THESIS DECLARATION

The undersigned

Michele Peruzzi
PhD Registration Number: *1248830*

Thesis Title:
Bayesian Modular Approaches for Regression

PhD in Statistics
30^{th} Cycle
External Advisor: *Professor David Dunson*
Internal Advisor: *Professor Sonia Petrone*
Year of Discussion: *2019*

DECLARES

Under his responsibility:

1) that, according to the President's decree of 28.12.2000, No. 445, mendacious declarations, falsifying records and the use of false records are punishable under the penal code and special laws, should any of these hypotheses prove true, all benefits included in this declaration and those of the temporary embargo are automatically forfeited from the beginning;

2) that the University has the obligation, according to art. 6, par. 11, Ministerial Decree of 30th April 1999 protocol no. 224/1999, to keep copy of the thesis on deposit at the Biblioteche Nazionali Centrali di Roma e Firenze, where consultation is permitted, unless there is a temporary embargo in order to protect the rights of external bodies and industrial/commercial exploitation of the thesis;

3) that the Servizio Biblioteca Bocconi will file the thesis in its 'Archivio istituzionale ad accesso aperto' and will permit on-line consultation of the complete text (except in cases of a temporary embargo);

4) that in order keep the thesis on file at Biblioteca Bocconi, the University requires that the thesis be delivered by the candidate to Società NORMADEC (acting on behalf of the University) by online procedure the contents of which must be unalterable and that NORMADEC will indicate in each footnote the following information:

   - thesis *Bayesian Modular Approaches for Regression*;
   - by *Michele Peruzzi*;
   - discussed at Università Commerciale Luigi Bocconi - Milano in *2019*;
   - the thesis is protected by the regulations governing copyright (law of 22 April 1941, no. 633 and successive modifications). The exception is the right of Università Commerciale Luigi Bocconi to reproduce the same for research and teaching purposes, quoting the source;

5) that the copy of the thesis deposited with NORMADEC by online procedure is identical to those handed in/sent to the Examiners and to any other copy deposited in the University offices on paper or electronic copy and, as a consequence, the University is absolved from any responsibility regarding errors, inaccuracy or omissions in the contents of the thesis;

6) that the contents and organization of the thesis is an original work carried out by the undersigned and does not in any way compromise the rights of third parties (law of 22 April 1941, no. 633 and successive integrations and modifications), including those regarding security of personal details; therefore the University is in any case absolved from any responsibility whatsoever, civil, administrative or penal and shall be exempt from any requests or claims from third parties;

7) that the PhD thesis is not the result of work included in the regulations governing industrial property, it was not produced as part of projects financed by public or private bodies with restrictions on the diffusion of the results; it is not subject to patent or protection registrations, and therefore not subject to an embargo.

14 April 2019

Michele Peruzzi

Contents

List of Figures

List of Tables

Acknowledgements

This dissertation would not have been possible without the guidance and support from my advisors Prof. Petrone and Prof. Dunson, to whom I am very grateful. I would also like to thank Prof. Isadora Antoniano and Prof. Daniele Durante for their comments on an early draft of this dissertation. I have shared this journey with amazing colleagues, both at the Department of Statistical Science at Bocconi University – Amir, Paolo, Stefano, Veronica and Xuefei – and at the Department of Statistical Sciences at Duke University as visiting scholars – Emanuele, Massimiliano and Sally. In particular, I would like to recognize Stefano Rizzelli for his collaboration in developing the asymptotic results of Chapter 4. Additionally, I would like to mention Angela and Silvia at Bocconi University and Karen and Nicole at Duke University, who helped sort the logistical side of my journey.

I want to thank my parents Daniele and Lorenza for their invaluable and continuous support, never failing to teach me and make me grow as a person even when I am thousands of miles away from home. My gratitude also goes to those with whom I have shared important parts of my personal journey and who have been close to me during this time: Younis, Erta, Alessandro M., Monica, Federica, Nicolas, Alessandro A., Alessandro F.

Abstract

This thesis develops novel Bayesian modular methods for multiscale analysis in regression. In these settings, the regression function is estimated simultaneously at multiple levels of resolution of the inputs, with the goal of assessing their contribution in explaining the variability of the output. We implement our methodology in three cases. First, with the goal of measuring the contribution to the regression function of different resolutions of the data, assuming these resolutions are predetermined and observed simultaneously. Second, for efficient dimension reduction with structured high-dimensional predictors obtained by sensors. Third, as a flexible tool to facilitate interpretability in more standard regression settings with unstructured predictors. We consider these scenarios separately. In Chapter 1, we motivate multiscale analysis in regression settings by providing an overview of the relevant problems and the corresponding literature. We proceed in Chapter 2 by introducing a Bayesian modular approach for multiscale regression which can be used with high-dimensional structured predictors, outlining potential advantages and disadvantages compared to other established methods. We test our method on an applied problem using tfMRI data. Chapter 3 extends the framework developed previously to the case in which resolution is unknown and the predictor is a structured tensor. We revisit our applied analysis by recasting the problem as a scalar-on-image regression with the goal of estimating a multiresolution decomposition of the regression coefficients image. Finally, we change our point of view in Chapter 4, where multiple scalar predictors are considered in an additive model with the goal of obtaining interpretable results. Each additive component is further decomposed into resolution contributions.

Chapter 1

Introduction

This thesis is about multiscale regression and focuses on two scenarios. First, we focus on the case in which a multitude of predictors are studied in their relation to a scalar response variable. The predictors are structured, or indexed. For example, groups of predictors may be identified having a shared meaning (e.g. products belonging to the same category, brain regions clustering in lobes and hemispheres). Alternatively, predictors may be temporally indexed, in which case the goal is to use an entire time series as a predictor for each statistical unit, with the goal of identifying time intervals most associated to changes in the response. In this setting, the predictors' underlying structure or indexing can be exploited for dimension reduction to improve estimation and prediction. The second setting we consider, instead, is a more standard scenario in which the predictors are not structured or indexed, and the goal is to flexibly estimate the regression function. In this latter case, we make the assumption that the effect of each predictor on the response can be approximated by a step function, meaning it can be considered constant over regions of the predictor's space. To facilitate the search for such regions, we decompose the step functions into coarse-to-fine scale contributions. The second scenario can be described as deriving from the first, on which we focus now.

Suppose we consider a response variable y_i and a vector $\mathbf{x}_i = (x_{i,s})_{s \in S}$ of predictors, available for each statistical unit $i = 1, \ldots, n$. The aim is to study the relationship between the output vector y and the data matrix X_S via regression methods. In this context, S is the *measuring grid* determining the *scale*, or equivalently the *resolution*, of the problem. S also determines the problem dimension, as $|S| = d$, and typically $d > n$. There might not be a single S at which the data are measured. Instead, the researcher may face data at grids $\{S_1, \ldots, S_K\}$, each corresponding to a different measuring frequency, sampling rate, or image resolution. Notably, recent technological advancements allow modern recording devices to collect measurements at extremely high resolutions; such high resolutions might prove counterproductive if the aim is to use these data as inputs in regression. In fact, increasing the resolution of inputs results in $d \gg n$ problems, and likely high correlation between adjacent measurement locations. Similarly, considering data at higher resolutions might prove unfruitful in explaining the variation of the response y, given the noisy relationship assumed between outputs and inputs. For these reasons, researchers might opt to down-sample the data before analysis. However, any specific choice hides uncertainty on the resolution itself.

These issues motivate the use of multiscale models in regression. Models of such kind consider multiple resolutions of the data, with the aim of estimating their relative contribution to the regression function. These alternative resolutions do not necessarily correspond to existing measuring grids of the data; one can think of them as mathematical tools to achieve multiresolution interpretations of the results. We introduce a novel methodology for multiscale regression that can be used: first, on regression problems in which multiple measuring grids of the data concurrently exist, and the researcher is interested in assessing their relative contribution to the regression function. This setting lays the basis for the development of our methodology. We build the foundations of our approach in Chapter 2, where we motivate the modularization of the Bayesian posterior distribution to solve problems arising in multiresolution regression settings. In Chapter 3 we introduce a general algorithm to estimate multiscale decompositions of the regressors starting from data at a single high resolution. This new setting amounts to considering the resolution as unknown. In Chapter 4, we consider the classical case of nonparametric regression with unstructured predictors. We introduce a modularization strategy for a novel model which decomposes regression function into predictor and resolution contributions.

The rest of this introduction further motivates the problems we consider, and continues with an overview of the relevant literature.

1.1 Alternative data resolutions

In many practical settings, as we have mentioned above, data are recorded at multiple resolutions simultaneously. We can label the available grids as $\{S_1, \dots, S_K\}$. Researchers typically only choose to consider predictors x_{i,S_j} at some fixed intermediate resolution S_j. However, any specific resolution choice is arbitrary, and ideally one should account for resolution uncertainty when modeling the relationship between inputs and output. Uncertainty on the data resolution underpins many problems such as time series or image classification when data are generated by modern high-resolution sensors, but similar issues arise with non-sensor data having analogous structure.

We now introduce an illustrative example to show how resolution choice might be crucial to model specification. Consider data generated by $y_i = \mathbf{x}_i'\beta + \varepsilon_i$, where $\varepsilon \overset{iid}{\sim} N(0, \sigma^2)$ and σ^2 is a known constant. Suppose the response y_i is scalar, whereas $\mathbf{x}_i = (x_{s,i})_{s \in S}$ and $S = \{t_1, \dots, t_{2d}\}$ for some d. $\beta = (\beta_1, \dots, \beta_j, \dots, \beta_{2d})$ is the unknown vector of regression coefficients. β_j thus represents the linear effect of the measurement x at time j on the response y. Suppose that the X_S "high-resolution" matrix collecting all the subjects' time series is such that $\frac{X_S' X_S}{n}$ is a block-diagonal matrix as follows:

$$\frac{X_S' X_S}{n} = \begin{bmatrix} B_2 & & 0 \\ & \ddots & \\ 0 & & B_2 \end{bmatrix} \qquad B_2 = \begin{bmatrix} 1 & 1-\varepsilon \\ 1-\varepsilon & 1 \end{bmatrix},$$

where ε is a small, positive scalar. Strong correlation between measurements at adjacent locations leads to likely failures in estimating the unknown β because of the near-collinearity in the columns of X_S. Many methods exist in the literature to deal with this problem – we present an overview of the literature later in this chapter. However, a simple, yet effective solution is to model the regression by considering a low-resolution design matrix X^* of dimension $n \times d$ obtained by summing or averaging pairs of adjacent columns of X. This would result in a diagonal observed covariance matrix, and solve the multicollinearity problem. The coefficient estimates, however, will be "compressed" to a resolution lower than the original X_S; this evidences there is a trade-off between potentially richer results at high resolution, and more stable results at low resolution. A similar conclusion can be reached by casting the resolution problem along the lines of a bias-variance trade-off; we do so in Section 2.8.1. In other words, data resolution is a source of uncertainty which can be accounted for when modeling the regression function. Ultimately, failure to consider an existing multiresolution structure may lead to poor reproducibility or harm interpretability of the results. A natural way to build a multiscale regression model using grids $\{S_1, \dots, S_K\}$ is to consider the corresponding data matrices $\{X_1, \dots, X_K\}$ in the same regression model, as this could in principle allow to weigh the contribution of each scale to the regression function. Such an approach, however, fails to deliver reliable results in the simplest possible scenario of a linear regression model with simple down-samplings of the data (i.e. summing or averaging of adjacent columns of X_S). We explain this problem in Chapter 2, where we provide a solution by introducing our methodology.

1.2 Multiscale decompositions

The first point of view we mentioned arises when the interest lies in a multiscale interpretation of the regression function, obtained by using data at a multiple existing scales. However, sometimes the interest lies in finding a multiresolution decomposition of the regression function starting from data at a single resolution. In this scenario, no resolution of the data are considered other than S^*, and the regression function is decomposed in coarse-to-fine components $\{S_1, \dots, S_K\}$. In other words, the objective is to detect and estimate coexisting low- and high-resolution patterns, disentangling their effects. A partial solution for this kind of problem which is related to the methodology we propose is the discrete wavelet transform (DWT), with Haar wavelets being the simplest, most interpretable choice of wavelet basis. We review the relevant literature on wavelets in Section 1.4. With this methodology, each row of the design matrix X^* corresponding to resolution S^* is projected into the wavelet domain, resulting in a new design matrix $X_H = X^* H$, where H is the linear operator that applies the DWT. H is an orthogonal $p \times d$ matrix such that $HH' = I_d$, the identity matrix. In linear regression, an equivalent way to write the usual model is thus $y = X^* HH' \beta + \varepsilon$. If one were to replace X^* by $X_H = X^* H$, inference would refer to the transformed coefficients $\beta_H = H'\beta$. The wavelet coefficients β_H collect magnitudes of effects of regions of X^* at different scales, and therefore provide information on the multiscale structure of the regression function. Inference on the original β can still be performed by reversing the DWT, since $\beta = H\beta_H$.

The DWT is not a dimension reduction tool; it is merely a way to project the data onto a new space. Methods for high dimensional data can be used when $d > n$, and dimension reduction can be achieved by dropping the columns of X_H interpreted as being the least important resolution components. This can be achieved by imposing a Lasso penalty on the Haar coefficients. Setting some wavelet coefficients to zero corresponds to using $\tilde{H}$ in the model $y = X\tilde{H}\beta_{\tilde{H}} + \varepsilon$, where $\tilde{H}$ is a "compressed" DWT and only includes the columns of H corresponding to the non-zero coefficients. Since $\tilde{H}\tilde{H}' \neq I_p$, we are limited to studying $\tilde{H}\beta_{\tilde{H}}$.

As an alternative to using Lasso penalties on β_H, we can obtain $\tilde{H}$ by dropping the finest-resolution components of H (i.e. truncating the Haar basis representations of $x_{i,S}$). We show the effect of alternative choices for $\tilde{H}$ on the "compressed" least-squares estimates $\tilde{H}\hat{\beta}_{\tilde{H}}$ for simulated data in Figure 1.1. If we truncate the Haar-bases, $\tilde{H}\beta_{\tilde{H}}$ proceed in steps of equal length (Figure 1.1, last row). Equivalently, we could obtain the *same* results by summing columns of X corresponding to the same "steps." In other words, the coefficient estimate that we obtain by truncating a Haar basis is the same as the one we would obtain by simple sums of the corresponding columns of X. These simple sums correspond to down-sampling each row of X^* to a coarser grid. The advantage of using Haar bases is that we can simultaneously estimate multiple such representations jointly, making it possible to quantify their relative contribution to the regression function. However, a disadvantage is that the DWT typically requires $p = 2^L$ for some L, resulting in a multiscale structure with L layers that can only proceed by successively splitting S^* in two equally-sized regions. In practice, this has two undesirable effects: first, it restricts the interpretability of the results, especially in cases in which there is a multiscale structure not equivalent to the wavelet decomposition. For example, a wavelet decomposition of hourly

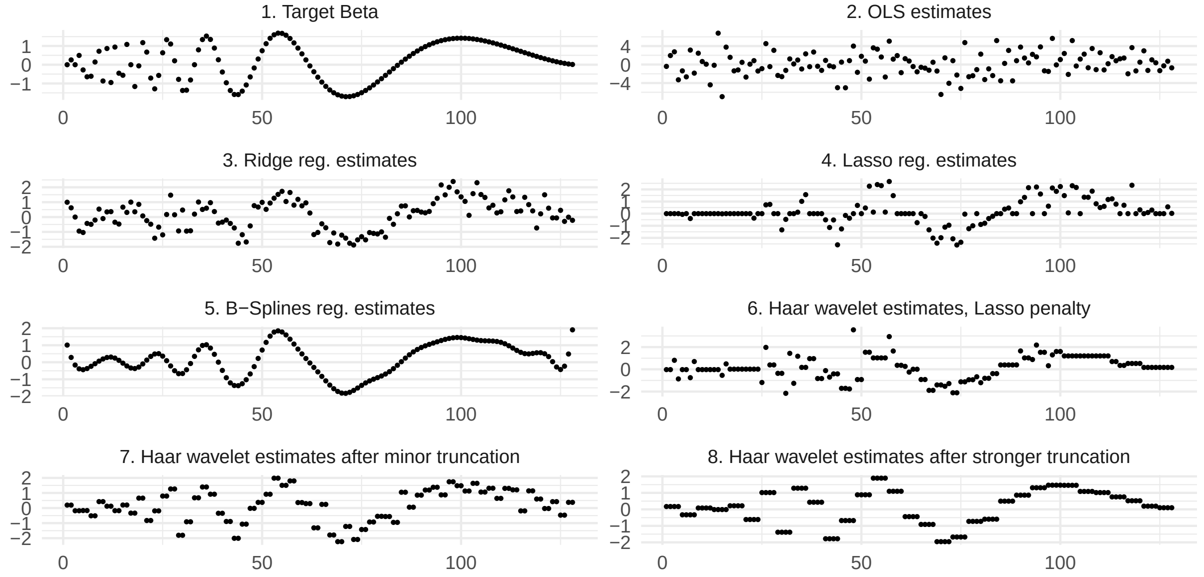


Figure 1.1: We simulate a linear regression model $y = X\beta + \varepsilon$, $\varepsilon \sim N(0, I_n)$, $n = 150, d = 128$, with highly correlated predictors; the horizontal axes represent the column index of X, on the vertical axes we report the value of the estimated β_j, $j = 1, \ldots, p$, from 7 models plus the truth.

time-series data cannot take into account the native hour/day/week structure. Second, it requires an additional pre-processing step whenever the number of data points is not a power of 2. In this case, the researcher can discard some observations, or interpolate the grid appropriately; these decisions might affect the results.

The methodology we introduce is similar to regression using Haar bases, as we consider multiple down-samplings of the data in a single regression model. However, it is not bound by the same constraints, and we can apply it on data with $p \neq 2^L$, considering $j = 1, \ldots, K$ layers, each corresponding to a grid S_j. The grids can be adaptively learned, thereby giving rise to a multiscale decomposition of the regression function. The same class of models can be applied to problems which are typically defined as scalar-on-function regression; we also introduce an extension to scalar-on-image regression problems in Chapter 3.

1.3 Nonparametric regression from a multiscale point of view

A special case arises from the previous point if we take $X^* = I_n$, the identity matrix of size n. The resulting model becomes $y_i = \beta_i + \varepsilon_i$, and β_i is thus the mean of the response. We may call $\beta_i = f(x_i)$ where x_i is a univariate input indexing the n columns of X^*. The problem thus transforms from the estimation of a *simple* relationship between the output and complex, very high-dimensional structured inputs, to a problem of estimating the *complex* relationship between output and a simple input. Multiscale estimation of $f(\cdot)$ becomes a simplification tool as it may be assumed that $f(\cdot)$ is flat along possibly wide ranges of x. The special case in which $n = 2^L$ for some L and all the x_i are equally spaced is the typical scenario in which wavelet shrinkage is applied. Analogously to our previous treatment of wavelet regression, a Haar wavelet decomposition of $f(\cdot)$ in this case results in the recursive partitioning of the space of x into equally-sized blocks, along which the regression function is assumed constant. Wavelet thresholding thus involves shrinking the wavelet coefficients or setting them to zero. This technique can be used when the sample sizes are very large and owes its efficiency to the speed of the fast wavelet transform algorithm; it has been successfully applied as image compression or denoising tool (i.e. with a bivariate regressor). Although wavelets provide multiscale interpretations, one may question their applicability in settings that do not conform to their rather inflexible requirements. In fact, methods that similarly partition the covariate space but forgo multiscale decompositions – such as regression trees – have enjoyed tremendous success in both the classical and Bayesian nonparametric regression literature.

1.4 Relevant literature

There are many branches of the literature that are related to the problems we have outlined above. While we refer to the individual chapters for more in-depth treatments, we provide here a general overview.

The idea of merging groups of correlated predictors in regression emerged with partial least squares and principal components regression (Wold, 1966, 1975; Frank and Friedman, 1993), which use data-driven projections to effectively "whiten" the data and solve collinearity-related problems. These methods operate at a single measurement scale; when only considering the covariance structure among the predictors, the resulting linear combinations will be unrelated to scaling and resolution. Alternative methods for correlated predictors include ridge regression (Hoerl, 1962) and the elastic net (Zou and Hastie, 2005; Hesterberg et al., 2008). The Lasso of Tibshirani (1996); Hastie et al. (2009) was not created to tackle predictor collinearity, and it can be shown that it selects only one among many correlated predictors, even if all appear in the regression function. Analogous results hold for Bayesian variable selection models (Mitchell and Beauchamp, 1988; George and McCulloch, 1997; Ishwaran and Rao, 2005). This shortcoming can be tackled by explicitly taking into account predictor structure. For example, one can specify groups of predictors to be selected or dropped via the group Lasso (Yuan and Lin, 2006). Alternatively, groups can be adaptively estimated under the assumption of an ordered measuring grid via the fused Lasso (Tibshirani et al., 2005). Projection-based methods can also take advantage of penalization or shrinkage priors to induce sparsity (Zou et al., 2006; Bhattacharya and Dunson, 2011). Other grouping mechanisms have been developed by Park et al. (2007); Bühlmann et al. (2013). While these methods take into account or estimate some kind of predictor structure, they are not multiscale methods as they only operate at a single resolution.

A multiscale interpretation can be achieved by using wavelets, as mentioned above (Zhao et al., 2012; Antoniadis, 2007). Wavelet thresholding has been used in non-parametric regression settings in Donoho and Johnstone (1994, 1995). With highly dimensional predictors, the methods developed by Zhao et al. (2012), and from a Bayesian perspective Brown et al. (2001); Jeong et al. (2013), are more closely related to our problem and approach of Chapters 2 and 3. The authors use L1-penalties and variable selection priors on the wavelet coefficients to obtain a sparse representation of the linear regression function. The selected subset of regression coefficients allow to quantify resolution uncertainty and result in a mixed, "best" resolution. In these settings, and imposing a specific multiscale structure of dyadic splits of the grid S, the method we propose is asymptotically related to regression using Haar bases. Other methods for multiscale analysis are reviewed in (Ferreira and Lee, 2007), with a focus on dynamic models.

Structured high-dimensional predictors in regression can be used within the general framework of functional data analysis (FDA; Ramsay and Silverman 1997). Morris (2015) and Reiss et al. (2017) provide overviews of FDA methods for regression with functional predictors. Yao et al. (2005); Comte and Johannes (2012) apply functional data regression methods on longitudinal data. FDA methods typically require the researcher to choose a basis function representation of the data, which are assumed to arise from the discretization of underlying functions. Regression methods using basis function representations are not uncommon in the Bayesian literature (Gelman et al., 2014). Alternative methods with scalar response include clustering of curves, see for example Bigelow and Dunson (2012), who apply this concept on an early pregnancy study.

As we mentioned, wavelets are typically considered the standard multiscale methodology in nonparametric regression settings (Antoniadis, 2007). However, "scaling" is at the core of classical methods such as histogram regression (Anderson, 1966; Györfi et al., 2002), in which the major goal is determining block sizes. By the same logic, there is a connection to be made with methods such as decision trees (Breiman et al., 1984; Chipman et al., 1998). These models are not multiscale, but recursively partition the covariate space into blocks of decreasing size. Their flexibility and predictive performance have made them popular tools in regression, especially in more complex implementations that hinder their potential for interpretability (Breiman, 2001; Chipman et al., 2010).

The approach we introduce can be interpreted as a stepwise procedure that estimates the regression function via a coarse-to-fine combination of scales. Each scale is partly separate from the others, and assigned a specific model for its estimation. In other words, the overall model is built upon a sequence of *modules*. Each module corresponds to a Bayesian model, and output of preceding modules is used as input in subsequent ones. Because of this, the overall model is not fully Bayesian. Our modularization approach can be seen as a Bayesian extension of two-step *plug-in* procedures that are common in econometrics (Murphy and Topel, 1985). Increasing attention has recently been devoted to so-called modular procedures that break the dependence among components of a Bayesian hierarchical model, and to the cases in which this might be advantageous. Liu et al. (2009) motivate the use of modular procedures in computer models, where breaking dependence is seen as a necessity arising from potential misspecification. Jacob et al. (2017) provides an overview of how modularization can be useful in other contexts, whereas Gerard and Stephens (2018) apply a modular approach in genomics, and Chen and Dunson (2017) focus on modular techniques to screen predictors for regression. Our method uses modularization to solve the identifiability issues arising in simple multiscale regression settings. Other advantages of modularization include flexible specification of the priors for the parameters of interest. The computational overhead of modularization is small, and the overall complexity follows from the modules being implemented.

We introduce our general approach in Chapter 2, where we consider the case of a prespecified multiresolution structure in regression. After introducing our modular approach as a general procedure that can be used with modules of different kinds, we give some theoretical guarantees in the form of an asymptotic result, with the goal of providing a link to frequentist tools. We apply our model to simulated and real data implementing a simple algorithm. Motivated by the real data application considered in Chapter 2, we extend our modular approach to the estimation of unknown multiresolution structures in Chapter 3, by taking advantage of properties of the modular posterior distribution. We introduce a modular and multiscale approach for regression of a scalar outcome on image predictors, and revisit our previous real data analysis application in this new light. We thus go a step further and create new algorithms that take advantage of the simple form of the modular posterior distribution when the modules are conjugate.

Finally, we consider in Chapter 4 the classical problem of nonparametric regression, and introduce a Bayesian modular approach which produces interpretable results by modeling the regression function as an additive decomposition of regressor and resolution contributions, implemented by an ensemble of multiscale trees. Our methodology hierarchically sorts effects of covariates on the output from coarse to fine scales, interpreted

as major and minor effects, respectively. We provide simulated and real world data applications to compare our model with state-of-the-art methods and show its predictive performance, along with its ability to produce interpretable “stylized facts” about the relationship between output and inputs.

Chapter 2

Bayesian modular and multiscale regression

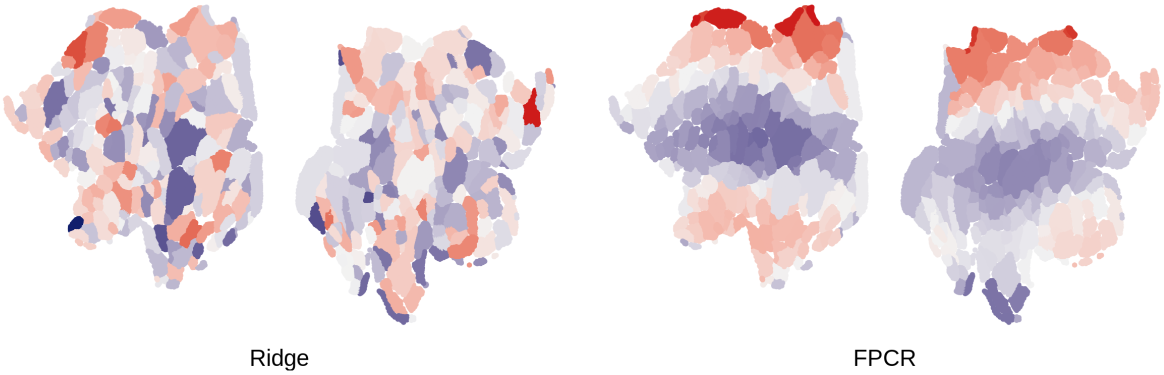


Figure 2.1: Brain hemispheres are parcellated in 333 regions according to Gordon et al. (2016) and used in a gender classification task performed via Ridge regression and functional principal components regression (FPCR). These two models achieve similar out-of-sample accuracy (Table 2.1), but it is challenging to reconcile interpretations arising from their estimates, as shown above.

2.1 Introduction

Researchers routinely collect very high-dimensional data that are spatially and/or temporally indexed, with the intention of using them as inputs in regression-type problems. For example, imaging or biomonitoring data may be collected for each patient, with the goal being prediction and understanding of relationships with health outcomes. In these cases, obtaining clear interpretation and accurate prediction simultaneously is notoriously difficult. In particular, challenges arise due to tendencies for adjacent measurement locations to be highly correlated, with huge numbers of measurements at a very high resolution.

In these settings, directly inputing such data into usual regression methods leads to poor results. Methods for dimensionality reduction that do not take advantage of predictor structure can have poor performance in estimating regression coefficients and sparsity patterns when the dimension is huge and predictors are highly correlated. Theoretical guarantees in such settings typically rely on strong assumptions on sparsity, low linear dependence, and high signal-to-noise (Zhao and Yu 2006; Wasserman and Roeder 2009; Bühlmann and van de Geer 2011, Chapter 7; Scarlett and Cevher 2017).

The above problems can be alleviated by down-sampling the data to lower resolutions before analysis. This is an appealing option because of the potential for huge dimensionality reduction. However, any specific resolution choice might be perceived as ad-hoc, and hide patterns at different scales that could instead be highlighted by a multiresolution approach. This problem cannot be solved by methods that somewhat take into account predictor structure, but only act on a single measurement scale, like the group Lasso of Yuan and Lin (2006), the fused Lasso of Tibshirani et al. (2005), the Bayesian method of Li and Zhang (2010), and data-driven projection approaches such as PCR (Delaigle and Hall, 2012).

Alternatively, one can use methods for “functional predictors" (Ramsay and Silverman, 1997; Morris, 2015; Reiss et al., 2017), which view temporally- or spatially-indexed predictors as corresponding to realizations of a function. Typically this involves estimating a time or spatially-indexed coefficient function, which is often represented as a linear combination of basis functions. In this context, a multiresolution decomposition of the predictors measures local, fine-scale effects, along with wider-range, coarse-scale ones. Achieving this via wavelets requires the specification of orthogonal basis functions; this leads to benefits in terms of identifiability and performance in estimating the individual coefficients, while also leading to disadvantages relative to “over-complete" specifications (Donoho and Elad, 2003; Mairal et al., 2009). Furthermore, traditional wavelet analysis specifies resolutions inflexibly and cannot be easily adapted to cases in which there is uncertainty on multiple pre-specified resolutions that do not conform to a wavelet decomposition (e.g. brain hemispheres/lobes/regions), or when the goal of analysis is to automatically and flexibly group the predictors to reduce dimension and achieve reproducible and interpretable results (Thirion et al., 2014; Schiffler et al., 2017). For classical references on wavelet regression, see Donoho and Johnstone (1994, 1995); Zhao et al. (2012); from a Bayesian perspective, see e.g. Brown et al. (2001) or Jeong et al. (2013).

To solve these issues, we propose a class of Bayesian modular and multiscale regression methods (*BM&Ms*), which express the regression function as an additive expansion of functions of data at increasing resolutions. In its simplest form, the regression function becomes an additive expansion of coarse to fine step functions. This implies that multiple down-samplings of the predictor are included within a single flexible multiresolution model. Our approach can be used when (1) there is a pre-determined multiscale structure, or uncertainty on a multiplicity of pre-specified resolutions, as in the case of brain atlases; (2) with temporally- or spatially-indexed predictors, when the goal is a multiscale interpretation of single-scale data. In the first case, our method can be directly used to ascertain the contribution of the pre-determined scales to the regression function. In the second case, *BM&Ms* are related to a Haar wavelet expansion but involve a simpler, non-orthogonal transformation that facilitates easy interpretation, and suggests a straightforward extension to scalar-on-tensor regression. We address the identifiability issues induced by this non-orthogonal expansion by taking a modularization approach. The resulting *BM&M* regression is stable, well identified, easily interpretable, and provides uncertainty quantification at different resolutions. We defer the treatment of temporally- or spatially-indexed predictors to Chapter 3. Here, we focus on multiscale regression settings in which the resolutions are specified a priori and are not assumed to have a spatiotemporal indexing.

The idea of modularization in Bayesian inference is that instead of using a fully Bayesian joint probability model for all components of the model, one "cuts" the dependence between different components or modules (see Liu et al. (2009); Jacob et al. (2017) and references therein). Modularization has been commonly applied in joint modeling contexts for computational and robustness reasons; latent factor models can be made more robust to model misspecification by cutting the dependence of the predictor factors on the response. Motivated by the practical improvements attributable to such an approach, probabilistic programming software WinBUGS has incorporated a `cut(.)` function to allow cutting of dependence in routine Bayesian inference (Plummer, 2015).

Our multiscale setting is different than previous work on Bayesian modularization, in that we use modules to combine information from the same data at increasingly higher resolutions. We introduce *BM&Ms* in a general setting in Section 2.2, highlighting the links with coherent Bayesian modeling. Section 2.4 focuses on linear regression. Section 2.6 outlines an algorithm to sample from the modular posterior distribution, and presents applications on simulated data and to a brain imaging classification task. Proofs and technical details are included in an Appendix.

2.2 A modular approach for multiscale regression

We consider a regression problem linking a scalar response y_i to an input vector $\mathbf{x}_i = (x_{s,i})_{s \in S}$ of dimension $d \gg n$, for each subject $i \in \{1, \dots, n\}$. For example, a subject's health outcome may be linked to the output from a high-resolution recording device. The goal is to predict the outcome variable and explain its variability across subjects by identifying specific patterns in the sensor recording at different resolutions.

We denote the vector of responses by y and the raw data matrix by X_S, where S is the resolution of the raw data. If predictors are temporally indexed, then S corresponds to the frequency of such time series; if they are spatially-indexed, then S corresponds to the area of each sampling unit. It is also relevant to speak of resolution if predictors can be partitioned into groups with an underlying meaning. For example, alternative brain parcellations assign group memberships to voxels based on brain functions in addition to taking into account their spatial location. In the case of product categories there may be no spatiotemporal indexing to speak of, but we may still refer as their alternative categorizations as resolutions. Therefore in these terms, we assume that each row of X_S can be down-sampled to get a new design matrix X_{S_j}, where S_j is a lower resolution grid such that $d_j = |S_j| < |S| = d$. Down-sampling can be achieved by summing or averaging adjacent columns, or subsetting them. Researchers might be interested in using down-sampled data to ease implementation and interpretation of standard models, but any such choice is arbitrary, as we have seen in Section 1.1.

We simplify the notation slightly by calling $X = X_S$ and $X_j = X_{S_j}$, and consider the same data at increasing resolutions in an additive model:

$$y = f_1(X_1) + \cdots + f_j(X_j) + \cdots + f_K(X_K) + \varepsilon, \tag{2.1}$$

where f_j is the resolution j contribution to the regression function. With this additive multiresolution expansion, it is difficult to disambiguate the impact of the coarse scales from the finer ones, leading to identifiability issues. If we were to attempt fully Bayes inferences by placing priors on the different component functions, large posterior dependence would lead to substantial computational problems (e.g. very poor mixing of Markov chain Monte Carlo algorithms) and it would not be possible to reliably interpret the different f_js. This happens in particular if each f_j is linear, as seen in Section 2.4.

We bypass these problems by adopting a modular approach, splitting model (2.1) into components or *modules* which are kept partly separate from each other, to get a *modular posterior*. In the following, for $j > i$ we use the notation $A_{i:j} = \{A_i, \dots, A_j\}$ and omit the dependence on X_j to simplify the notation.

Definition 2.2.1. Within the overall model (2.1), *module* j for data at resolution S_j consists of a prior for f_j, and a model for y, conditionally on $f_{1:j}$:

$$f_j | f_{1:j-1} \sim \pi(f_j | f_{1:j-1}) \qquad y | f_j, f_{1:j-1} \sim p_j(y | f_j, f_{1:j-1}),$$

where model p_j assumes the relationship $y = f_1(X_1) + \cdots + f_{j-1}(X_{j-1}) + f_j(X_j) + \varepsilon_j$, where $f_1, \dots, f_{j-1}$ are considered known. The output from module j is the (conditional) posterior distribution for f_j obtained by the prior distribution $\pi(f_j \mid f_{1:j-1})$ coupled with model p_j, and we denote it by $\pi_M(f_j | y, f_{1:j-1})$:

$$\pi_M(f_j | f_{1:j-1}, y) = \frac{\pi(f_j | f_{1:j-1}) p_j(y | f_{1:j})}{p_j(y | f_{1:j-1})}. \tag{2.2}$$

Thus for the full model (2.1) we build K modules, using increasingly higher resolution data, with each module being a *refinement* on previous output.

Definition 2.2.2. The modular prior distribution for $\mathbf{f} = (f_1, \dots, f_K)$ corresponds to $\pi_M(\mathbf{f}) = \pi(f_1) \cdots$
$\cdots \pi(f_j | f_{1:j-1}) \cdots \pi(f_K | f_{1:K-1})$, whereas the modular posterior distribution $\pi_M(\mathbf{f} | y)$ is:

$$\pi_M(\mathbf{f} | y) = \pi_M(f_1 \mid y) \cdots \pi_M(f_j | y, f_{1:j-1}) \cdots \pi_M(f_K | y, f_{1:K-1}).$$

The modular posterior distribution collects the posteriors in (2.2). Each module refines the output from previous modules by using higher resolution data, and the modular posterior is obtained by aggregating all refinements. Modularity is evidenced by resolution dependence, which is only allowed downwardly, as opposed to letting it be bidirectional as in a fully Bayes approach. Therefore, in the multiscale model outlined above our modular approach is not Bayes-coherent.

2.3 Connections with standard Bayesian inference

We have introduced a modular method above that updates (conditional) prior distributions on scale contributions by using multiple models. This way of updating prior distributions is non-standard, and it is therefore relevant to determine which models and prior distributions lead to a Bayesian posterior distribution that coincides with the modular posterior distribution, if this connection can be made.

In a fully Bayesian setting, a *single* model and joint prior distribution must be assumed for coherency. Thus, consider the multiscale regression model

$$y = f_1(X_1) + f_2(X_2) + \varepsilon \text{ where } \varepsilon \sim N(0, \sigma^2 I_n) \text{ and } \sigma^2 \text{ is known.} \tag{2.3}$$

Assign a prior $\pi(f_1, f_2) = \pi(f_1)\pi(f_2 \mid f_1)$ to the resolution contributions; coupled with the model $p(y \mid f_1, f_2)$ (where we omit the dependence on X_j, $j = 1, 2$ for simplicity), we derive the standard Bayesian posterior distribution as

$$\begin{aligned} d\Pi(f_1, f_2 \mid y) &= d\Pi(f_1 \mid y) d\Pi(f_2 \mid f_1, y) \\ &= \frac{d\Pi(f_1) \int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)}{\iint p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1) d\Pi(f_1)} \frac{d\Pi(f_2 \mid f_1) p(y \mid f_1, f_2)}{\int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)}. \end{aligned} \tag{2.4}$$

The above result highlights a two-stage Bayesian update scheme to obtain the posterior distribution $\pi(f_1, f_2 \mid y)$. We now use this decomposition of the posterior distribution to show (1) which setup of our modular approach achieves Bayesian coherency; (2) which setup of a Bayesian model corresponds to the modular posterior distribution.

Partial Bayesian coherency of the modular posterior distribution

Proposition 2.3.1. In model (2.3), the modular posterior distribution achieves full Bayesian coherency if and only if

$$p_1(y \mid f_1) = \int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)$$

and $p_2(y \mid f_1, f_2) = p(y \mid f_1, f_2)$, where $p(y \mid f_1, f_2)$ is the model corresponding to (2.3).

Proof. The result follows from Definition 2.2.2. In fact, the modular posterior distribution is factored as:

$$\begin{aligned} d\Pi_M(f_1, f_2 \mid y) &= d\Pi_M(f_1 \mid y) d\Pi_M(f_2 \mid f_1, y) \\ &= \frac{d\Pi(f_1) p_1(y \mid f_1)}{\int p_1(y \mid f_1) d\Pi(f_1)} \frac{d\Pi(f_2 \mid f_1) p_2(y \mid f_1, f_2)}{\int p_2(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)}, \end{aligned}$$

where $p_1(y \mid f_1)$ models the relationship between the output y and f_1 alone. Therefore, we obtain the Bayesian posterior of equation (2.4) only by choosing $p_1(y \mid f_1) = \int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)$ and $p_2(y \mid f_1, f_2) = p(y \mid f_1, f_2)$. □

Such a choice for p_1, while fully Bayesian, would negate the advantages of modularization, as it requires averaging f_2 from the full model $p(y \mid f_1, f_2)$. Other choices for p_1, instead, result in only partial Bayesian coherency. In fact, any modular posterior distribution $d\Pi_M(f_j \mid y)$ is the Bayesian update of a prior distribution with a model p_j, but this model does not in general correspond to the overall model. Lastly, we show that rescaling a Bayesian prior by a data-dependent factor is equivalent to a modular update.

Modularity via rescaled data-dependent prior distributions

The reverse problem is to obtain the modular posterior by means of specific choices of model/prior pairs.

Proposition 2.3.2. In model (2.3), consider a rescaled, data-dependent prior distribution $d\tilde{\Pi}(f_1, f_2) = d\tilde{\Pi}(f_1) d\Pi(f_2 \mid f_1)$, where

$$d\tilde{\Pi}(f_1) = \frac{d\Pi(f_1) p_1(y \mid f_1)}{\int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)}.$$

Then, Bayesian and modular posterior distributions coincide if $p_2(\cdot) = p(\cdot)$.

Proof. Considering (2.4), Bayesian and modular posterior distributions of $f_2 \mid f_1$ coincide if $p_2(\cdot) = p(\cdot)$. We are thus concerned with the posterior distributions on f_1. Updating the (rescaled) prior distribution $d\tilde{\Pi}(f_1)$ via model $p(\cdot)$, we obtain

$$\begin{aligned} \tilde{\pi}(f_1 \mid y) &= \frac{\tilde{\pi}(f_1) \int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)}{\iint p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1) d\tilde{\Pi}(f_1)} \\ &= \frac{d\Pi(f_1) \frac{p_1(y|f_1)}{\int p(y|f_1, f_2) d\Pi(f_2|f_1)} \int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)}{\iint p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1) \frac{p_1(y|f_1)}{\int p(y|f_1, f_2) d\Pi(f_2|f_1)} d\Pi(f_1)} \\ &= \frac{\pi(f_1) p_1(y \mid f_1)}{\int p_1(y \mid f_1) d\Pi(f_1)}, \end{aligned}$$

which coincides with the modular posterior distribution for f_1. □

Note that $\tilde{\pi}(f_1)$ is not necessarily a proper distribution, and it depends on the data via a scaling factor. As previously mentioned, this is not a practical choice as the scaling factor involves the marginalization over f_2. To reiterate the previous point, the scaling factor can be erased by choosing $p_1(y \mid f_1) = \int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1)$.

Example

Consider the setting in which the regression function is measured with Gaussian noise, and assume $\pi(f_2 \mid f_1) = N(0, \sigma^2\tau)$. From the full model we find the fully Bayesian first "module" by $\int p(y \mid f_1, f_2) d\Pi(f_2 \mid f_1) = \int N(y; f_1 + f_2, \sigma^2 I_n) N(f_2; 0, \sigma^2\tau) df_2 = N(y; f_1, \sigma^2(I_n + \tau))$. In other words f_1 is updated via $y = f_1(X_1) + u$, where $u \sim N(0, \sigma^2(I_n + \tau))$. The second fully Bayesian "module" uses the conditional prior $\pi(f_2 \mid f_1)$ specified previously, on the full model $p(y \mid f_1, f_2)$. This seemingly appealing modeling strategy has the advantage of proceeding in steps to isolate the low-resolution parameter. However, its disadvantages become clear once f_j's and τ take a more specific form and dependence on the multiresolution data matrices is made explicit. We analyze in depth these shortcomings in Section 2.4.2, where we show how full Bayesian inference may fail in linear multiscale regression.

2.4 *BM&Ms* for linear regression

We now assume that $f_j = X_j\theta_j$ and our goal is to study θ_j for $j = 1, \dots, K$ in the model

$$y = X_1\theta_1 + \cdots + X_K\theta_K + \varepsilon, \tag{2.5}$$

where $\varepsilon \sim N(0, \sigma^2)$, each of the K data matrices are such that $X_j = X_{j+1}L_j$ and $X_K = XL_K$. In this section we assume that σ^2 is known and that $(L_1, \dots, L_K)$ are predetermined. However, we will later consider unknown scale-specific σ_j^2 and unknown L_j. The $d_{j+1} \times d_j$ matrix $L_j = [L_j^1 \cdots L_j^{d_j}]$ can be simply built as a collection of vectors with binary entries. Each $L_j^{(l)}, l = 1, \dots, d_j$ thus indexes the columns of X_{j+1} to be grouped via summation. For example, if $d = 2^r$ for some r, a simple sequence of nested dyadic down-sampling is generated by $L_j = I_{2^{j-1}} \otimes \mathbf{1}_{d/2^{j-2}}$, where I_v and $\mathbf{1}_v$ are the identity matrix of size v and a size-v vector of ones, respectively. With the above assumptions, the multiresolution regression model in (2.5) can be written as:

$$y = X\left(\mathcal{L}_1\theta_1 + \cdots + \mathcal{L}_K\theta_K\right) + \varepsilon = X\boldsymbol{\mathcal{L}}\boldsymbol{\theta} + \varepsilon = X\beta_K + \varepsilon, \tag{2.6}$$

where $\mathcal{L}_j = \prod_{s \leq j} L_s$, $\boldsymbol{\mathcal{L}} = [\mathcal{L}_1, \ldots, \mathcal{L}_K]$, $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_K]'$. We thus decompose the regression coefficient vector $\beta_K = [\beta_K^{(1)} \cdots \beta_K^{(d)}]$ in resolution contributions. For $r = 1, \ldots, d, \beta_K^{(r)} = \sum_{j=1}^{K} \sum_{s=1}^{d_j} \theta_j^{(s)} \mathbf{1}\left(\mathcal{L}_j^{(s)} = 1\right)$, where $\mathbf{1}(A)$ is the indicator function at A. Therefore, the multiresolution decomposition of β_K involves $\sum_j d_j = d^*$ parameters, which is of the same order of magnitude of $d_K \leq d$. In Section 3.2.1 we consider the special interpretations that arise if the columns in X are temporally- or spatially-indexed.

2.4.1 Identifiability of the multiscale parameters

With the goal of estimating each θ_j separately, we can effectively write model (2.5) as:

$$y = \Big[X_1 \cdots X_K\Big] \Big[\theta_1 \cdots \theta_K\Big]' + \varepsilon,$$

but note that the effective design matrix $X := [X_1 \cdots X_K]$ is such that $det(X'X) = 0$, as the columns of X_j are linear combinations of columns of X_{j+1}. This means that infinitely many values of $\theta_1, \ldots, \theta_K$ share the same likelihood. To see this, suppose $p = 2$ and $K = 2$, i.e. we have only two measurements for each observation $i = 1, \ldots, n$. Then, fix $\mathcal{L}_1 = (1\ 1)'$ and $\mathcal{L}_2 = I_2$ as our down-sampling operators. The multiscale regression model is $y = X_2(\mathcal{L}_1\theta_1 + \mathcal{L}_2\theta_2) + \varepsilon$. We have $\theta_1 \in \mathbb{R}$ and $\theta_2 = (\theta_{2,1}, \theta_{2,2}) \in \mathbb{R}^2$. Then construct $(\theta_1^*, \theta_2^*) \neq (\theta_1, \theta_2)$ such that for $c \in \mathbb{R}$, $\theta_1^* = \theta_1 + c$ and $\theta_{2,j}^* = \theta_{2,j} - c$, $j = 1, 2$. Clearly $(\mathcal{L}_1\theta_1 + \mathcal{L}_2\theta_2) = (\mathcal{L}_1\theta_1^* + \mathcal{L}_2\theta_2^*)$, hence the likelihood is flat for all c, y, X_1, X_2.

The model in equation (2.5) is therefore ill-posed, leading to problems with standard techniques for model fitting. In this case, there are ways which could be used to restore identifiability of the parameters. First, one could drop one of the merged columns *at each scale*. In the previous example, we could drop the first (or second) column of X_2, estimate $\theta_{2,1}$ (resp. $\theta_{2,2}$), and obtain an estimate of $\theta_{2,2}$ (resp. $\theta_{2,1}$) by $\theta_1 - \theta_{2,1}$ (resp. $\theta_1 - \theta_{2,2}$). In more general settings, one can devise an ad-hoc fixed orthogonal basis, or rely on existing methods such as wavelet transforms, which decompose β_K into resolution contributions by specifying a fixed orthogonal basis. As we mentioned in the introduction, wavelets are restrictive and cannot be applied to problems in which resolutions of the data are specified in advance not conforming to wavelet decompositions. Analogously, the specification a fixed ad-hoc orthogonal basis becomes increasingly more complicated with many scales and high-dimensional data, resulting in convoluted interpretation and post-processing. Even when this process can be automated, the ultimate goal is to consider $\boldsymbol{\mathcal{L}}$ as an unknown parameter to be estimated. We anticipate that such goal requires the implementation of Markov-chain Monte Carlo methods that would likely be affected negatively by these restrictive orthogonality requirements. By relinquishing orthogonality requirements, instead, we can devise simple and efficient proposal functions when implementing MCMC methods.

Alternatively, one may be able to obtain a well defined posterior distribution through informative prior distributions for θ. We outline this procedure in the next Section. As anticipated by the flat regions in the likelihood, the resulting posterior distribution will be strongly dependent on prior information.

2.4.2 Bayesian inference in linear multiscale regression

We have argued above that we may use model (2.5) for multiscale analysis when it is undesirable to impose restrictive orthogonality conditions. We now show how a coherent fully Bayesian model fails in this case. Consider model (2.5) and assign a prior $\pi(\theta_2 \mid \theta_1) = N(0, \sigma^2 V_2)$. The marginal posterior distribution of θ_1 is

$$\pi(\theta_1 \mid y) = \frac{d\Pi(\theta_1) \int p(y \mid \theta_1, \theta_2) d\Pi(\theta_2 \mid \theta_1)}{\iint p(y \mid \theta_1, \theta_2) d\Pi(\theta_2 \mid \theta_1) d\Pi(\theta_1)}.$$

In this particular case:

$$\begin{aligned}\int p(y \mid \theta_1, \theta_2) d\Pi(\theta_2 \mid \theta_1) &= \int N(y; X_1\theta_1 + X_2\theta_2, \sigma^2 I_n) N(\theta_2; 0, \sigma^2 V_2) d\theta_2 \\ &= N(y; X_1\theta_1, \sigma^2(I_n + X_2 V_2 X_2')).\end{aligned}$$

In other words, one can obtain the marginal posterior distribution of θ_1 in a fully Bayesian model by updating its prior $\pi(\theta_1) = N(\theta_1; m_1, V_1)$ via a "module" using $p_1(y \mid f_1) = N(y; X_1\theta_1, \sigma^2(I_n + X_2 V_2 X_2'))$. Call $V_y = I_n + X_2 V_2 X_2' = (I_n - X_2(V_2^{-1} + X_2'X_2)^{-1}X_2)^{-1}$. The posterior mean of θ_1 is thus $E(\theta_1 \mid y) = (V_1^{-1} + X_1'V_y^{-1}X_1)^{-1}(V_1^{-1}m_1 + X_1'V_y^{-1}y)$. Since we assume $X_1 = X_2 L_1$, columns at coarse scales are obtained as linear combinations of columns at finer scales. This realistic assumptions implies that

$$\begin{aligned}X_1'V_y^{-1}X_1 &= X_1'(I_n - X_2(V_2^{-1} + X_2'X_2)^{-1}X_2')X_1 \\ &= X_1'X_1 - X_1'X_2(V_2^{-1} + X_2'X_2)^{-1}X_2'X_1 \\ &= L_1'X_2'X_2L_1 - L_1'X_2'X_2(V_2^{-1} + X_2'X_2)^{-1}X_2'X_2L_1.\end{aligned}$$

This term is near zero if V_2^{-1} is "small." To more clearly see this, assume $V_2 = g(X_2'X_2)^{-1}$ corresponding to Zellner's g-prior for $\theta_2 \mid \theta_1$. Then $(V_2^{-1} + X_2'X_2)^{-1} = (\frac{1}{g}X_2'X_2 + X_2'X_2)^{-1} = \frac{g}{g+1}(X_2'X_2)^{-1}$. Large values of g corresponding to desirable high prior uncertainty lead to $V_1'V_y^{-1}X_1 \approx 0$, meaning that posterior inference at the coarse scale will be greatly influenced by its prior distribution. To directly solve this, one may use a highly-informative prior distribution on $\theta_2 \mid \theta_1$, but this is questionable as it moves the same kind of problem to the finer scale. Finally, our overall conclusions would not change if we assumed that $X_1 \approx X_2 L_1$, in which case the problem might only be slightly less severe. An even weaker assumption is that only for some s and k, $X_{1,s} \approx aX_{2,k} + bX_{2,k+1}$, where $X_{j,k}$ is the k^{th} column of X_j, corresponding to the case in which at least a column at low resolution can be obtained as a linear combination or aggregate of columns at high resolution. In this latter case, the problem would be localized to the affected columns. To bypass all of these issues, we modularize the posterior distribution.

2.4.3 Modularization of the posterior distribution

$\boldsymbol{\mathcal{L}}$ is overcomplete and non-orthogonal by construction and as we have seen above, this will lead to problems with standard techniques when fitting model (2.5). Our modularization approach of Section 2.2 results in a well defined *modular* posterior distribution:

$$\pi_M(\theta|y) = \frac{\pi(\theta_1)p_1(y|\theta_1)}{p_1(y)} \cdots \frac{\pi(\theta_K|\theta_{1:K-1})p_K(y|\theta_{1:K})}{p_K(y|\theta_{1:K-1})},$$

where $p_j(\cdot)$ is the likelihood for the response y under $y = \sum_{h=1}^{j} X_h\theta_h + \varepsilon$, $\varepsilon \sim N(0, \sigma^2)$, and $\theta_1, \dots, \theta_{j-1}$ are considered known. The posterior distribution for the coarsest scale coefficient θ_1 is derived treating all the finer scale coefficients as equal to zero; this makes θ_1 identifiable and interpretable as producing the best coarse scale approximation to the regression function. For $j > 1$, we can write $p_j(\cdot)$ as

$$y - \sum_{h=1}^{j-1} X_h\theta_h = X_j\theta_j + \varepsilon, \qquad \varepsilon \sim N(0, \sigma^2).$$

We are thus using the residuals $e_{j-1} = y - \sum_{h=0}^{j-1} X_h\theta_h$ as responses for the j^{th} module, X_j as design matrix, and $\theta_{1:j-1}$ from the previous modules' posterior distributions. Hence, the modular posterior for (2.5) is built using simpler, well-identified single-scale models as modules.

Proposition 2.4.1. Consider the modularization of (2.5) in K modules, with known σ^2 and fixed resolutions $\boldsymbol{\mathcal{L}}$. For $j = 1, \dots, K$, we consider $\theta_j \mid \sigma^2 \sim N(m_j, \sigma^2 V_j)$ and $p_j(\cdot \mid \theta_{1:j-1}, \sigma^2) = N(X_{1:j-1}\theta_{1:j-1} + X_j\theta_j, \sigma^2 I_n)$.
Then for $1 < j \leq K$, the modular posterior of $\theta_{1:j}$ is $\pi_M(\theta_j, \theta_{1:j-1} \mid y, \sigma^2) = N(\mu_{1:j}, \sigma^2\Sigma_{1:j})$ where

$$\mu_{1:j} = \begin{bmatrix} \mu_{1:j-1} \\ \mu_j \end{bmatrix} = \begin{bmatrix} \mu_{\beta_{1:j-1}} \\ \mu_{\beta_j} - Q_{1:j-1}\mu_{\beta_{1:j-1}} \end{bmatrix} \tag{2.7}$$

$$\Sigma_{1:j} = \begin{bmatrix} \Sigma_{1:j-1} & -\Sigma_{1:j-1}Q'_{1:j-1} \\ -Q_{1:j-1}\Sigma_{1:j-1} & \Sigma_j + Q_{1:j-1}\Sigma_{1:j-1}Q'_{1:j-1} \end{bmatrix}, \tag{2.8}$$

with $Q_{1:j-1} = \Sigma_j X_j' X_{1:j-1}$, and $\mu_{\beta_s} = \Sigma_s^{-1}(V_s m_s + X_s' y)$ for $s = 1, \dots, K$ the posterior mean we would obtain from the single resolution model $y = X_s\beta_s + \varepsilon_s$ with priors $\beta_s \sim N(m_s, \sigma^2 V_s)$.

Proof. Suppose $\pi_M(\theta_{1:j-1} \mid y, \sigma^2) = N(\mu_{1:j-1}, \sigma^2\Sigma_{1:j-1})$. Then module j involves the derivation of the posterior distribution of $\theta_j \mid \theta_{1:j-1}$ from the model $y - X_{1:j-1}\theta_{1:j-1} = X_j\theta_j + \varepsilon$, $\varepsilon \sim N(0, \sigma^2 I_n)$. With conjugate priors, we obtain $\pi_M(\theta_j \mid y, \theta_{1:j-1}, \sigma^2) = N(\mu_j, \sigma^2\Sigma_j)$, where $\Sigma_j = (V_j^{-1} + X_j'X_j)^{-1}$ and $\mu_j = \Sigma_j(V_j^{-1}m_j + X_j'(y - X_{1:j-1}\theta_{1:j-1})) = \mu_{\beta_j} - Q_{1:j-1}\theta_{1:j-1}$. The final result is then obtained by applying the properties of the Normal distribution. □

2.4.4 Asymptotics of *BM&Ms* in linear regression

The goal of this section is to study frequentist asymptotic properties of the *BM&Ms* posterior. We assume that (y, X) are i.i.d. and generated according to a process P_0 such that

$$\frac{1}{n}\begin{bmatrix} y'y & y'X \\ X'y & X'X \end{bmatrix} \xrightarrow[a.s.]{} \begin{bmatrix} \omega_{yy} & \omega_{y\mathbf{x}} \\ \omega_{\mathbf{x}y} & \Omega \end{bmatrix} = \mathbf{\Omega},$$

with $\mathbf{\Omega}$ positive definite. We then assume that $p_0(y|X, \sigma^2) = N(Xb, \sigma^2 I_n)$ is the regression model corresponding to P_0, where $b \in \mathbb{R}^p$ with dimension not depending on the sample size and σ^2 is known. We consider input data at a finite set of predetermined resolutions $X_1, \ldots, X_K$ as defined previously. For simplicity and without loss of generality we consider $K = 2$.

Proposition 2.4.2. The modular estimator (μ_1, μ_2) is consistent for $\mu^* = (\beta_1^*, \beta_2^* - L_1\beta_1^*)$, where β_j^* is the pseudo-true value of b under the model $y = X_j\beta_j + \varepsilon$. Furthermore, if $\underline{\theta} \sim \pi_M(\theta \mid y, X_{1:2})$ then $\sqrt{n}(\underline{\theta} - \mu) \xrightarrow{\text{d}} N(0, \Sigma^*)$, where

$$\Sigma^* = \begin{bmatrix} \Omega_1^{-1} & -\Omega_1^{-1}L_1' \\ -L_1\Omega_1^{-1} & \Omega_2^{-1} + L_1\Omega_1^{-1}L_1' \end{bmatrix}$$

and $\Omega_j = \mathcal{L}_j\Omega\mathcal{L}_j' = \mathcal{L}_jX'X\mathcal{L}_j'$ for $j = 1, 2$.

Proof. Consider a regression model p_j represented by $y = X_j\beta_j + \varepsilon_j$ where $\varepsilon \sim N(0, \sigma^2 I_n)$ and $X_j = X\mathcal{L}_j$. Then

$$\begin{aligned} E_{P_0}\log\frac{p_0}{p_j} &\propto E_{P_0}\left(-(y - Xb)'(y - Xb) + (y - X_j\beta_j)'(y - X_j\beta_j)\right) \\ &= c + E_{P_0}\left((y - X_j\beta_j)'(y - X_j\beta_j)\right) \\ &= c + E_{P_0}\left((y - Xb + Xb - X_j\beta_j)'(y - Xb + Xb - X_j\beta_j)\right) \\ &= c + (Xb - X_j\beta_j)'E_{P_0}\left(y - Xb\right) + E_{P_0}\left(y - Xb\right)'\left(Xb - X_j\beta_j\right) \\ &\qquad + (Xb - X_j\beta_j)'(Xb - X_j\beta_j) \\ &= c + (Xb - X_j\beta_j)'(Xb - X_j\beta_j) \end{aligned}$$

The pseudo-true value of b using model p_j is thus

$$\beta_{j,(n)}^* = \inf_{\beta_j} E_{P_0}\log\frac{p_0}{p_j}$$

$$
\begin{aligned}
&= (X_j'X_j)^{-1}X_j'Xb \\
&= (\mathcal{L}_jX'X\mathcal{L}_j)^{-1}\mathcal{L}_jX'Xb \longrightarrow_{n\to\infty} (\mathcal{L}_j\Omega\mathcal{L}_j)^{-1}\mathcal{L}_j\Omega b =: \beta_j^*.
\end{aligned}
$$

With conjugate priors and normal modules with known variance σ^2 we can directly take the modular posterior distribution in (2.4.1). Hence take $\mu_{1:2}$ as defined in (2.4.1). By the law of large numbers we obtain:

$$
\begin{aligned}
\mu_1 - \beta_1^* &= \mu_{\beta_1} - \beta_1^* \\
&= (V_1^{-1} + X_1'X_1)^{-1}(V_1^{-1}m_1 + X_1'y) - \beta_1^* \\
&= (V_1^{-1} + \mathcal{L}_1'X'X\mathcal{L}_1)^{-1}(V_1^{-1}m_1 + \mathcal{L}_1'X'y) - \beta_1^* \\
&= (\frac{1}{n}V_1^{-1} + \frac{1}{n}\mathcal{L}_1'X'X\mathcal{L}_1)^{-1}(\frac{1}{n}V_1^{-1}m_1 + \frac{1}{n}\mathcal{L}_1'X'(Xb+\varepsilon)) - \beta_1^* \\
&\to_{n\to\infty} E_{P_0}(\beta_{1,(n)}^*) - \beta_1^* + E_{P_0}((V_1^{-1} + \mathcal{L}_1'X'X\mathcal{L}_1)^{-1}(V_1^{-1}m_1 + \mathcal{L}_1'X'\varepsilon)) \\
&= 0
\end{aligned}
$$

Hence μ_1 is consistent for β_1^*. Subsequently, noting that $Q_1 = \Sigma_2X_2'X_1 = \Sigma_2X_2'X_2L_1 \to_{n\to\infty} L_1$, we have:

$$
\begin{aligned}
\mu_2 - \beta_2^* + L_1\beta_1^* &= \mu_{\beta_2} - Q_1\mu_{\beta_1} \\
&= (\frac{1}{n}V_2^{-1} + \frac{1}{n}X_2'X_2)^{-1}(\frac{1}{n}V_2^{-1}m_2 + \frac{1}{n}X_2'y) \\
&\qquad - Q_1(\frac{1}{n}V_1^{-1} + \frac{1}{n}X_1'X_1)^{-1}(\frac{1}{n}V_1^{-1}m_1 + \frac{1}{n}X_1'y) - \beta_2^* + L_1\beta_1^* \\
&\to_{n\to\infty} E_{P_0}(\beta_{2,(n)}^*) - \beta_2^* - L_1E_{P_0}(\beta_{1,(n)}^*) + L_1\beta_1^* \\
&= 0.
\end{aligned}
$$

Finally consider $\sqrt{n}(\underline{\theta} - \mu^*)$ where $\underline{\theta} \sim N(\mu_{1:2}, \Sigma_{1:2})$. For $j = 1, 2$ and as $n \to \infty$ we have $n\Sigma_j = (\frac{1}{n}V_j^{-1} + \frac{1}{n}X_j'X_j)^{-1} = (\frac{1}{n}V_j^{-1} + \frac{1}{n}\mathcal{L}_j'X'X\mathcal{L}_j)^{-1} \to (\mathcal{L}_j'\Omega\mathcal{L}_j)^{-1} = \Omega_j^{-1}$. Hence $\sqrt{n}(\underline{\theta} - \mu^*) \overset{\text{d}}{\to} N(0, \Sigma^*)$. □

Notice the asymptotic correspondence of μ_1 and μ_2 with $\hat{\theta}_1$ and $\hat{\theta}_2$, where $\hat{\theta}_1 = \hat{\beta}_1$ and $\hat{\theta}_2$ is the least-squares estimator obtained by regressing $y - X_1\hat{\beta}_1$ on X_2. In smaller samples, uncertainty at a coarse scale is propagated forward to finer scales across multiple stages.

Corollary 2.4.1. The large sample distributions of θ_1 and $L_1\theta_1 + \theta_2$ are approximated by $N(\beta_1^*, \frac{\sigma^2}{n}\Omega_1^{-1})$ and $N(\beta_2^*, \frac{\sigma^2}{n}\Omega_2^{-1})$, respectively. In other words, accumulating the modular posterior mean components up to j results in a consistent and asymptotically efficient estimator for β_j^*.

Proof. This is a direct consequence of the properties of the Normal distribution. □

2.5 Computation of the modular posterior distribution

We sample from the modular posterior distribution of Definition 2.2.2 by sequentially sampling from each module.

```
for $t \in \{1, \dots, T\}$ do
    Start: Draw sample $f_1^{(t)}$ from the posterior distribution of module 1 i.e.
        $\pi_M(f_1 \mid y, X_1)$
    for $j \in \{2, \dots, K\}$ do
        Draw sample $f_j^{(t)}$ from the posterior distribution of module $j$
            $m(f_j|f^{(t)}_{1:j-1}) = \pi_M(f_j|f^{(t)}_{1:j-1}, y) = \pi_M(f_j|\mathbf{f}^{(t)}_{-j}, y))$    (2.9)
    end
end
```

Algorithm 1: Sampling $\left\{\mathbf{f}^{(t)}\right\}_{t \in \{1,\dots,T\}}$ from the modular posterior $\pi_M(\mathbf{f} \mid y, X_{1:K})$

In general, sampling from the posterior distribution of module j, i.e. $\pi_M(f_j \mid y, f^{(t)}_{1:j-1})$ depends on the particular choices of priors and models. Therefore, in a multiscale linear regression with conjugate priors as in Section 2.4 we can easily sample from each individual module taking advantage of (2.7). Sampling from the posterior distribution of module-specific parameters is also simplified by the overall modular dependence structure. For example, the low resolution error variance of a fully Bayesian "module" should coherently depend on the higher-resolutions. However, we have seen above that fully Bayesian analysis may fail in the multiscale setting in which we are interested. At the same time, the assumption we have made so far was of a single noise variance among all modules, and this may be unnecessarily restrictive. We may relax this assumption within a modular approach by building *BM&Ms* with scale-specific error variances, i.e. $p_j(y \mid \theta_{1:j-1}, \sigma^2_j) = N(X_{1:j-1}\theta_{1:j-1} + X_j\theta_j, \sigma^2_j I_n)$. Each σ^2_j will capture the model error variance σ^2 alongside the *constant* variance induced by omitted higher-resolutions, $\varepsilon_j \sim N(0, (\sigma^2 + \sigma^2_{X_{j+1:K}})I_n)$ where $\sigma^2_{X_{j+1:K}} \in \mathbb{R}$. Sampling from the posterior distribution of σ^2_j in module j is achieved via routine methods. In classification problems, we may use probit or logit link functions. In both cases, the modular posterior is sampled from via a Gibbs sampler that alternates draws from a latent variable, given the linear predictor (as in Albert and Chib (1993) for probit regression, or as Polson et al. (2013) for logit), and from the linear predictor, given the latent variable. The latter step follows Algorithm 1 using the latent variable as response.

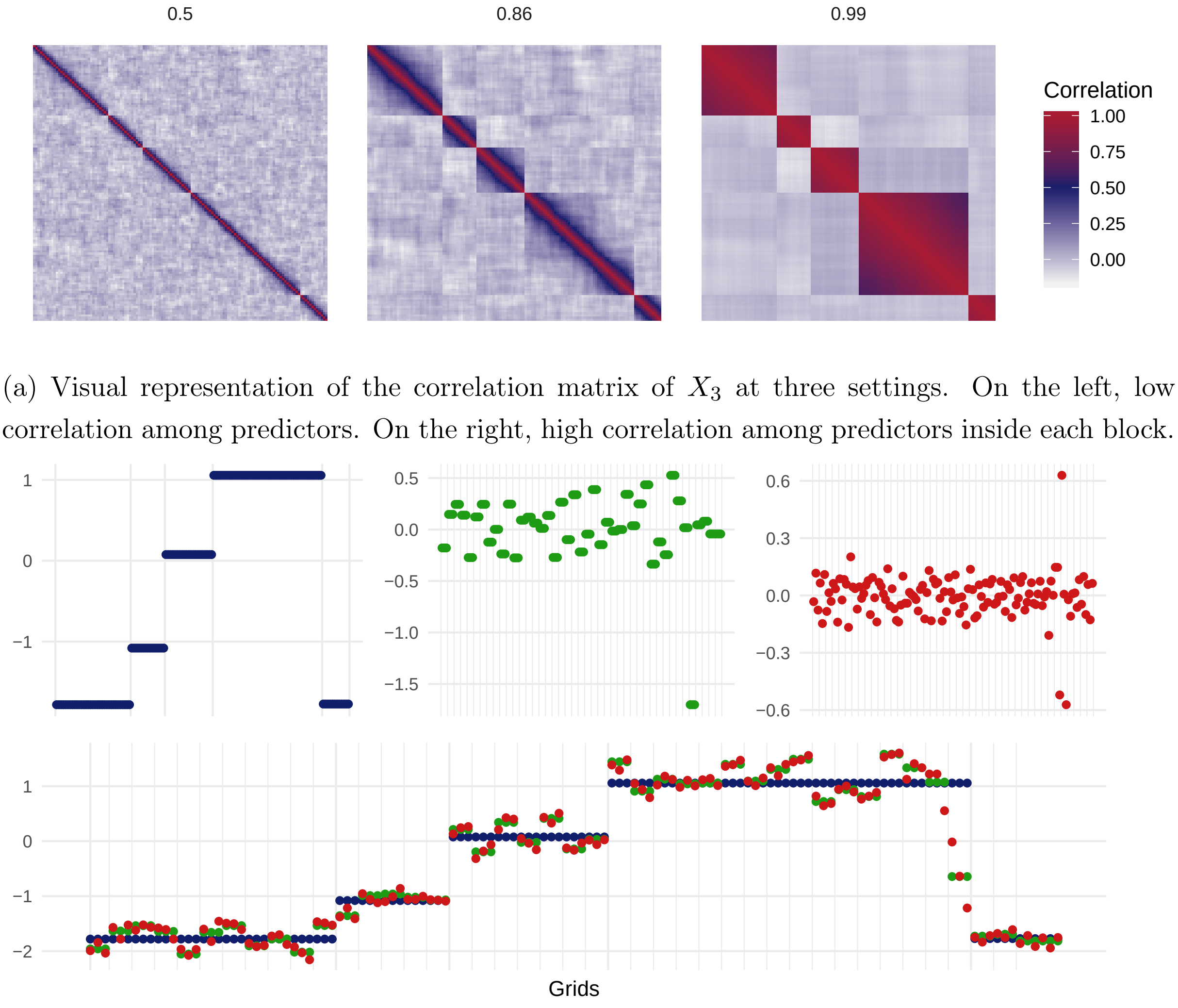
2.6 Applications

We now apply our methodology in a regression problem on synthetic data and on a classification task using real-world brain imaging data. In both cases, we assume that the resolutions are predetermined in advance. In Chapter 3, instead, we consider the case in which such resolution structure is unknown with the goal of estimating it using the data.

2.6.1 Simulation study

The goal of this section is to provide an illustration of multiscale regression on a scalar response. To this end, we generate data via the model $y = X_1\theta_1 + X_2\theta_2 + X_3\theta_3 + \varepsilon$, where $\sigma^2 = 1$, and $X_1 = X_3\mathcal{L}_1$, $X_2 = X_3\mathcal{L}_2$ are two down-samplings of the high-resolution data matrix X_3. The three matrices have dimensions $p_1 = 5, p_2 = 42, p_3 = 129$. To generate each dataset, we use $X_3 \sim N_n(0, \Sigma)$ where Σ is a block-diagonal matrix as shown in Figure 2.2a. The near-block-diagonal correlation suggests low resolution components are easily estimable. A parameter controls the correlation inside each block, making the estimation of higher-resolution contributions increasingly difficult. The values of θ_j corresponding to the true decomposition of the regression coefficient are shown in Figure 2.2b. Coupled with possibly high correlation among predictors, the high-resolution contributions will be difficult to estimate; we address the estimation of a multiresolution regression function via *BM&Ms* and competing models.

The decomposition $\beta = \mathcal{L}_1\theta_1 + \mathcal{L}_2\theta_2 + \theta_3$ is shown in Figure 2.2b. The regression relationship is largely, but not exhaustively, determined by the low-resolution component. The intermediate resolution provides finer detail, while the high-resolution data might only prove useful if there is either low correlation among regressors, or a large sample size. We could imagine X_1 to represent activity levels in p_1 brain lobes, while X_2 and X_3 represent activity at gradually smaller subregions. The regressors are *not* assumed to be temporally or spatially indexed. In figure 2.3, we cluster predictors by their θ_j values only for illustration purposes. However, there is an existing group-membership relationship among regressors (e.g. using the brain analogy, many regions belong to the same lobe). This prevents us from obtaining clear interpretation of the estimates from FDA models for linear regression, hence we postpone their comparison with *BM&Ms* until Chapter 3, where we tackle the problem of decomposing a regression function with temporally- and spatially-indexed predictors. We estimate the resolution contributions with *BM&Ms*, choosing Bayesian variable selection modules. In each module, we assign the prior $\theta_j \mid \sigma_j^2 \sim N(0, \sigma^2 g(X'_{\gamma_j,j}X_{\gamma_j,j})^{-1})$ where γ_j indexes the columns of X_j being selected (Marin and Robert, 2007), and fixing $g = n$. We compare *BM&Ms* to ridge and Lasso multiscale regression, where all resolutions are used jointly, and whose optimal penalty parameters are obtained via cross-validation. The goal is the estimation of



(a) Visual representation of the correlation matrix of X_3 at three settings. On the left, low correlation among predictors. On the right, high correlation among predictors inside each block.

(b) The coarse-to-fine scale contributions in blue, green, and red (above) generate the true coefficient vector β (in red below) when summed together.

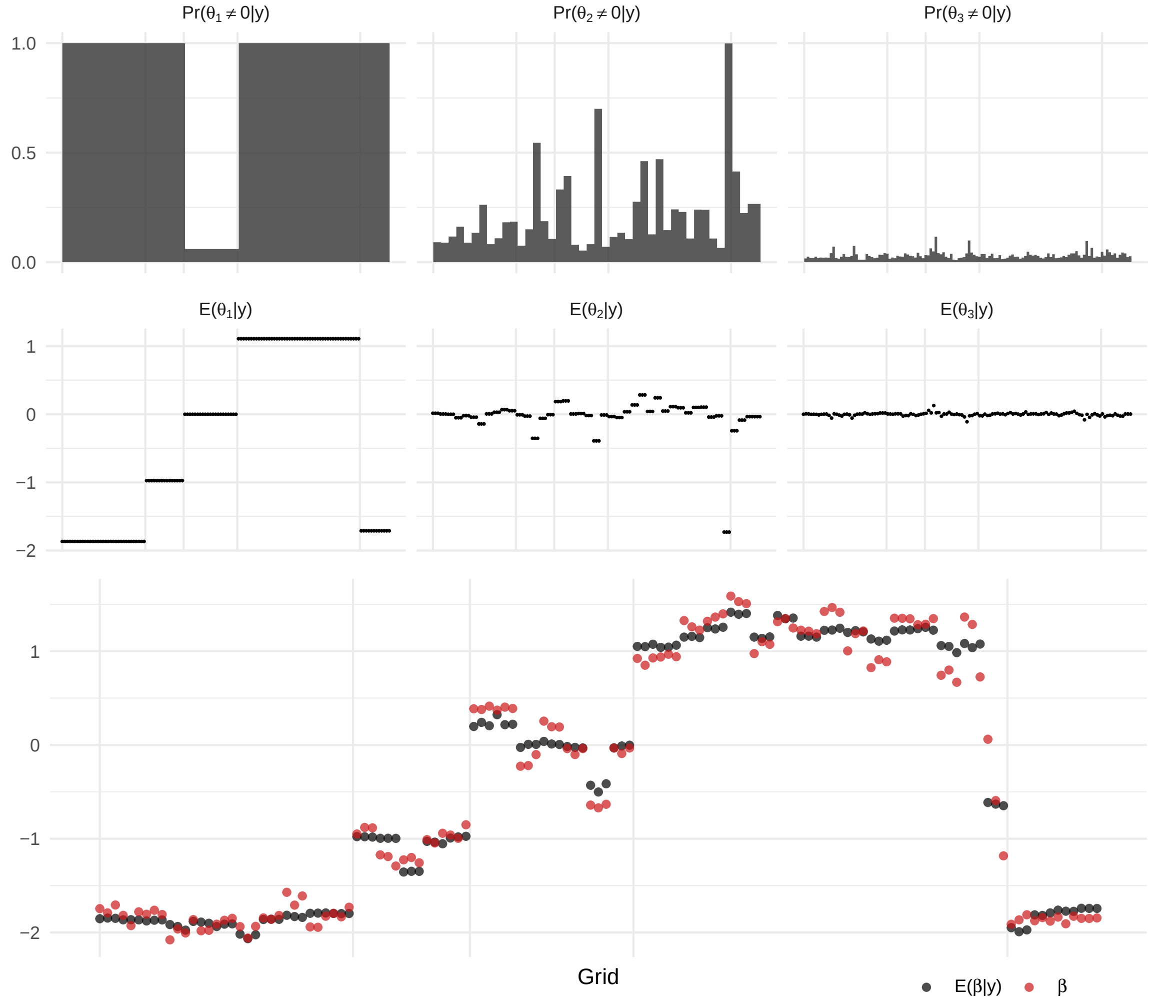


Figure 2.3: From the top, *BM&Ms* posterior selection probability and posterior means of the resolution contributions θ_j, approximated via MCMC. Bottom: posterior mean and true values of the overall regression coefficients β, approximated via MCMC.

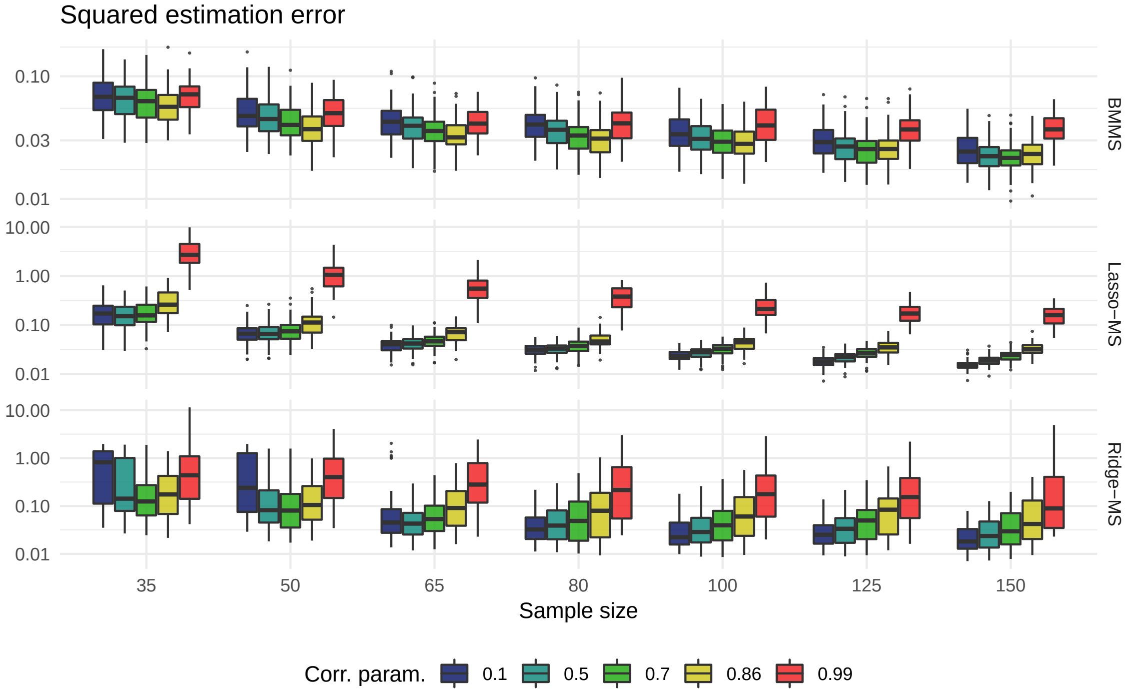


Figure 2.4: Squared errors in the estimation of β via multiresolution decomposition of the regression relationship, on 50 simulated datasets.

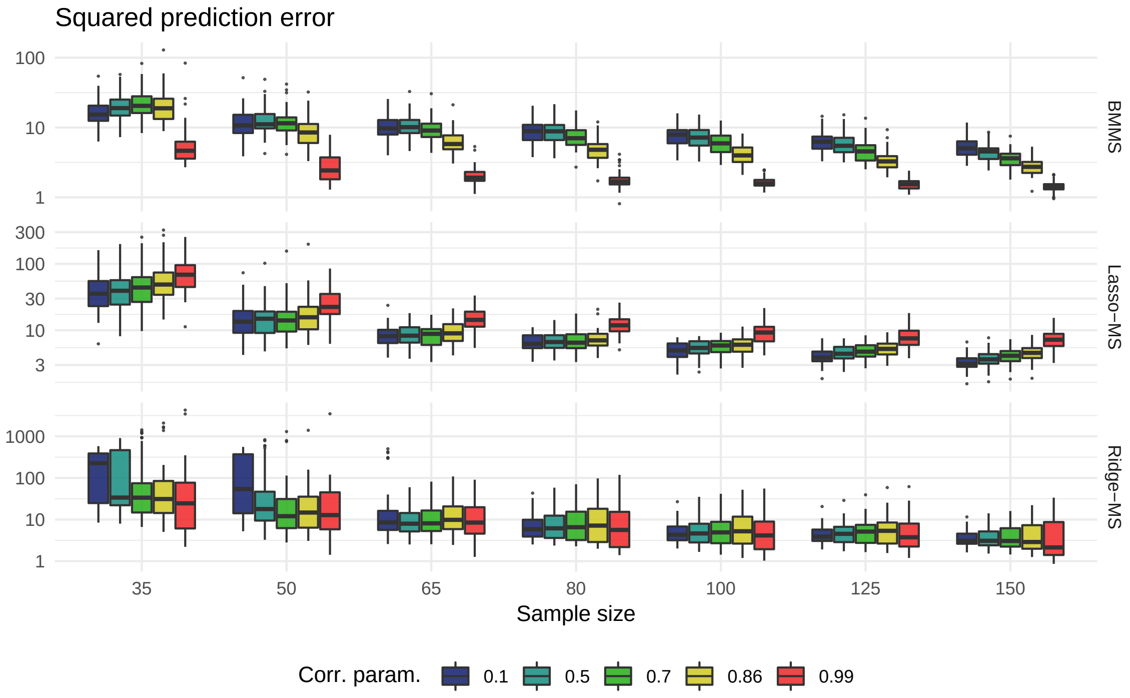


Figure 2.5: Squared error in the prediction of $n_{\text{out}} = 100$ new observations generated with the same model, on 50 simulated datasets.

$\theta_1, \theta_2, \theta_3$. For each configuration of sample size and level of correlation we generate 50 datasets and apply the three above-mentioned models. We report box-plots of the estimation results in Figure 2.4; Figure 2.5 is the analogous for the out-of-sample prediction task. One could alternatively apply ridge and lasso on single scale data to estimate β directly using X_j at one of the available resolutions; we are not concerned with these approaches as they are not multiresolution, but we report their performance in Appendix 2.9 for completeness.

2.6.2 Gender differences in multiresolution tfMRI data

Brain activity and connectivity data play a central role in neuroscience research today, but increasingly high-resolution medical imaging devices make management and analysis of such data challenging. We use data from the *Human Connectome Project* (HCP) (Essen et al., 2013; Glasser et al., 2013), considering a sample of $n = 100$ subjects, with the goal of classifying subjects' genders using brain activation data during task-based functional Magnetic Resonance Imaging (*tfMRI*). Gender differences have been observed in neuroscience and linked to brain morphology and connectivity (Tomasi and Volkow, 2012; Gong et al., 2011), or task-based activity patterns (Kaisera et al., 2009; Lee et al., 2014). We use *BM&Ms* on tfMRI data recorded during the *social* task in HCP, preprocessed according to the *Gordon333* (Gordon et al., 2016) parcellation. This hierarchical partitioning splits the brain into a multiscale structure of 333 regions, 26 lobes, and 2 hemispheres. We further parcellate lobes into 80 *sublobes*. The regions within a sublobe are contiguous. We use sublobes as an intermediate level in our multiscale regression. We expect each region to have low explanatory power on its own, but it remains unclear whether grouping them to form a coarser structure might improve predictive power. In particular, while the coarse-scale 26 lobes are easily interpretable given their connection to known brain functions, they might not be an efficient coarsening for our predictive task. These questions cannot be addressed by single-scale models. We thus consider the lobes-sublobes-regions multiscale structure as specified by Gordon et al. (2016), on a binary regression model with logit link, i.e. $Pr(y_i = 1) = (1 + \exp(-\mathbf{x}_{1,i}\theta_1 - \cdots - \mathbf{x}_{K,i}\theta_K))^{-1}$, where y_i is the subject's gender, and $\mathbf{x}_{j,i}$ are their data at scale j.

X_j is a $n \times d_j$ matrix collecting subjects' brain activation data at resolutions $j \in \{1, 2, 3\}$ corresponding to regions and lobes, so that $d_1 = 26, d_2 = 80, d_3 = 333$. We use Bayesian variable selection modules to estimate the contributions of each resolution to the regression function: $\theta_j = (\theta_{j,1}, \cdots, \theta_{j,d_j})$ for $j \in \{1, 2, 3\}$. The estimated posterior means of θ_j and the final estimate of the coefficients β are shown in Figure 2.6. In analyzing the resulting multiscale decomposition, *BM&Ms* models highlights coarse-scale differences: the parietal lobes in the right hemisphere and areas around the parieto-occipital sulcus in

Model	Accuracy	AUC
BM&Ms (V. S. modules)	0.804	0.896
Bayes V. S.	0.8	0.892
Ridge	0.791	0.870
Lasso	0.714	0.821

Table 2.1: Correct classification rate (*Accuracy*), and area under ROC curve (AUC), on random samples of size $n_{\text{out}} = 385$, averaged across 100 resamples of the data.

the left hemisphere have the largest positive effect in predicting gender. These coarse-scale findings align to the literature (see e.g. Koscik et al. 2009). At the finer scale, increased cingulo-opercular activity during the *social* task are most predictive of gender differences.

We compare *BM&Ms* to competing models on the same data, noting that unlike our approach, the other provide no multiscale interpretation. Table 2.1 reports the out-of-sample performance of all models, whereas Figures 2.1 and 2.7 show the estimation output of the competitors. Refer to Appendix 2.9.2 for additional details on the implemented models. Ridge regression achieves good out-of-sample performance but interpretations are unclear as many regions have large estimated effects. By contrast, Lasso regression has poor out-of-sample performance. Its output may mislead researchers into believing that only a few isolated regions account for gender differences. The Bayesian variable selection model results paint a similar picture to ridge regression, with additional information on posterior selection probabilities of each brain region, which helps in quantifying uncertainty when compared to a Lasso model.

2.7 Discussion

In this chapter, we introduced a Bayesian modular approach that builds an overall model by the sequential application of increasingly more complex component modules. Our approach can be applied to assess the contribution to the regression function of multiple resolutions of the data. Compared to established methods for multiscale regression such as wavelets, our method is more flexible and with the potential of easier interpretation. Both simulations and real data analysis show that this is not achieved at the expense of performance.

The modular posterior distribution of *BM&Ms* is the product of each module's posterior distribution, and hence it inherits their properties. Choosing component modules requires clarity on what the objective of analysis is. In this chapter, we used variable selection modules, as the resolution structure was pre-specified by the parcellation of the

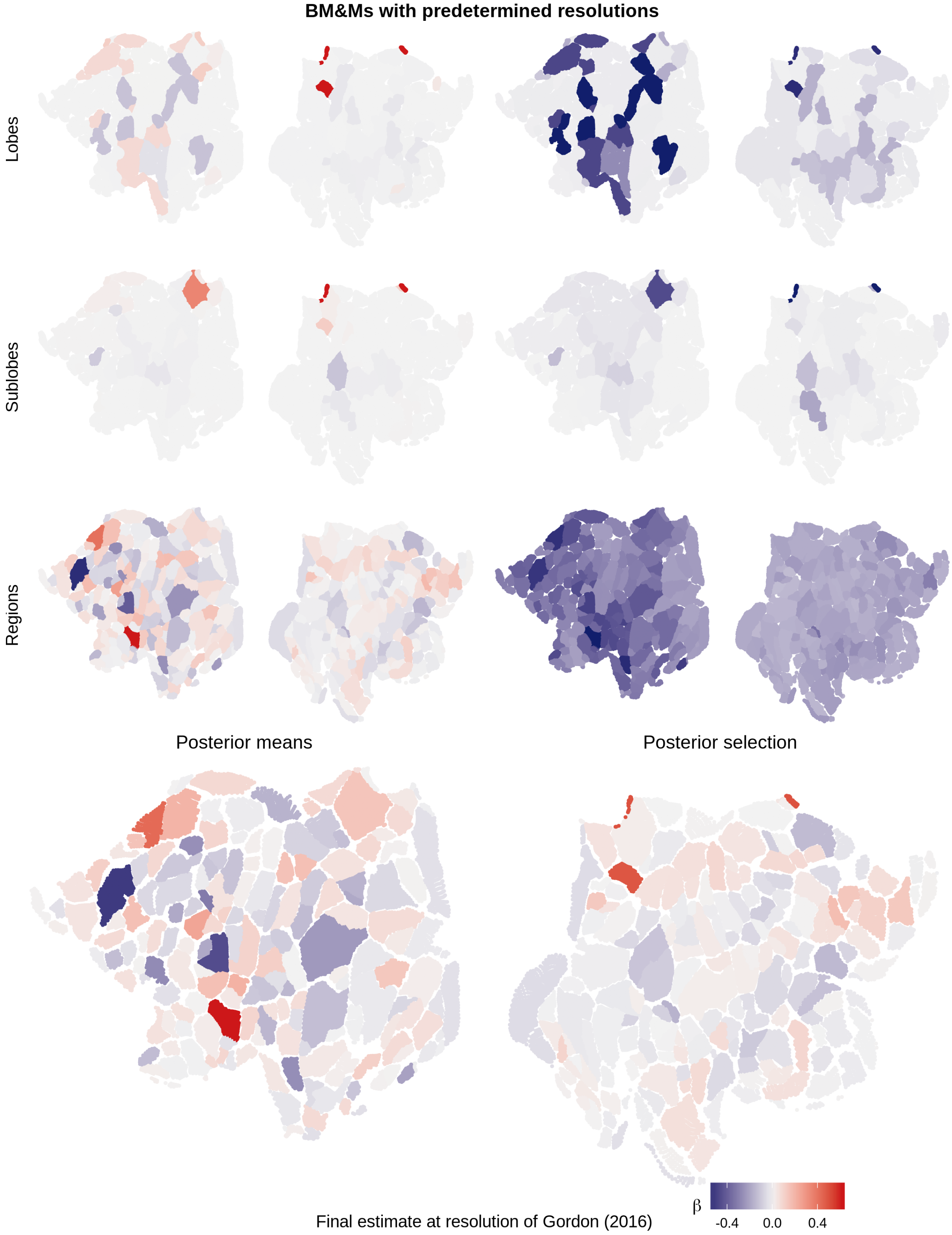

BM&Ms with predetermined resolutions
Lobes
Sublobes
Regions
Posterior means
Posterior selection
Final estimate at resolution of Gordon (2016)
β
-0.4
0.0
0.4

Figure 2.6: Upper left: posterior means of the contributions of lobes, sublobes, and regions, to the multiresolution regression function, estimated via *BM&Ms* using variable selection modules. Upper right: posterior selection probabilities. Bottom: estimated modular posterior mean after reconstruction of the multiscale structure.

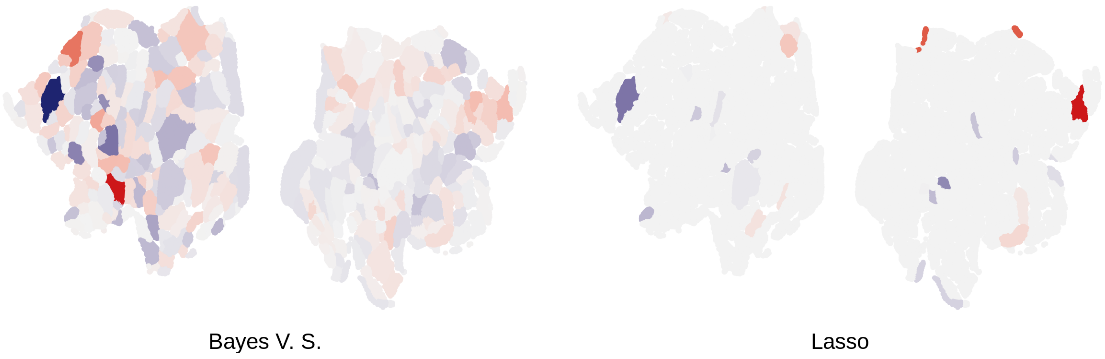


Figure 2.7: Coefficient on brain regions estimated via Bayesian variable selection and Lasso.

brain cortex of Gordon et al. (2016). The same modules could be used when predictors have a group-membership or treed structure. If the predictors are temporally- or spatially-indexed, however, we can take advantage of the structure and more efficiently decompose the regression function into resolution contributions. For example, we may consider the spatial location of each brain region to adaptively group neighboring regions into coarser resolutions that aid in interpreting the results. However, methods that do consider the regions' spatial locations, such as Functional Principal Components Regression (FPCR) were shown to produce starkly different output when compared to other models (see Figure 2.1). Our challenge will thus be to reconcile the FPCR results with the output from other models. To achieve this goal, we develop *BM&Ms* in Chapter 3 for scenarios in which the predictors are temporally- or spatially-indexed.

2.8 Further topics

2.8.1 Practical implementation of *BM&Ms*

Suppose only two measurements are taken from a sensor at times t_1 and t_2, to be used as inputs in regression. We thus call $S = \{t_1, t_2\}$, $|S| = d = 2$, $X_S = \begin{bmatrix} x_{t_1} & x_{t_2} \end{bmatrix}$. For simplicity, we denote $x_1 = x_{t_1}$, $x_2 = x_{t_2}$, $X_2 = X_S$, $X_1 = x_{t_1} + x_{t_2}$. We assume the covariance of the design matrix X_2 depends on parameter r as follows:

$$C(r) = \frac{1}{n} X_2' X_2 = \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}$$

which implies $\frac{1}{n} X_1' X_1 = 2 + 2r$. We fix $\beta_1, \beta_2, \varepsilon \sim N(0, \sigma^2 I_n)$ and set $\bar{\beta}_1 = \frac{1}{2}(\beta_1 + \beta_2)$. We then consider two models

$$\text{(i)} \quad y = x_1\beta_1 + x_2\beta_2 + \varepsilon \qquad \text{(ii)} \quad y = (x_1 + x_2)\bar{\beta}_1 + \varepsilon$$

Models (i) and (ii) consider the data at the high and low resolutions, respectively. Note that the KL divergence of (ii) from (i) is

$$KL(r) = \frac{n}{2\sigma^2}(\bar{\beta} - \beta)'C(r)(\bar{\beta} - \beta).$$

This is increasing in $|\beta_1 - \beta_2|$ and decreasing on r, as $\frac{\delta}{\delta r}KL(r) = \frac{n}{\sigma^2}(\bar{\beta}_1 - \beta_1)(\bar{\beta}_1 - \beta_2) \leq 0$. This implies that similar regression coefficients and high observed correlations (large positive r) between covariates make the lower resolution model a good approximation of the high resolution one, and thus we might prefer it, given its increased parsimony.

In these settings, we can implement *BM&Ms* as in Section 2.4, letting σ_j^2 for $j = 1, 2$ be known, by setting up two modules. For **Module 1** we have:

$$y = \theta_1(x_1 + x_2) + \varepsilon_1 \quad \text{where} \quad \varepsilon_1 \sim N(0, \sigma_1^2 I_n)$$

with prior parameters $m_1 = 0$ and $V_1 = n((x_1 + x_2)'(x_1 + x_2))^{-1} = \frac{1}{2}(r + 1)^{-1}$. We get

$$\begin{aligned}
\Sigma_1 &= (2(r + 1) + 2n(r + 1))^{-1} = (2(r + 1)(n + 1))^{-1} \\
\mu_1 &= \frac{(x_1 + x_2)'y}{2(r + 1)(n + 1)} = \frac{n\frac{x_1'y}{n} + n\frac{x_2'y}{n}}{2(r + 1)(n + 1)} \xrightarrow[n\to\infty]{} \frac{\beta_1 + \beta_2}{2},
\end{aligned}$$

whereas **Module 2** is implemented as:

$$e_2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \theta_{21} \\ \theta_{22} \end{bmatrix} + \varepsilon_2 \quad \text{where} \quad \varepsilon_2 \sim N(0, \sigma_2^2)$$

where $e_2 = y - (x_1 + x_2)\frac{(x_1+x_2)'y}{2(r+1)(n+1)}$. In this case we set the prior parameters as

$$m_2 = \begin{bmatrix} m_{21} & m_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} \qquad V_2 = n \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}^{-1},$$

hence the posterior distribution parameters are

$$\begin{aligned}
\Sigma_2 &= \frac{1}{n + 1} \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}^{-1} \\
\mu_2 &= \frac{1}{n + 1} \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1'e_2 \\ x_2'e_2 \end{bmatrix} \\
&= \frac{1}{n + 1} \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1'y - x_1'(x_1 + x_2)\frac{(x_1+x_2)'y}{2(r+1)(n+1)} \\ x_2'y - x_2'(x_1 + x_2)\frac{(x_1+x_2)'y}{2(r+1)(n+1)} \end{bmatrix} \\
&\xrightarrow[n\to\infty]{} \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}^{-1} \begin{bmatrix} \beta_1 + r\beta_2 - (r + 1)\frac{\beta_1+\beta_2}{2} \\ r\beta_1 + \beta_2 - (r + 1)\frac{\beta_1+\beta_2}{2} \end{bmatrix} \\
&= \frac{1}{r + 1} \begin{bmatrix} 1 & -r \\ -r & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\beta_1 - \frac{1}{2}\beta_2 \\ \frac{1}{2}\beta_2 - \frac{1}{2}\beta_1 \end{bmatrix} \frac{1}{r + 1} \begin{bmatrix} 1 & -r \\ -r & 1 \end{bmatrix} (\beta - L_1C_1\beta) = \begin{bmatrix} \frac{\beta_1-\beta_2}{2} \\ \frac{\beta_2-\beta_1}{2} \end{bmatrix},
\end{aligned}$$

where $C_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$. Σ_2 is the posterior covariance of $\theta_2 \mid \theta_1$, whereas the asymptotic variance of $\theta_2 = (\theta_{21}, \theta_{22})$ is

$$\begin{aligned} AVar(\theta_2) &= \Omega_2^{-1} + \sigma^2 L_1 \Omega_1^{-1} L_1' \\ &= \sigma^2 \left(\begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}^{-1} + \frac{1}{2(r+1)} L_1 L_1' \right) \\ &= \sigma^2 \left(\frac{1}{(1-r)(r+1)} \begin{bmatrix} 1 & -r \\ -r & 1 \end{bmatrix} + \frac{1}{2(r+1)} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \right) \\ &= \frac{\sigma^2}{r+1} \left(\frac{1}{1-r} \begin{bmatrix} 1 & -r \\ -r & 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \right) \\ &= \frac{\sigma^2}{r+1} \left(\begin{bmatrix} \frac{1}{1-r} & \frac{-r}{1-r} \\ \frac{-r}{1-r} & \frac{1}{1-r} \end{bmatrix} + \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \right) \\ &= \frac{\sigma^2}{r+1} \begin{bmatrix} \frac{2+1-r}{2(1-r)} & \frac{-2r+1-r}{2(1-r)} \\ \frac{-2r+1-r}{2(1-r)} & \frac{2+1-r}{2(1-r)} \end{bmatrix} = \frac{\sigma^2}{r+1} \begin{bmatrix} \frac{3-r}{2(1-r)} & \frac{1-3r}{2(1-r)} \\ \frac{1-3r}{2(1-r)} & \frac{3-r}{2(1-r)} \end{bmatrix} \\ &= \frac{\sigma^2}{2(1-r)(r+1)} \begin{bmatrix} 3-r & 3r-1 \\ 3r-1 & 3-r \end{bmatrix} \end{aligned}$$

In summary:

$$\begin{array}{ll} \Sigma_1 = (2(r+1)(n+1))^{-1} & \Sigma_2 = \frac{1}{n+1} \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}^{-1} \\ \mu_1 = \frac{x_1' y + x_2' y}{2(r+1)(n+1)} & \mu_2 = \frac{1}{n+1} \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1' e_2 \\ x_2' e_2 \end{bmatrix} \\ \mu_1 \xrightarrow[n\to\infty]{} \frac{\beta_1+\beta_2}{2} & \mu_2 \xrightarrow[n\to\infty]{} \begin{bmatrix} \frac{\beta_1-\beta_2}{2} \\ \frac{\beta_2-\beta_1}{2} \end{bmatrix} \end{array}$$

Note how μ_1 roughly corresponds to the average of the high-resolution coefficient vector, whereas μ_2 – which is interpreted as the added detail from the higher resolution – to half differences. Finally, the asymptotic variance of $\theta_2 = (\theta_{21}, \theta_{22})$ reported above has determinant $\frac{8}{(1-r)(1+r)}$, which diverges for $r \approx 1$ making the higher resolution worthless. For $r < 1$, the coefficient vector at the highest resolution can be estimated consistently with *BM&Ms* via $\beta_M = L_1\mu_1 + \mu_2$. In fact, notice that:

$$\beta_M = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} \mu_1 + \mu_{2,1} \\ \mu_1 + \mu_{2,2} \end{bmatrix} \xrightarrow[n\to\infty]{} \begin{bmatrix} \frac{\beta_1+\beta_2}{2} \\ \frac{\beta_1+\beta_2}{2} \end{bmatrix} + \begin{bmatrix} \frac{\beta_1-\beta_2}{2} \\ \frac{\beta_2-\beta_1}{2} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$$

We now consider a case in which $m_1 = m_2 = 0, V_1 = n(X_1'X_1)^{-1}, V_2 = n(X_2'X_2)^{-1}$. Define $\beta_M^c = L_1\mu_1 + c\mu_2$. Then

$$\beta_M^c = L_1\mu_1 + c\mu_2 = L_1\mu_1 + cl\hat{\beta} - c\Sigma_2 X_2' X_1 \mu_1 = (1-cl)L_1\mu_1 + cl\hat{\beta}$$

with $l = \frac{n}{n+1}$ and $c \in \{0, 1\}$. From a frequentist perspective, we can consider the two alternative choices for c. If we select $c = 0$, we are estimating β through $L_1\mu_1$, meaning that we completely discard the contribution of the high resolution. Instead, choosing $c = 1$ results in a slightly shrunk version of $\hat{\beta}$. The MSE of this estimator for $c = 0$ and $c = 1$ is, respectively:

$$\begin{aligned} \text{MSE}(\beta_M^{c=0}) &= ((l-2)^2 + l^2)\frac{\beta_1^2 + \beta_2^2}{4} + l(l-2)\beta_1\beta_2 + \frac{\sigma_2^2}{n(1+r)} \\ \text{MSE}(\beta_M^{c=1}) &= (1-l)^2((l-2)^2 + l^2)\frac{\beta_1^2 + \beta_2^2}{4} + \\ &\quad + (1-l)^2 l(l-2)\beta_1\beta_2 + \frac{2l^2\sigma_2^2}{n(1+r)(1-r)} \end{aligned}$$

First, for $n \to \infty$ the bias term approaches 0 only for $c = 1$, whereas the variance will decrease in both cases with n. A more relevant scenario is $r \approx 1$ and/or $\beta_1 \approx \beta_2$: if $c = 1$ the variance diverges when $r \to 1$. In other words, if r is large, considering the two measurements separately leads to a large expected error. Similarly, $\beta_1 \approx \beta_2$ results in

$$\begin{aligned} \text{MSE}(\beta_M^{c=0}) &= 2(1-l)^2\beta_1^2 + \frac{\sigma_2^2}{n(1+r)} \\ \text{MSE}(\beta_M^{c=1}) &= 2(1-l)^4\beta_1^2 + \frac{2l^2\sigma_2^2}{n(1+r)(1-r)} \end{aligned}$$

meaning that the bias term is almost equalized, but favoring $c = 0$, whereas the comparison on variance entirely depends on r: the closer the two coefficients β_1 and β_2 are to each other, the closer to zero r must be to make it worth it to consider the high resolution. Ultimately, this shows how considering the data at the highest resolution might be counterproductive. An alternative way to visualize why $c = 1$ may be suboptimal is to look at β_M^c (with $c \in [0, 1]$) as an estimator that shrinks towards the lower resolution coefficient function.

2.8.2 Derivation of the Mean Squared Error of 2.8.1

The expected value of the modular estimator for β is

$$\begin{aligned} E\left[\beta_M^c\right] &= E\left[(1-cl)L_1\mu_1 + cl\hat{\beta}\right] \\ &= E\left[l(1-cl)L_1(X_1'X_1)^{-1}X_1'y + cl\hat{\beta}\right] \\ &= E\left[l(1-cl)L_1(X_1'X_1)^{-1}X_1'(X_2\beta + \varepsilon_2)\right] + cl\beta \end{aligned}$$

$$\begin{aligned} &= l(1-cl)L_1(X_1'X_1)^{-1}X_1'X_2\beta + cl\beta \\ &= (l(1-cl)L_1(X_1'X_1)^{-1}X_1'X_2 + cl)\beta \end{aligned}$$

$$\begin{aligned}
&= (l(1-cl)\begin{bmatrix}\frac{1}{2n(r+1)}\\ \frac{1}{2n(r+1)}\end{bmatrix}\begin{bmatrix}n(1+r) & n(1+r)\end{bmatrix} + cl)\beta\\
&= l(1-cl)\begin{bmatrix}\frac{\beta_1+\beta_2}{2}\\ \frac{\beta_1+\beta_2}{2}\end{bmatrix} + cl\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}
\end{aligned}$$

so that the estimator has bias

$$E\left[\beta_M^c\right] - \beta = \frac{1-cl}{2}\begin{bmatrix}l-2 & l\\ l & l-2\end{bmatrix}\beta$$

which, as expected, is smaller for $l \approx 1$. We also obtain

$$\begin{aligned}
\text{Bias}^2 &= \frac{(1-cl)^2}{4}\beta'\begin{bmatrix}(l-2)^2+l^2 & 2(l-2)l\\ 2(l-2)l & (l-2)^2+l^2\end{bmatrix}\beta\\
&= (1-cl)^2((l-2)^2+l^2)\frac{\beta_1^2+\beta_2^2}{4} + (1-cl)^2 l(l-2)\beta_1\beta_2
\end{aligned}$$

We then move to calculating the variance of μ_β.

$$\begin{aligned}
Var(\beta_M^c) &= Var((1-cl)L_1(X_1'X_1)^{-1}X_1'y + cl(X_2'X_2)^{-1}X_2'y)\\
&= Var(((1-cl)L_1(X_1'X_1)^{-1}X_1' + cl(X_2'X_2)^{-1}X_2')\varepsilon)\\
&= \sigma_2^2((1-cl)L_1(X_1'X_1)^{-1}X_1'\\
&\quad + cl(X_2'X_2)^{-1}X_2')((1-cl)L_1(X_1'X_1)^{-1}X_1' + cl(X_2'X_2)^{-1}X_2')'\\
&= \sigma_2^2((1-cl)^2L_1(X_1'X_1)^{-1}L_1'\\
&\quad + cl(1-cl)L_1(X_1'X_1)^{-1}X_1'X_2(X_2'X_2)^{-1}\\
&\quad + cl(1-cl)(X_2'X_2)^{-1}X_2'X_1(X_1'X_1)^{-1}L_1' + c^2l^2(X_2'X_2)^{-1})
\end{aligned}$$

$$\begin{aligned}
&= \sigma_2^2\left(\frac{(1-cl)^2}{2n(r+1)}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} + \frac{cl(1-cl)}{2n(r+1)(1-r^2)}\begin{bmatrix}1\\ 1\end{bmatrix}\begin{bmatrix}1+r & 1+r\end{bmatrix}\begin{bmatrix}1 & -r\\ -r & 1\end{bmatrix} +\right.\\
&\quad\left. + \frac{cl(1-cl)}{n(1-r^2)(r+1)}\begin{bmatrix}1 & -r\\ -r & 1\end{bmatrix}\begin{bmatrix}1+r\\ 1+r\end{bmatrix}\begin{bmatrix}1 & 1\end{bmatrix} + \frac{c^2l^2}{n(1-r^2)}\begin{bmatrix}1 & -r\\ -r & 1\end{bmatrix}\right)\\
&= \sigma_2^2\left(\frac{(1-cl)^2}{2n(r+1)}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} + \frac{cl(1-cl)}{2n(1-r)(1+r)}\begin{bmatrix}1-r & 1-r\\ 1-r & 1-r\end{bmatrix} +\right.\\
&\quad\left. + \frac{cl(1-cl)}{2n(1+r)}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} + \frac{c^2l^2}{n(1-r^2)}\begin{bmatrix}1 & -r\\ -r & 1\end{bmatrix}\right)\\
&= \frac{\sigma_2^2}{n(1+r)}\left(\frac{(1-cl)^2}{2}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} + cl(1-cl)\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} + \frac{c^2l^2}{1-r}\begin{bmatrix}1 & -r\\ -r & 1\end{bmatrix}\right)
\end{aligned}$$

And then finally

$$\text{Tr}(\text{Var}(\beta_M^c)) = \frac{\sigma_2^2}{n(1+r)}\left((1-cl)(1+cl) + \frac{2c^2l^2}{1-r}\right)$$

So that the modular estimator has mean square error:

$$\begin{aligned}\mathrm{MSE}(\beta^c_M) = (1-cl)^2((l-2)^2+l^2)\frac{\beta_1^2+\beta_2^2}{4} + (1-cl)^2 l(l-2)\beta_1\beta_2 + \\ + \frac{\sigma_2^2}{n(1+r)}\left((1-cl)(1+cl) + \frac{2c^2l^2}{1-r}\right)\end{aligned}$$

2.9 Appendix

2.9.1 Additional simulated data analysis output

We report MSE and MSPE for the single-scale versions of the Lasso and Ridge models (alongside *BM&Ms* for reference), at different sample sizes and for different correlation parameter values. Therefore, *X1*, *X2*, *X3* refer to single resolution models using the low, medium, high resolution, respectively.

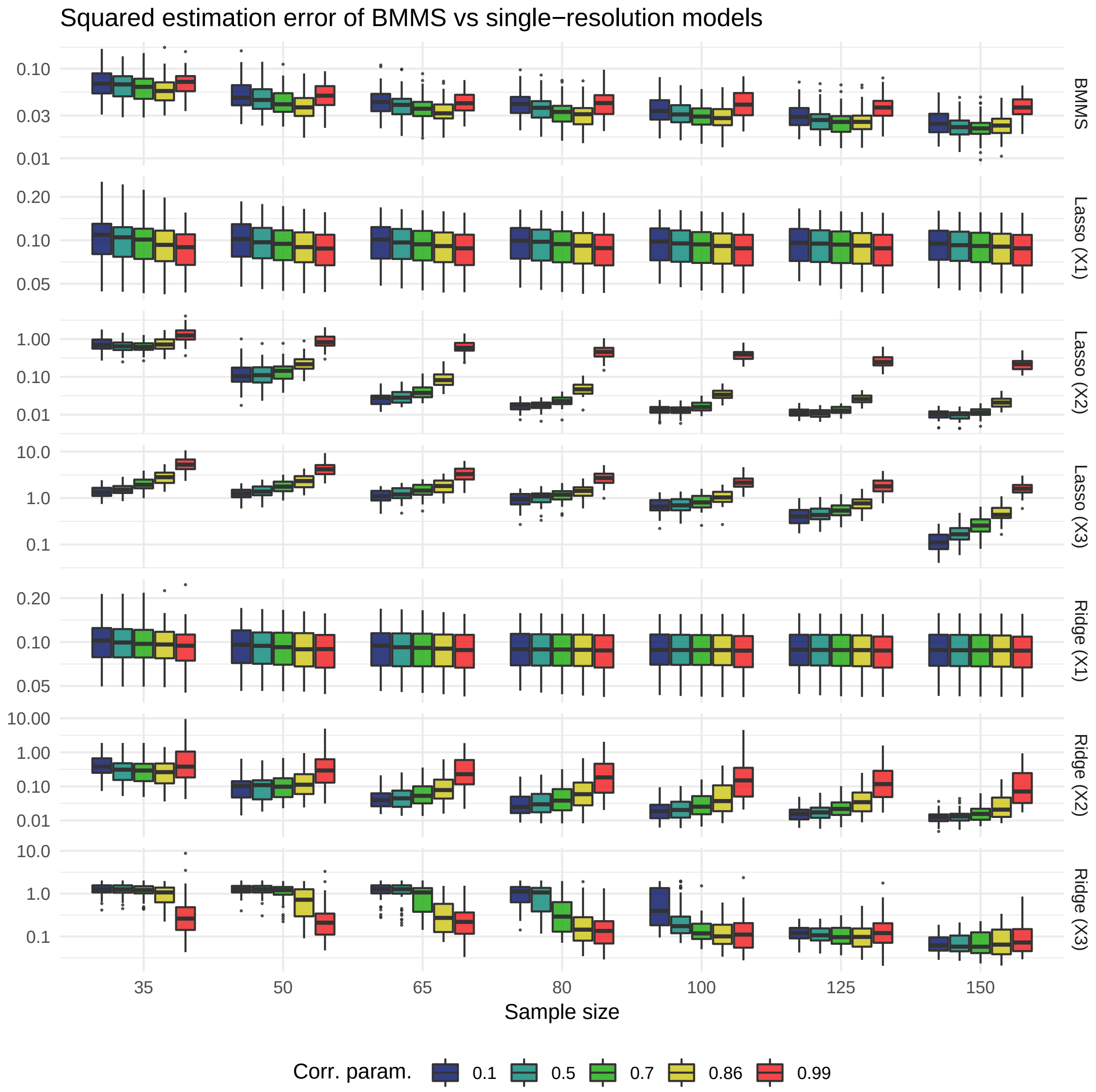


Figure 2.8: Mean Square Error for the tested models at different settings, estimated as average over 100 simulations.

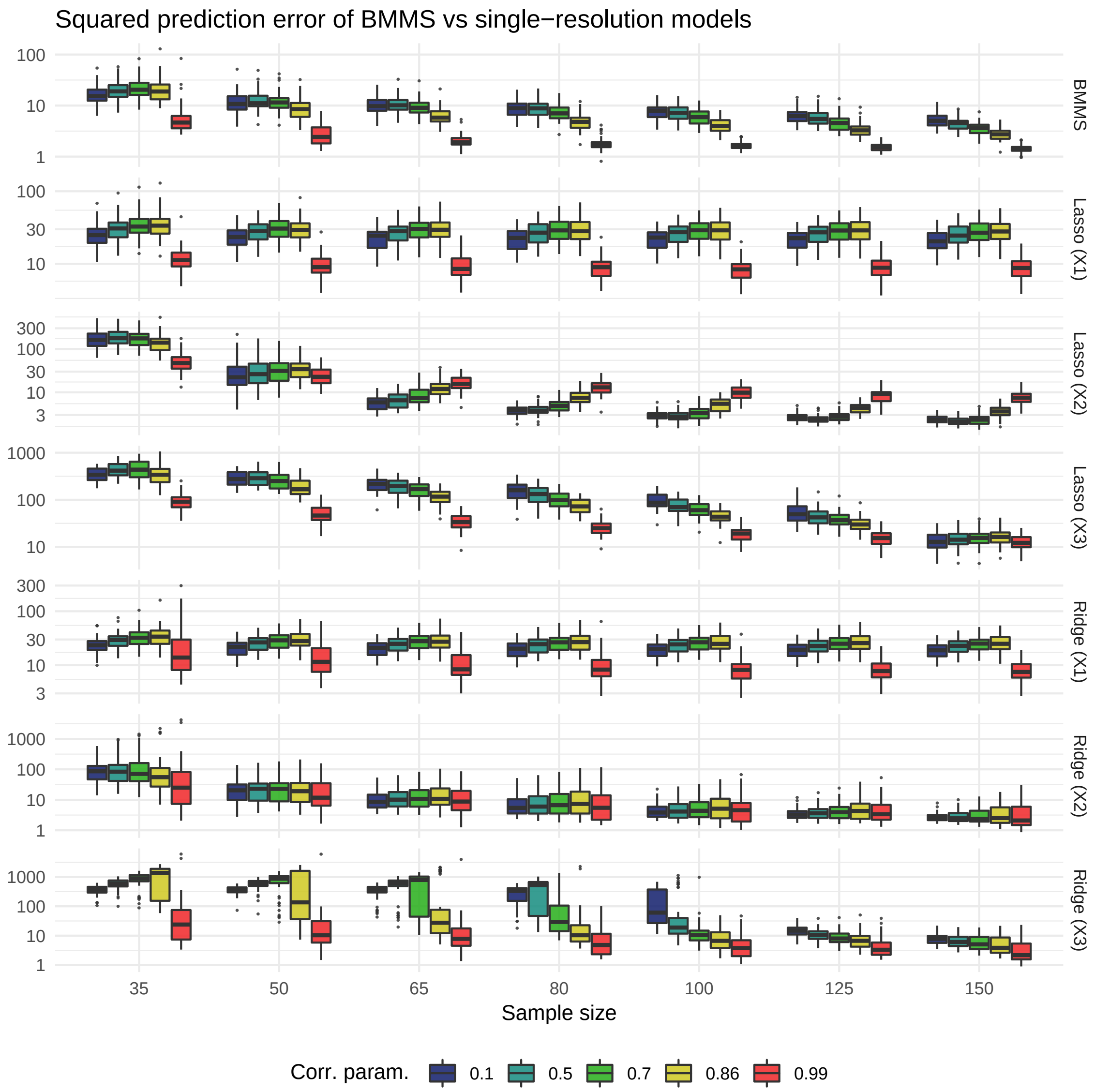


Figure 2.9: Mean Square Prediction Error for the tested models at different settings, estimated as average over 100 simulations.

2.9.2 Implemented models

- *BM&Ms*: using Algorithm 1 with Bayesian variable selection modules. In the brain imaging application, each of the $K = 3$ resolution contributions θ_s is linked to a selection parameter γ_s. The number of columns of X_s selected by γ_s is $q_{\gamma_s} = \sum_j \gamma_{s,j}$. We assign the prior $\gamma_s \propto \exp(-\kappa_s q_{\gamma_s})$ where $\kappa_s = 100, 100, 3$, $g = .3$ for resolutions $s = 1, 2, 3$, respectively, and corresponding to lobes, sublobes, brain regions. For each γ_s, $\pi(\beta_{\gamma_s}) = N(0, g(X'_{s,\gamma_s} X_{s,\gamma_s})^{-1})$.

- *Lasso*: lasso regression (Tibshirani, 1996) using cross-validation for λ (from R package `glmnet`)

- *Ridge*: ridge regression (Hoerl, 1962) using cross-validation for the ridge parameter (from R package `MXM`)

- *Ridge-* and *Lasso-MS* (Section 2.6.1): the models (Hoerl 1962; Tibshirani 1996) are fit using $X = [X_1 \cdots X_K]$ as design matrix.

- *Bayesian V. S.*: Bayesian variable selection using g-priors (Marin and Robert, 2007). This is equivalent to *BM&Ms* as described above, using the single-scale data corresponding to brain regions and setting $\kappa = 3$, $g = 1$.

Chapter 3

Multiscale regression on structured tensor predictors

3.1 Introduction

The increasing dimensionality and complexity of the data modern researchers collect typically require simplifying choices in either the preprocessing of data or in the modeling approach itself. In the former case, data are often down-sampled, flattened, and aligned so that a significant dimension reduction is achieved and complex models can be estimated using the simplified data. The latter scenario, instead, involves a much lower degree of data preprocessing, favoring the estimation of simplified models instead. For example, functional Magnetic Resonance Imaging (fMRI) data correspond to a high-resolution volumetric video (a four-way tensor) for each study participant.

Modeling brain imaging data at the voxel level requires a simplification of the modeling approach; cortical activation maps at voxel level can be estimated by considering each voxel separately, plus a subsequent correction for false discoveries (Genovese et al., 2002). Complex models of brain connectivity using these data are typically fit to down-sampled data obtained by low-resolution parcellations (Desikan et al., 2006) and their subsequent flattening using correlations. A line of research in this sense is devoted to the identification and optimization of cortical parcellations for network analysis (Dennis et al., 2011; Bassett et al., 2011). Using the data at multiple predetermined resolutions was the focus of Chapter 2, where we implemented *BM&Ms* in a case where a multiscale cortical parcellation (Gordon et al., 2016) was predetermined and contrasted the results to single-scale models. However, the resolution structure was taken as given, and there was no consideration on the spatial positioning of brain regions. If we take the spatial structure into account, we obtain a brain image predictor for each subject.

A general approach to this kind of problem is to consider the data to be used as input in regression as a tensor (or multidimensional array). We focus on the case of scalar-on-tensor regression, similarly to Guhaniyogi et al. (2017); Zhou et al. (2013); Li et al. (2018) (a more general overview of tensor regression and related algebraic operations is in Hoff 2015). In this context, each observation i corresponds to a scalar response y_i and a D-dimensional array-valued predictor $\mathbf{X}_i \in \mathbb{R}^{d_1 \times \cdots \times d_D}$. Raw fMRI data corresponds to $D = 4$. Each element $x_{i,(j_1,\dots,j_D)}$ of $\mathbf{X}_i$ is associated to y_i via the corresponding regression coefficient $\beta_{j_1,\dots,j_D}$. These coefficients can be collected in a tensor $\mathbf{B}$ having the same size as $\mathbf{X}_i$. The resulting regression model can be written, for $i = 1, \dots, n$, as:

$$y_i = \text{vec}(\mathbf{X_i})\text{vec}(\mathbf{B}) + \varepsilon_i$$

where $\varepsilon \overset{iid}{\sim} N(0, \sigma^2)$ and vec$(\mathbf{B})$ is the vectorization of the $\mathbf{B}$ tensor, of dimension $d = \prod_{j=1}^{D} d_j$ which is not infrequently multiple orders of magnitude larger than the sample size n. As we mentioned in previous chapters, these settings require some form of structure to be imposed on the regression coefficient and/or the predictor, in order to drastically reduce the problem dimensionality. For example, a greatly simplifying assumption is that

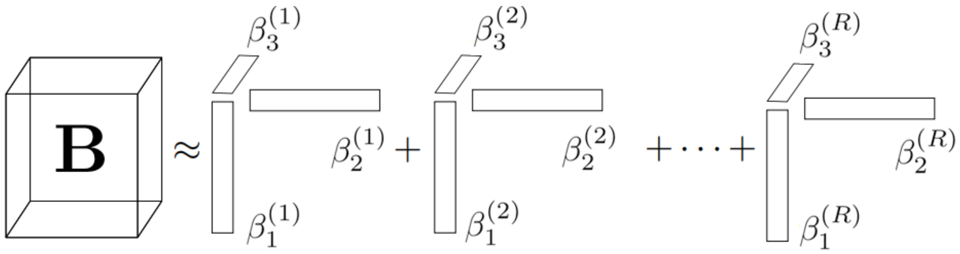


Figure 3.1: A rank-R decomposition of a 3-way tensor $\mathbf{B}$ assumes $\mathbf{B}_{i,j,k} = \beta_i^{(1)}\beta_j^{(1)}\beta_k^{(1)} + \cdots + \beta_i^{(R)}\beta_j^{(R)}\beta_k^{(R)}$

$\mathbf{B}$ admits a rank-R decomposition:

$$\mathbf{B} = \sum_{r=1}^{R} \beta_1^{(r)} \circ \cdots \circ \beta_D^{(r)}$$

where $\beta_r \in \mathbb{R}^{d_r}$, and "$\circ$" corresponds to the outer product between two vectors;[1] see Figure 3.1. Modeling the coefficient tensor $\mathbf{B}$ using a rank-R parafac decomposition reduces the number of parameters to be estimated from $\prod_{j=1}^{D} d_j$ to $R\sum_{j=1}^{D} d_j$. This approach was used by Zhou et al. (2013) on neuroimaging data with a scalar response, while Guhaniyogi et al. (2017) make similar assumptions but adopt a Bayesian perspective. Note that even with $D = 2$, the dimension of the unknown parameter may easily exceed the sample size n even after assuming a low-rank decomposition of $\mathbf{B}$. For example, the Human Connectome Project data on brain activation can be represented by an image (hence $D = 2$) of size $d_1 = 341 \times d_2 = 896$ for each subject (Essen et al., 2013), and therefore even a rank-1 decomposition will correspond to a high-dimensional setting. For this reason, it is customary to shrink the margins of a decomposition of $\mathbf{B}$ to obtain a sparse representation of the regression function.

The above low-rank representations might be unnecessarily restrictive when the predictors are spatiotemporally indexed, and ineffective at high resolutions; instead, one could approach the problem by assuming that adjacent measurement locations have similar effects on the output. This scenario is realistic when the predictors are time series, images, videos. In these settings, multiresolution decompositions of the spatiotemporally-indexed regression tensor $\mathbf{B}$ allow to identify coexisting low- and high-resolution patterns in the regression function.

We develop a Bayesian modular and multiscale methodology to decompose the regression function into low-to-high resolution components using spatiotemporally-indexed tensor predictors and a scalar response, under the hypothesis that each margin of $\mathbf{B}$ is ordered on a one-dimensional discrete grid. Thus if $D = 1$ we may interpret $\mathbf{X}_i$ as a time-series predictor, if $D = 2$ as an image, and so on. We do not make the assumption

[1] If $\mathbf{a} \in \mathbb{R}_a^d$ and $\mathbf{b} \in \mathbb{R}_b^d$, then $\mathbf{a} \circ \mathbf{b} = (a_i b_j)_{i=1,\ldots,d_a; j=1,\ldots,d_b}$.

that $\mathbf{B}$ admits a low-rank decomposition as outlined above. The overall regression model extends the one presented in Chapter 2, eq. (2.5):

$$\begin{aligned} y_i &= \mathrm{vec}(\mathbf{X}_i)\mathrm{vec}(\mathbf{B}) + \varepsilon_i \\ &= \mathrm{vec}(\mathbf{X}_i)\beta_K + \varepsilon_i = \mathrm{vec}(\mathbf{X}_i)(\mathcal{L}_1\theta_1 + \cdots + \mathcal{L}_K\theta_K) + \varepsilon_i = \\ &= \mathrm{vec}(\mathbf{X}_i)\mathrm{vec}(\Theta_1) + \cdots + \mathrm{vec}(\mathbf{X}_i)\mathrm{vec}(\Theta_K) + \varepsilon_i \end{aligned} \tag{3.1}$$

where $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, for all $i = 1, \dots, n$. In words, the vectorized regression tensor β_K is decomposed into resolution contributions. Each of these can be interpreted as being the vectorization of a resolution component. Having K pre-specified resolutions is equivalent to assuming that $\mathcal{L}_j$ are known operators for $j = 1, \dots, K$, each corresponding to data resolution $S_1, \dots, S_K$. If the available data resolutions are uninteresting, or in cases in which the goal is to *learn* a multiscale structure, we can consider S_j, and the corresponding $\mathcal{L}_j$, as unknown parameters, and construct a hierarchical model by assigning a prior to the multiresolution structure itself $\mathcal{S} = \{S_1, \dots, S_K\}$. In this latter case, $\mathcal{L}_j$ for $j = 1, \dots, K$ need to be defined in such a way as to represent a resolution of the data. We analyze possible alternatives with $D = 1$ and $D \geq 2$ later in this chapter.

Our modeling approach results in multiscale decompositions of the regression coefficient tensor $\mathbf{B}$, and can be generalized for tensors of size $D > 2$. After analyzing modeling alternatives in Section 3.1.1, we outline an efficient algorithm for linear multiscale tensor regression. We then apply our method on simulated and real data, first by detailing our method in the simple case of $D = 1$ in Section 3.2; later, we extend it to $D \geq 2$, with applications on scalar-on-image regression.

3.1.1 Unknown multiscale structures

Our previous treatment of multiscale regression assumed that $\mathcal{S}$ was known. With temporally- and/or spatially-indexed predictors, one can easily devise multiple ways to down-sample the data; for example, high-frequency time series are straightforwardly down-sampled by binning adjacent measurement locations. Even without mentioning the possible advantages of a mixed resolution – i.e. one that changes along the covariate space – one is faced with uncertainty on the data resolution. Regression estimates should thus take into account this underlying uncertainty. For this reason, we now consider $\mathcal{S}$ as unknown. Our general strategy to implement *BM&Ms* in this case is straightforward and boils down to embedding the modular posterior distribution of Chapter 3.2 into a larger hierarchical model.

$$
\begin{aligned}
\mathcal{S}|K &\sim \pi(\mathcal{S}, K) \\
\sigma_j^2 &\sim \pi(\sigma_j^2) \\
\Theta|\mathcal{S}, K &\sim \mathit{BM\&Ms}(\Theta, \mathcal{S}) \\
\beta &= \mathcal{L}_1\theta_1 + \cdots + \mathcal{L}_K\theta_K \\
y|\beta, \sigma_K^2 &\sim N(X\beta, \sigma_K^2 I_n))
\end{aligned} \tag{3.2}
$$

where X is the data matrix,[2] $\Theta = \{(\theta_1, \sigma_1^2), \ldots, (\theta_K, \sigma_K^2)\}$, and $\mathit{BM\&Ms}(\Theta, \mathcal{S})$ is shorthand for the use of our modular approach for inference on the multiscale decomposition, given $\mathcal{S}$ as in Chapter 2. In typical cases, sampling from the posterior distribution $p(\mathcal{S}|y)$ requires Markov-chain Monte Carlo approximations; this suggests the usage of simple modules, as it will alleviate the overall computational burden. We thus assume $\sigma_j^2 \sim \text{N.InvG}(a_j, b_j), \theta_j|\sigma_j^2 \sim N(m_j, \sigma_j^2 V_j)$, and $p_j(y|\theta_{1:j-1}, \mathcal{L}_{1:j-1}, X) = N(X(\mathcal{L}_1\theta_1 + \cdots + \mathcal{L}_{j-1}\theta_{j-1} + \mathcal{L}_j\theta_j), \sigma_j^2 I_n)$. Given $\mathcal{S}$, we can use Algorithm 1 to sample from the modular posterior distribution $\pi_M(\theta_{1:K}, \sigma_{1:K}^2|y, \mathcal{S}, X)$.

Algorithm 2 uses the modular posterior distribution in (3.2) inside a reversible-jumps MCMC scheme (Green, 1995). This kind of algorithm is made necessary by the fact that the proposal $\mathcal{S}^*$ is associated with θ_j^* which can be of different dimension from θ_j of $\mathcal{S}$. In fact $\mathcal{S}^*$ can be interpreted as being either a refinement or coarsening over a previous $\mathcal{S}$. In these scenarios, dimension-matching mechanisms and adjustments to the Metropolis-Hastings acceptance ratio need to be made to maintain detailed balance and ensure convergence to the stationary distribution. However, an advantage of using conjugate modules is that we can marginalize over θ_j and σ_j^2 at each module, obtaining $p_M(y|\mathcal{S})$ and $p_M(y|\mathcal{S}^*)$; this implies that we do not need to propose a new θ_j^* and perform dimension-matching (Dellaportas et al., 2002; Denison et al., 1998). The acceptance ratio α for $\mathcal{S}^*$ against $\mathcal{S}$ is thus:

$$
\alpha = \min\left(1, \frac{p_M(y|\mathcal{S}^*)\pi(\mathcal{S}^*)q(\mathcal{S}|\mathcal{S}^*)}{p_M(y|\mathcal{S})\pi(\mathcal{S})q(\mathcal{S}^*|\mathcal{S})}\right),
$$

where $q(i|j)$ is the probability of moving to state i from state j, and $\pi(\cdot)$ is the prior imposed on the multiscale structure parameter $\mathcal{S}$. Reversible-jumps MCMC typically suffers from mixing issues and it is often challenging to devise and implement efficient proposals. Models such as the Bayesian CART (Chipman et al., 1998) often lead to poorly-mixing MCMC and the prescription is to run multiple, relatively short chains. A similar outcome is typical with proposals for treed unknown parameters. One such example is $\mathcal{S}$ itself, which can be described as a sequence of increasingly finer scales, each refining on the previous ones; however, cutting the scale dependence via modularity allows one to greatly simplify the MCMC proposals as we outline below.

[2] In the tensor regression case, each row of X corresponds to $\text{vec}(\mathbf{X})_i$.

```
Initialize: Start with $\mathcal{S}^{(0)} = \{S_1, \dots, S_K\}$, and sample $\left\{(\theta_j, \sigma_j^2)^{(0)}\right\}_{j=1}^K$ using
            Algorithm 1.
for $t \in \{1, \dots, T\}$ do
    for $j \in \{1, \dots, K\}$ do
        • Randomly select a transition type for grid $S_j$ among
        change, increase, decrease;
        • With the selected transition, obtain proposal
         $\mathcal{S}^* = \{\dots, S_{j-1}, S_j^*, S_{j+1}, \dots\}$;
        • Evaluate the acceptance ratio $\alpha$;
        • If accepted, set $\mathcal{S}^{(t)} = \mathcal{S}^*$, else $\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)}$;
        • Sample $\left\{(\theta_j, \sigma_j^2)^{(t)}\right\}_{j=1}^K$ using Algorithm 1;
    end
end
```

Algorithm 2: Sampling $\left\{\mathcal{S}^{(t)}\right\}_{t=1}^T$ from $\pi(\mathcal{S}|y, X)$.

We specify the three proposals for $\mathcal{S}$ by noting that each grid $S_j \in \mathcal{S}$ can be described as a partitioning of the index space into cells and their respective centers $s_h \in S_j$ for $s = 1, \dots, h_j$, regardless of the dimension of the tensor. This means that these proposals are general and work for all $D \geq 1$. First, define A as the *inactive set*: $A = \{1, \dots, d_1 - 1\} \times \cdots \times \{1, \dots, d_K - 1\} \setminus S_K$ and includes all centers that do not appear at any scale. Then, we use the proposal functions:

- *Change*: from $S_j = \{s_1, \dots, s_m, \dots, s_{h_j}\}$ propose S_j^* such that $S_j^* = \{s_1, \dots, s_m^*, \dots, s_{h_j}\}$ where $s_m^* = s_m + d$, $s_m^* \in A$, and $d \in \mathbb{Z}^{d_1 \cdots d_K}$;

- *Increase*: from $S_j = \{s_1, \dots, s_{h_j}\}$ propose S_j^* such that $S_j^* = \{s_1, \dots, s_{h_j}, s_{h_j+1}\}$ where s_{h_j+1}, $s_{h_j+1} \in A$ generated at random from a probability mass function with support in A;

- *Decrease* from $S_j = \{s_1, \dots, s_{h_j}\}$ propose $S_j^* = S_j \setminus \{s_m\}$ where $m \in \{1, \dots, h_j\}$ is sampled uniformly.

Before outlining a general practical implementation for tensors of any dimension, we present the special case of $D = 1$. We consider it separately because in the simple setting of a single tensor margin (e.g. time-series predictors), we can further enrich our representation.

3.2 Tensor predictors with $D = 1$

The simplest case of scalar-on-tensor regression arises when $D = 1$. In this case, $\mathbf{B} = \beta \in \mathbb{R}^d$ is the usual regression coefficient vector, and its regularization may proceed with routine methods. However, we are interested here in a case in which regressors – and thus β – can be interpreted as having a temporal structure and possibly high correlation between adjacent measurement locations. Therefore, while one can use shrinkage methods such as ridge (Hoerl, 1962), Lasso (Tibshirani, 1996), or others, methods that do impose some structure are more appropriate. For example, the fused Lasso (Tibshirani et al., 2005) is a single-scale approach that selects and shrinks differences in coefficients at adjacent measurement locations. Alternatively, one can use functional data analysis methods (Ramsay and Silverman, 1997), which provide a useful view of this problem. The typical prescription from these models is to pursue sparsity by truncating or penalizing an orthogonal basis function representation of the data via shrinkage methods. The researcher must choose the basis and the shrinkage method. As we have seen in Chapter 1, one may use Haar wavelets coupled with a L1 penalty (see e.g. Zhao et al. 2012) as a multiresolution FDA method that achieves a sparse representation of β. All the approaches mentioned above usually calibrate parameters and quantify uncertainty via cross-validation and bootstrapping methods, but these may prove unreliable in small sample sizes or not trivial to implement (Chatterjee and Lahiri, 2011). For in-depth treatments of FDA, refer to Ramsay and Silverman (1997); in regression settings, Morris (2015) and Reiss et al. (2017) provide an overview. More recently Grollemund et al. (2018) modeled the coefficient vector as a step function with sparse support in a single resolution setting. Bayesian methods typically rely on sparsity-inducing priors on the transformed data (e.g. Brown et al. 2001; Goldsmith et al. 2014). However, orthogonal transformations of the data reduce flexibility and are usually obtained via predetermined schemes; relaxing these assumptions may be computationally challenging.

3.2.1 Temporally-indexed predictors

With temporally-indexed predictors, one may view the regression model on observed data as the discretization of a functional linear regression with scalar output,

$$y = \int X(t)\beta(t)dt + \varepsilon.$$

Our construction of *BM&Ms* for functional linear regression rests on decomposing $\beta(t)$. We model the coefficient function as

$$\beta(t) = \tilde{\theta}_1(t) + \cdots + \tilde{\theta}_K(t),$$

where the collection of $\tilde{\theta}_j(\cdot)$ for $j = 1, \dots, K$ builds an overcomplete basis. In the simplest setting, we may assume that they are a set of coarse-to-fine step functions with jumps at locations $S_j = \{t_1^{(j)}, \dots, t_{d_j-1}^{(j)}\}$ such that $S_j \subset S_{j+1}$, and with values $\theta_j = (\theta_r^{(j)})'$, $r = 1, \dots, d_j$. This assumption corresponds to the idea that the regression coefficient function can be decomposed into resolution contributions. Each contributing function can be represented as

$$\tilde{\theta}_j = \sum_{r=1}^{d_j} A^0_{jr}(t)\theta_r^{(j)} \quad \text{where} \quad A^0_{jr}(t) = \mathbf{1}(t_{r-1}^{(j)} < t < t_r^{(j)}),$$

with the indicator function $A^0_{jr}(t)$ describing the support of $\tilde{\theta}_j$. Since we assume that the predictors x_{it} are temporally-indexed on a finely-spaced grid $t = \frac{h}{p}$, $h = 1, \dots, d$ for all observations $i = 1, \dots, n$, we can represent each $\tilde{\theta}_j(t)$ as $\tilde{\theta}_j(t) = \mathcal{L}_j\theta_j$ for $t = \frac{h}{p}$. The jth column of $\mathcal{L}_j$ is thus the discretization of $A^0_{jr}(t)$ and corresponds to a binary vector taking value 1 at locations in the grid falling between $t_{r-1}^{(j)}$ and $t_r^{(j)}$, 0 otherwise. For example, $d = 5$, $K = 2$, $S_1 = \{1, 4\}$ and $S_2 = \{1, 3, 4\}$, corresponding to $d_1 = |S_1| = 3$, and $d_2 = |S_2| = 4$, respectively, generate

$$\mathcal{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathcal{L}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The *BM&Ms* basis $\boldsymbol{\mathcal{L}}$ thus encodes the discretizations of the A^0_{jr} functions and is completely determined by the jump locations in $S_1, \dots, S_K$. Each S_j is free from orthogonality requirements; the jump locations can thus be considered unknown and their posterior distribution approximated via MCMC. Contrast this to wavelet methods, which are equivalent to fixing S_j *a priori* at $t = \frac{2r-1}{2^j}$ for $j = 1, \dots, \log_2(d)$, $r = 1, \dots, 2^{j-1}$ to create the orthogonal multiscale basis.

Another advantage of this representation is that it can be easily generalized. In fact, we can replace A^0_{jr} with sigmoid functions which control jump smoothness via a parameter δ_j:

$$A^{\delta_j}_{jr}(t) = \left(e^{\frac{1}{\delta_j}(t_{r-1}^{(r)} - t)} + 1\right)^{-1} - \left(e^{\frac{1}{\delta_j}(t_r^{(r)} - t)} + 1\right)^{-1}. \tag{3.3}$$

An overcomplete basis $\boldsymbol{\mathcal{L}}^\delta$ with scale-specific smoothness levels may thus be constructed via discretization of $A^{\delta_j}_{jr}(t)$ functions. Figure 3.2 shows a visual representation of $A^{\delta_j}_{jr}(t)$ and compares *BM&Ms* bases with orthogonal wavelet bases.

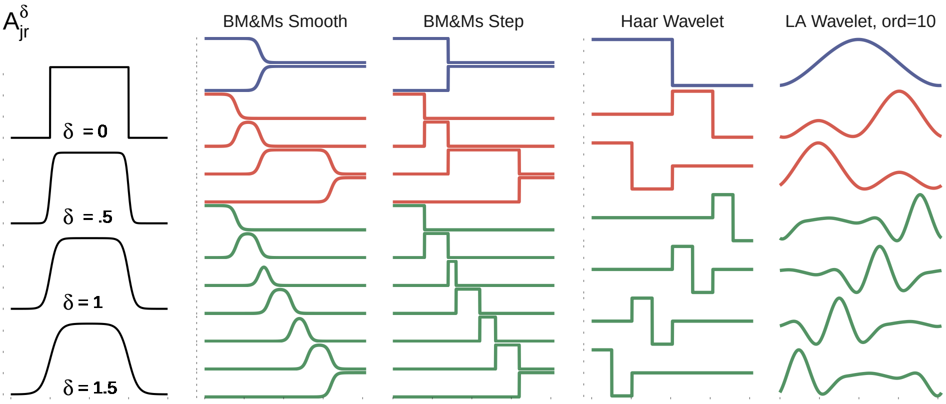


Figure 3.2: From the left: $A_{jr}^{\delta_j}(t)$; example of arbitrary $\boldsymbol{\mathcal{L}}$ with $K = 3$ and $\delta = .7$ or $\delta = 0$; orthogonal wavelets. Each color corresponds to a resolution component; each line to an element of the basis.

We finalize *BM&Ms* for functional linear regression by introducing a prior on the grid S_j and the scale-dependent smoothness δ_j in the larger hierarchical model outlined in (3.4).

Assume that $S_j = \{t_1^{(j)}, \ldots, t_{d_j}^{(j)}\}$ are such that $t_s^{(j)} < t_{s+1}^{(j)}$ for all s. Define $W_j = (w_s^{(j)})_{s=1,\ldots,d_j-1}$ such that $w_s^{(j)} = t_{s+1}^{(j)} - t_s^{(j)}$, $t_{d_j} = d$.

$$\begin{aligned}
\pi(S_j) &\propto \left(W_j' W_j\right)^{-\alpha} \exp\{\lambda 2^{K-j} |S_j| \log(|S_j|)\} \\
\sigma_j^2 \mid S_j &\sim N.InvG(a_j, b_j) \\
\pi(\delta) &\propto 1/\delta \\
X_j &= X \boldsymbol{\mathcal{L}}_j^{\delta} \\
\theta_j \mid \theta_{1:j-1}, \sigma_j^2, S_j &\sim N(0, n(X_j' X_j)^{-1}) \\
p_j(y_j \mid \theta_{1:j-1}, \sigma_j^2, S_j) &= N(X_{1:j}\theta_{1:j}, \sigma_j^2 I_n) \text{ (Module } j)
\end{aligned} \tag{3.4}$$

In other words, given S_j we assume a linear *BM&Ms* with module-dependent error variances, as in Section 2.4 and 3.2.1. We fix $a_j = a$, $b_j = b$ for all $j = 1, \ldots, K$. α penalizes grids S_j that place jumps closer to each other, λ favors more parsimonious grids and 2^{K-j} favors placing more jumps on later grids. We sample from the posterior distribution of δ via Metropolis-Hastings steps with log-normal proposals. Finally, the absence of orthogonality requirements facilitates posterior sampling of $\mathcal{S}$ and δ_j via MCMC, making it possible to estimate $\beta(t)$ of varying degrees of smoothness with little to no tuning of the prior hyperparameters. Figure 3.3 depicts the output of *BM&Ms* in two opposing cases.

3.2.2 Simulation study

We generate $n \in \{60, 200\}$ observations from a linear regression model $y = X\beta + \varepsilon$, with

$$\begin{array}{lll} \varepsilon \sim N(0, \sigma^2 I_n) & \Omega = (\omega_{h,j}) \quad h, j = 1, \dots, d & d \in \{128, 1024\} \\ \mathbf{x}_i \sim N(0, \Omega) & \omega_{h,j} = \exp\{-(1-\rho)|h-j|\} & \rho \in \{.7, .99\}. \end{array}$$

When $d = 128$ we fix $\sigma^2 = 1$, whereas for $d = 1024$ we fix $\sigma^2 = 100$. The coefficient vector β has dimension $d \times 1$ and is obtained by discretizing the *Doppler* and *Blocks* of Donoho and Johnstone (1994, 1995) on a regular grid. We control the correlation among the random covariates via ρ. This setup will thus include a "simple" scenario with many observations, low predictor dimension, low predictor correlation, low error variance ($n = 200, d = 128, \rho = .7, \sigma^2 = 1$), and a much more challenging scenario in which the signal is confounded by large predictor correlation, small sample size, high dimensionality and high error variance ($n = 60, d = 1024, \rho = .99, \sigma^2 = 100$).

We implement *BM&Ms* as in Section 3.2.1 with the goal of estimating the coefficient function $\beta(t)$, its multiscale decomposition $\beta(t) = \tilde{\theta}_1(t) + \cdots + \tilde{\theta}_K(t)$ for $K = 5$, and predicting the output on a test set of size 100. Posterior information on the scale-dependent smoothness of $\beta(t)$ is obtained by assigning a prior on δ_j. We fix $\delta_j = (K-j)\delta$ and assign a prior $\pi(\delta) \propto 1/\delta$. We fix the prior parameters detailed in 3.4 to $\lambda = .1$ and $\alpha = 1$ in all cases, and run the MCMC chain for 4000 iterations. With $n = 200, d = 1024, K = 5, \rho = .7$, the `R` package available at `github.com/mkln/bmms` takes on average about 90 seconds on a 4-core Linux workstation with Intel Core i7-3700 CPU to run 1000 MCMC iterations once compiled to take advantage of the Intel MKL libraries. The typical output for *BM&Ms* in this kind of problem is shown in Figure 3.3. We underline that the reduced coverage of the estimates in the *Doppler* case on the lower end of the index range is due to the high correlation among the predictors. In fact, opposing β effects of adjacent highly-correlated predictors on the response cancel each other out, and the resulting model is equivalent to one in which all the effects are set at zero.

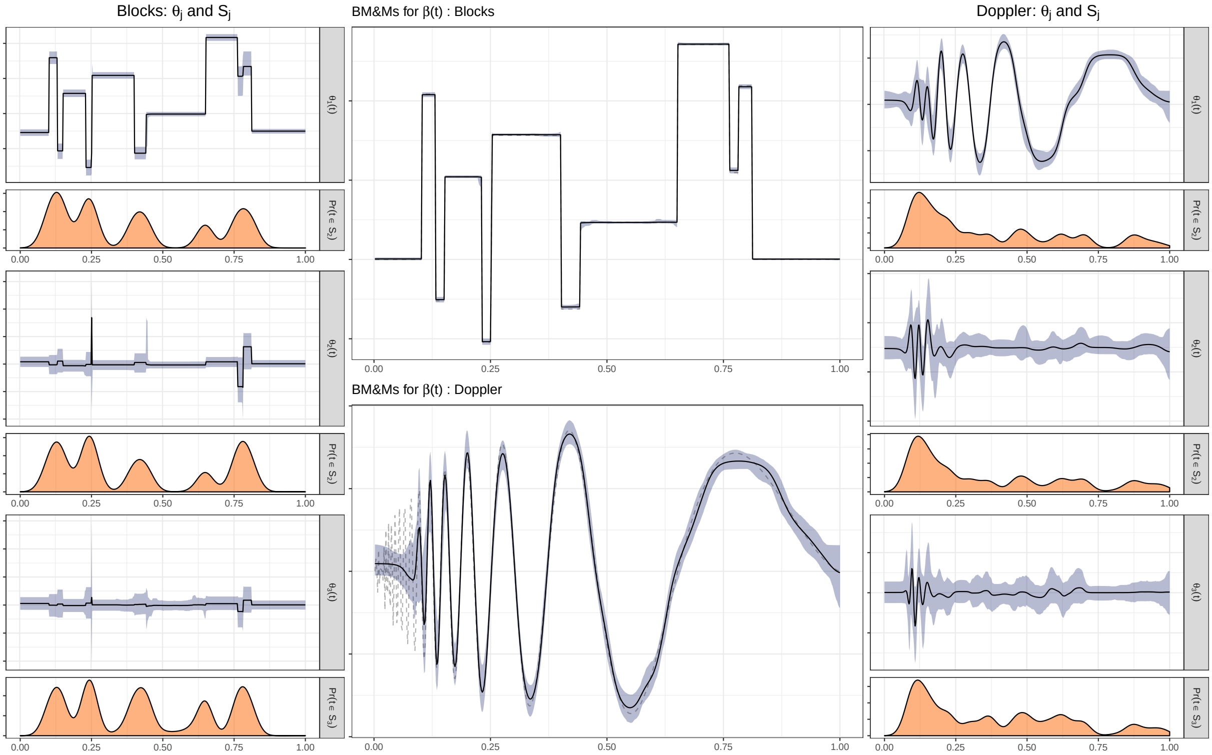


Figure 3.3: Center: *BM&Ms* estimate of β after 3000 iterations of MCMC for Blocks (top), and Doppler (bottom). On the sides, each of the 6 subplots depicts the estimated $\mathcal{L}_j\theta_j$ (top), along with a kernel-density estimate of the jump locations S_j (bottom). A jump location t_s identifies regions in the predictors' index space that are most related to a change in the effect on the outcome. The dashed line corresponds to the true regression coefficients. The smoothness parameter δ was estimated at $7.2 \cdot 10^{-4}$ for *Blocks*, .213 for *Doppler*.

We test *BM&Ms* against state-of-the-art models for functional linear regression on 1600 simulated datasets; for each combination of (d, ρ, n, β) we simulate 100 datasets and estimate errors in estimating β, and in the prediction of 100 out-of-sample observations generated by the same model. Table 3.1 reports the average performance of *BM&Ms* and competing models. Box-plots of the results for all datasets are in Figure 3.4. Details on each implemented model are in Figure 3.4. *BM&Ms* outperform all other tested approaches in all tested configurations of *Blocks*. Performance in the smooth *Doppler* case is generally equivalent to the best alternative method, if not better.

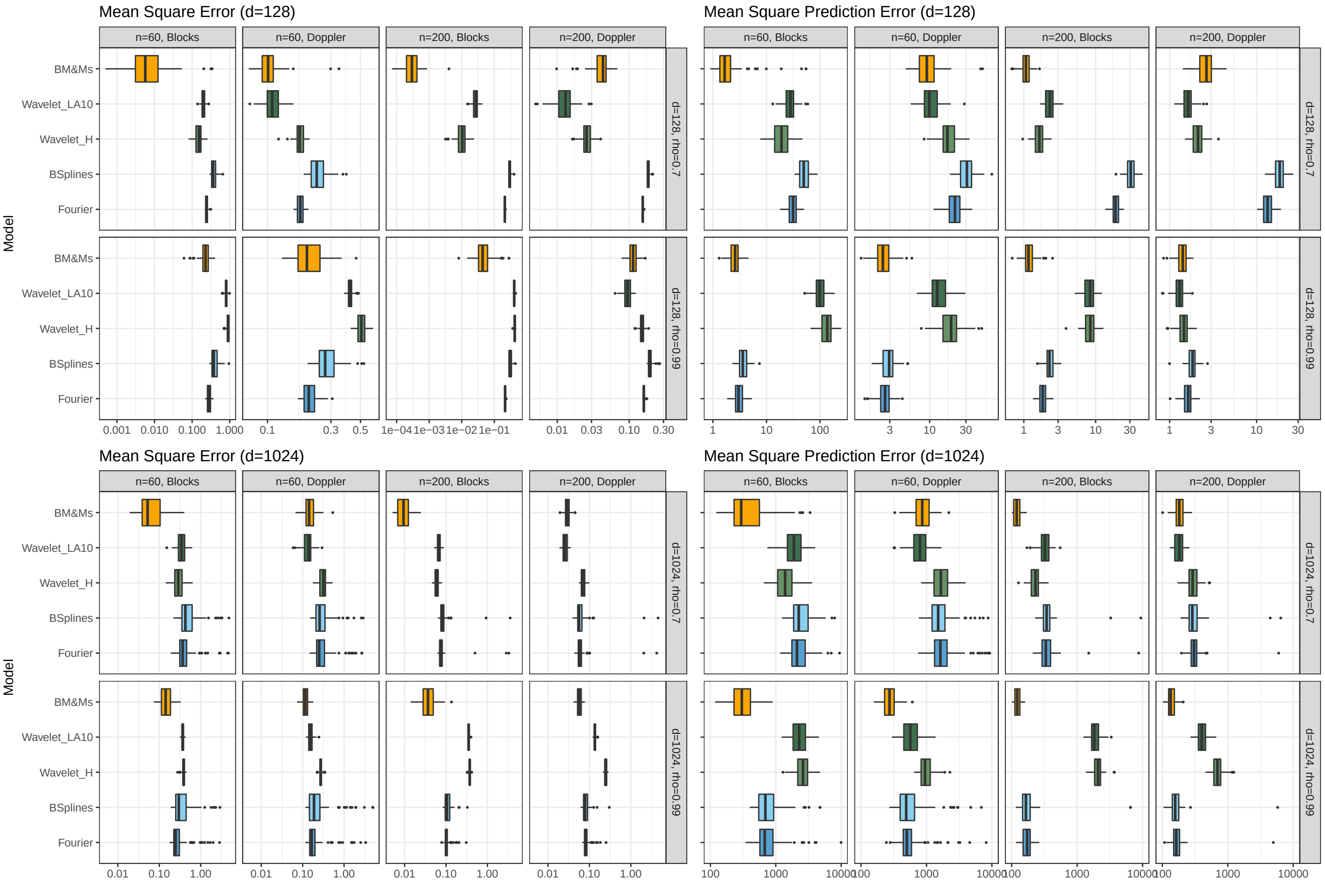


Figure 3.4: Box-plots for MSE and MSPE over 100 datasets for each configuration. Averages from these boxplots are reported in Table 3.1. 17 datasets resulted in errors for the FDA routines. For illustrative purposes, we also omit the results for 98 datasets with $d = 1024$, which resulted in very large outliers from the FDA packages.

		Estimation of $\beta(t)$ – MSE					Prediction of y_{out} – MSPE				
		FDA Fourier	FDA B-Splines	Haar Wavelets	LA10 Wavelets	***BM&Ms***	FDA Fourier	FDA B-Splines	Haar Wavelets	LA10 Wavelets	***BM&Ms***
d=128, *Blocks*											
n=60	ρ =0.7	0.240	0.361	0.152	0.196	**0.006**	31.4	49.7	19.1	28.0	**1.66**
	ρ =0.99	0.279	0.378	0.920	0.807	**0.227**	3.01	3.61	137.0	98.5	**2.6**
n=200	ρ =0.7	0.21	0.294	0.01	0.027	**0.0003**	18.9	30.8	1.65	2.3	**1.07**
	ρ =0.99	0.215	0.298	0.424	0.409	**0.044**	1.83	2.28	8.43	8.36	**1.16**
d=128, *Doppler*											
n=60	ρ =0.7	0.176	0.234	0.174	0.108	**0.101**	21.5	30.08	17.1	9.9	**9.18**
	ρ =0.99	0.204	0.271	0.507	0.415	**0.198**	2.6	2.94	19.1	12.6	**2.43**
n=200	ρ =0.7	0.154	0.182	0.026	**0.013**	0.043	13.4	18.5	2.12	**1.63**	2.65
	ρ =0.99	0.159	0.192	0.150	**0.095**	0.113	1.64	1.83	1.46	**1.30**	1.42
d=1024, *Blocks*											
n=60	ρ =0.7	0.368	0.441	0.287	0.344	**0.052**	2180.8	2459.8	1388.2	1893.2	**296.3**
	ρ =0.99	0.249	0.297	0.387	0.368	**0.143**	677.5	710.2	2609.1	2281.1	**302.5**
n=200	ρ =0.7	0.075	[1]0.081	0.06	0.068	**0.009**	333.3	[1]341.3	231.1	321.8	**114.3**
	ρ =0.99	[3]0.101	[5]0.107	0.374	0.353	**0.037**	[3]172.2	[5]166.4	2085.6	1830.1	**118.4**
d=1024, *Doppler*											
n=60	ρ =0.7	0.254	0.257	0.316	**0.140**	0.143	1670.9	1524.9	1658.9	**795.4**	862.2
	ρ =0.99	0.165	0.191	0.274	0.151	**0.117**	501.5	496.8	951.5	566.7	**272.3**
n=200	ρ =0.7	0.058	0.056	0.07	**0.026**	0.030	305.1	286.9	289.5	181.6	**181.1**
	ρ =0.99	[1]0.081	[7]0.079	0.249	0.135	**0.057**	[1]163.0	[7]157.0	693.9	398.7	**132.8**

Table 3.1: For each simulation setup, median of MSE and MSPE over 100 randomly generated datasets. We consider the median performance considering the occasionally unreliable implementation of FDA methods, which resulted in 17 failures (marked by square brakets in the table) or extreme low performance skewing the averages upwardly. A more complete picture is available in Section 3.5, where we show MSPE and MSPE box-plots and a summary table with mean MSE and MSPE.

3.3 Tensor predictors with $D \geq 2$

3.3.1 Overview

We now construct *BM&Ms* for multiscale scalar-on-image or scalar-on-tensor regression. Our goal is thus to obtain a multiscale decomposition of $\mathbf{B}$, following model (3.1); this requires us to define $\mathcal{L}_j$ appropriately for tensors of dimensions larger than 1. By doing so, we will be able to apply Algorithm 2 as we did for the $D = 1$ case, to compute the hierarchical model (3.2). Compared to the previous section, we now directly detail the construction of the overcomplete basis on the discretized grid, noting that smooth extensions can be analogously defined.

3.3.2 Construction of $\mathcal{S}$ for $D \geq 2$

Suppose for each subject we observe a tensor $\mathbf{X} = (b_{i_1,\dots,i_D})_{i_1,\dots,i_D=1}^{p_1,\dots,p_D}$. Our objective is to flatten this tensor into a vector $\mathbf{x}$ to be used in standard linear regression and apply Algorithm 2 to sample from the posterior distribution of $\mathcal{S}$. Therefore we call:

- $\mathbf{J}$ the *tensor of indices* of $\mathbf{X}$

$$\mathbf{J} = \{\mathbf{s} = (s_1, \dots, s_D) \text{ s.t. } s_1 \in \{1, \dots, p_1\}, \dots, s_D \in \{1, \dots, p_D\}\}$$

- a *center* of $\mathbf{X}$ is an element of $\mathbf{J}$;
- a *set of centers* is $S = \{\mathbf{s}_1, \dots, \mathbf{s}_h\}$, where $h < \prod_{j=1}^{D} p_j$;
- a *multiresolution structure* is $\mathcal{S} = \{S_1, \dots, S_K\}$ where K is the number of levels and $S_{j-1} \subset S_j$;
- the resolution increments are defined as $\mathcal{G} = \{G_1, \dots, G_K\}$ where $G_1 = S_1$ and $G_h = S_h \setminus S_{h-1}$;
- the inactive set is $A = \mathbf{J} \setminus S_K$;
- a *cell* $C_{\mathbf{s}^*,V}$ corresponding to a center $\mathbf{s}^* \in S$ and indices V is defined as $C_{\mathbf{s}^*,V} = \{\mathbf{j} \in V : d(\mathbf{j}, \mathbf{s}^*) < d(\mathbf{j}, \mathbf{s}) \text{ for all } \mathbf{s} \neq \mathbf{s}^*)$. In other words, $C_{\mathbf{s}^*,V}$ is a discrete Voronoi cell around $\mathbf{s}^*$ within indices tensor V.
- a *hierarchical Voronoi tessellation* of tensor $\mathbf{X}$ with indices $\mathbf{J}$ is thus the sequential construction of cells using S_1 within $\mathbf{J}$ at first, and subsequent subcells using G_j, $j > 1$ for finer partitioning.

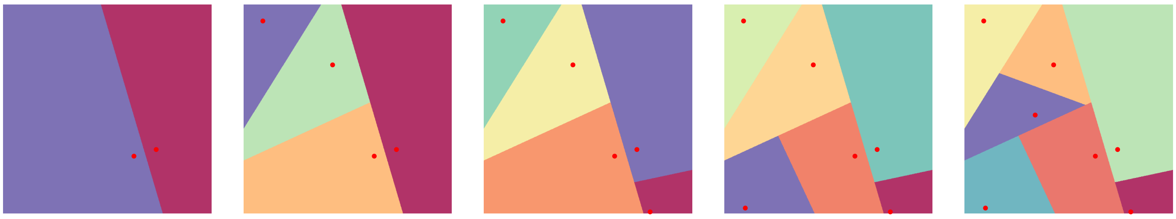


Figure 3.5: Hierarchical Voronoi tessellation of a tensor of size $D = 2$, corresponding to $\mathcal{L}_j$ for $j = 1, \dots, 5$. Centers are sequentially added and previously-identified areas are further divided into smaller piecer. The step-wise construction implies that only the first plot on the left is a Voronoi tessellation of the 2D surface.

For a given set of centers S_j, we obtain vector $\mathbf{x}_{S_j} = (x_1, \dots, x_h)$ from tensor $\mathbf{X}$ as:

$$\mathbf{x}_{S_j} = \mathbf{X}\mathcal{L}_j = (x_j)_{j=1}^h \quad \text{where} \quad x_j = \sum_{r \in C} \mathbf{X}_r.$$

where $C = C_{\mathbf{s_j},\mathbf{J}} \cap \cdots \cap C_{\mathbf{s_j},V_{j-1}} \cap C_{\mathbf{s_j},V_j}$. In other words, each element of $\mathbf{x}_{S_j}$ is the sum of elements of $\mathbf{X}$ with indices in the cell $C_{\mathbf{s_j},V_h}$ for all $h \in \{1, \dots, j\}$. A nested multiresolution decomposition can be obtained by considering sequences of subcells. From the application of $\mathcal{L}_1, \dots, \mathcal{L}_K$ corresponding to $\mathcal{S} = \{S_1, \dots, S_K\}$ we thus obtain $\mathbf{x}_1, \dots, \mathbf{x}_K$. This construction is equivalent to assuming that the regression coefficient tensor $\mathbf{B}$ is a piecewise constant "step-surface," defined by the locations of the centers. Holmes et al. (2005) use a standard Voronoi construction in a non-parametric regression setting, and note that the sharp edges produced by the Voronoi tessellation are undesirable as likely to produce implausible single model estimates. The same caveat applies to our layered construction. However, the individual model estimates will be smoothed by Bayesian model averaging.

Finally, we remark that the above construction can also be used with $D = 1$, but it is less parsimonious and slightly harder to interpret compared to a construction via *splits* (or jumps) as in Section 3.2. On the other hand, considering centers for $D \geq 2$ results in stable partitioning when running MCMC. Nevertheless, alternative constructions can be defined and be potentially better at down-sampling tensors to lower resolutions, especially if large portions of the tensor predictor have no effect on the output. This is a possible direction for future research.

3.3.3 Simulation study

We target the estimation and multiscale decomposition of $\mathbf{B}$ in the tensor regression model (3.1). We set $D = 2$ and generate the true $\mathbf{B}$ as one of the square images in Figure 3.6; they have dimensions $d_j = 40, 50, 64$ for $j = 1, 2$. The resulting dimension of $\text{vec}(\mathbf{B})$ is $d = 1600, 2500, 4096$ respectively.

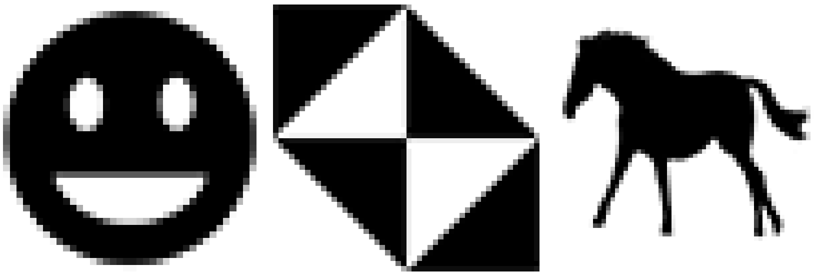


Figure 3.6: True regression tensor $\mathbf{B}_j$, where j is one of *Smile40*, *Geometric50*, *Horse64*.

For each of the 50 simulated datasets we generate $\mathbf{X}_i$ from a matrix-variate normal distribution with zero mean and exponential covariances as in Section 3.2.2, with $\rho_1 = \rho_2 = 0.9$. We fix a sample size of $n = 200$ for the training set, $n_{\text{out}} = 100$ for the test set. Finally, we set $\sigma^2 = 1$.

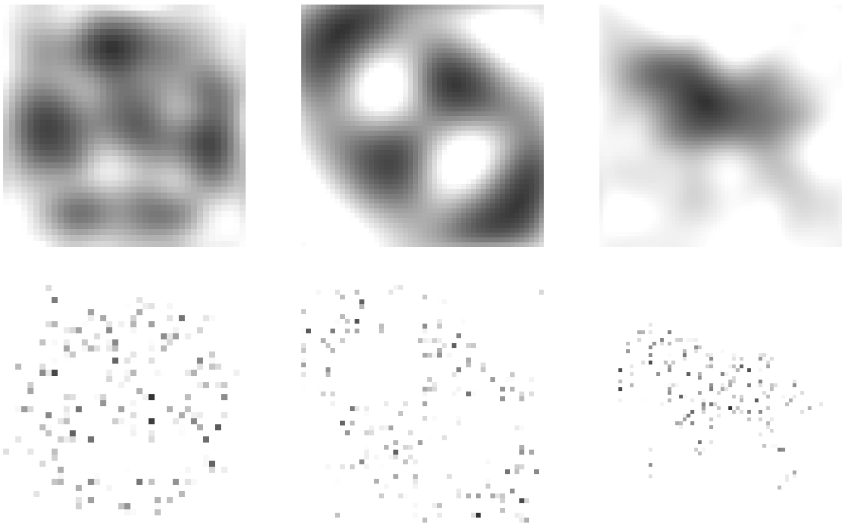
Estimation and prediction results of *BM&Ms* compared to Lasso regression and Functional Principal Components Regression (FPCR) are in Figure 3.2. A Lasso model on unstructured data struggles to estimate $\mathbf{B}$; FPCR and *BM&Ms* produce similar estimation output, validating our methodology. However, *BM&Ms* also decompose the regression tensor into resolution components which allow to sequentially reconstruct the estimate of $\mathbf{B}$ in four resolution steps, as in Figure 3.7b.

	Smile40		Geometric50		Horse64	
Model	MSE	MAPE	MSE	MAPE	MSE	MAPE
BM&Ms	**0.748**	**0.929**	**0.387**	**0.403**	**0.671**	**0.533**
Funct. PCR	1.606	14.972	1.289	0.853	1.047	1.145
Lasso	47.183	13.999	48.382	1.679	21.714	1.66

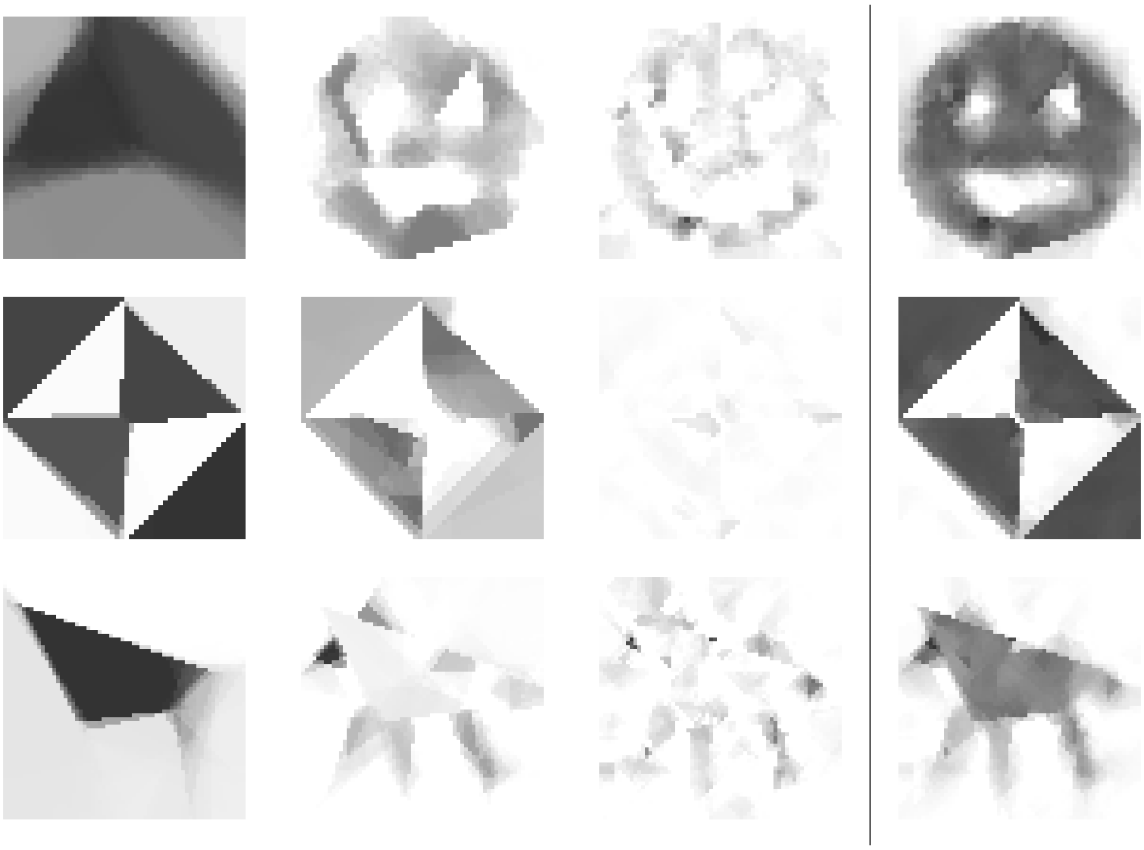
Table 3.2: Mean squared error in the estimation of $\mathbf{B}$, and mean average prediction error on a sample of size $n_{\text{out}} = 100$, averaged over 50 datasets.

3.3.4 Multiscale tensor classification using tfMRI data

Our analysis of brain imaging for classification in Chapter 2 used information on brain parcellations to develop a multiscale model without considering the actual spatial location of each region on the brain cortex. We can take advantage of this information to adaptively merge regions into bigger groups at multiple levels, thereby creating a multiscale structure. By this logic, instead of an unstructured list of brain regions, we now have a two-dimensional tensor or image, for each subject. In order to compare the new specification with other models, we use the same *Gordon333* parcellation of the cortex. This amounts to placing the activation value of each region on a matrix of size $p_1 = 23 \times 57 = p_2$,



(a) Estimation of the **B** regression tensor in each simulation type via FPCR and Lasso. From the left: *Smile40*, *Geometric50*, and *Horse64*. Top row: FPCR; bottom: Lasso.



(b) Posterior means of the resolution contributions for the *Smile40*, *Geometric50*, and *Horse64* regression tensors using *BM&Ms* for $j = 1, \dots, K = 3$ scale estimates and obtained via MCMC. On the right, the resulting posterior mean for the **B** tensor is the sum of the $K = 3$ components on the left.

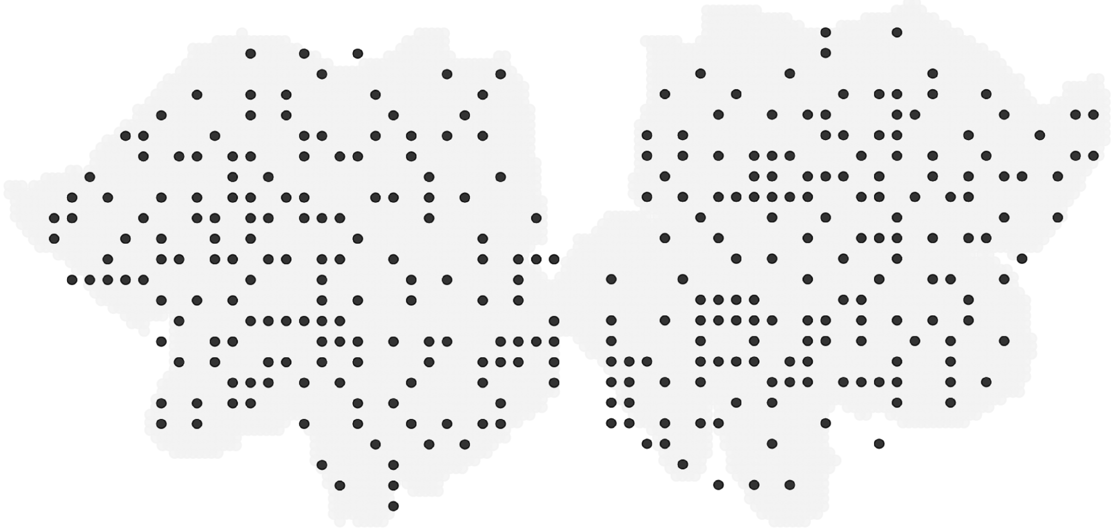


Figure 3.8: Activation values of each of region of *Gordon333* are placed in a 2D matrix (locations in black), so as to approximate their relative position on the brain cortex, shown in grey.

Model	Accuracy	AUC
BM&Ms	0.791	0.883
Functional PCR	0.778	0.853

Table 3.3: Correct classification rate (*Accuracy*), and area under ROC curve (AUC), on random samples of size $n_{\text{out}} = 385$, averaged across 100 resamples of the data.

with the relative location of each region approximated by the matrix indices; see Figure 3.8. The effective dimension of the problem is maintained at $p = 333$, as many locations inside the matrix do not correspond to the centroid of any region and their coefficients are thus fixed at zero.

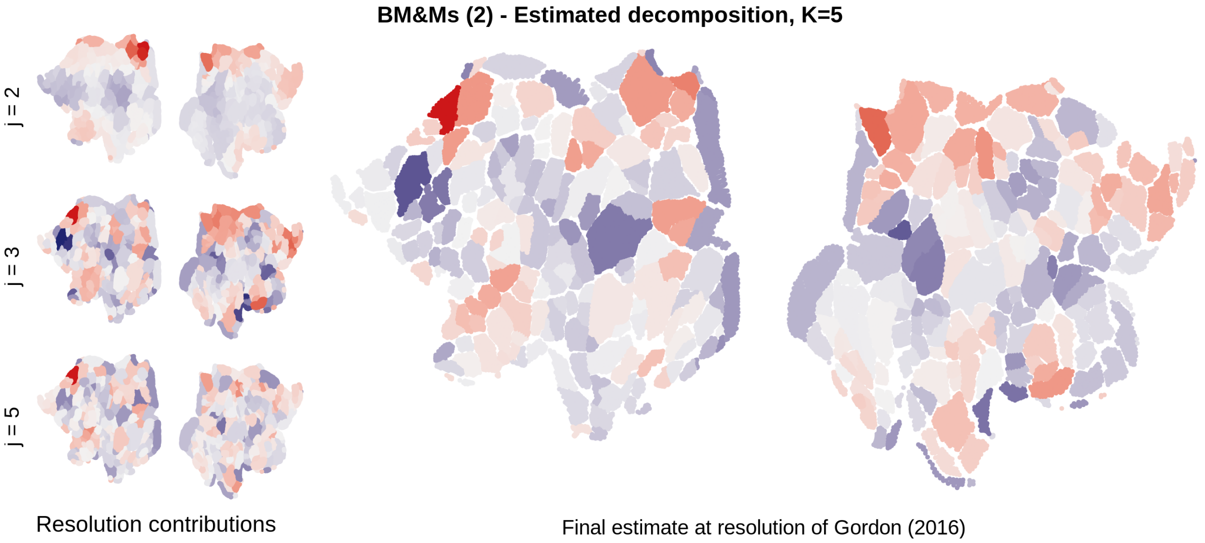

BM&Ms (2) - Estimated decomposition, K=5
j = 2
j = 3
j = 5
Resolution contributions
Final estimate at resolution of Gordon (2016)

Figure 3.9: Posterior means and multiresolution structure of the coefficient matrix, estimated via *BM&Ms*. The image on the right is decomposed as an additive expansion of $K = 5$ components, of which 3 are shown on the left.

We implement *BM&Ms* fixing $K = 5$ to decompose the original higher-resolution parcellation following the construction in Section 3.3.2. We introduce latent variables to appropriately account for the binary nature of the response, as in Chapter 2. The estimated posterior means of three of the five estimated resolution components are shown at the left of Figure 3.9; their sum corresponds to the final estimate of $\mathbf{B}$, shown on the right. *BM&Ms* compare favorably to FPCR, obtaining out-of-sample predictive performance in line with the models we had seen in Section 3.3.3. Overall, models using variable selection priors had the best accuracy; however, standard models for Bayesian variable selection scale poorly. Instead, our scalar-on-image specification of *BM&Ms* easily scales to much larger data.

In fact, we have so far restricted our analysis to the *Gordon333* brain parcellation of Gordon et al. (2016) in order to make comparisons across models. In Figure 3.10 we show the decomposition of the $\mathbf{B}$ tensor obtained using minimal preprocessing of the voxel-level activation data, corresponding to 69×180 image-predictors for each subject and an effective dimensionality of $p = 7716$. *BM&Ms* achieves averaged out-of-sample accuracy of approximately 77% on 50 resamples of these data. Lasso models on the same data resulted on less than 65% accuracy.

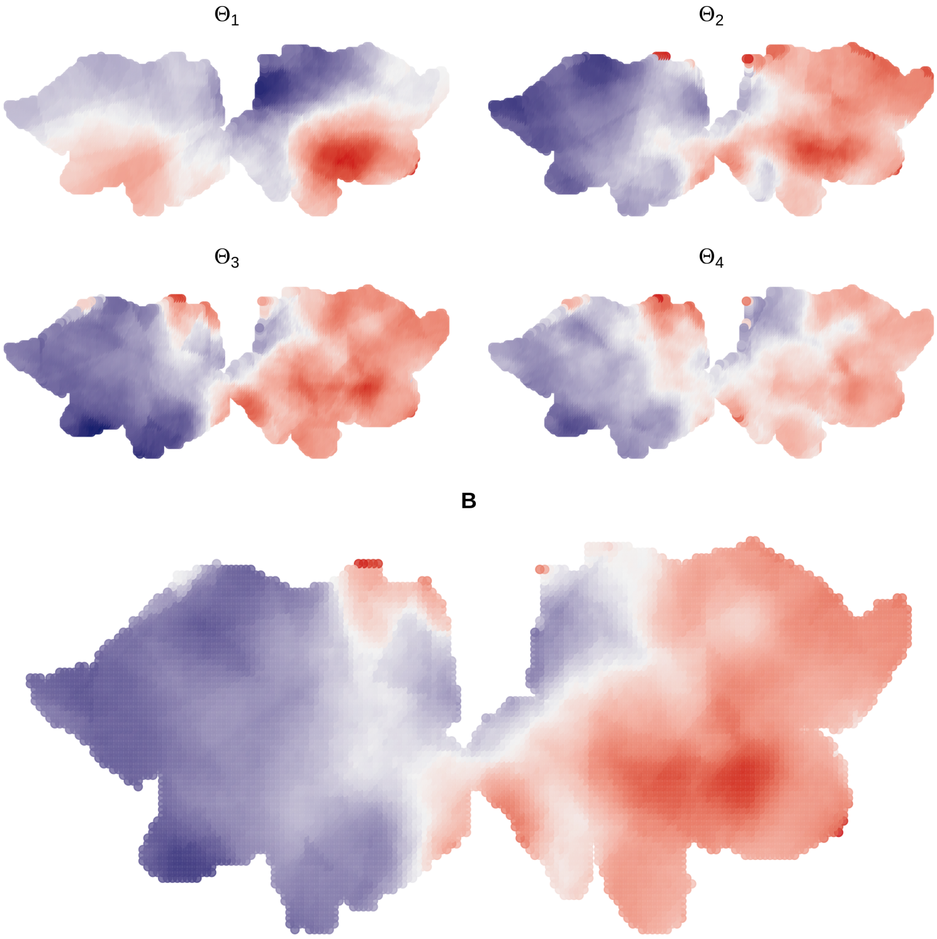


Figure 3.10: Estimate of $\mathbf{B}$ using *BM&Ms* on 69×180-dimensional images, reconstructed as the sum of $K = 4$ resolution contributions (shown on top), obtained by averaging 5000 Markov-chain Monte Carlo iterations.

3.4 Discussion

In this chapter, we have extended our Bayesian modular approach to regression models with structured tensor predictors and unknown multiscale structures. *BM&Ms* can take advantage of Algorithm 2 to decompose the regression coefficient tensor $\mathbf{B}$ into additive resolution components, facilitating easy interpretation of the results. We have applied our methodology to scalar-on-function and scalar-on-image regression and classification problems using simulated and brain imaging data. We note that our hierarchical Voronoi tessellation represents the regression coefficient tensor $\mathbf{B}$ by grouping it in cells. If large regions of $\mathbf{B}$ are zero, i.e. the corresponding locations in $\mathbf{X}_i$ have no effect on the output, this representation might be inefficient. However, it is not straightforward to unify variable selection ideas with multiscale analysis. Future research may thus focus on representations of the multiscale regression function that better adapt in cases in which $\mathbf{B}$ is sparse.

3.5 Appendix

3.5.1 Average simulation results from Section 3.2.2

FDA models occasionally failed at producing reasonable estimates, and we previously reported the median performance values in order to give a more accurate representation of the typical performance from the tested model. Below, we also report the average values for completeness.

			Estimation of $\beta(t)$ – MSE					Prediction of y_{out} – MSPE				
			FDA Fourier	FDA B-Splines	Haar Wavelets	LA10 Wavelets	***BM&Ms***	FDA Fourier	FDA B-Splines	Haar Wavelets	LA10 Wavelets	***BM&Ms***
d=128, *Blocks*												
	n=60	ρ =0.7	0.244	0.383	0.151	0.200	**0.021**	31.77	51.62	20.03	28.86	**3.66**
		ρ =0.99	0.281	0.414	0.901	0.805	**0.230**	3.13	3.81	137.23	102.41	**2.62**
	n=200	ρ =0.7	0.211	0.301	0.010	0.027	**0.0004**	19.22	30.85	1.66	2.35	**1.09**
		ρ =0.99	0.216	0.312	0.424	0.409	**0.059**	1.85	2.34	8.52	8.40	**1.22**
d=128, *Doppler*												
	n=60	ρ =0.7	0.177	0.243	0.176	0.110	**0.107**	21.68	31.45	18.07	10.98	**10.41**
		ρ =0.99	0.210	0.288	0.511	0.420	**0.214**	2.63	2.99	20.11	13.89	**2.35**
	n=200	ρ =0.7	0.154	0.185	0.026	**0.013**	0.042	13.48	18.43	2.13	**1.64**	2.66
		ρ =0.99	0.160	0.196	0.150	**0.095**	0.115	1.63	1.83	1.45	**1.30**	1.40
d=1024, *Blocks*												
	n=60	ρ =0.7	1.122	1.720	0.301	0.359	**0.098**	6847.1	9683.8	1545.0	2031.7	**558.6**
		ρ =0.99	0.454	0.964	0.385	0.370	**0.157**	960.0	1632.7	2625.7	2382.6	**349.2**
	n=200	ρ =0.7	0.140	[1]0.127	0.060	0.068	**0.01**	537.3	[1]464.6	230.9	323.8	**117.9**
		ρ =0.99	[3]0.108	[5]22.367	0.372	0.354	**0.040**	[3]174.5	[5]3852.7	2100.3	1918.4	**119.7**
d=1024, *Doppler*												
	n=60	ρ =0.7	1.010	0.502	0.317	**0.142**	0.159	5860.5	2922.7	1746.2	**840.0**	921.1
		ρ =0.99	0.300	1.076	0.274	0.156	**0.118**	751.0	1799.6	1023.7	604.6	**284.7**
	n=200	ρ =0.7	6.600	0.126	0.071	**0.026**	0.030	17873.5	400.8	304.0	**180.5**	184.1
		ρ =0.99	[1]0.415	[7]3.697	0.249	0.136	**0.058**	[1]216.8	[7]603.8	720.7	410.3	**137.1**

Table 3.4: MSE and MSPE averaged over 100 datasets for each setup

3.5.2 Estimated resolution decomposition with $K = 5$

We have previously only shown $j = 2, 3, 5$. In Figure 3.11 we report the posterior means of the complete decomposition, as approximated via MCMC.

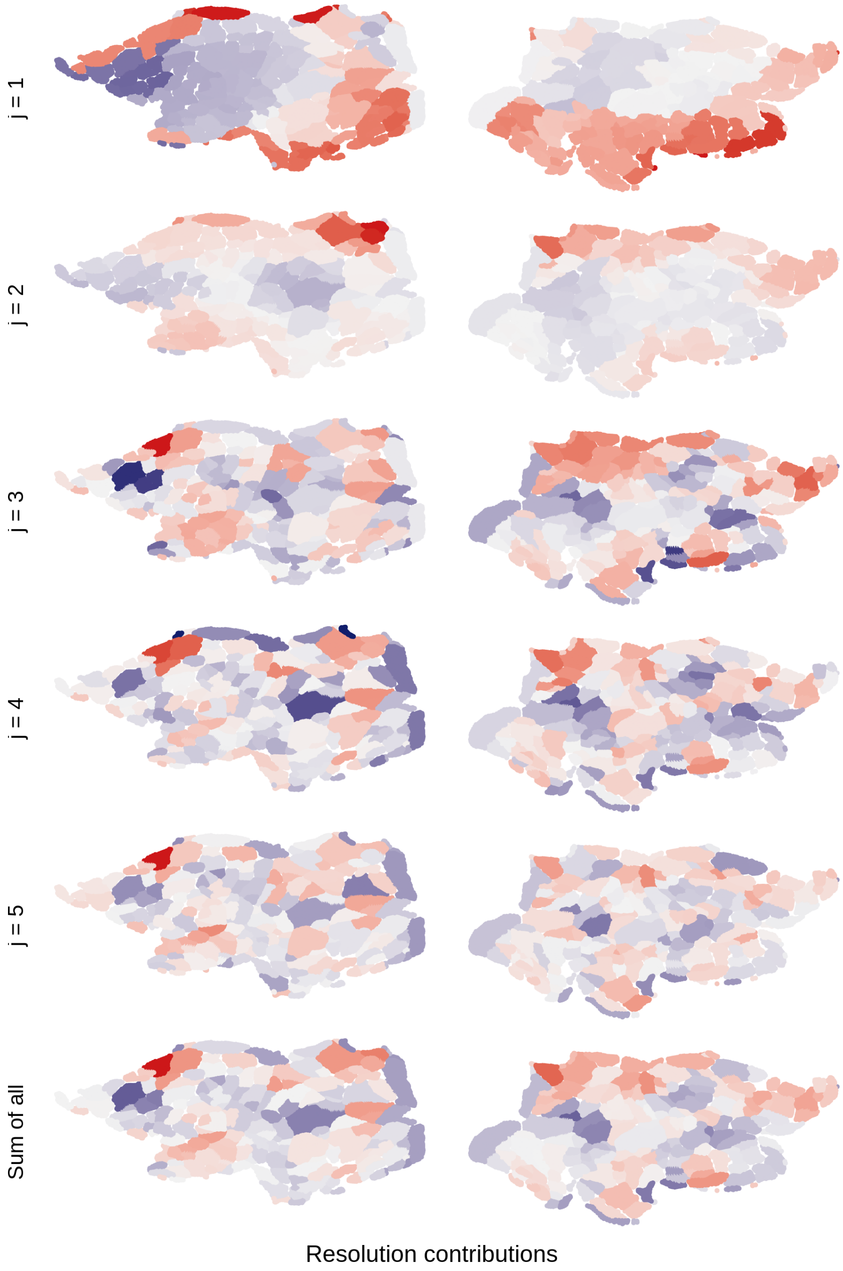


Figure 3.11: Estimated posterior means of the resolution contributions in the brain imaging classification task of Section 2.6.2.

3.5.3 Implemented models

- *BM&Ms* are implemented in Section 3.2.2 using hierarchical model (3.4) with $\alpha = 1$ and $\lambda = .1$. In Sections 3.3.3 and 3.3.4, we use our hierarchical Voronoi tessellation, $alpha = 0$ and $\lambda = .1$.

- *Haar Wavelets*: L1-penalized regression using wavelet-transformed data (Haar wavelets). See Nason (2008); Zhao et al. (2012). R package `refund.wave`.

- *LA10 Wavelets*: L1-penalized regression using wavelet-transformed data (Daubechies' Least Asymmetric wavelets of order 10). See Nason (2008); Zhao et al. (2012). R package `refund.wave`.

- *FDA Fourier* and *FDA B-Splines*: functional linear model using a Fourier-basis representation of the data, or cubic B-Splines, using cross-validation to estimate the number of basis elements; R package `fda.usc` with default parameters.

- *Functional PCR*: functional principal component regression, with cross-validated number of bases (from R package `refund`).

Chapter 4

Modular multiscale ensembles for multivariate regression

4.1 Introduction

We have focused in previous chapters on the analysis of multiscale regression methods for high-dimensional predictors, under the assumption that they shared a common structure or spatiotemporal indexing. In other terms, a single data source (e.g. an imaging device) produced all the available predictors. In Chapter 2, in fact, $x_i = (x_{i,1}, \dots, x_{i,d})$ was interpreted as output from a high-resolution sensor or device, so that a structural relationship between $x_{i,t}$ and $x_{i,s}$ could be identified. For example, we considered $x_{i,t}$ to be the activation of brain region t for subject i in Chapter 3; brain regions have a spatial arrangement which we used to construct image predictors and efficiently reduce dimensionality. Given the complex data structure, we made simplifying assumptions of linearity and normality of the errors. We now assume that predictors are unstructured, and our goal is to model their complex relationship with the output in a highly interpretable way. Our real-world data analysis will focus on the *Boston Housing* dataset, which has been extensively used in the literature.

We consider the unknown function $f(\cdot)$, which is measured with error on a sample $i = 1, \dots, n$ at inputs $x_i \in \mathbb{R}^d$:

$$y_i = f(x_i) + \varepsilon_i,$$

where y_i is a univariate output, $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2(x_i))$ is the measurement error, possibly depending on inputs through an unknown function $\sigma^2(\cdot)$. Restrictive assumptions on $f(\cdot)$ – such as linearity – would result in interpretable models, possibly at the expense of performance. On the other hand, black-box methods may greatly improve performance without providing the tools to interpret their results. Our goal is to estimate $f(\cdot)$ flexibly in a way that allows to easily disseminate "stylized facts" about the results which can be understood by non-experts or practicioners with null or minimal background on statistical learning methods, with a small cost to performance compared to state-of-the-art methods.

We approximate $f(x)$ by an additive model $f(x) \approx h(x) = \sum_{j=1}^{d} h_j(x)$, where each h_j is decomposed additively into coarse-to-fine scale contributions – i.e. $h_j(x) = \sum_{r=1}^{R_j} g_{j,r}(x_j)$. We label $g_{j,r}$ as the r^{th} scale contribution of variable j to the regression function, and assume it is a step-function. Therefore, $g_{j,r}$ splits the j^{th} covariate space into regions with comparable effects on the response. At coarser scales, these regions will be larger. We assume $g_{j,r}$ has jumps at $S_{j,r} = \{s_1, \dots, s_{k_{j,r}}\}$ such that $S_{j,r} \subset S_{j,r+1}$; in other words, we recursively partition the regressors' space.

Our approach is most related to methods based on local averaging of partitions which are possibly found recursively, and methods for multiscale regression. A first example is classical histogram regression (see e.g. Györfi et al. 2002, Ch. 4, Nobel 1996), which is based on partitioning the input space into cells and performing local averages, thus modeling the regression function as piecewise constant along partitions of the covariate space. The resulting estimator is $m(x) = \sum_{i=1}^{n} y_i I_{\{x_i \in A(x)\}} / \sum_{i=1}^{n} I_{\{x_i \in A(x)\}}$, where $A(x)$

is the cell that includes x and $I\{\cdot\}$ is the indicator function. If $d = 1$, one can view this problem from the perspective of change-point detection models; these methods date back to Scott and Knott (1974); Auger and Lawrence (1989); Green (1995), and more recently Frick et al. (2014).

In more general cases, dependence on x can be introduced by generating covariate-dependent partitions, by using covariate-dependent models for each partition, or a combination of both. A piecewise constant model with Normal errors results in outputs in cell A being modeled as $y|\{x \in A\} \sim N(\mu_A, \sigma_A^2)$. A Bayesian researcher can conveniently assign conjugate priors to the local mean and variance at each cell. Given the partition, cells may be assumed independent a priori and their parameters marginalized out. This implies that (1) the partition itself is the only parameter of interest and (2) prior parameters remain important. Finding "good" partitions is difficult. One can consider data-dependent *statistically equivalent* cells, which include approximately the same number of observations (Anderson, 1966), but this may be inefficient or restrictive. In fact, smaller cell sizes potentially provide more accurate estimates, but reducing cell dimension inevitably decreases the local sample size; therefore, inference will be heavily influenced by the prior parameters.

As a partial solution, decision trees approach the problem via greedy recursive partitions of the space of predictors. These methods date back to Friedman (1977) and (Breiman et al., 1984) and require multiple rounds of post-processing to avoid overfitting. They have led to the development of a multitude of similarly popular models based on tree ensembles, including boosting, bagging, and random forests Freund and Schapire (1997); Friedman (2001); Breiman (1996, 2001). Bayesian additive regression tree models such as the Bayesian CART (Chipman et al., 1998) and the BART (Chipman et al., 2010) assign a prior to the tree or tree ensemble and sample from its posterior distribution by stochastic search. These models can accommodate non-constant variances of the errors, but such extensions in tree ensemble models have only recently been developed (Pratola et al., 2017). Also recent is their theoretical support in the form of asymptotic results (van der Pas and Rockova, 2017; Rockova and van der Pas, 2017). However, as noted by Chipman et al. (1998) and Müller and Quintana (2004), mixing of MCMC is problematic in tree models given the complex parameter space. Furthermore, direct interpretations on the trees are made more difficult when many different trees have equivalent performance. This issue is exacerbated when tens of trees are used jointly in ensembles, as each of them carries minimal information. Interpretability is difficult to define, but has received increasing attention in machine learning and computer science as a consequence of the success and proliferation of black-box methods. See e.g. Letham et al. (2015); Doshi-Velez and Kim (2017); Samek et al. (2017); Weller (2017); Guidotti et al. (2018).

In order to simplify interpretations with decision trees, one could pair one tree with each of the d predictors, resulting in an ensemble of d trees. The improvement in interpretability follows from the resulting implicit multiscale decompositions along each axis. For example, the tree root and larger branches for predictor j would be linked to areas of *major* impact on the response. Smaller branches and leaves would instead correspond to *minor* effects. In other words, roots, branches, and leaves represent increasing resolution levels. Trees corresponding to unimportant predictors would not grow. Unfortunately, decision tree methods only assign values to terminal nodes; this prevents multiscale interpretations as the split order does not matter.

With our goal of simultaneously estimating the regression function at different scales, one can use wavelet methods; Antoniadis (2007) and Vidakovic (2009) provide in-depth treatments. These methods are popular as denoising tools in signal processing, as they usually require a univariate or bivariate regressor sampled on a grid of equally-spaced points, a number of observations that is power of 2 (dyadic data), and a constant measurement error. Given these assumptions, wavelet methods enjoy good properties (Donoho and Johnstone, 1994, 1995). From a signal processing perspective, Sardy and Tseng (2004) use wavelets in additive models on very large samples and with high signal-to-noise. Other extensions of wavelet methods to allow for more flexibility have been developed in the literature at the expense of interpretability or increased preprocessing complexity (Kovac and Silverman, 2000; Kerkyacharian and Picard, 2004; Fryzlewicz, 2007).

In this Chapter we propose a Bayesian *MOdular NOnparametric Multiscale Ensemble Regression* (`MONOMER`) model to estimate the regression function flexibly and simultaneously at multiple scales of the predictors. We use an ensemble of univariate trees, constructed modularly to simultaneously obtain multiscale inferences on each margin of the regression function. Our additive representation decomposes the regression function into regressor and resolution contributions. Our approach is similar to Bayesian tree ensemble models, as we use multiple trees to recursively split the predictor space into and estimate local means and variances. Unlike Bayesian tree models, we rearrange the trees to keep each covariate separate. Like Haar-wavelet regression, `MONOMER` decomposes the regression function into coarse-to-fine step functions; our approach is more flexible, requires less preprocessing, and can be used with $d > 2$, irregularly spaced predictors, heteroscedastic errors. We obtain well-defined multiscale decompositions and improve mixing of MCMC by modularity in our Bayesian model. If we do not consider interactions, the contribution of each regressor to $E(Y|x)$ can be isolated. The multiresolution decomposition of each individual h_j hierarchically identifies *major* and *minor* (if any) effects of a predictor on the output, corresponding to low and high resolutions, respectively. In `MONOMER`, Bayesian model averaging will smooth the estimates of each piecewise constant component in regions whose boundary is uncertain. The final estimate for each margin h_j is quickly obtained by summing over all resolution contributions.

4.2 The MONOMER model

Our approach corresponds to the following overall model for the regression function:

$$E(Y|x) = \sum_{j=1}^{d} h_j(x) = \sum_{j=1}^{d} \sum_{r=1}^{R_j} g_{j,r}(x_j), \tag{4.1}$$

where we assume $x \in \mathbb{R}^d$, and for $r = 1, \dots, R_j$, the functions $g_{j,r}$ are piecewise constant with jumps at $S_{j,r} = \{s_1, \dots, s_{k_{j,r}}\}$. Along the x_j axis this implies that $h_j(x_j) = \sum_{r=1}^{R_j} g_{j,r}(x_j)$ is also a step function, and it has jumps at $\cup_{r=1}^{R_j} S_{j,r}$. R_j represents the number of scales for variable j, we take $R_j = R$ for all j for simplicity. We observe a sample (y_i, x_i), with $i = 1, \dots, n$, and assume that $x_i \in \mathbb{R}^d$.

Since our goal is to obtain a *multiresolution* decomposition, we assume that $S_{j,r} \subset S_{j,r+1}$. While not strictly necessary, this assumption facilitates the representation of the multiresolution structure by a tree of sequential splits. There is a one-to-one correspondence between flat areas of $g_{j,r}$, split points $S_{j,r}$ and the induced partition $\mathcal{A}_{j,r}$ of $\mathbb{R}$. Furthermore, the j-variable resolution structure $\mathcal{S}_j = \{S_{j,r} : \ r = 1, \dots, R_j\}$ can be represented by a decision tree that recursively splits the regressor space into finer partitions. Consequently, when considering multivariate predictors, we obtain an ensemble of scaling trees. Each of the d trees can be "cut" at some resolution $S_j^{(r)} \in \mathcal{S}_j$, which corresponds to partitioning $\mathbb{R}$ into adjacent intervals using the partition $\mathcal{A}_j^{(r)} = \{A_{j,1}^{(r)}, \dots, A_{j,k_j^{(r)}}^{(r)}\}$, where $k_j^{(r)} = 1 + |S_j^{(r)}|$ is the number of intervals into which the space of regressor j is split. The resulting collection of regressor trees cut at resolutions $S_1^{(r)}, \dots, S_p^{(r)}$ will correspond to a partition $\mathcal{A}^{(r)} = \{\mathbf{A}_1^{(r)}, \dots, \mathbf{A}_q^{(r)}\}$ of $\mathbb{R}^d$, where $q = \prod_{j=1}^d k_j^{(r)}$. Suppose $A = (A_1 \times \cdots \times A_d) \in \mathcal{A}^{(r)}$. We model the mean of Y falling in A as the sum of the contributions of resolutions up to $S_1^{(r)}, \dots, S_d^{(r)}$ from each regressor. In other words,

$$E(Y|A, \mu, \sigma^2) = \sum_{\substack{l<r \\ s\in A_1}} \mu_{1,s}^{(l)} + \cdots + \sum_{\substack{l<r \\ s\in A_d}} \mu_{d,s}^{(l)},$$

where $\mu_{j,k}^{(z)}$ is variable j's contribution to cell k at resolution z. We focus here on the decomposition of the mean; conditional on the partition, we can similarly model the errors to be additive and independent on each margin. Observations in a cell are thus modeled as sharing a single mean and variance. An alternative is to unlink mean and variance scaling trees and model the variances via product decompositions as in Pratola et al. (2017), or keep it constant. We see a graphical depiction of the decomposition of a two-dimensional surface into resolution contributions in Figure 4.1.

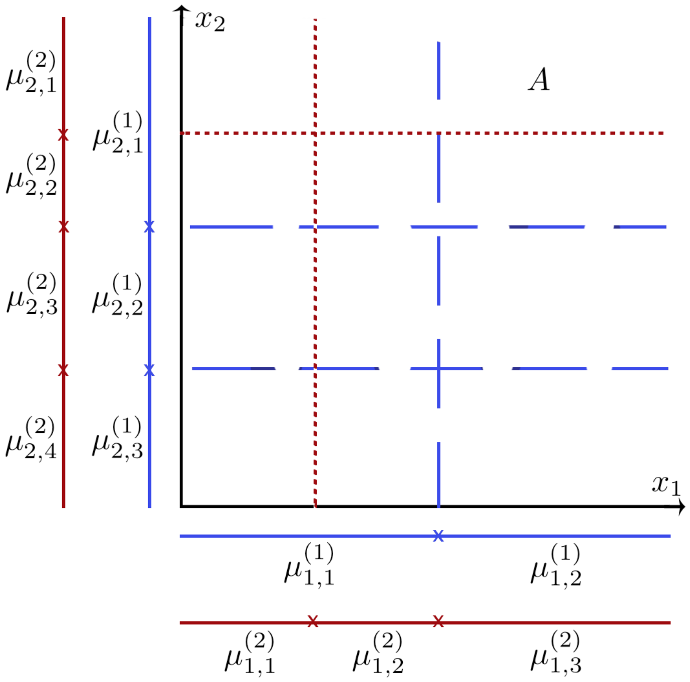


Figure 4.1: If $d = 2$ we can represent multiscale splitting rules in the cartesian plane. If $(x_1, x_2) \in A$ then $E(Y|x, \mu) = \mu_{1,2}^{(1)} + \mu_{1,3}^{(2)} + \mu_{2,1}^{(1)} + \mu_{2,1}^{(2)}$.

Figure 4.1 highlights an identifiability problem which is analogous to the one presented in Chapter 2. In fact, considering the x_1 margin, it is difficult to disambiguate between $\mu_{1,1}^{(1)}$ at the coarse scale and $\mu_{1,1}^{(2)}, \mu_{1,2}^{(2)}$ at the finer if all are considered jointly in a single model. We may be able to solve this problem by careful prior specification; however, it may not be possible to use simple computational algorithms to sample from the posterior distribution when the resolution is unknown. A more straightforward and effective way is to instead modularize the computation of the posterior distribution for scale contribution r.

4.3 Computing the posterior distribution

4.3.1 Univariate case

We start by considering the case $d = 1$; for $d > 1$ we detail Algorithm 3 below. For this reason, we temporarily drop the j subscripts. Given the relative simplicity of the data at hand, we can model the relation ship between output and input as an additive decomposition on the mean, and multiplicative on the variance. In other words, fixing the number of resolutions to $R_1 = R$, we assume for all $i = 1, \dots, n$:

$$y_i = g_1(x_i) + \cdots + g_R(x_i) + \sigma_1(x_i) \cdots \sigma_R(x_i)\varepsilon_i,$$

where $\varepsilon_i \overset{iid}{\sim} N(0,1)$, $g_r(x) = \sum_{s=1}^{S_r} \theta_{r,s}\mathbb{1}_{A_{r,s}}(x)$ and $\sigma_r(x) = \sum_{s=1}^{S_r} \tau_{r,s}\mathbb{1}_{A_{r,s}}(x)$ with $\tau_{r,s} > 0$. Intuitively, all observations (y_i, x_i) with inputs falling in the s-th cell of resolution r, i.e. $x_i \in A_{r,s}$, are associated to a mean $\theta_{r,s}$ and a standard deviation $\tau_{r,s}$. A modeling approach of this kind is overcomplete, since we assume that all step functions g_r with $r < R$ are such that $S_r \subset S_{r+1} \subset \cdot \subset S_R$. For this reason, it is difficult to disambiguate the contribution of each resolution; this modeling approach results in poorly identifiable parameters, refer to Chapter 2, sections 2.1 and 2.4 for a discussion.

Given a resolution structure $\mathcal{S}$, we pair a module to each resolution $S^{(r)} \in \mathcal{S}$. For each observation (y, x) we fix $e^{(0)} = \frac{y-\bar{y}}{\sigma_y}$, and construct module r by considering:

- a model $e^{(r-1)} \sim N\left(\mu_1^{(r)} + \cdots + \mu_{k^{(r)}}^{(r)}, \sigma_1^{2(r)} \cdots \sigma_{k^{(r)}}^{2(r)}\right)$,
- setting $\theta^{(r)} = \left(\mu_h^{(r)}, \sigma_h^{2(r)}\right)$, a prior for $\theta^{(r)}|(\theta^{(1)}, \ldots, \theta^{(r-1)})$, for $h = 1, \ldots, k^{(r)}$, where $k^{(r)}$ is the number of cells in which $\mathbb{R}$ is split at resolution S_r. To facilitate computation, we choose independent conjugate priors for each cell, i.e. $\sigma_h^{2(r)} \sim \text{N.InvG}(\alpha, \beta)$, and $\mu_h^{(r)}|\sigma_h^{2(r)} \sim N(0, \sigma_h^{2(r)}/\kappa)$
- standardized residuals $e^{(r)} = \frac{e^{(r-1)} - \sum_{r<k^{(r)}} \mu_r}{\prod_{r<k^{(r)}} \sigma_r}$.

Given resolution $S^{(r)}$, observations on different cells will be modeled as independent; as a result, the posterior distribution of $\theta^{(r)}$ arises from this model as the collection of the posterior distributions from each cell. The *modular* posterior distribution instead collects all resolutions up to (r), i.e. for $\Theta_{1:r} = (\theta^{(1)}, \ldots, \theta^{(r)})$, given $\mathcal{S}$:

$$p_M(\theta^{(1)}, \ldots, \theta^{(r)}|y) = p_1(\theta^{(1)}|e^{(0)}, S_1) \cdots p_r(\theta^{(r)}|e^{(r-1)}, S_r)$$

Computing the posterior distribution in this case is immediate, as with $p = 1$ we can directly use Algorithm 2 as implemented in Chapter 3.2. The regression "tensor" in this case is nothing other than the identity matrix of size n. Also notice that it is unnecessary to transform x, as only the quantiles of the observed $(x_1, \ldots, x_n)$ matter.

4.3.2 Multivariate case

We can extend the above to the case $d > 1$. The overall model becomes an ensemble of d trees. We can implement a Gibbs sampler to sample the full conditional distribution of h_{j^*}, given the other $(d-1)$ trees and y, which will update the j^{th} tree parameters parameters in block. This is similar to Chipman et al. (2010), who link it to the algorithm by Hastie and Tibshirani (2000). Since our focus is on the (additive) regression function, we make the simplifying assumption of constant but unknown variance, i.e. $\sigma^2(x) = \sigma^2$ for all $x \in \mathbb{R}$. We defined $\mathcal{S}_j$ earlier as the resolution structure for regressor j, which can be described by a tree, and $\mathcal{A}_j$ as the set of partitions of $\mathbb{R}$ corresponding to $\mathcal{S}_j$. We can similarly introduce $\mathcal{M}_j$ as the set of multilayer terminal node means for a given $\mathcal{S}_j$.

```
Initialize: Start with $h_j(x_j; \mathcal{S}_j^{(0)}, \mathcal{M}_j^{(0)}, \sigma^{2(0)})$.
for $t \in \{1, \dots, T\}$ do
    for $j \in \{1, \dots, p\}$ do
        • Calculate $y_{-j^*} = \frac{y - \sum_{j \neq j^*} h_j(x_j; \mathcal{S}_j^{(t)}, \mathcal{M}_j^{(t)})}{\sqrt{\sigma^{2(t)}}}$;
        • Sort the data using the order of $x_j$;
        • Sample $\mathcal{S}_j^{(t)} | y_{-j^*}$ using Algorithm 2 as implemented in Chapter 3.2. It
          suffices to set the design matrix to the identity matrix of size $n$;
        • Sample $\mathcal{M}_j^{(t)}$ given $y_{-j^*}, \mathcal{S}_j^{(t)}, \sigma^{2(t)}$;
    end
    • Sample $\sigma^{2(t)}$ given $y, \mathcal{S}_j^{(t)}$;
end
```

Algorithm 3: Sampling from $p(\mathcal{S}, \mathcal{M}, \sigma^2 | y)$.

4.4 Asymptotic properties

We now provide a consistency result: in words, if f_0 is the true regression function generating the data, under some conditions the modular posterior distribution concentrates around f_0 as the sample size increases.

Assumption 4.4.1. The observables $Y^{(n)} = (Y_1, \dots, Y_n)'$ are assumed of the form $Y_i = f_0(x_i) + \varepsilon_i$, where $\varepsilon_i \sim N(0, \sigma_0^2)$, for $i = 1, \dots n$, and $x_i = i/n$.

Assumption 4.4.2. The sequence of sample sizes is $n \equiv n_m = 2^m$ for $m = 1, 2, \dots$.

For any $f : [0,1] \to \mathbb{R}$, define $\|f - f_0\|_n^2 = n^{-1} \sum_{i=1}^n (f(x_i) - f_0(x_1))$. We require in Assumption 4.4.3 that f_0 admits an approximation via step-functions on dyadic intervals in terms of the $\|\cdot\|_n$ metric; it is satisfied e.g. by any $f_0 \in C[0,1]$.

Assumption 4.4.3. There exists a sequence of functions $\left(f_{\beta_0^{(n)}}\right)_{n=1}^{\infty}$ of the form $f_{\beta_0^{(n)}}(x) = \sum_{j=1}^{2^{K_0-1}} \beta_{0,j}^{(n)} \mathbb{1}_{\Omega_j^{(n)}}(x)$, where $\Omega_1^{(n)} = \left[0, \frac{1}{2^{K_0-1}}\right]$ and $\Omega_j^{(n)} \left(\frac{j-1}{2^{K_0-1}}, \frac{j}{2^{K_0-1}}\right]$, for $j = 2, \dots, 2^{K_0-1}$, such that $\|f_{\beta_0^{(n)}} - f_0\|_n = o(1)$ as $n \to \infty$. Here, $\mathbb{1}_A(x) = 1$ if $x \in A$, 0 otherwise.

Set $e_0^{(0)} = 0$ and $e_1^{(n)} = Y^{(n)}$. Then for $j = 1, \dots, K_n$ we configure a `MONOMER` model to use modules of this form:

$$
\begin{aligned}
&X_j \equiv X_j^{(n)} \equiv I_{p_j} \otimes \mathbf{1}_{\frac{n}{p_j}} \qquad p_j = 2^{j-1}, \quad j = 1, \dots, K_n \\
&\sigma_j^{2(n)} \propto \frac{1}{\sigma_j^2} \qquad \theta_j^{(n)} \mid \sigma_j^{2(n)} \sim N\left(0, n\sigma_j^{2(n)} (X_j' X_j)^{-1}\right) \\
&e_j^{(n)} | \theta_j^{(n)}, \sigma_j^{2(n)} \sim N\left(\theta_j^{(n)}, \sigma_j^{2(n)}\right)
\end{aligned}
\tag{4.2}
$$

where $e_j^{(n)} = e_{j-1}^{(n)} - X_j\theta_{j-1}^{(n)}$. As a result, we obtain

$$\theta_j^{(n)} \mid Y^{(n)}, \theta_{1:j-1}^{(n)} \sim \mathcal{T}\left(n, \frac{n}{n+1}\hat{\mu}_j^{(n)}, S^{(n)}\right)$$

where $\hat{\mu}_j^{(n)} = (X_j'X_j)^{-1}X_j'e_j^{(n)}$, $S^{(n)} = \frac{(X_j'X_j)^{-1}}{n+1}\left(s_j^{2(n)} + \hat{\mu}_j^{(n)'}X_j'X_j\hat{\mu}_j^{(n)}/(n+1)\right)$, and $s_j^{2(n)} = (e_j^{(n)} - X_j\hat{\mu}_j^{(n)})'(e_j^{(n)} - X_j\hat{\mu}_j^{(n)})$.

Lemma 4.4.1. For $l = 2, \dots, K_n$,

$$P_{f_0}^{(n)} E\left(\sum_{p=1}^{2^{l-1}} Var(\tilde{\theta}_{l,p} \mid Y^{(n)}, \tilde{\theta}_{1:l-1}^{(n)}) \mid Y^{(n)}\right) \leq$$
$$\frac{2^l}{(n+1)^2} P_{f_0}^{(n)} E\left(\sum_{t=1}^{p_l-1} Var(\tilde{\theta}_{l-1,t}^{(n)} \mid Y^{(n)}, \tilde{\theta}_{1:l-2}^{(n)}) \mid Y^{(n)}\right) + 2^{l-1}R_{n,2}$$

where

$$R_{n,l} = P_{f_0}\left(\frac{1}{n+1}\sum_{t=1}^{p_l}\left(\frac{p_l}{n}\sum_{i\in I_{l,t}^{(n)}}\left(Y_i - \frac{p_l}{n}\sum_{r\in I_{l,t}^{(n)}} Y_i\right)\right)^2\right)$$
$$+\frac{1}{2(n+1)^2}\sum_{t=1}^{p_l-1}\left(\frac{p_l}{n}\sum_{i\in I_{l,2t-1}^{(n)}} Y_i + \frac{p_l}{n}\sum_{i\in I_{l,2t}^{(n)}} Y_i\right)^2\Bigg)$$

Proof. Observe that

$$E\left(\theta_l^{(n)} \mid Y^{(n)}, \tilde{\theta}_{1:l-1}^{(n)}\right) =$$
$$\frac{n}{n+1}\left(\sum_{i\in I_{l,1}^{(n)}} Y_i - \sum_{r=1}^{l-1}\tilde{\theta}_{r,\Omega_{l,1}}, \dots, \sum_{i\in I_{l,p_l}^{(n)}} Y_i - \sum_{r=1}^{l-1}\tilde{\theta}_{r,\Omega_{l,p_l}}\right)'$$

and

$$\begin{aligned}
&Var\left(\theta_l^{(n)} \mid Y^{(n)}, \tilde{\theta}_{1:l-1}^{(n)}\right) = \\
&= \frac{n}{n+1}\left(\frac{s_{n,l}^2}{n} + \frac{e_l^{(n)'}X_l(X_l'X_l)^{-1}X_l'e_l^{(n)}}{n(n+1)}\right) I_{p_l}\frac{p_l}{n} \\
&= \frac{n}{n+1}\left(\frac{1}{n}\sum_{t=1}^{p_l}\sum_{i\in I_{l,t}^{(n)}}\left(Y_i - \frac{p_l}{n}\sum_{r\in I_{l,t}^{(n)}} Y_r\right)^2 + \frac{p_l}{n^2(n+1)}e_l^{(n)'}X_lX_l'e_l^{(n)}\right) I_{p_l}\frac{p_l}{n} \\
&= \frac{n}{n+1}\left(\frac{1}{n}\sum_{t=1}^{p_l}\left(\frac{p_l}{n}\sum_{i\in I_{l,t}^{(n)}}\left(Y_i - \frac{p_l}{n}\sum_{r\in I_{l,t}^{(n)}} Y_r\right)\right)^2\right)
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{1}{n(n+1)} \sum_{t=1}^{p_l} \left(\frac{p_l}{n} \sum_{i \in I_{l,t}^{(n)}} Y_i - \sum_{r=1}^{l-1} \tilde{\theta}^{(n)}_{r,\Omega_{l,t}} \right)^2 \Bigg) I_{p_l} \\
&= \left(\frac{1}{n+1} \sum_{t=1}^{p_l} \left(\frac{p_l}{n} \sum_{i \in I_{l,t}^{(n)}} \left(Y_i - \frac{p_l}{n} \sum_{r \in I_{l,t}^{(n)}} Y_r \right)^2 \right) \right) I_{p_l} + \\
&\quad \left(\frac{1}{(n+1)^2} \sum_{t=1}^{p_l} \left(\frac{p_l}{n} \sum_{i \in I_{l,t}^{(n)}} Y_i - \sum_{r=1}^{l-1} \tilde{\theta}^{(n)}_{r,\Omega_{l,t}} \right)^2 \right) I_{p_l} \\
&= T_{n,l} I_{p_l} + U_{n,l} I_{p_l}.
\end{aligned}
$$

Above, $\tilde{\theta}_{r,\Omega_{l,t}}$ is the coefficient corresponding to the unique integer h in the set $\{1, \ldots, 2^{r-1}\}$ such that $\Omega_{r,h} \cap \Omega_{l,t} \neq \emptyset$. Consequently, we have that

$$
\begin{aligned}
& P_{f_0}^{(n)} E \left(\sum_{t=1}^{p_l} Var \left(\tilde{\theta}_{l,t} \mid Y^{(n)}, \tilde{\theta}^{(n)}_{1:l-1} \right) \mid Y^{(n)} \right) \\
& = p_l P_{f_0}^{(n)} E \left(T_{n,l} + E_{n,l} \mid Y^{(n)} \right) \\
& \leq p_l P_{f_0}^{(n)} T_{n,l} + \frac{2 p_l}{(n+1)^2} P_{f_0}^{(n)} E \left(\sum_{t=1}^{p_{l-1}} Var \left(\tilde{\theta}^{(n)}_{l-1,t} \mid Y^{(n)}, \tilde{\theta}^{(n)}_{1:l-2} \right) \mid Y^{(n)} \right) \\
& + \frac{p_l}{(n+1)^2} \sum_{t=1}^{p_{l-1}} P_{f_0}^{(n)} \left(\frac{1}{2} \left(\frac{p_l}{n} \sum_{i \in I_{l,2t-1}^{(n)}} Y_i + \frac{p_l}{n} \sum_{i \in I_{l,2t}^{(n)}} Y_i \right)^2 \right)
\end{aligned}
$$

which is the result. □

Proposition 4.4.1. *Let K_n be a sequence of powers of two satisfying $K_n \to \infty$, $n \to \infty$, $K_n < log_2 n$. Then, under the assumptions above, for all $\varepsilon > 0$*

$$
P_{f_0}^{(n)} \Pi^{(n)} \left\{ \| f_0 - f_{\tilde{\theta}^{(n)}} \|_n > 2\varepsilon \right\} = o(1) \qquad \text{as } n \to \infty,
$$

where $f_{\tilde{\theta}^{(n)}}(x) = \sum_{j=1}^{K_n} \sum_{s=1}^{2^{j-1}} \theta^{(n)}_{j,s} \mathbb{1}_{\Omega_{j,s}}(x)$, $x \in [0,1]$, and $\Omega_1 = [0,1]$, $\Omega_{j,s} = \Omega_{j+1,2s-1} \sqcup \Omega_{j+1,2s}$ for $s = 1, \ldots, 2^{j-1}$ and $j = 1, \ldots, K_n - 1$.

Proof. By Assumption 4.4.3, there exists $n_0 \in \mathbb{N}$ such that $\forall n \geq n_0$, $\| f_{\beta_0^{(n)}} - f_0 \|_n \leq \varepsilon$. Without loss of generality, we can consider $K_0(n) = K_n$. For large n and using Markov's inequality:

$$
\begin{aligned}
& P_{f_0}^{(n)} \Pi^{(n)} \left(\| f_0 - f_{\tilde{\theta}^{(n)}} \|_n > 2\varepsilon \right) \\
& \leq P_{f_0}^{(n)} \Pi^{(n)} \left(\| f_0 - f_{\beta_0^{(n)}} \|_n + \| f_{\beta_0^{(n)}} - f_{\tilde{\theta}^{(n)}} \|_n > 2\varepsilon \right) \\
& \leq P_{f_0}^{(n)} \Pi^{(n)} \left(\| f_{\beta_0^{(n)}} - f_{\tilde{\theta}^{(n)}} \|_n > \varepsilon \right)
\end{aligned}
$$

$$
\begin{aligned}
&\leq P_{f_0}^{(n)} \sum_{j=1}^{2^{K_n-1}} \Pi^{(n)} \left(|\beta_{0,j}^{(n)} - \sum_{t=1}^{K_n} \tilde{\theta}^{(n)}_{t,\Omega_{K_n,j}}| > n^{1/2}\varepsilon \right) \\
&\leq P_{f_0}^{(n)} \frac{1}{n\varepsilon^2} \sum_{j=1}^{2^{K_n-1}} E\left((\beta_{0,j}^{(n)} - \sum_{t=1}^{K_n} \tilde{\theta}^{(n)}_{t,\Omega_{K_n,j}})^2 \mid Y^{(n)} \right) \\
&= P_{f_0}^{(n)} \frac{1}{n\varepsilon} \sum_{j=1}^{2^{K_n-1}} E\left(E\left((\beta_{0,j}^{(n)} - \sum_{t=1}^{K_n} \tilde{\theta}^{(n)}_{t,\Omega_{K_n,j}})^2 \mid Y^{(n)}, \tilde{\theta}^{(n)}_{1:K_n-1} \right) \mid Y^{(n)} \right) \\
&= \frac{1}{n\varepsilon^2} P_{f_0}^{(n)} \sum_{j=1}^{p_{K_n}} \left(\beta_{0,j}^{(n)} - \frac{p_{K_n}}{n} \sum_{i \in I^{(n)}_{K_n,j}} Y_i^{(n)} \right)^2 + \\
&\quad + \frac{1}{n\varepsilon^2} P_{f_0}^{(n)} E\left(\sum_{j=1}^{p_{K_n}} Var\left(\tilde{\theta}^{(n)}_{K_n,j} \mid Y^{(n)}, \tilde{\theta}^{(n)}_{1:K_n-1} \right) \mid Y^{(n)} \right).
\end{aligned}
$$

The first term

$$
\begin{aligned}
&\frac{1}{n\varepsilon^2} P_{f_0}^{(n)} \sum_{j=1}^{p_{K_n}} \left(\beta_{0,j}^{(n)} - \frac{p_{K_n}}{n} \sum_{i \in I^{(n)}_{K_n,j}} Y_i^{(n)} \right)^2 \\
&= \frac{1}{n\varepsilon^2} P_{f_0}^{(n)} \sum_{j=1}^{p_{K_n}} \left(\frac{\sum_{i \in I^{(n)}_{K_n,j}} (Y_i - \beta_{0,j}^{(n)})}{n/p_{K_n}} \right)^2 \\
&\leq \frac{1}{n\varepsilon^2} P_{f_0}^{(n)} \sum_{j=1}^{p_{K_n}} \frac{p_{K_n}}{n} \left(\sum_{i \in I^{(n)}_{K_n,j}} \left(Y_i - \beta_{0,j}^{(n)} \right)^2 \right) \\
&= \frac{p_{K_n}}{n} \frac{\|f_0 - f_{\beta_0^{(n)}}\|^2}{\varepsilon^2} + \frac{p_{K_n}\sigma_0^2}{n\varepsilon^2} = o(1).
\end{aligned}
$$

The second term

$$
\begin{aligned}
&\frac{1}{n\varepsilon^2} P_{f_0}^{(n)} E\left(\sum_{j=1}^{p_{K_n}} Var\left(\tilde{\theta}^{(n)}_{K_n,j} \mid Y^{(n)}, \tilde{\theta}^{(n)}_{1:K_n-1} \right) \mid Y^{(n)} \right) \\
&= \frac{1}{n\varepsilon^2} P_{f_0}^{(n)} \sum_{j=1}^{p_{K_n}} E\left(Var\left(\tilde{\theta}^{(n)}_{K_n,j} \mid Y^{(n)}, \tilde{\theta}^{(n)}_{1:K_n-1} \right) \mid Y^{(n)} \right) \\
&\leq \frac{2^{K_n-1}}{n\varepsilon^2} R_{n,K_n} + \frac{2^{K_n}}{n(n+1)^2\varepsilon^2} P_{f_0}^{(n)} E\left(\sum_{t=1}^{p_{K_n-1}} Var\left(\tilde{\theta}_{K_n-1,t} \mid Y^{(n)}, \tilde{\theta}^{(n)}_{1:K_n-1} \right) \mid Y^{(n)} \right) \\
&\cdots \\
&\leq \frac{1}{\varepsilon^2} \sum_{l=0}^{K_n-2} \frac{2^{(l+1)K_n-2l}}{n(n+1)^{2l}} R_{n,K_n-l} + \frac{2^{\sum_{l=2}^{K_n} l}}{n(n+1)^{2(K_n-1)}} P_{f_0}^{(n)} Var(\tilde{\theta}_1^{(n)} \mid Y^{(n)}) \\
&\leq \frac{1}{\varepsilon^2} \left(\max_{2 \leq l \leq K_n} R_{n,l} \right) \sum_{l=0}^{K_n-2} \frac{2^{(l+1)K_n-2l}}{n(n+1)^{2l}} + o(1).
\end{aligned}
$$

To conclude, $\max_{2\leq l\leq K_n} R_{n,l} = o(1)$; then, using $n = 2^m$ and $K_n < \log_2 n$, we get, for m large:

$$\sum_{l=0}^{K_n-2} \frac{2^{(l+1)K_n-2l}}{n(n+1)^{2l}} \leq \sum_{l=0}^{K_n-2} \frac{2^{(l+1)m-2l}}{2^{m(1+2l)}}$$
$$= \sum_{l=0}^{K_n-2} 2^{-(m+2)l} \leq \frac{1-2^{-(m+2)(m-1)}}{1-2^{-(m+2)}} \approx 1$$

□

We proceed by showing asymptotic normality of the posterior distributions from each module. In the following, we rephrase the assumptions of Theorem 2.1 in Kleijn and van der Vaart (2012) in order to suit `MONOMER`. As usual, we deal with a two-module setting for the sake of clarity and without loss of generality.

Denote by $\|\cdot\|$ the Euclidean norm and let $B(u,\epsilon)$ represent an Euclidean distance ball centered at u, with radius ϵ.

Assumption 4.4.4. (First module) For some $\theta_1^* \in \Theta_1 \subset \mathbb{R}^{d_1}$, there exist: a sequence of random vectors Δ_{n,θ_1^*} (bounded in probability), a non-singular matrix $V_{\theta_1^*}$ and a norming rate $\delta_{1,n}$ such that, for every compact $K_1 \subset \mathbb{R}^{d_1}$, the misspecified likelihood for the first module $p_{\theta_1}^{(n)}$ satisfies

$$\sup_{h\in K}\left|\log\frac{p^{(n)}_{\theta_1^*+\delta_{1,n}h}}{p^{(n)}_{\theta_1^*}}(Y^{(n)}) - h'V_{\theta_1^*}\Delta_{\theta_1^*,n} + \frac{1}{2}h'V_{\theta_1^*}h\right| \to 0, \quad (n\to\infty)$$

in $P_0^{(n)}$-probability and, for every sequence of constants $M_n \to \infty$, it holds that

$$P_0^{(n)}\Pi_1^{(n)}\left(\|\theta_1-\theta_1^*\| > \delta_{1,n}M_n\right) \to 0, \quad (n\to\infty).$$

Assumption 4.4.5. (Second module) There exist: a map $\theta_2^* : \Theta_1 \mapsto \Theta_2 \subset \mathbb{R}^{d_2}$; a norming rate $\delta_{2,n}$; a sequence of balls $K_n \subset \mathbb{R}^{d_2}$ containing 0, with radius $W_n \to \infty$, such that for any small $\epsilon > 0$

$$\sup_{u_1\in B(0,\epsilon)} \Pi_2^{(n)}\left(\theta_2 - \theta_2^*(\delta_{1,n}u_1+\theta_1^*) \notin \delta_{2,n}K_n)|Y^{(n)}, \delta_{1,n}u_1+\theta_1^*\right) \to 0 \tag{4.3}$$

as $n \to \infty$, in $P_0^{(n)}$ probability. The pseudo-prior density π_2 is assumed positive and continuous at $\theta_2^*(\theta_1^*)$. Moreover, there exist: a sequence of random vectors $\Delta_{n,\theta_2^*(\theta_1*)}$ (bounded in probability) and a non-singular matrix $V_{\theta_2^*(\theta_1^*)}$ such that, for any compact $K \subset \mathbb{R}^{d_2}$, the misspecified likelihood for the second module given θ_1 , $p_{\theta_2}^{(n)}(\cdot|\theta_1)$, satisfies

$$\begin{aligned}\sup_{u_1\in B(0,\epsilon)}\sup_{g\in K}&\left|\log\frac{p^{(n)}_{\theta_2^*(u_1\delta_{1,n}+\theta_1^*)+\delta_{2,n}g}(Y^{(n)}|u_1\delta_{1,n}+\theta_1^*)}{p^{(n)}_{\theta_2^*(u_1\delta_{1,n}+\theta_1^*)}(Y^{(n)}|u_1\delta_{1,n}+\theta_1^*)}\right.\\ &\left.-g'V_{\theta_2^*(\theta_1^*)}\Delta_{\theta_2^*(\theta_1^*),n} - \frac{1}{2}g'V_{\theta_2^*(\theta_1^*)}g\right|\\ &= c\epsilon(1+o_p(1)),\end{aligned} \tag{4.4}$$

for a positive constant c as $n \to \infty$.

Proposition 4.4.2. Let $\pi_1^{(n)}$ and $\pi_{2,n}^{(n)}$ denote the modular pseudo-posterior distributions of $(\theta_1 - \theta_1^*)/\delta_{1,n}$ (conditionally on $Y^{(n)}$) and $(\theta_2 - \theta_2^*(\theta_1))/\delta_n$ (conditionally on $Y^{(n)}$ and θ_1). Let $\pi^{(n)}(u_1, u_2) = \pi_1^{(n)}(u_1)\pi_{2,u_1\delta_n+\theta_1^*}^{(n)}(u_2)$. Then, under Assumptions 4.4.4-4.4.5 it holds that

$$\|\pi^{(n)} - \phi_1(\cdot|\Delta_{\theta_1^*,n}, V_1^{-1})\phi_2(\cdot|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})\|_1 \to 0, \quad (n \to \infty) \tag{4.5}$$

in $P_0^{(n)}$-probability.

Proof. Analogously to the previous section, we have that for an arbitrarily small $\epsilon > 0$

$$\begin{aligned}&\|\pi^{(n)} - \phi_1(\cdot|\Delta_{\theta_1^*,n}, V_1^{-1})\phi_2(\cdot|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})\|_1\\ &\quad\le \int_{B(0,\epsilon)} \|\pi_{2,\delta_n u_1+\theta_1^*}^{(n)}(u_2) - \phi_2(u_2|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})\|_1 \pi_1^{(n)}(u_1)du_1 + o_p(1).\end{aligned}$$

We show once more that the integrand in the above display converges uniformly to 0 in probability. To do it, we exploit several arguments from (Kleijn and van der Vaart, 2012, pp. 358-360), thus we outline only the salient steps. We use the following short notations: $\phi_{2,n}(u_2) \equiv \phi_2(u_2|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})$; $\pi_n(u)$ denotes the pseudo-prior density of $(\theta_2 - \theta_2^*(\delta_{1,n}u_1 + \theta_1^*))/\delta_{2,n}$ at u_2; we set

$$s_n(u) := \log \frac{p^{(n)}_{\theta_2^*(u_1\delta_{1,n}+\theta_1^*)+\delta_{2,n}g}(Y^{(n)}|u_1\delta_{1,n} + \theta_1^*)}{p^{(n)}_{\theta_2^*(u_1\delta_{1,n}+\theta_1^*)}(Y^{(n)}|u_1\delta_{1,n} + \theta_1^*)}.$$

Under Assumption 4.4.5, the random function

$$f_n(g, h) = \sup_{u_1\in B(0,\epsilon)} \left|1 - \frac{\phi_{2,n}(h)s_n(u_1, g)\pi_n(u_1, g)}{\phi_{2,n}(g)s_n(u_1, h)\pi_n(u_1, h)}\right|$$

defined on the compact set $K \times K$, for a compact $K \subset \mathbb{R}^{d_2}$, satisfies

$$\sup_{h,g\in K\times K} f_n(h, g) \le C\epsilon(1 + o_p(1))$$

as $n \to \infty$, for some positive constant C. Denote by P^K the restriction on K of a generic probability measure P on $\mathbb{R}^{d_2}$. Let Ξ_n denote the event that $\inf_{u_1\in B(0,\epsilon)} \Pi^{(n)}_{2,u_1\delta_{1,n}+\theta_1^*}(K) > 0$. Let $\Omega_{\epsilon,n}$ define the inner measurable cover set of the event $\sup_{h,g\in K\times K} f_n(h, g) \le C\epsilon$. Then, assuming that K contains a neighborhood of 0, for any $u_1 \in B(0, \epsilon)$

$$\begin{aligned}&P_0^{(n)}\|\pi^{(n)K}_{2,\delta_{1,n}u_1+\theta_1^*} - \phi^K_{2,n}\|_1 \mathbb{1}_{\Xi_n}\\ &\quad\le P_0^{(n)}\|\pi^{(n)K}_{2,\delta_{1,n}u_1+\theta_1^*} - \phi^K_{2,n}\|_1 \mathbb{1}_{\xi_n\cap\Omega_{\epsilon,n}} + 2P_0^{(n)}(\Xi_n \setminus \Omega_{\epsilon,n}),\end{aligned}$$

where the second summand on the r.h.s. converges to zero. As for the first summand, it holds that

$$\begin{aligned}&\frac{1}{2}P_0^{(n)}\|\pi^{(n)K}_{2,\delta_{1,n}u_1+\theta_1^*}-\phi^K_{2,n}\|_1\mathbb{1}_{\xi_n\cap\Omega_{\epsilon,n}}\\&\qquad\le P_0^{(n)}\int\int\sup_{h,g\in K\times K}f_n(h,g)\mathbb{1}_{\xi_n\cap\Omega_{\epsilon,n}}\phi^K_{2,n}(g)dg\pi^{(n)K}_{2,\delta_{1,n}u_1+\theta_1^*}(h)dh\\&\qquad\le C\epsilon.\end{aligned}$$

Thus, $\sup_{u_1\in B(0,\epsilon)}P_0^{(n)}\|\pi^{(n)K}_{2,\delta_{1,n}u_1+\theta_1^*}-\phi^K_{2,n}\|_1\mathbb{1}_{\Xi_n}$ can be made arbitrarily small. Finally, observe that for any $u_1\in B(0,\epsilon)$

$$\begin{aligned}&\|\pi^{(n)}_{2,\delta_{1,n}u_1+\theta_1^*}-\phi_{2,n}\|_1\\&\qquad\le 2\Pi^{(n)}_{2,\delta_{1,n}u_1+\theta_1^*}(K^c)+2\Phi_{2,n}(K^c)+\|\pi^{(n)K}_{2,\delta_{1,n}u_1+\theta_1^*}-\phi^K_{2,n}\|_1,\end{aligned}\tag{4.6}$$

where the third term can be made arbitrarily small, uniformly with respect to u_1. The result now follows from (4.3), by replacing K with K_n. □

Arguably, condition (4.4) can be deduced from the continuity of the map $\theta_1\mapsto\theta_2^*(\theta_1)$ and the "regularity" of the misspecified model $p^{(n)}_{2,\theta_2}(\cdot|\theta_1)$. Therefore, it does not appear to be too restrictive. A more intriguing problem consists in finding sufficient conditions to guarantee the "uniform" contraction condition in (4.3).

Since we expect the pseudo-posterior of module 1 to concentrate at a pseudo-true value θ_1^*, it seems reasonable that, asymptotically, the pseudo-posterior for the second module inherits θ_1^* only from the first one, θ_2 becoming independent from θ_1. Though, modular procedures might have very different natures and it would be interesting to find counterexamples (if any), in which dependence is preserved also asymptotically.

Example

Consider the basic problem:

$$y_i=f_0(x_i)+\varepsilon_i,\quad \varepsilon_i\overset{iid}{\sim}N(0,1),$$

where $y^{(n)}=(y_1,\ldots,y_n)'$, $x_i=i/n$, $i=1,\ldots,n$, and n is a power of 2. We assume f_0 is the true smooth unknown function. As we did previously, we implement `MONOMER` in 2 modules proceeding in dyadic splits. After assigning zero-mean g-priors, we get:

$$\theta_1|y^{(n)}\sim N\left(\frac{n}{n+1}\bar{Y},\frac{1}{n+1}\right),\quad \theta_2|\theta_1,y^{(n)}\sim N_2\left(\frac{n}{n+1}\hat{\beta}_n,\frac{2}{n+1}I_2\right),$$

where $\hat{\beta}_n = (\bar{Y}_1 - \theta_1, \bar{Y}_2 - \theta_1)'$, $\bar{Y}_1 = 2/n \sum_{i \le n/2} y_i$ and $\bar{Y}_2 = 2/n \sum_{i \ge n/2} y_i$. We immediately obtain that the modular posterior distribution of θ_1 asymptotically concentrates at $\theta_1^* = \int_0^1 f_0$, with rate $\delta = 1/\sqrt{n}$. Similarly, for a given θ_1, the modular posterior distribution of θ_1 asymptotically concentrates at

$$\theta_2^*(\theta_1) = \left(\int_0^{1/2} f_0 - \theta_1, \int_{1/2}^{1} f_0 - \theta_1 \right)',$$

again with rate $\delta_n = 1/\sqrt{n}$. It is also immediate to verify that the first module satisfies the conditions of Theorem 2.1 in Kleijn and van der Vaart (2012); the modular posterior density of the standardized random variable $(\theta_1 - \theta_1^*)/\delta_n$, denoted by $\pi_1^{(n)}$, satisfies

$$\|\pi_1^{(n)} - \phi_1(\cdot | \Delta_{\theta_1^*,n}, V_1^{-1})\|_1 \to 0$$

in $P_0^{(n)}$-probability as $n \to \infty$, with $P_0^{(n)}$ denoting the true probability measure associated to $y^{(n)}$. Herein, $\phi_1(\cdot|\mu, \sigma^2)$ is the density of a univariate normal distribution, $\Delta_{\theta_1^*,n} = \sqrt{n}(\bar{Y} - \theta_1^*)$ and $V_1 = 1$. As for the second module, it makes use of the misspecified likelihood $p_{\theta_2}(y^{(n)} - \theta_1 \mathbf{1}_n)$ satisfying, for any $h \in \mathbb{R}^2$,

$$\log \frac{p_{\theta_2^*(\theta_1) + \delta_n h}(y^{(n)} - \theta_1 \mathbf{1}_n)}{p_{\theta_2^*(\theta_1)}(y^{(n)} - \theta_1 \mathbf{1}_n)} = \Delta'_{\theta_2^*(\theta_1),n} V_2 h - \frac{1}{2} h' V_2 h,$$

where

$$\Delta_{\theta_2^*(\theta_1),n} = \sqrt{n}(\bar{Y}_1 - \theta_1 - \theta_{2,1}^*(\theta_1), \bar{Y}_1 - \theta_1 - \theta_{2,1}^*(\theta_1))', \quad V_2 = \frac{1}{2} I_2.$$

Consequently, condition (2.8) in Kleijn and van der Vaart (2012) is satisfied and we can resort once more to their Theorem 2.1 to conclude that for any given θ_1,

$$\|\pi_{2,\theta_1}^{(n)} - \phi_2(\cdot | \Delta_{\theta_2^*(\theta_1),n}, V_2^{-1})\|_1 \to 0 \tag{4.7}$$

in $P_0^{(n)}$-probability as $n \to \infty$, where $\pi_{2,\theta_1}^{(n)}$ is the modular posterior density of $(\theta_2 - \theta_2^*(\theta_1))/\delta_n$ (conditionally on $y^{(n)}$ and θ_1). In fact, for this very specific modular setup, we have that

$$\Delta_{\theta_2^*(\theta_1),n} = \Delta_{\theta_2^*(\theta_1^*),n}, \quad \forall \theta_1 \in \mathbb{R}, \tag{4.8}$$

therefore the Gaussian density in (4.7) does not depend on θ_1. This implies that the parameters θ_1 and θ_2 are asymptotically independent a posteriori. Formally, the following result can be established.

Proposition 4.4.3. Let $\pi^{(n)}(u_1, u_2) = \pi_1^{(n)}(u_1) \pi_{2, u_1 \delta_n + \theta_1^*}^{(n)}(u_2)$, then as $n \to \infty$

$$\|\pi^{(n)} - \phi_1(\cdot | \Delta_{\theta_1^*,n}, V_1^{-1}) \phi_2(\cdot | \Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})\|_1 \to 0, \tag{4.9}$$

in $P_0^{(n)}$-probability.

Proof. Using the triangular inequality and the concentration of the modular posterior of θ_1 at θ_1^*, we obtain

$$\begin{aligned}
&\|\pi^{(n)} - \phi_1(\cdot|\Delta_{\theta_1^*,n}, V_1^{-1})\phi_2(\cdot|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})\|_1 \\
&\quad\leq \int_{-\epsilon}^{\epsilon} \|\pi^{(n)}_{2,\delta_n u_1+\theta_1^*}(u_2) - \phi_2(u_2|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})\|_1 \pi_1^{(n)}(u_1)du_1 + o_p(1).
\end{aligned}$$

Next, observe that for any $u_1 \in (-\epsilon, \epsilon)$

$$\begin{aligned}
&\|\pi^{(n)}_{2,\delta_n u_1+\theta_1^*}(u_2) - \phi_2(u_2|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})\|_1 \\
&\quad\leq \frac{1}{n+1} + \int_{(-\infty,+\infty)^2} \left|e^{-D_n(u)} - 1\right| \phi_2(u_2|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})du_2,
\end{aligned} \tag{4.10}$$

where $u = (u_1, u_2)$, $\hat{\beta}_n^* = (\bar{Y}_1 - \theta_1^*, \bar{Y}_2 - \theta_1^*)'$ and

$$\begin{aligned}
D_n(u) = &\frac{1}{4}\left(u_2 - \Delta_{\theta_2^*(\theta_1^*),n}\right)'\left(u_2 - \Delta_{\theta_2^*(\theta_1^*),n}\right) \\
&- \frac{n+1}{4}\left(\delta_n u_2 + \theta_2^*(\delta_n u_1 + \theta_1^*) - \frac{n}{n+1}\hat{\beta}_n\right)' \cdot \\
&\left(\delta_n u_2 + \theta_2^*(\delta_n u_1 + \theta_1^*) - \frac{n}{n+1}\hat{\beta}_n\right) \\
= &-\frac{1}{2}\left(u_2 - \Delta_{\theta_2^*(\theta_1^*),n}\right)'\left(\hat{\beta}_n^* + \delta_n u_1 \mathbf{1}_2\right)\left(\frac{\sqrt{n}}{1+n} + \frac{\sqrt{n}}{n(1+n)}\right) \\
&+ \left(\hat{\beta}_n^* + \delta_n u_1 \mathbf{1}_2\right)'\left(\hat{\beta}_n^* + \delta_n u_1 \mathbf{1}_2\right)\frac{1}{n+1} \\
&+ \left(u_2 - \Delta_{\theta_2^*(\theta_1^*),n}\right)'\left(u_2 - \Delta_{\theta_2^*(\theta_1^*),n}\right)\frac{1}{4n}.
\end{aligned}$$

In particular, $|D_n(u)| \leq O_p(1/\sqrt{n})$ uniformly in u_1 (the upper bound depending on ϵ and u_2 only). Thus, a Taylor expansion at 0 of the exponential in (4.10) allows to conclude that

$$\sup_{u_1 \in (-\epsilon,+\epsilon)} \|\pi^{(n)}_{2,\delta_n u_1+\theta_1^*}(u_2) - \phi_2(u_2|\Delta_{\theta_2^*(\theta_1^*),n}, V_2^{-1})\|_1 \to 0$$

as $n \to \infty$ and the result in (4.9) follows. □

Model	MSPE	MAPE
BART	1.036	1.850
MONOMER	1.083	2.232
GAM	1.467	1.970
MARS	1.823	2.764
Random Forest	2.035	3.316
CART	2.844	4.025
Lasso	2.968	5.088
Linear Reg.	4.090	6.951
Bayes. CART	5.486	6.578

Table 4.1: Mean squared (MSPE) or absolute percentage (MAPE) prediction error in the out-of-sample prediction on $n_{\text{out}} = 50$, averaged over 100 datasets generated using the setup of Section 4.5.2, and a sample size of $n = 200$.

4.5 Applications

4.5.1 Univariate regression

We simulate 500 observations from the model $y_i = f(x_i) + .2e^{2x_i} Z_i$, where $f(\cdot)$ is the *Blocks* function of Donoho and Johnstone (1994, 1995), $x_i \overset{iid}{\sim} U[0,1]$ and $Z_i \overset{iid}{\sim} N(0,1)$. $\sigma(x) = .2e^{2x_i}$ is the standard deviation of the errors also used in Pratola et al. (2017). We run MCMC of Algorithm 3 for 2000 iterations after 1000 of burn-in. Figure 4.2 shows the output of our model in this case. `MONOMER` allows to not only estimate the regression function and the covariate-dependent uncertainty due to the measurement error (bottom left of Figure 4.2), but also to sequentially reconstruct the regression function bottom-up by partial sums of resolution contributions, quantifying their uncertainty.

4.5.2 Multivariate regression

We simulate 250 observations from the model $y_i = h_1(x_{1,i}) + \cdots + h_4(x_{4,i}) + \varepsilon_i$, where $\varepsilon_i\ N(0, \sigma^2)$ with $\sigma^2 = 0.1$, and $x_j \overset{iid}{\sim} U[0,1]$ for $j = 1, \ldots, 4$. We set $h_1(x) = 4x_1^2$, while h_2, h_3, h_4 correspond to *Blocks*, *Doppler*, *HeaviSine* from Donoho and Johnstone (1994, 1995). The goal is to estimate $f(\cdot)$ along with its additive components, and provide their multiresolution decomposition. We show the results of `MONOMER` in Figure 4.3. In terms of out-of-sample predictive performance on this data, `MONOMER` compares favorably with well-established models, as seen in Table 4.1.

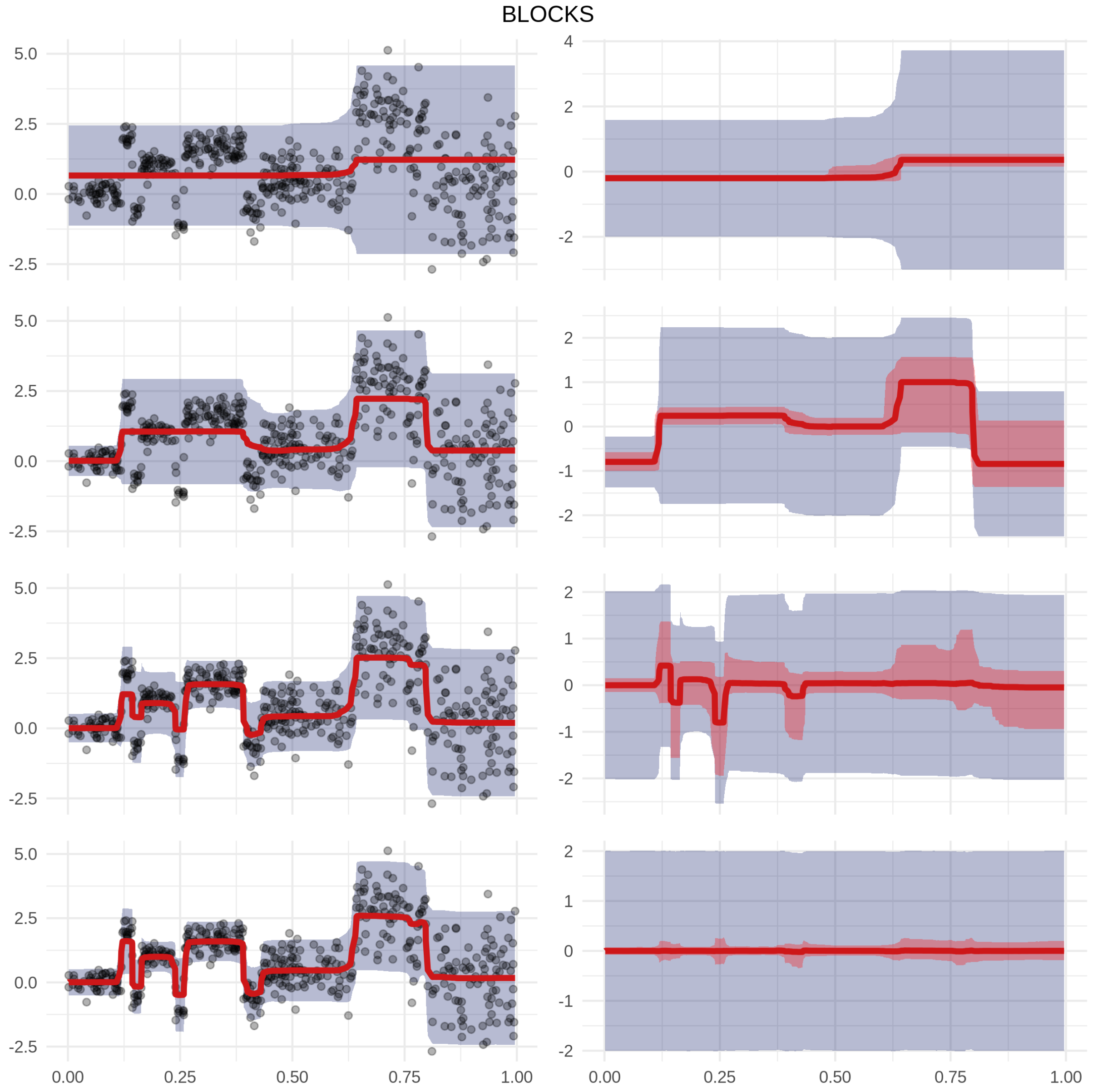


Figure 4.2: MONOMER output for the simulated data application of Section 4.5.1. On the right and from the top, the output data y_i (dots) is overlayed by the sequential estimates of $E(Y|x)$ at increasing resolutions. On the left, we decompose the regression function and show the contributions of increasingly higher resolution. The red shaded areas correspond to uncertainty in the regression function, the blue shaded correspond to the multiplicative contribution of $\sigma^{2(r)}$. All the contributions are estimated simultaneously in a single MCMC run.

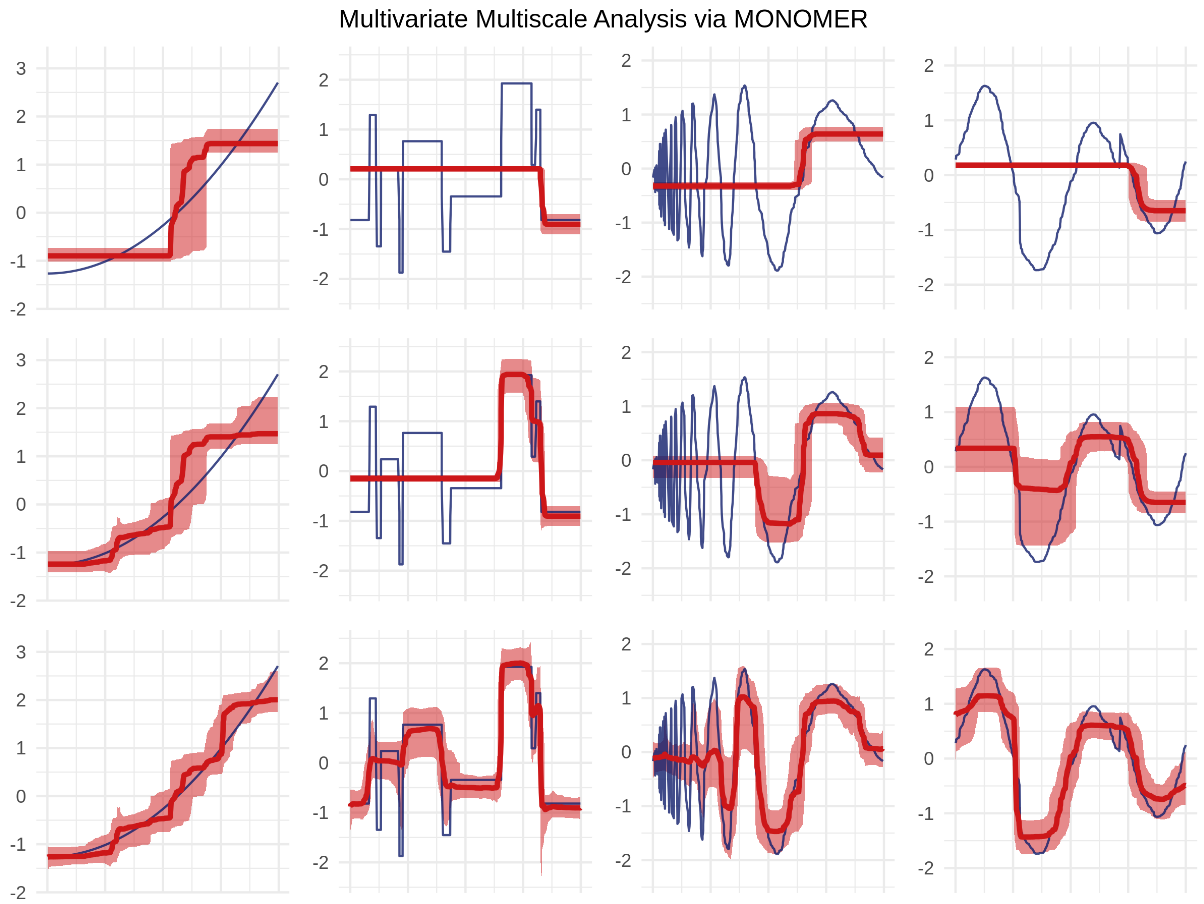


Figure 4.3: MONOMER output for the simulated data application of Section 4.5.2. In this plot matrix, row r, column j is the model-averaged estimate of $\sum_{l<r} g_{j,r}(x_j)$, i.e. the estimate of h_j obtained by summing all resolution contributions up to r. We consider here decomposing each function in at most 10 resolution contributions, and we show $r = 1, 3, 10$. Uncertainty in the estimation of the partial regression function at each resolution is displayed as a red shaded area that corresponds to 95% credible bands.

Model	MSPE	MAPE
BART	13.29	12.32%
Random Forest	14.58	12.53%
MONOMER	17.17	13.71%
MARS	17.81	15.14%
Neural Net	20.37	16.05
GAM	22.11	15.46%
CART	25.63	17.40
Lasso	32.29	20.08%
Linear Reg.	84.14	36.04%

Table 4.2: Mean squared (MSPE) or absolute percentage (MAPE) prediction error in the out-of-sample prediction of *MPG*, with $n_{\text{out}} = 306$, averaged over 100 simulations using a sample size of $n = 200$ randomly sampled observations.

4.5.3 The Boston Housing dataset

We consider the Boston Housing data, which has been used extensively throughout the literature to benchmark numerous algorithms. This standard dataset reports median house prices (*medv*) in the Boston, MA area, along with other covariates. We fit models to predict *medv* based on 10 continuous covariates, namely *lstat*, *b*, *crim*, *indus*, *nox*, *rm*, *age*, *dis*, *tax*, and *ptratio*. We compare MONOMER with state-of-the-art methods for regression by testing the models' predictive performance and show the results of this comparison on Table 4.2. On this task, the loss in predictive performance by MONOMER compared to the best alternative is 1.5%. We may attribute this loss to the fact that by additively separating the predictors' effects on the response, we are not considering the potential interaction effect. However, this choice is partly responsible for the resulting easy interpretability of the estimates. We show the multiscale decomposition of the predictors' effect on *medv* in Figure 4.4. The low resolution components identify the areas in the predictor spaces most related to large changes in the outcome. The high resolution components, instead,

4.6 Discussion

In this Chapter, we proposed a novel Bayesian semiparametric method for multivariate regression which decomposes the regression function into regressor and resolution components, and can be used in standard regression settings to model the complex relationship between predictors and outcome. By implementing Bayesian modularization techniques, our approach can produce easily communicable "stylized facts" about major and minor features in the relationship between inputs and output. We showed in a real-world data application that it also enjoys comparable performance to state-of-the-art methods for

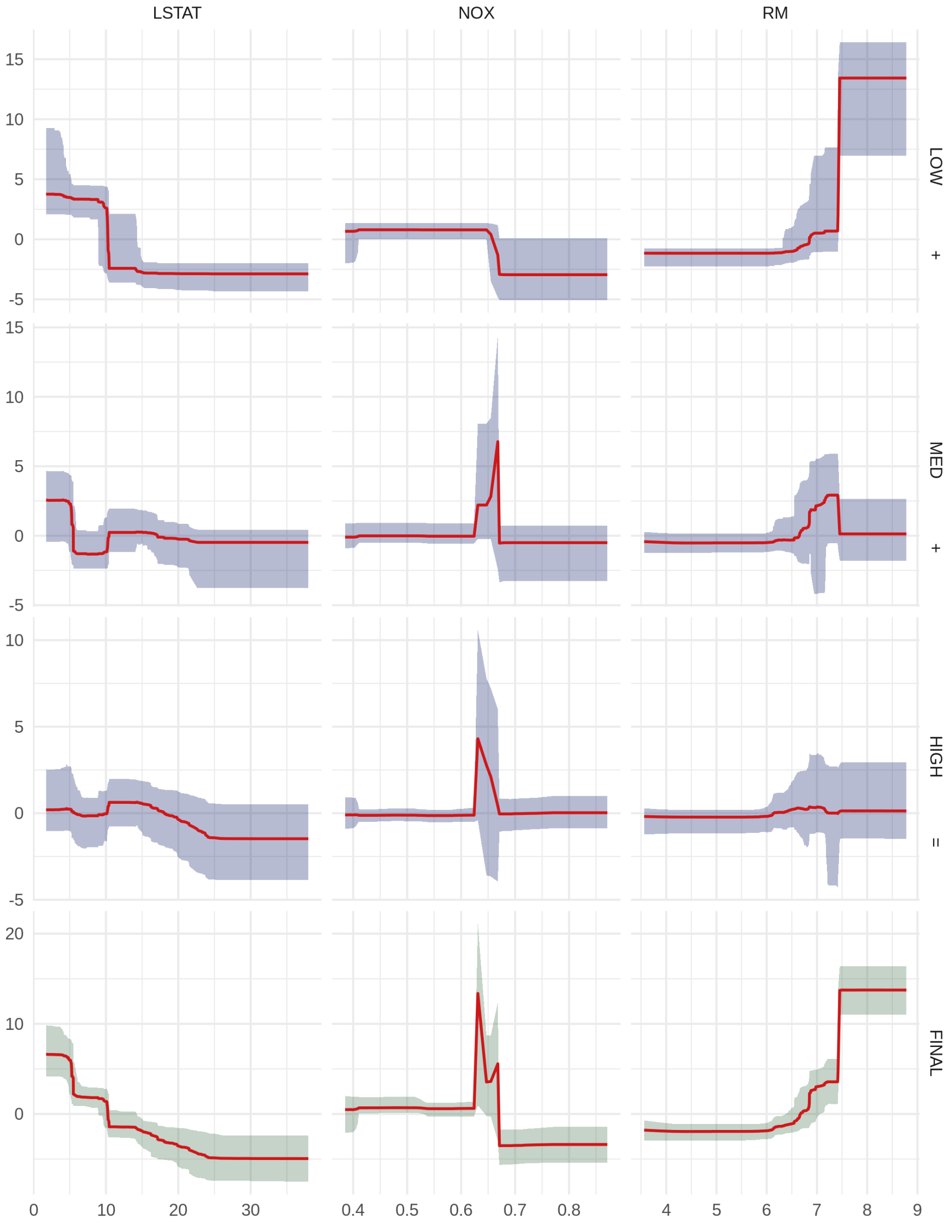


Figure 4.4: `MONOMER` estimates of the net effects of three regressors on *medv*, the median value of homes in a census tract, at low, medium, and high resolution. Row r, column j of the 3x3 matrix of made of dashed-line plots thus corresponds to $g_{j,r}$ in the `MONOMER` decomposition of (4.1). The final row corresponds to $h_j = \sum_{r=1}^{R} g_{j,r}(x_j)$, i.e. the total effect of regressor j on the response. All the plots were obtained from a single run of MCMC as averages over 3000 iterations. The top row of low-resolution contributions highlights the regions in the covariates' spaces linked to the largest differences in houses' median value. In particular, census tracts with a large average number of rooms per dwelling (*rm*$>$ 7.5), and a lower poverty rate (*lstat*$<$ 10%) are linked to higher home value.

multivariate regression. However, these advantages come at a cost, and there are cases in which we expect our approach to be lacking. First, estimating the full extent of the covariate contributions h_j becomes increasingly complex in finite samples as the number of covariates d grows; it would be desirable to only focus on a subset of variables. Second, our assumption of additive and separable effects of the covariates on the output is unable to fully capture the extent of possible interactions among inputs; therefore, we expect our model to perform comparatively worse if interactions play a major role in regression. Third, a multiscale decomposition via step-functions might be inefficient if the relationship between inputs and outputs is best approximated by a smooth function. These points suggest avenues for future research.

4.7 Appendix

4.7.1 Linking `MONOMER` to *BM&Ms*

We can find a simple connection between the models of Chapters 2 and 4. Suppose we target the estimation of $f(\cdot)$ in the following model

$$y_i = f(x_i) + \varepsilon_i$$

where errors are $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, and $x \in \mathbb{R}$. The function $f(\cdot)$ is assumed as piecewise constant, hence the cells partitioning the predictor space are nothing but intervals in one dimension. Therefore, for a given partition the only important information in the inputs is their ordering. This fact may be limiting in some cases, but removes the need to consider transforming the data in the pre-processing phase. For this reason, we assume without loss of generality that the data are sorted by x, i.e. if $i_1 < i_2$ then $x_{i_1} \leq x_{i_2}$. Then, the regression relationship can be expressed as a linear model $y = \mathcal{L}_j \theta_j + \varepsilon$, where y is now the vector collecting all observations sorted by x, $\varepsilon \sim N(0, \sigma^2 I_n)$, θ_j are the heights of the steps, and $\mathcal{L}_j$ is a $n \times p_j$ matrix of zeros and ones representing the cell memberships of the sorted observations when $f(\cdot)$ has $p_j - 1$ jumps.

For illustrative purposes, a step function with a single jump at a value x_1 out of a sample of size $n = 4$ would correspond to:

$$\mathcal{L}_j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Then, a univariate `MONOMER` decomposing $f(\cdot)$ in resolution components, with fixed jumps, can be written as:

$$y = \mathcal{L}_0 \theta_0 + \cdots + \mathcal{L}_K \theta_K + \varepsilon.$$

Hence `MONOMER` in one dimension can be represented as *BM&Ms* using $X_K = I_n$; the coarsening operators $\mathcal{L}_j$ are defined the same way (see e.g. Section 3.2). A special case arises if $n = 2^q$ for some q and we deterministically choose $\mathcal{L}_j = I_{2^j} \otimes \mathbf{1}_{2^{q-j}}$. The resulting multiresolution structure successively splits the observed x in statistically equivalent blocks of halved sizes; decomposing $f(\cdot)$ in low-to-high resolution step-functions is in principle analogous to implementing the discrete Haar wavelet transform (see the discussion in Chapter 1).

This simplicity in the setup does not follow in the multivariate case, as regressors imply ordering observations differently. However, Algorithm 3 is implemented via Gibbs steps that only involve univariate models, and hence ordering and reordering can be done locally at each Gibbs step, making the multivariate case an easy algorithmic extension of the univariate one.

Example

Suppose we consider a univariate `MONOMER` and decompose $f(\cdot)$ into K resolutions, using a simple dyadic split of the data. Each module $k \leq K$ will use data X_k. The setup is thus the following:

$$\begin{aligned} &X = I_n & & \\ &\mathbf{1}_k := \begin{bmatrix} 1 \cdots 1 \end{bmatrix}'_{1\times k} & & n_k = 2^k \\ & & & L_k := I_{n_k} \otimes \mathbf{1}_{n/n_k} \\ &\mathbf{0}_k := \begin{bmatrix} 0 \cdots 0 \end{bmatrix}'_{1\times k} & & X_k := XL_k \\ &n = 2^K & & \end{aligned}$$

We take σ^2 as known. We then define $X_j = XL_j$ and use $\pi(\theta_j) = N(m_j, \sigma^2 V_j)$, where for simplicity $V_j = n(X_j'X_j)^{-1}$. This results in $\pi_M(\theta_j \mid y) = N(\mu_j, \sigma^2\Sigma_j)$. For illustration, consider just the first two modules – finer-scale ones can be obtained by induction.

Module 1:

$$\begin{aligned} X_1 &= XL_1 = I_n I_1 \otimes \mathbf{1}_n = \mathbf{1}_n \\ V_1 &= n(X_1'X_1)^{-1} = n(\mathbf{1}_n'\mathbf{1}_n)^{-1} = 1 \\ \mu_1 &= \Sigma_1 X_1' y = \frac{1}{n+1} 1_n' y = \frac{1}{n+1} \sum_{i=1}^{n} y = \frac{n}{n+1} \bar{y} \end{aligned}$$

Module 2:

$$\begin{aligned} X_2 &= XL_2 = I_n I_2 \otimes \mathbf{1}_2 = I_n \begin{bmatrix} \mathbf{1}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{0}_{n/2} & \mathbf{1}_{n/2} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{0}_{n/2} & \mathbf{1}_{n/2} \end{bmatrix} \\ V_2 &= n(X_2'X_2)^{-1} = n \left(\begin{bmatrix} \mathbf{1}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{0}_{n/2} & \mathbf{1}_{n/2} \end{bmatrix}' \begin{bmatrix} \mathbf{1}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{0}_{n/2} & \mathbf{1}_{n/2} \end{bmatrix} \right)^{-1} \end{aligned}$$

$$
\begin{aligned}
&= n \begin{bmatrix} \frac{n}{2} & 0 \\ 0 & \frac{n}{2} \end{bmatrix}^{-1} = 2I_2 \\
\Sigma_2 &= (V_2^{-1} + X_2'X_2)^{-1} = \frac{2}{n+1} I_2 \\
\mu_2 &= \Sigma_2 X_2' e_2 = \frac{2}{n+1} \begin{bmatrix} \mathbf{1}_{n/2} & \mathbf{0}_{n/2} \\ \mathbf{0}_{n/2} & \mathbf{1}_{n/2} \end{bmatrix}' e_2 = \begin{bmatrix} \frac{2}{n+1} \sum_{i=1}^{\frac{n}{2}} \left(y_i - \frac{n}{n+1}\bar{y}\right) \\ \frac{2}{n+1} \sum_{i=\frac{n}{2}+1}^{n} \left(y_i - \frac{n}{n+1}\bar{y}\right) \end{bmatrix} \\
&= \begin{bmatrix} \frac{2}{n+1} \sum_{i=1}^{\frac{n}{2}} y_i - \frac{n^2}{(n+1)^2}\bar{y} \\ \frac{2}{n+1} \sum_{i=\frac{n}{2}+1}^{n} y_i - \frac{n^2}{(n+1)^2}\bar{y} \end{bmatrix}
\end{aligned}
$$

and note that for n large

$$
\mu_1 = \tfrac{n}{n+1}\bar{y} \approx \bar{y} \qquad \mu_2 = \begin{bmatrix} \frac{2}{n+1} \sum_{i=1}^{\frac{n}{2}} y_i - \frac{n^2}{(n+1)^2}\bar{y} \\ \frac{2}{n+1} \sum_{i=\frac{n}{2}+1}^{n} y_i - \frac{n^2}{(n+1)^2}\bar{y} \end{bmatrix} \approx \begin{bmatrix} \bar{y}_{[i \leq \frac{n}{2}]} - \bar{y} \\ \bar{y}_{[i > \frac{n}{2}]} - \bar{y} \end{bmatrix}
$$

Each module k increases the detail of module $k-1$ by calculating increasingly smaller group means of the output and their difference with corresponding means from the previous module.

4.7.2 Output from some implemented models

We report a visual depiction of the output from Random Forests (Breiman, 2001), MARS (Friedman, 1991), and CART (Breiman et al., 1984) on the analysis of the *MPG* data of Section 4.5.3. In all cases, marginal effects are displayed by keeping other variables fixed at their median values, and this fact highlights how the ability to efficiently consider interactions comes at a cost for interpretability.

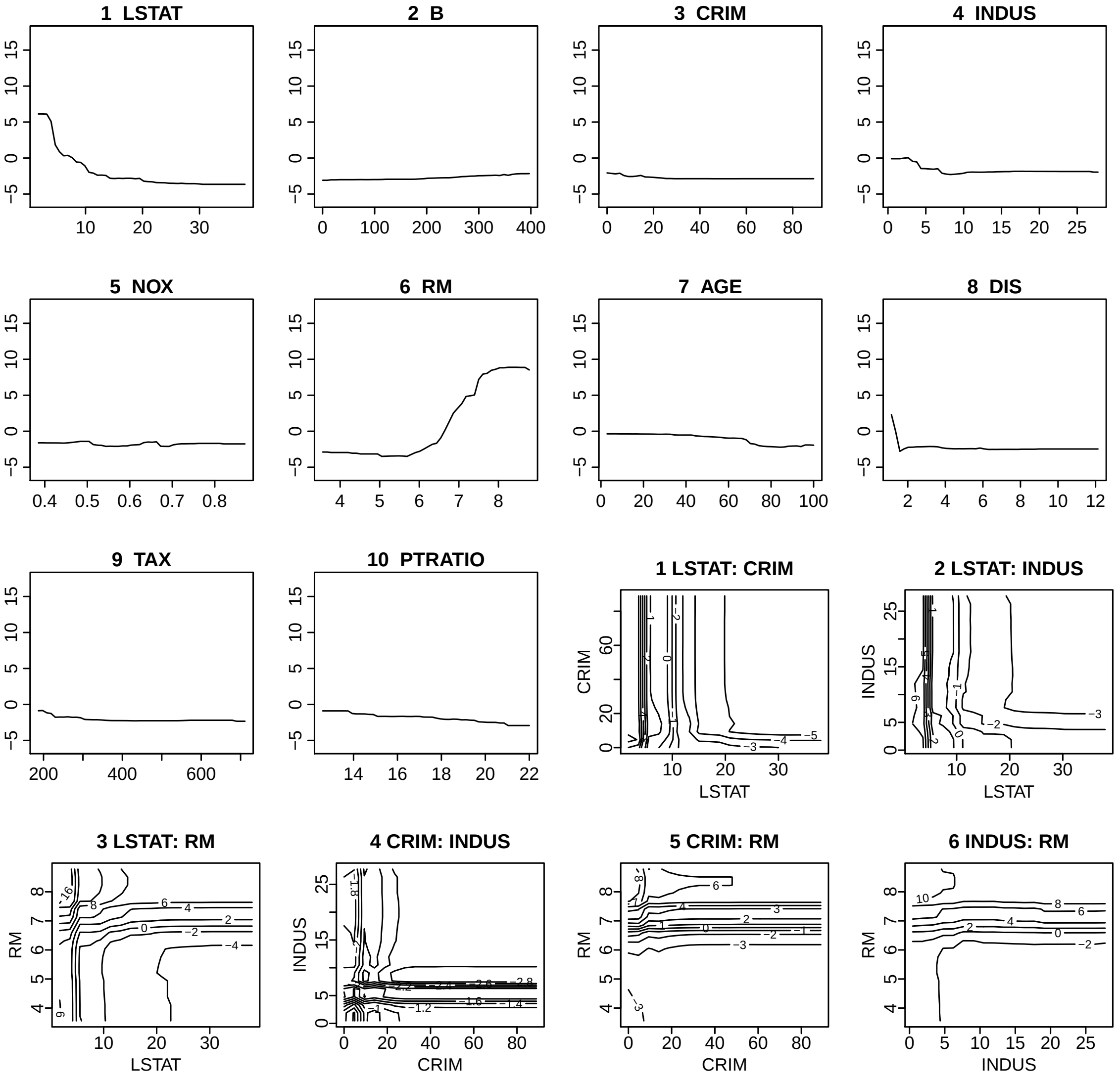


Figure 4.5: Output from Random Forests (Breiman, 2001) on the analysis of the *Boston Housing* data.

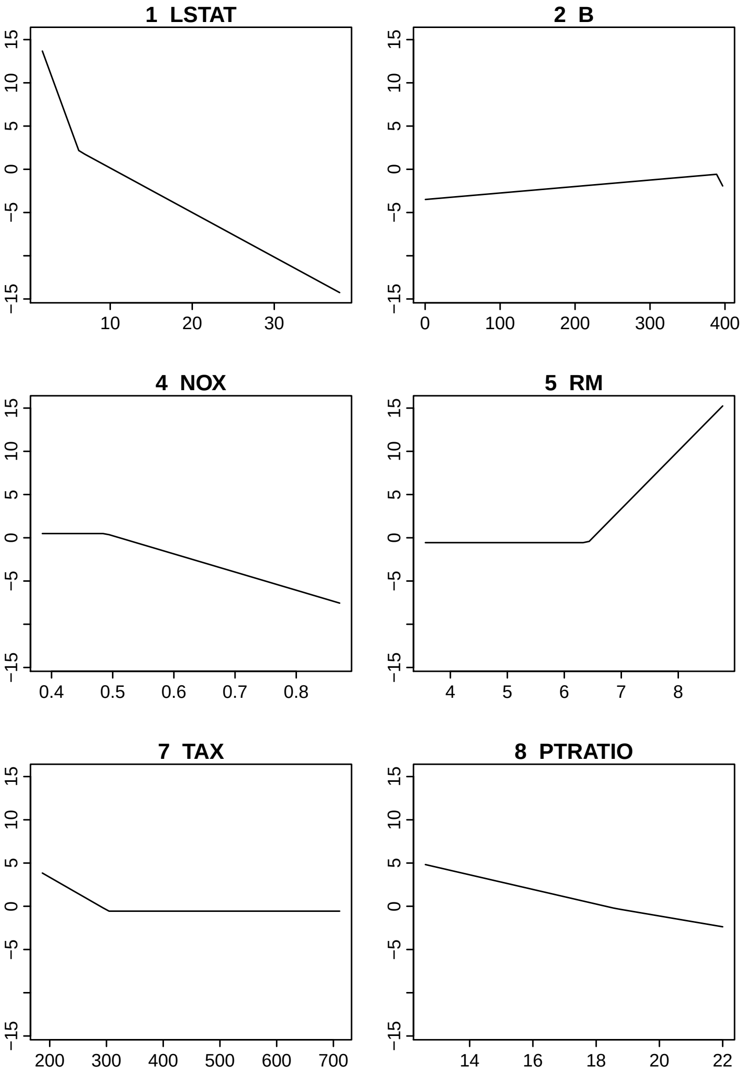


Figure 4.6: Output from MARS (Friedman, 1991) on the analysis of *Boston Housing* data.

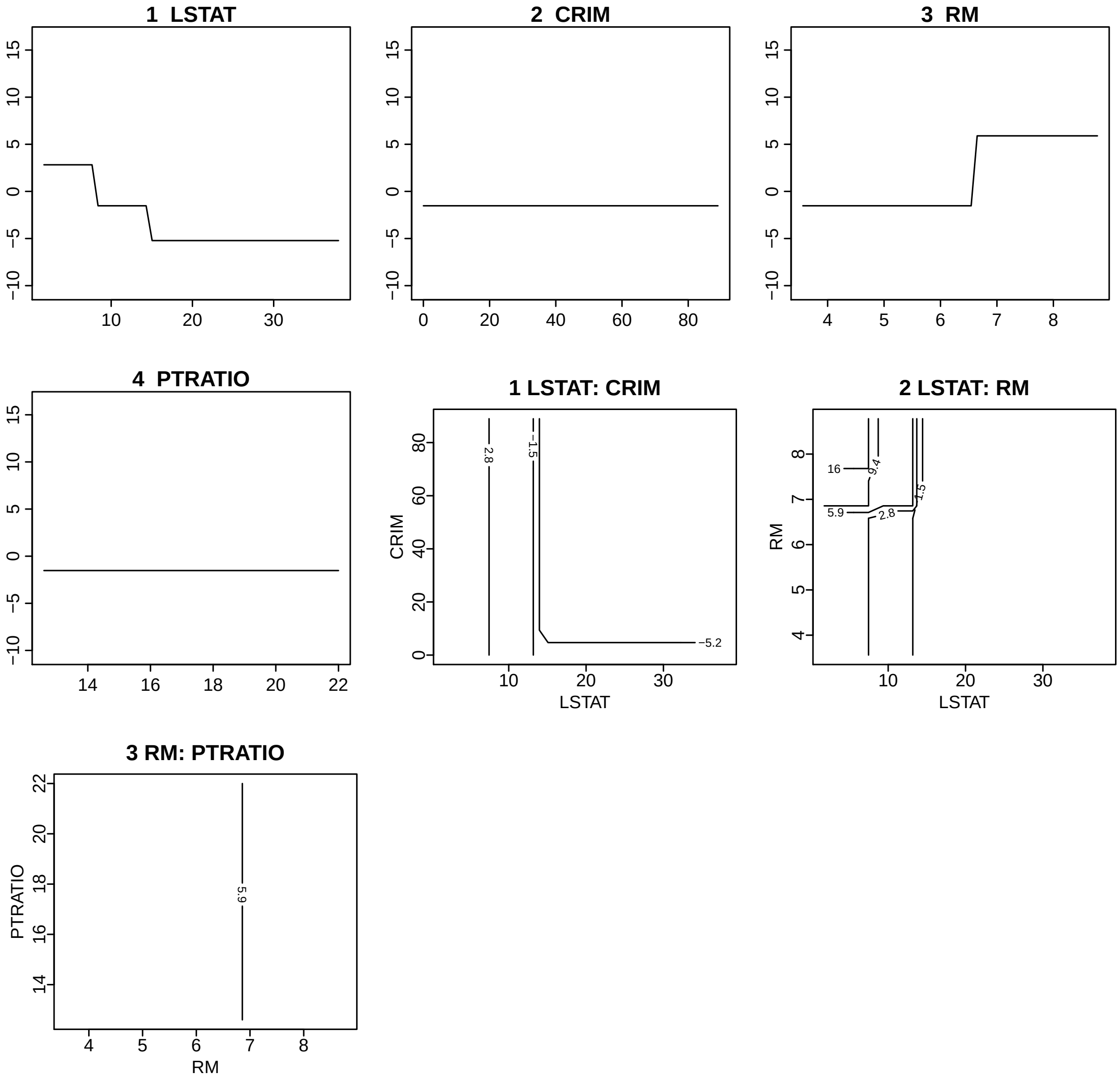


Figure 4.7: Output from CART (Breiman et al., 1984) on the analysis of the *Boston Housing* data.

4.7.3 Implemented models

- `MONOMER`: Modular Nonparametric Multiscale Ensembles Regression as in Chapter 4.
- *Linear regression*: Ordinary least squares
- *Lasso*: Lasso regression (Tibshirani, 1996) using cross-validation for λ (from R package `glmnet`)
- *BART*: Bayesian Additive Regression Trees (Chipman et al., 2010), implemented using R function `BART::wbart` and increasing the default chain length to 5000 iterations
- *CART*: Regression trees (Breiman et al., 1984), implemented using R package `rpart`
- *GAM*: Generalized Additive Models (Hastie and Tibshirani, 1990), implemented using R function `mgcv::gam` with spline bases on each covariate
- *MARS*: Multivariate Adaptive Regression Splines (Friedman, 1991), implemented using R package `earth`
- *Random Forests* (Breiman, 2001): from R package `randomForest`
- *Neural Net*: implemented using R package `nnet` with parameters `size`= 6 and `decay`= .1 found by cross-validation.

Bibliography

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679. doi:10.2307/2290350.

Anderson, T. W. (1966). *Some nonparametric multivariate procedures based on statistically equivalent blocks*, pages 5–27. Academic Press, New York.

Antoniadis, A. (2007). Wavelet methods in statistics: Some recent developments and their applications. *Statistics Surveys*, 1:16–55. doi:10.1214/07-SS014.

Auger, I. E. and Lawrence, C. E. (1989). Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1):39–54. doi:10.1007/BF02458835.

Bassett, D. S., Brown, J. A., Deshpande, V., Carlson, J. M., and Grafton, S. T. (2011). Conserved and variable architecture of human white matter connectivity. *NeuroImage*, 54(2):1262–1279. doi:10.1016/j.neuroimage.2010.09.006.

Bhattacharya, A. and Dunson, D. B. (2011). Sparse bayesian infinite factor models. *Biometrika*, 98(2):291–306. doi:10.1093/biomet/asr013.

Bigelow, J. L. and Dunson, D. B. (2012). Bayesian semiparametric joint models for functional predictors. *Journal of the American Statistical Association*, 104(485):26–36. doi:10.1198/jasa.2009.0001.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 26:123–140. doi:10.1007/BF00058655.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32. doi:10.1023/A:1010933404324.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and regression trees*. The Wadsworth statistics/probability series, Springer, New York.

Brown, P. J., Fearn, T., and Vannucci, M. (2001). Bayesian wavelet regression on curves with application to a spectroscopic calibration problem. *Journal of the American Statistical Association*, 96(454):398–408. doi:10.1198/016214501753168118.

Bühlmann, P., Rütimann, P., van de Geer, S., and Zhang, C.-H. (2013). Correlated variables in regression: clustering and sparse estimation. *Journal of Statistical Planning and Inference*, 143:1835–1858. doi:10.1016/j.jspi.2013.05.019.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data.* Springer, Dordrecht. doi:10.1007/978-3-642-20192-9.

Chatterjee, A. and Lahiri, S. N. (2011). Bootstrapping lasso estimators. *Journal of the American Statistical Association*, 106:608–625. doi:10.1198/jasa.2011.tm10159.

Chen, Y. and Dunson, D. B. (2017). Modular bayes screening for high-dimensional predictors. arXiv:1703.09906.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–960. doi:10.1080/01621459.1998.10473750.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *Annals of Applied Statistics*, 4(1):266–298. doi:10.1214/09-AOAS285.

Comte, F. and Johannes, J. (2012). Adaptive functional linear regression. *Annals of Statistics*, 40(6):2765–2797. doi:10.1214/12-AOS1050.

Crainiceanu, C., Ruppert, D., and Wand, M. (2005). Bayesian analysis for penalized spline regression using WinBUGS. *Journal of Statistical Software*, 14:1–22. doi:10.18637/jss.v014.i14.

Delaigle, A. and Hall, P. (2012). Methodology and theory for partial least squares applied to functional data. *Annals of Statistics*, 40(1):322–352. doi:10.1214/11-AOS958.

Dellaportas, P., Forster, J. J., and Ntzoufras, I. (2002). On Bayesian model and variable selection using MCMC. *Statistics and Computing*, 12:27–36. doi:10.1023/A:1013164120801.

Denison, D. G. T., Mallick, B. K., and Smith, A. F. M. (1998). Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60:333–350. doi:10.1111/1467-9868.00128.

Dennis, E. L., Jahanshad, N., Rudie, J. D., Brown, J. A., Johnson, K., McMahon, K. L., de Zubicaray, G. I., Montgomery, G., Martin, N. G., Wright, M. J., Bookheimer, S. Y., Dapretto, M., Toga, A. W., and Thompson, P. M. (2011). Altered structural brain connectivity in healthy carriers of the autism risk gene, CNTNAP2. *Brain Connectivity*, 1(6):447–459. doi:10.1089/brain.2011.0064.

Desikan, R. S., Segonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., Buckner, R. L., Dale, A. M., Maguire, R. P., Hyman, B. T., Albert, M. S., and Killiany, R. J. (2006). An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage*, 31(3):968–980. doi:10.1016/j.neuroimage.2006.01.021.

Donoho, D. L. and Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202. doi:10.1073/pnas.0437847100.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455. doi:10.1093/biomet/81.3.425.

Donoho, D. L. and Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90:1200–1224. doi:10.1080/01621459.1995.10476626.

Doshi-Velez, F. and Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. arXiv:1702.08608.

Essen, D. C. V., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., and Kamil Ugurbil, f. t. W.-M. H. C. (2013). The WU-Minn Human Connectome Project: An overview. *NeuroImage*, 80(2013):62–79. doi:10.1016/j.neuroimage.2013.05.041.

Ferreira, M. A. and Lee, H. K. (2007). *Multiscale Modeling: A Bayesian Perspective*. Springer Publishing Company, Incorporated, 1st edition.

Frank, I. E. and Friedman, J. H. (1993). A Statistical View of Some Chemometrics Regression Tools. *Technometrics*, 35:109–135. doi:10.2307/1269656.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139. doi:10.1006/jcss.1997.1504.

Frick, K., Munk, A., and Sieling, H. (2014). Multiscale change point inference. *Journal of the Royal Statistical Society Series B*, 76:495–580. doi:10.1111/rssb.12047.

Friedman, J. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, 19:1–67. doi:10.1214/aos/1176347963.

Friedman, J. H. (1977). A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers*, 26:404–408. doi:10.1109/TC.1977.1674849.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232. doi:10.1214/aos/1013203451.

Fryzlewicz, P. (2007). Unbalanced Haar technique for nonparametric function estimation. *Journal of the American Statistical Association*, 102(480):1318–1327. doi:10.1198/016214507000000860.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2014). *Bayesian Data Analysis, Third Edition*. Chapman and Hall/CRC, third edition.

Genovese, C. R., Lazar, N. A., and Nichols, T. (2002). Thresholding of statistical maps in functional neuroimaging using the false discovery rate. *NeuroImage*, 15:870–878. doi:10.1006/nimg.2001.1037.

George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, pages 339–374.

Gerard, D. and Stephens, M. (2018). Empirical Bayes shrinkage and false discovery rate estimation, allowing for unwanted variation. *Biostatistics*, page kxy029. doi:10.1093/biostatistics/kxy029.

Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., Essen, D. C. V., and Jenkinson, M. (2013). The minimal preprocessing pipelines for the human connectome project. *NeuroImage*, 80(2013):105–124. doi:10.1016/j.neuroimage.2013.04.127.

Goldsmith, J., Huang, L., and Crainiceanu, C. M. (2014). Smooth scalar-on-image regression via spatial Bayesian variable selection. *Journal of Computational and Graphical Statistics*, 23:46–64. doi:10.1080/10618600.2012.743437.

Gong, G., He, Y., and Evans, A. C. (2011). Brain connectivity: gender makes a difference. *The Neuroscientist*, 17:575–591. doi:10.1177/1073858410386492.

Gordon, E. M., Laumann, T. O., Adeyemo, B., Huckins, J. F., Kelley, W. M., and Petersen, S. E. (2016). Generation and evaluation of a cortical area parcellation from resting-state correlations. *Cerebral Cortex*, 26(1):288–303. doi:10.1093/cercor/bhu239.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Grollemund, P.-M., Abraham, C., Baragatti, M., , and Pudlo, P. (2018). Bayesian functional linear regression with sparse step functions. arXiv:1604.08403.

Guhaniyogi, R., Qamar, S., and Dunson, D. B. (2017). Bayesian tensor regression. *Journal of Machine Learning Research*, 18(79):1–31.

Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., and Giannotti, F. (2018). A survey of methods for explaining black box models. arXiv:1802.01933.

Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics. Springer, New York. doi:10.1007/b97848.

Hastie, T. and Tibshirani, R. (1990). *Generalized additive models*. Chapman & Hall, London, 1 edition. doi:10.1002/sim.4780110717.

Hastie, T. and Tibshirani, R. (2000). Bayesian backfitting (with comments and a rejoinder by the authors). *Statistical Science*, 15(3):196–223. doi:10.1214/ss/1009212815.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York. doi:10.1007/978-0-387-84858-7.

Hesterberg, T., Choi, N. H., Meier, L., and Fraley, C. (2008). Least angle and l_1 penalized regression: A review. *Statistics Surveys*, 2:61–93. doi:10.1214/08-SS035.

Hoerl, A. E. (1962). Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58:54–59.

Hoff, P. D. (2015). Multilinear tensor regression for longitudinal relational data. doi:10.1214/15-AOAS839.

Holmes, C. C., Denison, D. G. T., Ray, S., and Mallick, B. K. (2005). Bayesian prediction via partitioning. *Journal of Computational and Graphical Statistics*, 14(4):811–830. doi:10.1198/106186005X78107.

Ishwaran, H. and Rao, J. S. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. *Annals of Statistics*, 33(2):730–773. doi:10.1214/009053604000001147.

Jacob, P. E., Murray, L. M., Holmes, C. C., and Robert, C. P. (2017). Better together? statistical learning in models made of modules. arXiv:1708.08719.

Jeong, J., Vannucci, M., and Ko, K. (2013). A wavelet-based Bayesian approach to regression models with long memory errors and its application to fMRI data. *Biometrics*, 69:184–196. doi:10.1111/j.1541-0420.2012.01819.x.

Kaisera, A., Hallerb, S., Schmitzd, S., and Nitscha, C. (2009). On sex/gender related similarities and differences in fMRI language research. *Brain Research Reviews*, 61:49–59. doi:10.1016/j.brainresrev.2009.03.005.

Kerkyacharian, G. and Picard, D. (2004). Regression in random design and warped wavelets. *Bernoulli*, 10(6):1053–1105. doi:10.3150/bj/1106314850.

Kleijn, B. and van der Vaart, A. (2012). The Bernstein-Von-Mises theorem under misspecification. *Electron. J. Statist.*, 6:354–381.

Koscik, T., O'Leary, D., Moser, D. J., Andreasen, N. C., and Nopoulos, P. (2009). Sex differences in parietal lobe morphology: Relationship to mental rotation performance. *Brain and Cognition*, 69(3):451–459. doi:10.1016/j.bandc.2008.09.004.

Kovac, A. and Silverman, B. W. (2000). Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *Journal of American Statistical Association*, 95(449):172–183. doi:10.1080/01621459.2000.10473912.

Lee, M. R., Cacic, K., Demers, C. H., Haroon, M., Heishman, S., Hommer, D. W., Epstein, D. H., J.Ross, T., Stein, E. A., Heilig, M., and Salmeron, B. J. (2014). Gender differences in neural–behavioral response to self-observation during a novel fMRI social stress task. *Neuropsychologia*, 53:257–263. doi:10.1016/j.neuropsychologia.2013.11.022.

Letham, B., Rudin, C., McCormick, T. H., , and Madigan, D. (2015). Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Annals of Applied Statistics*, 9(3):1350–1371. doi:10.1214/15-AOAS848.

Li, F. and Zhang, N. R. (2010). Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics. *Journal of the American Statistical Association*, 105(491):1202–1214. doi:10.1198/jasa.2010.tm08177.

Li, X., Xu, D., Zhou, H., and Li, L. (2018). Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, 10(3):520–545. doi:10.1007/s12561-018-9215-6.

Liu, F., Bayarri, M. J., and Berger, J. O. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150. doi:10.1214/09-BA404.

Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., and Bach, F. R. (2009). Supervised dictionary learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 1033–1040. Curran Associates, Inc.

Marin, J.-M. and Robert, C. P. (2007). *Bayesian core: a practical approach to computational Bayesian statistics.* Springer Publishing Company, Incorporated. doi:10.1007/978-0-387-38983-7.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032. doi:10.2307/2290129.

Morris, J. S. (2015). Functional regression. *Annual Review of Statistics and Its Application*, 2(1):321–359. doi:10.1146/annurev-statistics-010814-020413.

Müller, P. and Quintana, F. (2004). Nonparametric Bayesian data analysis. *Statistical Science*, 19(1):95–110. doi:10.1214/088342304000000017.

Murphy, K. M. and Topel, R. H. (1985). Estimation and inference in two-step econometric models. *Journal of Business & Economic Statistics*, 3(4):370–379. doi:10.1198/073500102753410417.

Nason, G. (2008). *Wavelet Methods in Statistics with R.* Springer Publishing Company, Incorporated, 1st edition. doi:10.1007/978-0-387-75961-6.

Nobel, A. (1996). Histogram regression estimation using data-dependent partitions. *Annals of Statistics*, 24(3):1084–1105. doi:10.1214/aos/1032526958.

Park, M. Y., Hastie, T., and Tibshirani, R. (2007). Averaged gene expressions for regression. *Biostatistics*, pages 212–227. doi:10.1093/biostatistics/kxl002.

Plummer, M. (2015). Cuts in Bayesian graphical models. *Statistics and Computing*, 25(1):37–43. doi:10.1007/s11222-014-9503-z.

Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using Polya-Gamma latent variables. *Journal of the American Statistical Association*, 108:1339–1349. doi:10.1080/01621459.2013.829001.

Pratola, M., Chipman, H., George, E., and McCulloch, R. (2017). Heteroscedastic BART Using Multiplicative Regression Trees. arXiv:1709.07542.

Quinlan, R. (1993). Combining instance-based and model-based learning. In *Proceedings on the Tenth International Conference of Machine Learning*, pages 236–243. University of Massachusetts, Amherst. Morgan Kaufmann. doi:10.1.1.34.6358.

Ramsay, J. and Silverman, B. (1997). *Functional Data Analysis*. Springer-Verlag New York. doi:10.1007/b98888.

Reiss, P. T., Goldsmith, J., Shang, H. L., and Ogden, R. T. (2017). Methods for scalar-on-function regression. *International Statistical Review*, 85(2):228–249. doi:10.1111/insr.12163.

Rockova, V. and van der Pas, S. (2017). Posterior concentration for Bayesian regression trees and forests. arXiv:1708.08734.

Samek, W., Wiegand, T., and Müller, K. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv:1708.08296.

Sardy, S. and Tseng, P. (2004). AMlet, RAMlet, and GAMlet: Automatic nonlinear fitting of additive models, robust and generalized, with wavelets. *Journal of Computational and Graphical Statistics*, 13(2):283–309. doi:10.1198/1061860043434.

Scarlett, J. and Cevher, V. (2017). Limits on support recovery with probabilistic models: An information-theoretic framework. *IEEE Transaction on Information Theory*, 63:593 – 620. doi:10.1109/TIT.2016.2606605.

Schiffler, P., Tenberge, J.-G., Wiendl, H., and Meuth, S. G. (2017). Cortex parcellation associated whole white matter parcellation in individual subjects. *Frontiers in Human Neuroscience*, 11:352. doi:10.3389/fnhum.2017.00352.

Scott, A. J. and Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 30(3):507–512. doi:10.2307/2529204.

Thirion, B., Varoquaux, G., Dohmatob, E., and Poline, J.-B. (2014). Which fMRI clustering gives good brain parcellations? *Frontiers in Neuroscience*, 8:167. doi:10.3389/fnins.2014.00167.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58:267–288. doi:10.1.1.35.7574.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, pages 91–108. doi:10.1111/j.1467-9868.2005.00490.x.

Tomasi, D. and Volkow, N. D. (2012). Gender differences in brain functional connectivity density. *Human Brain Mapping*, 33:849–860. doi:10.1002/hbm.21252.

van der Pas, S. and Rockova, V. (2017). Bayesian dyadic trees and histograms for regression. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 2089–2099. Curran Associates, Inc.

Vidakovic, B. (2009). *Statistical modelling by wavelets.* John Wiley & Sons. doi:10.1002/9780470317020.

Wasserman, L. and Roeder, K. (2009). High dimensional variable selection. *Annals of Statistics*, 37(5A):2178–2201. doi:10.1214/08-AOS646.

Weller, A. (2017). Challenges for transparency. arXiv:1708.01870.

Wold, H. (1966). *Estimation of Principal Components and Related Models by Iterative Least squares*, pages 391–420. Academic Press, New York.

Wold, H. (1975). Soft modelling with latent variables: The non-linear iterative partial least squares (nipals) approach. In Gani, J., editor, *Perspectives on Probability and Statistics, Festschrift (65th Birthday) for M. S. Bartlett*, pages 117–142. Academic Press.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional linear regression analysis for longitudinal data. *Annals of Statistics*, 33(6):2873–2903. doi:10.1214/009053605000000660.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67. doi:10.1111/j.1467-9868.2005.00532.x.

Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563.

Zhao, Y., Ogden, R. T., and Reiss, P. T. (2012). Wavelet-based lasso in functional linear regression. *Journal of Computational and Graphical Statistics*, 21(3):600–617. doi:10.1080/10618600.2012.679241.

Zhou, H., Li, L., and Zhu, H. (2013). Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552. doi:10.1080/01621459.2013.776499.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320. doi:10.1111/j.1467-9868.2005.00503.x.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286. doi:10.1198/106186006X113430.