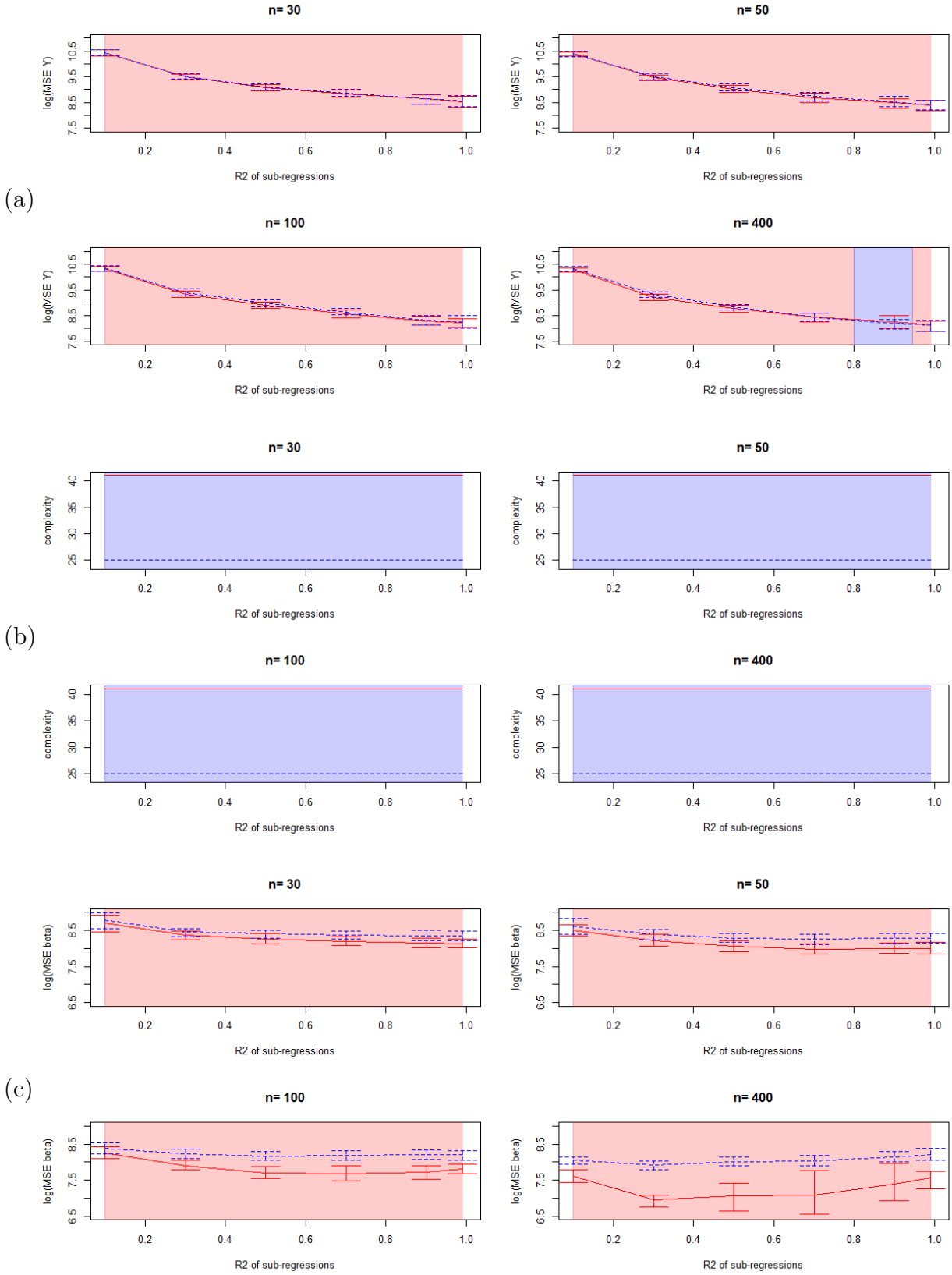


Figure 4.16: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model

4.9 Conclusion

Explicit modeling of the correlations is good for interpretation but also leads to the construction of the marginal model that is correlations-free and seem to significantly improve the efficiency of the model. These first results really are encouraging. Improvement in prediction is significant and we hope that it will be sufficient to obtain good results even when we will use $\hat{\boldsymbol{S}}$ instead of $\boldsymbol{S}$. Further results are provided with estimated $\hat{\boldsymbol{S}}$ (Chapter 6) and then with real industrial datasets (Chapter 7) in the following but we need first to obtain $\hat{\boldsymbol{S}}$ (Chapter 5).

Remark: The marginal model proposed in this chapter is conditional to $\boldsymbol{X}_f$ which is defined with $\boldsymbol{S}$. In this chapter we suppose $\boldsymbol{S}$ to be known but in real case we will have to estimate it and the estimator used $\hat{\boldsymbol{S}}$ will depend on whole $\boldsymbol{X}$ (see Chapter 5) so the marginal model will be in fact conditional to $\boldsymbol{X}$ but in a specific manner. It will be a sequential estimation: we will first estimate the structure of correlations $\hat{\boldsymbol{S}}$ based on $\boldsymbol{X}$ before estimating $\hat{\boldsymbol{Y}}$ conditionally to $\boldsymbol{X}_f$ and $\hat{\boldsymbol{S}}$.

Chapter 5

Estimation of the structure of sub-regression by MCMC

Abstract: In the previous chapters, $\boldsymbol{S}$ was supposed to be known, but in fact we have to find it. We define a generative model on the dataset, but also a probabilistic hypothesis on the structure that allows us to introduce a new criterion to evaluate the quality a structure of sub-regression. This new criterion takes into account the huge number of feasible structures. Then we have developed an MCMC algorithm to choose the best $\boldsymbol{S}$.

5.1 Choice of model: Brief state of the art

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 offers a method [Friedman et al., 2008] 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 5.4.3). But the resulting matrix is symmetric and we need an oriented structure for $\boldsymbol{S}$ to avoid cycles.

5.1.1 Cross validation

We want a model that would remain good for new individuals. 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 individuals 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.

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

5.1.2 Bayesian Information Criterion

Cross-validation depends on a criterion to choose a model. The Mean Squared Error is not the only criterion. Probabilistic criteria can also be used when we have an hypothesis on the distribution of the studied model. Such criteria can also be used without cross-validation. The Akaike Information Criterion [Akaike, 1974] known as AIC is asymptotically optimal in selecting the model with the least mean squared error (see [Stone, 1977]). The Bayesian Information Criterion [Lebarbier and Mary-Huard, 2006, Schwarz et al., 1978, Yang, 2005] 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 number of parameters to estimate (complexity is then penalized) and the number of individuals in the dataset. BIC is consistent so it asymptotically points out the true model when n grows. The Risk Inflation Criterion [Foster and George, 1994] (RIC) can also be used, or any other criterion [George and McCulloch, 1993] thought to be better in a given context. In this work we decided to start with the BIC that is given by:

$$\text{BIC} = -2\ln(L(\boldsymbol{\theta}|\boldsymbol{X})) + |\boldsymbol{\theta}|\ln(n) \tag{5.1}$$

where $L(\boldsymbol{\theta}|\boldsymbol{X}) = \mathbb{P}(\boldsymbol{X}|\boldsymbol{\theta})$ is the estimated likelihood of the parameters $\boldsymbol{\theta}$ given the dataset $\boldsymbol{X}$, $|\boldsymbol{\theta}|$ is the number of free parameters to estimate and n the number of individuals in the dataset. This choice comes from the popularity of BIC and from the fact that it makes a strong penalization on the complexity and we want to obtain a model that is easy to understand so parsimony is a real stake.

5.2 Revisiting the Bayesian approach for an over-penalized BIC

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

$$\mathbb{P}(\boldsymbol{S}|\boldsymbol{X}) \quad \propto \quad \mathbb{P}(\boldsymbol{X}|\boldsymbol{S})\mathbb{P}(\boldsymbol{S}) = \mathbb{P}(\boldsymbol{X}_r|\boldsymbol{X}_f, \boldsymbol{S})\mathbb{P}(\boldsymbol{X}_f|\boldsymbol{S})\mathbb{P}(\boldsymbol{S}) \tag{5.2}$$

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

5.2.1 Probability associated to the redundant covariates (responses)

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

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

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 $\boldsymbol{\psi}$.

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

product of independent Gaussians.

Estimation of the numerical parameters: $\boldsymbol{S}$ is a discrete parameter but it is associated to numerical parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\varepsilon}$ through Hypothesis 1. We can estimate the coefficients of sub-regression by OLS for example because sub-regression are small, imply independent covariates and do not need variable selection (parsimony comes directly from $\boldsymbol{S}$):

$$\forall j \in \{1, \ldots, d_r\}, \hat{\boldsymbol{\alpha}}_j = ((\boldsymbol{X}^{J_p^j})' \boldsymbol{X}^{J_p^j})^{-1} (\boldsymbol{X}^{J_p^j})' \boldsymbol{X}^{J_r^j}. \tag{5.3}$$

And then we get the estimation of $\boldsymbol{\varepsilon}$ relying on $\hat{\boldsymbol{\alpha}}^*$:

$$\hat{\boldsymbol{\varepsilon}} = \boldsymbol{X}_r - \boldsymbol{X}_f \hat{\boldsymbol{\alpha}}^*. \tag{5.4}$$

These estimators will be used in Chapter 8.

5.2.2 Probability associated to the free covariates (predictors)

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

Hypothesis 4 *All covariates $\boldsymbol{X}^j$ with $j \in J_f$ are mutually independent and arise from the following Gaussian mixture of K_j components*

$$\forall 1 \leq i \leq n, \mathbb{P}(x_{i,j}|\boldsymbol{S}; \boldsymbol{\pi}_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \sum_{h=1}^{K_j} \pi_{j,h} \Phi(x_{i,j}; \mu_{j,h}, \Sigma_{j,h}),$$

where $\boldsymbol{\pi}_j = (\pi_{j,1}, \ldots, \pi_{j,K_j})$ is the vector of mixing proportions with $\forall 1 \leq h \leq k_j, \pi_{j,h} > 0$ and $\sum_{h=1}^{K_j} \pi_{j,h} = 1$, $\boldsymbol{\mu}_j = (\mu_{j,1}, \ldots, \mu_{j,K_j})$ is the vector of centres and $\boldsymbol{\Sigma}_j = (\Sigma_{j,1}, \ldots, \Sigma_{j,K_j})$ is the vector of variances and Φ is the Gaussian density function.
We stack together all these mixture parameters in $\boldsymbol{\theta} = (\boldsymbol{\pi}_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j; j \in J_f)$. We now have a full generative model on $\boldsymbol{X}$.

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

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

5.2.3 Probability associated to the discrete parameter $\boldsymbol{S}$

Defining $\mathbb{P}(\boldsymbol{S})$ The most standard choice consists of putting a uniform distribution on the model space $\mathcal{S}_d$, this choice being noted $\mathbb{P}_U(\boldsymbol{S}) = |\mathcal{S}_d|^{-1}$, with $|\mathcal{S}_d|$ the space dimension of $\mathcal{S}_d$ as defined in section 4.6.

Remarks

- Hypothesis 4 is the keystone to define a full generative model on the whole covariates $\boldsymbol{X}$. 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 $\boldsymbol{X}^j$ ($j = 1, \ldots, d$). 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.

5.2.4 Penalization of the integrated likelihood by $\mathbb{P}(\boldsymbol{S})$

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

$$\begin{aligned}
\mathbb{P}_H(\boldsymbol{S}) &= \mathbb{P}_H(\boldsymbol{J}_r, \boldsymbol{J}_p) \\
&= \mathbb{P}_H(\boldsymbol{J}_r, \boldsymbol{J}_p, d_r, \boldsymbol{d}_p) \\
&= \mathbb{P}_U(\boldsymbol{J}_p|\boldsymbol{d}_p, \boldsymbol{J}_r, d_r) \times \mathbb{P}_U(\boldsymbol{d}_p|\boldsymbol{J}_r, d_r) \times \mathbb{P}_U(\boldsymbol{J}_r|d_r) \times \mathbb{P}_U(d_r) \\
&= \left[\prod_{j=1}^{d_r} \binom{d - d_r}{d_p^j}\right]^{-1} \times [d - d_r]^{-d_r} \times \left[\binom{d}{d_r}\right]^{-1} \times [d+1]^{-1},
\end{aligned}$$

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. $\mathbb{P}_H(\boldsymbol{S})$ 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 $d_p^j < d/2$ ($j = 1, \ldots, d_r$).*

These two thresholds are sufficiently large to be wholly realistic.

Final approximation of $\mathbb{P}(\boldsymbol{S}|\boldsymbol{X})$ 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 $\mathbb{P}_U(\boldsymbol{S})$ or $\mathbb{P}_H(\boldsymbol{S})$ respectively:

$$\begin{aligned}
\text{BIC}_U(\boldsymbol{S}) &= \text{BIC}_r(\boldsymbol{S}) + \text{BIC}_f(\boldsymbol{S}) - 2\ln \mathbb{P}_U(\boldsymbol{S}) \\
\text{BIC}_H(\boldsymbol{S}) &= \text{BIC}_r(\boldsymbol{S}) + \text{BIC}_f(\boldsymbol{S}) - 2\ln \mathbb{P}_H(\boldsymbol{S}).
\end{aligned}$$

In the following, we will denote by BIC_* any of both BIC_U and BIC_H. Numerical results in Section 6.2 will allow to compare behaviour of both criteria.

Remarks

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

5.3 Random walk to optimize the criterion

We have to choose a model from a finite space of models ($\boldsymbol{S}$ is a discrete parameter), with a criterion to compare them. We choose to use a random walk (Markov Chain [Robert and Casella, 2005])with some properties:

- The walk has to be able to go from any structure to any other.
- Each step has rely on the chosen quality criterion for the structure.

Then we have to define neighbourhoods and probability transitions that verify these two properties.

5.3.1 Transition probabilities

Once we have neighbourhoods $\mathcal{V}(\boldsymbol{S})$ that allow to go from any structure to any other, we have to choose a candidate for the next step. The walk follows a time-homogeneous Markov Chain whose transition matrix has $|\mathcal{S}_d|$ rows and columns (extremely wide and sparse matrix for $d > 10$ so we just compute the probabilities when we need them). At each step the Markov chain moves with probability:

$$\forall \boldsymbol{S} \in \mathcal{S}_d, \forall \boldsymbol{S}^+ \in \mathcal{V}(\boldsymbol{S}) : \mathbb{P}(\boldsymbol{S}^+ | \mathcal{V}(\boldsymbol{S})) \quad = \quad \frac{\exp\left(-\text{BIC}_*(\boldsymbol{S}^+)\right)}{\sum_{\tilde{\boldsymbol{S}} \in \mathcal{V}(\boldsymbol{S})} \exp\left(-\text{BIC}_*(\tilde{\boldsymbol{S}})\right)} \tag{5.5}$$

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 $\boldsymbol{S}$ and $\mathcal{V}(\boldsymbol{S})$ with stationary distribution $\pi \propto \exp(-\text{BIC}_*(\cdot))$ on the space $\mathcal{S}_d$.

The output of the algorithm will be the best structure in terms of $P(\boldsymbol{S}|\boldsymbol{X})$ which weights each candidate.

5.3.2 Deterministic neighbourhood

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

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

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

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

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

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

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

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

$$\mathcal{V}_{p-}(\boldsymbol{S}) = \Big\{\tilde{\boldsymbol{S}} : \tilde{\boldsymbol{S}} \in \mathcal{S}_d, \qquad \tilde{\boldsymbol{J}}_r = \boldsymbol{J}_r, \tilde{J}_p^{\{1,\dots,d_r\}\setminus\{j\}} = J_p^{\{1,\dots,d_r\}\setminus\{j\}}, \\ \tilde{J}_p^j = J_p^j\setminus\{h\}, j \in \{1,\dots,d_r\}, h \in J_p^j\Big\}.$$

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

5.3.3 Stochastic neighbourhood

We see that $|\mathcal{V}(\boldsymbol{S})|$ can be very large. Large neighbourhoods are bad in terms of computational cost but also in terms of efficiency: the denominator in equation (5.5) increases with the number of candidates. A classical way to avoid such a problem is to reduce the neighbourhood to a random subset and then to allow stationarity (to keep the walk's properties). We redefine a neighbourhood based on the modification of the adjacency matrix $\boldsymbol{G}$ associated to $\boldsymbol{S}$ instead of using the four previous neighbourhoods.

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

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

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 the directed edge of the graph to modify (add or remove). And we have for $\tilde{\boldsymbol{S}} = \boldsymbol{S}^{(i,j)}$:

$$\begin{aligned} \forall (k,l) \neq (i,j), \quad \tilde{\boldsymbol{G}}_{k,l} &= \boldsymbol{G}_{k,l} \\ \tilde{\boldsymbol{G}}_{i,j} &= 1 - \boldsymbol{G}_{i,j} \end{aligned}$$

where $\tilde{\boldsymbol{G}}$ is the adjacency matrix associated to $\tilde{\boldsymbol{S}}$. 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-regressions. 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.

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

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

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. For large values of d, the number of candidate at each step can become critical. Each candidate requires to re-estimate the $\boldsymbol{\alpha}^j$'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. In such cases one solution (almost as a warm-up phase) would be to only consider a random part of the neighbourhood to reduce the computational cost of each step.

5.3.4 Active relaxation of constraints

In practice, for some of the $(i,j) \in \mathcal{A}_{(q)}$, we have $\boldsymbol{S}^{(i,j)} \notin \mathcal{S}_d$ 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 can reach any feasible structure from any other feasible structure, local extrema may significantly slow down the research of the optimal structure. Thus we propose a constraints relaxation method by a new definition of $\tilde{\boldsymbol{S}} = \boldsymbol{S}^{(i,j)}$ relying on a new definition of $\tilde{\boldsymbol{G}}$:

Modification of the selected directed edge (i,j) on the graph: (Figure 5.2)

$$\begin{aligned} & \tilde{\boldsymbol{G}}_{i,j} &=& \ 1 - \boldsymbol{G}_{i,j} \text{ as usual and} \\ \forall k \neq i, l \neq j, \quad & \tilde{\boldsymbol{G}}_{k,l} &=& \ \boldsymbol{G}_{k,l} \end{aligned}$$

Column-wise relaxation : newly predictive covariate cannot be regressed anymore (Figure 5.3):

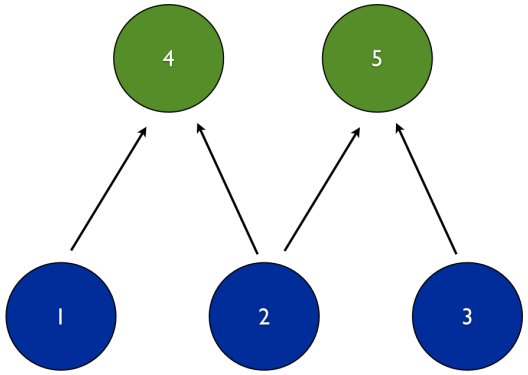
$$\forall k \in \{1,\dots,d\} \setminus \{i\}, \tilde{\boldsymbol{G}}_{k,j} \ = \ \boldsymbol{G}_{i,j}\boldsymbol{G}_{k,j} \tag{5.6}$$

Row-wise relaxation: newly regressed covariate cannot be predictive anymore (Figure 5.4):

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

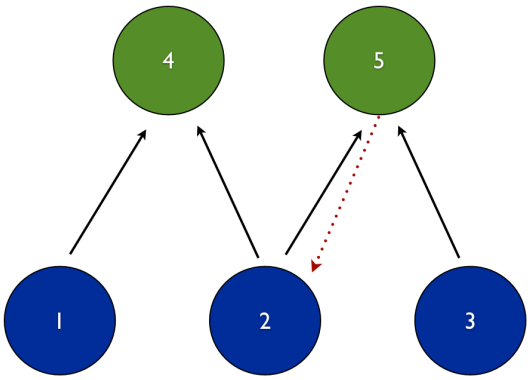
It can be seen as forcing the modification by removing what would have made the structure not feasible. So in one step we can test a model that remove completely a sub-regression, remove the explicative role of a covariate in all sub-regressions (that was not possible with the deterministic neighbourhood) and create a new pairwise sub-regression. It drastically increases the scope of the neighbourhood (Figure 5.6) 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 5.10), 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).



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

Figure 5.1: We start from a structure $\boldsymbol{S} = ((4,5),(\{1,2\},\{2,3\}))$ and its associated matrix G

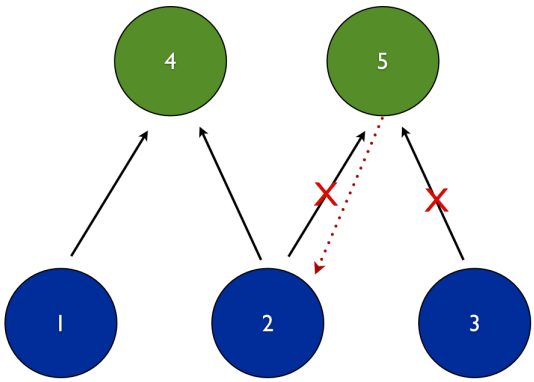


$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 \end{pmatrix}$$

Figure 5.2: We want to define the candidate $\tilde{\boldsymbol{S}} = \boldsymbol{S}^{(5,2)}$ and its associated matrix but the structure obtained would not be feasible (breaking the uncrossing rule).

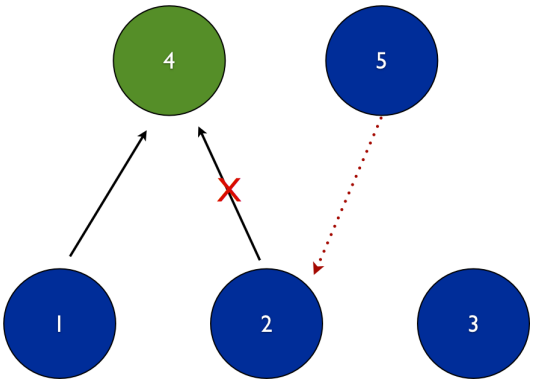
Numerical results (Section 4) illustrates the efficiency of the walk when the true model contains structures with various strength (section 6.2) and an example with a non-linear structure (Figure 6.20(a)).

We give an example to better illustrate how it works in Figures 5.1 to 5.6:



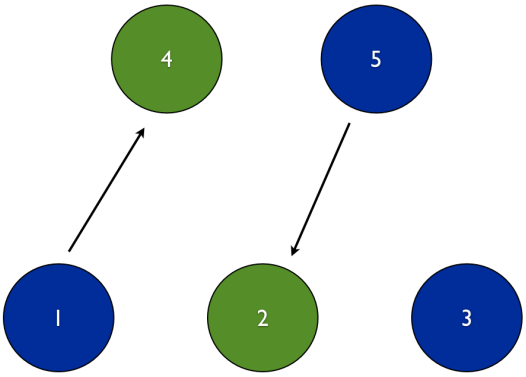
$$\begin{pmatrix} 0 & 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 0 & 0 & \mathbf{0} \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{0} \end{pmatrix}$$

Figure 5.3: Column-wise relaxation: newly predictive covariate cannot be regressed anymore.



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

Figure 5.4: Row-wise relaxation: newly predictive covariate cannot be regressed anymore.



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

Figure 5.5: We get a feasible candidate $\tilde{\boldsymbol{S}} = ((2,4),(\{5\},\{1\}))$ that does differ from $\boldsymbol{S}$ in many points.

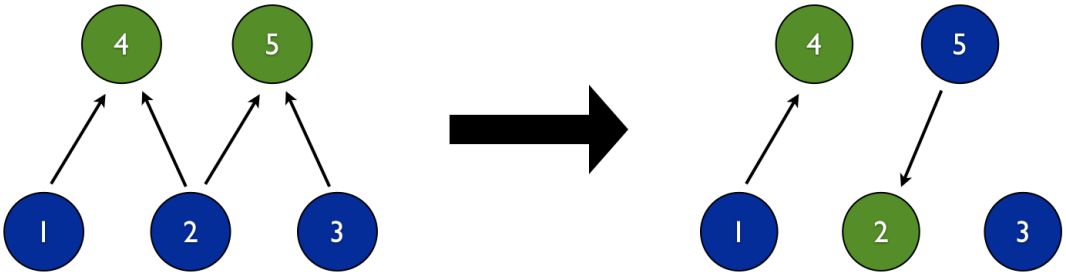


Figure 5.6: All this modifications are made in only one step in the MCMC, meaning an increased scope for the neighbourhoods.

5.4 Initialization

5.4.1 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 $\boldsymbol{G}$, 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 5.7 and 5.8. 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.

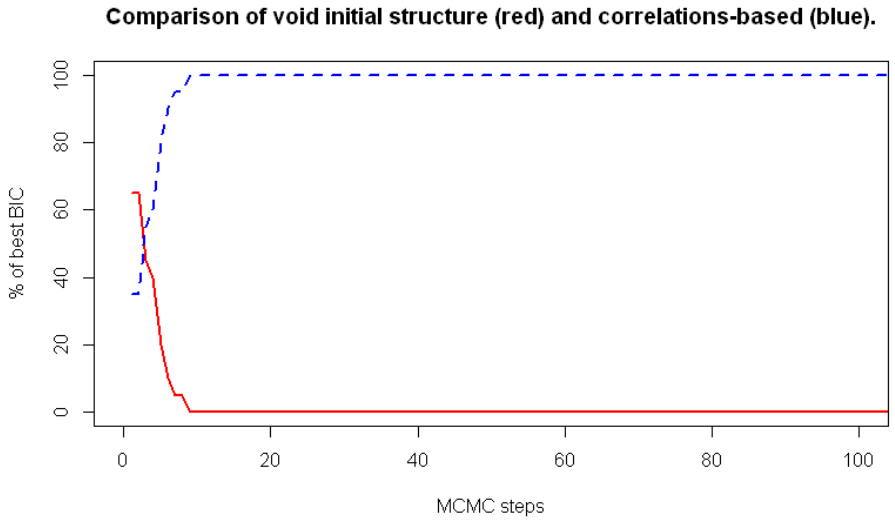


Figure 5.7: Amount of time each method is better for the 100 first steps of the MCMC.

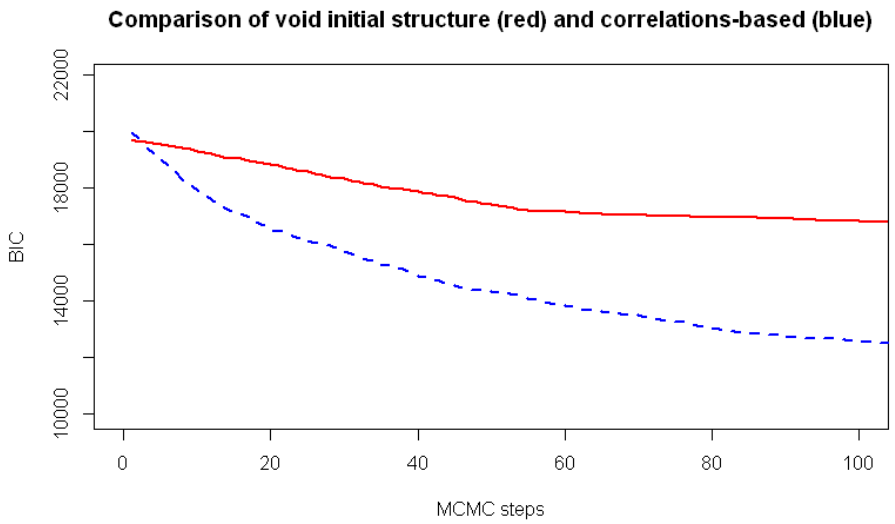


Figure 5.8: Evolution of the BIC (criterion to minimize in the MCMC) for each method.

5.4.2 Multiple intialization

Local extrema are a known issue for most of optimization methods, and one would rather test multiple short chains than lose time in initialisation or long chains [Gilks et al., 1996]. We also compare the results obtained with several number of chains. Figure 5.9 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 (then curves get lower as the number of initializations does increase, by construction). 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.

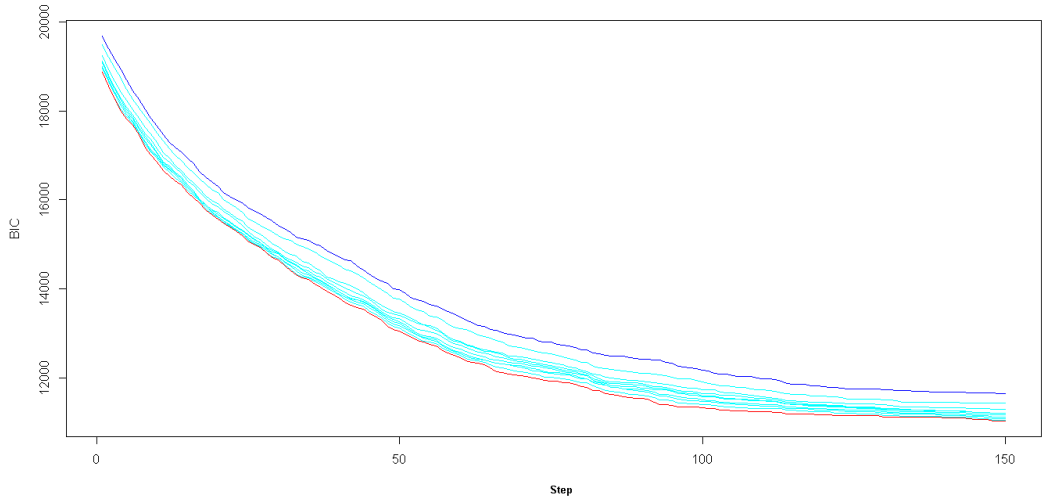


Figure 5.9: Comparison of distinct number of correlation-based initialisations for the MCMC. Dark blue=1 intialization, red=10 initializations, the cyan curves represent all intermediate values for the number of initializations (from 2 to 9).

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

5.4.3 Graphical LASSO

Graphical LASSO ([Friedman et al., 2008], [Witten et al., 2011], [Tibshirani et al.,] and [Friedman et al., 2010]) is set to give undirected (thus symmetric) graphs by selection in the precision matrix (the inverse of the variance-covariance matrix). It cycles through the variables, fitting a modified LASSO regression to each variable in turn. The individual LASSO problems are solved by coordinate descent. It is a variant of another method ([Meinshausen and Bühlmann, 2006]) that computes d LASSO estimation (each covariate regressed by all the others) and put zeros in the precision matrix when covariates seem to be conditionally independent (zero as regression coefficient). It does make sense for exponential family because in these cases, zeros in the precision matrix $\mathbf{\Sigma}^{-1}$ can be interpreted in terms of conditional independence between covariates [Dempster, 1972]. But we have supposed Gaussian mixture on $\boldsymbol{X}$ (not exponential family so the precision matrix cannot be interpreted as easily) and we search an oriented graph. So it is not adapted to find the structure of sub-regressions.

However, we could still try use it for initialization, for example by a Hadamard product with $\boldsymbol{G}^{(0)}$ the adjacency matrix of the initial structure. Another idea would be to make the Hadamard product with the correlations matrix before computing 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.

5.5 Pruning

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

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:

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

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. So the criterion is the same, only the neighbourhood is changed. 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`.

But if n is small, then BIC might suffer from overfitting. "Trust, but verify" says a Russian proverb. So we propose some other pruning methods (once again, they only potentially reduce complexity of the structure).

Variable selection

We can use variable selection methods like the LASSO on $\boldsymbol{X}^{J_p^j}$ to estimate the coefficients $\boldsymbol{\alpha}_j$ and obtain some supplementary zeros. Working on $\boldsymbol{X}^{J_p^j}$ 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-regressions that offer a F-statistic under a minimal value.

5.6 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 $\boldsymbol{S}$ as the number of elements in the adjacency matrix, that is the number of links between covariates and is obtained by:

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

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 5.10 and 5.11 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.

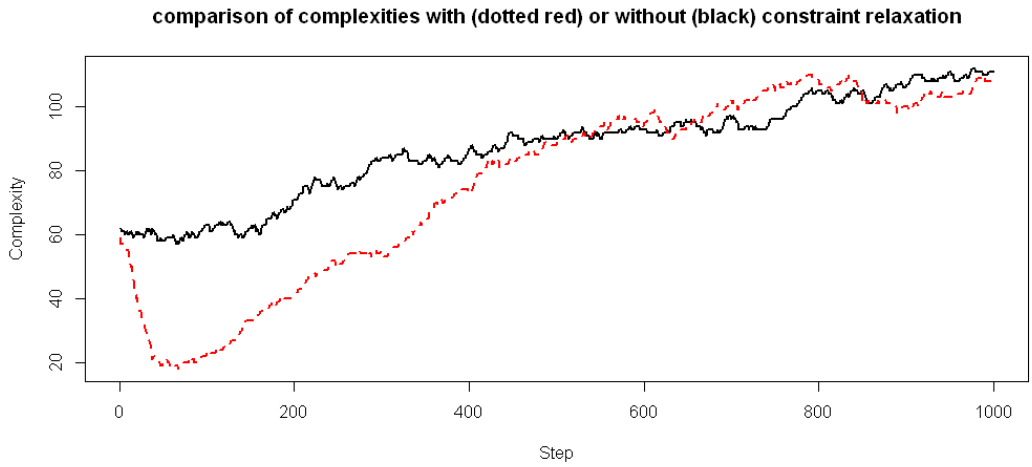


Figure 5.10: Comparison of complexity evolution with or without constraint relaxation.

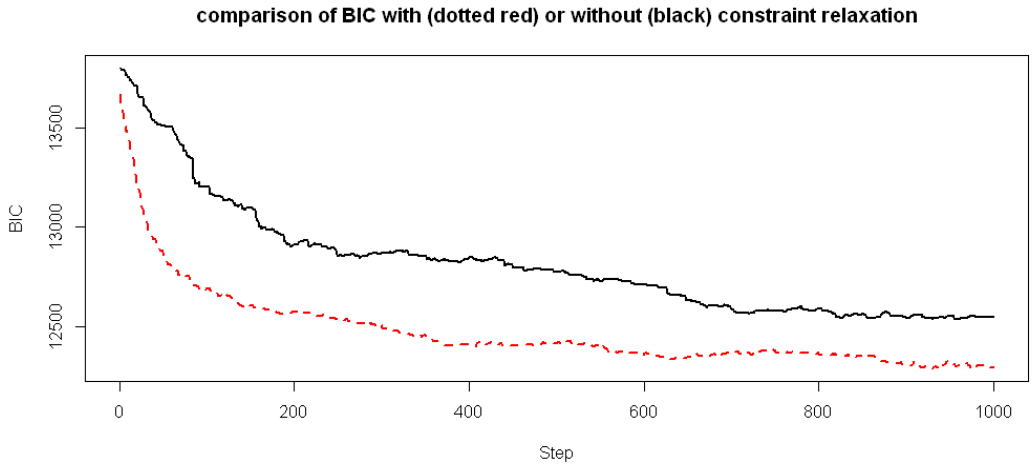


Figure 5.11: Comparison of BIC evolution with or without constraint relaxation.

5.6.1 Some indicators for proximity

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

- $T_r = |J_r \cap \hat{J}_r|$ ("True Responses"): it corresponds to the number of estimate response covariates in $\hat{\boldsymbol{S}}$ which are *truly* response covariates in the true model $\boldsymbol{S}$.
- $W_r = |\hat{J}_r| - T_r$ ("Wrong Responses"): it corresponds to the number of estimate response covariates in $\hat{\boldsymbol{S}}$ which are are *wrongfully* response covariates in the true model $\boldsymbol{S}$.
- $M_r = d_r - T_r$ ("Missing Responses"): 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 $\hat{\boldsymbol{S}}$ 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 models.

5.7 Conclusion

We now have an algorithm and a criterion to find the best structure of correlations. This algorithm is extremely flexible: no heavy hypotheses on the dataset, no crucial parameter to tune by hand. We want then to know how efficient it is on simulated datasets first (Chapter 6), and then on real industrial datasets from ArcelorMittal to confirm the utility of both the new model and the algorithm in the real life (Chapter 7).

Remark: Choosing whether we have to use or not the structure to make a pre-treatment 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. Our explicit structure describes in details the complexity of the situation so we can then act knowing what we do.

Chapter 6

Numerical results on simulated datasets

Abstract: Here are the numerical results obtained for an unknown structure $\boldsymbol{S}$ on simulated datasets. It illustrates efficiency of the MCMC algorithm used to find a relevant structure between the covariates. The fact that the structure is unknown makes the simulations more similar to a real case study where we do not even know if there is a linear structure of correlations between the covariates.

6.1 Simulated datasets

Now we have defined the model and the way to obtain it, we can have a look on some numerical results to evaluate the proposed strategy to find the structure and the efficiency of the resulting marginal model. The package `CorReg` has been tested on the simulated datasets from section 4.8.1. Section 6.2 shows the results obtained in terms of $\hat{\boldsymbol{S}}$. Section 6.3 shows the results obtained using only `CorReg`, or `CorReg` combined with other methods. The graph in section 6.3 give both mean, first and third quartiles of the chosen indicator. The MSE on $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{Y}}$ were computed on a validation sample of 1000 individuals. Several pattern for $\boldsymbol{Y}$ were tested to evaluate the impact of irrelevant covariates.

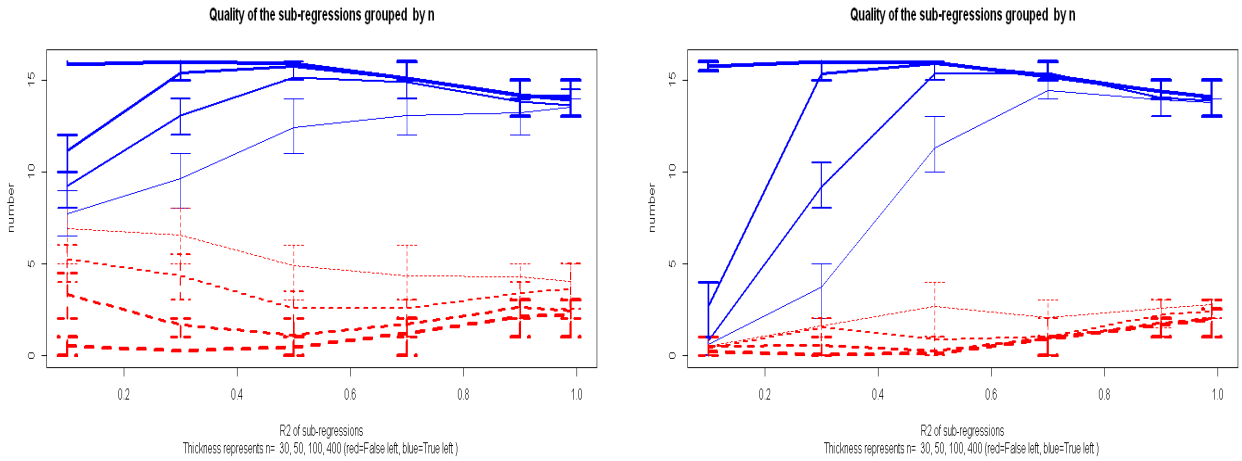
We used the package `Rmixmod` from CRAN [Auder et al., 2014] to estimate the densities of each covariate. For each configuration, the MCMC walk was launched on 10 initial structures with a maximum of 1 000 steps each time. When $n < d$, a frequently used method is the Moore-Penrose generalized inverse [Katsikis and Pappas, 2008], thus OLS can obtain some results even with $n < d$. We compare different methods with and without `CorReg` as a pre-treatment. All the results are provided by the `CorReg` package. Associated figures will display both mean and inter-quartile intervals.

6.2 Results on $\hat{S}$

Figure 6.1 illustrates the impact of large samples. For $n >> d$ 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 is 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 6.3.
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 `Rmixmod`, the gap between the marginal and dependent likelihood might be small and thus the walk can be slowed whereas we use a finite number of steps.



(a) With classical BIC_U criterion

(b) With our BIC_H criterion

Figure 6.1: Quality of the sub-regressions 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.

6.2.1 Comparison with `Selvarclust`

Maugis provides results[1] of `SelvarClust` on a dataset *Data2.txt* containing 2 000 data points from a mixture of four Gaussian distributions $\mathcal{N}(\mu_k, \boldsymbol{I}_2)$ with $\mu_1 = (-2, -2), \mu_2 = (-2, 2), \mu_3 = (2, -2), \mu_4 = (2, 2)$ and with a proportion vector $\boldsymbol{\pi} = (0.3, 0.2, 0.3, 0.2)$.

[1] `http://www.math.univ-toulouse.fr/~maugis/SelvarClustHomepage.html`

Variables $\boldsymbol{X}^1, \boldsymbol{X}^2$ corresponds to the coordinates. Eight irrelevant variables (for clustering) are appended, simulated according to $\forall 1 \leq i \leq 2\,000$:

$$\boldsymbol{X}_i^{\{3,\dots,6\}} | \boldsymbol{X}_i^{\{1,2\}}; \boldsymbol{\alpha}^*, \boldsymbol{\Omega} \sim \mathcal{N}(\boldsymbol{X}_i^{\{1,2\}} \boldsymbol{\alpha}^*, \boldsymbol{\Omega})$$

$$\text{with } \boldsymbol{\alpha}^* = \begin{pmatrix} 0.5 & 0 & 2 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix} \text{ and } \boldsymbol{\Omega} = \text{diag}(1, 1, 0.5, 0.5)$$

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

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 $\boldsymbol{X}^0 = \mathbf{1}$ 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 $\hat{\boldsymbol{\alpha}}^*_{Selvarclust} =$

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

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

We then compare the two methods on theoretical aspects.

- **Distinct goals:** `CorReg` gives better results because it does not suffer from correlations like stepwise does in the algorithm from Maugis but also because the goal is not the same. `Selvarclust` aims to find the relevant covariates for clustering 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. To obtain better results is just a necessary confirmation of that.

- **Distinct hypotheses:**

  - `Selvarclust` allows dependencies between the regressors and also between the noises of the sub-regressions. We suppose independence between the regressors and between the conditional distributions (noises of the sub-regressions are supposed to be independent). Our algorithm does not allow to find complex distributions with dependencies as `Selvarclust` does.
  - We estimate each marginal distribution separately (with `Rmixmod` for example). It can lead to a joint distribution with a huge number of components (see Chapter 9) whereas `Selvarclust` has to choose directly the number of components of the joint distribution and then this number is limited.

Both models use a sub-regression structure and propose an algorithm to find it, but the comparison stops here. `CorReg` does not really beat `Selvarclust` because hypotheses and objectives are not the same, so the result cannot really be compared. Our algorithm will allow more components than `Selvarclust` and `Selvarclust` will allow more dependencies between the covariates.
`Selvarclust` is not a competitor to `CorReg` but a confirmation that the concept of sub-regressions within the covariates and the concept of redundant and irrelevant covariates are pertinent.

6.2.2 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 $(\boldsymbol{X}^{J_p^j})'\boldsymbol{X}^{J_p^j}$ matrices.
Figure 6.2 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 $n = 400$.

The main impact of a change of d will be the expansion of $\mathcal{S}_d$ that will require more

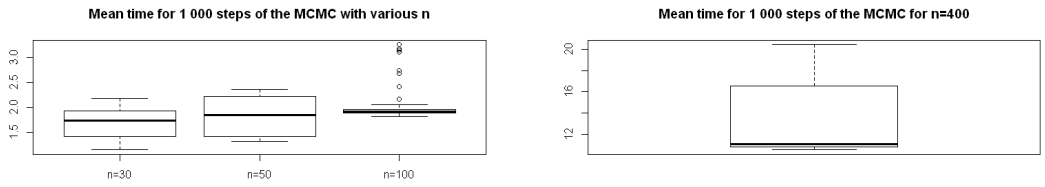


Figure 6.2: Evolution of the mean time for 10 times 1 000 steps of the MCMC, 39 candidates each time (constraint relaxation). $d = 40$ covariates with $d_r = 16$ sub-regressions

iteration/candidates to find the true structure. But the cost of each sub-regression remains the same. Figure 6.3 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 $\boldsymbol{G}$, 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.

Without constraint relaxation (Figure 6.4) 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.

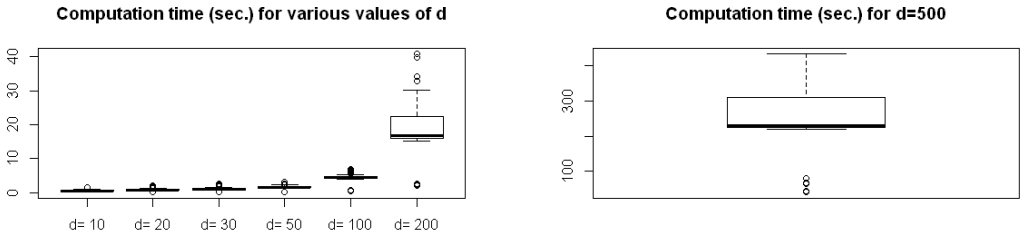


Figure 6.3: Evolution of the mean time for 10 times 1 000 steps of the MCMC, 40 candidates each time (constraint relaxation). $d_r = 0.4 \times d$ sub-regressions, $n = 30$.

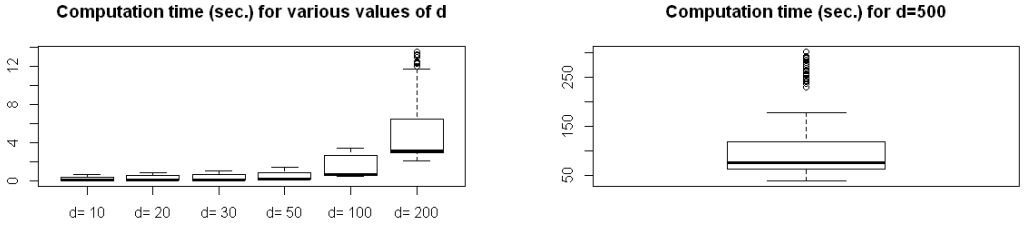


Figure 6.4: Evolution of the mean time for 10 times 1 000 steps of the MCMC, 40 candidates each time (without constraint relaxation). $d_r = 0.4 \times d$ sub-regressions, $n = 30$.

6.3 Results on prediction

6.3.1 $\boldsymbol{Y}$ depends on all variables in $\boldsymbol{X}$

We try the method with a response variable depending on all covariates to compare the results with those from section 4.8.1 (same $\boldsymbol{X}$ and $\boldsymbol{Y}$). (`CorReg` reduces the dimension and cannot 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 $\hat{\boldsymbol{S}}$ are still good so the MCMC is efficient enough to be useful for the study of the response variable $\boldsymbol{Y}$. Results are mostly the same than when using the true structure.

Results for OLS (Figure 6.5) are similar to those from section 4.8.1 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 6.6 to 6.8) where covariates not deleted by the structure are deleted by the variable selection.
Ridge regression results (Figure 6.9) are also very similar to the previous (Figure 4.16).

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.

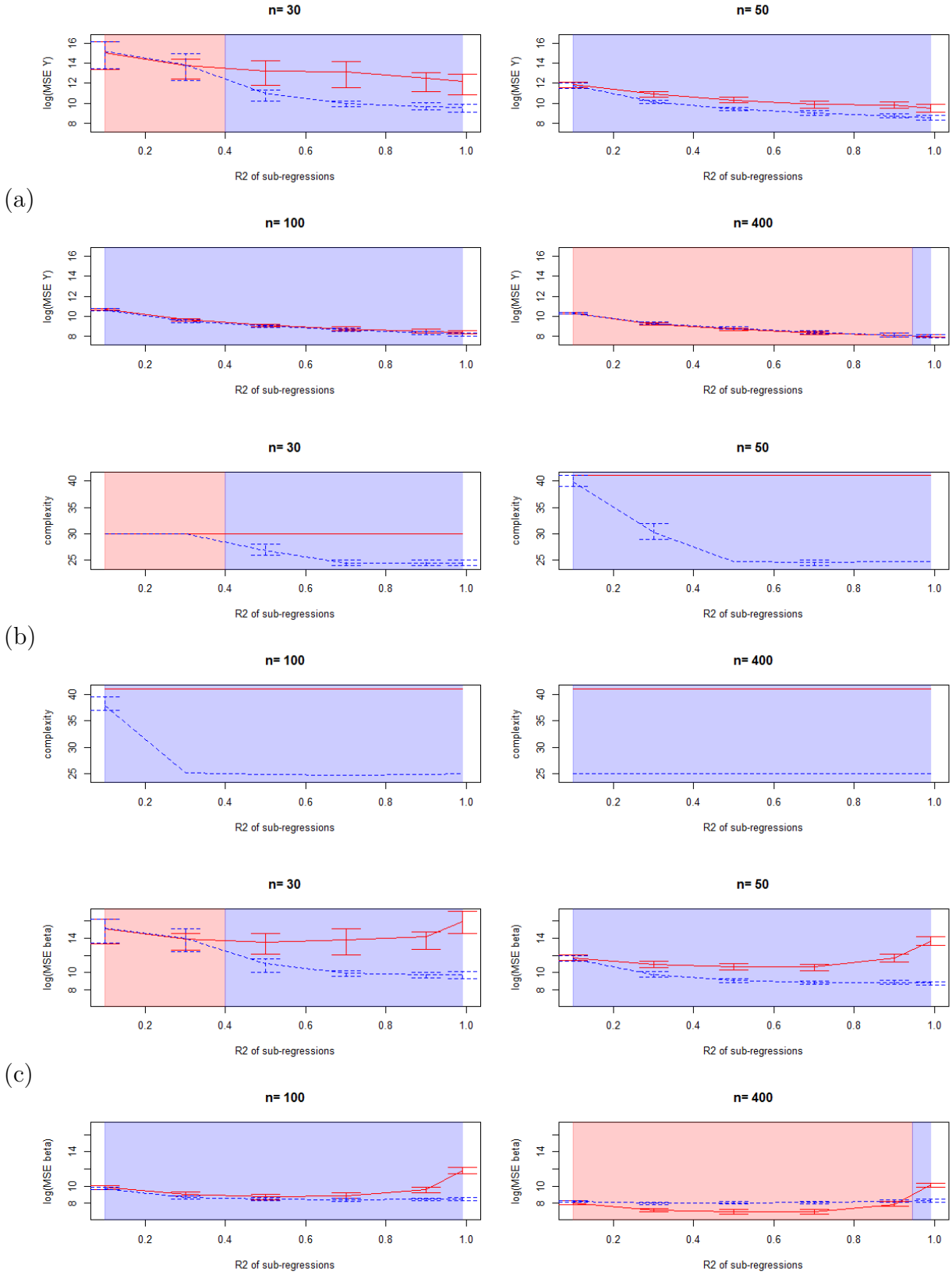


Figure 6.5: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

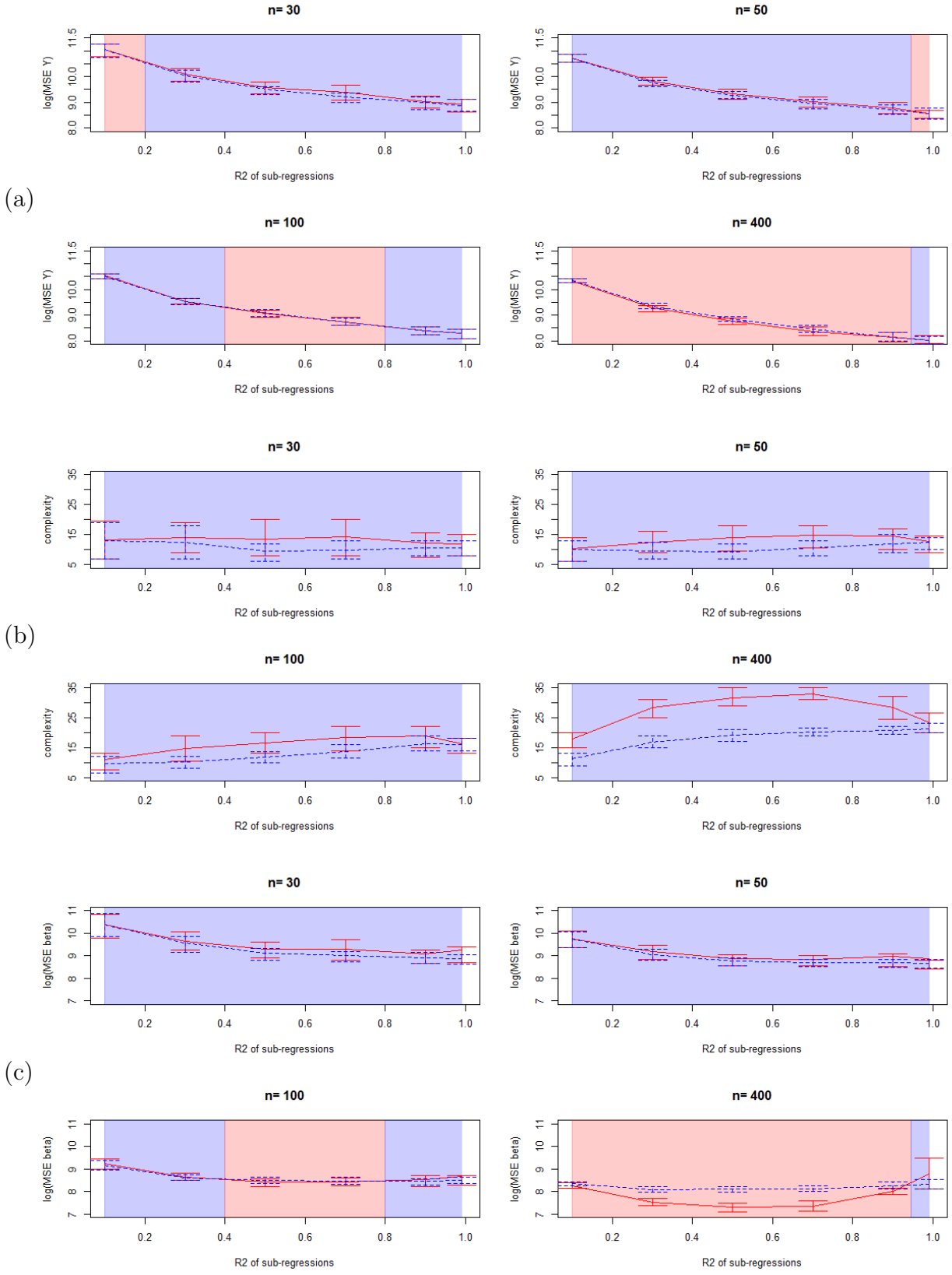


Figure 6.6: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when Y depends on all variables in X

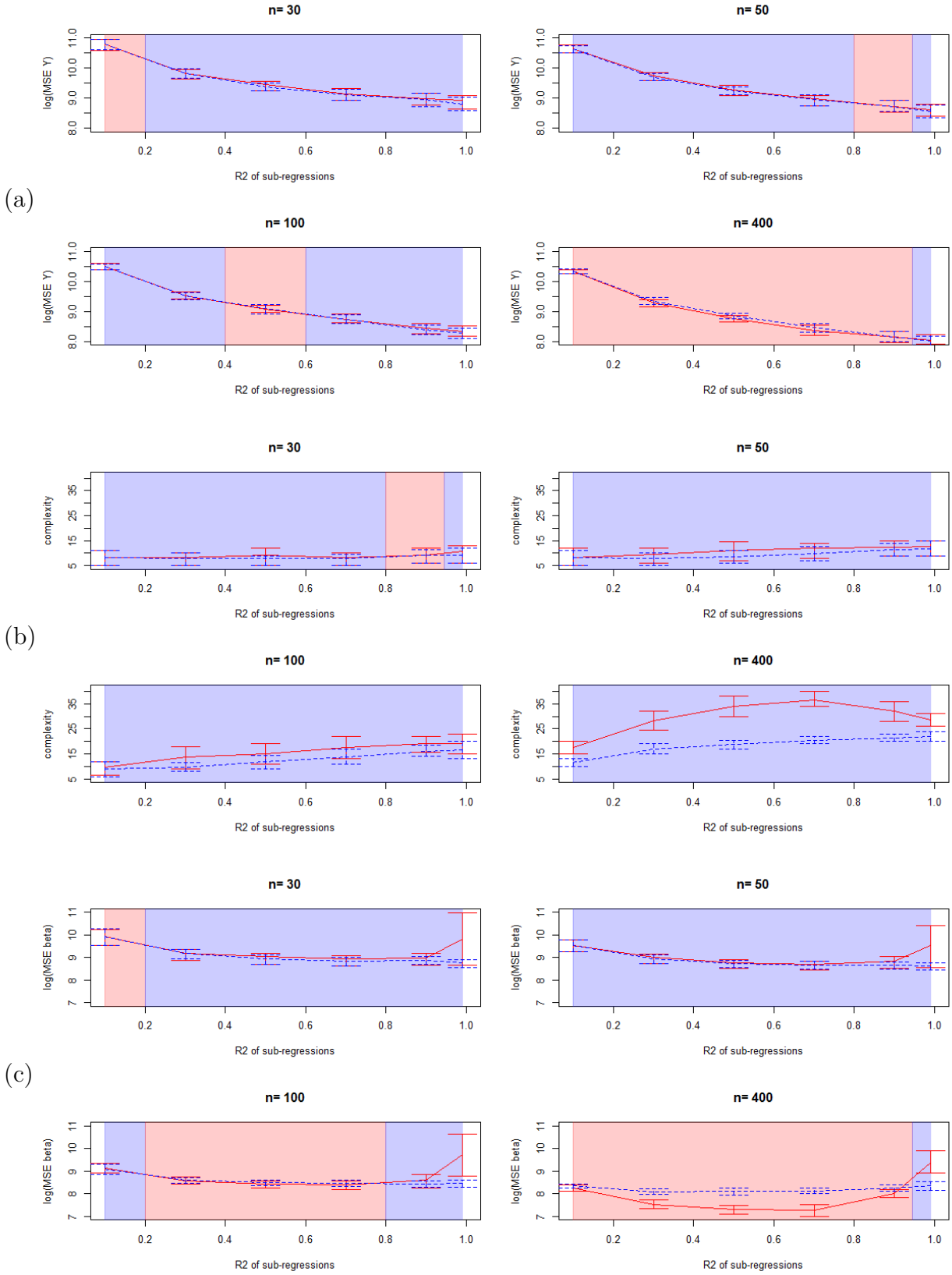


Figure 6.7: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends on all variables in X

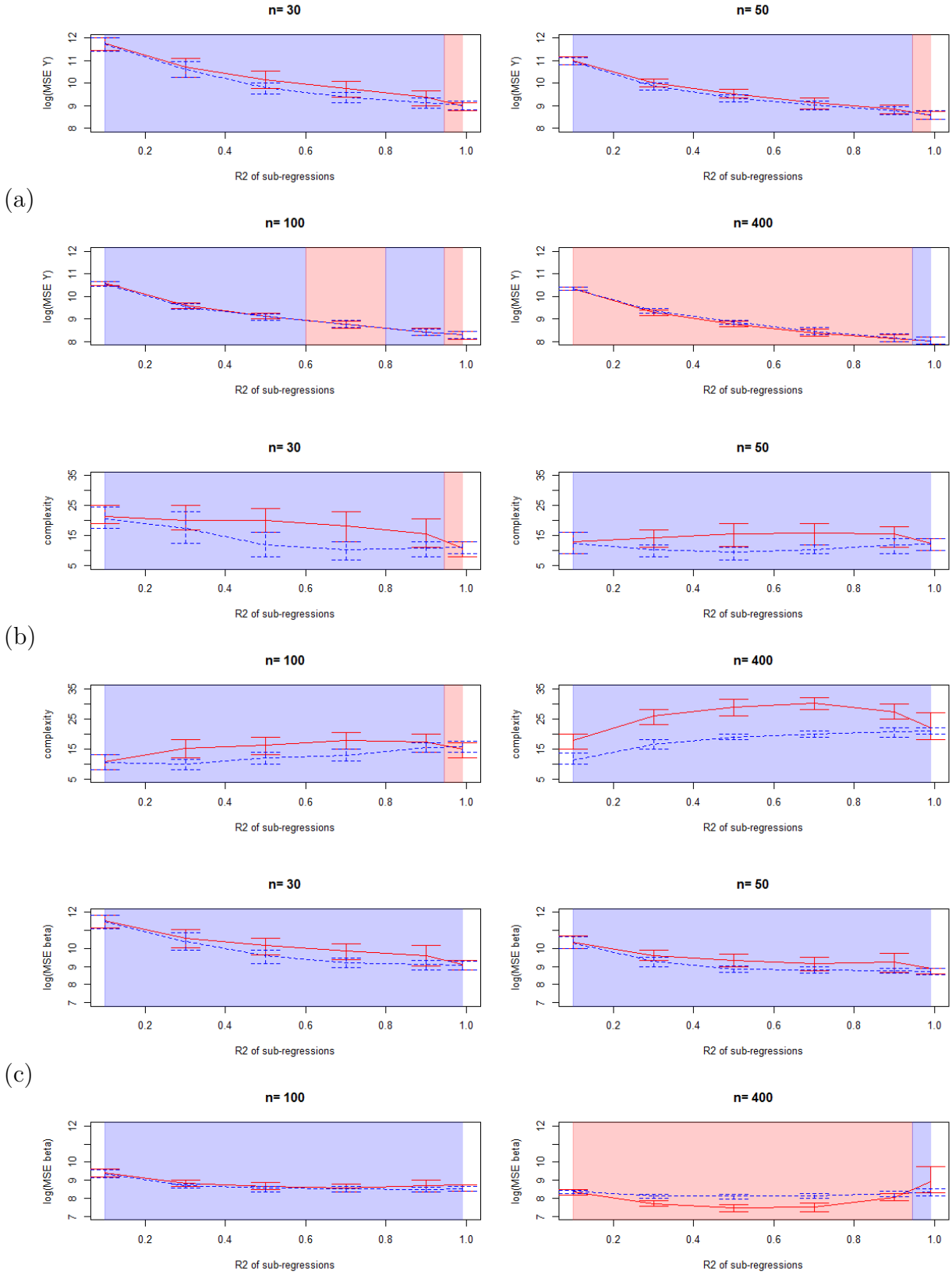


Figure 6.8: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when Y depends on all variables in X

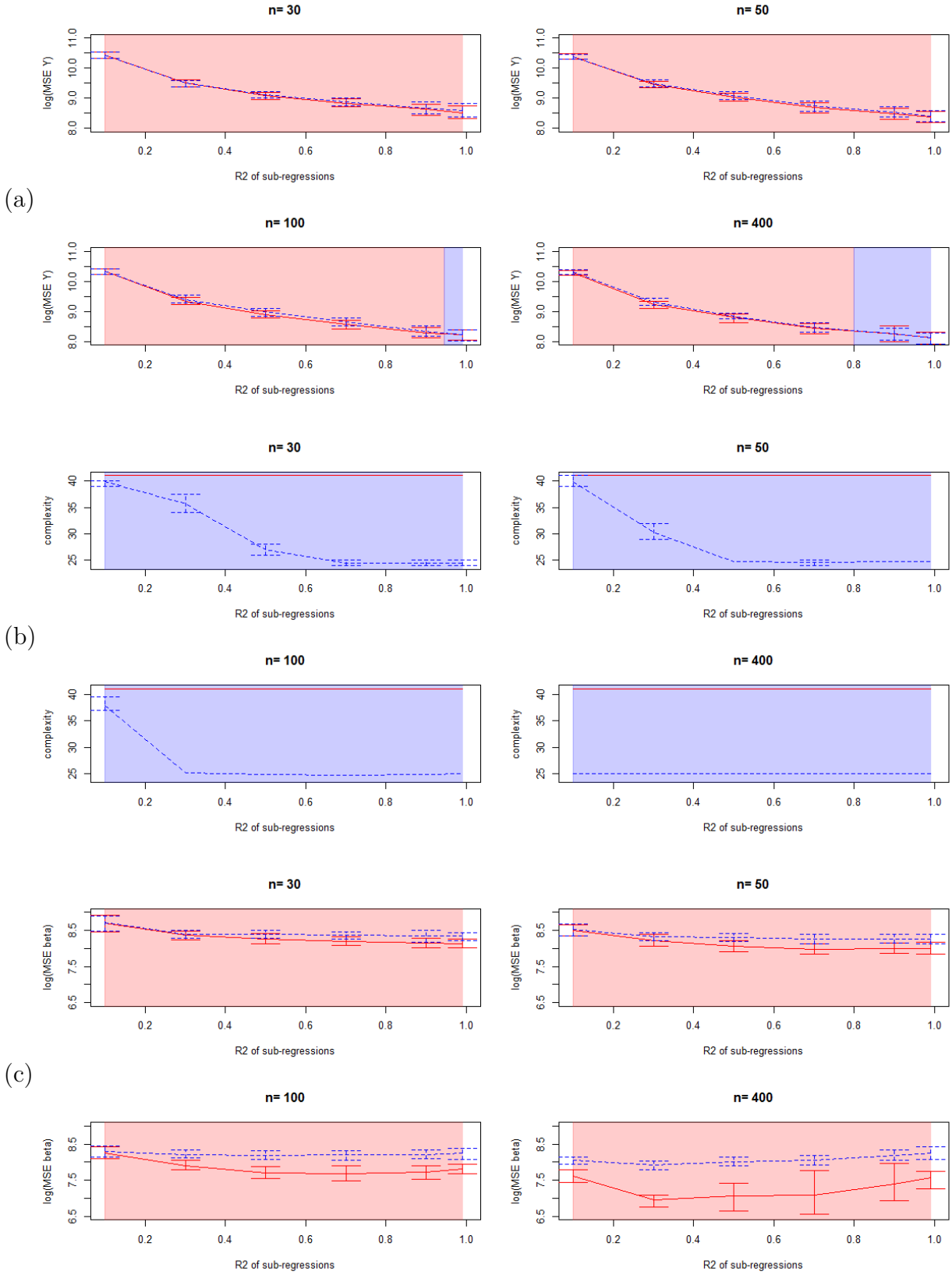


Figure 6.9: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

6.3.2 Y depends only on covariates in X_f

We want to see what happens in real life, when some covariates are irrelevant to describe $\boldsymbol{Y}$ according to the given dataset. We could generate $\boldsymbol{Y}$ with a random subset of $\boldsymbol{X}$ but in such a case, it would be impossible to say whether results come from sparsity or from the ratio of covariates in the subset that are in $\boldsymbol{X}_r$. Moreover, we will study real datasets in the next chapter so the only pattern to test here are those with some irrelevant covariates and relevant covariates only in one part of the partition on $\boldsymbol{X}$.

We start with $\boldsymbol{Y}$ depending only on covariates in $\boldsymbol{X_f}$. 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 $\boldsymbol{Y}$ depends on the 24 covariates in $\boldsymbol{X}_f$ 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 6.3.1.

Ordinary Least Squares: In figure 6.10 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 6.11 to 6.13)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 $\boldsymbol{Y}$ depending on the whole dataset $\boldsymbol{X}$.

Ridge regression: Ridge regression (figure 6.14) is finally improved here by our pre-treatment by selection, like if we had added variable selection feature to the ridge regression. It is the method that provides the best results, but only because $\boldsymbol{Y}$ depends on all covariates in $\boldsymbol{X}_f$. Our pre-treatment is limited in terms of variable selection.

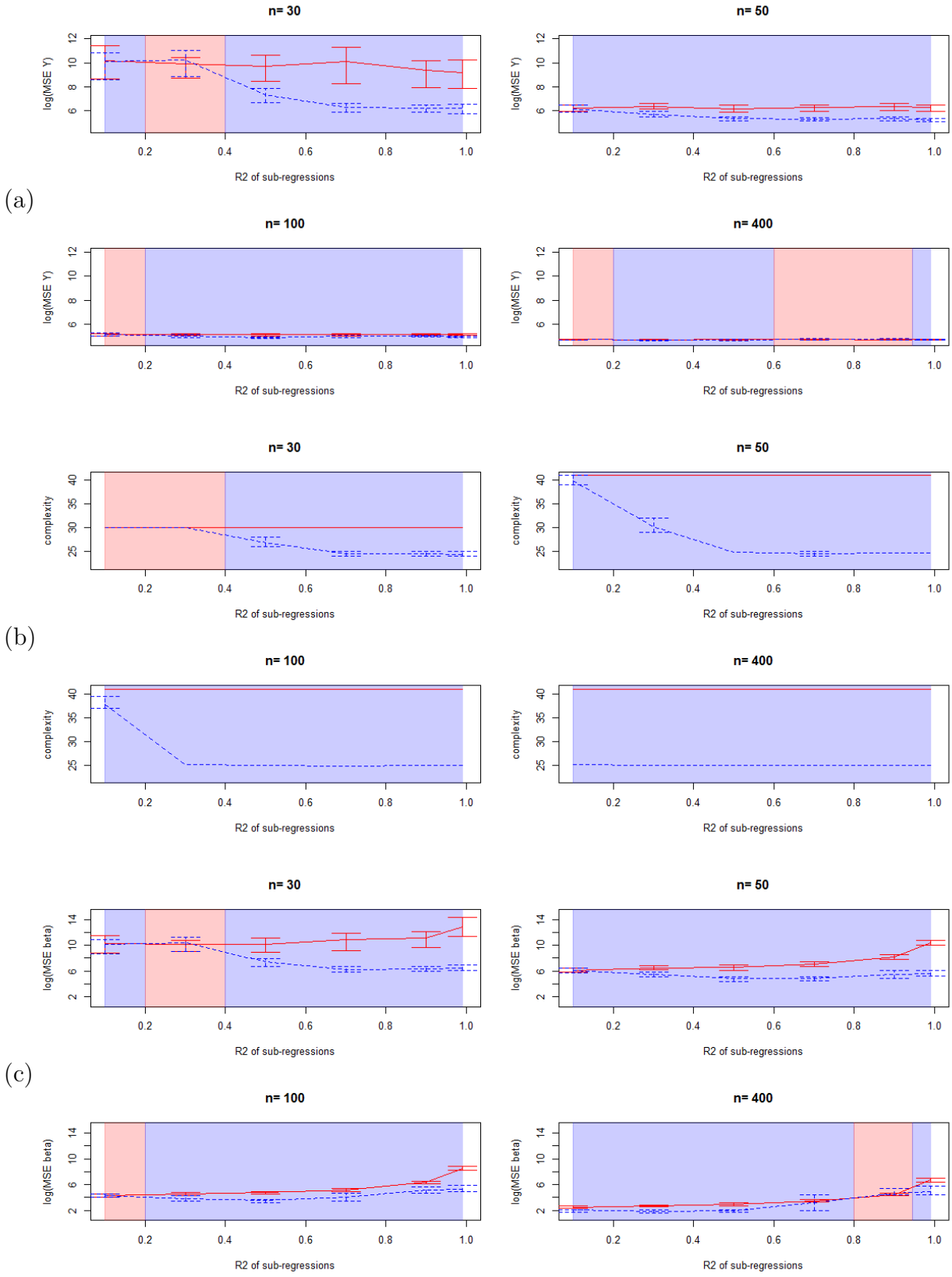


Figure 6.10: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

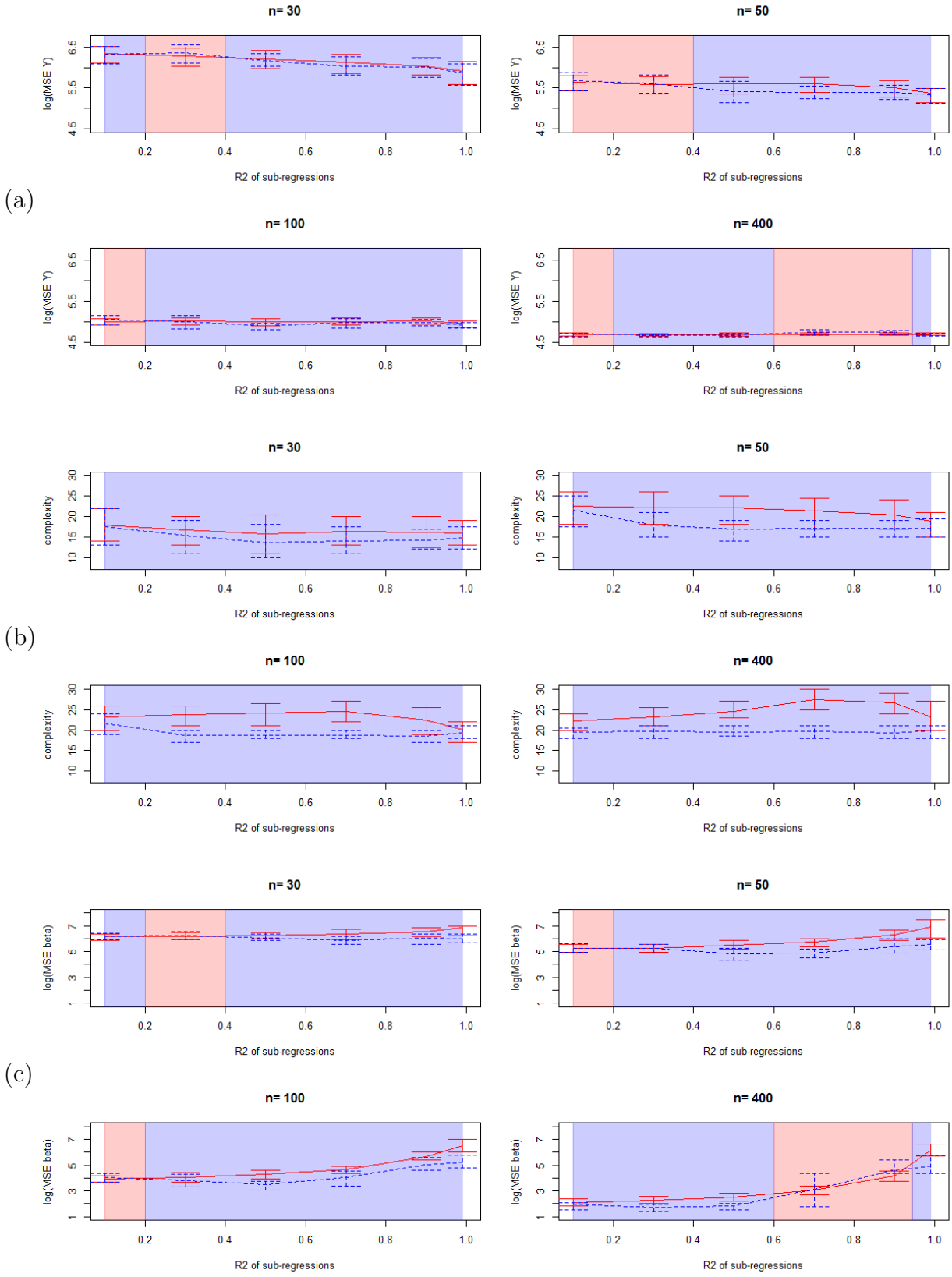


Figure 6.11: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when Y depends only on covariates in X_f

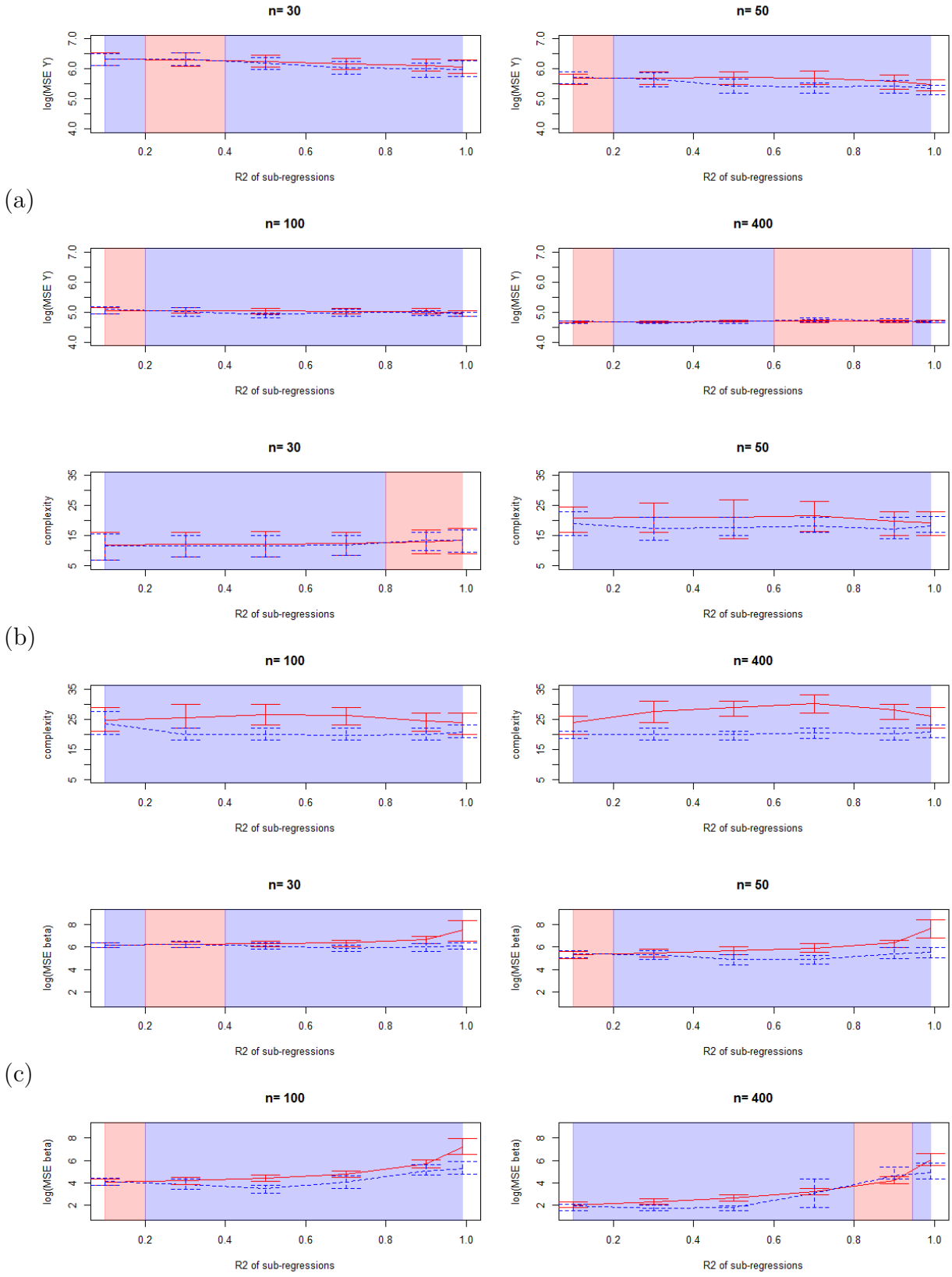


Figure 6.12: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends only on covariates in X_f

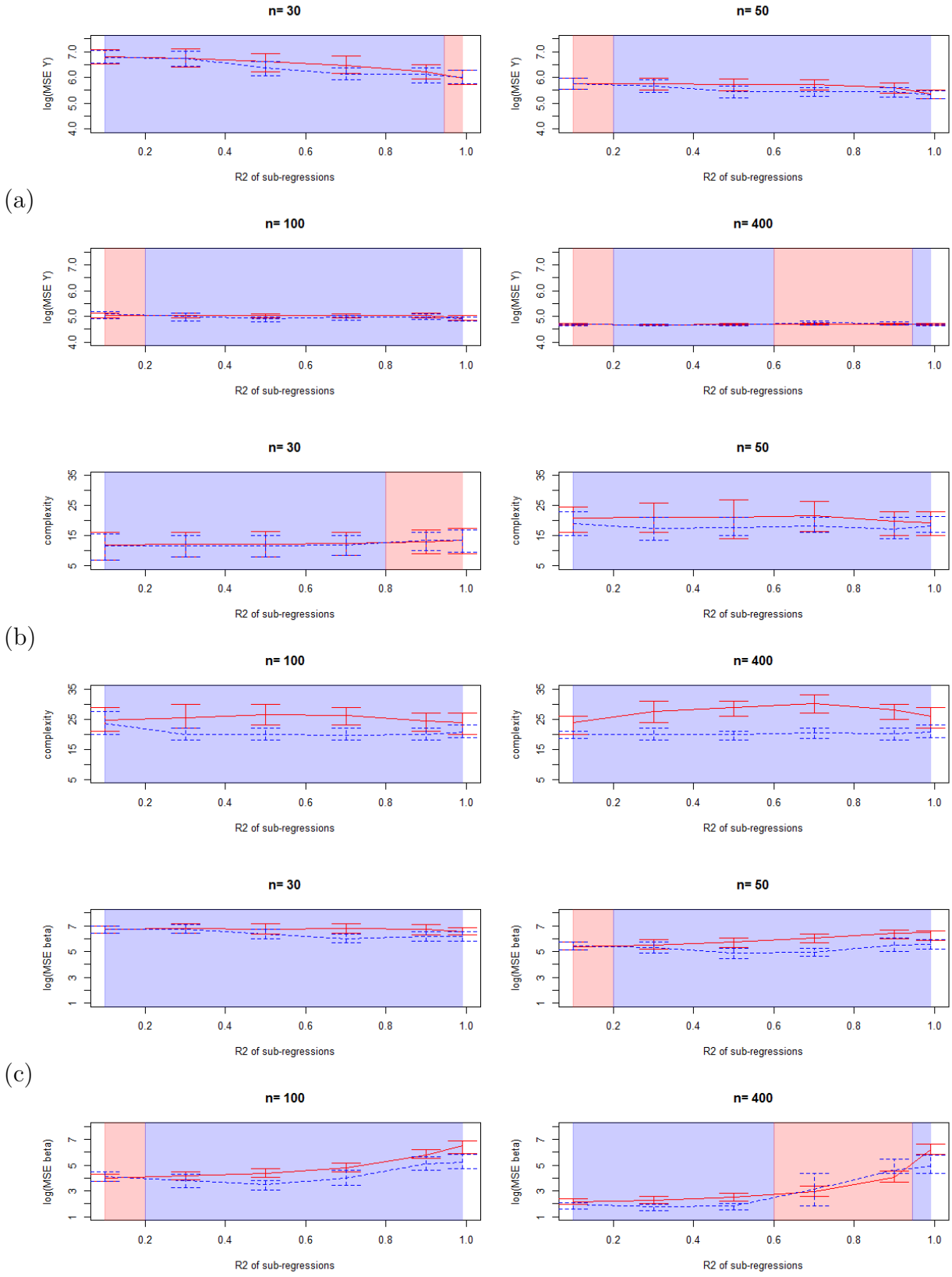


Figure 6.13: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when Y depends only on covariates in X_f

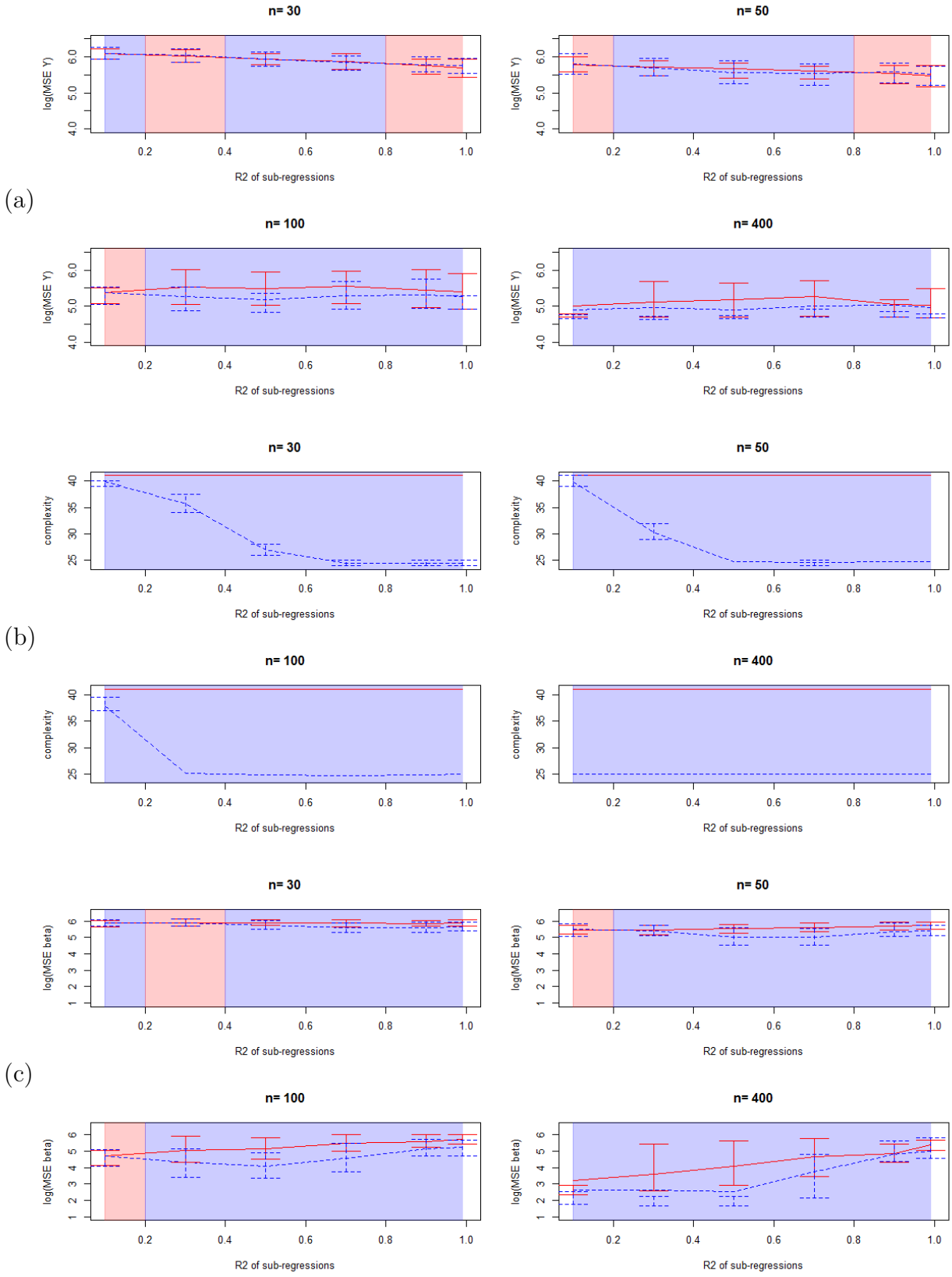


Figure 6.14: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

6.3.3 Y depends only on covariates in X_r

We now try the method with a response depending only on variables in $\boldsymbol{X}_r$. The datasets used here were still the same. Depending only on $\boldsymbol{X}_r$ implies sparsity and impossibility to obtain the true model when using the true structure. We get unbiased models but with an increase in the variance as described in equation 4.5.
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 6.15) 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 helps whatever the true model is. This is a really encouraging result. Improvement for small correlations but $n < d$ comes from dimension reduction. When you do not 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 $\boldsymbol{X}_r$ 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 6.16 to 6.18) still are impacted by correlations but not enough to be improved by our marginal model. Neither is the ridge regression (Figure 6.19). The results confirm that this is the worst case for our marginal model.

Real datasets will provide $\boldsymbol{Y}$ depending on a mix of covariates from both $\boldsymbol{X}_f$ and $\boldsymbol{X}_r$ so our marginal model could help even with variable selection methods or ridge regression. We also recall that the structure $\boldsymbol{S}$ 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 $\hat{\boldsymbol{S}}$ is found, trying the marginal model has no cost and should be tested.

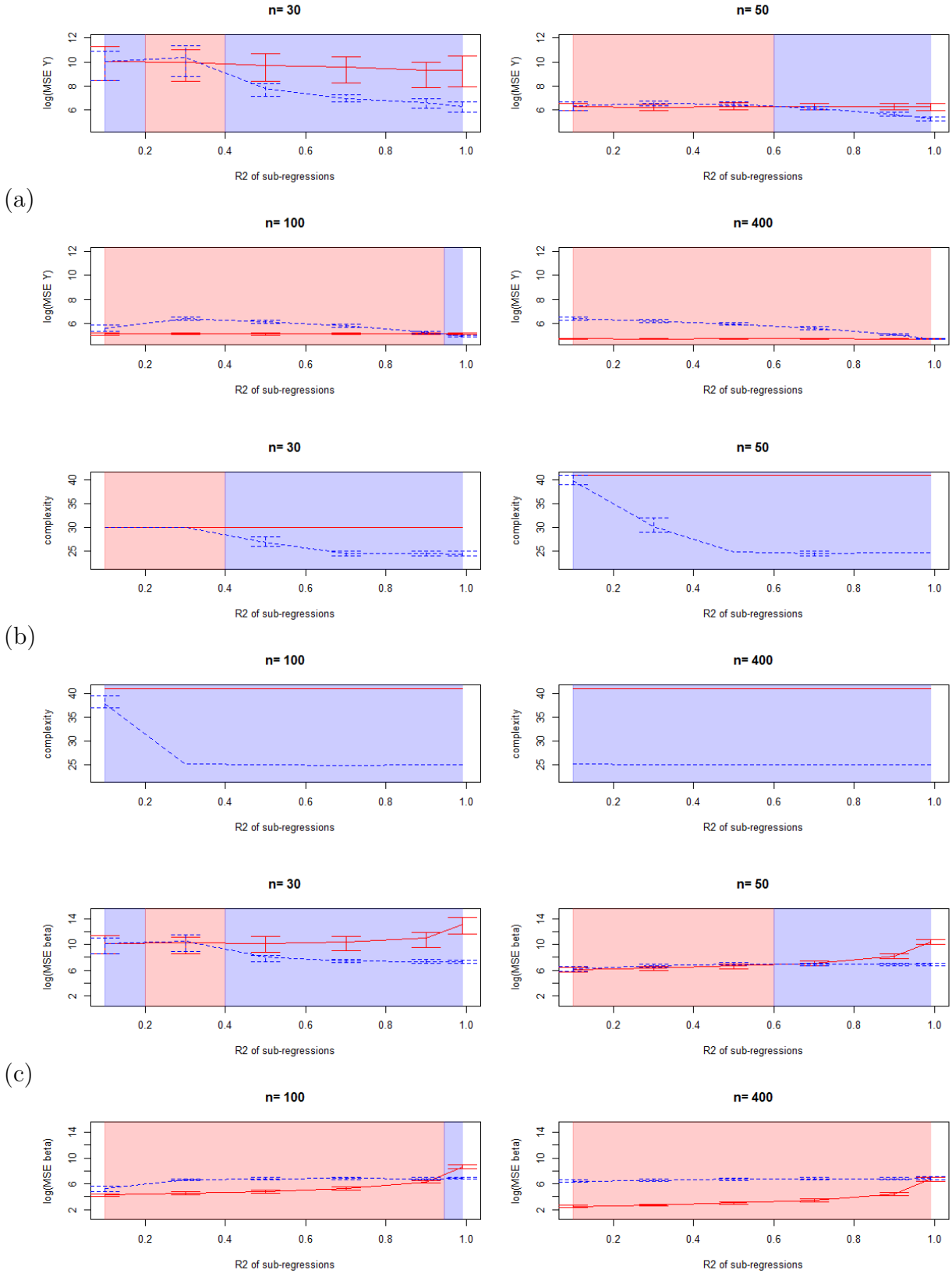


Figure 6.15: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

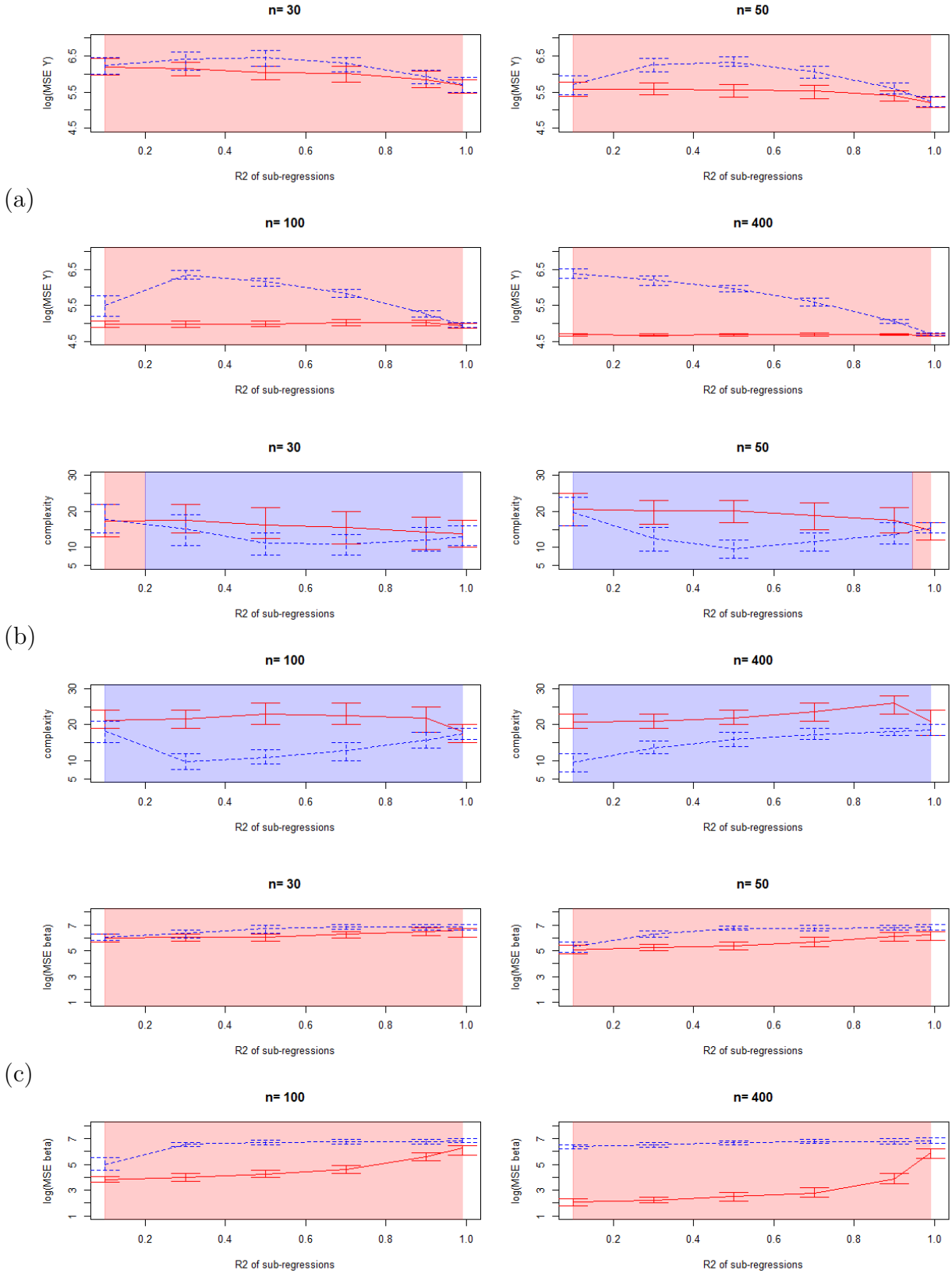


Figure 6.16: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Elasticnet when Y depends only on covariates in X_r

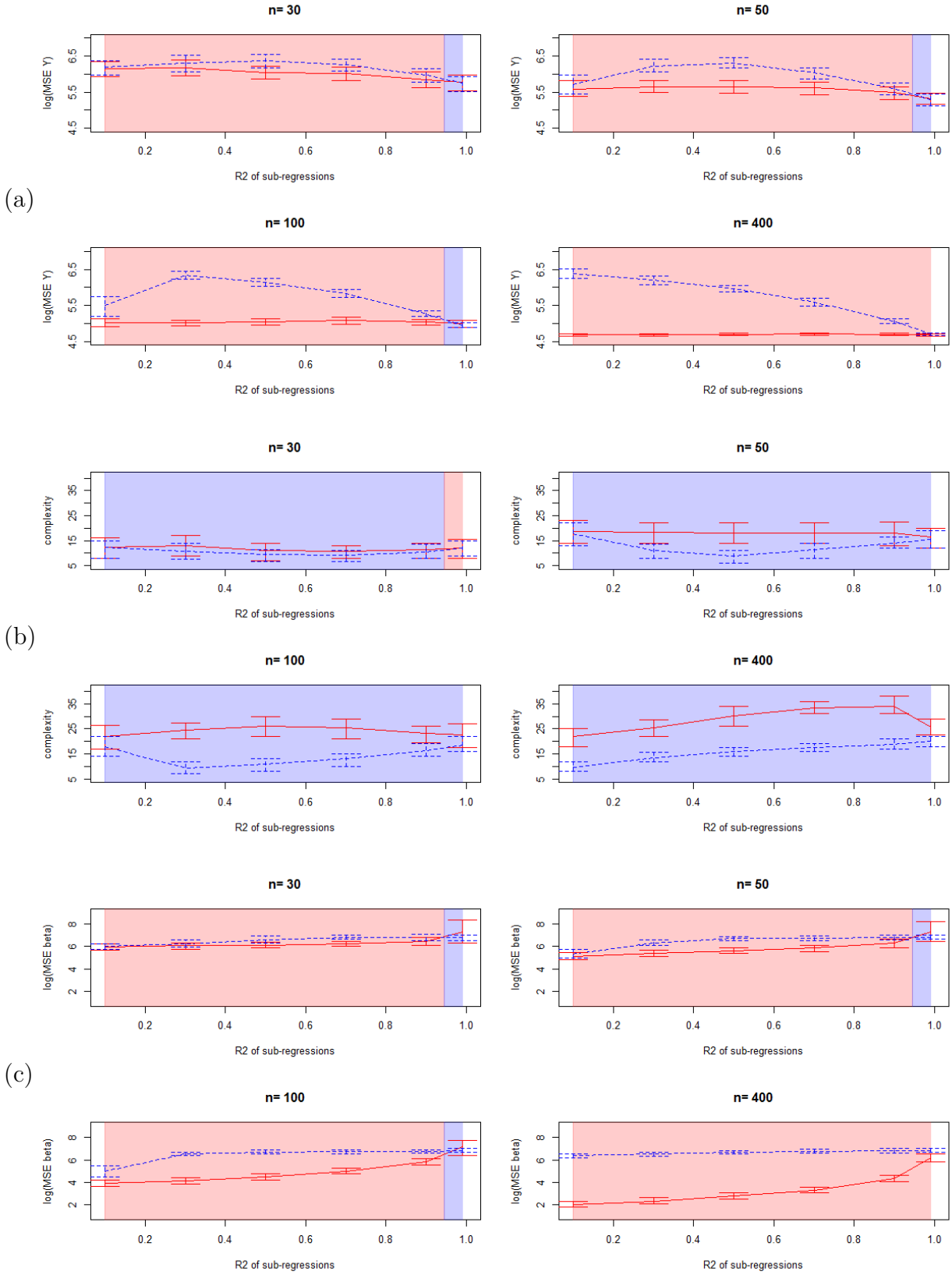


Figure 6.17: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Stepwise when Y depends only on covariates in X_r

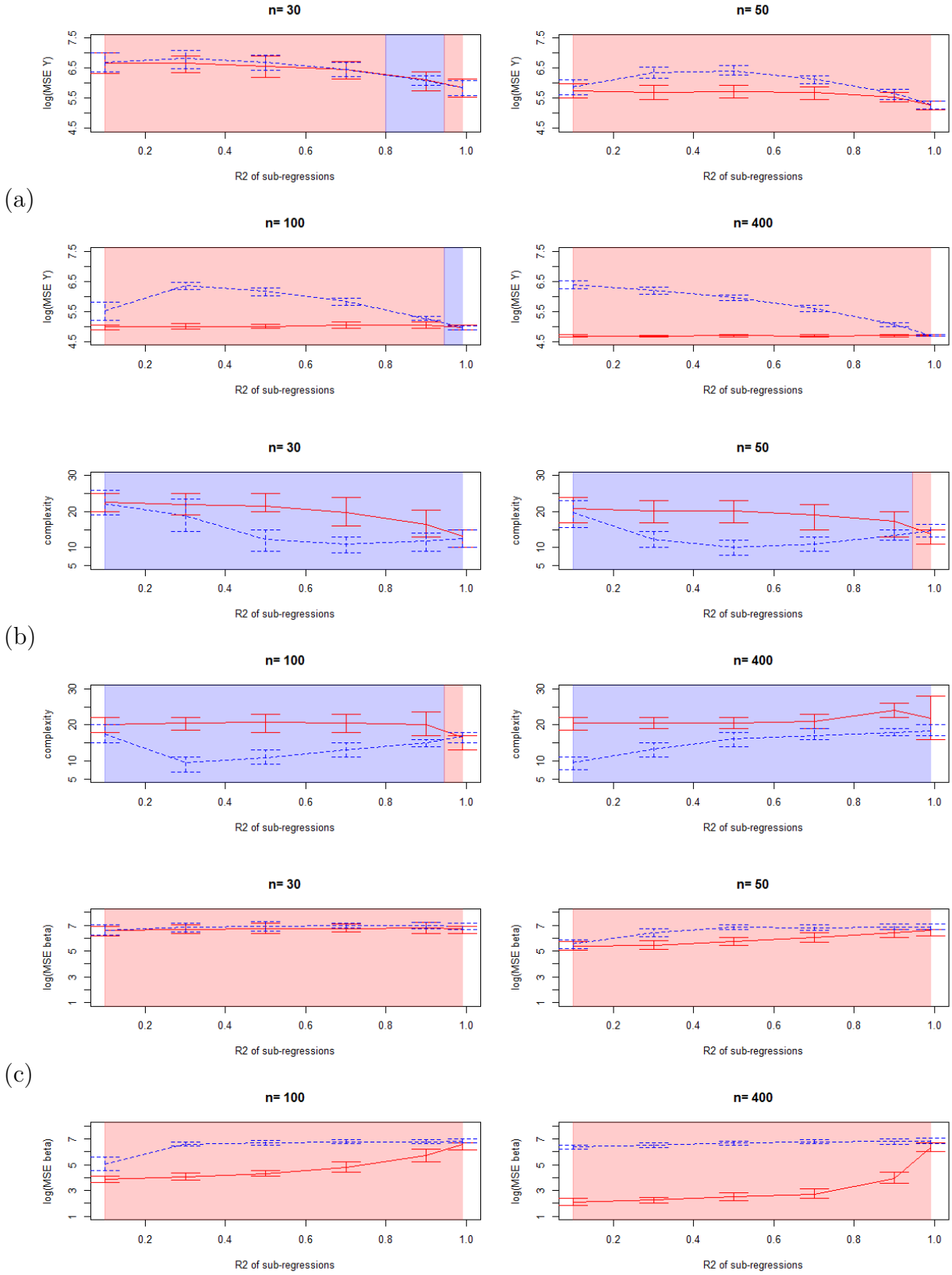


Figure 6.18: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

Ridge regression when $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X}_r$

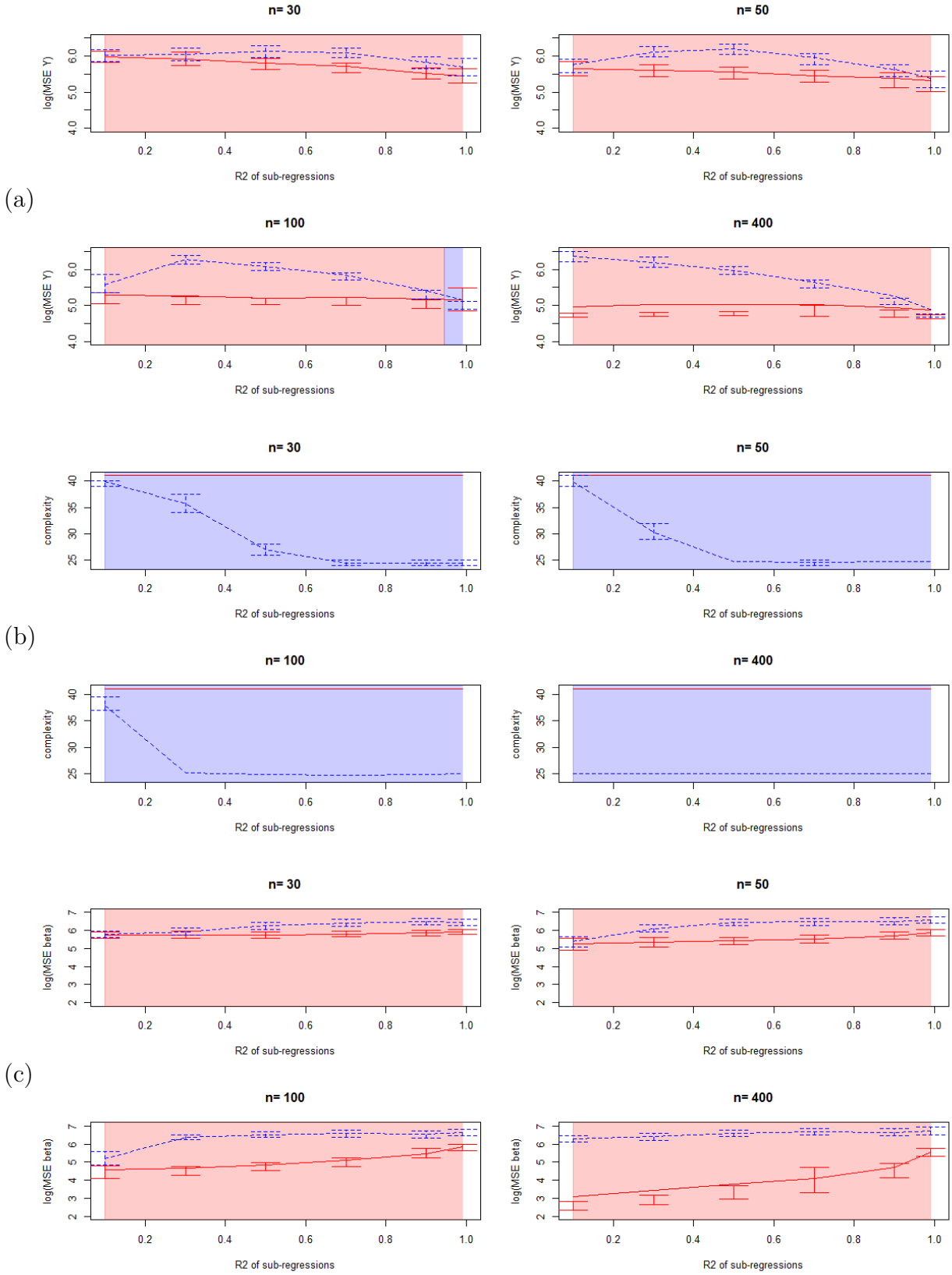


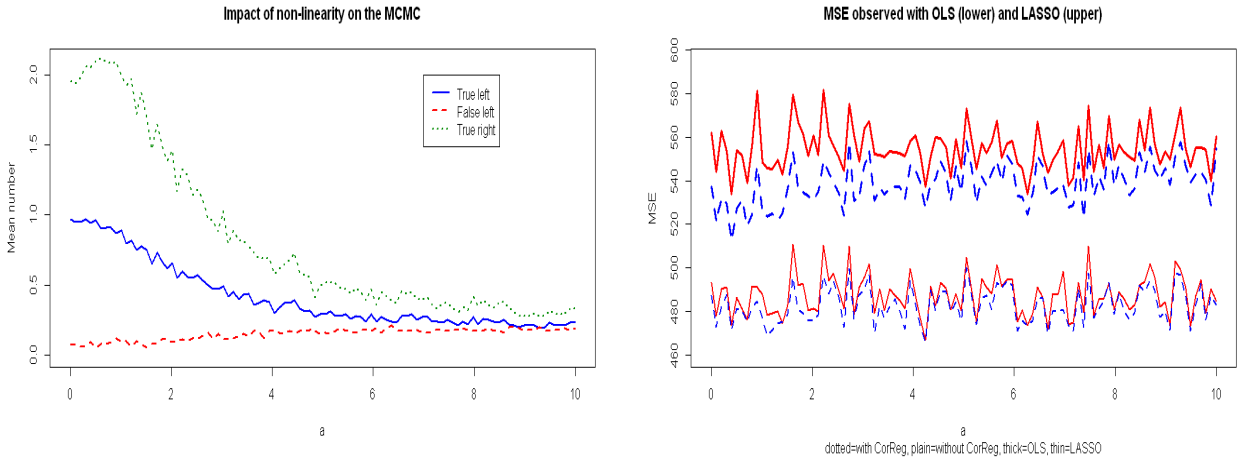
Figure 6.19: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), red=classical (complete) model, blue=marginal model

6.3.4 Robustness with non-linear case

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

$$\boldsymbol{Y} = \sum_{i=1}^{7} \boldsymbol{X}^i + \boldsymbol{\varepsilon}_Y.$$

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.



(a) Evolution of the quality of $\hat{\boldsymbol{S}}$ when the paramater a increases

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

Figure 6.20: Non-linear case analysis.

Figure 6.20(b) illustrates the advantage of using `CorReg` even with non-linear structures. Figure 6.20(a) 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).

6.4 Conclusion

The marginal model still gives good results even with $\hat{\boldsymbol{S}}$. We also observe that the estimation of the structure is satisfying. But these are only simulated datasets and we have to confirm that our hypotheses can face the reality of industrial datasets. Moreover, we see that the pattern of the true regression on $\boldsymbol{Y}$ leads to fundamental changes in the quality of the prediction. So we will test `CorReg` on real datasets (Chapter 7) and then try to improve the marginal model to better fit the patterns that are not favorable (those whith $\boldsymbol{Y}$ depending only on redundant covariates) to the marginal model (Chapter 8).

Chapter 7

Experiments on steel industry

Abstract: Here are the numerical results obtained from real datasets. As in the previous chapter we have to estimate the structure. But here we do not know if the true model follows our hypotheses or not. We will see if the model does make sense in real life as we supposed previously. These experiments were made on real datasets from ArcelorMittal and some informations are confidential so the reader can experiment a kind of frustration but efforts were made to keep these experiments interesting.

7.1 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 :

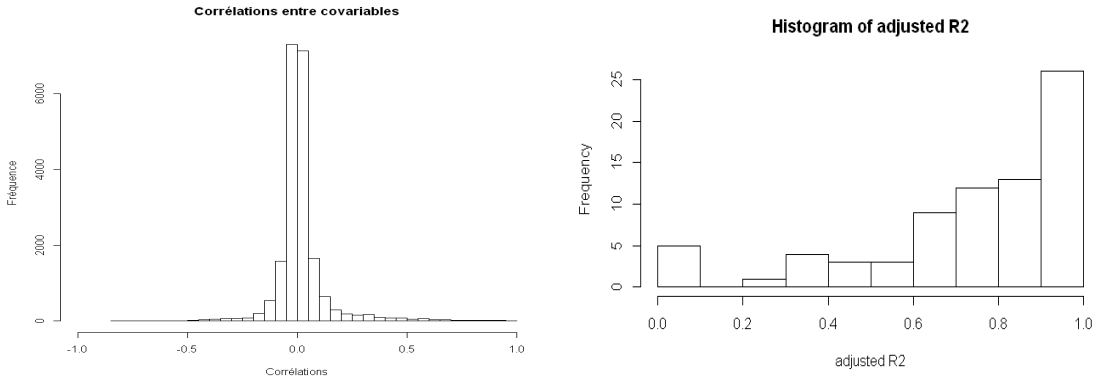
- a quality parameter (confidential) as response variable,
- $d = 205$ variables from the whole process to explain it.

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 7.1(a) gives an histogram of ρ where we can see that, however, many other variables are not so highly correlated.

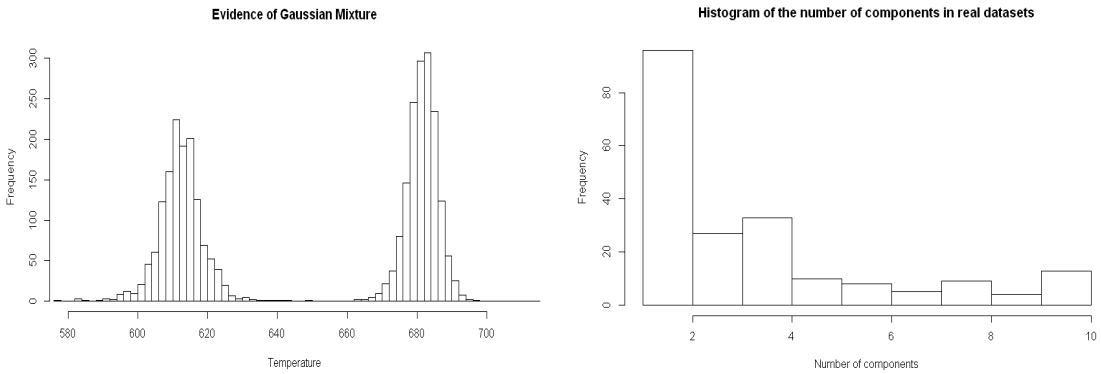
CorReg estimated a structure of $d_r = 76$ sub-regressions with a mean of $\bar{\boldsymbol{d}}_p = 5.17$ predictors. In the resulting uncorrelated covariate set $\boldsymbol{X}_f$ the number of values $|\rho| > 0.7$ is 79.33% smaller than in $\boldsymbol{X}$. Indeed, Figure 7.1(b) displays the histogram of adjusted R^2 value (R^2_{adj}) and we can see that essentially large values of R^2_{adj} 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).



(a) (b)

Figure 7.1: Quality case study: (a) Histogram of correlations ρ in $\boldsymbol{X}$, (b) histogram of the adjusted R^2_{adj} 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 7.2(a). In particular, the number of components is quite moderate as seen in Figure 7.2(b).



(a) (b)

Figure 7.2: 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 7.1 displays predictive results associated to different estimation methods with and without pre-treatment. We can see that the reduced model improves the results for

each method tested in terms of prediction, with generally a more parsimonious regression on $\boldsymbol{Y}$. 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	On $\hat{\boldsymbol{X}}_f$ (`CorReg`'s reduced model)	On $\boldsymbol{X}$ (complete model)
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 7.1: Quality case study: Results obtained on a validation sample ($n = 847$ individuals). In bold, the best MSE value.

7.2 Production case study

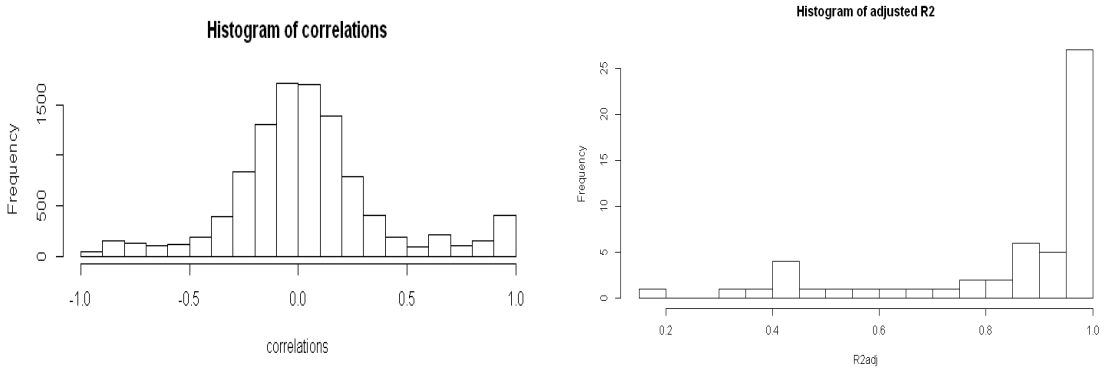
This second example is about a phenomenon that impacts the productivity of a steel plant. We have:

- 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 7.3(a) shows that many variables are highly correlated. CorReg found $d_r = 55$ sub-regressions and corresponding R^2_{adj} values are displayed in Figure 7.3(b). One of them seems to be weak ($R^2_{adj} = 0.17$) 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.

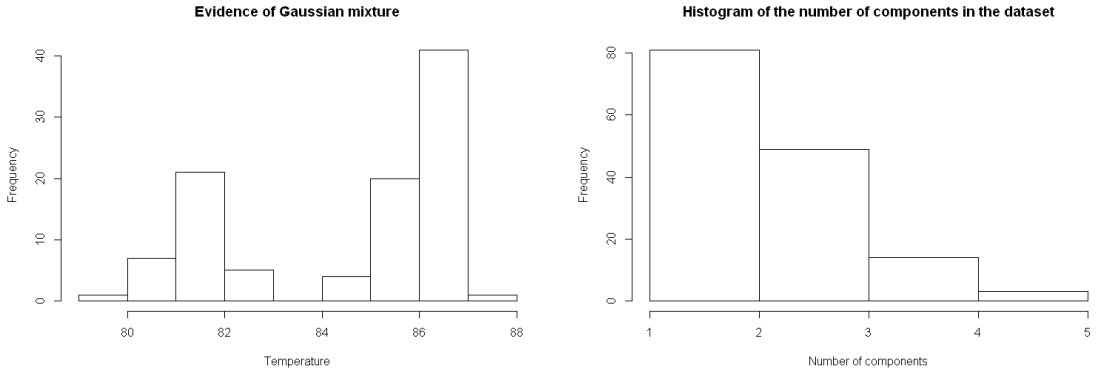
As in the previous quality case study, we note that the uncorrelated variables can be very well-modeled by parsimonious Gaussian mixtures as it is illustrated by Figure 7.4(a). In particular, the number of components is really moderate as seen in Figure 7.4(b).

Table 7.2 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 $\boldsymbol{Y}$.



(a) (b)

Figure 7.3: Production case study: (a) Histogram of correlations ρ in $\boldsymbol{X}$, (b) histogram of the adjusted R^2_{adj} for the $d_r = 55$ sub-regressions.



(a) (b)

Figure 7.4: 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.

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 7.2: Production case study: Results obtained with leave-one out cross-validation ($n = 100, d = 145$). Predictive 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 $\boldsymbol{Y}$ as a continuous variable and then made imputation by 1 when $\hat{\boldsymbol{Y}} > 0.5$ and by 0 else.

In this precise case, `CorReg` found a structure that helped to decorrelate covariates in interpretation and to find the relevant part of the process to optimize. This product is

made by a long process that requires several steel plants so it was necessary to point out the steel plant where the problem occurred. We now have the proof that our method can help on real statistical studies.

7.3 Conclusion

We have used the package `CorReg` with real datasets and the results obtained by our model were better than those from classical methods in terms of prediction and parsimony. But more than that we have found structures between the covariates that did make sense for the metallurgists. The interpretation of both the sub-regressions structure and the main regression model have helped to improve the process so it does confirm that explicit modeling of the correlations and marginalization were good choices. However, we want to investigate other manners to take benefits from the sub-regression structures: sequential estimation and missing values management.

Part II

Usage of the residuals by plug-in and extension to missing data

Chapter 8

Using coefficients of regression and sub-regression to improve prediction

Abstract: We have seen that eviction of redundant covariates can often improve the results by an efficient trade-off between dimension reduction and better conditioning versus keeping all the covariates. But the fact is that we have lost some information and we would like to find a way to use this information. So we propose a plug-in model to use the redundant covariates in a second estimation step with an estimate of the residuals. It is a sequential estimation of $\boldsymbol{\beta}$.

8.1 Motivations

The structure $\boldsymbol{S}$ of sub-regressions between the covariates has given us the opportunity to introduce a marginal model that in fact removes the response covariates from the model. 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 7). But even if the sub-regressions are strong, we face a loss of information that can be damageable as we have seen in Section 6.3.3.

The redundant covariates $\boldsymbol{X}_r$ were only used to estimate $\boldsymbol{S}$ and only the partition (given by J_r) was used to estimate $\boldsymbol{Y}$. With the marginalization we have not used the part of $\boldsymbol{X}_r$ that is independent of $\boldsymbol{X}_f$ (the noise of the sub-regressions). We will try to use it with a sequential approach relying on the coefficients of sub-regression $\hat{\boldsymbol{\alpha}}^*$ and on the estimate $\hat{\boldsymbol{\beta}}_f^*$ by profile likelihood.

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 (equation 4.7) that makes appear the coefficient $\boldsymbol{\beta}_r$ in both the noise and $\boldsymbol{\beta}_f^*$. Thus we are able to obtain by plug-in better estimates of $\boldsymbol{\beta}_r$ and $\boldsymbol{\beta}_f$ in terms of bias. Once again final result will depend on the bias-variance trade-off and numerical results will show what to expect.

In this chapter we note $\longrightarrow$ the convergence in probability when n grows to $+\infty$.

8.2 A plug-in model to reduce the noise

We propose a plug-in model to reduce the noise of the marginal model. We had the true complete model:

$$\boldsymbol{Y} = \boldsymbol{X}_f\boldsymbol{\beta}_f + \boldsymbol{X}_r\boldsymbol{\beta}_r + \boldsymbol{\varepsilon}_Y,$$

with a structure $\boldsymbol{S}$ of sub-regressions on $\boldsymbol{X}$:

$$\boldsymbol{X}_r = \boldsymbol{X}_f\boldsymbol{\alpha}^* + \boldsymbol{\varepsilon}. \tag{8.1}$$

Then by marginalization on $\boldsymbol{X}_r$ we obtained:

$$\boldsymbol{Y} = \boldsymbol{X}_f \underbrace{(\boldsymbol{\beta}_f + \boldsymbol{\alpha}^*\boldsymbol{\beta}_r)}_{\boldsymbol{\beta}_f^*} + \underbrace{\boldsymbol{\varepsilon}\boldsymbol{\beta}_r + \boldsymbol{\varepsilon}_Y}_{\boldsymbol{\varepsilon}_Y^*}. \tag{8.2}$$

Plug-in approach: We get from equation (8.2):

$$\boldsymbol{\varepsilon}_Y^* = \boldsymbol{\varepsilon}\boldsymbol{\beta}_r + \boldsymbol{\varepsilon}_Y. \tag{8.3}$$

Then the Best Linear Unbiased Estimator (BLUE) for $\boldsymbol{\beta}_r$ is given (OLS estimator) by:

$$\hat{\boldsymbol{\beta}}_r = (\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon})^{-1}\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}_Y^*. \tag{8.4}$$

But it depends on $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}_Y^*$ that are unknown. The plug-in estimator of $\boldsymbol{\beta}_r$ we propose in this chapter does rely on the fact that we already have the following estimators (from equations (8.1) and (8.2)):

$$\begin{aligned} \hat{\boldsymbol{\varepsilon}} &= \boldsymbol{X}_r - \boldsymbol{X}_f\hat{\boldsymbol{\alpha}}^* \text{ and} \\ \hat{\boldsymbol{\varepsilon}}_Y^* &= \boldsymbol{Y} - \boldsymbol{X}_f\hat{\boldsymbol{\beta}}_f^* \end{aligned}$$

that we can use by plug-in.

Estimation of $\boldsymbol{\beta}_r$: We define a plug-in estimator for $\boldsymbol{\beta}_r$:

$$\begin{aligned} \hat{\boldsymbol{\beta}}_r^{\varepsilon} &= (\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}_Y^* \\ &= ((\boldsymbol{X}_r - \boldsymbol{X}_f\hat{\boldsymbol{\alpha}}^*)'(\boldsymbol{X}_r - \boldsymbol{X}_f\hat{\boldsymbol{\alpha}}^*))^{-1}(\boldsymbol{X}_r - \boldsymbol{X}_f\hat{\boldsymbol{\alpha}}^*)'(\boldsymbol{Y} - \boldsymbol{X}_f\hat{\boldsymbol{\beta}}_f^*) \end{aligned} \tag{8.5}$$

that depends on all covariates in $\boldsymbol{X}$ and relies on the estimated coefficients of sub-regressions $\hat{\boldsymbol{\alpha}}^*$ and on the estimate $\hat{\boldsymbol{\beta}}_f^*$ of the coefficients in the marginal model. It is the Ordinary Least squares estimator but we can use any other linear regression estimator, allowing then to make variable selection again, that is to decide which covariates will come back in the model. This second linear regression does not have any intercept (even if it was added in $\boldsymbol{X}$ it won't depend on any other covariate and then won't be part of $\boldsymbol{X}_r$). Then we can estimate $\boldsymbol{Y}$ by:

$$\hat{\boldsymbol{Y}}_{plug-in} = \boldsymbol{X}_f\hat{\boldsymbol{\beta}}_f^* + \hat{\boldsymbol{\varepsilon}}\hat{\boldsymbol{\beta}}_r^{\varepsilon}. \tag{8.6}$$

We have now (sequentially) used all the covariates to estimate the parameters of the regression on $\boldsymbol{Y}$. In the following we suppose that $\hat{\boldsymbol{\alpha}}^*$ and $\hat{\boldsymbol{\beta}}_f^*$ are OLS estimators.

Properties: The estimators $\hat{\boldsymbol{\alpha}}^*$ and $\hat{\boldsymbol{\beta}}_f^*$ sequentially used here by plug-in are consistent (OLS estimators):

$$\hat{\boldsymbol{\alpha}}^* \longrightarrow \boldsymbol{\alpha}^* \text{ and } \hat{\boldsymbol{\beta}}_f^* \longrightarrow \boldsymbol{\beta}_f^*.$$

Then, by the continuous mapping theorem, our new estimator from equation (8.5) converges in probability to the OLS estimator (defined in equation (8.4)):

$$\begin{aligned}((\boldsymbol{X}_r - \boldsymbol{X}_f\hat{\boldsymbol{\alpha}}^*)'(\boldsymbol{X}_r - \boldsymbol{X}_f\hat{\boldsymbol{\alpha}}^*))^{-1}(\boldsymbol{X}_r - \boldsymbol{X}_f\hat{\boldsymbol{\alpha}}^*)'(\boldsymbol{Y} - \boldsymbol{X}_f\hat{\boldsymbol{\beta}}_f^*) - (\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon})^{-1}\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}_Y^* &\longrightarrow \mathbf{0},\\ \hat{\boldsymbol{\beta}}_r^{\varepsilon} - \hat{\boldsymbol{\beta}}_r &\longrightarrow \mathbf{0}.\end{aligned}$$

We know that OLS estimators are consistent:

$$(\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon})^{-1}\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}_Y^* \longrightarrow \boldsymbol{\beta}_r$$

and then we obtain

$$\hat{\boldsymbol{\beta}}_r^{\varepsilon} \longrightarrow \boldsymbol{\beta}_r,$$

so the plug-in estimator $\hat{\boldsymbol{\beta}}_r^{\varepsilon}$ is consistent. We deduce then

$$\lim_{n\to+\infty} \mathbb{E}(\hat{\boldsymbol{\beta}}_r^{\varepsilon}) = \boldsymbol{\beta}_r \text{ (asymptotically unbiased estimator)},$$

and we have:

$$\begin{aligned}\mathbb{E}(\hat{\boldsymbol{\beta}}_r^{\varepsilon}) &= \mathbb{E}[(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{Y} - \boldsymbol{X}_f\hat{\boldsymbol{\beta}}_f^*)]\\ &= \mathbb{E}[(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{Y} - \boldsymbol{X}_f(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'\boldsymbol{Y})]\\ &= \mathbb{E}[(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \underbrace{\boldsymbol{X}_f(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'}_{\boldsymbol{H}_f})\boldsymbol{Y})]\\ &= (\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\mathbb{E}(Y)\\ &= (\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\boldsymbol{X}\boldsymbol{\beta}.\end{aligned}$$

The variance of the estimator is given by:

$$\begin{aligned}\mathrm{Var}(\hat{\boldsymbol{\beta}}_r^{\varepsilon}) &= \mathrm{Var}[\underbrace{(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)}_{(\boldsymbol{X}_r^{\varepsilon})'}\boldsymbol{Y})]\\ &= (\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\hat{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\,\mathrm{Var}(\boldsymbol{Y})\\ &= \sigma_Y^2(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\hat{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\\ &= \sigma_Y^2\left[\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}[\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\hat{\boldsymbol{\varepsilon}}]^{-1}\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}\right]^{-1}\\ &= \sigma_Y^2\left[(\boldsymbol{X}_r^{\varepsilon})'\boldsymbol{X}_r^{\varepsilon}\right]^{-1}\end{aligned} \tag{8.7}$$

where $\boldsymbol{X}_r^{\varepsilon} = [\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\hat{\boldsymbol{\varepsilon}}]^{-\frac{1}{2}}\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}$ is a $d_r \times d_r$ matrix.
It is clear that when the σ_j^2 tends to 0 then the plug-in estimator explodes. But in such cases the marginal model tends to be the true model without bias. So when a sub-regression is almost exact it is preferable to keep the redundant covariate out of the model. The plug-in model was defined only to use the residuals of the sub-regressions when they are not too small.

Remark: We can improve estimation of $\boldsymbol{\beta}_f$ (in terms of bias) by doing an additional identification step. We have $\boldsymbol{\beta}_f^* = \boldsymbol{\beta}_f + \boldsymbol{\alpha}^*\boldsymbol{\beta}_r$ so we define the following estimator:

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

Properties of $\hat{\boldsymbol{\beta}}_f^{\varepsilon}$ are the following:

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_f^{\varepsilon} &\longrightarrow \boldsymbol{\beta}_f^* - \boldsymbol{\alpha}^*\boldsymbol{\beta}_r \text{ (consistent estimator)},\\
\lim_{n\to+\infty} \mathbb{E}(\hat{\boldsymbol{\beta}}_f^{\varepsilon}) &= \boldsymbol{\beta}_f^* - \boldsymbol{\alpha}^*\boldsymbol{\beta}_r = \boldsymbol{\beta}_f \text{ (asymptotically unbiased), and}\\
\mathbb{E}(\hat{\boldsymbol{\beta}}_f^{\varepsilon}) &= \mathbb{E}[\hat{\boldsymbol{\beta}}_f^* - \hat{\boldsymbol{\alpha}}^*\hat{\boldsymbol{\beta}}_r^{\varepsilon}]\\
&= \mathbb{E}[(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'\boldsymbol{Y} - \hat{\boldsymbol{\alpha}}^*(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{Y} - \boldsymbol{X}_f\hat{\boldsymbol{\beta}}_f^*)]\\
&= \mathbb{E}[(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'\boldsymbol{Y} - \hat{\boldsymbol{\alpha}}^*(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{Y} - \boldsymbol{X}_f(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'\boldsymbol{Y})]\\
&= \mathbb{E}\left[\left((\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f' - \hat{\boldsymbol{\alpha}}^*(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\right)\boldsymbol{Y}\right]\\
&= \left[(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f' - \hat{\boldsymbol{\alpha}}^*(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\right]\boldsymbol{X}\boldsymbol{\beta}.
\end{aligned}$$

The variance is given by:

$$\begin{aligned}
\mathrm{Var}(\hat{\boldsymbol{\beta}}_f^{\varepsilon}) &= \mathrm{Var}[\hat{\boldsymbol{\beta}}_f^* - \hat{\boldsymbol{\alpha}}^*\hat{\boldsymbol{\beta}}_r^{\varepsilon}]\\
&= \mathrm{Var}\left[\left((\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f' - \hat{\boldsymbol{\alpha}}^*(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\right)\boldsymbol{Y}\right]\\
&= \sigma_Y^2[(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1} + \hat{\boldsymbol{\alpha}}^*(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}\hat{\boldsymbol{\varepsilon}}'(\boldsymbol{I}_n - \boldsymbol{H}_f)\hat{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}})^{-1}(\hat{\boldsymbol{\alpha}}^*)']\\
&= \sigma_Y^2\left[(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1} + \hat{\boldsymbol{\alpha}}^*\left[(\boldsymbol{X}_r^{\varepsilon})'\boldsymbol{X}_r^{\varepsilon}\right]^{-1}(\hat{\boldsymbol{\alpha}}^*)'\right]. \qquad (8.8)
\end{aligned}$$

Comparison with OLS: Up to a permutation of the columns of $\boldsymbol{X}$ we have $\boldsymbol{X} = (\boldsymbol{X}_f, \boldsymbol{X}_r)$ and then:

$$\boldsymbol{X}'\boldsymbol{X} = (\boldsymbol{X}_f, \boldsymbol{X}_r)'(\boldsymbol{X}_f, \boldsymbol{X}_r) = \begin{pmatrix} \boldsymbol{X}_f'\boldsymbol{X}_f & \boldsymbol{X}_f'\boldsymbol{X}_r \\ \boldsymbol{X}_r'\boldsymbol{X}_f & \boldsymbol{X}_r'\boldsymbol{X}_r \end{pmatrix}.$$

We have $\mathrm{Var}_{ols}(\hat{\boldsymbol{\beta}}) = \sigma_Y^2(\boldsymbol{X}'\boldsymbol{X})^{-1} = (\mathrm{Var}_{ols}(\hat{\boldsymbol{\beta}}_f), \mathrm{Var}_{ols}(\hat{\boldsymbol{\beta}}_r))'$ so we obtain, using bloc inversion with Schur complement:

$$\begin{aligned}
\mathrm{Var}_{ols}(\hat{\boldsymbol{\beta}}_f) &= \sigma_Y^2\left[(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1} + (\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'\boldsymbol{X}_r[\boldsymbol{X}_r'\boldsymbol{X}_r - \boldsymbol{X}_r'\boldsymbol{X}_f(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'\boldsymbol{X}_r]^{-1}\boldsymbol{X}_r'\boldsymbol{X}_f(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\right],\\
&= \sigma_Y^2\Big[(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1} + \underbrace{(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'\boldsymbol{X}_r}_{\hat{\boldsymbol{\alpha}}_{ols}^*}[\boldsymbol{X}_r'(\boldsymbol{I}_n - \boldsymbol{H}_f)\boldsymbol{X}_r]^{-1}\underbrace{\boldsymbol{X}_r'\boldsymbol{X}_f(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}}_{(\hat{\boldsymbol{\alpha}}_{ols}^*)'}\Big], \qquad (8.9)
\end{aligned}$$

where $\hat{\boldsymbol{\alpha}}_{ols}^*$ is the sub-regression coefficients matrix estimated by OLS without parsimony (all the covariates in $\boldsymbol{X}_r$ supposed to depend on all the covariates in $\boldsymbol{X}_f$). We also have

$$\mathrm{Var}_{ols}(\hat{\boldsymbol{\beta}}_r) = \sigma_Y^2\left[\boldsymbol{X}_r'\boldsymbol{X}_r - \boldsymbol{X}_r'\boldsymbol{X}_f(\boldsymbol{X}_f'\boldsymbol{X}_f)^{-1}\boldsymbol{X}_f'\boldsymbol{X}_r\right]^{-1} = \sigma_Y^2\left[\boldsymbol{X}_r'(\boldsymbol{I}_n - \boldsymbol{H}_f)\boldsymbol{X}_r\right]^{-1}. \qquad (8.10)$$

Variances in equations (8.7) and (8.8) rely on the estimated residuals of the sub-regressions that may give better conditioned matrices ($\boldsymbol{\varepsilon}_j$'s are supposed to be independent) to invert than $\boldsymbol{X}_r$ that appears with OLS applied on $\boldsymbol{X}$ (equations (8.10) and (8.9)) because redundant covariate can have common predictors. When n rises to $+\infty$ both models are consistent so convergence rates will make the difference. Compared to OLS we have a new bias-variance trade-off to study with numerical results.

Compared to the marginal model, the plug-in model uses all the covariates. But when sub-regressions tends to be exact, the variance of the marginal estimator shrinks and $\hat{\boldsymbol{\varepsilon}}$ tends to the zero matrix (that is bad for conditioning). So the marginal model would be better for extreme correlations. Exact sub-regressions ($\boldsymbol{\varepsilon} = \boldsymbol{0}$) make the marginal model the exact true model without additional noise (equation (8.2)) so the plug-in model is not

needed in such cases.
Equations (8.2) and (8.3) are linear regressions and both of them can use any estimator for linear regression. Then the plug-in model can have two variable selection steps: one for $\boldsymbol{X}_f$ and then another for $\boldsymbol{X}_r$ so there is no restriction for the final model.

Figure 8.1 illustrates the bias-variance trade-off followed by this plug-in model. We logically observe that the plug-in model gives better results than OLS in the cases with enough correlations to have problem when using whole $\boldsymbol{X}$ (ill-conditioned). The plug-in model gives better results than the marginal when there are 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 and enlarges the range of cases where we can beat classical OLS.

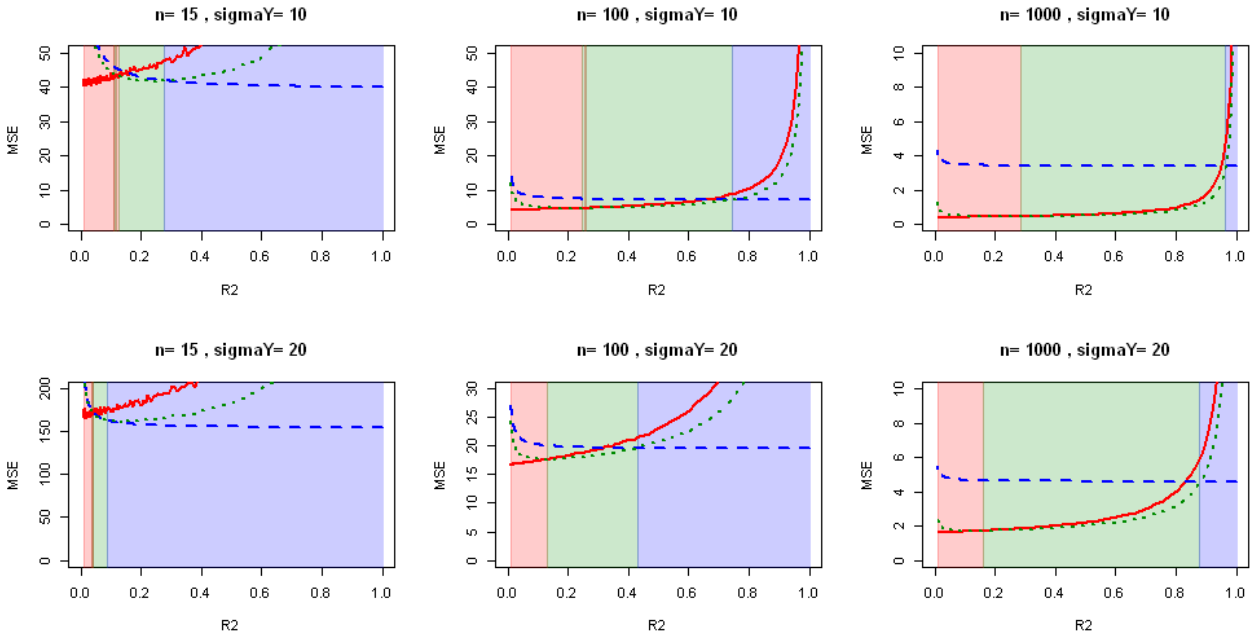


Figure 8.1: MSE on $\hat{\boldsymbol{\beta}}$ of OLS (plain red) and `CorReg` marginal (blue dashed) and `CorReg` plug-in (green dotted) estimators for varying R^2 of the sub-regression, n and σ_Y. Results obtained on the running example with $d = 5$ covariates.

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 in the numerical results (Section 8.4) that it is not always the better choice because even if the plug-in estimator can always be consistent (each covariate can be used) contrary to the marginal model, cumulated variances are a real problem and the marginal model is often better in prediction.

Remarks:

- $\boldsymbol{\beta}_r$ can be interpreted as the proper effect of $\boldsymbol{X}_r$ on $\boldsymbol{Y}$ in that it is the effect of the part of $\boldsymbol{X}_r$ that is independent of other covariates. Then if $\boldsymbol{X}_r$ is correlated to $\boldsymbol{Y}$ only through its correlation with $\boldsymbol{X}_f$ this sequential estimation will point it out and give a parsimonious model ($\hat{\boldsymbol{\beta}}_r = \boldsymbol{0}$).
- If $\boldsymbol{Y}$ depends only on $\boldsymbol{\varepsilon}$ (latent variables) then the plug-in model written as in equation (8.6) will show it. Moreover, we have an estimator of $\boldsymbol{\varepsilon}$ and we know that it is the proper of effect of $\boldsymbol{X}_r$ on $\boldsymbol{Y}$ and that it is independent of $\boldsymbol{X}_f$. Thus we

have an idea of the values and meaning of $\boldsymbol{\varepsilon}$. It can help to name the latent variable and to add it in the dataset if possible.

8.3 Model selection consistency of LASSO improved

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 better to find the true relevant covariates. Here $d = 3$ and $n = 1\,000$.
We define $\boldsymbol{X}^1, \boldsymbol{X}^2, \boldsymbol{\varepsilon}_Y, \boldsymbol{\varepsilon}_1 \quad i.i.d. \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ and then

$$\begin{aligned} \boldsymbol{X}^3 &= \frac{2}{3}\boldsymbol{X}^1 + \frac{2}{3}\boldsymbol{X}^2 + \frac{1}{3}\boldsymbol{\varepsilon}_1 \text{ and} \\ \boldsymbol{Y} &= 2\boldsymbol{X}^1 + 3\boldsymbol{X}^2 + \boldsymbol{\varepsilon}_Y. \end{aligned}$$

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 do not find the true structure but a permuted one so we both look at the results obtained with the true $\boldsymbol{S} = ((3), (\{1, 2\}))$ (but $\hat{\boldsymbol{\alpha}}$ is used) and with the structure found by the Markov chain after a few seconds.

True $\boldsymbol{S}$ was found 991 times on 1 000 tries.

	Classical LASSO	`CorReg` marginal + LASSO	`CorReg` full plug-in + LASSO
True $\boldsymbol{S}$	1.003303 (0.046)	**1.002273** (0.046)	**1.002812** (0.046)
$\hat{\boldsymbol{S}}$	1.003303 (0.046)	1.017622 (0.17)	**1.002812** (0.046)

Table 8.1: MSE observed on a validation sample (1 000 individuals) and their standard deviation (between brackets).

We observe as we hoped that our marginal model is better when using true $\boldsymbol{S}$ (coercing real zeros) and that marginal with $\hat{\boldsymbol{S}}$ is penalised (coercing wrong coefficients to be zeros when true $\boldsymbol{S}$ is not found). We also see that the plug-in model stays better than the classical one with the true $\boldsymbol{S}$ and corrects enough the marginal model to be better than the classical LASSO when using $\hat{\boldsymbol{S}}$.

But sadly, improvements in MSE are very small and the MSE in Table 8.1 are not significantly distinct (Student's t-Tests).

We look at the consistency (Table 8.2), that is the real stake of these numerical results:

	Classical LASSO	`CorReg` marginal + LASSO	`CorReg` full plug-in + LASSO
True $\boldsymbol{S}$	0	**1000**	**835**
$\hat{\boldsymbol{S}}$	0	**991**	**829**

Table 8.2: Number of consistent models found ($\boldsymbol{Y}$ depending on $\boldsymbol{X}^1, \boldsymbol{X}^2$ and only them) on 1 000 tries.

It is clear that the plug-in model significantly improves the consistency of the model estimated on $\boldsymbol{S}$ when compared to the classical LASSO. Both models have the choice to keep or not each covariate but only the plug-in model find sometimes (and in most of the cases) the true set of relevant covariates. Model using the true structure cannot 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 do not only improve this situation, we give consistent models in most of the cases.

8.4 Numerical results

We test the plug-in model with datasets generated the same way as for section 6.3.

8.4.1 Y depends on all variables in X

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

Ordinary Least Squares We observe for OLS (Figure 8.2) 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 found a model with more than n coefficients when each estimation step computes less than n coefficients. It means that we estimate more coefficients than the classical OLS and keep a smaller variance so the plug-in model can also be an alternative to the complete model. It is interesting to see that OLS combined with the plug-in model is a sort of sequential estimation that allows to estimate more than n coefficients.

Other methods: Combined with variable selection methods (Figures 8.3 to 8.5) 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 8.6) leads to the same conclusion.

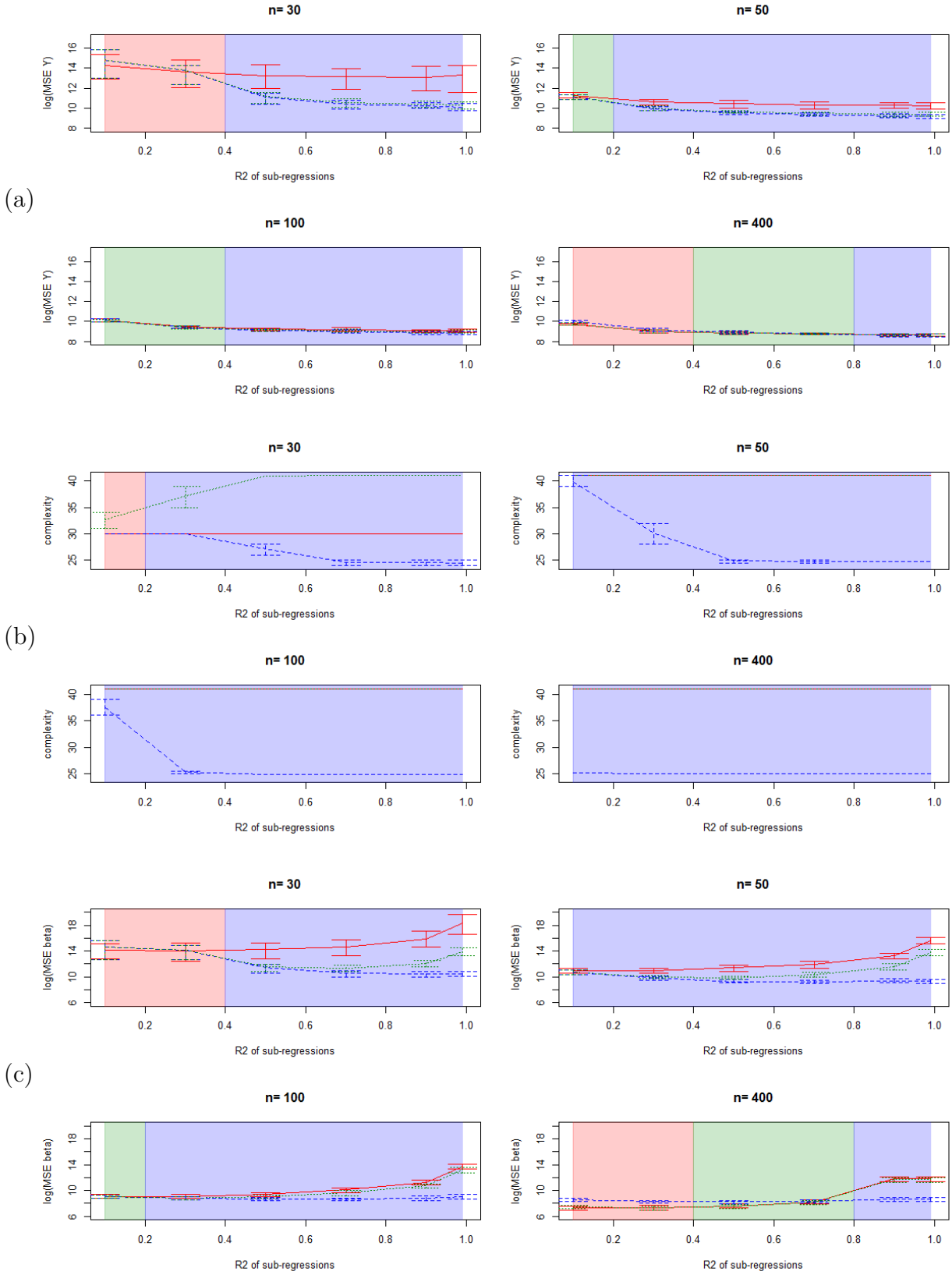


Figure 8.2: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

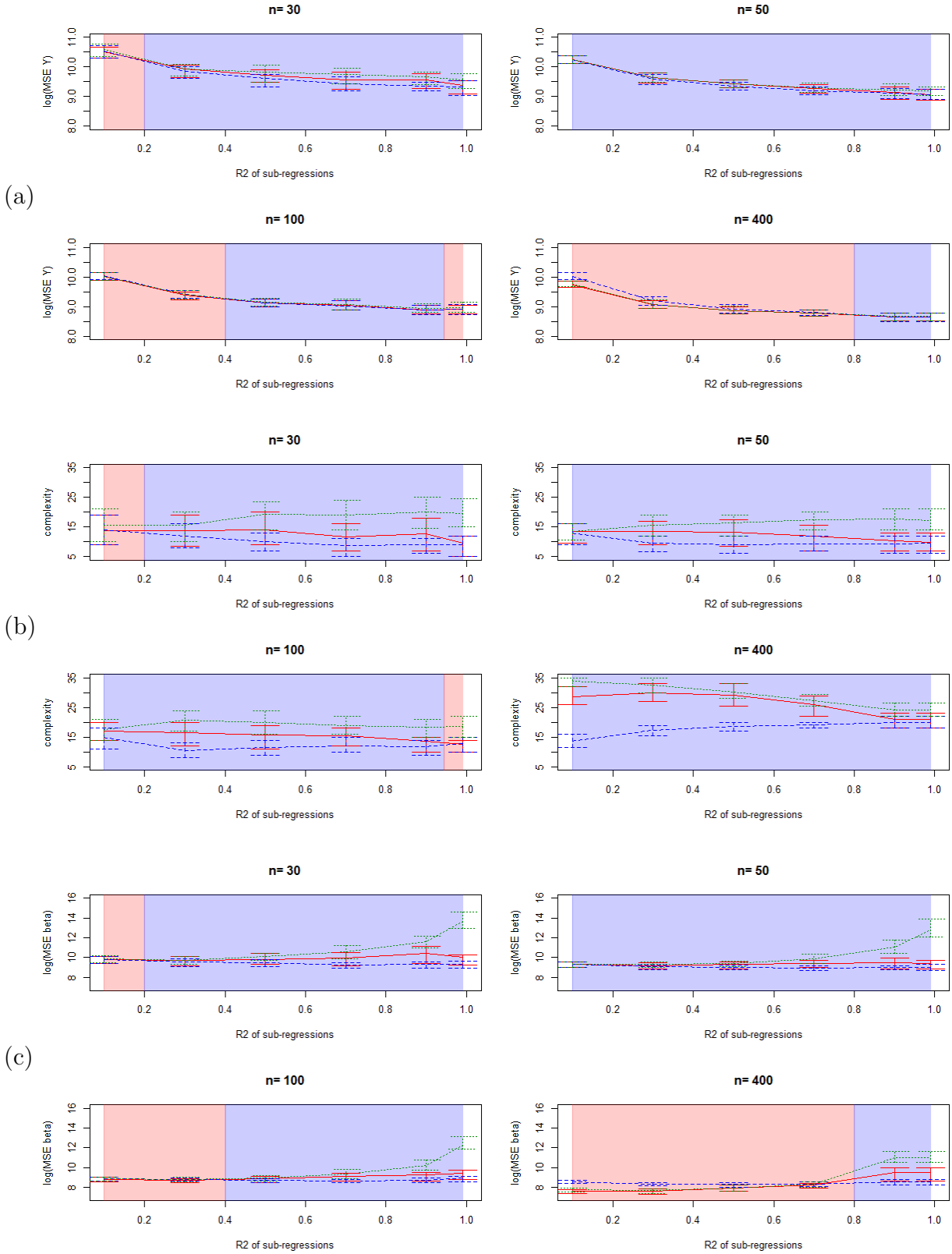


Figure 8.3: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

Elasticnet when Y depends on all variables in X

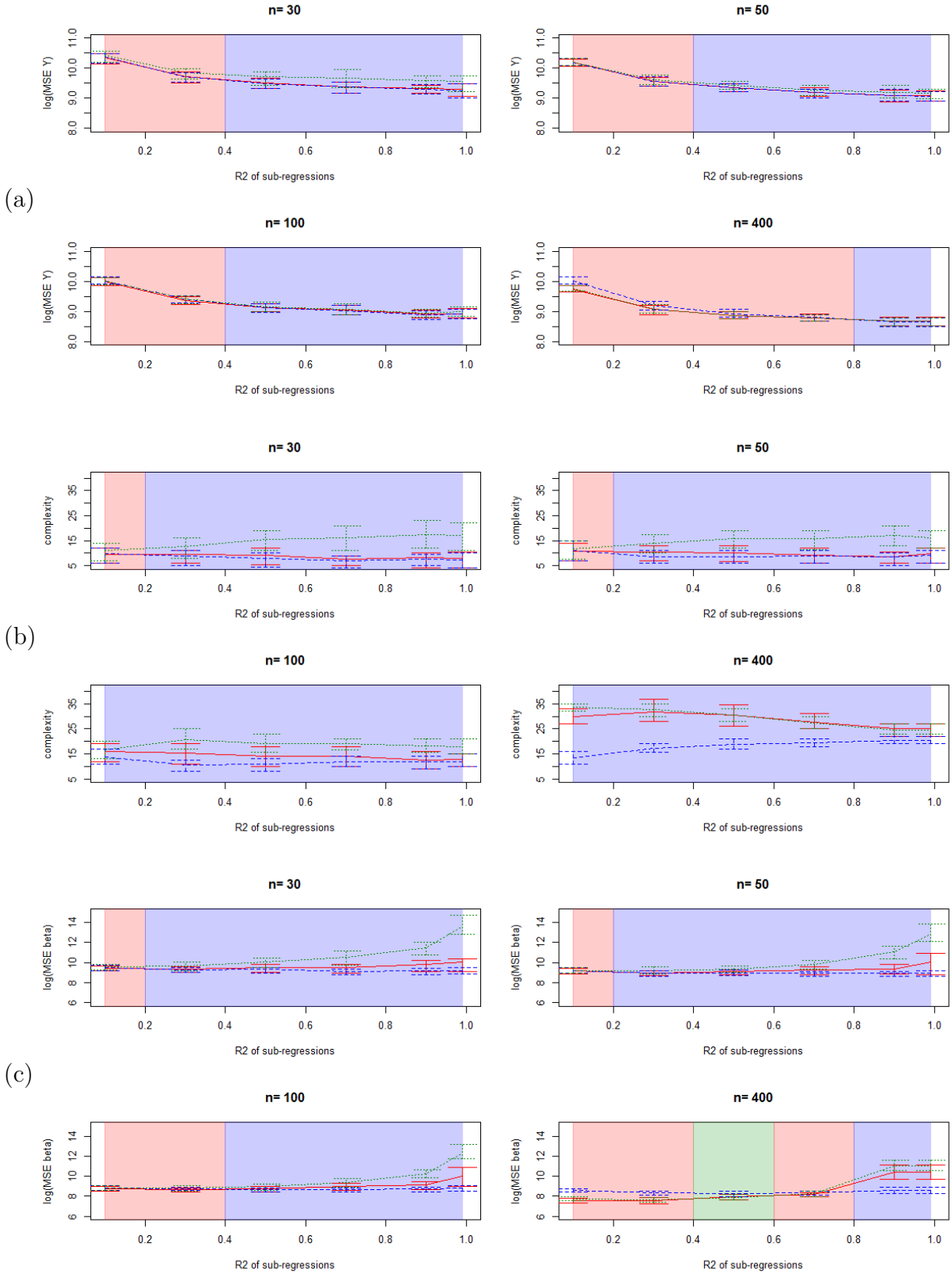


Figure 8.4: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

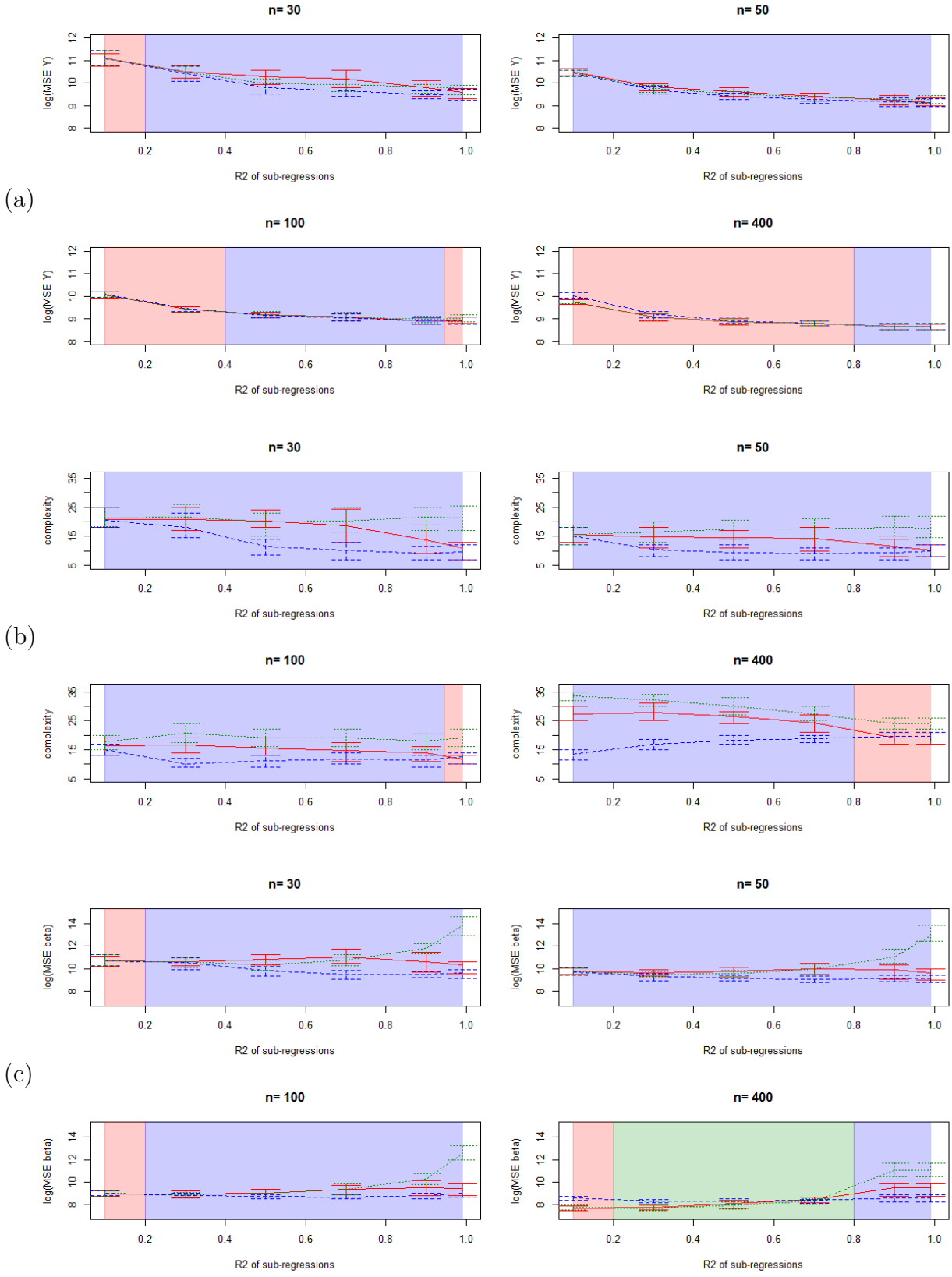


Figure 8.5: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

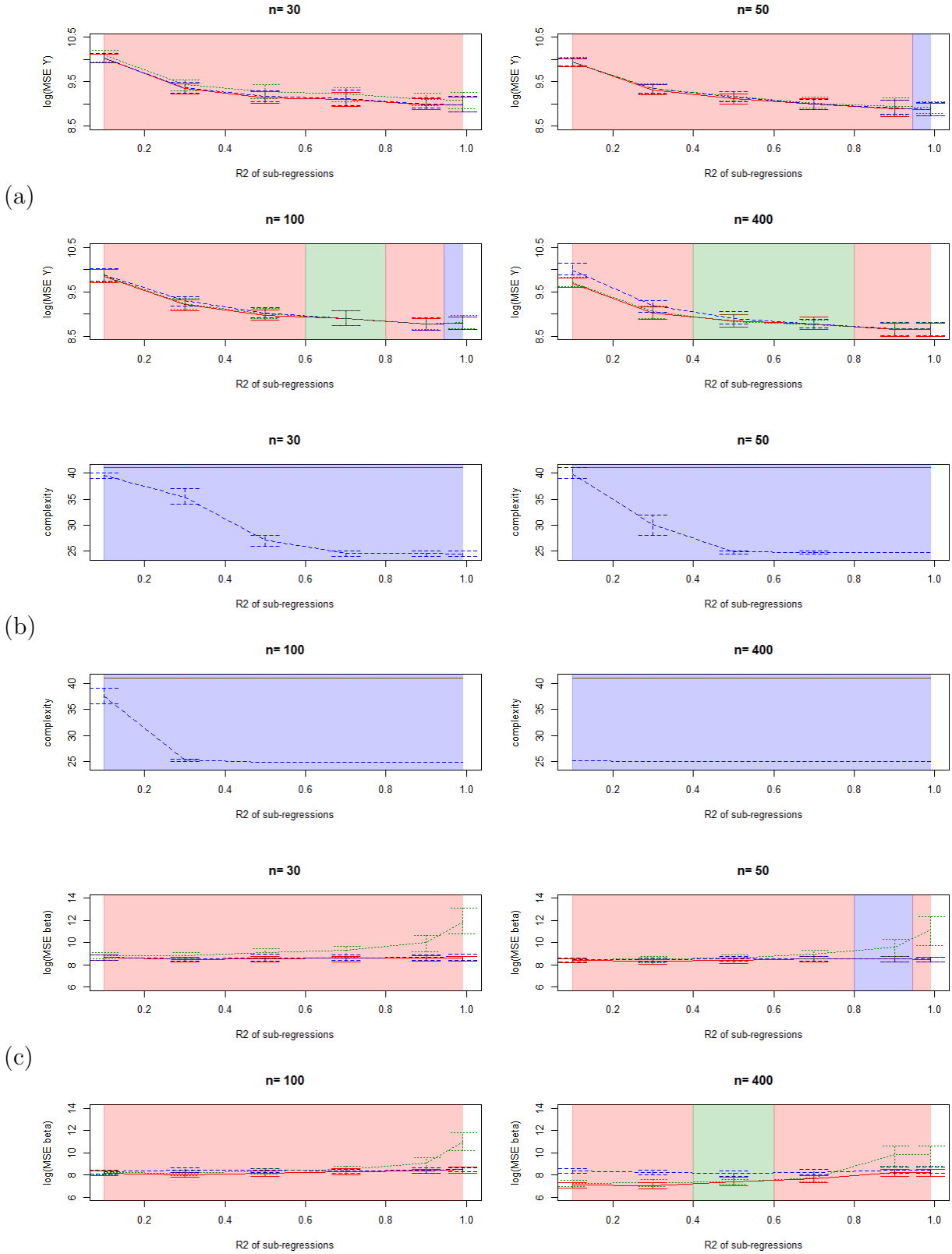


Figure 8.6: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

8.4.2 Y depends only on covariates in X_f

We look then at the case where $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X_f}$.

Ordinary Least Squares: Figure 8.7 shows that the plug-in model stays better than OLS even when $\boldsymbol{Y}$ depends only on covariates in $\boldsymbol{X_f}$. 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 8.8 to 8.10)are not improved by the plug-in model and the marginal model remains the best. But ridge regression (Figure 8.11) is much similar to OLS and then the plug-in model gives good results, even if the marginal model is sufficient.

8.4.3 Y depends only on covariates in X_r

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

Better but not sufficient: Concerning OLS (Figure 8.12) 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 8.13 to 8.15) even if it is not sufficient to improve the complete model. This phenomenon is still observed with the ridge regression (Figure 8.16) with more efficiency but the complete model stays better.

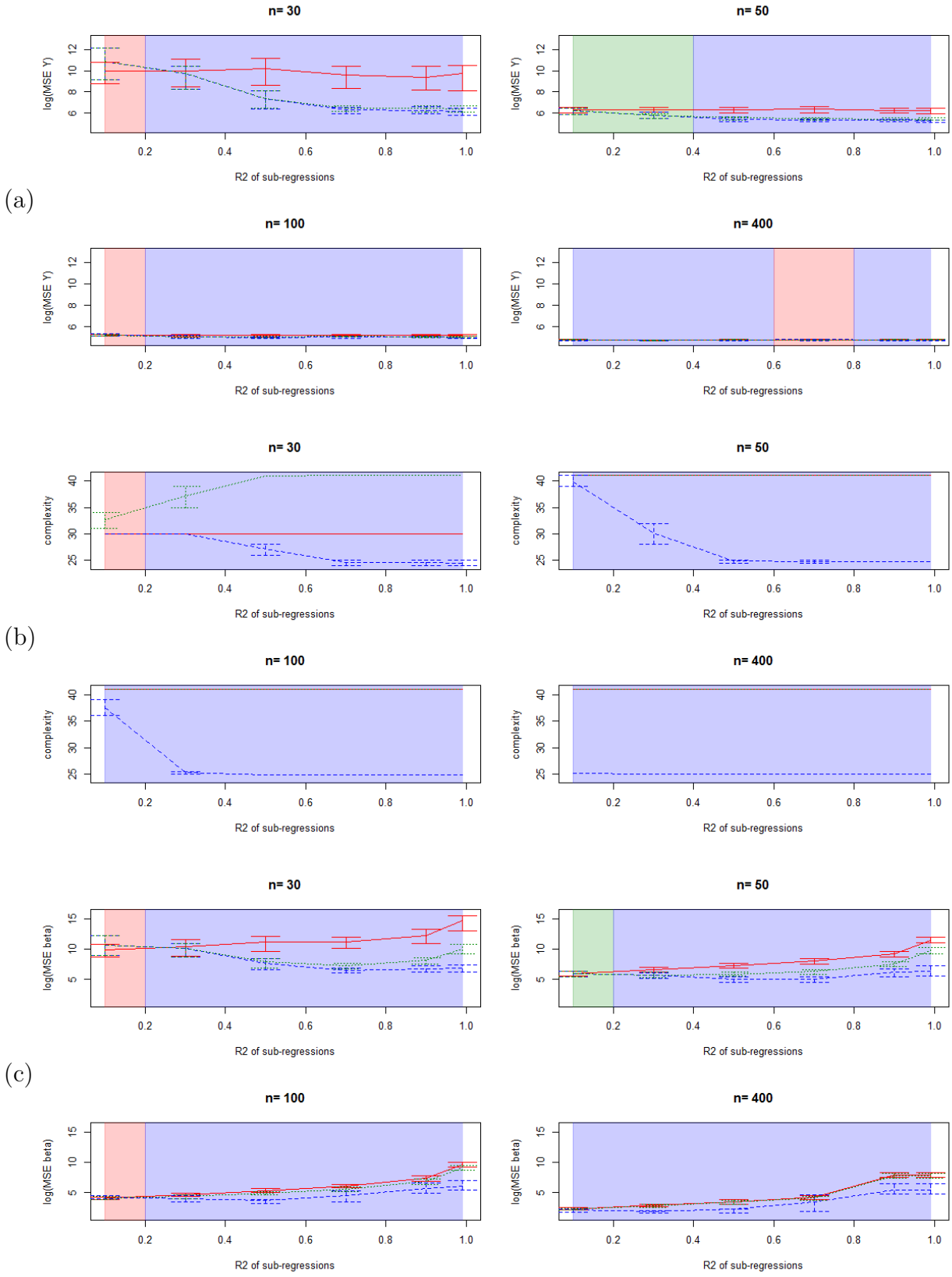


Figure 8.7: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

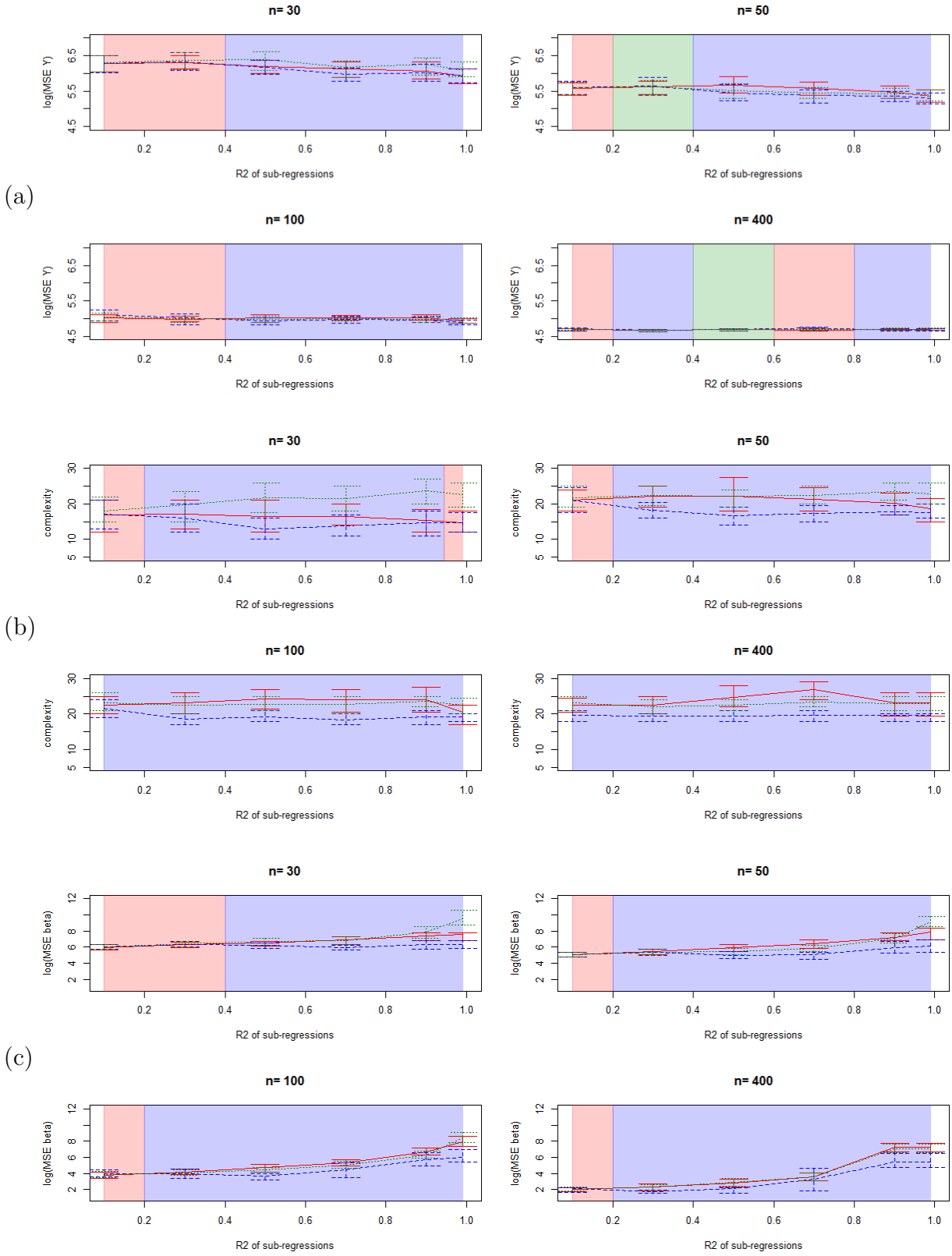


Figure 8.8: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

Elasticnet when Y depends only on covariates in X_f

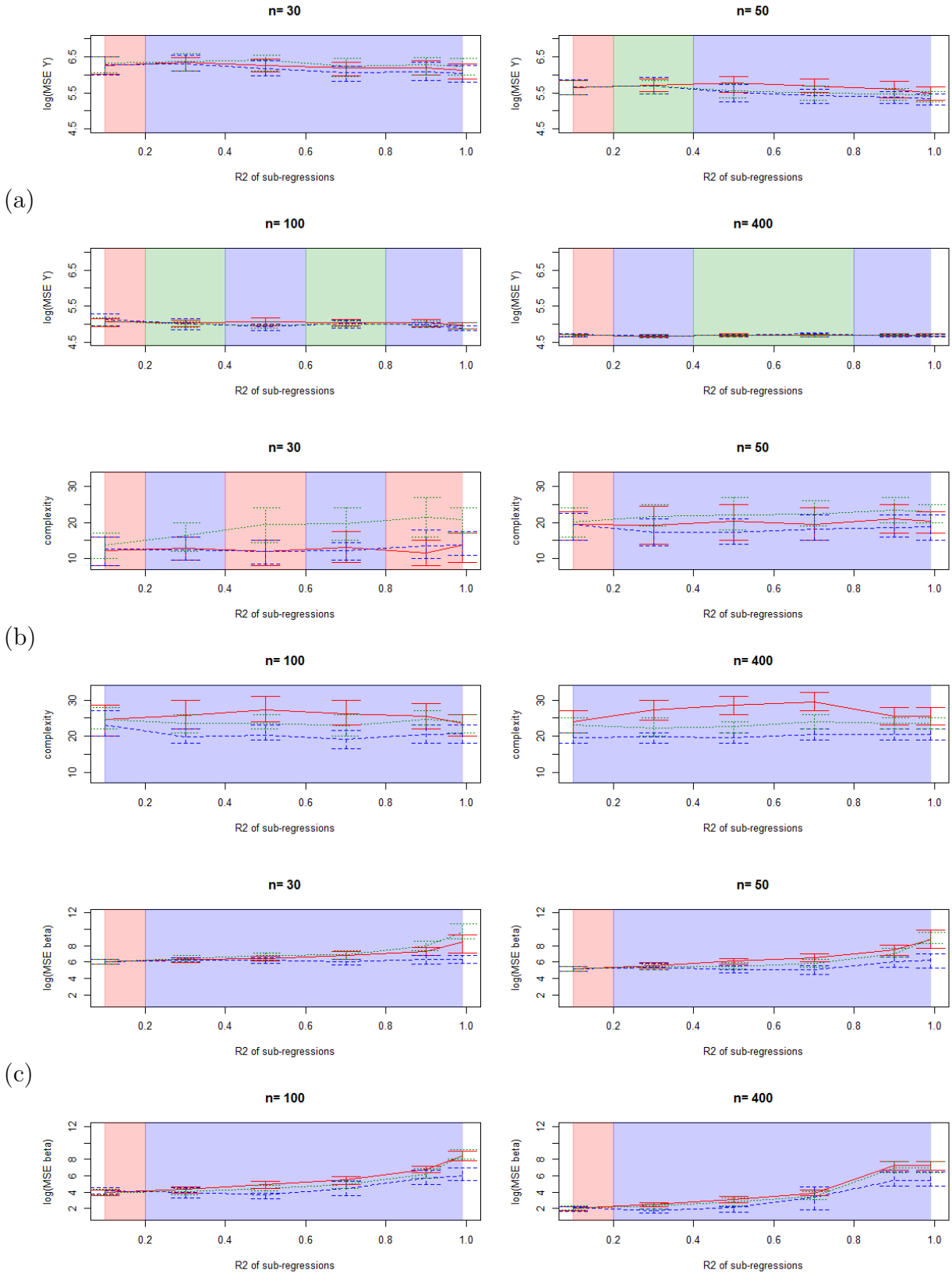


Figure 8.9: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

Stepwise when Y depends only on covariates in X_f

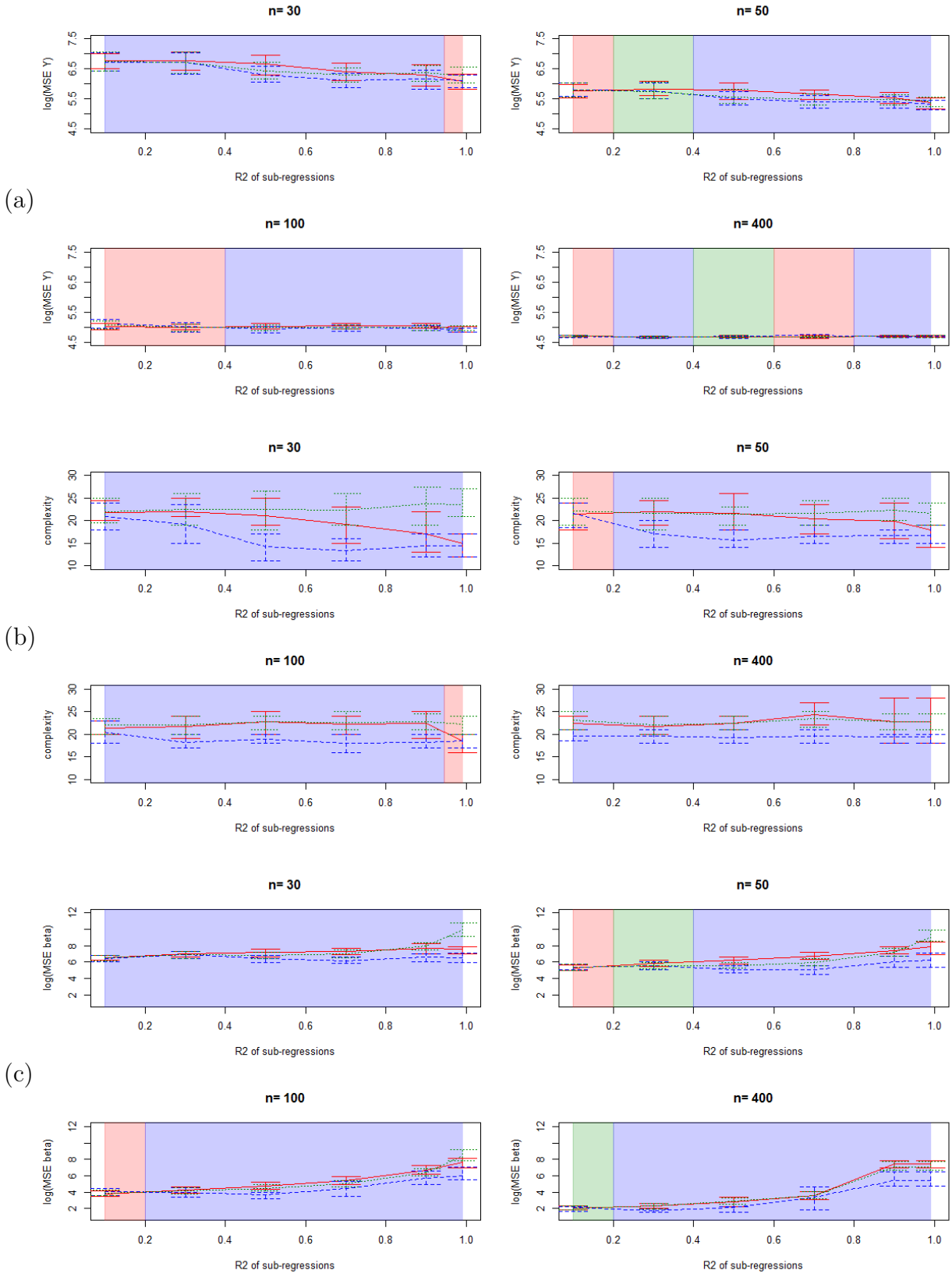


Figure 8.10: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

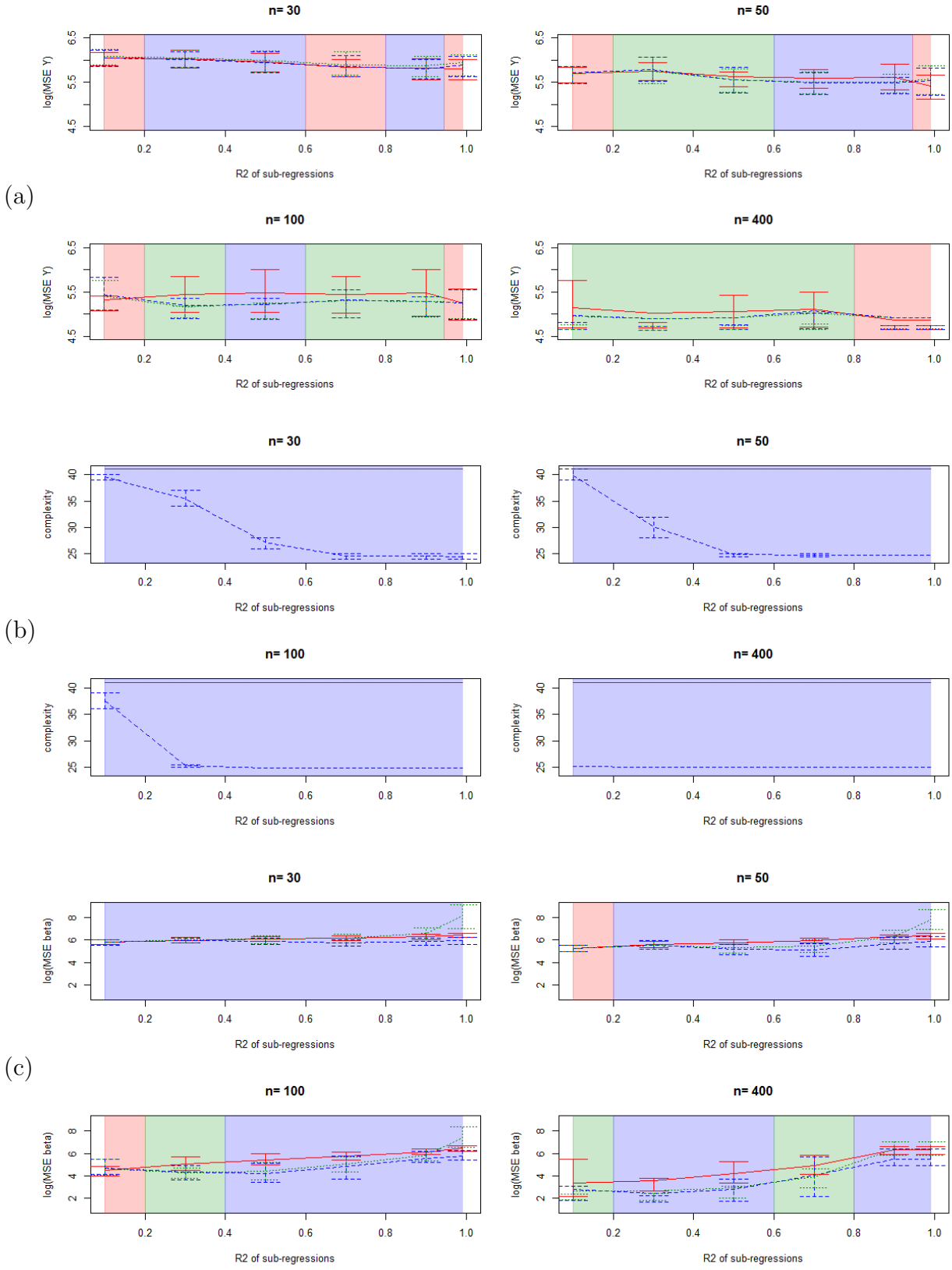


Figure 8.11: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

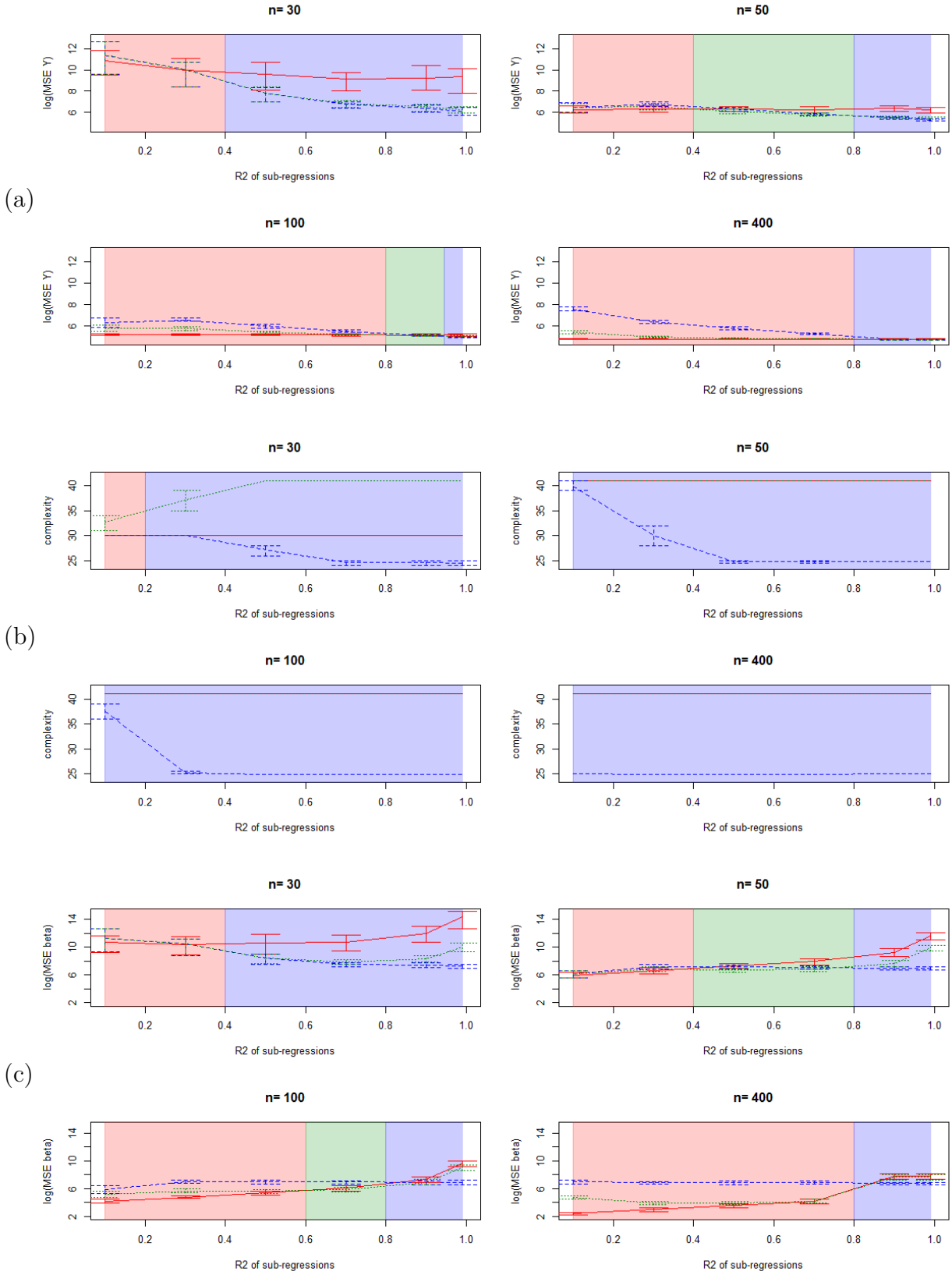


Figure 8.12: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

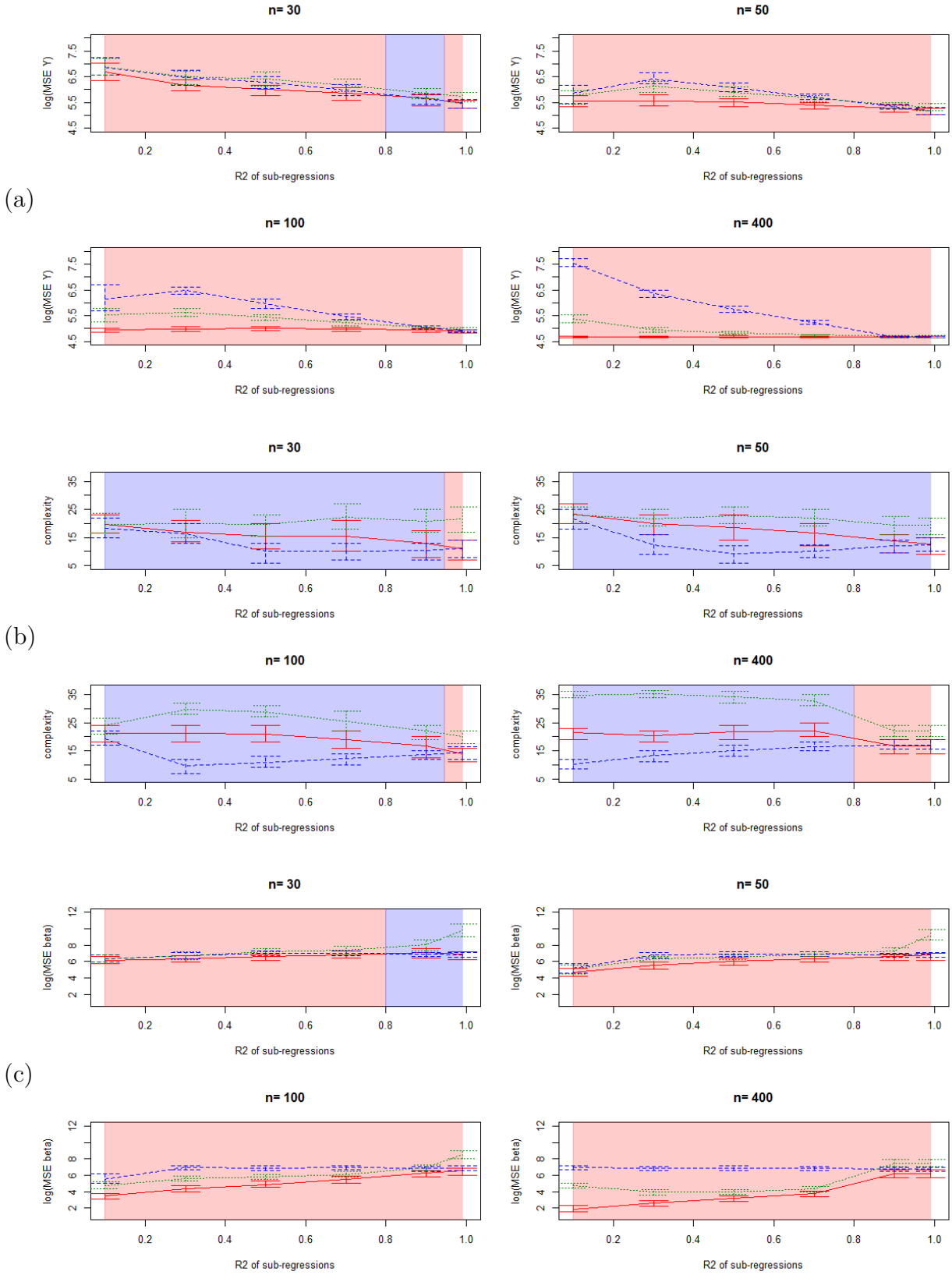


Figure 8.13: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

Elasticnet when Y depends only on covariates in X_r

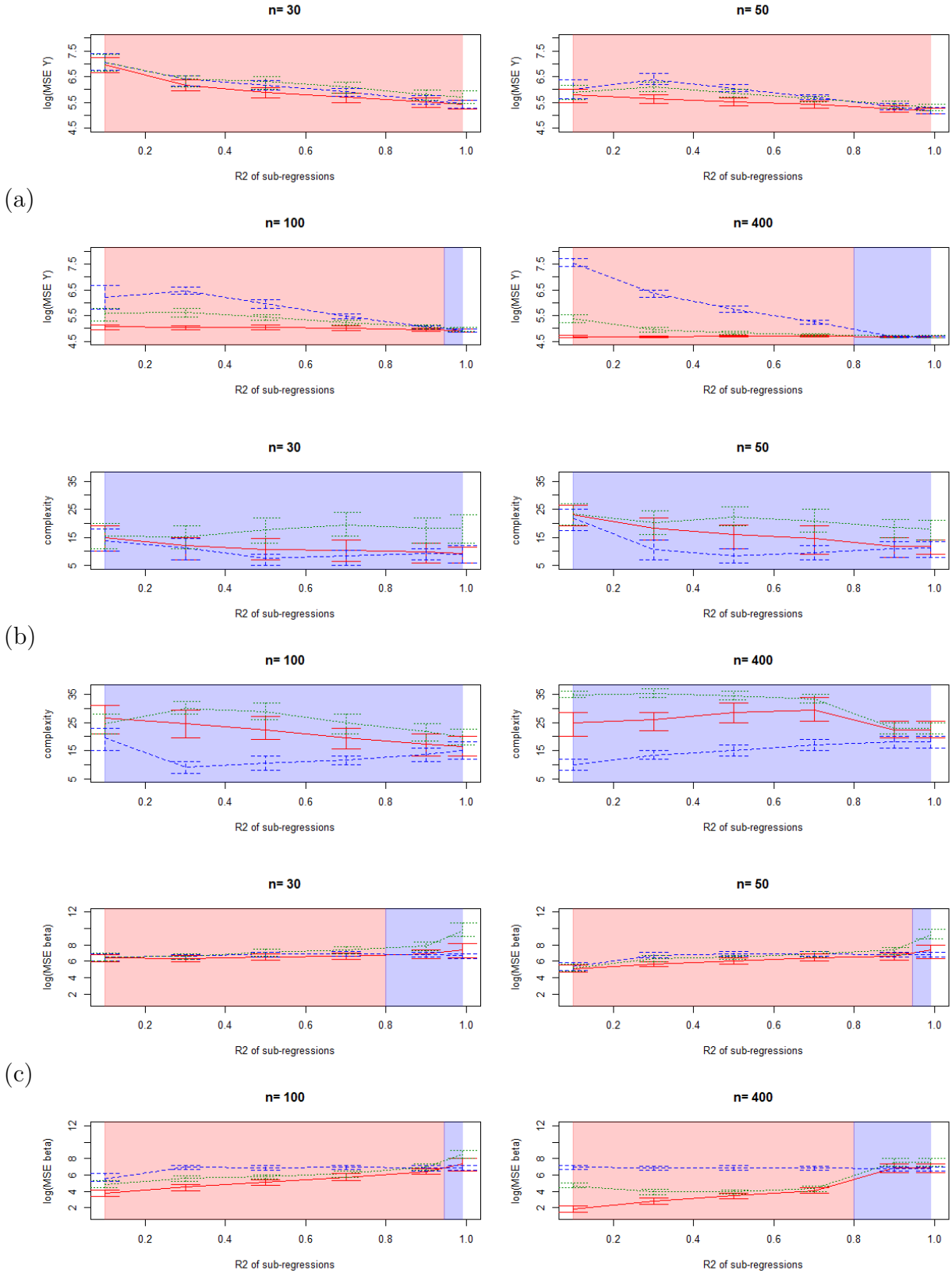


Figure 8.14: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

Stepwise when Y depends only on covariates in X_r

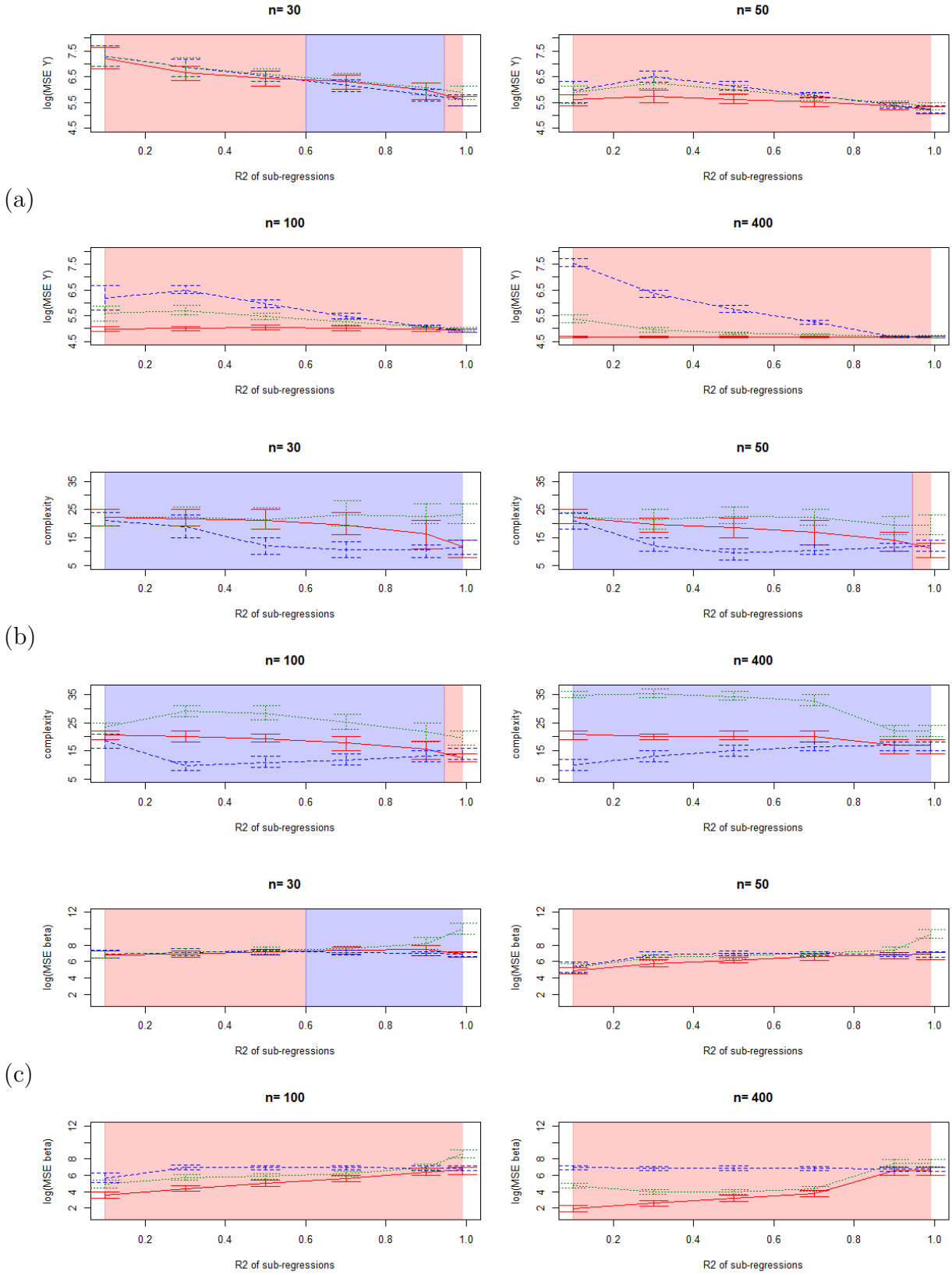


Figure 8.15: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

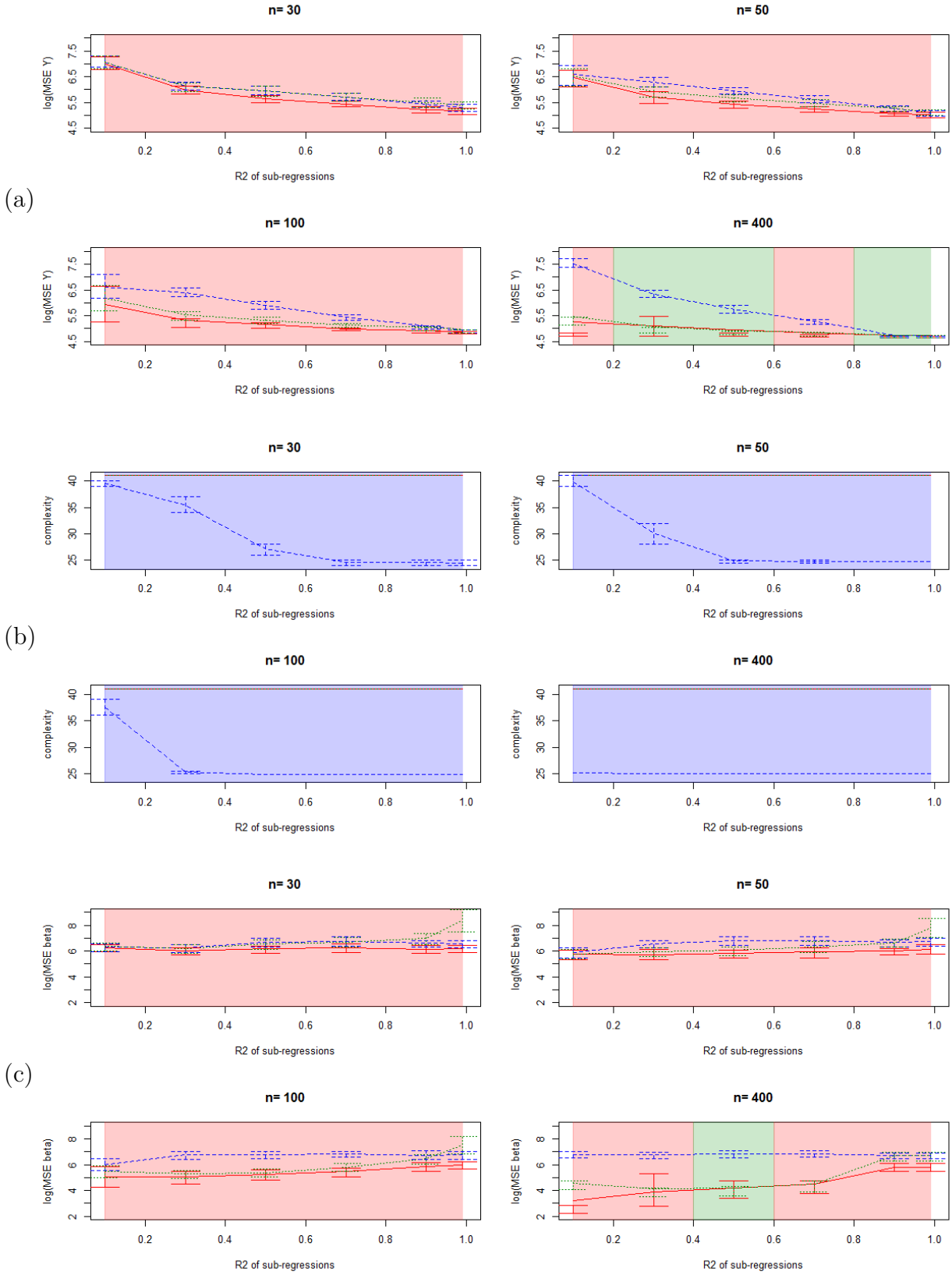


Figure 8.16: Comparison of the MSE on $\hat{\boldsymbol{Y}}$ (a), complexities (b) and MSE on $\hat{\boldsymbol{\beta}}$ (c), plain red=classical (complete) model, dashed blue=marginal model, dotted green=plug-in model.

8.5 Conclusion

The plug-in model sounds good when described theoretically and figure 8.1 makes us hope to obtain good results with it. But the reality is that the plug-in model (by definition) relies on other estimators so it has a slow convergence speed.
Moreover, if $\hat{\boldsymbol{S}} \neq \boldsymbol{S}$ but $\hat{J}_r = J_r$, then the marginal model is not impacted whereas the plug-in model depends on both $\hat{\boldsymbol{\beta}}_f^*$ and $\hat{\boldsymbol{\alpha}}$. 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 $\boldsymbol{S}$. So we recommend to compute both complete, marginal and plug-in model and to compare the results in a second time.

Chapter 9

Using the full generative model to manage missing values

Abstract: The full generative model defined on $\boldsymbol{X}$ with $\boldsymbol{S}$ gives the conditional distribution of missing values. Thus we are able to manage missing values. Because the joint distribution of $\boldsymbol{X}$ can be very complex (Gaussian mixture with a huge number of components), we propose to use a SEM algorithm with multiple imputations by Gibbs sampling. Computation of the BIC in the MCMC will also rely on Gibbs sampling. To resume: we propose to manage missing values by multiple imputations based on the structure $\boldsymbol{S}$ and on the marginal Gaussian mixtures estimated independently on each covariate in $\boldsymbol{X}_f$.

9.1 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 depend 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 do 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 $\boldsymbol{X}$ or based on $\boldsymbol{X}$ and $\boldsymbol{Y}$. These methods are often used with weighted least-squares to minimize the influence of imputed values.

4. Other methods are 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 $\boldsymbol{X}$ with explicit dependencies within the covariates:

$$\mathbb{P}(\boldsymbol{X}|\boldsymbol{S}) = \mathbb{P}(\boldsymbol{X}_r|\boldsymbol{X}_f, \boldsymbol{S})\mathbb{P}(\boldsymbol{X}_f|\boldsymbol{S}).$$

So when a value is missing, we know its distribution but in addition, 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 we have defined on $\boldsymbol{X}$.

SelvarclustMV: We have compared our method to `Selvarclust` in Section 6.2.1. But Maugis has developed another software [Maugis-Rabusseau 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: $\forall j \in \{1,\dots,d_r\}, \forall k \in J_p^j, \forall 1 \leq i \leq n, x_{i,k}$ is observed). 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 $\boldsymbol{S}$ and $\boldsymbol{\alpha}^*$ was inconclusive (see section 6.2.1) 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 $\boldsymbol{\alpha}$ by Maximum Likelihood as `SelvarclustMV` does but without supposing that any covariate is fully observed.

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

$\bar{\boldsymbol{\alpha}}$ is the $d \times d$ matrix of the sub-regression coefficients with $\bar{\alpha}_{i,j}$ the coefficients associated to $\boldsymbol{X}^i$ in the sub-regression explaining $\boldsymbol{X}^j$ and zeros where there is no sub-regression, not to be confused with the $\boldsymbol{\alpha}^*$ previously defined. $\bar{\boldsymbol{\alpha}}$ is $\boldsymbol{\alpha}^*$ completed with rows and columns of zeros. As in previous part, we note Φ the Gaussian density function.

9.2 Estimation of S with missing values

We know that $\boldsymbol{X}$ follows a Gaussian mixture because of hypothesis 4 in Section 5.2 page 62 (*i.i.d.* individuals, vectors of orthogonal Gaussian mixtures $\boldsymbol{X}_f$ and linear combinations of these Gaussian mixtures and some Gaussian for $\boldsymbol{X}_r$) with K components. We stack together all the mixture parameters in $\boldsymbol{\Theta}$ (that depends on $\boldsymbol{\alpha}$). The number of component k can be huge (combinations of all the components of the covariates in $\boldsymbol{X}_f$). In fact we have

$$K = \prod_{j \in J_f} K_j$$

where K_j is the number of components of the Gaussian mixture followed by $\boldsymbol{X}_j$ as defined in Hypothesis 4 page 62. It is clear that K can really explode even if some components may be identical (but it happens with zero probability).

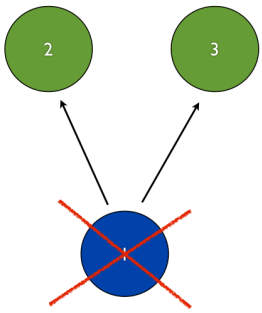


Figure 9.1: Pattern of missing values that does not allow to use equation (9.2): one missing value for a covariate that is common to 2 observed response covariates.

9.2.1 Marginal (observed) likelihood

The first thing we do with $\boldsymbol{X}$ in the CorReg process is to estimate $\boldsymbol{S}$. It is done by comparison of the BIC that rely on the likelihood. Because covariates in $\boldsymbol{X}_f$ 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 likelihoods 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 sub-regressions 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_*.

Observed likelihood: When missing values occur, we restrict the likelihood to the observed likelihood, that is the likelihood of the known values. All individuals are supposed *i.i.d.* in the following. We let $\boldsymbol{\alpha}$ appear to recall that $\boldsymbol{\Theta}$ depends on it.

$$\begin{aligned} L(\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S};\boldsymbol{X}_O) &= f(\boldsymbol{X}_O;\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S}) \\ &= \prod_{i=1}^{n} f(\boldsymbol{X}_{i,O}^{J_r}|\boldsymbol{X}_{i,O}^{J_f};\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S}) f(\boldsymbol{X}_{i,O}^{J_f};\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S}) \\ &= \prod_{i=1}^{n} f(\boldsymbol{X}_{i,O}^{J_r}|\boldsymbol{X}_{i,O}^{J_f};\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S}) \prod_{\substack{j\in J_f \\ \boldsymbol{M}_{i,j}=0}} f(x_{i,j};\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S}). \end{aligned} \tag{9.1}$$

And if there is no missing values in covariates that regress several observed response covariates (see Figure 9.1) we can continue the decomposition of equation (9.1):

$$L(\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S};\boldsymbol{X}_O) = \prod_{i=1}^{n} \prod_{\substack{j=1 \\ \boldsymbol{M}_{i,J_r^j}=0}}^{d_r} f(x_{i,J_r^j}|\boldsymbol{X}_{i,O}^{J_p^j};\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S}) \prod_{\substack{j\in J_f \\ \boldsymbol{M}_{i,j}=0}} f(x_{i,j};\boldsymbol{\alpha},\boldsymbol{\Theta},\boldsymbol{S}). \tag{9.2}$$

If each sub-regression is a distinct connected component then we can always use equation (9.2) but it is not a necessary condition.

For the general case we need to manage the dependencies implied by missing values in common covariates in the J_p^j. Then we use the global Gaussian mixture on $\boldsymbol{X}_O$ whose parameters depends on those of $\boldsymbol{X}$. Each component is Gaussian so conditional distributions are explicit. But we have seen that the number of components can be huge so in practice we won't compute the likelihood directly. We will work instead with multiple imputations (Section 9.3.2) to estimate $\boldsymbol{\alpha}$ and the likelihood for BIC (Section 9.3.2).

9.2.2 Weighted penalty for BIC

Once defined a way to compute the likelihood (Section 9.3.2), other questions will 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)? For each parameter we have to estimate, the number of individuals that we can use is not the same according to the position of the missing values. Then penalization of the complexity of a model is not obvious.

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. We define

$$c_j = \frac{1}{n}\sum_{i=1}^{n} c_{i,j},$$

where $c_{i,j}$ is the number of parameters to estimate in $\mathbb{P}(x_{i,j}|\boldsymbol{X}_i \setminus \boldsymbol{X}_i^j)$. Then we have

$$\begin{aligned} -2\log\mathbb{P}(\boldsymbol{X}|\boldsymbol{S}) &\approx \text{BIC} = -2\ln\left(L(\boldsymbol{\Theta},\boldsymbol{S};\boldsymbol{X})\right) + |\boldsymbol{\Theta}|\ln(n) \\ &= -2\ln\left(L(\boldsymbol{\Theta},\boldsymbol{S};\boldsymbol{X})\right) + (\sum_{j=1}^{d} c_j)\ln(n). \end{aligned}$$

Thus if a structure is only touched by one missing value the penalty will be smaller than if the same structure had 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 $\log(d)$.

9.3 Estimation of $\boldsymbol{\alpha}$ and the observed likelihood

If we do not have an estimate of $\boldsymbol{\alpha}$ we can just take the values that maximize the likelihood. Sometimes the maximization of a likelihood is too complex to be solved analytically. In such a case, we can use algorithms of Expectation-Maximization (EM [McLachlan and Krishnan, 2007]) family [McLachlan and Krishnan, 2007]. This kind of algorithm allows to manage missing values [Dempster et al., 1977] but faces local extrema problems so it is recommended to make multiple initializations and runs 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]. In the following we use the Stochastic EM.

9.3.1 Stochastic EM

The observed likelihood depicted above (equation (9.1)) depends on the $\boldsymbol{\alpha}_j$'s which were formerly estimated by OLS when there were no missing values. But when missing values occur in a sub-regression we need another solution.

We use a Stochastic Expectation Maximization (SEM [Celeux and Diebolt, 1986]) algorithm to estimate $\boldsymbol{\alpha}$ because missing values do not allow to use OLS and the Expectation-Maximization (EM) algorithm necessitates to first write the observed likelihood whose number of component can explode (as we have seen Section 9.2) so it would be difficult to compute.

Another method would be to estimate the $\boldsymbol{\alpha}_j$ with OLS applied on sub-matrix of $(\boldsymbol{X}^{J_r^j}, \boldsymbol{X}^{J_p^j})$ 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.

Here 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 of $\boldsymbol{X}$) is generated (component of the marginal Gaussian mixture followed by $x_{i,j}$ and defined by Hypothesis 4 page 62) with $1 \leq i \leq n$ and $1 \leq j \leq d$.

The Stochastic EM algorithm is an iterative procedure that starts with an initialization and then alternates SE steps (imputation of the missing values and Z) and the M step that maximizes the likelihood of the completed dataset on the parameter $\boldsymbol{\alpha}$.

Initialization: We start with some imputation (for example by the mean) for each missing value (done only once for the MCMC) to get $\boldsymbol{X}_M^{(0)}$. $\boldsymbol{\alpha}^{(0)}$ can be initialized by complete-case method (sparse structure) or using imputed values in $\boldsymbol{X}$ and then OLS. And the at iteration (h):

SE step: We generate the missing values $\hat{\boldsymbol{X}}_M^{(h)}$ according to $\mathbb{P}(\boldsymbol{X}_M|\boldsymbol{X}_O; \boldsymbol{\alpha}^{(h)}, \boldsymbol{\Theta}^{(h)}, \boldsymbol{S})$, that is stochastic imputation. It can be done for example by Gibbs sampling, as we propose in Section 9.3.2.

M step: We estimate

$$\boldsymbol{\alpha}^{(h+1)} = \text{argmax}_{\boldsymbol{\alpha}}\, E\left[\ln L(\boldsymbol{\alpha}, \boldsymbol{S}, \boldsymbol{\Theta}; \boldsymbol{X}_O, \boldsymbol{X}_M^{(h)})\right]$$

where $\boldsymbol{\Theta}$ depends on $\boldsymbol{\alpha}$ and on the mixture parameters of the marginal distributions of $\boldsymbol{X}_f$ (estimated once by `Rmixmod`). We can use the same method as the one for classical case without missing values (OLS, SUR, *etc.*). Then we deduce $\boldsymbol{\Theta}^{(h+1)}$

We continue during m_1 iterations. Then we make m_2 iterations and take $\hat{\boldsymbol{\alpha}}$ as the mean of these m_2 last iterations.

The stochastic imputation made at the SE step is computed by Gibbs sampling in the following.

9.3.2 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 $\boldsymbol{X}^{(0)}$ and then for each iteration (q) and successively each variable $x_j^{(q+1)}$

to draw, it draws from $\mathbb{P}(x_j|x_1^{(q+1)},\ldots,x_{j-1}^{(q+1)},x_{j+1}^{(q)},\ldots,x_d^{(q)})$ using the most recent drawn values each time.

Gibbs sampling method can be used to generate the missing values at each h^{th} SE step of the previous Stochastic EM algorithm.

Initialisation: $\boldsymbol{X}_M^{(0,h)}$ are imputed by the marginal means or drawn independently from the univariate distribution estimated for the MCMC (by `Rmixmod` for example, following hypothesis 4 page 62). All the $Z_{i,j}^{(0,h)}$ are then randomly set based on the $t_{i,j,k}^{(0,h)}$ as explained below, replacing $(q+1)$ by 0 (we just need those of $\boldsymbol{X}_f^{(0,h)}$).

Iteration (q+1): At each iteration of the Gibbs sampler we first make successive imputations on the missing values. We note $\boldsymbol{X}_{\bar{ij}}^{(q+1,h)} = (x_{i,1}^{(q+1,h)},\ldots,x_{i,j-1}^{(q+1,h)},x_{i,j+1}^{(q,h)},\ldots,x_{i,d}^{(q,h)})$ and $Z_i^{(q,h)} = \{Z_{i,j}^{(q,h)}|1\leq j\leq d, j\in J_f\}$. Then:

- Regressed covariates: $\forall 1\leq i\leq n, \forall j\in\{1,\ldots,d_r\}, \boldsymbol{M}_{i,J_r^j}=1$: $x_{i,J_r^j}^{(q+1,h)}$ is generated using the corresponding sub-regression, that gives the Gaussian

$$\mathbb{P}(x_{i,J_r^j}|\boldsymbol{X}_{i\bar{J}_r^j}^{(q+1,h)}, Z_i^{(q,h)};\boldsymbol{\alpha}^{(h)},\boldsymbol{\Theta}^{(h)},\boldsymbol{S}) \;=\; \Phi(x_{i,J_r^j};(\boldsymbol{X}_i^{J_p^j})^{(q+1,h)}\boldsymbol{\alpha}_j^{(h)},(\sigma_j^2)^{(h)}).$$

- And for regressors: $\forall(i,j)\in\{1,\ldots,n\}\times J_f, \boldsymbol{M}_{i,j}=1$: $x_{i,j}^{(q+1,h)}$ is generated according to the Gaussian $\mathbb{P}(x_{i,j}|\boldsymbol{X}_{\bar{ij}}^{(q+1,h)}, Z^{(q,h)};\boldsymbol{\alpha}^{(h)},\boldsymbol{\Theta}^{(h)},\boldsymbol{S}) = \Phi\Big(x_{i,j};\mu_{i,j}^{(q,h)},(\sigma_{i,j}^{(q,h)})^2\Big)$ using Gaussian conditional distribution with Schur complement. $\mu_{i,j}^{(q,h)}$ and $(\sigma_{i,j}^{(q,h)})^2$ are the mean and variance associated to $x_{i,j}$ knowing the components $Z_i^{(q,h)}$ and are defined below (Section 9.3.3).

Then, $\forall 1\leq i\leq n, \forall j\in J_f$ we draw the $Z_{i,j}^{(q+1,h)}$ according to the K_j-multinomial distribution whose parameters are the $(t_{i,j,1}^{(q+1,h)},\ldots,t_{i,j,K_j}^{(q+1,h)})$ defined by:

$$t_{i,j,k}^{(q+1,h)} \;=\; \frac{\pi_{j,k}\Phi(x_{i,j}^{(q+1,h)};\mu_{j,k},\sigma_{j,k}^2)}{\sum_{l=1}^{K_j}\pi_{j,l}\Phi(x_{i,j}^{(q+1,h)};\mu_{j,l},\sigma_{j,l}^2)}.$$

We see that $Z_{i,j}$ are not used if there is no missing values in $\boldsymbol{X}_i$ and others are not all needed so we can also optimize computation time by computing only the $Z_{i,j}$ that are needed in the Gibbs. For the last iteration of the Gibbs, in the last iteration of the 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 the number of component to compute in the likelihood can be very large we search a fast way to estimate the likelihood. We use the previous Gibbs algorithm with:

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

Here Q is the number of iterations of the Gibbs sampler and $\hat{\boldsymbol{\Theta}}$ is the global parameter associated to $\hat{\boldsymbol{\alpha}}$.

9.3.3 Parameters computation for the Gibbs sampler

For the step q of the Gibbs sampler at the step h of the Stochastic EM, $\forall (i,j) \in \{1,\dots,n\} \times J_f, \boldsymbol{M}_{i,j} = 1$ we have for the mean:

$$\mu_{i,j}^{(q,h)} = \mu_{j,Z_{i,j}^{(q,h)}} + \boldsymbol{\Sigma}_{j,X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}} \boldsymbol{\Sigma}^{-1}_{X_{\bar{ij}}^{(q+1,h)},X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}} (\boldsymbol{X}_{\bar{ij}}^{(q+1,h)} - \boldsymbol{\mu}_{X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}}),$$

and for the variance:

$$(\sigma_{i,j}^{(q,h)})^2 = \sigma^2_{j,Z_{i,j}^{(q,h)}} - \boldsymbol{\Sigma}_{j,X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}} \boldsymbol{\Sigma}^{-1}_{X_{\bar{ij}}^{(q+1,h)},X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}} (\boldsymbol{\Sigma}_{j,X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}})'.$$

we have $\mu_{j,Z_{i,j}^{(q,h)}}$ the mean of the component $Z_{i,j}^{(q,h)}$ of $\boldsymbol{X}^j$ (estimated once by `Rmixmod`) and $\sigma^2_{j,Z_{i,j}^{(q,h)}}$ the variance of the component $Z_{i,j}^{(q,h)}$ of $\boldsymbol{X}^j$ (estimated once by `Rmixmod`).
$\boldsymbol{\Sigma}_{j,X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}}$ is the vector of the covariances between $x_{i,j}^{(q+1,h)}$ and the other covariates $(X_{\bar{ij}}^{(q+1,h)})$ knowing the components $Z_i^{(q,h)}$.
$\boldsymbol{\Sigma}_{X_{\bar{ij}}^{(q+1,h)},X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}}$ is the variance-covariance matrix of the $X_{\bar{ij}}^{(q+1,h)}$ knowing the components $Z_i^{(q,h)}$.

$\boldsymbol{\mu}_{X_{\bar{ij}}^{(q+1,h)},Z_i^{(q,h)}}$ is the vector of the means associated to $\boldsymbol{X}_{\bar{ij}}^{(q+1,h)}$ knowing the components $Z_i^{(q,h)}$.

Means:

- Means associated to $\boldsymbol{X}_f$ are estimated once (by `Rmixmod`) following hypothesis 4 page 62.
- $\forall j \in \{1,\dots,d_r\}$, the mean associated to $x_{i,J_r^j}^{(q+1,h)}$ knowing $Z_i^{(q,h)}$ is:

$$\mu_{i,J_r^j,Z_i^{(q,h)}} = \sum_{l \in J_p^j} \bar{\alpha}_{l,J_r^j} \mu_{l,Z_{i,l}^{(q,h)}}.$$

Variances:

- The variances $\sigma^2_{l,Z_i^{(q,h)}}$ of the free covariates $\boldsymbol{X_f}$ are estimated once (by `Rmixmod`) following hypothesis 4 page 62.
- $\forall j \in \{1,\dots,d_r\}$, the variance associated to $x_{i,J_r^j}^{(q+1,h)}$ knowing $Z_i^{(q,h)}$ is:

$$\sigma^2_{i,J_r^j,Z_i^{(q,h)}} = \sigma_j^2 + \sum_{l \in J_p^j} \bar{\alpha}^2_{l,J_r^j} \sigma^2_{l,Z_{i,l}^{(q,h)}}.$$

Covariances: We look then at the covariances between the covariates:

- The covariance between two regressors is always zero:

$$\forall j_1 \in J_f, \forall j_2 \in J_f, \quad \mathrm{Cov}(x_{i,j_1}, x_{i,j_2}) = 0.$$

- The covariance between a response covariate and a free covariate that does not regressed it is always zero by Hypothesis 1 page 40:

$$\forall j_1 \in \{1,\dots,d_r\}, j_2 \notin J_p^{j_1} \cup J_r, \quad \mathrm{Cov}(x_{i,J_r^{j_1}}, x_{i,j_2}) = 0.$$

- The covariance between two response covariates without any common regressors is always zero by Hypotheses 1 and 3 page 41:

$$\forall j_1 \in \{1,\dots,d_r\}, j_2 \in \{1,\dots,d_r\}\backslash\{j_1\}, \text{ with } J_p^{j_1} \cap J_p^{j_2} = \emptyset, \quad \mathrm{Cov}(x_{i,J_r^{j_1}}, x_{i,J_r^{j_1}}) = 0.$$

- The covariance between two response covariates with common regressors is: $\forall j_1 \in \{1,\dots,d_r\}, j_2 \in \{1,\dots,d_r\} \setminus \{j_1\}$, with $J_p^{j_1} \cap J_p^{j_2} \neq \emptyset$:

$$\mathrm{Cov}_k(x_{i,J_r^{j_1}}, x_{i,J_r^{j_1}}) = \sum_{l \in J_p^{j_1} \cap J_p^{j_2}} \bar{\alpha}_{l,J_r^{j_1}} \bar{\alpha}_{l,J_r^{j_2}} \sigma^2_{l,Z_{i,l}^{(q,h)}}.$$

  where $\sigma^2_{l,Z_{i,l}^{(q,h)}}$ is the variance of $\boldsymbol{X}^l$ knowing $Z_{i,l}^{(q,h)}$, estimated once (by `Rmixmod`) following hypothesis 4 page 62.

- The covariance between a response covariate and one of its regressors is:

$$\forall j \in \{1,\dots,d_r\}, l \in J_p^j, \quad \mathrm{Cov}_k(x_{i,J_r^j}, x_{i,l}) = \bar{\alpha}_{l,J_r^j} \sigma^2_{l,Z_{i,l}^{(q,h)}}.$$

Then we are able to compute the Gibbs sampler.

9.4 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.

9.5 Numerical results on simulated datasets

9.5.1 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 $\boldsymbol{S}$. 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.

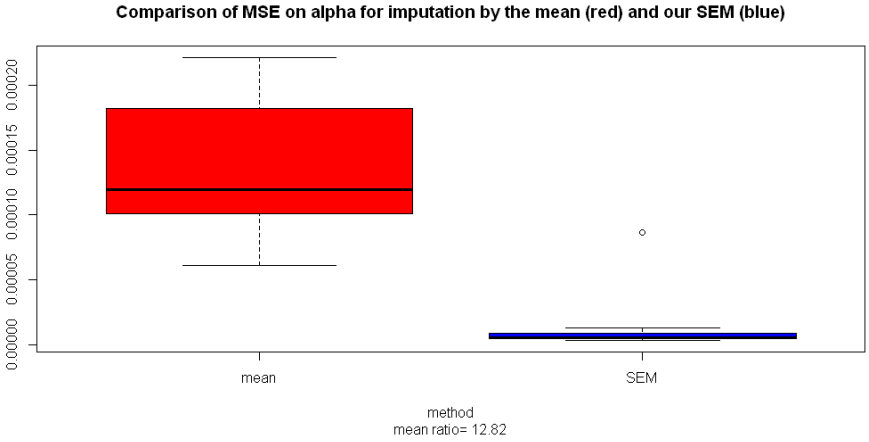


Figure 9.2: MSE on $\boldsymbol{\alpha}$ is significantly lower and with smaller variance with our Stochastic EM than with imputation by the mean.

We see (Figure 9.2) that our Stochastic EM is nearly 13 times more efficient in mean that estimation based on imputation by the mean. Our results are extremely good because each sub-regression is true and we have 30 individuals (even if missing values kind of reduce this number) to estimate 3 coefficients only each time. Although, using imputed values lead to learn a true regression with a factually incorrect dataset. Thus we should prefer to work without imputing the missing values but using the full generative model and the dependencies it implies. Imputation will always introduce some noise.

9.5.2 Multiple imputation

We have then imputed missing values in $\boldsymbol{X}_r$ by using the corresponding sub-regressions after $\boldsymbol{\alpha}$ has been estimated by the Stochastic EM. Missing values in $\boldsymbol{X}_f$ are estimated by the mean of 50 Gibbs iterations after the Stochastic EM and 2 warming steps of the Gibbs. Figure 9.3 shows the significant gain in MSE produced by our method.

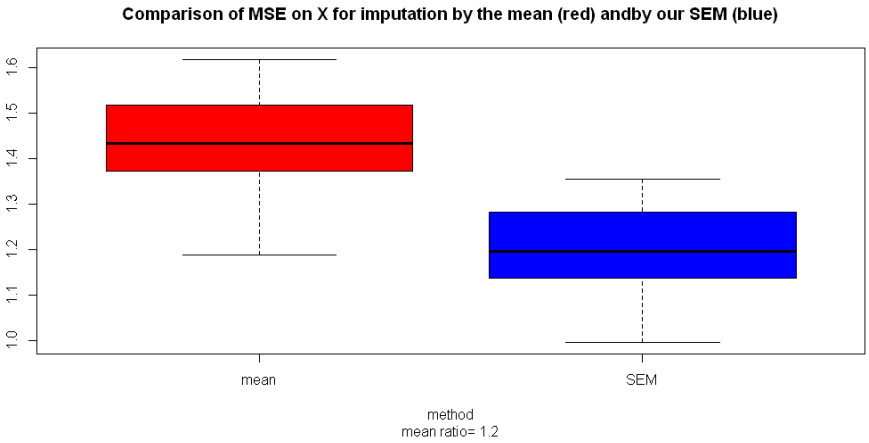


Figure 9.3: MSE on $\hat{\boldsymbol{X}}$ is significantly lower when using our Stochastic EM than with imputation by the mean.

9.5.3 Results on the main regression

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

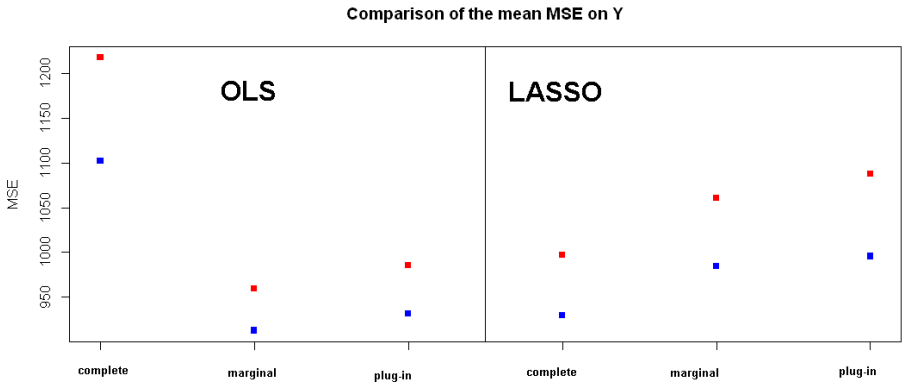


Figure 9.4: MSE on $\hat{\boldsymbol{Y}}$ are lower when using our Stochastic EM (blue) than with imputation by the mean (red) for the three model (complete, marginal and plug-in) using OLS or LASSO

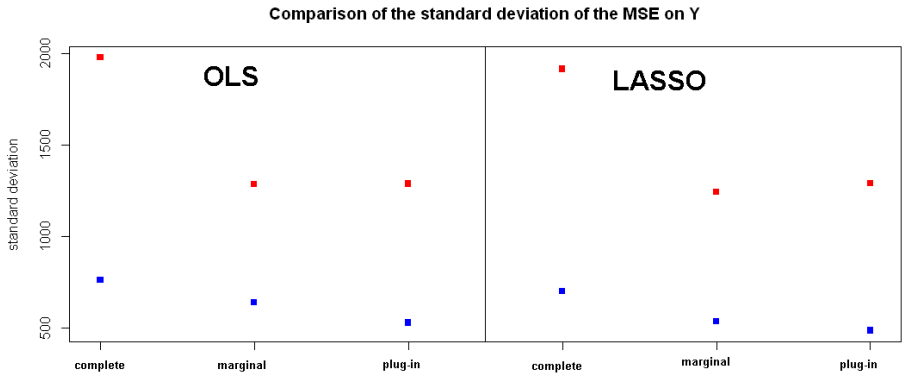


Figure 9.5: our Stochastic EM (blue) provides more robust results than imputation by the mean (red) but the variances are still too wide.

We obtain on a validation sample of 1000 individuals a predictive MSE smaller in mean with our method (Figure 9.4). But the variances are too important to really conclude (Figure 9.5). 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 $\boldsymbol{\alpha}$ and makes possible to find $\boldsymbol{S}$ based on a dataset with missing values.

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

9.6 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 7.2 to obtain 10% of missing values. Figure 9.6 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.

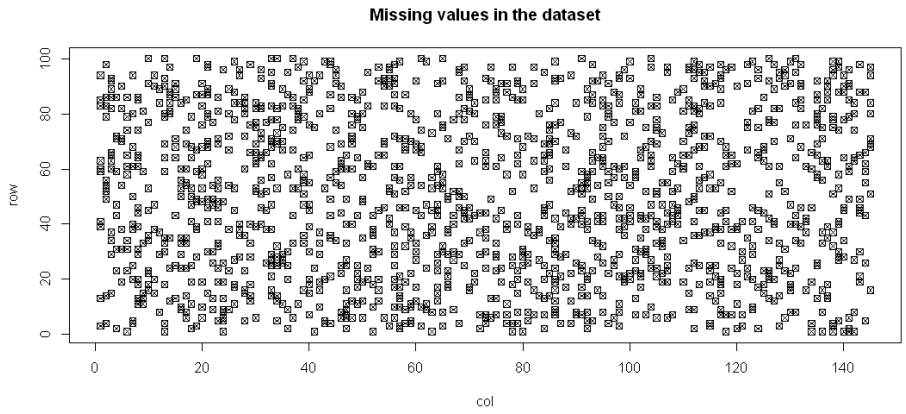


Figure 9.6: Graphical representation of the dataset with 10% of missing values

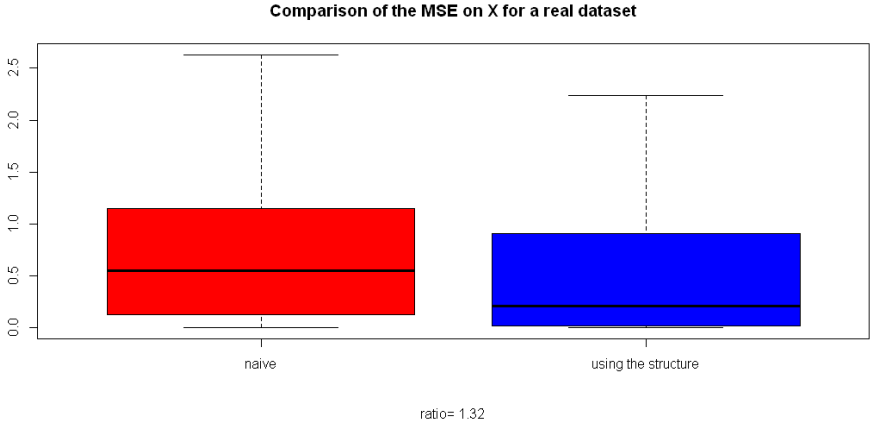


Figure 9.7: MSE on $\hat{\boldsymbol{X}}$ is 1.32 times lower in mean when using our method (blue) than with imputation by the mean (red).

We see in figure 9.7 that our Stochastic EM gives a smaller MSE with a smaller variance than imputation by the mean.

9.7 Conclusion

We can see that our generative model on the dataset and the explicit modeling of the correlations are powerful enough to manage missing values either to estimate $\boldsymbol{\alpha}$, $\boldsymbol{S}$, the main regression coefficients or even to impute missing values with an indicator of the reliability of these imputations. But we have to face the sad reality: we cannot create information and if too much values are missing we won't get good results. Moreover, the algorithms used here have good asymptotic properties but time is not infinite during a statistical study. However, if there are not too much missing values or if they are not too badly placed in the dataset we can hope to obtain good results in a reasonable amount of time.
There is still a lot of work to do in the field of missing values and we have only walked the firsts steps of this huge perspective, but these first results are encouraging and make us feel like this orientation is worthy to be followed.

Chapter 10

Conclusion and perspectives

10.1 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, the marginal model can be seen as a pre-treatment 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-regressions 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). Last but not least, missing values management is a positive side-effect of our explicit modeling of the correlations with promising perspectives. So we hope that statistics will continue to be a central tool for engineers.

To conclude:

- We provide an full generative model on $\boldsymbol{X}$ with explicit dependencies within the covariates by a structure of sub-regressions.
- We provide an efficient algorithm to find the structure of sub-regressions.
- We propose a variable pre-selection that takes into account the correlations and has proved (on both simulated and real datasets) to significantly improve predictive results when correlations are strong.
- We propose a plug-in model that allows to manage correlations by sequential estimation without deleting any covariates and that improves consistency (compared

to the LASSO).

- We provide a package `CorReg` (on CRAN) that implements our method.
- The full generative model on $\boldsymbol{X}$ also allows to manage missing values.
- Variable pre-selection is just a pretreatment so it can be used with any other statistical tool.
- The structure of sub-regression itself is useful for interpretation and makes the user more confident.
- Every step of the proposed method is simple to understand even by non-statistician.
- The method can be fully automated.

We have made the bet that explicit modeling of the correlations between the covariates would be valuable. Obtained results confirm that it was a good choice.

10.2 Perspectives

This work was focused on linear regression and the ways to avoid correlations issues. But statistics cover an extremely wider range of methods that sometimes also suffer from correlations. Then the perspectives of this work are wide.

10.2.1 Logistic regression

Logistic regression [Hosmer and Lemeshow, 2000] is in fact a linear regression with a post-treatment on $\boldsymbol{Y}$ so it is clear that we would obtain the same kind of phenomenon. It would be an easy generalization of CorReg to binary response variables. Introduction of qualitative covariates would be another big improvement for our method.

10.2.2 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. In such a model, the contribution of the structure itself would be even greater for the final user. An additional algorithm to search for distinct or equal coefficients within the classes would be to develop.

10.2.3 Using $\boldsymbol{Y}$ to estimate $\boldsymbol{S}$

As in PLS regression, we could imagine to estimate $\boldsymbol{S}$ using also $\boldsymbol{Y}$ to find a structure that will give the best results (in terms of a given criterion) when applied on $\boldsymbol{Y}$.

But it would probably give distinct structures according to the chosen model (marginal or plug-in) and also for distinct response variables (a same dataset about the same process to study distinct response variables) so it would not be very user-friendly. For this reason it was not implemented, but we can look a bit further.

This perspective would require to rethink the criterion used to choose the structure and (maybe) the algorithm to find it. The global criterion could be the BIC associated to the combination of both $\boldsymbol{S}$ and the chosen regression on $\boldsymbol{Y}$, that is BIC associated to $\mathbb{P}(\boldsymbol{S}|\boldsymbol{Y},\boldsymbol{X})$. In fact it would be like considering $\boldsymbol{Y}$ as a covariate in $\boldsymbol{X}$ but with distinct structural constraints (no need for the uncrossing-rule (Hypothesis 2 page 40) for example). Because $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X})$ is independent from $\mathbb{P}(\boldsymbol{X}_r|\boldsymbol{X}_f,\boldsymbol{S})$ (by Hypothesis 3 page 41) then, to be of any interest, global optimization would require to constrain $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{S})$ according to $\boldsymbol{S}$ (in terms of zeros in $\boldsymbol{\beta}$).
$\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{S})$ could be chosen to be the marginal model without variable selection, or the marginal model with variable selection (by the LASSO for example), or the plug-in model with or without variable selection at each step.

One possible consequence of global optimization with such a constraint would be a reduced marginalization with some response covariates kept when the correlations implied are not too strong, as a trade-off between the marginal and the plug-in model. A simpler way to achieve this would be a kind of stepwise algorithm with forward variable selection starting with our marginal model and allowing variable selection for each model tested (by LASSO for example). But it implies to use correlated covariates and we know that it can be a source of problems (it is the main motivation of this thesis). Removing more covariates (by global optimization) in the marginal model would not be of real interest because we already propose to use variable selection methods (like the LASSO) to estimate the coefficients and the implied covariates are supposed to be uncorrelated. Global optimization could be replaced by a suitable variable selection algorithm for the regression of $\boldsymbol{Y}$ by $\boldsymbol{X}$ taking the structure $\boldsymbol{S}$ into account, it would be preferable (for interpretation) to have the same structure for every response variable and to only adapt the way to use the structure to estimate the regression coefficients.

However, to model $\mathbb{P}(\boldsymbol{X},\boldsymbol{Y})$ instead of $\mathbb{P}(\boldsymbol{X})$ would also help to better manage missing values by using $\mathbb{P}(\boldsymbol{X}_M|\boldsymbol{X}_O,\boldsymbol{Y}_O)$ instead of $\mathbb{P}(\boldsymbol{X}_M|\boldsymbol{X}_O)$. Then we would have more information to make imputations. But it would use in the Gibbs sampler $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X})$ that is what we search. So it would need some additional work.

10.2.4 Pre-treatment for non-linear regression

Polynomial regression, Classification and Regression Trees, and any other method could also benefit from the variable selection pre-treatment implied by our marginal model. Definition of a plug-in model for these methods would be of great interest so it is a challenging perspective.

10.2.5 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.

10.2.6 Improved programming and interpretation

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).

Ergonomics of the software should be improved to better fit industrial needs. This work is in progress and further work will be provided just after this thesis to get closer to this goal as CorReg will continue to be used and taught at ArcelorMittal's steel plants of Dunkerque and Florange.

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.

[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.

[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.

[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-Rabusseau et al., 2012] Maugis-Rabusseau, 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.

[Meinshausen and Bühlmann, 2006] Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462.

[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.

[Robert and Casella, 2005] Robert, C. P. and Casella, G. (2005). Monte carlo statistical methods.

[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.

[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.

[Stone, 1977] Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 44–47.

[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.

[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.

[Yang, 2005] Yang, Y. (2005). Can the strengths of aic and bic be shared? a conflict between model indentification and regression estimation. *Biometrika*, 92(4):937–950.

[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.

Appendices

Appendix A

Identifiability

A.1 Definition

We call identifiability:

$$\nexists(\boldsymbol{S}, \tilde{\boldsymbol{S}}) \in \mathcal{S}_d \times \mathcal{S}_d \text{ with } \boldsymbol{S} \neq \tilde{\boldsymbol{S}} \text{ and } \mathbb{P}(\boldsymbol{X}; \boldsymbol{S}) = \mathbb{P}(\boldsymbol{X}; \tilde{\boldsymbol{S}}) \tag{A.1}$$

To avoid label-switching consideration, we suppose here (without loss of generality) that $\boldsymbol{J}_r$ 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.

A.2 Sufficient condition for identifiability

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

$$\forall j \in \{1, \ldots, d_r\}, d_p^j > 1. \tag{A.2}$$

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 $\boldsymbol{X}$ and $\boldsymbol{S}$ 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

$$j \in \{1, \ldots, d_r\}, i \in J_p^j \text{ then } \operatorname{cov}(\boldsymbol{X}^i, \boldsymbol{X}^{J_r^j}) \neq 0 \tag{A.3}$$

2. Both variables are regressed by a common covariate in their respective sub-regression: $\exists k \in J_f, \exists (i,j) \in \{1, \ldots, d_r\} \times \{1, \ldots, d_r\}$ with $i \neq j, k \in J_p^i$ and $k \in J_p^j$ then:

$$\operatorname{cov}(\boldsymbol{X}^{J_r^i}, \boldsymbol{X}^{J_r^j}) \neq 0.$$

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

- if $\exists(i,j) \in \{1,\dots,d_r\} \times \{1,\dots,d_r\}, \text{cov}(\boldsymbol{X}^{J_r^i}, \boldsymbol{X}^{J_r^j}) \neq 0$ 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 $\text{cov}(\boldsymbol{X}^i, \boldsymbol{X}^j) = 0$ because covariates in $\boldsymbol{X}_f$ are orthogonal (by hypotheses 1 and 2).

- if $j \in \{1,\dots,d_r\}, i \in J_f$ and $\text{cov}(\boldsymbol{X}^{J_r^j}, \boldsymbol{X}^i) \neq 0$ then $i \in J_p^j$ by hypotheses 1 and 3 and equation 4.3.

□

proof of the identifiability criterion: We suppose that A.2 is verified and the model is not identifiable:

$$\exists(\boldsymbol{S}, \tilde{\boldsymbol{S}}) \in \mathcal{S}_d \times \mathcal{S}_d \text{ with } \boldsymbol{S} \neq \tilde{\boldsymbol{S}} \text{ and } \mathbb{P}(\boldsymbol{X}; \boldsymbol{S}) = \mathbb{P}(\boldsymbol{X}; \tilde{\boldsymbol{S}}) \tag{A.4}$$

$\tilde{\boldsymbol{S}} = (\tilde{\boldsymbol{J}}_r, \tilde{\boldsymbol{J}}_p)$ contains $\tilde{d}_r$ sub-regressions and is characterized by $\tilde{\boldsymbol{J}}_r = (\tilde{J}_r^1, \dots, \tilde{J}_r^{\tilde{d}_r}), \tilde{\boldsymbol{J}}_p = (\tilde{J}_p^1, \dots, \tilde{J}_p^{\tilde{d}_r})$.
Because $\boldsymbol{S} \neq \tilde{\boldsymbol{S}}$ we have $\boldsymbol{J}_r \neq \tilde{\boldsymbol{J}}_r$ or $\boldsymbol{J}_p \neq \tilde{\boldsymbol{J}}_p$.

- If $\boldsymbol{J}_r = \tilde{\boldsymbol{J}}_r$ and $\boldsymbol{S} \neq \tilde{\boldsymbol{S}}$ 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 $\text{cov}_S(\boldsymbol{X}^{J_r^j}, \boldsymbol{X}^i) \neq 0$ and $\text{cov}_{\tilde{S}}(\boldsymbol{X}^{J_r^j}, \boldsymbol{X}^i) = 0$ (from the lemma) so the two structure do not give the same joint distribution, leading to a contradiction.

- If $\boldsymbol{J}_r \neq \tilde{\boldsymbol{J}}_r$ 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 $J_r^j \notin \tilde{J}_r$ then $J_r^j \in \tilde{J}_f$ (recall $J_f = \{1,\dots,d\} \setminus J_r$). We note that $J_r^j \in J_r$ means $\exists k_1 \neq k_2, \{k_1,k_2\} \subset J_p^j \subset J_f$. Then $\text{cov}_{\boldsymbol{S}}(\boldsymbol{X}^{J_r^j}, \boldsymbol{X}^{k_1}) \neq 0$ and $\text{cov}_{\boldsymbol{S}}(\boldsymbol{X}^{J_r^j}, \boldsymbol{X}^{k_2}) \neq 0$ so by the lemma k_1 and k_2 are responses variables in $\tilde{\boldsymbol{S}}$: $\exists(l_1,l_2) \in \{1,\dots,\tilde{d}_r\} \times \{1,\dots,\tilde{d}_r\}, \tilde{J}_r^{l_1} = k_1, \tilde{J}_r^{l_2} = k_2$ 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 $\text{cov}_{\tilde{\boldsymbol{S}}}(\boldsymbol{X}^{k_1}, \boldsymbol{X}^{k_2}) \neq 0$ that is not possible because $\{k_1,k_2\} \subset J_p^j \subset J_f$ and covariates in $\boldsymbol{X}^{J_f}$ are orthogonal by hypotheses.

Finally, condition A.2 is sufficient for identifiability of $\boldsymbol{S}$. □

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.

Appendix B

CorReg: Existing and coming computer tools

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 complete, marginal and plug-in models with many regression tools (OLS, stepwise, LASSO, elastic-net, 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 2015, using Basic Excel R Toolkit (BERT[1]). Figure B.1 gives a glimpse of what it will look like. Another project is to propose `CorReg` in Gretl[2] (Gnu Regression, Econometrics and Time Series Library) that already have tools for Simultaneous Equation Modeling.

The package 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 (as an 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 (see Figure 3.2) 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 ArcelorMittal Dunkerque to provide ergonomic and powerful statistical tools to non-statisticians.

[1] `https://github.com/StructuredDataLLC/Basic-Excel-R-Toolkit`
[2] `http://gretl.sourceforge.net/`

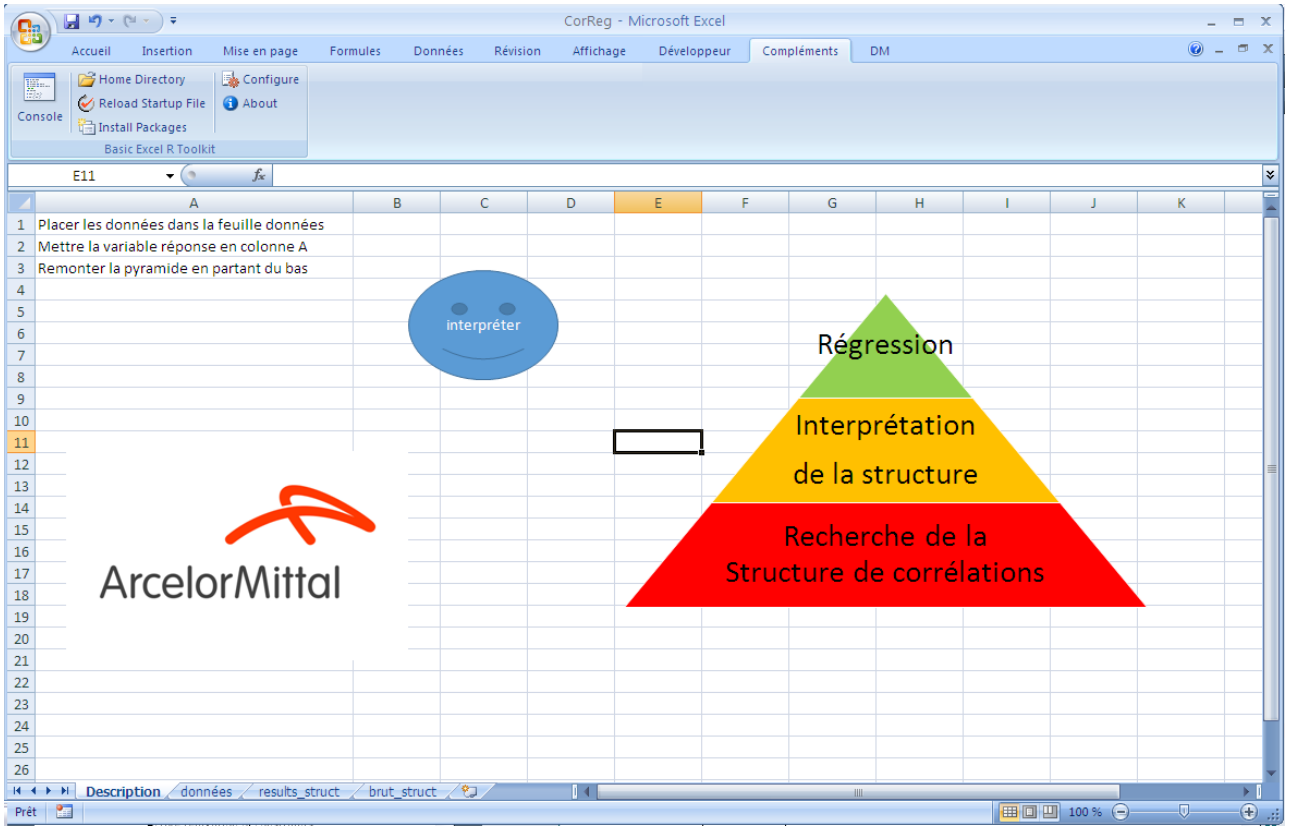


Figure B.1: Screenshot of the Graphical User Interface of `CorReg` in Excel.

Quick overview of the package: The following script gives an overview of `CorReg 0.15.8`.

We first generate a dataset:

```
> rm(list=ls()) #clear the workspace
> require(CorReg) #load CorReg if not already done
#we first generate a dataset
> base=mixture_generator(
        n=15, #number of individuals
        p=10, #number of covariates
        ratio=0.4, #ratio of covariates that are response variables
        tp1=1,#ratio of predictor covariates allowed to
        #appear in the regression of Y
        tp2=1,#ratio of response covariates allowed to
        #have a non-zero coefficient in the regression of Y
        tp3=1,#ratio of strictly independent covariates allowed to
        #have a non-zero coefficient in the regression of Y
        positive=0.5,#ratio of positive coefficients in the regressions
        R2Y=0.8, #R-squared of the main regression
        R2=0.9, #R-squared of the sub-regressions
        scale=TRUE,#to scale the covariates
        #(then responses do not have a greater variance or mean)
        max_compl=3,#maximum number of predictors in each sub-regression
    )
  #learning set (n individuals)
   > X_appr=base$X_appr
   > Y_appr=base$Y_appr
```

```
#validation sample (1 000 individuals by default)
 > Y_test=base$Y_test
 > X_test=base$X_test

 TrueZ=base$Z#True generative structure (binary adjacency matrix)
```

Then we can start the study by estimation of the marginal densities:

```
#density estimation of each covariate with a max of 10 components
> density=density_estimation(X=X_appr,nbclustmax=10,detailed=TRUE)
> Bic_null_vect=density$BIC_vect# vector of the d BIC
```

The BIC obtained comes as an input for the MCMC:

```
#MCMC to find the structure
> res=structureFinder(
              X=X_appr,#the dataset
              verbose=0,#several levels of details during the walk
              reject=0,#constraint relaxation
              Maxiter=900,#max number of steps for each initialization
      nbini=30,#number of initializations
      candidates=-1,#strategy for the neighbourhood
      Bic_null_vect=Bic_null_vect,
      star=TRUE,
      p1max=15,#maximum complexity of sub-regressions
      clean=TRUE#additional cleaning steps at the end
    )
> hatZ=res$Z_opt#best adjacency matrix found
> hatBic=res$bic_opt#associated BIC
```

Practically speaking, `CorReg` returns the best structure seen during the walk (even if the corresponding candidate has never been chosen) as an adjacency matrix. The package also gives the local structure when the walk stops so the 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 iterations but `CorReg` can also stop the walk after a given number of steps on the best found model. Exact sub-regressions are directly given to the user during the walk to allow to stop the walk and relaunch it without one of the implied covariates (without any loss of information).
`CorReg` gives to the user the choice with stationarity, included in the neighbourhood by default. Moreover, the package let the user choose many 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 $\boldsymbol{G}$.

Once a structure is found, we can compare it to the true structure (see section 5.6.1):

```
  #Structure comparison
> compZ=compare_struct(trueZ=TrueZ,Zalgo=hatZ)#qualitative comparison
> compZ$true_left #number of "True Responses"
[1] 4
> compZ$false_left #number of "Wrong responses "
[1] 0
> compZ$ratio_true1 #no correlations are missing
[1] 1
> compZ$false1 #3 correlations were added (over-fitting).
[1] 3
```

So here we have found all the 4 sub-regressions. No correlations are missing but 3 were added (some over-fitting).
We can also look for further details:

```
#interpretation of the structure found, ordered by increasing R2:
> readZ(Z=hatZ,crit="R2",X=X_appr,output="all",order=1)
   # (<NA>line: name (or index) of the response covariate)
[[1]]
                coefs       var
1   0.9110810299677        R2
2               <NA>         6
3  -0.60389362304509 intercept
4   -0.5080670369077         2
5  0.262896800316838         3
6  -0.41335932330995         5

[[2]]
                coefs       var
1   0.942274685071874        R2
2               <NA>        10
3   0.802402741595218 intercept
4  -0.532932485555117         2
5  0.0972300933365791         4
6  -0.409558991808605         5

[[3]]
                coefs       var
1   0.947429765365844        R2
2               <NA>         9
3  -0.881772084905585 intercept
4  0.0743241715086655         2
5  -0.153741473798903         3
6  -0.371811407183514         4
7   0.192198139511508         5

[[4]]
                coefs       var
1   0.954402650897285        R2
2               <NA>         8
3  -0.867797429793816 intercept
4   0.213533111330489         1
5  -0.0367999079013308         2
6  -0.0405454729730624         3
7   0.312013168373898         5
8   0.265126747497754         7
```

Then we use the structure for the main regression on the response variable:

```
   #Regression coefficients estimation
> resY=correg(X=X_appr,Y=Y_appr,Z=hatZ,#we give the dataset and the structure
              compl=TRUE,expl=TRUE,pred=TRUE,#we want the 3 models
              select="NULL",#we will use OLS
              K=10# number of groups for the K-fold cross-validation
              )
```

And we can compare the coefficients of the three models (intercept in first position):

```
> resY$compl$A #coefficients of the complete model
 [1] -25.511666 -27.237825  24.890817  -7.268666 -27.829301
 [6]  37.510213  34.882207 -23.717979  48.738350 -41.933443
[11]  62.007268
```

```
> resY$expl$A #marginal model
 [1]  -6.769296 -10.152404 -32.502943   8.031446  -2.939350
 [6]   5.535478   0.000000 -19.134011   0.000000   0.000000
[11]   0.000000
> resY$pred$A #plug-in model
  [[1] -29.008358 -13.188767  -6.823660   5.941027  -9.369883
  [6]  21.484112   6.201073 -22.904017  14.219635  -6.743260
 [11]  40.350803
> base$A #true model
 [1]  -8 -10 -10  -5  -6   8  14 -13   9  -7  14
```

And we can use the validation sample to compare the models in terms of predictive efficiency:

```
> MSE_complete=MSE_loc(Y=Y_test,X=X_test,A=resY$compl$A)#complete
> MSE_marginal=MSE_loc(Y=Y_test,X=X_test,A=resY$expl$A)#marginal
> MSE_plugin=MSE_loc(Y=Y_test,X=X_test,A=resY$pred$A)#plug-in
> MSE_true=MSE_loc(Y=Y_test,X=X_test,A=base$A)#true model

> data.frame(MSE_complete,MSE_explicative,MSE_predictive,MSE_true)
  MSE_complete  MSE_marginal     MSE_plugin       MSE_true
    628.0226       467.119        454.7964       203.2154
```

We observe that the marginal model is better than the complete model but the plug-in model is able to improve the results from the marginal model by using all the covariates.

This script describes the whole process of `CorReg`. We can see that it is really adapted to an automated process as we have done for Excel.