Regularized Tensor Quantile Regression for Functional Data Analysis with Applications to Neuroimaging

by

Matthew Pietrosanu

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Statistics

Department of Mathematical & Statistical Sciences

University of Alberta

Abstract

With accelerating technological advancements, functional data analysis is of ever-growing importance in statistics, engineering, and medicine as the collection, storage, and analysis of data becomes ingrained in every aspect of modern life beyond any one field of study. From continuous biometric measurements to environmental monitoring to medical imaging, data collected across time, space, and mathematical manifolds in general are increasingly common. The advent of "big data" ensures that the rate, volume, and complexity of collected data will only continue to grow. Despite its richness, the high dimensionality of functional data is inherently challenging to incorporate into statistical models in both theory and practice.

This thesis is primarily concerned with incorporating multiple scalar-valued functional covariates, each with a high-dimensional domain but observed in a discrete, uniform grid, into functional quantile regression models for a scalar response. This type of functional data is ubiquitous in science and medicine as imaging data observed across spatiotemporal dimensions in a tensor format. The existing accommodation of functional and tensor-valued covariates in generalized linear models does not extend immediately to the quantile setting, although such a development would be useful in practice. Indeed, quantile models are well-known to be more efficient and robust for non-Gaussian, heavy-tailed error distributions or when outliers are present—typically the case with real-world data. Throughout this work, we emphasize direct regularization of tensor effects for more generalizable models and interpretable signal recovery

for imaging data.

The Tucker tensor decomposition is the main tool used in this thesis: we assume a low-dimensional representation of a tensor with a particular polyadic structure to reduce the dimension of the parameter space and make model estimation feasible. We obtain this decomposition in a supervised manner, relying on partial quantile covariance between tensor covariates and the scalar response, similar to partial least squares in traditional linear regression. We estimate decomposition parameters and fit the proposed q-dimensional tensor quantile regression (qD-TQR) model by minimizing quantile loss. To address the non-convexity and non-differentiability of the loss in Tucker tensor decomposition parameters, we use a block relaxation technique and a continuously differentiable smoothing approximation of the quantile loss. After proposing a new algorithm and gradient-based implementation for models with one functional covariate, we extend our approach to multiple functional covariates and discuss simplifications exploiting the Tucker decomposition's nonsingular transformation indeterminacy. We consider convex penalty functions that, unlike previous approaches, directly regularize the estimated tensor effect through the assumed structure rather than only its decomposition parameters.

We establish theoretical properties for the proposed model, including global, local, and approximation convergence for the proposed algorithm and, using empirical process theory, asymptotic statistical results regarding estimator consistency and normality.

Finally, we demonstrate the performance of our model in simulated and real-world settings. Through a simulation study proposed in previous works that attempt to recover image signals of various geometric shape, we highlight the superiority of quantile-based methods for heavy-tailed error distributions. We examine the effect of tensor decomposition rank, quantile level, signal-to-noise ratio, and sample size on model estimates, further improving signal

recovery by using a LASSO-type penalty. Second, we apply our methods to a real-world neuroimaging dataset from the Alzheimer's Disease Neuroimaging Initiative. Our model relates clinical covariates and and four functional covariates obtained from magnetic resonance imaging scans to mini-mental state examination score, a screening tool for Alzheimer's disease. After LASSO regularization leaves more be desired in estimate interpretability, we explore fused LASSO penalization to enforce estimate smoothness in a post-hoc analysis. Results show improvement that would not be possible with previous work through direct penalization of decomposition parameters.

The major work in this thesis fills the need for an extension of existing functional quantile methods to tensor and high-dimensional functional data. Our results furthermore address the practical issue of multiple functional covariates—typically ignored in other work—and demonstrate the utility of direct regularization in tensor effect interpretability for imaging data.

To Lisa and to my parents, Dan and Dolores,
For their endless encouragement, patience, and sacrifice in supporting me and my education.

Acknowledgements

I am extremely grateful to my supervisors, Drs. Linglong Kong and Bei Jiang, for their support throughout my graduate studies and for the opportunities they continue to provide for my learning. This work and my studies in general have benefited greatly from their insights. I am grateful to the administrative staff in the Department of Mathematical & Statistical Sciences, particularly Ms. Tara Schuetz, for her work with graduate students and for facilitating the graduate program.

Beyond the Faculty of Graduate Studies and Research, Faculty of Science, and the Department of Mathematical & Statistical Sciences, my work towards this thesis and related conference/travel expenses are externally supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Servus Credit Union, the Graduate Students' Association, and the Mathematical Sciences Research Institute (MSRI).

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.;

Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (`www.fnih.org`). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Contents

List of Tables

List of Figures

List of (Select) Symbols

$\times$	Cartesian product
$\triangleq$	Definition
$\mathbb{E}$	Expected value
$Q_\tau(Y\|\boldsymbol{X})$	Conditional level-τ quantile of scalar random variable Y given random vector $\boldsymbol{X}$
$\mathrm{Cov}_\tau(Y, Z \,\|\, \boldsymbol{X})$	Quantile covariance of scalar random variables Y and Z given random vector $\boldsymbol{X}$
$\nabla_{\boldsymbol{\theta}} f$	Gradient (column) vector of a function f with respect to (the vectorization of) parameters $\boldsymbol{\theta}$
$\frac{\partial f}{\partial \boldsymbol{x}}$	Derivative matrix in $\mathbb{R}^{q\times p}$ of function f mapping to $\mathbb{R}^q$ with respect to $\boldsymbol{x} \in \mathbb{R}^p$, or in $\mathbb{R}^{I_1\times\cdots\times I_d}$ for scalar-valued f with $\boldsymbol{x} \in \mathbb{R}^{I_1\times\cdots\times I_d}$
$\xrightarrow{p}$	Convergence in probability (with limiting variable contextually clear)
o_p	Order in probability (small o): $x_n = o_p(a_n) \iff \frac{x_n}{a_n} = o_p(1) \iff \lvert\frac{x_n}{a_n}\rvert \xrightarrow{p} 0$
ρ_τ	Quantile loss function for τ-th quantile
$\rho_{\tau\nu}$	Smoothed quantile loss function for the τ-th quantile with smoothing parameter ν
$\otimes$	Matrix Kronecker product
$\overline{\cdots}^{-d}$	Sequence omitting index d
$\circ$	Tensor outer product
$\times_d$	Mode-d tensor multiplication
vec	Tensor vectorization
$\langle\cdot,\cdot\rangle$	Tensor inner product
$\cdot_{[[d]]}$	Mode-d tensor matricization
$[[\boldsymbol{\Lambda} \,\|\, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}]]$	Tucker decomposition with core tensor $\boldsymbol{\Lambda}$ and factor matrices $\boldsymbol{\Gamma}^{(d)}$
I_d	Identity matrix in $\mathbb{R}^{d\times d}$
I_Q	Indicator taking value 1 if the predicate Q is true, and 0 otherwise.

Chapter 1

Introduction

We first provide a general overview of and introduction to relevant topics and existing results. Our intention is to give sufficient context for the specific research presented in this thesis, but also to more cleanly separate our work from more recent results. An overview of this thesis and its contributions is provided at the end of this chapter in Subsection 1.4.

1.1 Functional Data Analysis

Functional data analysis (FDA) is concerned with data generated from curves, manifolds, and other structures that can be cast as functions of one or more variables [65]. This includes, among myriad examples, the growth patterns of children as a function of age [38]; annual precipitation or temperature patterns as a function of day [31] (Figure 1.1); and medical imaging scans as a function of spatial position and, potentially, another temporal variable [29]. In the functional data setting, as first called by Ramsay [64], observations are functions or "curves" that, in theory, could be evaluated anywhere on the function's domain [26], [69]. This is fundamentally different from more traditional settings where observations are scalar. Despite its differences, FDA shares the same goals as other data analytic approaches [65]. FDA aims to represent data and highlight salient characteristics; separate true, underlying signal from noise to identify sources of variation; and model variation in a functional [53], [94] or scalar [67] response using functional or scalar explanatory variables. These goals highlight the exploratory, confirmatory, and predictive objectives

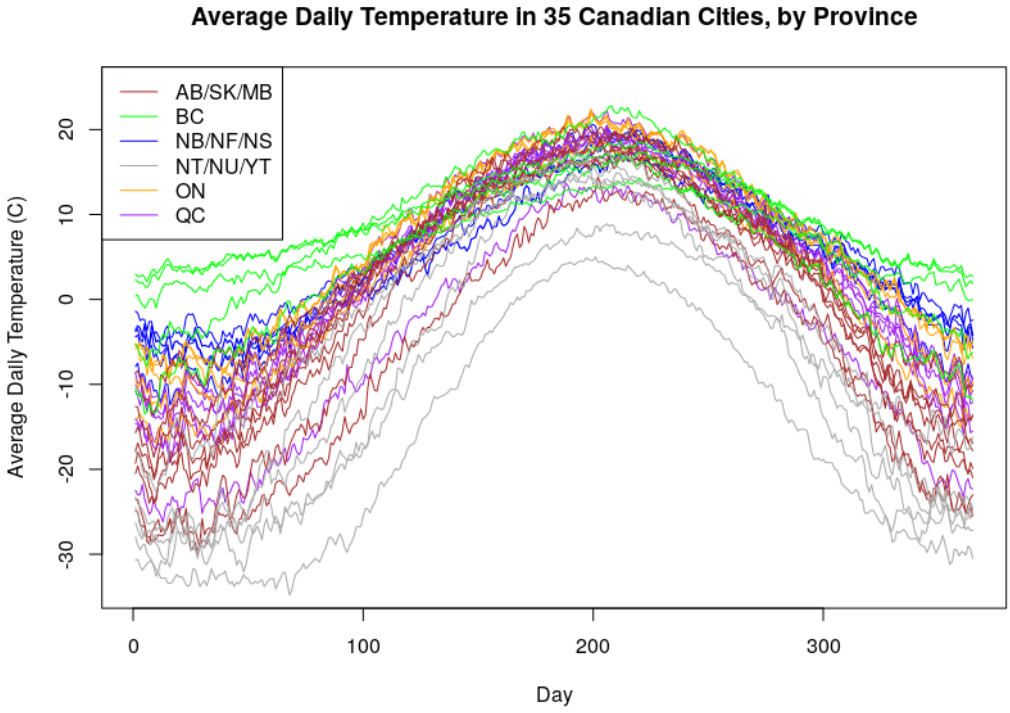

Figure 1.1: An example of functional data: average daily temperature in 35 Canadian cities over one year, coloured according to the province grouping specified in the legend. Data is available in the `fda` package [66] for `R` [63]

of FDA, respectively.

Functional data present unique analytic and computational challenges due to the intrinsically high dimensionality of functional space. While the setting with pure functional observations has been studied since the beginnings of FDA [28], it is most common in theory and practice to deal with observations at certain (fixed or random) points of a domain. Whether dense, sparse, or neither, these observations need not be evaluated at the same points across observations: this variability in the sampling design presents additional theoretical challenges and can impact the convergence rates of statistical estimators [32], [84]. While FDA is well-suited to address variation—generally referred to as *measurement error* whether stochastic or due to actual measurement error—other difficulties appear in "next generation" functional data [75], [84] when strong covariance structures exists within functional observations, as with spatial correlation in images, or when functional variables are correlated. This

problem is particularly relevant in the area of imaging genetics, which aims to combine highly correlated neuroimaging and genetic data into statistical models [58].

FDA has received much attention in the Statistical literature. This focus has certainly not waned in recent years with the ever-growing attention to "big data" and technological advances proliferating complex data structures to every field of application from chemometrics [27] to climate science [31] to neuroimaging [2]. In particular, functional regression models, including linear [11], [57], non-linear [56], and non-parametric models [24], are arguably the most popular tools in FDA and have progressed substantially since their initial development. Common approaches to FDA include kernel smoothing, local regression, and splines [8], [20], [33], [83], also drawing from theory in functional analysis and stochastic processes.

Functional regression models can be characterised by the role of functional data in the model—whether as a predictor, response, or both. The simplest of these is perhaps the traditional *functional linear regression* model with a scalar response and a single scalar functional predictor [67], given by

$$\mathbb{E}[Y|\boldsymbol{x}, z] = \alpha + \boldsymbol{x}^\top \boldsymbol{\beta} + \int_D \gamma(t) z(t) dt.$$

In this model, $Y \in \mathbb{R}$ is a scalar response, $\boldsymbol{x}, \boldsymbol{\beta} \in \mathbb{R}^p$ are scalar observation and effect vectors, and $z, \gamma : D \to \mathbb{R}$ represent the functional response and effect. The form of even this simple model immediately suggests inherent theoretical and computational challenges.

First, the functional effect γ most generally lies in infinite-dimensional functional space and cannot be estimated in practice without further restrictions. One immediate solution is to project γ into a finite-dimensional functional subspace satisfying desired smoothness conditions [42], but this raises the question of an appropriate functional basis. This choice of basis has been thoroughly investigated: existing literature has many examples of functional regression using general functional bases (e.g., B-splines [9] and wavelets [95]), functional principal components (FPC) bases [16], [28], and even functional partial least squares bases [68]—an analogue to partial least squares in traditional linear

regression [34], [89]. For a more thorough review of the functional linear model and recent developments, see the surveys by Morris [54] and Wang [84].

As a second notable point, the domain of integration D in the model above is usually taken to be the unit interval $[0, 1] \subseteq \mathbb{R}$ (after proper scaling of the inputs of z), although we can also consider D in higher-dimensional spaces, typically $D = [0, 1]^q \subseteq \mathbb{R}^q$. The former suggests a functional response observed as a vector such as (for example) growth over time or fractional anisotropy (a neuroimaging measure) along the brain's corpus callosum fiber tract as a function of arc length (1-dimensional spatial location) [91]. A domain in $\mathbb{R}^q$ suggests a functional response observed as a *tensor* (which we take in the usual sense to mean a q-dimensional array and generalization of a matrix). This suggests a fundamental relationship between functional and tensor regression models. As the generalization from the $q = 1$ to $q > 1$ case is not immediate, less work has been conducted in this area at present. However, major results by Zhou et al. [97] and Li et al. [49] establish generalized linear models (GLMs) incorporating tensor covariates, later adding the capacity for tensor effect regularization [96]. These works take advantage of low-dimensional tensor approximations such as the CP or Tucker decompositions [46] to reduce the generally high dimension of the tensor effect and make model estimation feasible. Earlier work obtained tensor decompositions in an unsupervised manner [3], [15], [37], while Zhou et al. and Li et al. take a supervised approach. In terms of regularization, these authors consider typical penalties such as elastic net (which includes LASSO [78] and L_2 losses) [98], SCAD [21], and MCP [93] applied to individual parameter blocks of the tensor decomposition (namely, elements of the factor matrix for CP decompositions and elements of the core tensor for Tucker decompositions). Most recently, Zhou et al. [96] implemented spectral regularization in the tensor GLM for matrix predictors ($q = 2$) based on the nuclear norm, which penalizes the sum of the singular values of the estimated tensor (matrix) effect.

1.2 Functional Quantile Regression

Linear quantile regression was first proposed in the seminal work of Koenker and Bassett [45] as an alternative to traditional least-squares regression robust to outlier contamination and misspecification of the (typically non-Gaussian) error distribution. Rather than modelling a conditional mean response, quantile regression considers the conditional level-τ quantile

$$Q_\tau(Y|\boldsymbol{x}) \triangleq \inf\{\pi_\tau \,|\, F_Y(\pi_\tau) \geq \tau\}$$

for a fixed $\tau \in (0,1)$ through the linear model

$$Q_\tau(Y|\boldsymbol{x}) = \alpha + \boldsymbol{x}^\top \boldsymbol{\beta}.$$

Estimates for model parameters are defined through M-estimation via

$$\hat{\alpha}, \hat{\boldsymbol{\beta}} \triangleq \arg\min_{\alpha,\boldsymbol{\beta}} \sum_{i=1}^{n} \rho_\tau(y_i - \alpha - \boldsymbol{x}^\top \boldsymbol{\beta}),$$

where $\rho_\tau(u) \triangleq u\big(\tau - I(u<0)\big)$ is the *quantile loss function* (also called the *check function*).

It is a well-established result that these quantile regression estimators, relative to least-squares estimators, are more efficient and robust when the error distribution is non-Gaussian and/or heavy-tailed or when outliers are present. Furthermore, quantile regression is capable of dealing with error heteroscedasticity and can give greater insight into the distribution of the response: this is particularly true of *composite quantile regression* [99], which simultaneously models multiple conditional quantiles $\tau_1, \ldots, \tau_K$, assuming that $\boldsymbol{\beta}$ remains constant over these quantile levels.

Quantile regression has since been extended to the 1-variable functional data setting through recent developments by Kato [43] and Yu et al [91]. Kato implemented FPC bases while also noting the tractability of Fourier and wavelet bases. B-splines [10] and other general bases along with FPC bases [50], [77] have been investigated by numerous authors. Based on the work of Dodge and Whittaker [18] on partial bases in the non-functional quantile setting, Yu et al. proposed a functional partial quantile (FPQ) basis for

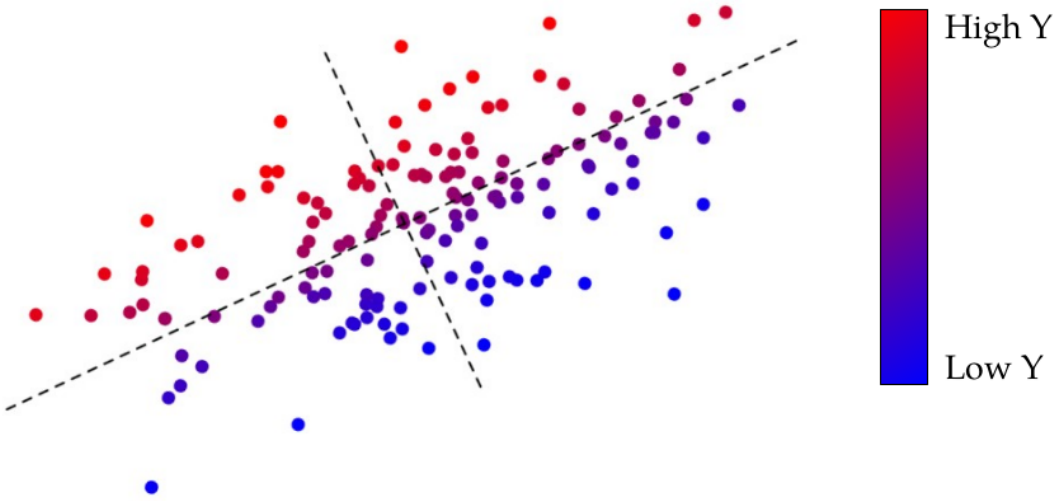

Figure 1.2: Illustration of a dataset (in $\mathbb{R}^2$) where projections onto the first principal component (the longer of the two dotted lines) are uncorrelated with the response Y (with values indicated by the colour scale on the right). In truth, response values are assigned according to the scalar projection of each point onto the second principal component (the shorter of the two dotted lines)—perpendicular to the first.

vector-valued functional covariates that, like other partial bases, is obtained in a supervised manner and is thus more data-driven than FPC methods by virtue of its incorporation of response information. That is, while an FPQ basis is optimally predictive of the response conditional quantile (in the sense of *partial quantile covariance*, defined below), the FPC basis, which only represents orthogonal directions of maximal variation in the functional predictor z, will generally not have this property. In the most extreme case, projections of the functional response onto the FPC basis will be useless as predictors in the functional quantile model. This scenario is illustrated (for vectors rather than functions for interpretability) in Figure 1.2.

Yu et al.'s FPQ basis is defined through the notion of partial quantile covariance, first proposed by Dodge and Whittaker [18] for the traditional quantile regression model. In particular, the partial quantile covariance between random variables Y and Z after (optionally) accounting for a random

vector $\boldsymbol{X}$ is

$$\mathrm{Cov}_\tau(Y, Z|\boldsymbol{X}) \triangleq \arg\inf_{\gamma} \inf_{\alpha,\boldsymbol{\beta},\gamma} \mathbb{E}\rho_\tau(Y - \alpha - \boldsymbol{X}^\top\boldsymbol{\beta} - \gamma Z).$$

Intuitively, partial quantile covariance measures the contribution of Z to the level-τ quantile of Y after accounting for $\boldsymbol{X}$.

Yu et al. propose using an FPQ basis $\Phi = \{\phi_1, \ldots, \phi_K\}$ in the model with a single functional covariate for some pre-specified basis size K by solving the optimization problem

$$\arg\max_{\Phi} \mathrm{Cov}_\tau\Big(Y, \sum_{k=1}^{K} \int_0^1 z(t)\phi_k(t)dt|\boldsymbol{X}\Big), \tag{1.1}$$

although the authors do not estimate the ϕ_k, but instead obtain evaluations of these basis elements on discrete, uniform grid.

Following Kato [43] and others [50], [77], Yu et al. empirically select an optimal basis size K using a goodness-of-fit criteria such as BIC or cross-validated test quantile loss. The authors' major contribution includes an algorithm for solving the optimization problem 1.1. Their approach employs block relaxation [47] to permit sequential updates of the estimated FPQ basis, similar to Li et al. and Zhou et al. [49], [97], as well as a smooth approximation of the non-differentiable quantile loss ρ_τ [12], [55].

Yu's doctoral thesis [90] begins generalizing this approach to q-dimensional functional data $z : [0, 1]^q \to \mathbb{R}$. Applying the same techniques as above and the previous results of [49], [97], Yu proposes partial quantile regression for the multidimensional functional linear model. To our knowledge, no other approaches to FPQ basis estimation exist in the literature. Relationships to the work in this thesis are discussed in Subsection 1.4, where we discuss the necessity of our investigations in formalizing the qD-FLQR model and theory for publication and future development.

With the proliferation of "big data" came rising interest in methods for sparse regression, variable selection, and estimate regularizarion in high-dimensional settings. As discussed in the previous subsection, this has typically involved penalties such as LASSO, SCAD, and MCP. These are well-studied in the

GLM setting, including with functional predictors [21], [78], [93], [96]. In the traditional quantile model, interior point approaches are well-established and widely implemented in existing software packages [44]. In light of the non-differentiability of the quantile loss, Pietrosanu et al. [62] most recently provided three reformulations of the (composite) quantile regression problems, both with and without adaptive LASSO regularization, enabling the application of different computational algorithms. These include alternating direction method of multipliers, majorize-minimization, and coordinate descent. The amount of improvement over interior point methods in estimation accuracy and/or runtime depends on context (e.g., $p > n$, $p \gg n$, $n \gg p$, etc.).

To our knowledge, no work has been done to implement the above penalties in the functional quantile setting. Furthermore, excepting the nuclear norm noted in Subsection 1.1 for matrix predictors, no work exists on methods for penalizing tensor effect estimates (as opposed to individual components of a corresponding low-dimensional representation) in either the quantile or GLM setting. Despite this, noise in estimated effects is a common problem, particularly in imaging data analysis, suggesting a need to penalize functional effect estimates and obtain less-noisy effects that are visually easier to interpret. Unlike individual tensor decomposition parameters, the tensor effect is typically of primary interest to applied researchers. In neuroimaging, for example, tensor effects can identify regions of the brain that are associated with the development or severity of a disease or other outcomes of clinical interest. Consequently, directly regularizing tensor effect estimates rather than just their low-dimensional representation may yield more interpretable results.

1.3 Neuroimaging Applications

The analysis of neuroimaging data is crucial to an understanding of human brain function and development [30] as well as neurological disorders such as substance abuse [70], anxiety [25], schizophrenia [81], Alzheimer's disease [88], attention deficit hyperactivity disorder [14], autism [4], and other neurodegenerative conditions. Statistical methods for analyzing complex modes of data

generated by modern medical imaging technologies further the development of preventative, diagnostic, and treatment procedures for these conditions.

Modern neuroimaging permits the examination of brain function and structure through computed tomography, magnetic resonance imaging (MRI), functional magnetic resonance imaging (fMRI) [29], [40], diffusion tensor imaging (DTI) [2], electroencephalography, positron emission tomography, and many other data modes [61]. For example, fMRI and DTI use blood oxygen levels and the diffusivity of water molecules in fibre tracts, respectively, to map brain activity and region activation. These measures can be used to assess and compare brain connectivity as well as the functional integration of numerous brain regions in different populations such as neuodegenerative cases and controls. Some well-known studies that make use of neuroimaging data include the Alzheimer's Disease Neuroimaging Initiative (ADNI) [88], the ADHD-200 Sample Initiative Project [7], and the NIH Human Connectome project [19].

MRI (including fMRI and DTI) modalities have demonstrated a wide range of applications. By describing three-dimensional molecular properties such as diffusion—a random transport phenomenon across cellular membranes—as a function of spatial location [2], these methods are able to detect changes in membrane microstructure. For example, Basser et al. [6] modelled the diffusivity of water travelling in the brain's fibre tracts along orthogonal coordinate axes x, y, and z using a multivariate normal distribution. The covariance matrix $\boldsymbol{D}$ in this model, called the *diffusion tensor*, describes three-dimensional diffusion covariance along these axes. Diffusion is called *isotropic* if the eigenvalues of $\boldsymbol{D}$ are roughly equal. *Anisotropy*, differences in these eigenvalues, can be caused by local changes in brain tissue structure due to normal (e.g., aging) or abnormal (e.g., injury and disease) neurophysiological changes.

Continuing the previous interpretable example, while DTI data is extremely rich, with a 3×3 diffusion tensor $\boldsymbol{D}$ at every voxel, visualizing or interpreting this data is a challenging task [2]. It is typical to summarize $\boldsymbol{D}$ at each voxel using a scalar map, resulting in a 3-dimensional tensor summary of all voxels. The trace $\mathrm{Tr}(\boldsymbol{D})$ of $\boldsymbol{D}$ (equal to the sum of the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of $\boldsymbol{D}$) or the mean diffusivity (MD) (simply the average eigenvalue $\bar{\lambda} = \frac{1}{3}\mathrm{Tr}(\boldsymbol{D})$)

are commonly used. Many measures of anisotropy exist, although *fractional anisotropy* (FA), given by

$$\text{FA} \triangleq \sqrt{\frac{\sum_{i=1}^{3}(\lambda_i - \bar{\lambda})^2}{\sum_{i=1}^{3}\lambda_i^2}},$$

is popular. These measures have been used previously to study ischemic stroke, demyelination, inflammation, edema, neoplasia, and other conditions. Different summary measures or combinations of eigenvalues (such as the minimum, maximum, or other linear combinations) speak to different pathologies [2].

Similar problems are present for other neuroimaging data modes. T1-weighted MRI measurements are most commonly used to assess atrophy in different parts of the brain [23]. Structural properties including volume [41] and surface-based measures such as radial distance [76], local area differences, and tensor-based morphemetry (TBM) [17] convey morphometric information on brain structures of interest such as the hippocampus and are known to be indicative of atrophy in these structures. Multivariate TBM (mTBM) [48], [85], a generalization of TBM, examines spatial deformation in multiple directions along a surface and has shown greater statistical power and improved signal detection than other measures. This is especially true when combined with complementary measures such as radial distance, which examines deformation in the surface normal direction [85], [86].

In general, statistical methods to automatically analyze neuroimaging data depend on tensor-valued summary measures extracted from raw data and neurological reconstructions. The previous discussions illustrate how one measure is not sufficient to summarize relevant information in neuroimaging data [1], [52], [74], [87]. Different measures are sensitive to different pathologies, while one measure can suggest multiple pathologies and might yield the same value for different eigenvalue configurations. In this light, statistical methods that efficiently accommodate multiple tensor covariates are necessary to take full advantage of the richness of neuroimaging data, but are not well-developed in the literature.

1.4 Thesis Overview and Contributions

This thesis proceeds with further theoretical background material regarding tensors, tensor operations, and tensor decompositions in Chapter 2. Chapter 3 follows with our formulation of the q-dimensional functional linear quantile regression (qD-FLQR) model (with one tensor covariate) and discusses, in Section 3.2, details pertaining to a gradient-based implementation with supervised tensor decomposition, including an algorithm for model estimation. Section 3.3 extends the qD-FLQR model to multiple tensor covariates, not necessarily of the same size, and a penalty term to regularize tensor covariate effect estimates. We examine algorithm convergence and estimator asymptotics in Chapter 4. A simulation study and analysis using a real-world neuroimaging dataset are carried out in Chapter 5. Finally, we summarize and discuss our results and future work in Chapter 6.

The major work in this thesis extends existing results accommodating tensor covariates in functional GLMs [49], [97] to the functional quantile setting to create a tensor quantile model. The approach using tensor decompositions and theoretical justification is similar to that for GLMs, as pointed out by Yu in his doctoral thesis on partial functional bases and supervised decompositions for functional data analysis in quantile regression [90]. However, we found that more attention was required to establish theoretical properties of the algorithm and asymptotics of model estimators. These results do not follow from the tensor GLM due to the non-strict convexity of the quantile loss; the inapplicability of standard arguments on quantile regression estimators established by Koenker et al. [45] due to the differentiability of our smoothing approximation; and the equivalency of minimizing quantile loss and maximizing the log-likelihood of independent and identically distributed asymmetric Laplace observations with unknown location parameter, which does not form an exponential family and thus does not fall under the GLM umbrella. These differences necessitate separate treatment of the qD-FLQR model, which we address in adequate detail here.

We are also the first to implement a smoothing approximation of the quan-

tile loss in our tensor quantile model algorithm to examine the performance of the proposed estimators. Previous work only implements (unsmoothed) quantile loss and applies existing software packages [44]. We accommodate multiple tensor covariates (a non-trivial extension wholly ignored in existing literature) as well as convex, differentiable penalties (or convex penalties that can be appropriately smoothed such as LASSO). Unlike previous work, we focus on estimate interpretability and directly regularize tensor effect estimates through the proposed tensor decomposition rather than just decomposition parameter blocks. We demonstrate the utility and, in real-world settings, necessity, of these additions through a simulation study and a novel application to an ADNI neuroimaging dataset. In general, our work further suggests directions for future improvement, including improving the proposed algorithm in its approach to block relaxation, the smoothing parameter update rules to be approximation-specific, and the optimization method in order to accommodate non-convex penalties.

Chapter 2

Tensors and the Tucker Decomposition

The following chapter provides background information on tensors, their properties as generalizations of matrices, and low-dimensional representations necessary for the qD-FLQR model later on. Section 2.1 introduces basic tensor operations while Section 2.2 discusses two *tensor decompositions*, that is, low-dimensional representations of a tensor, that will be of primary interest when discussing the qD-FLQR. The definitions and results in this chapter are consistent with the detailed work of Kolda and Bader [46], but we highlight the necessary components here in order for clarity.

Throughout this thesis, we treat a tensor as a q-dimensional array of real numbers (for some known q) with entries indexed by q-tuples $(i_1, \ldots, i_q)$, where $i_j = 1, \ldots, I_j$. For such a tensor $\boldsymbol{A}$, we write $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$, with $(i_1, \ldots, i_q)$-th element denoted by $\boldsymbol{A}_{i_1 \ldots i_2}$ or $a_{i_1,\ldots,i_q}$. We say that $\boldsymbol{A}$ is *q-dimensional* with *size* $(I_1, \ldots, I_q)$ and d-th *mode* I_d. As with matrices, which we view simply as 2-dimensional tensors, we define addition and subtraction element-wise on tensors of the same size in the usual way. Section 2.1 defines some useful product and matricization operations on tensors.

As a brief remark on notation, we use bold upper- and lower-case type to denote tensors and vectors, respectively. Furthermore, for a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we use $\boldsymbol{a}_i \in \mathbb{R}^n$ for $i = 1, \ldots, n$ to denote the i-th column vector of $\boldsymbol{A}$. These standard conventions hold throughout the rest of this thesis.

2.1 Tensor Operations

The operations presented in this section will be useful when manipulating tensor data later on. We set out notation here to avoid confusion with different conventions across the literature.

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p \times q}$. Define the *Kronecker product* of $\boldsymbol{A}$ and $\boldsymbol{B}$ as

$$\boldsymbol{A} \otimes \boldsymbol{B} \triangleq \begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{bmatrix} = [\boldsymbol{a}_1 \otimes \boldsymbol{B} \dots \boldsymbol{a}_n \otimes \boldsymbol{B}] \in \mathbb{R}^{mp \times nq},$$

where the second equality follows by definition of $\otimes$ in the first.

While the above operations preserve the dimension of their operands, we now introduce a tensor-valued *outer product* $\circ$ between tensors that does not. With $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \dots \times I_q}$ and $\boldsymbol{B} \in \mathbb{R}^{J_1 \times \dots \times J_{q'}}$, define

$$\boldsymbol{A} \circ \boldsymbol{B} \in \mathbb{R}^{I_1 \times \dots \times I_q \times J_1 \times \dots \times J_{q'}}$$

with $(i_1, \dots, i_q, j_1, \dots, j_{q'})$-th element $a_{i_1,\dots,i_q} b_{j_1,\dots,j_{q'}}$. It is clear that this operation is associative but not commutative. In this thesis, we are only interested in the case where $\boldsymbol{A}$ and $\boldsymbol{B}$ are vectors. It follows from the above definition that, with $\boldsymbol{a}_d \in \mathbb{R}^{p_d}$ for $d = 1, \dots, q$,

$$\boldsymbol{a}_1 \circ \dots \circ \boldsymbol{a}_q \in \mathbb{R}^{p_1 \times \dots \times p_q}$$

with $(i_1, \dots, i_q)$-th element $\prod_{d=1}^{q} (\boldsymbol{a}_d)_{i_d}$.

Tensors are inherently difficult to deal with due to their generally high dimensionality. The following operations rearrange the elements of a tensor to create a vector or matrix, to which we can apply standard operations.

To reshape a tensor into a vector, we define the *tensor vectorization* operator

$$\text{vec} : \mathbb{R}^{I_1 \times \dots \times I_q} \to \mathbb{R}^{\prod_{d=1}^{q} I_d}$$

(for given q) so that, for $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \dots \times I_q}$, $\text{vec}\, \boldsymbol{A}$ has elements

$$(\text{vec}\, \boldsymbol{A})_{1 + \sum_{d=1}^{q} (i_d - 1) \prod_{d' < d} I_{d'}} \triangleq \boldsymbol{A}_{i_1,\dots,i_q},$$

for $i_d = 1, \ldots, I_d$. This operation is well-defined, invertible, and generalizes the usual vec operation for matrices that "stacks" the column of the input matrix.

Vectorization induces an intuitive *inner product* $\langle \cdot, \cdot \rangle$ between tensors of the same size. For $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$, define

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle \triangleq \langle \operatorname{vec} \boldsymbol{A}, \operatorname{vec} \boldsymbol{B} \rangle = (\operatorname{vec} \boldsymbol{A})^\top \operatorname{vec} \boldsymbol{B} \in \mathbb{R}.$$

When both arguments are matrices, the tensor inner product has a useful duality property that we will use later on.

Lemma 1 (Duality, Lemma 1, Li et al. [49])**.** *For conformable matrices* $\boldsymbol{A} \in \mathbb{R}^{m \times p}$*,* $\boldsymbol{B} \in \mathbb{R}^{n \times p}$*, and* $\boldsymbol{C} \in \mathbb{R}^{m \times n}$*,*

$$\langle \boldsymbol{A}\boldsymbol{B}^\top, \boldsymbol{C} \rangle = \langle \boldsymbol{A}, \boldsymbol{C}\boldsymbol{B} \rangle.$$

Proof. The proof follows directly, with

$$\langle \boldsymbol{A}\boldsymbol{B}^\top, \boldsymbol{C} \rangle = \sum_{i,j} (\boldsymbol{A}\boldsymbol{B}^\top)_{ij} \boldsymbol{C}_{ij} = \sum_{i,j,k} \boldsymbol{A}_{i,k} \boldsymbol{B}_{j,k} \boldsymbol{C}_{i,j} = \sum_{i,k} \boldsymbol{A}_{i,k} (\boldsymbol{C}\boldsymbol{B})_{i,k} = \langle \boldsymbol{A}, \boldsymbol{C}\boldsymbol{B} \rangle.$$

□

We can similarly define an operation to reshape a tensor into a matrix through the *mode-d matricization* operator

$$\cdot_{[[d]]} : \mathbb{R}^{I_1 \times \cdots \times I_q} \to \mathbb{R}^{I_d \times \prod_{d' \neq d} I_{d'}}.$$

For $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$, $\boldsymbol{A}_{[[d]]}$ has elements

$$(\boldsymbol{A}_{[[d]]})_{i_d,\, 1 + \sum_{d' \neq d} (i_{d'} - 1) \prod_{d'' < d', d'' \neq d} I_{d''}} \triangleq \boldsymbol{A}_{i_1 \ldots, i_q}.$$

Intuitively, this matricization operation creates a matrix $\boldsymbol{A}_{[[d]]}$ in which any given column contains the I_d entries of $\boldsymbol{A}$ with some fixed indices $i_{d'}$ for $d' \neq d$, arranged in some fixed order. The arrangement of rows or columns in $\boldsymbol{A}_{[[d]]}$ is arbitrary, provided that this ordering remains consistent (and compatible with other reshaping operations such as vec). Figure 2.1 gives a concrete example of tensor matricization.

$$\boldsymbol{A} = \left\{ \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix} \right\}$$

$$\boldsymbol{A}_{[[1]]} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}$$

$$\boldsymbol{A}_{[[2]]} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix}$$

$$\boldsymbol{A}_{[[3]]} = \begin{bmatrix} 1 & 4 & 7 & 10 & 2 & 5 & 8 & 11 & 3 & 6 & 9 & 12 \\ 13 & 16 & 19 & 22 & 14 & 17 & 20 & 23 & 15 & 18 & 21 & 24 \end{bmatrix}$$

Figure 2.1: An example demonstrating tensor reshaping operations. (Top) A tensor $\boldsymbol{A} \in \mathbb{R}^{3\times4\times2}$ with example element $\boldsymbol{A}_{3,2,1} = 6$. The vectorization $\operatorname{vec} \boldsymbol{A}$ of $\boldsymbol{A}$ is $(1, 2, \ldots, 24)^\top$. (Middle and bottom) The mode-1, mode-2, and mode-3 matricizations of $\boldsymbol{A}$.

The high dimensionality of tensors makes it cumbersome to define a tensor multiplication analogous to the standard matrix multiplication. By holding all but one index constant, however, we can obtain a 1-dimensional *fibre* of a tensor to which the usual matrix-vector multiplication applies. This is the motivation behind the *mode-d product of a tensor by a matrix*,

$$\times_d : \mathbb{R}^{I_1\times\cdots\times I_q} \times \mathbb{R}^{J\times I_d} \to \mathbb{R}^{I_1\times\cdots\times I_{d-1}\times J\times I_{d+1}\times\cdots\times I_q}.$$

For $\boldsymbol{A} \in \mathbb{R}^{I_1\times\cdots\times I_q}$ and $\boldsymbol{U} \in \mathbb{R}^{J\times I_d}$, define

$$(\boldsymbol{A} \times_d \boldsymbol{U})_{i_1,\ldots,i_{d-1},j,i_{d+1},\ldots,i_q} \triangleq \sum_{i_d=1}^{I_d} \boldsymbol{A}_{i_1,\ldots,i_q} \boldsymbol{U}_{j,i_d}.$$

The above is suggestive of matrix multiplication: indeed, it is clear that

$$(\boldsymbol{A} \times_d \boldsymbol{U})_{[[d]]} = \boldsymbol{U}\boldsymbol{A}_{[[d]]}$$

since any fixed column of $\boldsymbol{B}_{[[d]]}$ has elements with constant indices $i_{d'}$ for all $d' \neq d$ in $\boldsymbol{A}$, as in the above summation.

It is easy to see that mode-d products for different modes commute. That is, for $d \neq d'$,

$$(\boldsymbol{A} \times_d \boldsymbol{U}) \times_{d'} \boldsymbol{V} = (\boldsymbol{A} \times_{d'} \boldsymbol{V}) \times_d \boldsymbol{U}. \tag{2.1}$$

Indeed,

$$\begin{aligned}
\big((\boldsymbol{A} \times_d \boldsymbol{U}) \times_{d'} \boldsymbol{V}\big)_{i_1,\ldots,i_{d'-1},j,i_{d'+1},\ldots,i_q} &= \sum_{i_{d'}} (\boldsymbol{A} \times_d \boldsymbol{U})_{i_1,\ldots,i_q} \boldsymbol{V}_{j,i_{d'}} \\
&= \sum_{i_{d'}} \sum_{j_d} \boldsymbol{A}_{i_1,\ldots,i_{d-1},j_d,i_{d+1},\ldots,i_q} \boldsymbol{U}_{i_d,j_d} \boldsymbol{V}_{j,i_{d'}} \\
&= \sum_{j_d} (\boldsymbol{A} \times_{d'} \boldsymbol{V})_{i_1,\ldots,i_{d-1},j_d,i_{d+1},\ldots,i_{d'-1},j,i_{d'+1},i_n} \boldsymbol{U}_{i_d,j_d} \\
&= \big((\boldsymbol{A} \times_{d'} \boldsymbol{V}) \times_d \boldsymbol{U}\big)_{i_1,\ldots,i_{d'-1},j,i_{d'+1},\ldots,i_q}.
\end{aligned}$$

A different property holds when $d = d'$, namely,

$$(\boldsymbol{A} \times_d \boldsymbol{U}) \times_d \boldsymbol{V} = \boldsymbol{A} \times_d (\boldsymbol{V}\boldsymbol{U}). \tag{2.2}$$

Again, this is clear from direct computation, since

$$\begin{aligned}
\big(\boldsymbol{A} \times_d (\boldsymbol{V}\boldsymbol{U})\big)_{i_1,\ldots,i_{d-1},j,i_{d+1},\ldots,i_q} &= \sum_{i_n} \boldsymbol{A}_{i_1,\ldots,i_q} (\boldsymbol{V}\boldsymbol{U})_{j,i_d} \\
&= \sum_{i_d} \boldsymbol{A}_{i_1,\ldots,i_q} \sum_k \boldsymbol{V}_{j,k} \boldsymbol{U}_{k,i_d} \\
&= \sum_k (\boldsymbol{A} \times_d \boldsymbol{U})_{i_1,\ldots,i_{d-1},k,i_{d+1},\ldots,i_q} \boldsymbol{V}_{j,k} \\
&= \big((\boldsymbol{A} \times_d \boldsymbol{U}) \times_d \boldsymbol{V}\big)_{i_1,\ldots,i_{d-1},j,i_{d+1},\ldots,i_q}.
\end{aligned}$$

2.2 Tensor Decompositions

2.2.1 CP and Tucker Decompositions

Tensor decompositions allow for a low-dimensional representation of a large q-dimensional tensor by assuming some sort of underlying structure—typically that the tensor can be written as some finite linear combination of the outer product of q vectors. In a regression setting, decomposing tensor effects (either exactly or approximately) allows tensor observations to be practically included as model covariates.

In this section, we explore two well-known tensor decompositions, namely, the *CP* and *Tucker decompositions*. The former was used by Zhou et al. [97] and Li et al. [49] to develop the CP and Tucker tensor GLMs, respectively. We focus on properties of the Tucker decomposition throughout this thesis as it is a flexible generalization of the CP decomposition.

The CP decomposition was first proposed by Hitchcock [35] as the *polyadic form* of a tensor. Historically, this decomposition has been introduced multiple times into the literature, also under the names *parallel factors decomposition* (PARAFAC) and *canonical decomposition* (CANDECOMP), hence the name CP (CANDECOMP/PARAFAC). A CP decomposition of a tensor $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$ takes the form

$$\boldsymbol{A} \triangleq \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \cdots \circ \boldsymbol{a}_r^{(q)}$$

for some R and $\boldsymbol{a}_r^{(d)} \in \mathbb{R}^{I_d}$. Where desireable, unit norms can be enforced on the $\boldsymbol{a}_r^{(d)}$ and the the CP decomposition becomes

$$\boldsymbol{A} = \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r^{(1)} \circ \cdots \circ \boldsymbol{a}_r^{(q)}.$$

We say that $\boldsymbol{A}$ has a *rank-R_0* CP decomposition if an exact CP decomposition is possible for $R = R_0$ but not for any $R < R_0$. For a full overview of tensor rank (including other types of tensor rank) and correponding theorems, see Kolda and Bader [46].

The Tucker decomposition was initially proposed for 3-dimensional tensors by Tucker [80] as *three-mode factor analysis* and later extended to general dimensions as *N-mode principal components analysis* or *higher-order singular value decomposition* (HOSVD). An (exact or approximate) Tucker decomposition of $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$ has form

$$\boldsymbol{A} \triangleq \sum_{r_1=1}^{R_1} \cdots \sum_{r_q=1}^{R_q} \lambda_{r_1,\ldots,r_q} \boldsymbol{a}_{r_1}^{(1)} \circ \cdots \circ \boldsymbol{a}_{r_q}^{(q)}$$

for some *(Tucker) decomposition ranks* $R_1, \ldots, R_q$ typically much smaller than $I_1, \ldots, I_q$, and $\boldsymbol{a}_{r_d}^{(d)} \in \mathbb{R}^{I_d}$. The tensor $\boldsymbol{\Lambda} \triangleq (\lambda_{r_1,\ldots,r_q})_{r_1,\ldots,r_q} \in \mathbb{R}^{R_1 \times \cdots \times R_q}$ is called

the *core tensor*, while the matrix $\boldsymbol{A}^{(d)} \triangleq [\boldsymbol{a}_1^{(d)}|\ldots|\boldsymbol{a}_{R_d}^{(d)}] \in \mathbb{R}^{I_d \times R_d}$ is called the *d-th factor matrix*, defined for $d = 1, \ldots, q$. Following this notation, we hereafter abbreviate the Tucker tensor decomposition as

$$[[\boldsymbol{\Lambda} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]] \triangleq \sum_{r_1=1}^{R_1} \cdots \sum_{r_q=1}^{R_q} \lambda_{r_1,\ldots,r_q} \boldsymbol{a}_{r_1}^{(1)} \circ \cdots \circ \boldsymbol{a}_{r_q}^{(q)}.$$

It is easy to see that the CP decomposition is a special case of the Tucker decomposition, namely, when $R_1 = \cdots = R_q = R$ and the core tensor is super-diagonal (i.e., $\lambda_{r_1,\ldots,r_q} = 0$ whenever $r_1, \ldots, r_q$ are not all equal). The Tucker decomposition is thus a more flexible, albeit more complex, low-dimensional representation of a tensor relative to the CP decomposition. This difference in dimension is clearer when (naively) comparing the number of elements in each decomposition. From the original tensor $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$ with $\prod_{d=1}^{q} I_q$ entries, a CP decomposition has $R \sum_{d=1}^{q} I_d$ entries while a Tucker decomposition has $\prod_{d=1}^{q} R_d + \sum_{d=1}^{q} R_d I_d$. Better would be to compare the number of effective parameters in these decompositions. We address this when discussing decomposition uniqueness in the following subsection.

Despite differences in (effective) number of parameters, both decompositions enforce a structure on the original tensor that, as we will see, can be exploited for dimension reduction when fitting a regression model. On the other hand, the Tucker decomposition is flexible, both by permitting greater interaction among factor matrices and by allowing different ranks $R_1, \ldots, R_q$ in each dimension. The latter property allows for more (or less) model complexity where (not) needed and can result in a more parsimonious model with even fewer parameters than the CP decomposition, which requires equal rank along each dimension. As a result, the Tucker decomposition might be preferred in practical application (such as neuroimaging), where sample size (number of patients) is typically limited.

2.2.2 Tucker Decomposition Properties

This subsection explores a number of technical properties regarding the Tucker decomposition that will become useful later on.

First, we establish a compact notation for the Tucker decomposition using the mode-d product convenient for computation, namely,

$$[[\boldsymbol{\Lambda} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]] = \boldsymbol{\Lambda} \times_1 \boldsymbol{A}^{(1)} \times_2 \cdots \times_q \boldsymbol{A}^{(q)}. \tag{2.3}$$

This follows by straightforward computation since

$$\begin{aligned}
&(\boldsymbol{\Lambda} \times_1 \boldsymbol{A}^{(1)} \times_2 \cdots \times_q \boldsymbol{A}^{(q)})_{i_1,\ldots,i_q} \\
&\qquad = \big((\boldsymbol{\Lambda} \times_1 \boldsymbol{A}^{(1)} \times_2 \cdots \times_{q-1} \boldsymbol{A}^{(q-1)}) \times_q \boldsymbol{A}^{(q)}\big)_{i_1,\ldots,i_q} \\
&\qquad = \sum_{r_q=1}^{R_q} (\boldsymbol{\Lambda} \times_1 \boldsymbol{A}^{(1)} \times_2 \cdots \times_{q-1} \boldsymbol{A}^{(q-1)})_{i_1,\ldots,i_{q-1},r_q} (\boldsymbol{A}^{(q)})_{i_q,r_q} \\
&\qquad = \sum_{r_q=1}^{R_q} \Big(\sum_{r_{q-1}=1}^{R_{q-1}} (\boldsymbol{\Lambda} \times_1 \boldsymbol{A}^{(1)} \times_2 \cdots \times_{q-2} \boldsymbol{A}^{(q-2)})_{i_1,\ldots,r_{q-1},r_q} (\boldsymbol{A}^{(q-1)})_{i_{q-1},r_{q-1}} \Big) (\boldsymbol{A}^{(q)})_{i_q,r_q} \\
\ldots &= \sum_{r_q=1}^{R_q} \cdots \sum_{r_1=1}^{R_1} \boldsymbol{\Lambda}_{r_1,\ldots,r_q} \prod_{d=1}^{q} (\boldsymbol{A}^{(d)})_{i_d,r_d} \\
&\qquad = \sum_{r_q=1}^{R_q} \cdots \sum_{r_1=1}^{R_1} \lambda_{r_1,\ldots,r_q} (\boldsymbol{a}_{r_1}^{(1)} \circ \cdots \circ \boldsymbol{a}_{r_q}^{(q)})_{i_1,\ldots,i_q} \\
&\qquad = \Big(\sum_{r_1=1}^{R_1} \cdots \sum_{r_q=1}^{R_q} \lambda_{r_1,\ldots,r_q} \boldsymbol{a}_{r_1}^{(1)} \circ \cdots \circ \boldsymbol{a}_{r_q}^{(q)} \Big)_{i_1,\ldots,i_q} \\
&\qquad = [[\boldsymbol{\Gamma} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]]_{i_1,\ldots,i_q}.
\end{aligned}$$

Second, we establish some facts regarding the mode-d matricization of a Tucker decomposition that will be critical to model estimation procedures later on.

Lemma 2 (Tucker mode-d matricization)**.** *Suppose that a tensor* $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$ *has Tucker decomposition* $[[\boldsymbol{\Lambda} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]]$*. Then*

$$\boldsymbol{A}_{[[d]]} = \boldsymbol{A}^{(d)} \boldsymbol{\Lambda}_{[[d]]} (\boldsymbol{A}^{(q)} \otimes \cdots \otimes \boldsymbol{A}^{(d+1)} \otimes \boldsymbol{A}^{(d-1)} \otimes \cdots \otimes \boldsymbol{A}^{(1)})^\top.$$

Proof. It is sufficient to show that, for every $i_1, \ldots, i_q$,

$$[[\boldsymbol{\Lambda} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]]_{i_1,\ldots,i_q} = \big(\boldsymbol{A}^{(d)} \boldsymbol{\Lambda}_{[[d]]} (\boldsymbol{A}^{(q)} \otimes \cdots \otimes \boldsymbol{A}^{(d+1)} \otimes \boldsymbol{A}^{(d-1)} \otimes \cdots \otimes \boldsymbol{A}^{(1)})^\top\big)_{i_d,i'},$$

where

$$i' = 1 + \sum_{d' \neq d} (i_{d'} - 1) \prod_{\substack{d'' < d' \\ d'' \neq d}} p_{d''}.$$

For notational convenience, define

$$\boldsymbol{A}^{(q)} \overline{\otimes} \cdots \overline{\otimes} \boldsymbol{A}^{(1)} \triangleq \boldsymbol{A}^{(q)} \otimes \cdots \otimes \boldsymbol{A}^{(d+1)} \otimes \boldsymbol{A}^{(d-1)} \otimes \cdots \otimes \boldsymbol{A}^{(1)}$$

On the left-hand side, by definition of the Tucker decomposition,

$$[[\boldsymbol{\Lambda} \mid \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]]_{i_1,\ldots,i_q} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_q=1}^{R_q} \lambda_{r_1,\ldots,r_q} \prod_{j=1}^{q} (\boldsymbol{a}_{r_j}^{(j)})_{i_j}. \tag{2.4}$$

On the right-hand side, denoting $\boldsymbol{B}_d \triangleq (\boldsymbol{A}^{(q)} \overline{\otimes} \cdots \overline{\otimes} \boldsymbol{A}^{(1)})^\top$, we have that

$$\begin{aligned}
\left(\boldsymbol{A}^{(d)} \boldsymbol{\Lambda}_{[[d]]} (\boldsymbol{A}^{(q)} \overline{\otimes} \cdots \overline{\otimes} \boldsymbol{A}^{(1)})^\top\right)_{i_d, i'} &= (\boldsymbol{A}^{(d)} \boldsymbol{\Lambda}_{[[d]]} \boldsymbol{B}_d)_{i_d, i'} \\
&= \sum_{j_d=1}^{R_d} (\boldsymbol{a}_{j_d}^{(d)})_{i_d} (\boldsymbol{\Lambda}_{[[d]]} \boldsymbol{B}_d)_{j_d, i'} \\
&= \sum_{j_d=1}^{R_d} (\boldsymbol{a}_{j_d}^{(d)})_{i_d} \sum_{j'=1}^{J_d{}'} (\boldsymbol{\Lambda}_{[[d]]})_{j_d, j'} (\boldsymbol{B}_d)_{j', i'},
\end{aligned} \tag{2.5}$$

where $J_d{}' = \prod_{j=1, j\neq d}^{q} R_j$. For a fixed j', we can find unique values of $j_1, \overline{\ldots}, j_q$ such that

$$j' = 1 + \sum_{\substack{d'=1 \\ d'\neq d}}^{q} (j_{d'} - 1) \prod_{\substack{d''<d' \\ d'\neq d}} R_{d''}$$

so that $(\boldsymbol{\Lambda}_{[[d]]})_{j_d, j'} = \lambda_{j_1,\ldots,j_q}$. From the same j', appealing to the definition of the Kronecker product, we have that

$$(\boldsymbol{B}_d)_{j', i'} = (\boldsymbol{A}^{(q)} \overline{\otimes} \cdots \overline{\otimes} \boldsymbol{A}^{(1)})_{i', j'} = \prod_{\substack{d'=1 \\ d'\neq d}}^{q} (\boldsymbol{A}^{(d')})_{i_{d'}, j_{d'}} = \prod_{\substack{d'=1 \\ d'\neq d}}^{q} (\boldsymbol{a}_{j_{d'}}^{(d')})_{i_{d'}}.$$

With this result, noting that the sum over j' in Equation 2.5 involves all elements of $\boldsymbol{\Lambda}$ with fixed d-th index j_d, Equation 2.5 can be rewritten as

$$\sum_{j_d=1}^{R_d} (\boldsymbol{a}_{j_d}^{(d)})_{i_d} \sum_{j_1, \overline{\ldots}, j_q}^{R_1, \overline{\ldots}, R_q} \lambda_{j_1,\ldots,j_q} \prod_{\substack{d'=1 \\ d'\neq d}}^{q} (\boldsymbol{a}_{j_{d'}}^{(d')})_{i_{d'}} = \sum_{j_1=1}^{R_1} \cdots \sum_{j_q=1}^{R_q} \lambda_{j_1,\ldots,j_q} \prod_{d'=1}^{q} (\boldsymbol{a}_{j_{d'}}^{(d')})_{i_{d'}},$$

which equals the left-hand side by Equation 2.4, thus completing the proof. □

Using the above lemma, a similar result for the vectorization of a Tucker decomposition follows. We rely on the trivial fact that the vectorization of the mode-1 matricization of a tensor $\boldsymbol{A}$ is equal the vectorization of $\boldsymbol{A}$, that is, $\operatorname{vec} \boldsymbol{A}_{[[1]]} = \operatorname{vec} \boldsymbol{A}$.

Lemma 3 (Tucker vectorization)**.** *Suppose that a tensor* $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$ *has Tucker decomposition* $[[\boldsymbol{\Lambda} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]]$*. Then*

$$\operatorname{vec} \boldsymbol{A} = (\boldsymbol{A}^{(q)} \otimes \cdots \otimes \boldsymbol{A}^{(1)}) \operatorname{vec} \boldsymbol{\Lambda}.$$

Proof. The proof follows easily with the results of Lemma 2 and the well-known relationship between vectorization and the Kronecker product that, for conformable matrices $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$, $\operatorname{vec}(\boldsymbol{XYZ}) = (\boldsymbol{Z}^\top \otimes \boldsymbol{X}) \operatorname{vec} \boldsymbol{Y}$. We have that

$$\begin{aligned} \operatorname{vec} \boldsymbol{A} = \operatorname{vec} \boldsymbol{A}_{[[1]]} &= \operatorname{vec}\left(\boldsymbol{A}^{(1)} \boldsymbol{\Lambda}_{[[1]]} (\boldsymbol{A}^{(q)} \otimes \cdots \otimes \boldsymbol{A}^{(2)})^\top\right) \\ &= (\boldsymbol{A}^{(q)} \otimes \cdots \otimes \boldsymbol{A}^{(1)}) \operatorname{vec} \boldsymbol{\Lambda}_{[[1]]} \\ &= (\boldsymbol{A}^{(q)} \otimes \cdots \otimes \boldsymbol{A}^{(1)}) \operatorname{vec} \boldsymbol{\Lambda}, \end{aligned}$$

completing the proof. □

2.2.3 Tucker Decomposition Uniqueness

A natural question regarding any tensor decomposition is whether such a representation is unique. This is not the case for either the CP or Tucker decompositions. For the latter, we have already shown in Equation 2.3 that the Tucker decomposition can be written as successive mode-d tensor products applied to the core tensor. Using the properties established in Equations 2.1 and 2.2, it follows immediately that, for any nonsingular matrix $\boldsymbol{O}_d \in \mathbb{R}^{R_d \times R_d}$,

$$\begin{aligned} [[\boldsymbol{\Lambda} \times_d \boldsymbol{O}_d^{-1} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(d)} \boldsymbol{O}_d, \ldots, \boldsymbol{A}^{(q)}]] &= (\boldsymbol{\Lambda} \boldsymbol{O}_d^{-1}) \times_1 \boldsymbol{A}^{(1)} \cdots \times_d (\boldsymbol{A}^{(d)} \boldsymbol{O}_d) \cdots \times_q \boldsymbol{A}^{(q)} \\ &= \boldsymbol{\Lambda} \times_1 \boldsymbol{A}^{(1)} \cdots \times_d (\boldsymbol{A}^{(d)} \boldsymbol{O}_d \boldsymbol{O}_d^{-1}) \cdots \times_q \boldsymbol{A}^{(q)} \\ &= [[\boldsymbol{\Lambda} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]]. \end{aligned}$$

The above example illustrates the more general *nonsingular transformation indeterminacy* for the Tucker decomposition,

$$[[\boldsymbol{\Lambda} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]] = [[\boldsymbol{\Lambda} \times_1 \boldsymbol{O}_1^{-1} \cdots \times_q \boldsymbol{O}_q^{-1} \,|\, \boldsymbol{A}^{(1)} \boldsymbol{O}_1, \ldots, \ldots, \boldsymbol{A}^{(q)} \boldsymbol{O}_q]],$$

where $\boldsymbol{O}_d$ is as defined above for $d = 1, \ldots, q$. In particular, this indeterminacy includes (non-zero) scaling and permutations of factor matrix columns.

One (of infinitely many) ways to resolve this non-uniqueness is to impose a restriction on the Tucker decomposition requiring, for $d = 1, \ldots, q$, the $(R_d)^2$ entries in the first R_d rows of $\boldsymbol{A}^{(d)}$ to be equal to one, ensuring that $\boldsymbol{O}_d$ is uniquely determined. As a result, the number of effective parameters in the Tucker decomposition is

$$p_q^{\text{eff}} \triangleq \prod_{d=1}^{q} R_d + \sum_{d=1}^{q} R_d I_d - \sum_{d=1}^{q} (R_d)^2. \tag{2.6}$$

For a similar examination of the CP decomposition, see Kolda and Bader [46].

The choice of restriction (to determine the $\boldsymbol{O}_d$ matrices) is arbitrary, although certain restrictions may be more or less appropriate depending on context and will affect log-likelihood gradients and Hessian matrices. We use the above set of restrictions where necessary in later chapters and define the resulting restricted parameter space as

$$\mathcal{G}_q \triangleq \{(\boldsymbol{\Gamma}, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}) \,|\, (\boldsymbol{a}_r^{(d)})_i = 1 \text{ for } i \leq R_d, d = 1, \ldots, q\}. \tag{2.7}$$

Topological equivalence of $\mathcal{G}_q$ and $\mathbb{R}^{p_q^{\text{eff}}}$ is clear.

Throughout this thesis, we say that two (or more) Tucker decompositions $[[\boldsymbol{\Lambda} \,|\, \boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(q)}]]$ and $[[\boldsymbol{\Pi} \,|\, \boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(q)}]]$ are *factor equivalent* if there exists nonsingular $\boldsymbol{O}_d \in \mathbb{R}^{R_d \times R_d}$ such that $\boldsymbol{A}^{(d)} = \boldsymbol{B}^{(d)}\boldsymbol{O}_d$ for $d = 1, \ldots, q$. This property will be useful in simplifying the qD-FLQR model later in certain circumstances.

Chapter 3

q-Dimensional Functional Partial Linear Quantile Regression

In this chapter, we formulate the basic q-dimensional functional linear quantile regression (qD-FLQR) model. We specifically consider the setting where the functional covariate is observed over a uniform grid of points in $[0,1]^q$ and we translate the qD-FLQR model into a q-dimensional tensor quantile regression (qD-TQR) model. In Section 3.2, we discuss an implementation of the model and propose an algorithm for estimating model parameters. In particular, we use a supervised Tucker decomposition to obtain a "partial" decomposition of tensor effects. Lastly, Section 3.3 extends the qD-FLQR model to include multiple functional covariates not necessarily of the same size and, furthermore, the capacity for tensor effect regularization.

3.1 qD-FLQR Model Formulation

We first consider the qD-FLQR model with a single functional covariate. For a fixed quantile level $\tau \in (0,1)$, we model the τ-th conditional quantile of a scalar response Y given scalar covariates $\boldsymbol{x} \in \mathbb{R}^p$ and q-dimensional functional covariate $z : [0,1]^q \to \mathbb{R}$ as

$$Q_\tau(Y|\boldsymbol{x}, z) = \alpha_\tau + \boldsymbol{x}^\top \boldsymbol{\beta}_\tau + \int_0^1 \cdots \int_0^1 z(t_1, \ldots, t_q)\gamma_\tau(t_1, \ldots, t_q)dt_q \ldots dt_1. \quad (3.1)$$

From here on, we suppress the subscript τ for convenience and assume τ fixed and known.

In many applications such as neuroimaging, functional covariate observations are made automatically using technology at regularly-spaced points (in time or space). With this setting in mind, we assume that functional data z is observed over a uniform grid in $[0,1]^q$. Formally, we only observe $z(t_{i_1},\ldots,t_{i_q})$ for $i_d = 1\ldots,I_d$ and $d = 1,\ldots,q$. As a result, we can store observations from a single realization of z in a q-dimensional tensor $\boldsymbol{Z} \in \mathbb{R}^{I_1\times\cdots\times I_q}$ with entries $\boldsymbol{Z}_{i_1,\ldots,i_q} = z(t_{i_1},\ldots,t_{i_q})$. We can similarly create a tensor holding evaluations of the functional effect γ as $\boldsymbol{\Gamma} \in \mathbb{R}^{I_1\times\cdots\times I_q}$ with entries $\boldsymbol{\Gamma}_{i_1,\ldots,i_q} = \gamma(t_{i_1},\ldots,t_{i_q})$.

Restructuring the data as a tensor in this way suggests the qD-TQR model

$$Q_\tau(Y|\boldsymbol{x},\boldsymbol{Z}) = \alpha + \langle \boldsymbol{x},\boldsymbol{\beta}\rangle + \langle \boldsymbol{Z},\boldsymbol{\Gamma}\rangle. \tag{3.2}$$

The last term in this model can be viewed as an approximation to the integral in the qD-FLQR model of Equation 3.1. However, Equation 3.2 ignores any scaling factors, here $(\prod_{d=1}^q I_d)^{-1}$. This is reasonable for computational reasons and due to our assumption of working over a uniform grid. As a general note, this approach does not impose any (e.g., smoothness) restrictions on the functional effect γ, which our present approach does not aim to estimate. Previous work by Yu et al. [91], for comparison, is similar in that it also does not estimate functional effects (or the partial basis elements ϕ_k) and instead only works with a matrix of evaluations of the ϕ_k on a discrete grid. In contrast, our work here will impose penalties able to enforce the smoothness of $\boldsymbol{\Gamma}$ (that may subsequently be viewed as a smoothness restriction on γ), while Yu et al. does not consider any such penalty.

Equation 3.2 seems to suggest a typical linear quantile regression model. However, the large number of parameters in the functional portion of the model is $\prod_{d=1}^q I_d$, yielding a computationally difficult and unacceptably high-dimensional regression problem. In addition, the vectorization operation in $\langle \boldsymbol{Z},\boldsymbol{\Gamma}\rangle = (\operatorname{vec}\boldsymbol{Z})^\top \operatorname{vec}\boldsymbol{\Gamma}$ ignores any structure assumed by a tensor decomposition and any patterns of spatiotemporal correlation in $\boldsymbol{Z}$ and $\boldsymbol{\Gamma}$ and by treating each element as an independent variable. As such, this first approach to to es-

timation comes at the cost of high computational complexity, reduced power and efficiency, and ignorance of spatial correlation structures that are typically present in imaging data. (In addressing the last of these, regularization is crucial and will be discussed later.)

To address this difficulty in model estimation, we use a structured, low-dimensional representation of $\mathbf{\Gamma}$. Assume that $\mathbf{\Gamma}$ has Tucker decomposition

$$\mathbf{\Gamma} = [[\mathbf{\Lambda} \,|\, \mathbf{\Gamma}^{(1)}, \ldots, \mathbf{\Gamma}^{(q)}]]$$

for some fixed $R_1, \ldots, R_q$, with the r-th column of $\mathbf{\Gamma}^{(d)}$ denoted by $\boldsymbol{\gamma}_r^{(d)}$. The qD-TQR model in Equation 3.2 becomes

$$Q_\tau(Y|\boldsymbol{x}, \boldsymbol{Z}) = \alpha + \langle \boldsymbol{x}, \boldsymbol{\beta} \rangle + \langle \boldsymbol{Z}, [[\mathbf{\Lambda} \,|\, \mathbf{\Gamma}^{(1)}, \ldots, \mathbf{\Gamma}^{(q)}]] \rangle.$$

To obtain this decomposition, we use a supervised approach that incorporates information from the response Y. This is unlike FPC or two-stage approaches that first estimate and fix the factor matrices $\mathbf{\Gamma}^{(d)}$ using only the observed $\boldsymbol{Z}$ and subsequently only estimate the core tensor $\mathbf{\Lambda}$ as model parameters [3], [15]. Specifically, the supervised Tucker decomposition is chosen to maximize partial quantile covariance with the response,

$$\underset{\mathbf{\Lambda}, \mathbf{\Gamma}^{(1)} \ldots, \mathbf{\Gamma}^{(q)}}{\arg\max} \operatorname{Cov}_\tau\big(Y, \langle \boldsymbol{Z}, [[\mathbf{\Lambda} \,|\, \mathbf{\Gamma}^{(1)}, \ldots, \mathbf{\Gamma}^{(q)}]] \rangle \,|\, \boldsymbol{X}\big). \tag{3.3}$$

Intuitively, the (approximate) Tucker decomposition of $\mathbf{\Gamma}$ is chosen to be maximally predictive of the level-τ quantile of the response Y. In the following section, we address the practical implementation of this supervised approach to fitting the qD-TQR model.

3.2 Implementation and Estimation

In the (size n) sample setting, we wish to solve the optimization problem

$$\underset{\alpha, \boldsymbol{\beta}, \mathbf{\Lambda}, \mathbf{\Gamma}^{(1)}, \ldots, \mathbf{\Gamma}^{(q)}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau\big(y_i - \alpha - \boldsymbol{x}_i^\top \boldsymbol{\beta} - \langle \boldsymbol{Z}_i, [[\mathbf{\Lambda} \,|\, \mathbf{\Gamma}^{(1)}, \ldots, \mathbf{\Gamma}^{(q)}]] \rangle\big). \tag{3.4}$$

Two major complications present themselves. We denote the objective function in Equation 3.4 by $l(\alpha, \boldsymbol{\beta}, \mathbf{\Gamma}^{(1)}, \ldots, \mathbf{\Gamma}^{(q)}, \mathbf{\Lambda})$.

The objective function l is clearly not convex in all of $\alpha, \boldsymbol{\beta}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}$. Furthermore, the quantile loss ρ_τ is not differentiable. The latter precludes many common optimization algorithms such as gradient descent. As discussed in Yu et al. [91], it is difficult to establish statistical properties of estimators in this setting. Noting that the objective function above is convex in each individual block of parameters $(\alpha, \boldsymbol{\beta}), (\boldsymbol{\Lambda}), (\boldsymbol{\Gamma}^{(1)}), \ldots, (\boldsymbol{\Gamma}^{(q)})$, however, we can apply block relaxation [47] to break down the problem and can use a continuously differentiable approximation of ρ_τ [12], [55].

3.2.1 Smoothing the Quantile Loss

The quantile loss function ρ_τ, shown in Figure 3.1, is not differentiable at zero, preventing the use of common optimization techniques such as gradient descent or Newton's method. Indeed, ρ'_τ is piecewise constant. One approach taken in the literature to circumvent this is to replace ρ_τ with a convex, continuously differentiable approximation $\rho_{\tau\nu}$, where ν is some nuisance parameter controlling the smoothness of the approximation. This approach allows gradient-based methods that have the advantage of being computationally inexpensive and readily extended to include (smooth) penalization terms.

For median regression (i.e., when $\tau = 0.5$), the Huber function [39],

$$H_\nu(u) \triangleq \begin{cases} \frac{u^2}{2\nu}, & |u| \leq \nu \\ |u| - \nu/2, & |u| > \nu \end{cases},$$

has been previously applied in the literature [13], [51]. Chen [12] proposed an extension to any arbitrary quantile level $\tau \in (0, 1)$ with the *generalized Huber function*,

$$H_{\tau\nu}(u) \triangleq \begin{cases} u(\tau - 1) - \frac{1}{2}(\tau - 1)^2\nu, & u < (\tau - 1)\nu \\ \frac{u^2}{2\nu}, & (\tau - 1)\nu \leq u < \tau\nu \\ u\tau - \frac{1}{2}\tau^2\nu, & u \geq \tau\nu \end{cases}. \tag{3.5}$$

Muggeo et al. [55] more recently used an approach based on iteratively weighted

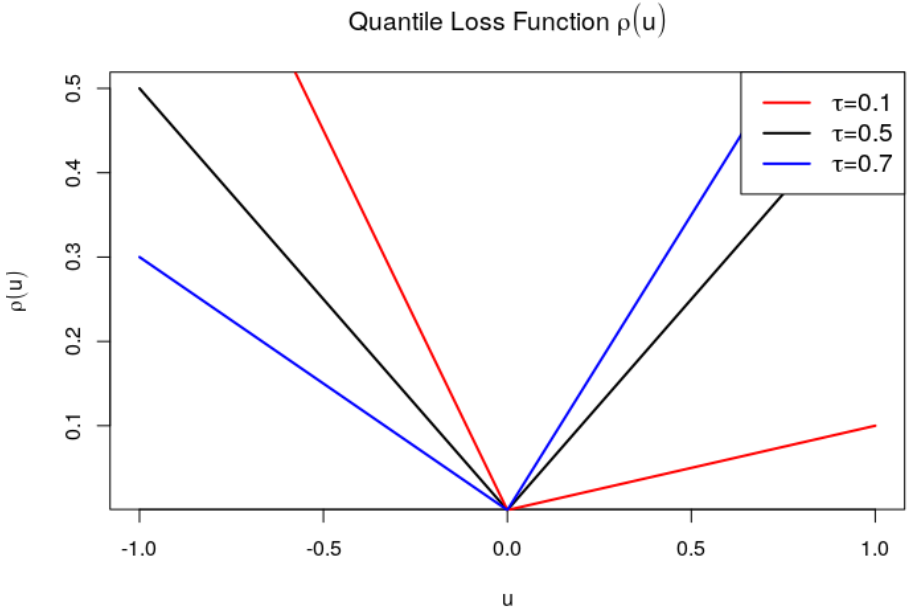

Figure 3.1: The quantile loss function ρ_τ for $\tau = 0.1, 0.5, 0.7$, coloured as per the legend.

least squares (IWLS) using the *IWLS approximator*,

$$S_{\tau\nu}(u) \triangleq \begin{cases} u(\tau - 1), & u \leq -\nu\tau \\ \frac{u^2(1-\tau)}{2\nu\tau} + \frac{\nu\tau(1-\tau)}{2}, & -\tau\nu < u \leq 0, \\ \frac{u^2\tau}{2\nu(1-\tau)} + \frac{\nu\tau(1-\tau)}{2}, & 0 < u < (1-\tau)\nu, \\ u\tau, & u \geq (1-\tau)\nu \end{cases}. \tag{3.6}$$

The quantile loss function and the two approximations above (and their derivatives) are shown in Figure 3.2. Muggeo et al. claim that $S_{\tau\nu}$ is superior to $H_{\tau\nu}$ by more closely approximating ρ'_τ for $u > 0$ ($u < 0$) when $\tau < 0.5$ ($\tau > 0.5$) where the derivative is "more important", and that the choice of ν in $S_{\tau\nu}$ has less impact on estimates than in $H_{\tau\nu}$. The authors don't provide an empirical comparison of the two claims, however.

Both Chen [12] and Muggeo et al. [55] derive regularity conditions similar to (but not following directly from) those presented in Theorem 3.3 of Koenker and Bassett [45]. Making use of a solution curve method, these conditions guarantee that for smoothing parameter $\nu > 0$ sufficiently close to zero, the solutions obtained by minimizing the smoothed loss is exactly the solution to the original, unsmoothed problem (i.e., the minimizer of l), and

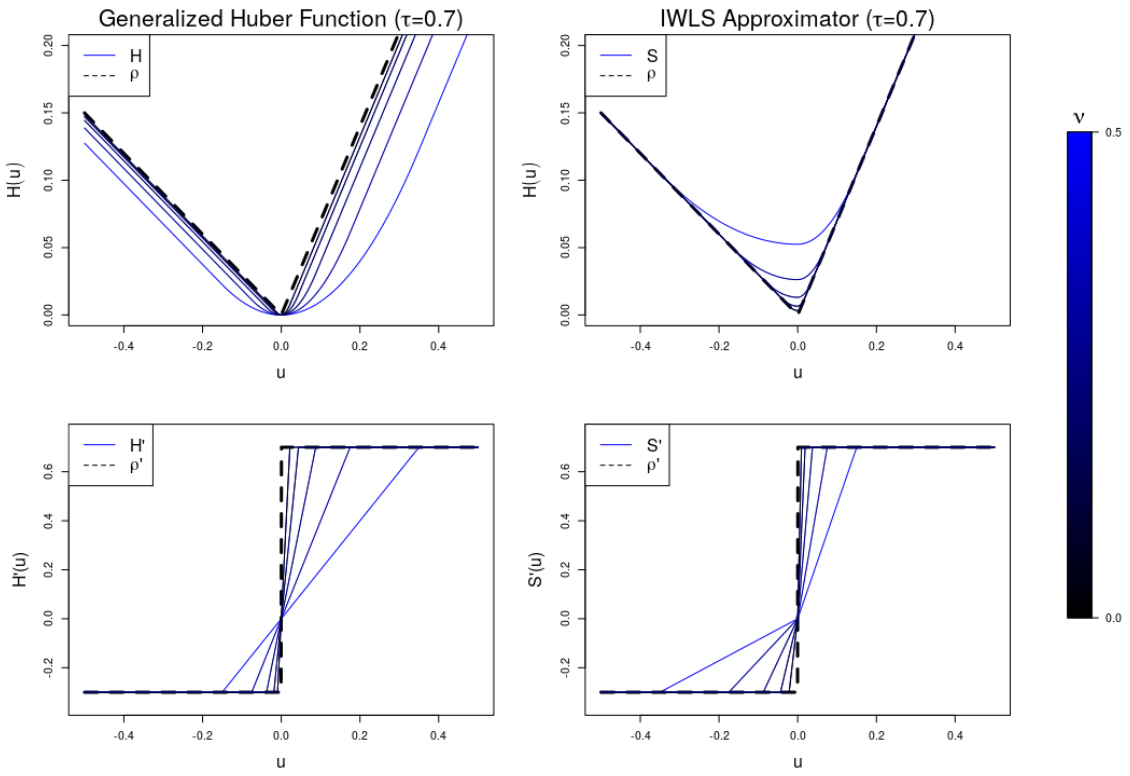

Figure 3.2: The generalized Huber function $H_{\tau\nu}$ (left) and iteratively weighted least squares (IWLS) approximator $S_{\tau\nu}$ (right) to the quantile loss ρ_τ (shown with dotted lines) and their derivatives (bottom). Both approximations are convex, continuously differentiable, and converge uniformly to ρ_τ as $\nu \to 0^+$.

that this solution interpolates some of the data. Both authors further propose an algorithm for updating (i.e., decreasing) ν with successive iterations. These updates generate a new smoothing parameter ν_{N+1} as a function of ν_N, current model estimates θ_N, and observed data.

We prefer to take $\rho_{\tau\nu} = H_{\tau\nu}$ in this thesis since it has ρ_τ as an upper bound. This property will be convenient later on in Subsection 4.2.3, but is not critical provided that the chosen approximation has a suitable envelope. Further motivating the choice of $H_{\tau\nu}$ is our focus on exploring regularized estimates: Chen gives heuristic guidelines for generic updates of ν_N dependent on dataset size but independent of θ_N. While increasing the number of iterations required, this update scheme fits well into our gradient-based implementation.

As an important point, note that both of these approximations converge uniformly to ρ_τ as $v \to 0^+$. We show in Subsection 4.1.3 that, as a result, the

solution to the optimization problem in 3.4 when ρ_τ is replaced by $\rho_{\tau\nu}$ will converge to the solution of the original problem in 3.4.

3.2.2 Block Relaxation

Block relaxation algorithms are a general class of optimization techniques that subsume the well-known majorization-minimization (MM), expectation-minimization (EM), and alternating least squares algorithms [47]. So-called *cyclic* examples allow the optimization of a scalar function ψ over several blocks of parameters $\omega_j \in \Omega_j$ for $j = 1, \ldots, p$ by repeating a cycle of p sequential block updates of the form

$$\omega_j^{k+1} = \underset{\omega_j \in \Omega_p}{\arg\min}\, \psi(\omega_1^{k+1}, \ldots, \omega_{j-1}^{k+1}, \omega_j, \omega_{j+1}^{k}, \ldots, \omega_{p-1}, \omega_p^{k}),$$

where superscript k denotes an estimate in the k-th update cycle. (On the other hand, *free-steering* variants do not specify a fixed order for block updates.) In general, parameter blocks are updated sequentially, holding other parameter blocks constant at their most current estimate. This technique comes with well-established global and local convergence results (under certain conditions).

In the qD-TQR setting, block relaxation proves extremely useful. The objective function $\psi = l$ is clearly not convex in all its parameters due to the Tucker decomposition. However, if we consider the parameter blocks $\omega_1 = (\alpha, \boldsymbol{\beta}), \omega_2 = (\boldsymbol{\Lambda}), \omega_3 = (\boldsymbol{\Gamma}^{(1)}), \ldots, \omega_{q+2} = (\boldsymbol{\Gamma}^{(q)})$, it is easy to see that the error

$$\eta \triangleq y - \alpha - \boldsymbol{x}^\top \boldsymbol{\beta} - \langle \boldsymbol{\mathcal{Z}}, [[\boldsymbol{\Lambda} \,|\, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}]] \rangle$$

is an affine linear combination (and thus convex) with respect to each block individually. Thus, $\rho_\tau(\eta)$ is *blockwise convex* as the composition of convex functions. The block updates are then convex optimization problems.

For notational simplicity, denote $\boldsymbol{\Gamma}^{\otimes} \triangleq \boldsymbol{\Gamma}^{(q)} \otimes \cdots \otimes \boldsymbol{\Gamma}^{(1)}$ and $\boldsymbol{\Gamma}_d^{\otimes} \triangleq \boldsymbol{\Gamma}^{(d)} \otimes \cdots \otimes \boldsymbol{\Gamma}^{(1)}$ from here on. The *scalar block* update for $(\alpha, \boldsymbol{\beta})$ is trivial as a standard linear quantile regression problem. The *factor block* updates for $(\boldsymbol{\Gamma}^{(d)})$ requires reformulation of the linear predictor. Lemma 2 allows us to write

$$\eta = y - \alpha - \boldsymbol{x}^\top \boldsymbol{\beta} - \langle \boldsymbol{\mathcal{Z}}_{[[d]]}, \boldsymbol{\Gamma}^{(d)} \boldsymbol{\Lambda}_{[[d]]} \boldsymbol{\Gamma}_d^{\otimes\top} \rangle$$

$$= y - \alpha - \boldsymbol{x}^\top \boldsymbol{\beta} - \langle \boldsymbol{Z}_{[[d]]} \boldsymbol{\Gamma}_d^{\otimes} \boldsymbol{\Lambda}_{[[d]]}^\top, \boldsymbol{\Gamma}^{(d)} \rangle, \tag{3.7}$$

where the second inequality follows from Lemma 1. Similarly for the *core block* update of ($\boldsymbol{\Lambda}$), we apply Lemma 3 to obtain

$$\begin{aligned} \eta &= y - \alpha - \boldsymbol{x}^\top \boldsymbol{\beta} - \langle \operatorname{vec} \boldsymbol{Z}, \boldsymbol{\Gamma}^{\otimes} \operatorname{vec} \boldsymbol{\Lambda} \rangle \\ &= y - \alpha - \boldsymbol{x}^\top \boldsymbol{\beta} - \langle {\boldsymbol{\Gamma}^{\otimes}}^\top \operatorname{vec} \boldsymbol{Z}, \operatorname{vec} \boldsymbol{\Lambda} \rangle. \end{aligned} \tag{3.8}$$

With this, each block update can be treated as a linear quantile regression problem in a relatively small number of parameters, specifically, $R_d I_d$ for factor matrix updates and $\prod_{d=1}^{q} R_d$ for core tensor updates (or fewer if estimates are restricted to $\mathcal{G}_q$).

3.2.3 Algorithm

Based on the previous two subsections, we propose Algorithm 1 to estimate the qD-FLQR model. We assume a preset rule for determining a decreasing sequence of positive smoothing parameters $(\nu_N)_N$, with ν_{N+1} possibly a function of ν_N and current model parameter estimates $\boldsymbol{\theta}_N$ that are generally unknown at initialization. We also assume preset Tucker decomposition ranks $R_1, \ldots, R_q$. Section 3.2.4 gives two criteria from existing literature to determine suitable decomposition ranks.

The inner loop in lines 3-8 of Algorithm 1 employs block relaxation to estimate model parameters using smoothed quantile loss $\rho_{\tau\nu_N}$ for a fixed smoothing parameter ν_N. We give a stopping criteria based on absolute change in the loss l_N for simplicity, although in practice we implement convergence criteria based on relative loss with tolerance 0.1%. The outer loop uses a given rule to decrease the smoothing parameter ν_N to ν_{N+1}, possibly as a function of current model residuals (as per Chen [12] and Muggeo et al. [55]). The convergence criteria might depend on the smoothing approximation used, so we leave this general.

We have implemented our algorithm entirely in R [63]. Because no package exists for solving the approximate quantile regression problem, we apply

Algorithm 1: qD-FLQR model estimation algorithm given fixed Tucker decomposition ranks $R_1, \ldots, R_q$, a single tensor covariate in $\mathbb{R}^{I_1 \times \cdots \times I_q}$, p scalar covariates, prespecified convergence tolerance ε, and a rule for generating a decreasing sequence of positive smoothing parameters $(\nu_N)_N$ potentially as a function of model estimates at the time of update.

for $N = 1, \ldots, N_{max}$ **do**
  Initialize: $\alpha_{(0)} \in \mathbb{R}, \boldsymbol{\beta}_{(0)} \in \mathbb{R}^p, \boldsymbol{\Lambda}_{(0)} \in \mathbb{R}^{R_1 \times \cdots \times R_q}, \boldsymbol{\Gamma}_{(0)}^{(d)} \in \mathbb{R}^{I_d \times R_d}$
  for $b = 1, 2, \ldots, b_{max}$ **do**
    $(\alpha_{(b)}, \boldsymbol{\beta}_{(b)}) = \arg\min_{\alpha, \boldsymbol{\beta}} l_N(\alpha, \boldsymbol{\beta}, \boldsymbol{\Gamma}_{(b-1)}^{(1)}, \ldots, \boldsymbol{\Gamma}_{(b-1)}^{(q)}, \boldsymbol{\Lambda}_{(b-1)})$
    for $d = 1, \ldots, q$ **do**
      $\boldsymbol{\Gamma}_{(b)}^{(d)} =$
      $\arg\min_{\boldsymbol{\Gamma}^{(d)}} l_N(\alpha_{(b)}, \boldsymbol{\beta}_{(b)}, \boldsymbol{\Gamma}_{(b)}^{(1)}, \ldots, \boldsymbol{\Gamma}_{(b)}^{(d-1)}, \boldsymbol{\Gamma}^{(d)}, \boldsymbol{\Gamma}_{(b-1)}^{(d+1)}, \ldots, \boldsymbol{\Gamma}_{(b-1)}^{(q)}, \boldsymbol{\Lambda}_{(b-1)})$
    end
    $\boldsymbol{\Lambda}_{(b)} = \arg\min_{\boldsymbol{\Lambda}} l_N(\alpha_{(b)}, \boldsymbol{\beta}_{(b)}, \boldsymbol{\Gamma}_{(b)}^{(1)}, \ldots, \boldsymbol{\Gamma}_{(b)}^{(q)}, \boldsymbol{\Lambda})$
    if $l_N(\alpha_{(b-1)}, \boldsymbol{\beta}_{(b-1)}, \boldsymbol{\Gamma}_{(b-1)}^{(1)}, \ldots, \boldsymbol{\Gamma}_{(b-1)}^{(q)}, \boldsymbol{\Lambda}) -$
     $l_N(\alpha_{(b)}, \boldsymbol{\beta}_{(b)}, \boldsymbol{\Gamma}_{(b)}^{(1)}, \ldots, \boldsymbol{\Gamma}_{(b)}^{(q)}, \boldsymbol{\Lambda}_{(b)}) < \varepsilon$ **then**
      break;
  end
  Save $(\hat{\alpha}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Gamma}}^{(1)}, \ldots, \hat{\boldsymbol{\Gamma}}^{(q)}, \hat{\boldsymbol{\Lambda}}) = (\alpha_{(b)}, \boldsymbol{\beta}_{(b)}, \boldsymbol{\Gamma}_{(b)}^{(1)}, \ldots, \boldsymbol{\Gamma}_{(b)}^{(q)}, \boldsymbol{\Lambda}_{(b)})$
  if *convergent in* N, **then**
    break;
end
return $(\hat{\alpha}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Gamma}}^{(1)}, \ldots, \hat{\boldsymbol{\Gamma}}^{(q)}, \hat{\boldsymbol{\Lambda}})$

a gradient descent method with Barzilai-Borwein adaptive step size [5], which is known to perform well for high-dimensional problems. A gradient-based method is flexible and convenient for our goal of exploring different penalty functions. Our implementation is able to restrict estimated Tucker decompositions to $\mathcal{G}_q$. We do not consider second-order methods due to the lack of curvature in the loss l_N in each block update for large N. To ensure decreasing loss across block updates, we implement a line search that shrinks step size if necessary. Based on the heuristics given by Chen [12], we set $\nu_N = 2^{-(N+1)}$ and $N_{max} = 15$ and use a relative loss convergence criteria based on l.

3.2.4 Choice of Decomposition Ranks

The Tucker decomposition ranks $R_1, \ldots, R_q$ determine the number of basis elements along each dimension (i.e., model size/complexity) of the qD-FLQR model and are to be fixed before before estimation. As usual, these hyperparameters should be large enough to give the model the capacity to learn the true signal, but not too large as to induce overfitting. Several criteria, including BIC and cross-validation criteria, exist for other models in the literature [43], [50], [77].

A cross-validation (CV) criteria is given by

$$\text{CV} = \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - \alpha^{(-i)} - \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}^{(-i)} - \langle \boldsymbol{\mathcal{Z}}_i, \hat{\Gamma}^{(-i)} \rangle)$$

for a dataset of size n, where superscripts $(-i)$ denote an estimate obtained after removing the i-th observation from the data. This leave-one-out CV criteria is easily adjusted to obtain a K-fold CV criteria [33], which we use in our analyses later on.

3.3 Extensions

In this section, we explore two extensions of the previous qD-FLQR model. First, we discuss incorporating more than one functional covariate, either of the same or different shapes. Second, we modify the model-fitting algorithm to allow for penalty terms that regularize the tensor effect estimate $\hat{\boldsymbol{\Gamma}}$ through

the structure imposed by the Tucker decomposition. Existing literature for the Tucker tensor GLM generally only consider regularizing $\hat{\boldsymbol{\Lambda}}$.

3.3.1 Multiple Functional Covariates

In practice, we might wish to use more than one functional covariate as predictors in the qD-FLQR model. Although these functional covariates are typically observed over the same grid in practice, this need not be the case in general. In this subsection, we briefly discuss ways to accommodate both of these cases. For simplicity, we only consider cases with $s = 2$ functional covariates. Generalizations for $s > 2$ follow immediately.

First, consider the case with two functional covariates with observation tensors $\boldsymbol{\mathcal{Z}}^{(1)}, \boldsymbol{\mathcal{Z}}^{(2)} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$ and respective tensor effects $\boldsymbol{\Gamma} = [[\boldsymbol{\Lambda} \,|\, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}]]$ and $\boldsymbol{\Delta} = [[\boldsymbol{\Pi} \,|\, \boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(q)}]]$. We do not require the "times" $t_{i_1}, \ldots, t_{i_q}$ at which the functional covariates are observed to be the same between $\boldsymbol{\mathcal{Z}}^{(1)}$ and $\boldsymbol{\mathcal{Z}}^{(2)}$. Without any assumptions on the decomposition ranks for each effect, we can write the new loss function as

$$
\begin{aligned}
&l(\alpha, \boldsymbol{\beta}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}, \boldsymbol{\Lambda}, \boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(q)}, \boldsymbol{\Pi}) \\
&\qquad = \frac{1}{n} \sum_{i=1}^{n} \rho_\tau\big(y_i - \alpha - \boldsymbol{x}_i^\top \boldsymbol{\beta} - \langle \boldsymbol{\mathcal{Z}}_i^{(1)}, [[\boldsymbol{\Lambda} \,|\, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}]] \rangle - \langle \boldsymbol{\mathcal{Z}}_i^{(2)}, [[\boldsymbol{\Pi} \,|\, \boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(q)}]] \rangle\big).
\end{aligned}
$$

Noting that the new loss is blockwise convex with respect to the blocks $(\boldsymbol{\Gamma}^{(1)}, \boldsymbol{\Delta}^{(1)})$, $\ldots$, $(\boldsymbol{\Gamma}^{(q)}, \boldsymbol{\Delta}^{(q)})$, $(\boldsymbol{\Lambda}, \boldsymbol{\Pi})$, we may simply modify Algorithm 1 to update both d-th factor matrices together (on line 5) and both core tensors together (on line 8). This approach increases the dimensionality of each problem but, on the other hand, does not require separate treatment of the second functional covariate (as described at the end of this subsection).

Further simplification is possible if we exploit the Tucker decomposition's nonsingular transformation indeterminacy. Here, we additionally assume that both functional covariates have the same decomposition ranks $R_1, \ldots, R_q$ and that, for $d = 1, \ldots, q$, we can write $\boldsymbol{\Delta}^{(d)} = \boldsymbol{\Gamma}^{(d)} \boldsymbol{O}_d$ for some nonsingular matrix $\boldsymbol{O}_d \in \mathbb{R}^{R_d \times R_d}$. Recall that we referred to this as factor equivalence in

Subsection 2.2.3. The Tucker decomposition of $\boldsymbol{\Delta}$ then can be written as

$$\begin{aligned}[[\boldsymbol{\Pi} \,|\, \boldsymbol{\Delta}^{(1)}, \ldots, \boldsymbol{\Delta}^{(q)}]] &= [[\boldsymbol{\Pi} \times_1 \boldsymbol{O}_1^{-1} \cdots \times_q \boldsymbol{O}_q^{-1} \,|\, \boldsymbol{\Delta}^{(1)}\boldsymbol{O}_1^{(1)}, \ldots, \boldsymbol{\Delta}^{(q)}\boldsymbol{O}_q]] \\ &= [[\boldsymbol{\Pi} \times_1 \boldsymbol{O}_1^{-1} \cdots \times_q \boldsymbol{O}_q^{-1} \,|\, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}]].\end{aligned}$$

Defining $\tilde{\boldsymbol{\Lambda}} \in \mathbb{R}^{R_1 \times \cdots \times R_q \times 2}$ with $\tilde{\boldsymbol{\Lambda}}_{r_1,\ldots,r_q,1} \triangleq \boldsymbol{\Lambda}_{r_1,\ldots,r_q}$ and $\tilde{\boldsymbol{\Lambda}}_{r_1,\ldots,r_q,2} \triangleq (\boldsymbol{\Pi} \times_1 \boldsymbol{O}_1^{-1} \cdots \times_q \boldsymbol{O}_q^{-1})_{r_1,\ldots,r_q}$, we can write

$$\tilde{\boldsymbol{\Gamma}} = [[\tilde{\boldsymbol{\Lambda}} \,|\, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}, \mathbb{1}_{2\times 2}]],$$

which is the tensor effect corresponding to $\tilde{\boldsymbol{Z}} \in \mathbb{R}^{I_1,\ldots,I_q,2}$ with $\tilde{\boldsymbol{Z}}_{i_1,\ldots,i_q,1} \triangleq \boldsymbol{Z}^{(1)}_{i_1,\ldots i_q}$ and $\tilde{\boldsymbol{Z}}_{i_1,\ldots,i_q,2} \triangleq \boldsymbol{Z}^{(2)}_{i_1,\ldots i_q}$. In other words, we have replaced both q-dimensional observation tensors $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)}$ with a single $q+1$-dimensional tensor $\tilde{\boldsymbol{Z}}$ and need only estimate the parameters for a single Tucker decomposition. While computationally convenient, it is unclear how to verify the assumption that $\boldsymbol{\Delta}^{(d)} = \boldsymbol{\Gamma}^{(d)}\boldsymbol{O}_d$ in practice. We explore the effect of this assumption in Section 5.1 in cases where it (not) appropriate.

In the most general case, where $\boldsymbol{Z}^{(1)} \in \mathbb{R}^{I_1 \times \cdots \times I_{q_1}}$ and $\boldsymbol{Z}^{(2)} \in \mathbb{R}^{J_1 \times \cdots \times J_{q_2}}$, we can trivially modify Algorithm 1 to consider blocks $(\alpha, \boldsymbol{\beta})$, $(\boldsymbol{\Gamma}^{(1)})$, $\ldots$, $(\boldsymbol{\Gamma}^{(q_1)})$, $(\boldsymbol{\Lambda})$, $(\boldsymbol{\Delta}^{(1)})$, $\ldots$, $(\boldsymbol{\Delta}^{(q_2)})$, $(\boldsymbol{\Pi})$. A slight modification might use blocks $(\boldsymbol{\Gamma}^{(d)}, \boldsymbol{\Delta}^{(d)})$ for $d = 1, \ldots, \min\{q_1, q_2\}$ and $(\boldsymbol{\Lambda}, \boldsymbol{\Pi})$.

3.3.2 Model Regularization

Previous work has demonstrated the need to regularize tensor effect estimates in the tensor GLM setting, whether using the CP [97] or Tucker [49] tensor decompositions, due to noise present in estimated tensor effects [96]. Our initial simulation results in Chapter 5 suggest this need for the tensor quantile setting as well. In this subsection, we extend the qD-FLQR model to include a convex, differentiable penalty. Unlike previous work, we directly penalize tensor effect estimates through the low rank structure imposed by the Tucker decomposition in a way that fits seamlessly into our gradient-based block relaxation approach. The results in this subsection allow us to implement LASSO and fused LASSO penalties in the data analyses in Chapter 5. Although we do

not directly estimate or penalize the functional effect γ, penalties on $\mathbf{\Gamma}$ may be viewed as having the same effect, being a restriction of γ to a discrete domain.

Consider the general setting where a differentiable scalar penalty function J is defined as a function of $\mathbf{\Gamma}$. The qD-TQR loss becomes

$$l(\alpha, \boldsymbol{\beta}, \mathbf{\Gamma}^{(1)}, \ldots, \mathbf{\Gamma}^{(q)}, \mathbf{\Lambda}) = \frac{1}{n}\sum_{i=1}^{n} \rho_\tau\big(y_i - \alpha - \boldsymbol{x}_i^\top \boldsymbol{\beta} - \langle \boldsymbol{Z}_i, \mathbf{\Gamma}\rangle\big) + \lambda J(\mathbf{\Gamma}),$$

where $\lambda > 0$ is the usual fidelity-penalty tradeoff hyperparameter.

As a useful first result, we derive

$$\begin{aligned}\frac{\partial \mathrm{vec}\, \mathbf{\Gamma}_{[[d]]}}{\partial \mathrm{vec}\, \mathbf{\Gamma}^{(d)}} &= \frac{\partial}{\partial \mathrm{vec}\, \mathbf{\Gamma}^{(d)}} \mathrm{vec}\, (\mathbf{\Gamma}^{(d)} \mathbf{\Lambda}_{[[d]]} \mathbf{\Gamma}_d^{\otimes\top}) \\ &= \frac{\partial}{\partial \mathrm{vec}\, \mathbf{\Gamma}^{(d)}} [\boldsymbol{I}_{I_d} \otimes (\mathbf{\Gamma}_d^{\otimes} \mathbf{\Lambda}_{[[d]]}^\top)] \mathrm{vec}\, \mathbf{\Gamma}^{(d)} \\ &= [\boldsymbol{I}_{I_d} \otimes (\mathbf{\Gamma}_d^{\otimes} \mathbf{\Lambda}_{[[d]]}^\top)] \in \mathbb{R}^{\prod_j I_j \times I_d R_d}. \end{aligned} \tag{3.9}$$

Equality above follows by Lemma 2 and the well-known result that $\mathrm{vec}\,(\boldsymbol{XYZ}) = (\boldsymbol{X} \otimes \boldsymbol{Z}^\top)\mathrm{vec}\, \boldsymbol{Y}$. It follows by the chain rule that

$$\begin{aligned}\frac{\partial J}{\partial \mathrm{vec}\, \mathbf{\Gamma}^{(d)}} &= \frac{\partial J}{\partial \mathrm{vec}\, \mathbf{\Gamma}_{[[d]]}} \frac{\partial \mathrm{vec}\, \mathbf{\Gamma}_{[[d]]}}{\partial \mathrm{vec}\, \mathbf{\Gamma}^{(d)}} \\ &= \left(\mathrm{vec} \left(\frac{\partial J}{\partial \mathbf{\Gamma}}\right)_{[[d]]}\right)^\top [\boldsymbol{I}_{I_d} \otimes (\mathbf{\Gamma}_d^{\otimes} \mathbf{\Lambda}_{[[d]]}^\top)] \in \mathbb{R}^{1 \times I_d R_d},\end{aligned}$$

and so

$$\nabla_{\mathbf{\Gamma}^{(d)}} J = [\boldsymbol{I}_{I_d} \otimes (\mathbf{\Gamma}_d^{\otimes} \mathbf{\Lambda}_{[[d]]}^\top)]^\top \mathrm{vec} \left(\frac{\partial J}{\partial \mathbf{\Gamma}}\right)_{[[d]]} = \mathrm{vec} \left[\left(\frac{\partial J}{\partial \mathbf{\Gamma}}\right)_{[[d]]} \mathbf{\Gamma}_d^{\otimes} \mathbf{\Lambda}_{[[d]]}^\top\right].$$

Similarly applying the results of Lemma 3, we see that

$$\frac{\partial \mathrm{vec}\, \mathbf{\Gamma}}{\partial \mathrm{vec}\, \mathbf{\Lambda}} = \frac{\partial}{\partial \mathrm{vec}\, \mathbf{\Lambda}} [\mathbf{\Gamma}^{\otimes} \mathrm{vec}\, \mathbf{\Lambda}] = \mathbf{\Gamma}^{\otimes}, \tag{3.10}$$

so that

$$\nabla_{\mathbf{\Lambda}} J = \mathbf{\Gamma}^{\otimes\top} \mathrm{vec}\, \frac{\partial J}{\partial \mathbf{\Gamma}}.$$

It is often the case that J sums the elements of $\mathbf{\Gamma}$ after a scalar penalty function $\zeta : \mathbb{R} \to \mathbb{R}$ is applied elementwise to $\mathbf{\Gamma}$, namely,

$$J(\mathbf{\Gamma}) = J = \sum_{i_1=1}^{I_1} \cdots \sum_{i_q=1}^{I_q} \zeta(\mathbf{\Gamma}_{i_1,\ldots,i_q}).$$

The previous assumption on J translates to a requirement that ζ be differentiable. Let $\boldsymbol{\zeta}$ and $\boldsymbol{\zeta}'$ denote the elementwise application of ζ and ζ', respectively, to a vector or tensor.

This setting yields the convenient notation

$$\frac{\partial J}{\partial \mathbf{\Gamma}} = \boldsymbol{\zeta}'(\mathbf{\Gamma}),$$

and gives the simplified gradients

$$\nabla_{\mathbf{\Gamma}^{(d)}} J = \text{vec}\left[\boldsymbol{\zeta}'(\mathbf{\Gamma})_{[[d]]}\mathbf{\Gamma}_d^{\otimes}\mathbf{\Lambda}_{[[d]]}^{\top}\right] = \text{vec}\left[\boldsymbol{\zeta}'(\mathbf{\Gamma}^{(d)}\mathbf{\Lambda}_{[[d]]}\mathbf{\Gamma}_d^{\otimes^{\top}})\mathbf{\Gamma}_d^{\otimes}\mathbf{\Lambda}_{[[d]]}^{\top}\right]$$

and

$$\nabla_{\mathbf{\Lambda}} J = \mathbf{\Gamma}^{\otimes^{\top}}\text{vec}\,\boldsymbol{\zeta}'(\mathbf{\Gamma}) = \mathbf{\Gamma}^{\otimes^{\top}}\boldsymbol{\zeta}'(\mathbf{\Gamma}^{\otimes}\text{vec}\,\mathbf{\Lambda})$$

since the elementwise application of ζ allows $\boldsymbol{\zeta}'$ to commute with vectorization and matricization.

The above gradients are simple to compute given current estimates of tensor decomposition parameters and are thus easily incorporated into Algorithm 1.

Imposing a LASSO penalty presents a difficulty in the non-differentiability of the L_1 loss. We address this problem in the same way as the non-differentiable quantile loss ρ_τ. Noting that the LASSO penalty function is $2\rho_{\tau=0.5}$, we can use the same smooth approximation $2\rho_{\tau=0.5,\nu}$ as before. Model estimates obtained from Algorithm 1 with this approximate LASSO penalty converge to the true LASSO-regularized estimates, as shown in Section 4.1.3.

Chapter 4

Model and Estimator Properties

In the following chapter, we examine properties of the proposed qD-FLQR estimators and algorithm. After establishing some basic properties regarding the approximated loss

$$l_N(\alpha, \boldsymbol{\beta}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}, \boldsymbol{\Lambda}) \triangleq \frac{1}{n} \sum_{i=1}^{n} \rho_{\tau\nu_N}\big(y_i - \alpha - \boldsymbol{x}_i^\top \boldsymbol{\beta} - \langle \boldsymbol{Z}_i, [[\boldsymbol{\Lambda} \,|\, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}]] \rangle\big),$$

we study convergence of the solution to the approximated problem as $\nu \overset{N\to\infty}{\to} 0$. Following this, we establish the asymptotic consistency and normality of the proposed estimators. While similar properties were examined in the tensor GLM setting for both the CP and Tucker decompositions [49], [97], these properties do not generalize immediately to the quantile case and thus require separate treatment.

Throughout this section, we assume a monotonic, zero-convergent sequence $(\nu_N)_N$ of positive smoothing parameters and fixed Tucker decomposition ranks $R_1, \ldots, R_q$. We define the the linear predictor $\eta = \alpha + \boldsymbol{x}^\top \boldsymbol{\beta} + \langle \boldsymbol{Z}, [[\boldsymbol{\Lambda} \,|\, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}]] \rangle$, possibly with a subscript i in the sample case. For simplicity, we let $\boldsymbol{\theta}$ denote the sequence of model parameters $(\alpha, \boldsymbol{\beta}, \boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}, \boldsymbol{\Lambda})$. We refer to the parameter blocks $(\alpha, \boldsymbol{\beta})$, $(\boldsymbol{\Gamma}^{(1)}), \ldots, (\boldsymbol{\Gamma}^{(q)})$, $(\boldsymbol{\Lambda})$ as the zeroth, first, ..., q-th, and $(q+1)$-th blocks, respectively. We continue to use the shorthand notation $\boldsymbol{\Gamma}^{\otimes}$ and $\boldsymbol{\Gamma}_d^{\otimes}$ defined in Subsection 3.2.2.

4.1 Smoothed Loss and Algorithm Properties

This section focuses on the approximated/smoothed qD-FLQR loss l_N: we lay out some basic properties of l_N as well as global and local convergence properties of the proposed algorithm. The former will be used later to establish *approximation convergence*—that the minimizer of l_N converges to the minimizer of l as $N \to \infty$.

We assume uniform convergence of l_N to l as $v \to 0$. This is clear for the generalized Huber loss $H_{\tau\nu}$ (Equation 3.5) since, for all $u \in \mathbb{R}$,

$$|H_{\tau\nu}(u) - \rho_\tau(u)| \leq \begin{cases} \frac{1}{2}(\tau-1)^2\nu, & u < 0 \\ \frac{1}{2}\tau^2\nu, & u \geq 0 \end{cases},$$

and so $\max_{u\in\mathbb{R}}|H_{\tau\nu}(u) - \rho_\tau(u)| \leq \max\{\frac{1}{2}(\tau-1)^2\nu, \frac{1}{2}\tau^2\nu\} \overset{v\to 0}{\to} 0$.

From the continuous differentiability of $\rho_{\tau\nu}$, it is trivial to see that l_N is continuously differentiable in its parameters. As noted previously, the smoothed loss is also (not strictly) convex in each block of its parameters.

4.1.1 Global Convergence

We wish to show that the sequence of iterates $\boldsymbol{\theta}_{(k)}$ generated by Algorithm 1 converges regardless of the initial estimate $\boldsymbol{\theta}_{(0)}$. We impose two regularity conditions also used in previous work [49], [97]. Assume that l_N is coercive (so that $\{\boldsymbol{\theta} \,|\, l_N(\boldsymbol{\theta}) \leq l_N(\boldsymbol{\theta}_0)\}$ is compact for any $\boldsymbol{\theta}_0$) and that the stationary points $\boldsymbol{\theta}^*$ of l_N are isolated. These are reasonable regularity conditions on the data. Note that we do not need to consider the nonsingular transformation indeterminacy (see Subsection 2.2.3) as each step holds constant all but at most one component of the Tucker decomposition.

Previous works also assume that each block update is strictly convex, although this is clearly not the case for $\rho_{\tau\nu}$ in general (e.g., for either $H_{\tau\nu}$ or $S_{\tau\nu}$). Furthermore, the regularity condition supposed in Theorem 3.3 of Koenker and Bassett [45], which in fact takes advantage of the non-differentiability of ρ_τ, no longer yields a unique solution (that would interpolate some of the data) if using the smoothed loss. Fortunately, continuity and convexity of l_N and the above regularity conditions guarantee a unique solution in each update step.

In each block update, a global minimum is unique since l_N is continuous and coercive. If two global minima exist, then by the convexity of l_N, the loss must be constant over the segment connecting them. This violates the assumption of isolated minima.

With the above results, we apply the work of Fiorot and Huard [22] (as discussed in [47]). Additionally noting that the loss l_N is strictly monotonic over block updates and that the feasible set in each update is hemicontinuous in the fixed model parameters, global convergence follows immediately. The same arguments hold when considering any convex penalty.

4.1.2 Local Convergence Rates

We are next interested in *local convergence rates*—how quickly algorithm iterates within some neighbourhood of a local mininum $\boldsymbol{\theta}^*$ converge to $\boldsymbol{\theta}^*$—and the rate of this convergence. Investigating this property requires stronger conditions and assumptions than before. We follow the general discussion of de Leeuw [47] that relies on the classical Ostrowski theorem [59], most notably requiring that, at the fixed point, the *algorithmic update map* be differentiable and the spectral radius of the *iteration Jacobian* be strictly less than one.

Let the i-th block update map be denoted by $\mathcal{B}_i$, for $i = 0, \ldots, q+1$. The *algorithmic update map* is then $\mathcal{A} = \mathcal{B}_{q+1} \circ \cdots \circ \mathcal{B}_0$. The individual block updates $\mathcal{B}_i$ are differentiable, as is $\mathcal{A}$ as a composition of differentiable functions.

The iteration Jacobian $\boldsymbol{M} = D\mathcal{A}$ is the derivative matrix of the algorithmic map. We assume that, at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, at least one of the residuals $y_i - \eta_i$ for $i = 1, \ldots, n$ lies within $\big((\tau-1)\nu_N, \tau\nu_N\big)$. With this assumption when using $H_{\tau\nu}$ (or an analogous condition given by Muggeo et al. [55] for $S_{\tau\nu}$), the loss l_N is strictly convex at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, ensuring that the conditions for the implicit function theorem are met.

Let $D_{ij}l_N$ be the mixed derivative matrix of l_N with respect to parameter blocks i and j. Applying the implicit function theorem, we can write $\boldsymbol{M} = -(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{L}^T$, where $\boldsymbol{L}$ and $\boldsymbol{D}$ are strictly lower and diagonal block matrices with blocks $D_{ij}l_N$ [47]. By strict convexity, the diagonal blocks of $\boldsymbol{D}$ (and thus $\boldsymbol{D}$ itself) are positive definite. It can be shown that the spectral radius

of $\boldsymbol{M} = -(\boldsymbol{L}+\boldsymbol{D})^{-1}\boldsymbol{L}^T$ is strictly less than one. This proves (by Ostrowski's theorem) that local convergence is linear.

4.1.3 Approximation Convergence

Algorithm 1 uses a smooth loss $\rho_{\tau\nu}$ that converges uniformly to the quantile loss ρ_τ as $\nu \to 0$. We show here that the iterates generated by the algorithm do indeed converge to a minimizer of the original problem with loss l. This follows from Lemma 2 of Hjort and Polland [36], albeit in a less general setting (e.g., with a sequence of deterministic rather than random functions), so we find it useful to reproduce here. We use the standard notation $B_\delta(\boldsymbol{x})$ to denote the open ball of radius $\delta > 0$ in $\mathbb{R}^p$ centred at $\boldsymbol{x} \in \mathbb{R}^p$.

Let $\boldsymbol{\theta}^*_N \in \mathbb{R}^p$ be the unique minimizer of the smoothed loss l_N and $\boldsymbol{\theta}^* \in \mathbb{R}^p$ a minimizer of l. Since l_N converges uniformly to l, we can choose $N_\varepsilon \in \mathbb{N}$ such that $\sup_{\boldsymbol{\theta}} |l_N(\boldsymbol{\theta}) - l(\boldsymbol{\theta})| < \varepsilon$ for all $N > N_\varepsilon$. Further define $\Delta_\delta \triangleq \inf_{\boldsymbol{s}:|\boldsymbol{s}-\boldsymbol{\theta}^*|=\delta} l(\boldsymbol{s}) - l(\boldsymbol{\theta}^*)$. Let $\boldsymbol{\theta} \triangleq \boldsymbol{\theta}^* + t\boldsymbol{v}$ for any $t > \delta$ and unit vector $\boldsymbol{u} \in \mathbb{R}^p$ so that $\boldsymbol{\theta} \notin B_\delta(\boldsymbol{\theta}^*)$.

By the convexity of l_N,

$$\begin{aligned}(1 - {}^{\delta}\!/_{t})l_N(\boldsymbol{\theta}^*) + ({}^{\delta}\!/_{t})l_N(\boldsymbol{\theta}) &\geq l_N\big((1 - {}^{\delta}\!/_{t})\boldsymbol{\theta}^* + ({}^{\delta}\!/_{t})\boldsymbol{\theta}\big) \\ &= l_N(\boldsymbol{\theta}^* + \delta\boldsymbol{u}).\end{aligned}$$

It follows that, for $N > N_\varepsilon$ with $\varepsilon < \frac{\Delta_\delta}{2}$,

$$\begin{aligned}&({}^{\delta}\!/_{t})\big[l_N(\boldsymbol{\theta}) - l_N(\boldsymbol{\theta}^*)\big] \\ &\qquad \geq l_N(\boldsymbol{\theta}^* + \delta\boldsymbol{u}) - l_N(\boldsymbol{\theta}^*) \\ &\qquad = \big[l(\boldsymbol{\theta}^* + \delta\boldsymbol{u}) - l(\boldsymbol{\theta}^*)\big] + \big[l_N(\boldsymbol{\theta}^* + \delta\boldsymbol{u}) - l(\boldsymbol{\theta}^* + \delta\boldsymbol{u})\big] - \big[l(\boldsymbol{\theta}^*) - l_N(\boldsymbol{\theta}^*)\big] \\ &\qquad \geq \Delta_\delta - 2\varepsilon \\ &\qquad > 0.\end{aligned}$$

Therefore, $l_N(\boldsymbol{\theta}) > l_N(\boldsymbol{\theta}^*)$ for all $\boldsymbol{\theta} \notin B_\delta(\boldsymbol{\theta}^*)$. Then it must be that $\boldsymbol{\theta}^*_N \in N_\delta(\boldsymbol{\theta}^*)$.

Thus, given $\delta > 0$, we can find $N'_\delta = N_\varepsilon$ (with $\varepsilon < \frac{\Delta_\delta}{2}$) so that $\boldsymbol{\theta}^*_N \in B_\delta(\boldsymbol{\theta}^*)$ for all $N > N'_\delta$, and so $\lim_{N\to\infty} \boldsymbol{\theta}^*_N = \boldsymbol{\theta}^*$. This proves approximation convergence.

4.2 Statistical Properties

We are now concerned with important statistical properties of qD-FLQR model estimators. It has been shown previously that the problem of minimizing the quantile loss l is equivalent to that of maximizing the log-likelihood of independent and identically-distributed asymmetric Laplace observations [72], [92]. Indeed, the general asymmetric Laplace density function is given by

$$f(y; m, \lambda, \kappa) = \left(\frac{\lambda}{\kappa + 1/\kappa}\right) \exp\left\{ -|y - m|\lambda\kappa^{\mathrm{sgn}(y-m)}\right\}$$

for $m \in \mathbb{R}$, $\lambda, \kappa > 0$. Setting $\lambda = \sqrt{\tau(1-\tau)}$, $\kappa = \sqrt{\tau/1-\tau}$, and $m = \eta$ (for known τ), we obtain

$$f(y; \eta) = \begin{cases} \tau(1-\tau)\exp\{-\tau(y-\eta)\}, & y \geq \eta \\ \tau(1-\tau)\exp\{-(\tau-1)(y-\eta)\}, & y < \eta \end{cases} = \tau(1-\tau)e^{-\rho_\tau(y-\eta)},$$

with negative log-likelihood equal (up to an additive constant in τ) to the quantile loss.

As such, it is meaningful to derive a score function and information matrix. For notational simplicity, we ignore the scalar part of the model, which is trivial to incorporate.

Following this, we examine qD-FLQR model identifiability as well as the consistency and asymptotic normality of model estimators. Since l is not differentiable but the smoothed loss l_N converges uniformly to l as $N \to \infty$, we derive these properties using l_N instead, adopting the standard asymptotic setup with sample size $n \to \infty$. This is valid as the uniform convergence of $\rho_{\tau\nu}$ to ρ_τ suggests that

$$f_{\tau\nu}(y; \eta) \triangleq C_{\tau\nu} e^{-\rho_{\tau\nu}(y-\eta)}$$

for some constant $C_{\tau\nu} > 0$ dependent on τ and $\nu > 0$ is a valid probability density. We refer to $f_{\tau\nu}$ as the *smoothed asymmetric Laplace density*.

Note that the (smoothed) asymmetric Laplace distribution is not in the exponential class when η is unknown (as is the case here), so the results of this chapter do not follow as a special case of previous work for the GLM model [49], although we take a similar approach using empirical process theory [82].

4.2.1 Score, Information, and Identifiability

For notational convenience throughout this subsection, we define, for $d = 1, \ldots, q$, the Jacobian matrices $J_d \triangleq \frac{\partial \operatorname{vec} \boldsymbol{\Gamma}}{\partial \operatorname{vec} \boldsymbol{\Gamma}^{(d)}}$. From the results of Subsection 3.3.2, it follows that

$$\boldsymbol{J}_d = \boldsymbol{\Pi}_d \frac{\partial \operatorname{vec} \boldsymbol{\Gamma}_{[[d]]}}{\partial \operatorname{vec} \boldsymbol{\Gamma}^{(d)}} = \boldsymbol{\Pi}_d [\boldsymbol{I}_{I_d} \otimes (\boldsymbol{\Gamma}_d^{\otimes} \boldsymbol{\Lambda}_{[[d]]}^{\top})]$$

where $\boldsymbol{\Pi}_d \in \mathbb{R}^{\prod_j I_j \times \prod_j I_j}$ is the (row) permutation matrix mapping $\operatorname{vec} \boldsymbol{A}_{[[d]]}$ to $\operatorname{vec} \boldsymbol{A}$ for any $\boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_q}$.

Letting η be the linear predictor as defined at the beginning of this chapter, it is easy to see that, by application of the chain rule,

$$\frac{\partial \eta}{\partial \operatorname{vec} \boldsymbol{\Gamma}^{(d)}} = \frac{\partial \eta}{\partial \operatorname{vec} \boldsymbol{\Gamma}} \frac{\partial \operatorname{vec} \boldsymbol{\Gamma}}{\partial \operatorname{vec} \boldsymbol{\Gamma}^{(d)}} = (\operatorname{vec} \boldsymbol{Z})^{\top} \boldsymbol{J}_d.$$

and

$$\operatorname{vec} \frac{\partial \eta}{\partial \boldsymbol{\Lambda}} = \frac{\partial \eta}{\partial \operatorname{vec} \boldsymbol{\Gamma}} \frac{\partial \operatorname{vec} \boldsymbol{\Gamma}}{\partial \operatorname{vec} \boldsymbol{\Lambda}} = (\operatorname{vec} \boldsymbol{Z})^{\top} \boldsymbol{\Gamma}^{\otimes}$$

It follows that the gradient of η with respect to Tucker decomposition parameters (which we take to be the vector of partial derivatives with respect to $(\operatorname{vec} \boldsymbol{\Gamma}^{(1)^{\top}}, \ldots, \operatorname{vec} \boldsymbol{\Gamma}^{(q)^{\top}}, \operatorname{vec} \boldsymbol{\Lambda}^{\top})^{\top}$) is

$$\nabla_{\boldsymbol{\theta}} \eta = [\boldsymbol{J}_1 | \ldots | \boldsymbol{J}_q | \boldsymbol{\Gamma}^{\otimes}]^{\top} \operatorname{vec} \boldsymbol{Z},$$

yielding the score function (for $n = 1$)

$$\nabla l_N(\boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}, \boldsymbol{\Lambda}) = -\rho'_{\tau \nu_N}(y - \eta)[\boldsymbol{J}_1 | \ldots | \boldsymbol{J}_q | \boldsymbol{\Gamma}^{\otimes}]^{\top} \operatorname{vec} \boldsymbol{Z}.$$

The (expected) Fisher Information follows as

$$\begin{aligned} I_N(\boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(q)}, \boldsymbol{\Lambda}) &= \mathbb{E}[\nabla l_N \nabla l_N^{\top}] \\ &= \mathbb{E}[(\rho'_{\tau \nu_N}(y - \eta))^2][\boldsymbol{J}_1, \ldots, \boldsymbol{J}_q, \boldsymbol{\Gamma}^{\otimes}]^{\top} \operatorname{vec} \boldsymbol{Z} (\operatorname{vec} \boldsymbol{Z})^{\top} [\boldsymbol{J}_1, \ldots, \boldsymbol{J}_q, \boldsymbol{\Gamma}^{\otimes}]. \end{aligned}$$

The non-uniqueness of the Tucker decomposition under nonsingular linear transformations of its components was discussed in Subsection 2.2.3. Consequently, the qD-TQR model is not identifiable without further restrictions. To address this, we gave one (of infinitely many) restricted parameter spaces $\mathcal{G}_q$

within which the Tucker decomposition would be unique. The above score and information matrix require adjustment to account for this. We don't attempt this here for $\mathcal{G}_q$ as the choice of parameter space is arbitrary and motivated by context.

We investigate model identifiability in $\mathcal{G}_q$ and refer to the results of Rothenberg [71], noting that all required assumptions are satisfied. Specifically, $\mathcal{G}_q$ may be viewed as $\mathbb{R}^{p_q^{\text{eff}}}$, where p_q^{eff} is the number of parameters in the Tucker decomposition not fixed at unity (as given in Equation 2.6), and is thus open (inside itself); $f_{\tau\nu}$ is a proper density for all $\boldsymbol{\theta} \in \mathcal{G}_q$; the support of $f_{\tau\nu}$ does not depend on $\boldsymbol{\theta}$; and $f_{\tau\nu}$ is continuously differentiable with respect to η, and thus, $\boldsymbol{\theta}$.

We say the qD-FLQR model with parameters $\boldsymbol{\theta}_0$ is *locally identifiable* in $\mathcal{G}_q$ if, in some open neighbourhood $U \subseteq \mathcal{G}_q$ of $\boldsymbol{\theta}_0$, no other $\boldsymbol{\theta} \in \mathcal{G}_q$ suggests the same distribution for the observed random variables Y. By Theorem 1 of Rothenberg, this holds if and only if $I(\boldsymbol{\theta})$ has full rank in U.

Global identifiability is difficult to establish outside the exponential class. Theorem 4 of Rothenberg provides a general criteria requiring, for all model parameters θ_i, that $\theta_i = \mathbb{E}[\phi_i(Y)]$ for some known function ϕ_i. Unfortunately, this is not the case for the qD-FLQR model, as η depends on all parameters of the Tucker decomposition and are not "interpretable characteristics" of $f_{\tau\nu}$. Fortunately, local identifiability is enough to establish the asymptotic properties in the next two subsections.

4.2.2 Consistency

The following two subsections primarily employ properties of M-estimation in empirical process theory [82]. We loosely follow the arguments of [97] and [49].

We establish estimator consistency via Theorem 5.7 of van der Vaart [82]:

Lemma 4. *for random functions M_n and a fixed function M of $\theta \in \Theta$ such that, for all $\varepsilon > 0$,*

$$\sup_{\theta \in \Theta} |M_n(\theta) - M(\theta)| \xrightarrow{p} 0, \tag{4.1}$$

$$\sup_{\theta:|\theta-\theta_0|\geq\varepsilon} M(\theta) < M(\theta_0), \tag{4.2}$$

any sequence of estimators $\hat{\theta}_n$ *with* $M_n(\hat{\theta}_n) \geq M_n(\theta_0) - o_P(1)$ *converges in probability to* θ_0*.*

In our setting, we take $\hat{\theta}_n = \hat{\boldsymbol{\theta}}_n$ to be the estimators of $\theta_0 = \boldsymbol{\theta}_0$ from a sample of size n so that $\hat{\boldsymbol{\Gamma}}_n = [[\hat{\boldsymbol{\Lambda}}_n \,|\, \hat{\boldsymbol{\Gamma}}_n^{(1)}, \ldots, \hat{\boldsymbol{\Gamma}}_n^{(q)}]]$ and $\boldsymbol{\Gamma}_0 = [[\boldsymbol{\Lambda}_0 \,|\, \boldsymbol{\Gamma}_0^{(1)}, \ldots, \boldsymbol{\Gamma}_0^{(q)}]]$, suppressing the implicit relationship between $\boldsymbol{\Gamma}$ and $\boldsymbol{\theta}$. It is enough to let $\boldsymbol{\theta}$ be any individual parameter block. Let

$$M(\boldsymbol{\theta}) = \mathbb{P} l_N(Y; \boldsymbol{\theta}) = \int_{\mathbb{R}} l_N(y; \boldsymbol{\theta}) f_{\tau\nu_N}(y; \boldsymbol{\theta}_0) dy$$

and

$$M_n(\boldsymbol{\theta}) = \mathbb{P}_n M(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} l_N(y_i; \boldsymbol{\theta}),$$

where $\mathbb{P}_n$ denotes the empirical measure. By the Strong Law of Large Numbers, $M_n(\boldsymbol{\theta}) \overset{\text{a.s.}}{\to} M(\boldsymbol{\theta})$ and the stochastic order property for $\hat{\boldsymbol{\theta}}_n$ is satisfied.

Condition 4.2 (possibly with equality) follows from the Cramer-Rao lower bound (i.e., the information inequality). Local identifiability is then sufficient to guarantee strict inequality, so Condition 4.2 holds.

We verify Condition 4.1 using the Glivenko-Cantelli Theorem, the statement for which we omit here for brevity (Theorem 19.13, van der Vaart [82]). Note that $\{\langle \boldsymbol{Z}, \boldsymbol{\Gamma} \rangle \,|\, \boldsymbol{\Gamma} = \boldsymbol{\Gamma}(\boldsymbol{\theta}), \boldsymbol{\theta} \in \mathcal{G}_q\}$ is a *Vapnik-Červonekis class* as a collection of polynomials with finite degree, and is thus also *Glivenko-Cantelli* and *Donsker*. Previous work ensures that $\{l_N(y; \boldsymbol{\theta}) \,|\, \boldsymbol{\theta} \in \mathcal{G}_q\}$ is also Donsker by restricting the parameter space to a compact subset. Such a restriction is not necessary here, as l_N is already Lipschitz in its parameters, so this set is Donsker as the composition of a Donsker class with a Lischitz function. The function l provides a suitable envelope for this class (when $\rho_{\tau\nu} = H_{\tau\nu}$, the generalized Huber function, although other envelopes can be easily found for other approximations) as $\mathbb{P} l < \infty$ and $\mathbb{P} l^2 < \infty$. This set is therefore also $\mathbb{P}$-Glvenko-Cantelli and $\mathbb{P}$-Donsker (Chapter 19, van der Vaart [82]). The conditions for the Glivenko-Cantelli Theorem are thus satisfied, so Condition 4.1 holds.

We conclude that $\boldsymbol{\theta}_n$ converges in probability to $\boldsymbol{\theta}_0$. This also implies that $\hat{\boldsymbol{\Gamma}} \xrightarrow{p} \boldsymbol{\Gamma}_0$ by the Open Mapping Theorem.

It is easy to show that the consistency results also hold if we consider $n, N \to \infty$. Let $\boldsymbol{\theta}_n^{(N)}$ be the estimator of the solution $\boldsymbol{\theta}_0^{(N)}$ to the problem using smoothed loss l_N, and let $\boldsymbol{\theta}_0$ be the solution to the non-smoothed problem with loss l. Using known consistency results for the quantile regression estimators and approximation convergence from Subsection 4.1.3, it follows that, for all $\varepsilon > 0$,

$$\begin{aligned}
&\lim_{n\to\infty} \mathbb{P}(|\lim_{N\to\infty} \boldsymbol{\theta}_n^{(N)} - \boldsymbol{\theta}_0| \geq \varepsilon) \\
&\quad \leq \lim_{n\to\infty} \mathbb{P}(|\lim_{N\to\infty} \boldsymbol{\theta}_n^{(N)} - \boldsymbol{\theta}_0^{(N)}| \geq \varepsilon/2) + \lim_{n\to\infty} \mathbb{P}(|\lim_{N\to\infty} \boldsymbol{\theta}_0^{(N)} - \boldsymbol{\theta}_0| \geq \varepsilon/2) \\
&\quad = 0.
\end{aligned}$$

This verifies the consistency of the estimators proposed in Algorithm 1.

4.2.3 Asymptotic Normality

Quadratic mean differentiability of l_N (definition omitted for brevity) is central to the following proof of the asymptotic normality of $\hat{\boldsymbol{\theta}}_n$. By Theorem 7.6 of van der Vaart [82], continuous differentiability of l_N and the existence and continuity of the elements of I_N are sufficient for quadratic mean differentiability. This is clearly the case, so we conclude that the model estimated under loss l_N is differentiable in quadratic mean.

As a final step, we apply Theorem 5.39 of van der Vaart [82]:

Lemma 5. *Suppose that the model $(P_\theta : \theta \in \Theta \subset \mathbb{R}^k)$ is differentiable in quadratic mean at an interior point $\theta_0 \in \Theta$ and that there exists a measurable function $\dot{l}$ with $P_{\theta_0}\dot{l}^2 < \infty$ such that, for every θ_1, θ_2 in a neighbourhood of θ_0,*

$$|\log p_{\theta_1}(x) - \log p_{\theta_2}(x) \leq \dot{l}(x)|\theta_1 - \theta_2|.$$

If the Fisher information matrix $I(\theta_0)$ is nonsingular and $\hat{\theta}_n$ is consistent, then $\sqrt{n}(\hat{\theta}_n - \theta_0)$ is asymptotically normal with mean zero and covariance matrix $I(\theta_0)^{-1}$.

The stated conditions hold by the consistency proven in Subsection 4.2.2 and since l_N is Lipschitz in its parameters. In particular, for a neighbourhood U of $\theta_0 = \boldsymbol{\theta}_0$ we take $\log p_\theta(y) = \log f_{\tau\nu}(y; \boldsymbol{\theta})$ with $\dot{l}(y) \triangleq \sup_{\boldsymbol{\theta} \in U} |\nabla_\theta \log f_\tau(y)|$, which satisfies the square-integrability condition. The above result applies, proving asymptotic normality of $\hat{\boldsymbol{\theta}}_n$.

Chapter 5

Data Analysis

In the following chapter, we examine the performance of the proposed model algorithm and estimators in two settings. The first, in Section 5.1, is a simulation study that has been previously used in the tensor regression literature [49], [96], [97] and the second, in Section 5.2, is an application to a real-world neuroimaging dataset derived from the ADNI [88]. Our simulation studies consider a variety of settings, including different Tucker decomposition ranks, quantile levels τ, response signal-to-noise ratios, response error distributions, numbers of functional covariates, and regularization strengths λ. In particular, our results highlight the superiority of our quantile approach relative to the tensor GLM in the case of heavy-tailed errors and the use of regularization even in ideal, simulated settings. We also examine the assumption of factor equivalence discussed in Subsection 5.1.4. Our real-world data analyses illustrate an application of our model to noisy data in non-ideal settings and the necessity of regularization in obtaining interpretable tensor effect estimates.

5.1 Simulation Studies

Our simulation setup is similar to that of Li et al. and Zhou et al. [49], [97]. Following the qD-TQR model notation used in Chapter 3, we take $\boldsymbol{\Gamma} \in \mathbb{R}^{64\times 64}$ (i.e., $q = 2$ with $I_1 = I_2 = 64$) as the true tensor effect with simulated tensor observations $\boldsymbol{Z}_i \in \mathbb{R}^{64\times 64}$, for $i = 1, \ldots, n$, with independent standard normal elements. We take $\alpha = 0$ and $\boldsymbol{\beta} = (1, \ldots, 5)^\top$ with scalar observations $\boldsymbol{x}_i \in \mathbb{R}^5$ with elements again sampled from independent standard normal distributions.

The scalar response Y_i is simulated via $Y_i = \eta_i + \varepsilon_i$ with $\eta_i = \alpha + \langle \boldsymbol{x}_i, \boldsymbol{\beta} \rangle + \langle \boldsymbol{Z}_i, \boldsymbol{\Gamma} \rangle$, where the ε_i are independent, identically distributed random variables. We consider normal, T, Cauchy, and χ^2 distributions for the ε, with parameters set to achieve a signal-to-noise ratio (SNR) of either 3 or 5. For simplicity, we set $R_1 = R_2 \triangleq R$ throughout.

Our model's ability to predict Y for new observations and estimate $\boldsymbol{\Gamma}$ is of primary interest, so we consider a few geometric shapes of varying complexity for $\boldsymbol{\Gamma}$. Shown in Figure 5.1, these are the square, T, triangle, circle, and star signals. As measures of model performance, we present (quantile) prediction error for both training and tests sets as well as estimator root-mean-square-error (RMSE) for $\boldsymbol{\Gamma}$, all averaged over five iterations. The same dataset $\mathcal{D}_k$ is used for the k-th iteration across all tests using the same sample size. Since we are working with simulated data, we use test set error as a model goodness-of-fit criteria for determining an optimal decomposition rank R.

5.1.1 Unregularized Estimation and Comparison to the Tensor GLM

We first apply the proposed qD-FLQR model without regularization (i.e, $\lambda = 0$) and the settings described previously, simulating training and test sets of size $n_{\text{train}} = 2000$ and $n_{\text{test}} = 200$, respectively. To demonstrate the advantages of the quantile model, we compare our results (for median regression, i.e., $\tau = 0.5$) to the Tucker tensor GLM proposed by Zhou et al. [49].

Figure 5.1 shows the true tensor effects $\boldsymbol{\Gamma}$ with example estimates $\hat{\boldsymbol{\Gamma}}$ obtained using Tucker decomposition ranks $R = 1, \ldots, 5$, for all five signals, at $\tau = 0.5$, SNR = 3, and Gaussian error. Figure 5.2 shows the same estimates but for Cauchy error. As expected, training loss decreases as R increases (i.e., as model capacity increases), although it is clear from the test loss that large values of R can result in overfitting. This can be seen in Figure 5.4, which plots training/test loss and estimator RMSE under Gaussian and Cauchy error. In particular, training loss decreases but test loss and estimate RMSE increases for the T signal around $R = 1$. Visually, this produces estimates $\hat{\boldsymbol{\Gamma}}$ with increasingly intense background noise, although the qD-FLQR estimates are

clearly much less noisy under Cauchy error. This further motivates estimate regularization, which we examine in Subsection 5.1.3.

Comparing the qD-FLQR model with the Tucker tensor GLM (which similarly uses a block relaxation approach), we find that relative performance is dependent on the true error distribution. Table 5.1 shows estimator RMSE at SNR $= 3$ and $\tau = 0.5$ for all shapes, Tucker decomposition ranks $R = 1, \ldots, 5$, error distributions, and signal shapes, while Figure 5.4 shows this graphically for only Gaussian and Cauchy errors. For Gaussian error (or T-distributed error to a lesser extent), we find that both approaches perform nearly identically. However, for Cauchy error, the qD-FLQR model clearly outperforms the Tucker tensor GLM. As both models use block relaxation in a similar way, this difference can be attributed to the general benefits of quantile regression, which we know to be superior to least-squares regression when outliers or heavy-tailed errors (such as Cauchy) are present. A comparison of example qD-FLQR estimates $\hat{\mathbf{\Gamma}}$ under different errors is available in Figure 5.3, which also compares qD-FLQR and Tucker tensor GLM estimates.

As expected, more complex signals require a higher value of R to be adequately recovered, as seen in Figures 5.1 and 5.4. For example, the square requires $R = 1$ while the T requires $R = 2$ to minimize estimate RMSE. Also unsurprisingly, model performance increases with increasing SNR, as illustrated in Figure 5.5. All performance measures worsened as τ deviated from 0.5 (Figure 5.6), as is the case in traditional quantile regression. Visually, estimates obtained under Cauchy error did not change as much as those under Gaussian error, with noticeable degradation only occurring at $\tau = 0.2$ or $\tau = 0.95$.

Overall, these result suggest that the proposed qD-FLQR model and estimators perform well under a variety of error distributions (which do not need to be specified *a priori*) and are particularly well-suited to heavy-tailed errors. Furthermore, Figure 5.2 and Table 5.1 suggest excellent performance when error follows a Cauchy distribution.

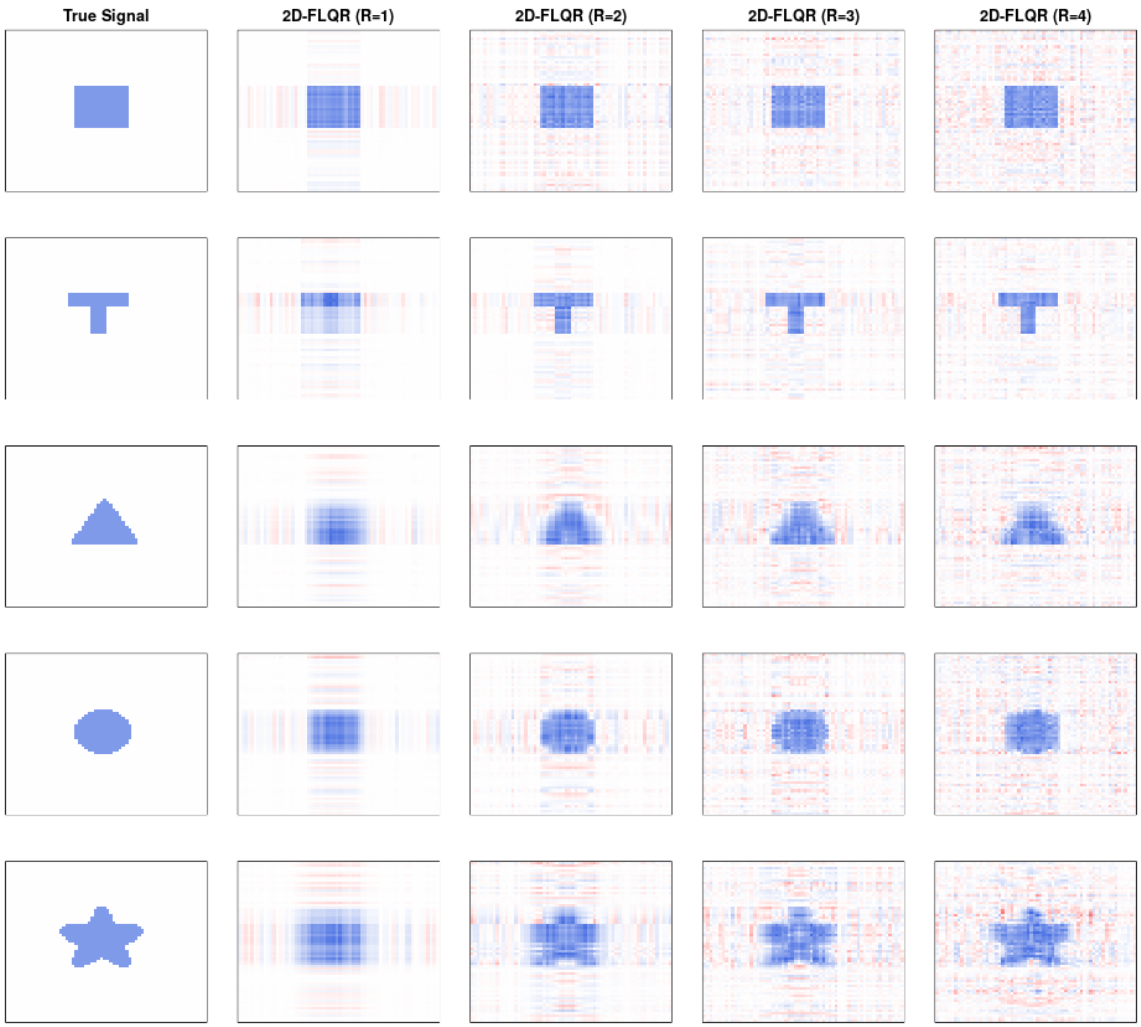

Figure 5.1: Examples of tensor effect estimates obtained for the 2D-FLQR model for five signals using Tucker decomposition ranks $R = 1, 2, 3, 4$, $\tau = 0.5$, SNR $= 3$, Gaussian error, and $n = 2000$. The colour scale indicates the effect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue). Each row corresponds to a different true signal $\mathbf{\Gamma}$. The leftmost column shows this true signal with individual pixels having value either 0 or 1. The four columns to the right show estimates $\hat{\mathbf{\Gamma}}$ for $R = 1, 2, 3, 4$, from left to right.

Figure 5.2: Examples of tensor effect estimates obtained for the 2D-FLQR model for five signals using Tucker decomposition ranks $R = 1, 2, 3, 4$, $\tau = 0.5$, SNR $= 3$, Cauchy error, and $n = 2000$. The colour scale indicates the effect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue). Each row corresponds to a different true signal $\mathbf{\Gamma}$. The leftmost column shows this true signal with individual pixels having value either 0 or 1. The four columns to the right show estimates $\hat{\mathbf{\Gamma}}$ for $R = 1, 2, 3, 4$, from left to right.

R	Shape	2D-FLQR				Tucker Tensor GLM			
		Gaussian	Cauchy	T	χ^2	Gaussian	Cauchy	T	χ^2
1	Square	0.052	0.001	0.040	0.046	0.041	0.042	0.039	0.040
	T	0.102	0.094	0.100	0.100	0.098	0.098	0.098	0.097
	Triangle	0.111	0.102	0.109	0.109	0.107	0.106	0.106	0.107
	Circle	0.099	0.086	0.096	0.096	0.093	0.093	0.093	0.093
	Star	0.149	0.139	0.148	0.149	0.145	0.146	0.145	0.144
2	Square	0.082	0.003	0.065	0.072	0.071	0.071	0.070	0.069
	T	0.055	0.002	0.045	0.049	0.045	0.042	0.043	0.043
	Triangle	0.116	0.077	0.102	0.099	0.092	0.093	0.091	0.092
	Circle	0.106	0.061	0.094	0.096	0.087	0.086	0.084	0.088
	Star	0.153	0.112	0.143	0.148	0.130	0.132	0.131	0.129
3	Square	0.099	0.004	0.086	0.089	0.092	0.092	0.094	0.094
	T	0.070	0.002	0.059	0.066	0.063	0.062	0.061	0.062
	Triangle	0.122	0.065	0.111	0.114	0.100	0.104	0.102	0.100
	Circle	0.117	0.051	0.110	0.106	0.101	0.102	0.099	0.101
	Star	0.161	0.084	0.141	0.139	0.128	0.124	0.125	0.129
4	Square	0.116	0.005	0.101	0.105	0.115	0.111	0.116	0.115
	T	0.084	0.003	0.074	0.075	0.081	0.078	0.078	0.077
	Triangle	0.131	0.062	0.124	0.119	0.117	0.120	0.118	0.116
	Circle	0.133	0.039	0.118	0.119	0.122	0.115	0.117	0.118
	Star	0.173	0.081	0.155	0.165	0.148	0.149	0.143	0.147
5	Square	0.131	0.006	0.115	0.120	0.134	0.127	0.135	0.136
	T	0.097	0.004	0.087	0.089	0.094	0.088	0.094	0.096
	Triangle	0.141	0.057	0.134	0.138	0.136	0.134	0.135	0.131
	Circle	0.145	0.025	0.137	0.136	0.139	0.136	0.139	0.140
	Star	0.191	0.078	0.166	0.179	0.172	0.168	0.168	0.173

Table 5.1: Estimate RMSE for $\hat{\mathbf{\Gamma}}$ using the 2D-FLQR model and the Tucker tensor GLM for various signals, error distributions, and tensor decomposition ranks R. We hold $\tau = 0.5$ (so that the conditional mean/median estimates are comparable), SNR = 3, and $n = 2000$. Each value in the table is an average over 5 simulations.

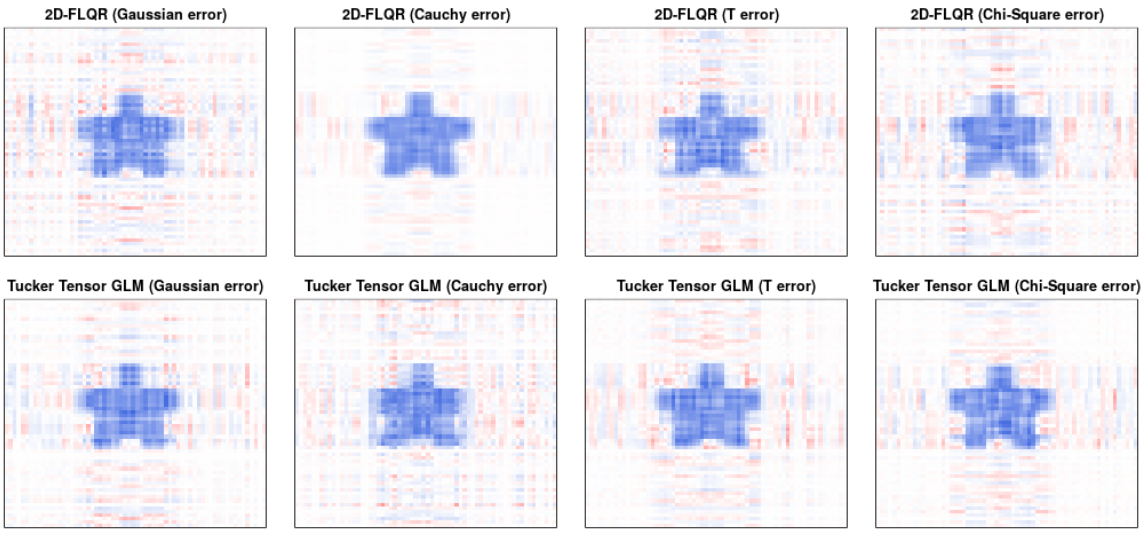

Figure 5.3: Examples of tensor effect estimates for the star signal obtained for the 2D-FLQR model (top) and for the Tucker tensor GLM (bottom) under various error distributions, Tucker decomposition rank $R = 3$, $\tau = 0.5$, SNR $= 3$ and $n = 2000$. The colour scale indicates the effect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue).

5.1.2 Sample size

In this subsection, we briefly consider the effect of sample size n on tensor effect estimates $\hat{\mathbf{\Gamma}}$. Figure 5.7 shows how estimates degrade as sample size decreases. Again we find that the qD-FLQR model performs well when the error distribution is heavy-tailed, as in the Cauchy distribution. Figure 5.8 shows how training and test loss and RMSE change over different values of n. Estimate performance degrades similarly regardless of signal complexity (e.g., the star vs. the T signal) or error distribution.

5.1.3 Regularization

We next consider regularizing model estimates using a LASSO penalty, as described in Subsection 3.3.2. In this setting, a LASSO penalty applied to individual entries of $\hat{\mathbf{\Gamma}}$ is motivated by estimate background noise observed in the previous subsection, while an explicit smoothness penalty does not (yet) seem necessary. Throughout this section, we fix SNR $= 3$, $R = 3$, $n = 2000$, and $\tau = 0.5$. We choose to use a Gaussian error here since we have

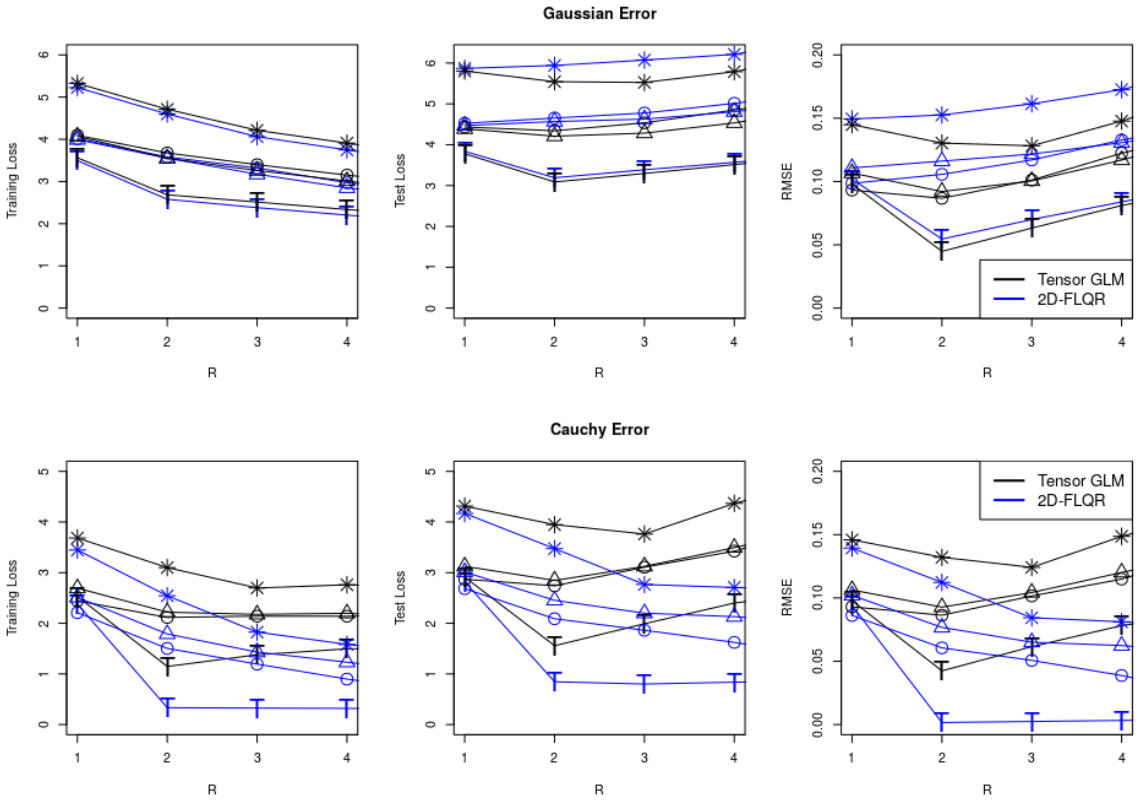

Figure 5.4: Model performance measures (averaged over 5 simulations) under Gaussian (top) and Cauchy (bottom) error distributions vs. Tucker decomposition rank R, with $\tau = 0.5$, SNR $= 3$, and $n = 2000$. The two panes in the left, centre, and right colums show training (quantile) loss, test (quantile) loss, and RMSE for $\hat{\mathbf{\Gamma}}$, respectively. Plot symbols represent the true signal (T, triangle, circle, or star) being estimated, while color indicates whether the estimate was obtained from the 2D-FLQR model (blue) or the Tucker tensor GLM (black).

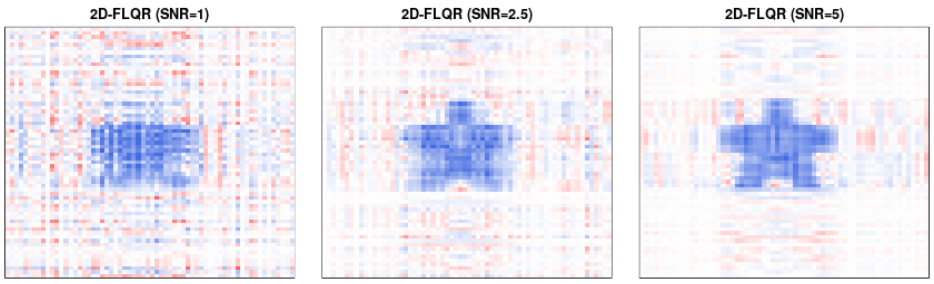

Figure 5.5: Examples of tensor effect estimates of the star signal for the 2D-FLQR model for SNR $= 1, 3, 5$, using Tucker decomposition ranks $R = 3$, $\tau = 0.5$, Gaussian error, and $n = 2000$. The colour scale indicates the estimated effect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue).

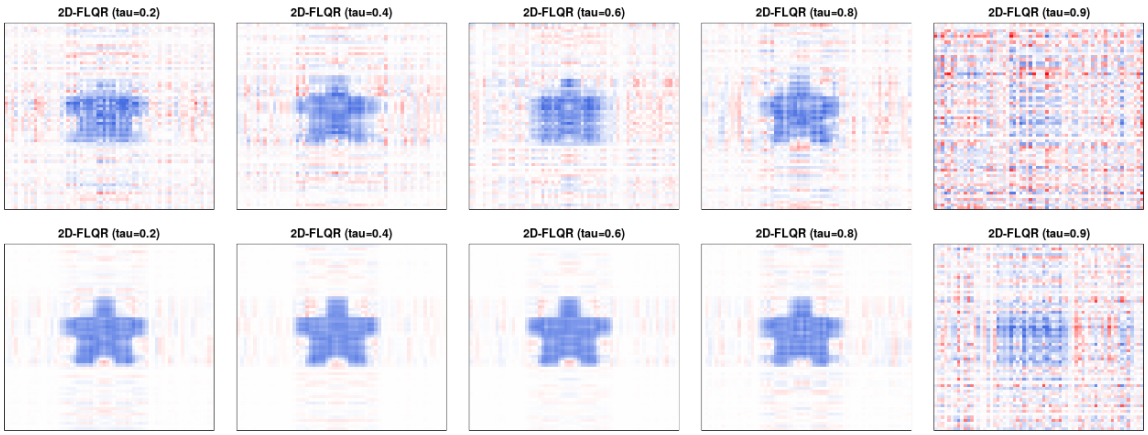

Figure 5.6: Examples of tensor effect estimates of the star signal for the 2D-FLQR model for quantile levels $\tau = 0.2, 0.4, 0.6, 0.8, 0.9$ and Gaussian (top) or Cauchy (bottom) error, using Tucker decomposition ranks $R = 3$ and $n = 2000$. The colour scale indicates the estimated effect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue).

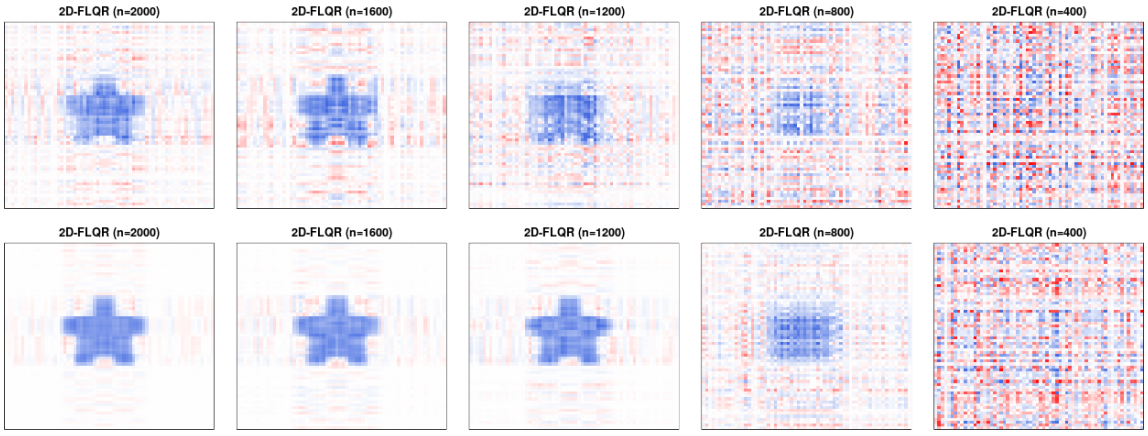

Figure 5.7: Examples of tensor effect estimates of the star signal for the 2D-FLQR model at various sample sizes $n = 2000, 1600, 1200, 800, 400$ and Gaussian (top) or Cauchy (bottom) error, using Tucker decomposition ranks $R = 3$, $\tau = 0.5$, and SNR $= 3$. The colour scale indicates the estimated effect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue).

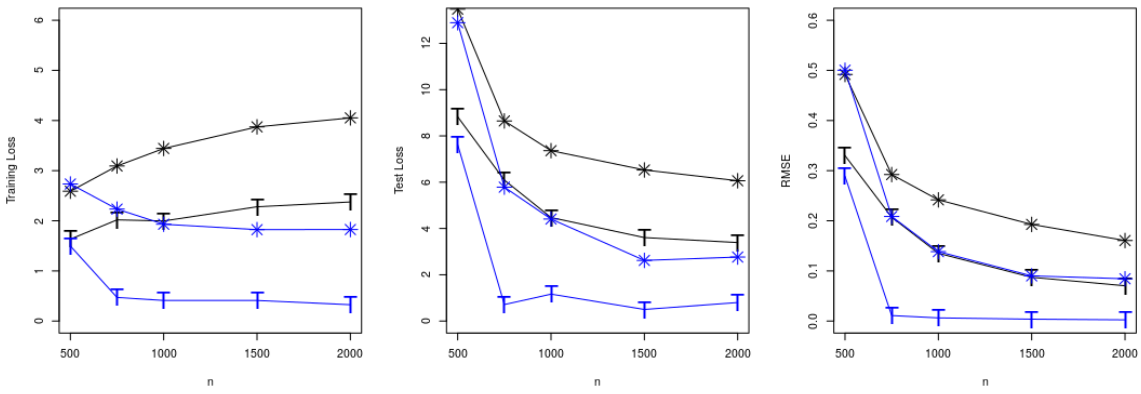

Figure 5.8: Model performance measures (averaged over 5 simulations) under Gaussian (black) and Cauchy (blue) error distributions vs. sample size n, with Tucker decomposition rank $R = 3$, $\tau = 0.5$, SNR $= 3$, and $n = 2000$. Subplots show training (quantile) loss (left), test (quantile) loss (centre), and RMSE for $\hat{\mathbf{\Gamma}}$ (right), respectively. Plot symbols represent the true signal (T or star) being estimated.

already found it can result in noisy effect estimates (Figure 5.1). Since we are simulating training and test data, we use test error (quantile loss plus penalty) to determine an optimal value of the regularization parameter λ.

Figure 5.12 shows example estimates $\hat{\mathbf{\Gamma}}$ for various values of λ, while Figure 5.10 plots test loss as a function of λ and presents the optimal tensor effect estimate. Visually, the regularized estimates are less noisy and the true signal is easier to identify. As we will see later, penalties enforcing the smoothness of tensor effect estimates will be necessary with real-world and/or less-ideal data.

5.1.4 Factor Equivalence and Multiple Covariates

Subsection 3.3.1 discussed ways to incorporate multiple functional covariates into the qD-FLQR model. We noted as a special case that tensor observations of the same size and with factor equivalent Tucker decompositions (Subsection2.2.3) can be combined into a single tensor (with an extra "variable selection" dimension). Relative to treating each tensor decomposition separately, this approach is convenient and computationally efficient in terms of

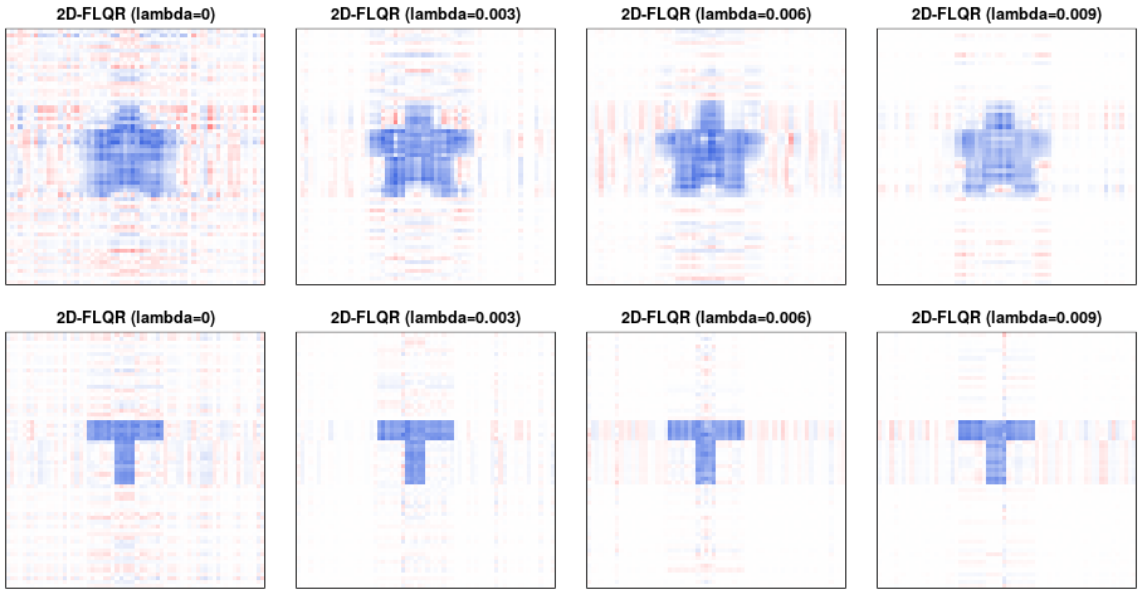

Figure 5.9: Examples of tensor effect estimates of the star (top) and T (bottom) signals for the 2D-FLQR model for various regularization parameter values λ, using Tucker decomposition ranks $R = 3$, $\tau = 0.5$, Gaussian error, and $n = 2000$. The colour scale indicates the estimatedeffect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue).

the proposed algorithm. In this subsection, we investigate the effect of assuming factor equivalence in cases where it is or is not appropriate to do so. We use the T signal as an example throughout since it has an exact Tucker decomposition with $q = 2$ and ranks $R_1 = R_2 = 2$.

The Tucker decomposition of a two-dimensional tensor (i.e., a matrix) has a convenient form. Namely, for $\boldsymbol{A} \in \mathbb{R}^{I_1 \times I_2}$,

$$\boldsymbol{A} = [[\boldsymbol{\Lambda} \,|\, \boldsymbol{\Gamma}^{(1)}, \boldsymbol{\Gamma}^{(2)}]] = \boldsymbol{\Gamma}^{(1)} \boldsymbol{\Lambda} {\boldsymbol{\Gamma}^{(2)}}^{\top},$$

suggesting that the application of any linear transformation $\boldsymbol{T} \in \mathbb{R}^{I_2 \times J}$ to $\boldsymbol{A}$ will yield Tucker decomposition

$$\boldsymbol{A}\boldsymbol{T} = [[\boldsymbol{\Lambda} \,|\, \boldsymbol{\Gamma}^{(1)}, \boldsymbol{T}^{\top}\boldsymbol{\Gamma}^{(2)}]] = \boldsymbol{\Gamma}^{(1)} \boldsymbol{\Lambda} \left(\boldsymbol{T}^{\top}\boldsymbol{\Gamma}^{(2)}\right)^{\top} = \boldsymbol{\Gamma}^{(1)} \boldsymbol{\Lambda} {\boldsymbol{\Gamma}^{(2)}}^{\top} \boldsymbol{T}.$$

These two decompositions are not factor equivalent, even when $J = I_2$. We demonstrate below that, even for very simple linear transformations, an incorrect assumption of factor equivalence will heavily affect tensor effect estimates.

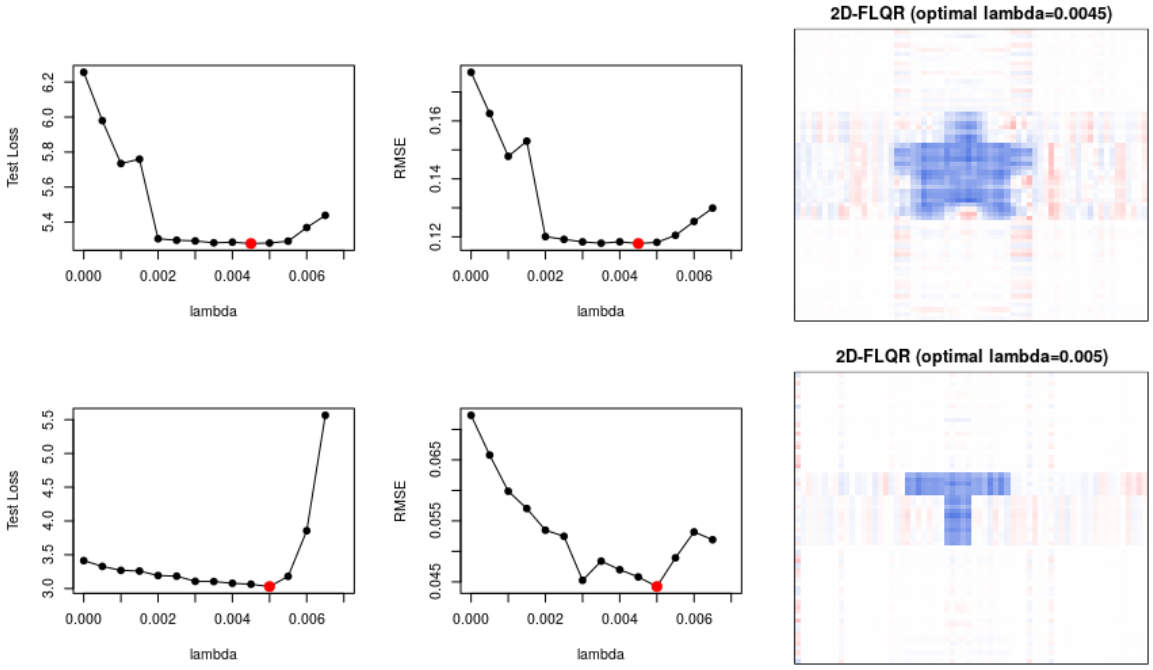

Figure 5.10: Test loss (left) and estimator RMSE (centre) as a function of regularization parameter λ for the star (top) and T (bottom) signals, with Tucker decomposition rank $R = 3$, $\tau = 0.5$, SNR $= 3$, and Gaussian error. The minimum in each graph is indicated in red. (Right) Tensor effect estimate $\hat{\mathbf{\Gamma}}$ for the 2D-FLQR model obtained at the optimal value of lambda, $\lambda = 0.0045$ and $\lambda = 0.005$, for the star and T signals, respectively. The colour scale indicates the estimated effect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue).

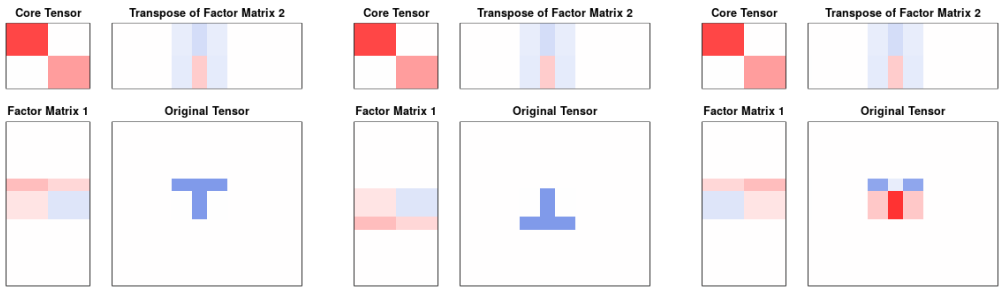

Figure 5.11: Exact Tucker decompositions (at decomposition ranks $R = 2$) for the T signal (left), vertically-reflected T signal (centre), and T signal with vertically-reflected second factor matrix (right). Only the first and last of these decompositions are factor equivalent. The colour scale indicates the effect of each pixel, from -1.5 (red) to 0 (white) to 1.5 (blue).

Figure 5.11 visualizes the exact Tucker decompositions for the T signal, the T signal transformed under a vertical reflection, and the T signal when its second factor matrix is vertically reflected. Among these three signals, only the first and last are factor equivalent.

The effect estimates in Figure 5.12 demonstrate that when factor equivalence holds, individual tensor effects can be adequately recovered when tensor observations are combined into one tensor (as per Subsection 3.3.1). On the other hand, if factor equivalence is incorrectly assumed (even when signals differ by only a linear transformation), we find individual effect estimates to be a mixture of the true signals.

5.2 Neuroimaging Application

We now consider an application of the qD-FLQR model to a real-world neuroimaging dataset derived from the ADNI [88]. The dataset used in this section consists of clinical information and neuroimaging data from 824 patients.

Clinical information includes the scalar covariates gender, handedness (left or right), marital status (married, widowed, divorced, or never married), years of education, binary retirement status, and age. We include all these variables in our analyses. The response of interest is *mini mental state examination* (MMSE) score—a convenient and widely-used criteria for screening neurodegenerative conditions such as dementia and Alzheimer's disease and for mon-

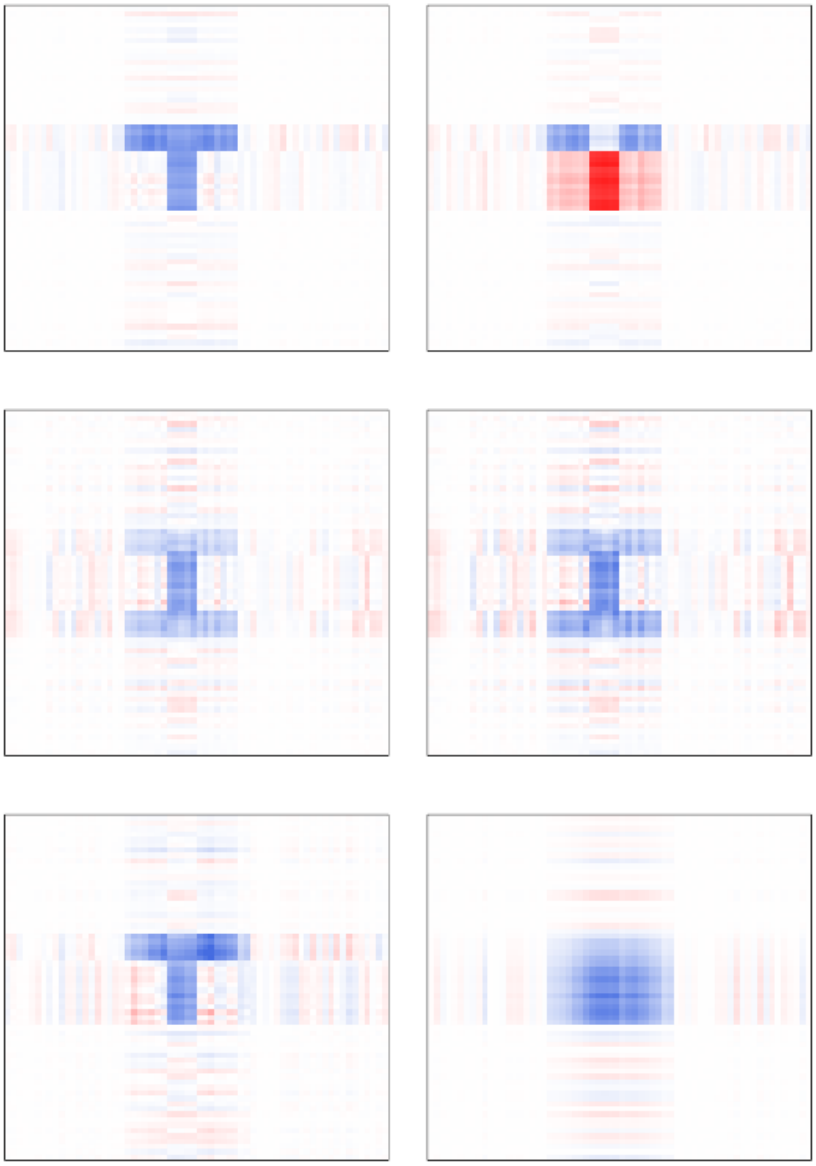

Figure 5.12: Examples of tensor effect estimates obtained for the 2D-FLQR model when two tensor covariates are combined into a single tensor (as per Subsection 3.3.1), with $R = 2$, $\tau = 0.5$, SNR $= 5$, Gaussian error, and $n = 1000$. (Top) Estimates when factor equivalence holds between the two tensor covariates. True signals are the left and right signals in Figure 5.11. (Middle) Estimates obtained when factor equivalence does not hold. True signals are the left and middle signals in Figure 5.11. (Bottom) Estimates when true signals are the (not factor equivalent) T and triangle signals, respectively.

itoring cognitive function over time [60]. MMSE is clinically evaluated as an integer from 0 to 30, inclusive, with high scores indicating normal cognitive function.

The functional neuroimaging data in this analysis is the same as that used by Wang et al. [87]. We highlight a few important contextual points here, but refer to the original paper for more detailed information on its extraction from raw imaging data. The dataset is derived from baseline T1-weighted MRI scans of the hippocampus and includes surface-based radial distance, (three) surface *multivariate tensor-based morphometry* (mTBM) features, and other Jacobian-based measures including the determinant and minimum and maximum eigenvalues. Each of these measures was computed along the surface of the hippocampus that, for each patient, was parameterized as a $30,000$-point mesh in two variables. Figure 5.13 visualizes this parameterization and illustrates how surface measures can be meaningfully interpreted as 3-dimensional tensors. Each side of the hippocampus yields a 150×100 matrix of measurements for each of the seven measures, resulting in a $\mathbb{R}^{7\times2\times150\times100}$ observation tensor for each patient.

Wang et al. [87] consider surface-based measures rather than the historically-used volume-based ones. Recent studies demonstrate numerous benefits of the former, which is better able to measure brain atrophy and, correspondingly, cognitive ability and other clinical outcomes. Wang et al. demonstrate that, together, radial distance and mTBM increase statistical power to detect neurodegenerative diseases relative to other Jacobian-based measures. Radial distance and the three mTBM measures (which we call *mTBM1*, *mTBM2*, and *mTBM3*) convey information on surface deformation in the hippocampal surface along the normal and tangent directions, respectively. Due to their complementary nature, we consider a model with these four functional variables.

We obtain an analytic sample of size $n = 798$ after cleaning and matching clinical and neuroimaging datasets. For each neuroimaging variable, we use Tucker decomposition ranks $R_1 = 2$ and $R_2 = R_3 = R$, where we apply 6-fold cross-validation to determine an optimal value of R. For computational

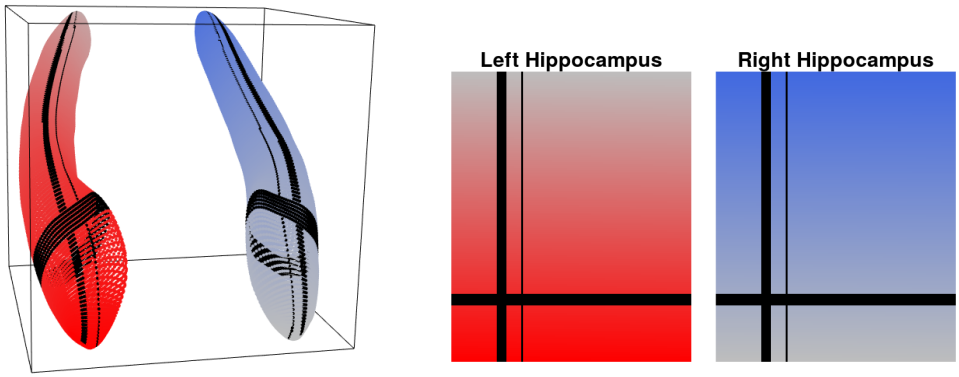

Figure 5.13: Visualization of our mapping from the hippocampal surface (left) to a 3-dimensional tensor ($\mathbb{R}^{2\times150\times100}$) (right). Each subplot on the right represents a 150×100 slice of the tensor observation matrix for a single neuroimaging measure. The colour gradient and black lines help illustrate the correspondence between the hippocampal surface and our parameterization. Specifically, the two 150×100 slices represents the left and right hippocampus. The rows and columns of these slices correspond to different level set rings around the hippocampal surface and position within these rings, respectively. Increasing column indices correspond to oppositely-oriented rotations around the surface, reflecting the stereoisometry of the left and right hippocampi.

convenience, we use a reduced dataset created by averaging neuroimaging measures in discrete 2×2 point partitions of the hippocampal surface, resulting in one $2 \times 75 \times 50$ tensor representing the hippocampal surface per variable per patient.

In the following analyses, we first obtain unregularized qD-FLQR model estimates. Results suggest the need for a tensor effect smoothness penalty, which we implement using fused LASSO. For comparison, we also consider a LASSO penalty despite fused LASSO being, *a priori*, better-motivated in practice.

5.2.1 Unregularized Estimation

Figure 5.15 shows that training loss decreases with increasing R in the (unregularized) neuroimaging model. This suggests that the model does indeed have the capacity to learn a relationship between the functional neuroimaging data and MMSE. In terms of 6-fold CV test error, however, we see very little

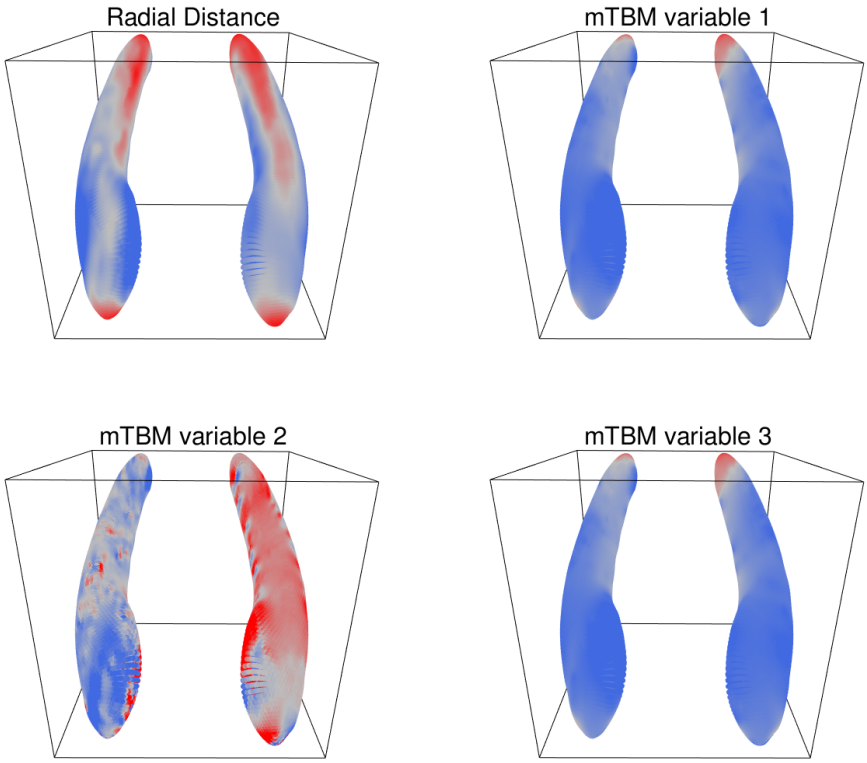

Figure 5.14: Visualization of the four functional variables considered in the model, as observed from a single patient. Radial distance is shown in the top-left subplot, while other subplots show the three mTBM measures. The colour scale differs for each subplot but ranges from red (for large, negative values) to grey (zero) to blue (large, positive values). Values are of similar numerical scale but are in different units, so we omit explicit gradient scales.

change across the models with $R = 1, \ldots, 5$ despite $R = 5$ having the lowest test error (but only by a minute margin). This suggests that $R = 1$ is sufficient for obtaining an estimate $\hat{\mathbf{\Gamma}}$ that is adequately predictive of MMSE across R. That the optimal R is not higher might also be due to low SNR. This result is not so surprising given our simulation studies in Section 5.1, specifically for the square or T signals, which exhibited constant or only slowly increasing test error for large R. Post-hoc analyses on the neuroimaging data show slowly increasing test error for $R \geq 6$ (not presented here), further supporting this conclusion.

Different algorithm settings do not influence these conclusions. Our convergence criteria is sufficiently strict and none of our fitted models reached the maximum allowed number of iterations. Taking a more slowly-decreasing sequence of smoothing parameters $(\nu_N)_N$ does not affect our conclusions. Different initialization schemes, such as randomly initializing *all* model parameters (rather than setting them to zero) at the start typically produce lower training error, but result in test error higher (up to 10-15 times as R increases beyond 5) than what we report here, suggesting greater susceptibility to overfitting. This makes sense given the backfitting nature of Algorithm 1, which would then be more prone to fit random noise.

Figure 5.16 compares tensor effect estimates for the unregularized $R = 1$ and $R = 5$ models. Higher R clearly allows for more complex estimates, although the amount of noise and variation in both is not amenable to interpretation. In a first attempt to address this, the next subsection applies the same LASSO penalty as Subsection 5.1.3. However, the unregularized estimates more strongly suggest that extreme variability across adjacent points on the hippocampal surface would be better addressed by a fused LASSO penalty. This is evident in the "checkerboard" patterns of Figure 5.16, suggesting that a more complex penalty should be used to control estimate smoothness. We explore this in Subsection 5.2.3 via fused LASSO penalty and continue to compare estimates for different R throughout.

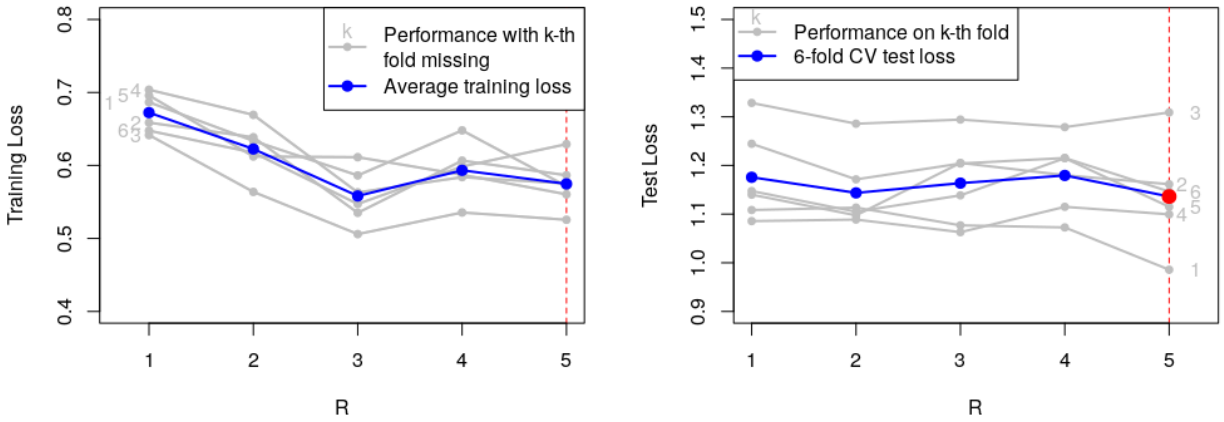

Figure 5.15: Training loss (left) and 6-fold CV test loss (right) when $R = 1, \ldots, 5$ for the unregularized (i.e., $\lambda = 0$) neuroimaging model. Grey lines (and associated numbers $k = 1, \ldots, 5$) indicate performance when the k-th fold is the test fold. Minimum test loss, occuring at $R = 5$, is indicated in red.

5.2.2 LASSO-Regularized Estimation

We apply LASSO penalization to the neuroimaging model with $R = 1$ and $R = 5$, as described in Subsection 3.3.2. An optimal value of $\lambda = 0.015$ and $\lambda = 0.010$, respectively, is obtained through 6-fold CV: Figure 5.17 displays training and test errors across various λ for the $R = 1$ model. Plots for $R = 3, 5$ are similar. Figure 5.18 illustrates differences between the optimal $R = 1$ and $R = 5$ tensor effect estimates.

Despite the LASSO penalty's elementwise application to $\hat{\mathbf{\Gamma}}$, we notice a substantial decrease in variability across adjacent points on the hippocampal surface. This is due to the structure assumed by the Tucker decomposition: elements of $\hat{\mathbf{\Gamma}}$ are penalized through Tucker decomposition parameters that, individually, control values of $\hat{\mathbf{\Gamma}}$ along a single row, column, or slice of our parameterization. This might suggest that the model is picking up on some true signal, although the still-present "checkerboard" pattern (albeit with larger regions) in the $R = 1$ model again suggests we impose a constraint on estimate smoothness. This pattern suggests that tensor effect estimates for this model are influenced by the choice of parameterization.

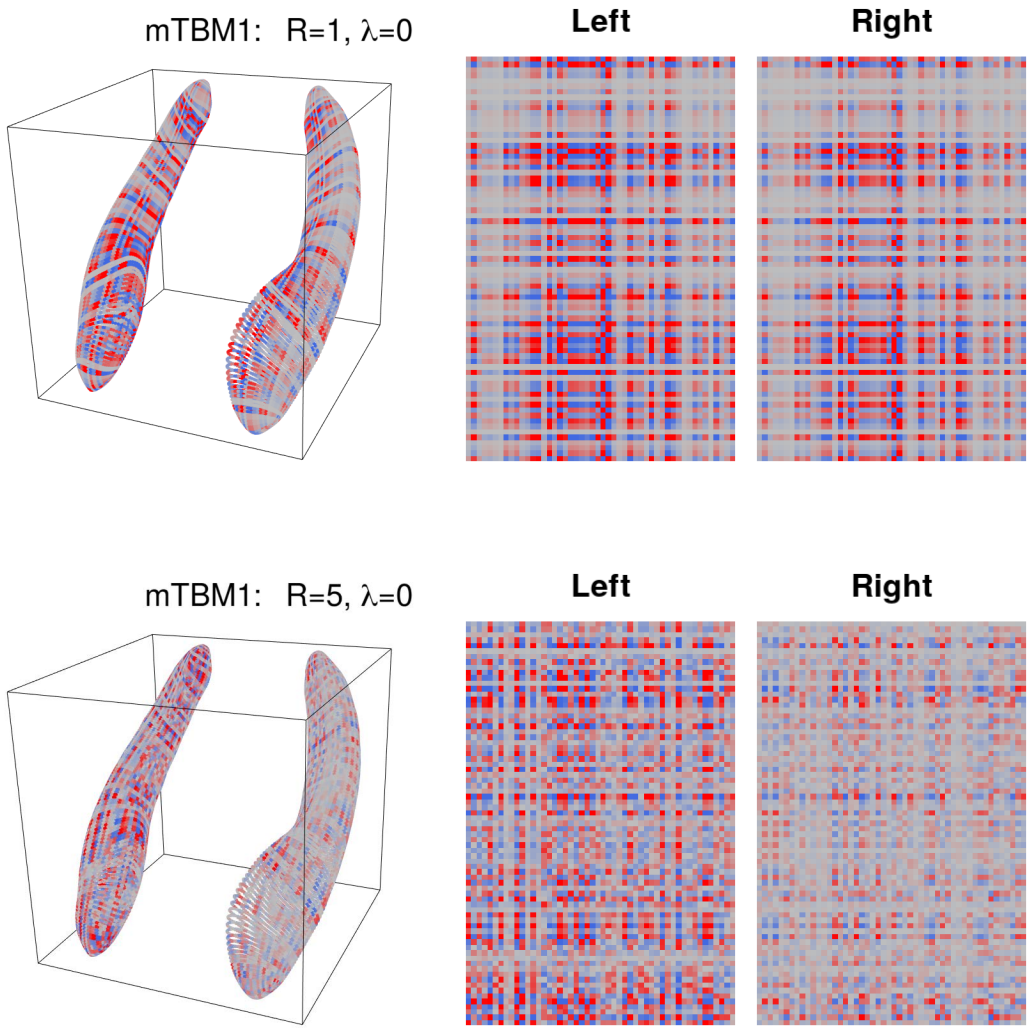

Figure 5.16: Comparison of the estimated effect of mTBM1, visualized on the hippocampal surface (left) and on our 3-dimensional parameterization (right), for the unregularized $R = 1$ (top) and $R = 5$ (bottom) neuroimaging models fit to the full dataset. Both subplots use the same colour gradient scale, so estimates are visually comparable.

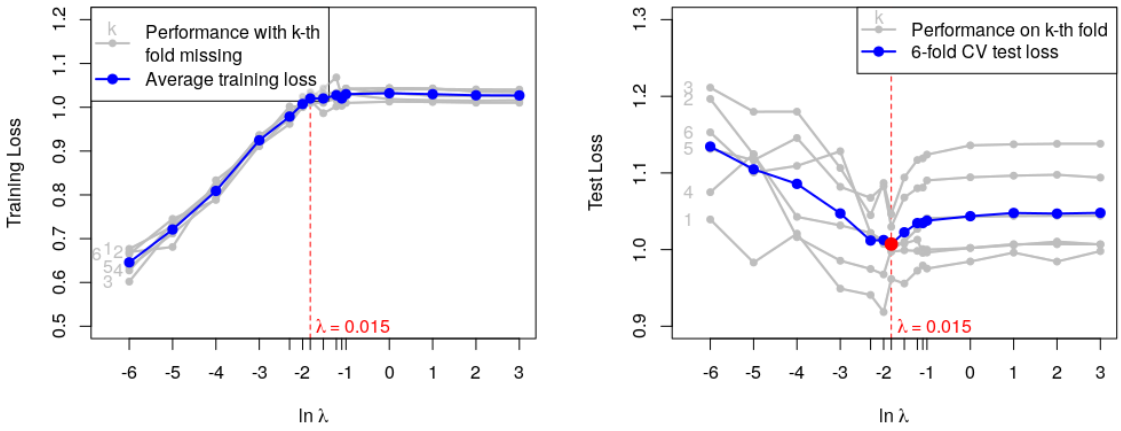

Figure 5.17: Training loss (left) and 6-fold CV test loss (right) as a function of the regularization parameter λ for the LASSO-penalized neuroimaging model with $R = 1$. Minimum test loss, occuring at $\lambda = 0.015$, is indicated in red.

Unregularized estimates in the previous subsection showed that higher R results in more complex estimates. This remains true with the LASSO penalty, as shown in Figure 5.18. The $R = 5$ estimate does not seem as restricted to our parameterization, although estimate's non-smoothness again motivates the fused LASSO penalty applied in the next subsection.

5.2.3 Enforcing Smoothness via Fused LASSO Penalty

The fused LASSO penalty is a well-known variant of LASSO as another L_1 penalty. In general, fused LASSO additionally penalizes absolute differences in pre-specified "fused" pairs of model parameters, whereas the LASSO penalty only penalizes parameter absolute values. For more detailed information, refer to the paper by Tibshrani et al. [79]. The fused LASSO encourages fused parameters to be similar in value. For computational simplicity, we do not incorporate the sparsity penalty (which sums absolute values) in our work here, as this would require another regularization hyperparameter λ_2 and would make exploration much more computationally intensive. Visually, our results below do not suggest the need to this sparsity penalty. We briefly outline the application to our neuroimaging data below, using the word *slice* to refer to

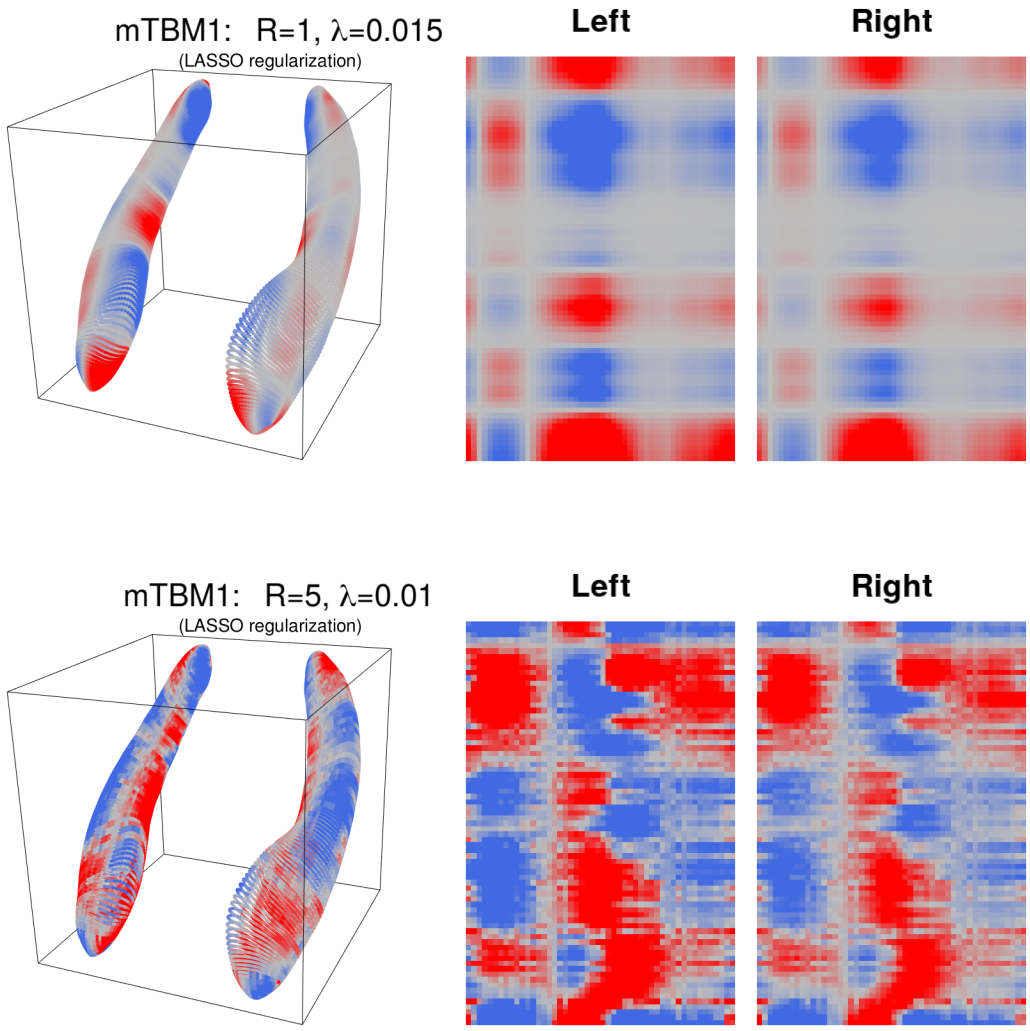

Figure 5.18: Comparison of the estimated effect of mTBM1, visualized on the hippocampal surface (left) and on our 3-dimensional parameterization (right), for the LASSO-regularized neuroimaging models with $R = 1$ ($\lambda = 0.03$) (top) and $R = 5$ ($\lambda = 0.03$) (bottom). Both subplots use the same colour gradient scale, so estimates are visually comparable.

a matrix formed by holding a constant position in the first dimension of a tensor.

Given the tensor effect estimate $\hat{\mathbf{\Gamma}} \in \mathbb{R}^{2\times I\times J}$ for the hippocampus data, define the fused LASSO penalty as

$$J(\hat{\mathbf{\Gamma}}) \triangleq \sum_{k=1}^{2}\sum_{i=1}^{I-1}\sum_{j=1}^{J}|\hat{\mathbf{\Gamma}}_{k,i,j} - \hat{\mathbf{\Gamma}}_{k,i+1,j}| + |\mathbf{\Gamma}_{k,i,j} - \hat{\mathbf{\Gamma}}_{k,i,j+1}|,$$

where $\hat{\mathbf{\Gamma}}_{k,i,J+1} \triangleq \hat{\mathbf{\Gamma}}_{k,i,1}$. As before, we approximate the absolute value $|\cdot|$ with $\rho_\nu \triangleq 2\rho_{\tau=0.5,\nu}$, denoting $\boldsymbol{\rho}_\nu$ as its elementwise application and J_ν the approximated loss. Thus, J penalizes differences between adjacent (not including diagonally-adjacent) elements in our tensor parameterization, with no penalty enforced between the the left and right hippocampi. The sum up to J and the definition of $\hat{\mathbf{\Gamma}}_{k,i,J+1}$ account for the fact that the first and J-th columns of each $I \times J$ slice are adjacent: each row in a slice corresponds to a closed loop on the hippocampal surface. Formally, define $\hat{\tilde{\mathbf{\Gamma}}} \triangleq (\hat{\mathbf{\Gamma}}_{k,i,j})_{k,i,j} \in \mathbb{R}^{2\times I\times(J+1)}$.

We briefly describe our implementation of the fused LASSO penalty. Denote $\boldsymbol{C} \triangleq [+1, -1] \in \mathbb{R}^{1\times 2}$ and $*$ the usual discrete convolutional operator applied along slices of the input image. This is the "same" convolutional operator that produces an output of the same size as its input (via zero-padding), given a discrete kernel. For simplicity, we do not "reflect" the kernel when applying a convolution with $*$, even though this is common convention. Define

$$\boldsymbol{C}^H \triangleq (\hat{\tilde{\mathbf{\Gamma}}} * \boldsymbol{C})_{[:,:,-(J+1)]} \in \mathbb{R}^{2\times I\times J}$$
$$\boldsymbol{C}^V \triangleq (\hat{\tilde{\mathbf{\Gamma}}} * \boldsymbol{C}^\top)_{[:,-I,:]} \in \mathbb{R}^{2\times(I-1)\times J},$$

where subscripts denote removal of the $(J+1)$-th column and I-th row of each slice. The (k,i,j)-th elements of $\boldsymbol{C}^H$ and $\boldsymbol{C}^V$ are thus $\hat{\mathbf{\Gamma}}_{k,i,j} - \hat{\mathbf{\Gamma}}_{k,i,j+1}$ and $\hat{\mathbf{\Gamma}}_{k,i,j} - \hat{\mathbf{\Gamma}}_{k,i+1,j}$, respectively. The loss J and its derivative with respect to $\hat{\mathbf{\Gamma}}$ depend only on the elements of these matrices.

It follows that

$$J_\nu(\hat{\mathbf{\Gamma}}) = \sum_{k,i,j}\boldsymbol{\rho}_\nu(\boldsymbol{C}^H)_{k,i,j} + \sum_{k,i,j}\boldsymbol{\rho}_\nu(\boldsymbol{C}^V)_{k,i,j},$$

Define $\boldsymbol{C}^{H+} \triangleq \boldsymbol{C}^H$, and $\boldsymbol{C}^{V+} \in \mathbb{R}^{2\times I\times J}$ equal to $\boldsymbol{C}^V$ but with slices augmented below by a zero row. These matrices each have (k,i,j)-th element

equal to one of the differences in J containing $+\hat{\boldsymbol{\Gamma}}_{k,i,j}$ (or 0 if such a difference does not exist). Similarly, define $\boldsymbol{C}^{H-} \in \mathbb{R}^{2\times I\times J}$ to be equal to $\boldsymbol{C}^{H}$ with slice columns permuted via $[1,\ldots,J] \mapsto [J,1,\ldots,J-1]$, and $\boldsymbol{C}^{V-} \in \mathbb{R}^{2\times I\times J}$ equal to $\boldsymbol{C}^{V}$ but with slices augmented above by a zero row. These matrices each have (k,i,j)-th element equal to one of the differences in J containing $-\hat{\boldsymbol{\Gamma}}_{k,i,j}$ (or 0 if such a difference does not exist).

From this, it is not hard to see that

$$\frac{\partial J_\nu}{\partial \hat{\boldsymbol{\Gamma}}} = \boldsymbol{\rho}_\nu{}'(\boldsymbol{C}^{H+}) + \boldsymbol{\rho}_\nu{}'(\boldsymbol{C}^{V+}) - \boldsymbol{\rho}_\nu{}'(\boldsymbol{C}^{H-}) - \boldsymbol{\rho}_\nu{}'(\boldsymbol{C}^{V-}).$$

As per Subsection 3.3.2, we have derived all expressions needed to implement the fused LASSO penalty (in a computationally convenient way). Our results follow for the $R = 1, 3, 5$ models.

Figure 5.19 plots training and 6-told CV test loss for the $R = 1, 3, 5$ models. Consistent with previous results, higher R results in marginally lower training error and little change in test error. Figure 5.20 visualizes tensor effect estimates at the optimal λ (separately cross-validated for each R).

The optimal $R = 1$ model still seems influenced by the parameterization (shown in the "checkerboard" pattern and the horizontal and vertical grey lines), although this effect seems to decrease as R increases. The effect of the fused LASSO penalty is visible in the smoother boundaries around the coloured regions.

On the other hand, the $R = 5$ model seems best able to break from the parameterized grid with curved nonlinear boundaries separating some areas of positive (blue) and negative (red) effect. With nearly no change in test error, the $R = 5$ model seems more practical and interpretable than the $R = 1$ models. Despite this, all models suggest the same general areas on the hippocampal surface to be associated with MMSE.

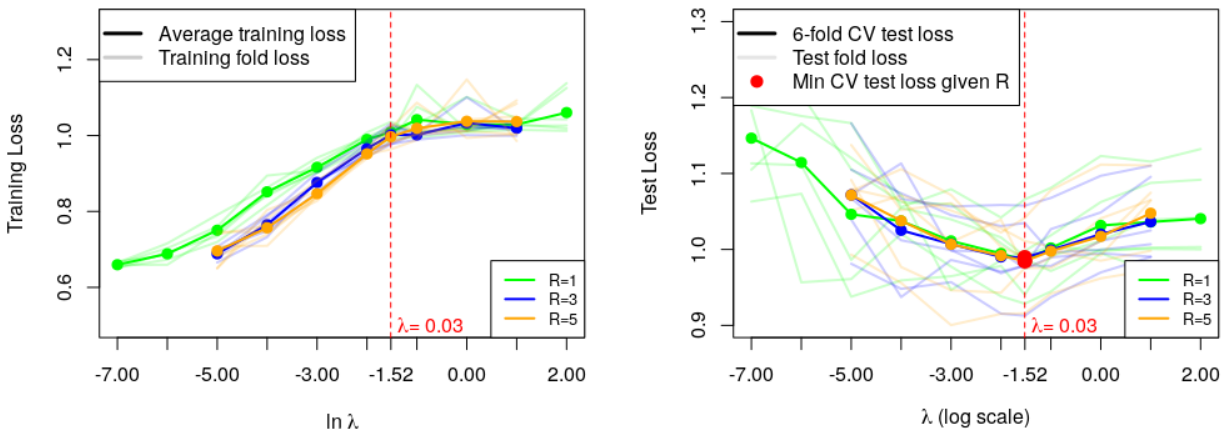

Figure 5.19: Training loss (left) and 6-fold CV test loss (right) as a function of the regularization parameter λ for the hippocampus model regularized via fused LASSO penalty. Tucker decomposition rank $R = 1, 3, 5$ is as indicated by the colour legend. Minimum test loss for each R is noted in red.

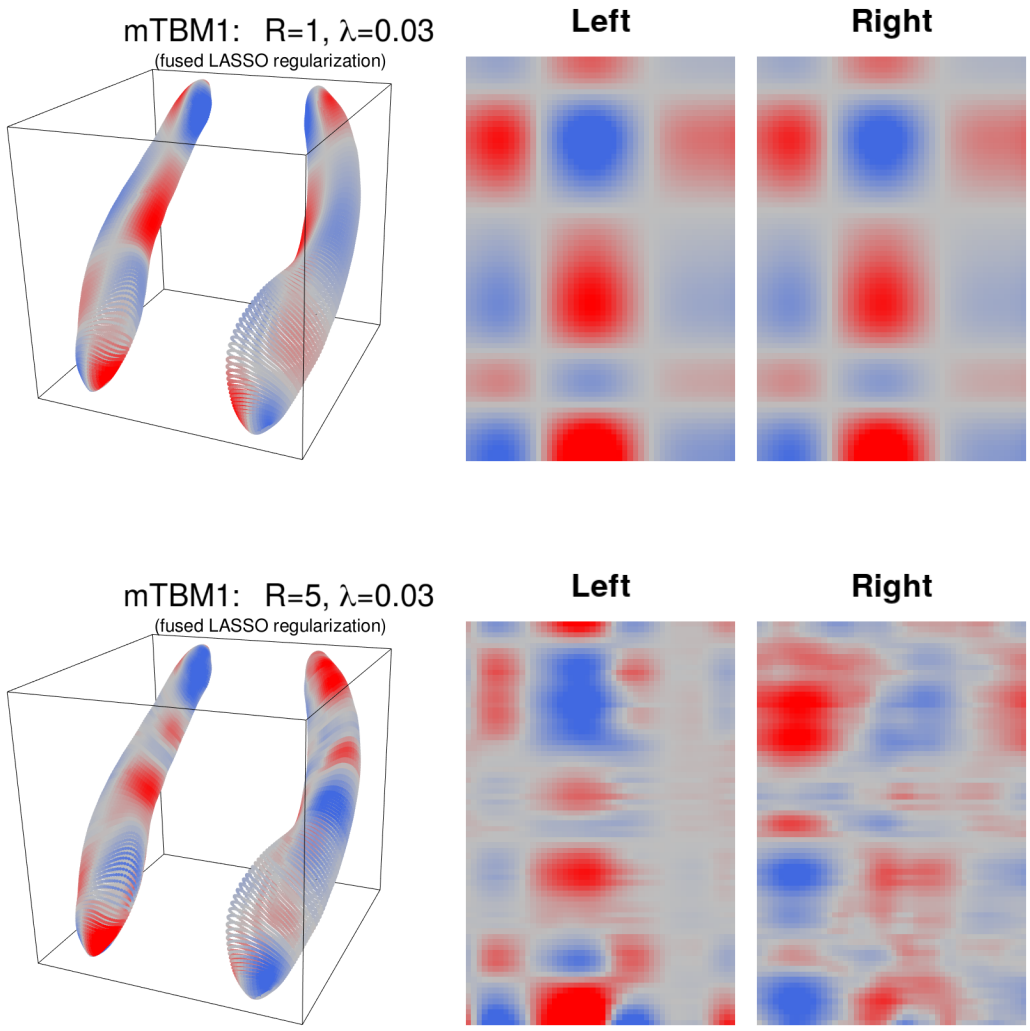

Figure 5.20: Comparison of the estimated effect of mTBM1, visualized on the hippocampal surface (left) and on our 3-dimensional parameterization (right), for the fused-LASSO-regularized neuroimaging models with $R = 1$ ($\lambda = 0.03$) (top) and $R = 5$ ($\lambda = 0.03$) (bottom). Both subplots use the same colour gradient scale, so estimates are visually comparable.

Chapter 6

Conclusion

The research presented in this thesis concerns quantile regression in functional data analysis. Specifically, we focused on incorporating functional variables with multidimensional input—observed on a discrete, uniform grid as tensors—into the quantile regression framework. Our work is motivated largely by tensor-valued neuroimaging data observed over a uniform spatial grid.

Our approach builds on the tensor GLM established by Zhou et al. [97], Li et al. [49], and Yu's doctoral work [90] on extensions to the quantile setting through the Tucker tensor decomposition [46], block relaxation techniques [47], and a smooth approximation to the quantile loss [12]. Building on their work, we provided a more rigorous formulation of the qD-FLQR model, a new estimation algorithm, and proofs of corresponding algorithmic and statistical properties which did not follow immediately from tensor GLM results. In our baseline/exploratory results, we have highlighted avenues for future work in this area, noted below.

The proposed algorithm uses block relaxation to split the non-convex model estimation problem into a fixed cycle of convex updates. Future work could improve algorithm efficiency by applying updates to parameter blocks most in need of improvement. As discussed by de Leeuw [47], this type of general block relaxation method, called *free-steering methods*, is less-studied in the literature and has fewer established theoretical guarantees. For generality, our algorithm currently updates the smoothing parameter ν_N as a function of only ν_{N-1}. However, previous works [12], [55] have laid out residual-dependent

rules for updating ν and convergence criteria specific to the smooth approximation used for the quantile loss. More work is required to incorporate this into the qD-FLQR model/algorithm and establish theoretical properties, but could improve computational efficiency in the long-run. Further improvements to computational efficiency might come from more complex optimization procedures such as Nesterov accelerated gradient descent or proximal gradient methods, which have garnered popularity in machine learning and non-convex optimization.

We further extended our model to allow multiple functional covariates and added the capacity for model regularization using a convex, differentiable penalty. We also enlarged the class of allowable penalties to ones approximable by convex, differentiable functions, particularly the LASSO and fused LASSO to address background estimate noise and non-smoothness, respectively. Further extending this framework to other non-convex and/or non-differentiable functions such as SCAD or MCP will be an important step forward. This development could be achieved through other optimization methods, as noted above, or by an IWLS approach as discussed by Zhou et al. [96] and Muggeo et al. [55].

In both simulation studies and real-world data analyses using neuroimaging data, we demonstrated the superiority of our method to existing tensor regression techniques, particularly when the error distribution is heavy-tailed. Our neuroimaging application showed the necessity of model regularization in producing interpretable tensor effect estimates. Future simulations or applied studies could examine model performance on different high-rank signals, such as the "butterfly signal" in Zhou et al [96]. Furthermore, different penalties could be explored and compared in greater detail. This could include both the sparsity and similarity penalties for the fused LASSO, of which the latter was only considered in this work.

References

[1] A. L. Alexander, K. Hasan, G. Kindlmann, D. L. Parker, and J. S. Tsuruda, "A geometric analysis of diffusion tensor measurements of the human brain," *Magnetic Resonance in Medicine*, vol. 44, no. 2, pp. 283–291, 2000.

[2] A. L. Alexander, J. E. Lee, M. Lazar, and A. S. Field, "Diffusion tensor imaging of the brain," *Neurotherapeutics*, vol. 4, no. 3, pp. 316–329, 2007.

[3] G. I. Allen, L. Grosenick, and J. Taylor, *A generalized least squares matrix decomposition*, 2011. eprint: `arXiv:1102.3074`.

[4] E. Anagnostou and M. J. Taylor, "Review of neuroimaging in autism spectrum disorders: What have we learned and where we go from here," *Molecular Autism*, vol. 2, no. 1, p. 4, 2011.

[5] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[6] P. Basser, J. Mattiello, and D. Lebihan, "MR diffusion tensor spectroscopy and imaging," *Biophysical Journal*, vol. 66, no. 1, pp. 259–267, 1994.

[7] P. Bellec, C. Chu, F. Chouinard-Decorte, Y. Benhajali, D. S. Margulies, and C. R. Craddock, "The Neuro Bureau ADHD-200 preprocessed repository," 2016.

[8] C. D. Boor, *A Practical Guide to Splines: With 32 Figures*. Springer, 2001.

[9] H. Cardot, F. Ferraty, and P. Sarda, "Spline estimators for the functional linear model," *Statistica Sinica*, vol. 13, pp. 571–591, 2003.

[10] H. Cardot, C. Crambes, and P. Sarda, "Quantile regression when the covariates are functions," *Journal of Nonparametric Statistics*, vol. 17, no. 7, pp. 841–856, 2005.

[11] H. Cardot, F. Ferraty, and P. Sarda, "Functional linear model," *Statistics & Probability Letters*, vol. 45, no. 1, pp. 11–22, 1999.

[12] C. Chen, "A finite smoothing algorithm for quantile regression," *Journal of Computational and Graphical Statistics*, vol. 16, no. 1, pp. 136–164, 2007.

[13] D. I. Clark and M. R. Osborne, "Finite algorithms for Huber's M-estimator," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 1, pp. 72–85, 1986.

[14] S. Cortese, C. Kelly, C. Chabernaud, E. Proal, A. D. Martino, M. P. Milham, and F. X. Castellanos, "Toward systems neuroscience of ADHD: a meta-analysis of 55 fMRI studies," *American Journal of Psychiatry*, vol. 169, no. 10, pp. 1038–1055, 2012.

[15] C. M. Crainiceanu, B. S. Caffo, S. Luo, V. M. Zipunnikov, and N. M. Punjabi, "Population value decomposition, a framework for the analysis of image populations," *Journal of the American Statistical Association*, vol. 106, no. 495, pp. 775–790, 2011.

[16] J. Dauxois, A. Pousse, and Y. Romain, "Asymptotic theory for the principal component analysis of a vector random function: Some applications to statistical inference," *Journal of Multivariate Analysis*, vol. 12, no. 1, pp. 136–154, 1982.

[17] C. Davatzikos, "Spatial normalization of 3D brain images using deformable models," *Journal of Computer Assisted Tomography*, vol. 20, no. 4, pp. 656–665, 1996.

[18] Y. Dodge and J. Whittaker, "Partial quantile regression," *Metrika*, vol. 70, no. 1, pp. 35–57, 2008.

[19] J. S. Elam and D. V. Essen, "Human Connectome Project," *Encyclopedia of Computational Neuroscience*, pp. 1408–1411, 2015.

[20] J. Fan and G. Irène, *Local Polynomial Modelling and its Applications*. CRC Press, 2003.

[21] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[22] J. C. Fiorot and P. Huard, "Composition and union of general algorithms of optimization," *Mathematical Programming Studies Point-to-Set Maps and Mathematical Programming*, pp. 69–85, 1979.

[23] N. C. Fox, R. I. Scahill, W. R. Crum, and M. N. Rossor, "Correlation between rates of brain atrophy and cognitive decline in AD," *Neurology*, vol. 52, no. 8, pp. 1687–1687, 1999.

[24] F. Frédéric and P. Vieu, *Nonparametric Functional Data Analysis: Theory and Practice*. Springer, 2011.

[25] M. Fredrikson, "Imaging genetics of anxiety disorders," *Neuroimaging Genetics*, pp. 223–232, 2016.

[26] T. Gasser, H.-G. Muller, W. Kohler, L. Molinari, and A. Prader, "Nonparametric regression analysis of growth curves," *The Annals of Statistics*, vol. 12, no. 1, pp. 210–229, 1984.

[27] C. Goutis, "Second-derivative functional regression with applications to near infra-red spectroscopy," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 60, no. 1, pp. 103–114, 1998.

[28] U. Grenander, "Stochastic processes and statistical inference," *Arkiv för Matematik*, vol. 1, no. 3, pp. 195–277, 1950.

[29] B. Grignon, L. Mainard, M. Delion, C. Hodez, and G. Oldrini, "Recent advances in medical imaging: Anatomical and clinical applications," *Surgical and Radiologic Anatomy*, vol. 34, no. 8, pp. 675–686, 2012.

[30] ——, "Recent advances in medical imaging: Anatomical and clinical applications," *Surgical and Radiologic Anatomy*, vol. 34, no. 8, pp. 675–686, 2012.

[31] P. Z. Hadjipantelis and H.-G. Müller, "Functional data analysis for big data: A case study on california temperature trends," *Handbook of Big Data Analytics Springer Handbooks of Computational Statistics*, pp. 457–483, 2018.

[32] P. Hall and J. L. Horowitz, "Methodology and convergence rates for functional linear regression," *The Annals of Statistics*, vol. 35, no. 1, pp. 70–91, 2007.

[33] T. Hastie, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

[34] I. Helland, "Partial least squares regression," *Encyclopedia of Statistical Sciences*, 2004.

[35] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, pp. 164–189, 1927.

[36] N. L. Hjort and D. Pollard, *Asymptotics for minimisers of convex processes*, 2011. eprint: `arXiv:1107.3806`.

[37] P. D. Hoff, "Hierarchical multilinear models for multiway data," *Computational Statistics & Data Analysis*, vol. 55, no. 1, pp. 530–543, 2011.

[38] Y. Hu, X. He, J. Tao, and N. Shi, "Modeling and prediction of children's growth data via functional principal component analysis," *Science in China Series A: Mathematics*, vol. 52, no. 6, pp. 1342–1350, 2009.

[39] P. J. Huber, "Robust regression: Asymptotics, conjectures and Monte Carlo," *The Annals of Statistics*, vol. 1, no. 5, pp. 799–821, 1973.

[40] S. A. Huettel, A. W. Song, and G. McCarthy, *Functional Magnetic Resonance Imaging*. Sinauer Associates, 2014.

[41] C. R. Jack, M. Slomkowski, S. Gracon, T. M. Hoover, J. P. Felmlee, K. Stewart, Y. Xu, M. Shiung, P. C. Obrien, R. Cha, D. Knopman, and R. C. Petersen, "MRI as a biomarker of disease progression in a therapeutic trial of milameline for AD," *Neurology*, vol. 60, no. 2, pp. 253–260, 2003.

[42] G. M. James, J. Wang, and J. Zhu, "Functional linear regression that's interpretable," *The Annals of Statistics*, vol. 37, no. 5A, pp. 2083–2108, 2009.

[43] K. Kato, "Estimation in functional linear quantile regression," *The Annals of Statistics*, vol. 40, no. 6, pp. 3108–3136, 2012.

[44] R. Koenker, *quantreg: Quantile regression*, R package version 5.36, 2018. [Online]. Available: `https://CRAN.R-project.org/package=quantreg`.

[45] R. Koenker and G. Bassett, "Regression quantiles," *Econometrica*, vol. 46, no. 1, pp. 33–50, 1978.

[46] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[47] J. D. Leeuw, "Block-relaxation algorithms in statistics," *Studies in Classification, Data Analysis, and Knowledge Organization Information Systems and Data Analysis*, pp. 308–324, 1994.

[48] N. Lepore, C. Brun, Y.-Y. Chou, M.-C. Chiang, R. Dutton, K. Hayashi, E. Luders, O. Lopez, H. Aizenstein, A. Toga, J. T. Becker, and P. M. Thompson, "Generalized tensor-based morphometry of HIV/AIDS using multivariate statistics on deformation tensors," *IEEE Transactions on Medical Imaging*, vol. 27, no. 1, pp. 129–141, 2008.

[49] X. Li, D. Xu, H. Zhou, and L. Li, "Tucker tensor regression and neuroimaging analysis," *Statistics in Biosciences*, vol. 10, no. 3, pp. 520–545, 2018.

[50] Y. Lu, J. Du, and Z. Sun, "Functional partially linear quantile regression model," *Metrika*, vol. 77, no. 2, pp. 317–332, 2013.

[51] K. Madsen and H. B. Nielsen, "A finite smoothing algorithm for linear l_1 estimation," *SIAM Journal on Optimization*, vol. 3, no. 2, pp. 223–235, 1993.

[52] N. Makris, A. J. Worth, G. M. Papadimitriou, J. W. Stakes, V. S. Caviness, D. N. Kennedy, D. N. Pandya, E. Kaplan, A. G. Sorensen, O. Wu, and et al., "Morphometry of in vivo human white matter association pathways with diffusion-weighted magnetic resonance imaging," *Annals of Neurology*, vol. 42, no. 6, pp. 951–962, 1997.

[53] N. Malfait and J. O. Ramsay, "The historical functional linear model," *Canadian Journal of Statistics*, vol. 31, no. 2, pp. 115–128, 2003.

[54] J. S. Morris, "Functional regression," *Annual Review of Statistics and Its Application*, vol. 2, no. 1, pp. 321–359, 2015.

[55] V. M. Muggeo, M. Sciandra, and L. Augugliaro, "Quantile regression via iterative least squares computations," *Journal of Statistical Computation and Simulation*, vol. 82, no. 11, pp. 1557–1569, 2012.

[56] H.-G. Muller, Y. Wu, and F. Yao, "Continuously additive models for nonlinear functional regression," *Biometrika*, vol. 100, no. 3, pp. 607–622, 2013.

[57] H.-G. Muller, "Functional modelling and classification of longitudinal data," *Scandinavian Journal of Statistics*, vol. 32, no. 2, pp. 223–240, 2005.

[58] F. S. Nathoo, L. Kong, and H. Zhu, "A review of statistical methods in imaging genetics," *Canadian Journal of Statistics*, vol. 47, no. 1, pp. 108–131, 2019.

[59] A. M. Ostrowski, *Solution of Equations and Systems of Equations*. Academic Press, 1966.

[60] V. C. Pangman, J. Sloan, and L. Guse, "An examination of psychometric properties of the mini-mental state examination and the standardized mini-mental state examination: Implications for clinical practice," *Applied Nursing Research*, vol. 13, no. 4, pp. 209–213, 2000.

[61] L. Pezawas and A. Meyer-Lindenberg, "Imaging genetics: Progressing by leaps and bounds," *NeuroImage*, vol. 53, no. 3, pp. 801–803, 2010.

[62] M. Pietrosanu, J. Gao, L. Kong, B. Jiang, and D. Niu, *cqrReg: An r package for quantile and composite quantile regression and variable selection*, 2017. eprint: `arXiv:1709.04126`.

[63] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2018. [Online]. Available: `https://www.R-project.org/`.

[64] J. O. Ramsay, "When the data are functions," *Psychometrika*, vol. 47, no. 4, pp. 379–396, 1982.

[65] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*. Springer, 2010.

[66] J. O. Ramsay, H. Wickham, S. Graves, and G. Hooker, *Fda: Functional data analysis*, R package version 2.4.8, 2018. [Online]. Available: `https://CRAN.R-project.org/package=fda`.

[67] P. T. Reiss, J. Goldsmith, H. L. Shang, and R. T. Ogden, "Methods for scalar-on-function regression," *International Statistical Review*, vol. 85, no. 2, pp. 228–249, 2016.

[68] P. T. Reiss and R. T. Ogden, "Functional principal component regression and functional partial least squares," *Journal of the American Statistical Association*, vol. 102, no. 479, pp. 984–996, 2007.

[69] J. A. Rice and B. W. Silverman, "Estimating the mean and covariance structure nonparametrically when the data are curves," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 53, no. 1, pp. 233–243, 1991.

[70] R. Rojas, R. Riascos, D. Vargas, H. Cuellar, and J. Borne, "Neuroimaging in drug and substance abuse part i," *Topics in Magnetic Resonance Imaging*, vol. 16, no. 3, pp. 231–238, 2005.

[71] T. J. Rothenberg, "Identification in parametric models," *Econometrica*, vol. 39, no. 3, pp. 577–591, 1971.

[72] B. Sánchez, H. Lachos, and V. Labra, "Likelihood based inference for quantile regression using the asymmetric Laplace distribution," *Journal of Statistical Computation and Simulation*, vol. 81, pp. 1565–1578, 2013.

[73] X.-H. Shao, H.-L. Shen, and C.-J. Li, "Applications of stair matrices and their generalizations to iterative methods," *Applied Mathematics and Mechanics*, vol. 27, no. 8, pp. 1115–1121, 2006.

[74] S.-K. Song, S.-W. Sun, M. J. Ramsbottom, C. Chang, J. Russell, and A. H. Cross, "Dysmyelination revealed through MRI as increased radial (but unchanged axial) diffusion of water," *NeuroImage*, vol. 17, no. 3, pp. 1429–1436, 2002.

[75] *Statistics & science - a report of the London workshop on the future of the statistical sciences*. [Online]. Available: `https://www.worldofstatistics.org/wos/pdfs/Statistics%5C&Science-TheLondonWorkshopReport.pdf`.

[76] M. Styner, "Boundary and medial shape analysis of the hippocampus in schizophrenia," *Medical Image Analysis*, vol. 8, no. 3, pp. 197–203, 2004.

[77] Q. Tang and L. Cheng, "Partial functional linear quantile regression," *Science China Mathematics*, vol. 57, no. 12, pp. 2589–2608, 2014.

[78] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[79] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[80] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[81] J. A. Turner, P. Smyth, F. Macciardi, J. H. Fallon, J. L. Kennedy, and S. G. Potkin, "Imaging phenotypes and genotypes in schizophrenia," *Neuroinformatics*, vol. 4, no. 1, pp. 21–50, 2006.

[82] A. W. van der Vaart, *Asymptotic Statistics*. Cambridge Univ. Press, 2007.

[83] M. P. Wand and M. C. Jones, *Kernel Smoothing*. Chapman & Hall, 2011.

[84] J.-L. Wang, J.-M. Chiou, and H.-G. Mueller, "Review of functional data analysis," 2015.

[85] Y. Wang, X. Gu, T. Chan, A. Toga, and P. Thompson, "Multivariate statistics of tensor-based cortical surface morphometry," *NeuroImage*, vol. 47, 2009.

[86] Y. Wang, T. F. Chan, A. W. Toga, and P. M. Thompson, "Multivariate tensor-based brain anatomical surface morphometry via holomorphic one-forms," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009 Lecture Notes in Computer Science*, pp. 337–344, 2009.

[87] Y. Wang, Y. Song, P. Rajagopalan, T. An, K. Liu, Y.-Y. Chou, B. Gutman, A. W. Toga, and P. M. Thompson, "Surface-based TBM boosts power to detect disease effects on the brain: An N=804 ADNI study," *NeuroImage*, vol. 56, no. 4, pp. 1993–2010, 2011.

[88] M. W. Weiner, D. P. Veitch, P. S. Aisen, L. A. Beckett, N. J. Cairns, J. Cedarbaum, R. C. Green, D. Harvey, C. R. Jack, J. Jagust William nd Luthman, J. C. Morris, R. C. Petersen, A. J. Saykin, L. Shaw, L. Shen, A. Schwarz, A. W. Toga, and J. Q. Trojanowski, "2014 update of the Alzheimer's disease neuroimaging initiative: A review of papers published since its inception," *Alzheimers & Dementia*, vol. 11, no. 6, 2015.

[89] H. Wold, "Soft modelling by latent variables: The non-linear iterative partial least squares (NIPALS) approach," *Journal of Applied Probability*, vol. 12, no. S1, pp. 117–142, 1975.

[90] D. Yu, "Quantile regression methods in functional data analysis," PhD thesis, University of Alberta Education & Research Archive, 2017.

[91] D. Yu, L. Kong, and I. Mizera, "Partial functional linear quantile regression for neuroimaging data analysis," *Neurocomputing*, vol. 195, pp. 74–87, 2016.

[92] K. Yu and R. A. Moyeed, "Bayesian quantile regression," *Statistics & Probability Letters*, vol. 54, no. 4, pp. 437–447, 2001.

[93] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, vol. 38, no. 2, pp. 894–942, 2010.

[94] X. Zhang, J.-M. Chiou, and Y. Ma, "Functional prediction through averaging estimated functional linear regression models," *Biometrika*, 2018.

[95] Y. Zhao, R. T. Ogden, and P. T. Reiss, "Wavelet-based LASSO in functional linear regression," *Journal of Computational and Graphical Statistics*, vol. 21, no. 3, pp. 600–617, 2012.

[96] H. Zhou and L. Li, "Regularized matrix regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 2, pp. 463–483, 2013.

[97] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *Journal of the American Statistical Association*, vol. 108, no. 502, pp. 540–552, 2013.

[98] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[99] H. Zou and M. Yuan, "Composite quantile regression and the oracle model selection theory," *The Annals of Statistics*, vol. 36, no. 3, pp. 1108–1126, 2008.