DISTRIBUTION REGRESSION: THEORY AND APPLICATION

by

Bohao Tang

A dissertation submitted to Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
January 2023

Abstract

In this dissertation we discuss the problem of distribution regression. That is, the problem of utilizing distributional covariates in predicting scalar outcomes. We first show an application in neuroimaging that relates functional connectivity measurements viewed as statistical distributions to outcomes. We consider 47 primary progressive aphasia (PPA) patients with various levels of language ability. These patients were randomly assigned to two treatment arms, tDCS (transcranial direct-current stimulation and language therapy) vs sham (language therapy only), in a clinical trial. We analyze the effect of direct stimulation on functional connectivity by treating connectivity measures as samples from individual distributions. As such, we estimate the density of correlations among the regions of interest (ROIs) and study the difference in the density post-intervention between treatment arms. This distributional approach gives the ability to drastically reduces the number of multiple comparisons compared to classic edge-wise analysis. In addition, it allows for the investigation of the impact of functional connectivity on the outcomes where the connectivity is not geometrically localized.

We next propose and study the theoretical properties of a related functional expectation model, where we show that optimal information rate bounds can be achieved by a distributional Gaussian process regression, without estimating any individual densities. The model can perform closed form posterior inference via a

Gaussian process prior on the regression function. We also propose a low-rank approximation method to accelerate the inference in real applications.

In the next chapter, we attached a less related work that reviews state-of-art algorithms to accelerate the convergence of fixed-point iteration problems. Fixed point iteration algorithms have a wide range of applications in statistics and data science. We propose a modified restart Nesterov accelerated gradient algorithm that can also be used for black-box acceleration of general fixed-point iteration problems and show that works well in practice via investigation under six different tasks.

Thesis Committee

Primary Readers

Brian S. Caffo (Primary Advisor)
Professor
Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Abhirup Datta
Associate Professor
Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Thomas Hartung
Professor
Department of Environmental Health and Engineering
Johns Hopkins Bloomberg School of Public Health

Joshua Ewen
Associate Professor
Department of Neurology
Johns Hopkins University School of Medicine

Alternate Readers

Vadim Zipunnikov
Associate Professor
Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Elizabeth Platz
Professor
Department of Epidemiology
Johns Hopkins Bloomberg School of Public Health

Acknowledgments

I would like to express my deep appreciation to my advisor Brian S. Caffo. The completion of this dissertation would not have been possible without his support.

Let me also express my sincere gratitude to Ravi Varadhan for his assistance on the work in Chapter 4 and for all the valuable discussions between us during my Ph.D. program. Also I would like to thank Abhirup Datta for his valuable guidance and, especially, for his assistance with Chapter 3.

I would like to extend my sincere thanks to all my committee members: Thomas Hartung, Joshua Ewen, Vadim Zipunnikov and Elizabeth Platz for their helpful advice and guidance to my thesis work.

Finally, I would also like to express my deepest gratitude to my parents and my girlfriend He, Ting for their invaluable support through my Ph.D. student life!

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This dissertation is primarily concerned with the distribution regression problem. This is the setting where the covariates from an analysis are distributional objects. In Chapter 2 we show how this kind of problem naturally arises in functional magnetic resonance imaging (fMRI) when one considers non-localized effects as a form of omnibus starting point. As an application, we consider 47 primary progressive aphasia (PPA) patients with various levels of language abilities. These patients were randomly assigned to two treatment arms, tDCS (transcranial direct-current stimulation and language therapy) vs sham (language therapy only), in a clinical trial. We utilize the connectivity measures as of they are samples from individual distributions. We estimate the density of correlations among the regions of interest (ROIs) and study the difference in the density post-intervention between treatment arms. We discover that it is the tail of the density, rather than the mean or lower order moments of the distribution, that demonstrates a significant impact in the classification.

In Chapter 3, we turn to study the theoretical properties of the same sort of distribution regression problems that from Chapter 2. However, we extend the

approach to propose a variant of Gaussian process regression that can be used for distributional covariates and show that the posterior contracts in optimal rates given the regression function being smooth to certain degree. The procedure we propose does not require one to estimate any individual densities and close form inference can be performed with only samples of subject distribution we observed. We also develop a low-rank approximation method to accelerate the algorithm that could be used in real life application where exact inference is not needed.

In Chapter 4, we take a different direction towards general purpose algorithms that are applicable in nearly any complex statistical computational setting using fixed point algorithms. Here, we perform a study that reviews the state-of-art black-box acceleration algorithms for fixed-point problems, which characterized many, if not most, statistics and data science optimization algorithms. We also propose a variant of the restart Nesterov accelerated gradient algorithm that can also be used to accelerate general fixed-point problems. We test all the algorithm with six tasks, tailored to probe performance along important benchmark domains that connect with some of the most popular uses of these algorithms. This include multivariate-t distribution estimation, mixture distributions estimation, LASSO, variational inference, matrix balancing and manifold embedding.

Chapter 2

Differences in functional connectivity distribution after transcranial direct-current stimulation: a connectivity density point of view

Bohao Tang [1], *Yi Zhao* [2], *Archana Venkataraman* [3], *Kyrana Tsapkini* [4], *Martin A Lindquist*[1], *James Pekar*[5], *Brian Caffo*[1]

2.1 Introduction

The study of resting state brain connectivity via functional magnetic resonance imaging (fMRI) involves the investigation of correlations between cortical seeds, regions or voxels (henceforth referred to as foci). Friston, in particular, defined functional connectivity as the correlations, over time, between spatially distinct brain regions (Friston, 2011). Nearly all inter-subject investigations of connectivity

[1] Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health
[2] Department of Biostatistics, Indiana University School of Medicine
[3] Department of Electrical and Computer Engineering, Johns Hopkins University
[4] Department of Neurology, Johns Hopkins School of Medicine
[5] Department of Radiology and Radiological Science, Johns Hopkins School of Medicine

have focused on *localized correlations*. That is, they consider correlations between foci treated consistently across subjects. Mathematically, this can be described as saying that the methods are not invariant to subject-specific relabeling of the foci. In fact, for most methods, such as pairwise regressions on correlations across subjects or decomposition methods, shuffling foci labels within subjects is a form of null distribution. Furthermore, this lack of invariance applies regardless of the degree of granularity of the analysis, from regions to seeds to voxels (Friston, 2011; Damoiseaux and Greicius, 2009; Bastos and Schoffelen, 2016). The methods and choice of granularity all center the focus on geographic consistency of correlations across groups of similar subjects. Individual topography (Kong et al., 2019) and functional connectivity alignment (Haxby et al., 2020) are another set of methods that allow for spatially inconsistent relationships beyond subject-specific structure. However, their effort of finding subject specific parcellation / transformation is still for the purpose of localization. Other exceptions include many variations of graph theory based methods, where graphical features may not be localized across subjects in the sense of summarising multiple connections (Shen et al., 2017) or being invariant to subject-specific foci labels (Koutra, Vogelstein, and Faloutsos, 2013; Vogelstein et al., 2012).

To illustrate the idea of label invariance, consider a scenario where one reduces the connectivity measures to subject-specific binary graphs (by thresholding). If the effect of the graphs on the outcomes is invariant to the nodes (foci) corresponding to the edges, then clearly it is sufficient to know the number of edges that are present for each subject's graph, since given that information one can create the set of equivalent graphs under node invariance. This is equivalent to saying the relationship between the outcome and connectivity graph, is solely dependent

on the estimated probability distribution for the edges under an assumed independent and identically distributed edge distribution, since that distribution only depends on the total number of edges. (This is the Erdős-Rényi random graph model.) Our approach formally builds on this idea. But we further consider a random weighted graph model rather than thresholding to obtain binary edges, and proposes a specific functional linear model for the relationship between outcomes and the connectivity density.

We demonstrate the benefits of using the distribution of resting state correlations as covariates using functional data analysis tools. We suggest the use of the quantile density, the density of connections evaluated at evenly spaced quantiles of the connections, as this improves performance. Regardless of these choices, utilizing connectivity density regression has several benefits. A primary one is the relaxation of the consistent localization assumption across subjects. In the appendix, we demonstrate mathematically how connection densities achieve this invariance. Localization analyses makes the, often unchallenged, assumption that pairs of foci represent the same correlated functional specialization across exchangeable subjects. This assumption is grounded in the neurological theory of functional specialization dating back to the foundational works of Broca and Weirnicke (Broca, 1861; Wernicke, 1874). However, it is clear that in specific applications and biological settings, the neural geography of functional specialization can vary. As an extreme example, subjects with brain damage in their youth often have the neuroplasticity that remaps a function to atypical areas (Finger and Almli, 1985).

Hyperalignment (Haxby et al., 2020) also allows for a high degree of subject-specific functional specialization. However, unlike connectivity density regression,

localization remains the goal in hyperalignment, and therefore, a multi-parameter alignment transformation must be estimated per subject. Connectivity density analysis can be seen as a complementary, technique that does not require estimation of subject specific alignment. Further, focusing on connectivity densities drastically simplifies the problem and reduces multiplicity concerns. Of course, these benefits come at the cost of not considering potentially relevant localization information, and so the technique can not be more sensitive to the detection of localized effects with a reduced search space and correct a priori localization hypotheses. It would be accurate to say that focusing on connectivity densities in analysis lies at one end of the spectrum of model localization assumptions, whereas pair at a time models lie at the other extreme and hyperalignment lying somewhere in the middle.

There are existing studies that utilize the distribution of resting state correlations. For example, Petersen, Müller, et al., 2016 consider the distribution of correlations between a seed voxel and all other voxels within a region of interest (ROI), to summarise the ROI state. Also, Scheinost et al., 2012 further considered such distributions across all pairs of voxels. This work derived a degree function from the connection density as a summary of the connectivity of each voxel. As a result, these studies continue to focus on localized effects, where the use of the connectivity density is mainly to achieve a more informative localized summary of brain connectivity.

This study is motivated by a resting-state fMRI study of primary progressive aphasia (PPA) patients, where it is feasible to want to relax the geometric localization assumption. In the study, the patients were randomly assigned into two treatment groups, 1) tDCS (transcranial direct-current stimulation Nitsche et al.,

2008) and language therapy versus 2) a sham tDCS and language therapy only. In the tDCS group, the nominal stimulation target was the left inferior frontal gyrus (IFG). Since the actual area of stimulation my vary, even if only slightly, it is relevant to consider models that are less dependent on localization. In addition, the stimulation electrode patches were size of $5 \times 5 = 25$ cm^2. Thus, the stimulation areas may have extended beyond the left IFG in a way that may induce additional variation across subjects that would also motivate considering techniques that are robust to violations of localization assumptions. Here, we propose a novel approach to represent the effect of stimulation on functional connectivity. By ignoring spatial heterogeneity, we directly study the change on the distribution of correlation between the ROIs.

The manuscript is organized as follows. In Section 2, the experimental design and approach are introduced. Results both for simulated and real data are shown in Section 3. Section 4 contains a summary and discussion.

2.2 Material and Methods

2.2.1 Experimental Design

The data analyzed in this study were part of a larger randomized, double-blinded, sham-controlled, crossover study on aphasia treatment using tDCS. All of the analyzed subjects had at least two years of progressive language deficit and no history of any other neurological condition that may have affected their language ability. Subjects had atrophy predominantly in the left hemisphere. Subjects were diagnosed via neuropsychological testing, language testing, MRI and clinical assessment according to consensus criteria (Gorno-Tempini et al., 2011). The study

was approved by the Johns Hopkins Hospital Institutional review board and all subjects provided informed consent to participate in the study.

Each subject was diagnosed with one of the PPA variant types: logopenic, nonfluent, or semantic. Randomization was conducted within each variant type with an equal probability assigned to either the tDCS or sham group. As shown in Table 2.1, the two groups are balanced in both demographic and clinical characteristics. The language component of severity was evaluated based on the revised fronto-temporal dementia clinical dementia rating (FTD-CDR) used to rate severity in PPA (Knopman et al., 2008). To calculate severity, three raters independently scored each item based on the interaction with the participant and family, language, cognitive testing, and questionnaires, followed by a discussion to produce a consensus score. In the tDCS group, the Soterix Transcranial Direct Current Stimulation 1×1 Clinical Trials device (Model 1500) was used to deliver tDCS (for tDCS setup details, see Tsapkini et al., 2018). The anode was placed over the left frontal lobe and the cathode was placed over the right cheek. The size of the nonmetallic, conductive rubber electrodes (fitted with saline-soaked sponges to limit skin-electrode reactions) is 5 cm $\times$ 5 cm, which covers the whole left IFG. In each tDCS session, the density of the delivered current was 2 mA and the delivery lasted for 20 minutes. Simultaneous with the tDCS delivery, language therapy was initiated and continued for an additional 20 to 25 minutes beyond the cessation of tDCS. The sham group had 30 seconds of current ramping up to 2 mA and then backing down to 0 mA simultaneous with the start of language therapy. These procedures have successfully blinded participants to the stimulation condition (Gandiga, Hummel, and Cohen, 2006), as well as the speech-language therapist. The protocol required 15 consecutive weekday

sessions for each participant. Efforts were made to adhere to the schedule, though some participants had to leave a few days earlier because of other commitments (median number of sessions: sham = 11, tDCS = 13). In the language therapy, we combined the spell-study-spell procedure with an oral and written naming paradigm and developed individualized trained and untrained word sets (Ficek et al., 2018), where trained and untrained sets (10 to 30 words depending on individual severity) were matched in length and frequency. Each participant was shown a picture on a computer, asked to orally name it and to write the name. If the participant could not name the picture (orally or in writing), they were asked to provide 3 characteristics of the item to evaluate and reinforce semantic knowledge. If they still could not describe the word orally, they were offered the correct word and asked to repeat for 3 times. Likewise, if the participant could not write the word, or wrote it incorrectly, the therapist would offer the correct spelling in a spell-study-spell procedure. That is, the therapist wrote the correct word, reviewed each letter's sound, and then asked the participant to copy the word three times. Letter accuracy was determined based on a scoring system (Goodman and Caramazza, 1985) that considered letter deletions, additions, substitutions, and movements. Rather than whole-word accuracy, letter accuracy was considered as a more precise evaluation as it captures the effects of different types of errors. Each letter was evaluated with 1 point, 0.5 points for correct identification and 0.5 points for correct position. Scores for trained and untrained words were transformed to percentage points for each participant.

A total of 50 right-handed, native English speaking patients had a pre-intervention scan (scan1) and 48 had a post-intervention scan (scan2). One patient was deleted from the analysis because of missing values in the connectivity matrix. Among the

remaining 47 post-intervention scanned patients, 25 had transcranial direct-current stimulation + language therapy and the remaining 22 patients had the sham treatment plus language therapy. Several baseline covariates were recorded including gender, disease onset (years), age at the start of therapy and language severity. These patients were diagnosed with three variant types, including: logopenic, nonfluent, and semantic. Diagnoses were based on which function(s) were compromised. Patients with the *Logopenic* variant PPA (lvPPA) present with word-finding difficulties and disproportionately impaired sentence repetition. Patients with *nonfluent* variant PPA (nfvPPA) present with difficulty producing grammatical sentences and/or exhibit motor speech impairment (apraxia of speech). Finally, patients with *semantic* variant PPA (svPPA) present with fluent speech, but impaired word comprehension. See Table 2.1 for a summary of demographic and clinical information on the participants.

	Combined (n = 47)	tDCS (n = 25)	Sham (n = 22)
Sex	22F, 25M	11F, 14M	11F, 11M
PPA variant	15L, 23N, 9S	9L, 12N, 4S	6L, 11N, 5S
Age	67.3 (6.8)	65.8 (8.1)	69.1 (5.0)
Year post onset	4.2 (2.8)	4.3 (3.2)	4.0 (2.3)
Language severity	1.7 (0.8)	1.7 (0.9)	1.8 (0.8)
Total severity	6.3 (4.5)	5.7 (3.9)	7.0 (5.2)

Table 2.1: Patient demographics. For age, years post onset, severity, values shown are mean (standard deviation). P-values are from the Welch two sample t-tests for continuous outcomes and Fisher's exact test for categorical outcomes. Language severity is based on the language subset from the FTD-CDR scale. Total severity refers to the sum of boxes, including language and behavior as added in Knopman et al., 2008.

2.2.2 Data Preprocessing

MRI scans were obtained at the Kennedy Krieger Institute at Johns Hopkins University, using a 3 T Philips Achieva MRI scanner equipped with a 32-channel head coil. Resting-state fMRI (rsfMRI) data were acquired for approximately 9 min (210 time-point acquisitions) post-intervention. We used a 2D EPI sequence with SENSE partial-parallel imaging acceleration to obtain an in-plane resolution of 3.3×3.3 mm^2 (64×64 voxels; TR/TE = 2500/30 ms; flip angle = 75°; SENSE acceleration factor = 2; SPIR for fat suppression, 3 mm slice thickness). The data were co-registered with structural scans into the same anatomical space. Structural scans, acquired axially with a scan time of 6 min (150 slices), used a T1-weighted MPRAGE sequence with 3D inversion recovery, magnetization-prepared rapid gradient, isotropic with a resolution of $1 \times 1 \times 1$ mm^3 (FOV = 224×224 mm^2; TR/TE = 8.1/3.7 ms; flip angle = 8°; SENSE acceleration factor = 2).

Using MRICloud, a cloud-platform for automated image parcellation approach (atlas-based analysis), the MPRAGE scan was parcellated into 283 structures (Mori et al., 2016). In detail, each participant's high resolution MPRAGE was segmented by using a multi-atlas fusion label algorithm (MALF) and large deformation diffeomorphic metric mapping, LDDMM (Ceritoglu et al., 2013; Miller et al., 2005; Tang et al., 2013). This highly accurate diffeomorphic algorithm, associated with multiple atlases, minimizes the mapping inaccuracies due to atrophy or local shape deformations. All analyses were performed in native space. To control for relative regional atrophy, volumes for each ROI were normalized by the total intracerebral volume (total brain tissue without myelencephalon and cerebrospinal fluid). The resting-state fMRI was also processed in MRICloud and analyzed in a

seed-by-seed manner. Image processing is described in Faria et al., 2012 including routines imported from the SPM connectivity toolbox for coregistration, motion, and slice timing correction, physiological nuisance correction using CompCor (Behzadi et al., 2007), and motion and intensity TR outlier rejection using ART (https://www.nitrc.org/projects/artifact_detect/). The MRICloud pipeline followed established steps for rsfMRI processing as follows. After exclusion of outlier TRs per the ART routine (parameters: 2 standard deviations for motion and 4 standard deviations for intensity, more severe than the default of 9), the movement matrix combined with the physiological nuisance matrix was used in the deconvolution regression for the remaining TRs. Outlier rejection and regression of motion parameters minimizes potential motion effects. The parcels resulting from the high resolution T1 segmentation were brought to the resting state dynamics by co-registration. Time-courses of 78 cortical and deep gray matter ROIs were extracted and the correlations among them were calculated.

2.2.3 Density regression

We propose to quantify the effect of possibly non-localized stimulation on functional connectivity through a density regression. Let $\mathbf{C}_i(u, v)$ be a connectivity measure, such as the correlation of the BOLD time series, between foci u and v for $u = 1 \dots p$ and $v = u \dots p$ and then let $\mathbf{C}_i$ be the collection of connectivity measurements, typically represented by a symmetric matrix, but in our case simply an ordered vector. We study the distributional summary of the collections of $\mathbf{C}_i$ exactly as if they were drawn independently from a distribution. Let $\hat{f}_i$ be the estimate of the associated density f_i of connections for subject i. Our proposal is to analyze f_i with functional regression methods. A motivation for studying

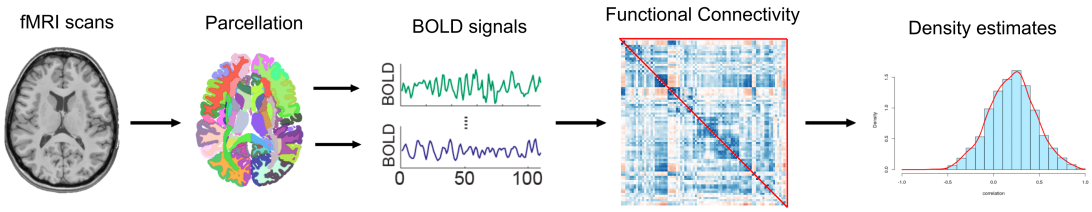


Figure 2.1: From MRI scan to connectivity density

f_i can be obtained by the weaker assumption of exchangeability of the labels. Such exchangeability translates in this context to the relevant information for predicting the outcome being in the proportion of stronger and weaker connections, regardless of where they occur.

The process of proceeding from fMRI scans to the connectivity density is outlined in Figure 2.1. We estimated the connectivity matrix via temporal correlations of BOLD signals between regions of interest (ROIs) after parcellation, which were then passed to a density estimation algorithm. Specifically, we used the vectorized elements in the upper triangular portion of the connectivity matrix to estimate the density using smoothing splines (Gu and Qiu, 1993). This performs maximum likelihood estimation on the spline coefficients for estimating the logarithm of the density function under a smoothness penalty. We chose this approach as it directly returns the splines, which are both mathematically and practically convenient, especially for performing a functional regression. In addition, it sets a boundary on the support for the estimated density, which is beneficial here, as correlation coefficients are bounded between -1 and 1. Kernel density estimators (Silverman, 1986) were also implemented as a comparison.

Our proposal is to use $\hat{f}_i$ to characterize $\mathbf{C}_i$ and subsequently study the relationship between $\hat{f}_i$ and variables of interest. In the tDCS study, the variable of interest is treatment status. Since the $\{\hat{f}_i\}$ are (infinite dimensional) functional data, we

employ functional data analysis tools (Ramsay, 2004; Ramsay and Silverman, 2007; McLean et al., 2014). Logically, one would model that treatment status predicts connectivity. However, treating complex data as covariates is typically more convenient than treating them as the outcomes. For example, the ability to incorporate other covariates is simply adding terms in a regression model. Unlike models for complex multivariate structured outcomes, an outcome reversed functional approach can be easily implemented with existing software tools available in any statistical package. As such, the method extends easily to longitudinal models, whereas longitudinal models for complex structured outcomes are not fully developed. Putting connectivity densities as covariates also makes the method directly extendable to predicting subject-specific behavior scores. Therefore, we adopt the ideas in case-control inverse regression (Prentice and Pyke, 1979; Rothman, Greenland, and Lash, 2008), and predict whether a subject is in the treatment arm using the connectivity density and the baseline covariates as predictors. Let A_i denote the treatment assignment with $A_i = 1$ for tDCS and $A_i = 0$ for sham, and $\mathbf{X}_i \in \mathbb{R}^q$ denote the q-dimensional covariate vector with the first element one for the intercept. The linear model considered is the following:

$$\text{logit}\{P(A_i = 1|\mathbf{X}_i, f_i)\} = \mathbf{X}_i^\top \beta + \int T(\hat{f}_i)g, \tag{2.1}$$

where T is a given operator from $\mathcal{L}^2$ to $\mathcal{L}^2$ aiming to capture a specific characteristic of the density functions. T can also be used to control the impact of possible outliers of connectivity measures, such as using quantile based transformations. The function g is a coefficient function representing the effect of the tDCS used in this experiment, which can potentially change for different simulation settings. The parameter $\beta \in \mathbb{R}^q$ is the coefficient vector of the covariates, both to be estimated.

Various choices of T and the shape of g have different interpretations on the resulting model. For example, setting $T(f) = f$, the identity function, the linear predictor is $\int T(f_i)g = E[g(Z_i)]$, where $E[\cdot]$ is the expectation of a random variable and Z_i is a random variable drawn from f_i. With a sufficiently flexible choice of g, Model (2.1) covers a broad range of possible model fits. However, many of them may not focus on non-central components of the density, where effects would likely occur because of the stimulation procedure. For example, if g is a polynomial, the model considers the moments of the density (mean, variance, skewness, etc.) as predictors. However, it offers no benefit over the direct usage of the moment estimates of the connectivities. Thus, polynomial bases will not be discussed further, though they do demonstrate an interesting special case of the approach.

As for the choice of T, using $T(f) = \log(f)$ is similar to the use of the identity function. It loses the expected value interpretation , while instead, performs regression on the space of densities with Aitchison geometry (Egozcue, Díaz-Barrero, and Pawlowsky-Glahn, 2006). Thus, it may better detect the influence of the tail behavior on the outcome.

Another choice is the quantile mapping, $T_q(f) = F^{-1}$, where F is the cumulative distribution function associated with the density f. With a sufficient number of foci, this approach is approximately equivalent to using the empirical quantiles of the connectivity data as the regressors. Our proposed approach is quite similar to this. However, we further propose to weight the quantiles via density quantile. Specifically, we set $T_{ldq}(f) = \log \circ f \circ F^{-1} = -\log\left[(dF^{-1}/dt)^{-1}\right]$ where $\circ$ is the function composition operator. The latter equality is easy to derive by taking derivatives via the chain rule to the identity function, $F \circ F^{-1}$. Note that

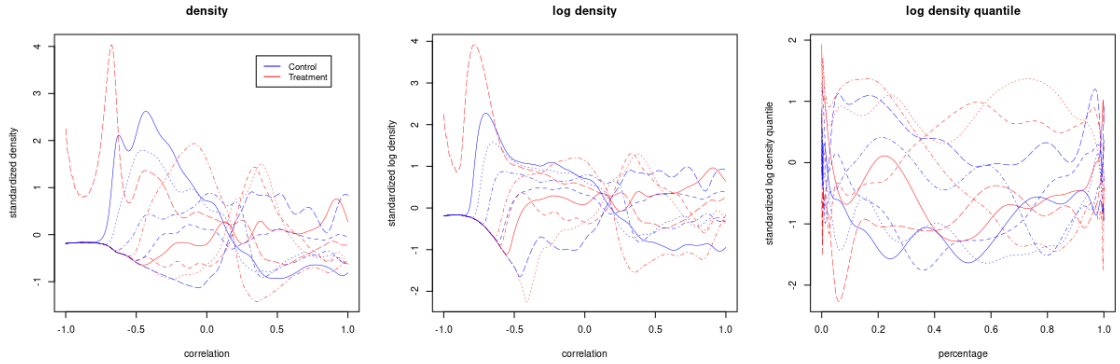


Figure 2.2: An illustration of connectivity densities, its log transformation and its log density quantiles. Plots shown for 10 random sampled subjects in our tDCS study and functions are standardized across all subjects to have similar y scales along x-axis.

the density quantile $f \circ F^{-1}$ can be regarded as a quantile synchronized version of the density function, and therefore is more sensitive to the changing tails. The logarithm transform maps density quantile to a Hilbert space, which is practically useful for linear models. This idea has been explored before as a potentially preferable method for utilizing quantiles as regressors. Specifically, it is equivalent to the Hilbert space mapping, suggested by Petersen, Müller, et al., 2016. Figure 2.2 shows original densities, log transformed densities and log density quantiles of 10 random sampled subjects in our tDCS study.

2.2.4 Reversing the predictor/response relationship

It is typical in regression models to consider the hypothetically functionally antecedent variable as a predictor, independent or exogenous variable, rather than an outcome, dependent or endogenous variable. A counterexample is in outcome dependent sampling, such as in retrospective studies. We utilize the same strategy of reversing the typical predictor / response relationship, as is more convenient

for modeling with high dimensional and complex quantities (such as brain connectivity) as the predictor. In the tDCS study, we model treatment assignment as the outcome using a logit model with the connectivity density and other covariates as the independent variables. This avoids the need to construct probability distributions on the connectivity densities themselves.

To elaborate, using Bayes' rule and $P(A_i = 1) = P(A_i = 0) = 0.5$ (due to the randomization), for any function g and transformation T, we have:

$$\text{Odds}(A_i = 1|\mathbf{X}_i, \langle T(f_i), g\rangle) = \frac{P(\langle T(f_i), g\rangle|A_i = 1, \mathbf{X}_i)}{P(\langle T(f_i), g\rangle|A_i = 0, \mathbf{X}_i)},$$

where $\langle \cdot, \cdot \rangle$ is any inner product of two functions. In our application we consider logit models on $P(A_i = 1|\mathbf{X}_i, T(f_i))$, which depend on f_i only though the form $\langle T(f_i), g\rangle$. Also, thanks to the randomized design, we can be aggressive in excluding potential confounders as covariates. This is especially helpful given the modest sample size. As the above relationship shows, our treatment assignment outcome model, $P(A_i|\mathbf{X}_i, T(f_i))$, is consistent with any connectivity outcome model, $P(\langle T(f_i), g\rangle|A_i, \mathbf{X}_i)$, where the likelihood ratio comparing treated to controls is approximately log linear with our linear separable density model given in Equation 2.1.

2.2.5 Estimation of the coefficient function

To estimate the coefficient function, g in Model (2.1) , we performed a functional principal components analysis (fPCA, see Reiss and Ogden, 2007, for a review). This reduces the dimension of the functional regressor using a set of data-derived bases. In this approach, one calculates the PCA decomposition of the functions, $\{T(\hat{f}_i)\}$, using the Karhunen/Loève transformation (Ghanem and Spanos, 2003),

where the covariance function is smoothed (Di et al., 2009). We selected the leading principal components which explained over 99% of the variation as the basis functions. Notice that the version of fPCA utilized here does not honor possible density implied constraints of $T(\hat{f}_i)$. Generalized cross validation (GCV) was used to choose the smoothing parameters (for detailed discussion, see Section 4.5.4 of Wood, 2004). Confidence bands were derived using a Bayes approach. (Wahba, 1983; Nychka, 1988; McLean et al., 2014).

2.2.6 Comparison

To illustrate the benefit of conducting a delocalized analysis, a simulation study based on the fMRI data collected in the tDCS study was conducted. We demonstrate an extreme example where non-localized brain stimulation decreases statistical power, or even makes it impossible to identify ROI pairs with a significant effect when implementing a localization method. However, using connectivity densities retains the relevant information. Our goal in this simulation was to create a caricature of non-localized effects, to demonstrate the statistical direction that the procedure highlights.

As a correlation coefficient, connectivity can be written as $\cos(\theta)$ where θ is the angle between two signals. In the simulation, consider a brain connectivity map with 20 regions, $R_1 \ldots R_{20}$. For every map, let θ_{ij} be the angle between signals in location i and j; we simplified the data generating distribution by assuming that the angles, θ_{ij}, are i.i.d. following a von-Mises distribution, $M(\mu, k)$, where the density is $f(\theta|\mu, k) = e^{k\cos(x-\mu)}/2\pi I_0(k)$, with I_0 as the modified Bessel function of order 0. The parameters, μ, k were estimated from pre-intervention patients by maximum likelihood. This was done to have a realistic null distribution on

densities.

A non-localized "stimulation" was simulated by perturbing region R_i with equal probability across i. After stimulation, we simplified the effect via a degree rotation, ϕ, for the signal at R_i. Correspondingly, all θ_{ij} change the same amount and the final post-stimulation connectivity was a convex combination of this stimulated matrix and the pre-stimulation matrix, where the weight was used to control the signal level and therefore controls the degree of difficulty in detecting the effect. Denote C_{ij} the pre-stimulation correlation between region i and j, that $C_{ij} = \cos(\theta_{ij})$. A stimulation on region i_0 yield a symmetric post-stimulation connectivity C_{ij}^{sti} as

$$C_{ij}^{sti} = \begin{cases} C_{ij} & i \neq i_0, j > i \\ wC_{ij} + (1-w)\cos(\theta_{ij} + \phi) & i = i_0, j > i \end{cases}$$

Notice that, although uniform stimulation on all regions of R_i is unpractical in many situations, this simulation is a boundary case to understand the effect of lacking localization. Mover, it is still consistent with other kinds of non-localized effects that are random mixture of localized effects. In the Appendix, we also describe and examine another intuitive simulation setting, and we observed similar results.

For every run of the simulation, we sampled 50 pre-stimulation maps from the pre-intervention scans and fit the parameters μ, k for each. We subsequently simulated 50 connectivity maps from samples of fitted von-Mises distributions, and applied the stimulation above for a random half of these maps. We chose $\phi = \pi$, and the weight w in the convex combination was chosen to be 75%. Other values, ranging from 90% to 50%, were also tried and similar patterns

were observed. Weights under 50% made the signal detection too easy and methods are indistinguishable. Significance results for edgewise testing, principal component regression and density regression were compared, with different density regression transformations for 1,000 simulations. For completeness, we also considered instances with no stimulation effect and when the stimulation was localized at a specific region.

The edgewise regression approach considers the following model:

$$\text{logit}\{P(A_i = 1|\mathbf{X}_i, f_i)\} = \mathbf{X}_i^\top \beta + \mathbf{C}_i(s,t)\alpha_{st}, \tag{2.2}$$

where $s > t$. The second approach was a regression model with dimension reduced predictors:

$$\text{logit}\{P(A_i = 1|\mathbf{X}_i, f_i)\} = \mathbf{X}_i^\top \beta + \mathbf{S}_i\alpha, \tag{2.3}$$

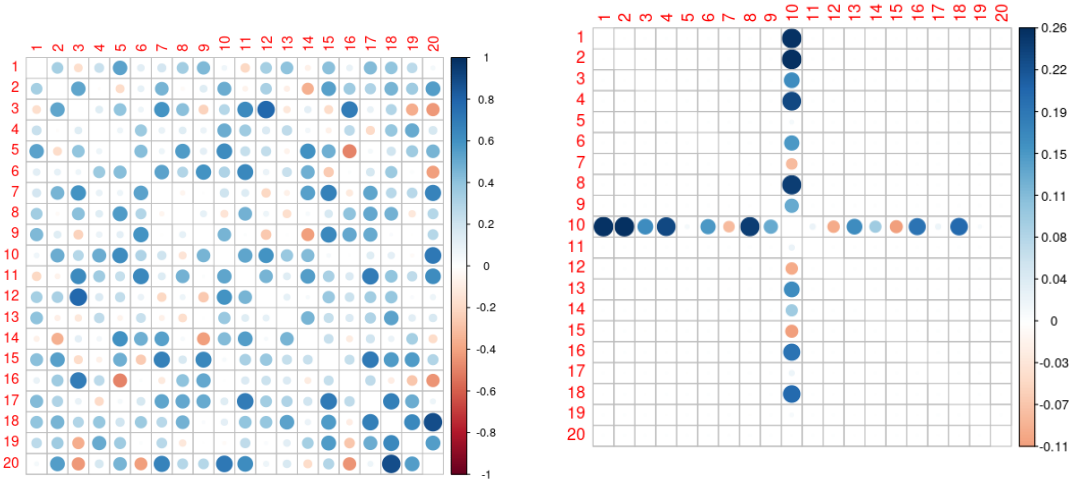
where $\mathbf{S}_i$ are the leading principal components of the vectorized connectivity matrix, $\mathbf{C}_i$. We refer to this model as the PC model.

2.3 Results

2.3.1 Simulation

Figure 2.3a, 2.4a shows example connectivity matrices and the difference after stimulation from an example simulation. The virtual stimulation was applied at region 10 in the right panel plot, while the left panel is the pre-stimulation map. We report the rate of positive findings for all methods. Results are shown in Figure 2.3c. Localization methods, including the dimension reduction method, do not find any significant region pairs in the non-localized simulations. In contrast, in this

setting, the density method detected the stimulation impact on the connectivity densities. Among all the transformations, the log density-quantile transformation was significantly better than others. We would like to reiterate that the simulation is contrived to highlight an extreme setting. Connectivity density methods will not necessarily increase the sensitivity of the analysis. If the true effect is localized, it cannot be better than well specified localized method.



(a) **(b)**

	Bonferroni	FDR	PC	T_0	T_l	T_{ldq}
Non-Localized	0.073	0.078	0.118	0.638	0.117	0.717
Localized	0.638	0.669	0.754	0.629	0.112	0.714
No-Stimulation	0.061	0.065	0.113	0.075	0.058	0.059

(c)

Figure 2.3: Figure (a) shows the simulated pre-stimulation connectivity matrix of a subject and Figure (b) is the simulated post-pre difference in the connectivity matrix of the same subject. Table (c) shows the ratio of significant positive findings over 1000 runs. T_0, T_l, T_{ldq} are density regressions with the identity, logarithm and log density-quantile transformation described in section 2.2.3. Bonferroni, FDR (Benjamini and Hochberg, 1995) refer to edgewise regression with different multiplicity correction procedures. PC refers to the principal component regression with the top 10 components, the number chosen by minimizing the sum of type I error (significance ratio in Non-Localized situation) and type II error (none significance ratio in Localized situation).

2.3.2 Analysis of the tDCS data using localized methods

For the tDCS data, we tested the significance of the edgewise regression [Model (2.2)], a principal components regression [Model (2.3)] and a LASSO post-inference model (Dezeure et al., 2015) using connectivity of all ROI pairs. No foci-pair or principal components was identified as significant in either regression model, at Type I error rate levels of 0.05 or even 0.1. Of note, previous localization work on related data (Ficek et al., 2018), yielded significant findings. However, the total number of regions were restricted, thus dramatically reducing multiplicity concerns. In this analysis, 78 regions were used, resulting in a more stringent correction factor based on 78 choose 2, or 3,003 comparisons. In addition, a more restrictive inclusion criteria in (Ficek et al., 2018) led to a different study population.

2.3.3 Analysis of the tDCS data using the density regression

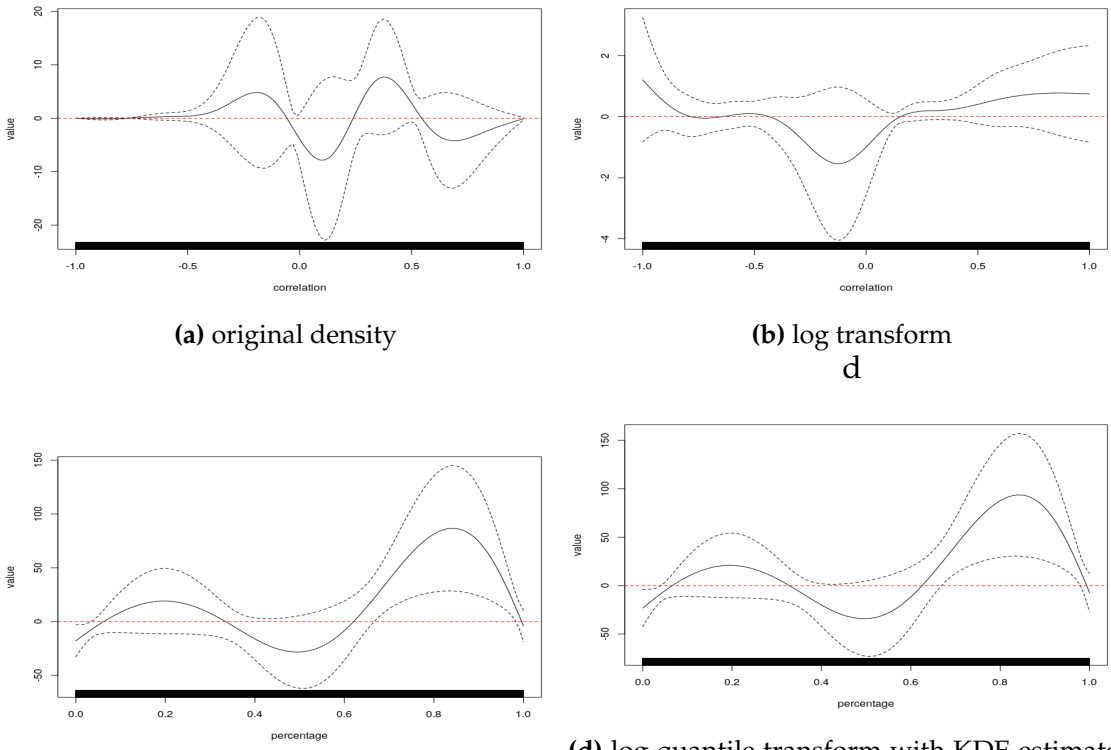
In this section, we present the analysis results of the tDCS study using the density regression Model (2.1) with different transformations (T). The fitted coefficient function, g, and its 95% confidence interval are presented in Figure 2.4. Functional linear regression was performed using the `refund` R-package with default parameter of smoothed covariance fPCA, which chooses the number of components that explains over 99% of the data variation.

Regressing on the density after applying the log-density quantile transform yielded the highest number of significant signals, which reaches its maximum around the 85^{th} percentile. This potentially indicates that stimulation has a consistent tail effect, which is more likely to be aligned by quantile, rather than absolute

value. Since the estimated coefficient function is significantly non-zero only in the positive tail this suggests that the tDCS group had higher connection densities in the tail than the sham group. That is, connectivity among the most connected regions was higher in the tDCS group.

A likelihood ratio test was performed to compare logistic regression with only baseline variables and our model including both the baseline variables and the log density quantile term. The resulting p-value was 0.0052, indicating a statistically significant gain of information from connectivity density at the 0.05 benchmark type I error rate. The conclusion remains true if one applies a Bonferroni p-value correction. Specifically, three transformations were compared and therefore the corrected p-value is 0.017. Notice that this is already a conservative value. The result agrees with a non-parametric permutation test where we do the same regression but connectivity densities of subjects are randomly shuffled. Using AUC as test statistic, we observe that the AUC of log density quantile model is also significant larger than that of null distribution, which is the AUCs with shuffled connectivity densities. The pvalue is 0.015 ± 0.0009 estimated from 20,000 runs. A further reanalysis of subgroups shows that the effect is driven primarily by the *nonfluent* subtype which comprises 23 over total 47 subjects. There is not enough data to investigate the possibility of different effects of other subtypes, the least of which only has 9 subjects. We also performed a sensitivity analysis examining the impact of hyperparameters in the density estimation. We changed the smoothing parameter in spline smoothing and bandwidth in kernel density estimation method, both in the range of $[\theta_0/2, 2\theta_0]$, where θ_0 is the corresponding default value. For smoothing splines this value was selected by the approximated cross validation method suggested in Gu and Wang, 2003 and for KDE this value is

suggest by Silverman, 1986. We observed that the log density quantile transformed model constantly gives significant information gain with p-value < 0.05 in all settings, comparing with the demographic only baseline model. Therefore, the method is not sensitive to reasonable deviations in hyperparameter selection.



(a) original density

(b) log transform

d

(c) log quantile transform

(d) log quantile transform with KDE estimated density

Figure 2.4: Model results on the tDCS experiment. The black solid line is the fitted coefficient function, g, with the black dashed line referencing the associated 95% confidence interval. Densities were estimate from smoothing splines implemented in the **fda** R-package with 19 degrees of freedom for the spline basis. A kernel density estimator (KDE,Figure 2.4d) is also computed and compared with smoothing spline (Panel 2.4c) method. Contrasting 2.4c and 2.4d shows that the density estimation technique did not impact results.

We also studied the effect of the estimated function on behavior change. We found that the variable $\int T(\hat{f}_i)\hat{\beta}$ is significant ($p < 0.05$) for predicting the change

of language ability, measured by untrained items, after transcranial direct-current stimulation. Here $\hat{\beta}$ is the coefficient function estimated above for $T = T_{ldq}$ and, recall, $\hat{f}_i$ are the connectivity densities for post-intervention scans. The result shows a necessary condition for connectivity density mediating the effect of stimulation on language ability, which can motivate a future formal mediation analysis.

2.3.4 Induced Connectivity

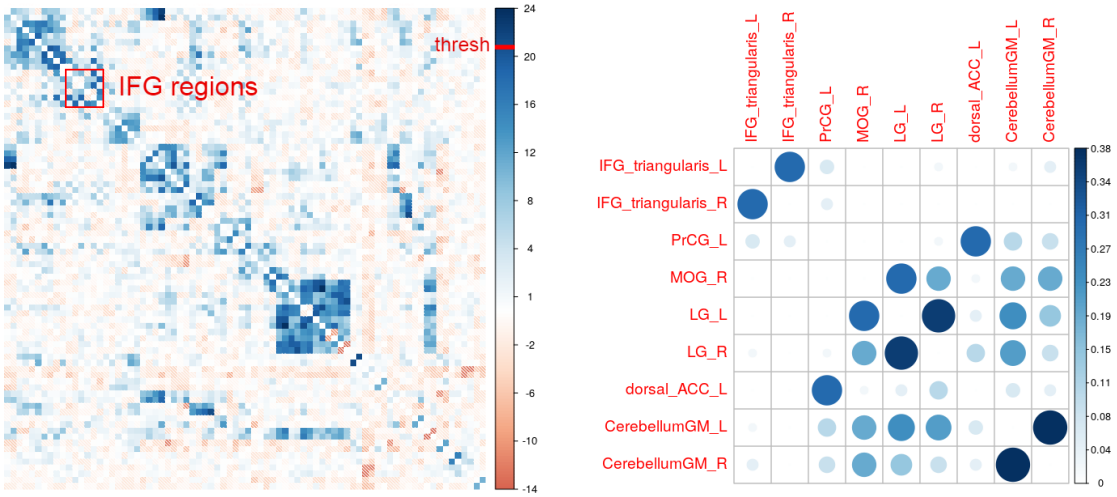
Consider the best model using the log density quantile transform, T_{ldq}. We have

$$\text{logit}\{P(A_i = 1|\mathbf{X}_i, f_i)\} = \mathbf{X}_i^\top \beta + \int_0^1 \log[f_i \circ F_i^{-1}(q)]g(q)dq.$$

Notice that for the connectivity matrix, $\mathbf{C}_i$, we have $F_i\{\mathbf{C}_i\} \sim U(0,1)$, a uniform distribution on $[0,1]$ via the probability integral transform. Let $\mathbf{Q}_i(s,t) = F_i\{\mathbf{C}_i(s,t)\}$. Then, it follows that:

$$\begin{aligned}\int_0^1 \log[f_i\{F_i^{-1}(q)\}]g(q)dq &= \mathbb{E}[g(\mathbf{Q}_i)\log f_i\{F_i^{-1}(\mathbf{Q}_i)\}] \\ &\asymp \frac{2}{N(N-1)}\sum_{t>s} g\{\mathbf{Q}_i(s,t)\}\log f_i[F_i^{-1}\{\mathbf{Q}_i(s,t)\}].\end{aligned}$$

Therefore, for this subject, one can assign $g\{\mathbf{Q}_i(s,t)\}\log f_i[F_i^{-1}\{\mathbf{Q}_i(s,t)\}]$ as the effect size for region pair (s,t). Averaging this effect across all patients yields an importance metric for every region pair in the model. We call this stimulation induced connectivity, since it describes how influential the correlation of each region pair is in predicting stimulation status. The induced connectivity matrix is shown in Figure 2.5, together with a summary of effect agreement across subjects, where for each patient, region pairs are selected with top 5% absolute effect size



(a) **(b)**

Figure 2.5: Figure 2.5a shows the induced connectivity described in section 2.3.4. IFG regions (the tDCS target) are noted in the red box. Figure 2.5b shows some region pairs with the most consistent contribution, measured by the frequency of having top 5% absolute effect size across all patients.

and the frequency of each region pair being selected is calculated.

This technique, of course, returns to a discussion of localized effects. However, by investigating this measure one can ascertain the degree of localization consistency across subjects - an impossibility with pure localization analysis.

2.4 Discussion

In this manuscript, a new framework for analysing functional connectivity was explored. Functional data analysis of log quantile connectivity densities investigates possible non-localized effects associated with subject level variables. It is clear that our method can be directly applied to other kinds of numerical measurements. For example, partial correlations or entropy based measures. However, it continues to be only useful suitable if connection exchangeability represents a useful model. A

sizable by-product of this style of analysis is the drastic reduction of multiplicity considerations. This is of great importance in connectivity analysis, where the number of comparisons grows at a rate of the square of the number of foci being considered. In the data application, we find associations between stimulation and connectivity density. In contrast, edgewise methods fail to find any results, because of multiplicity issues. This is partially due to a wide search of all possible region pairs from the parcellation. Of course, one could also reduce multiplicity concerns by restricting attention to regions associated with a priori hypotheses of interest, as was done in Ficek et al., 2018. In contrast, investigating connection densities is an omnibus approach that benefits from a reduction in the number of tests over exploratory edge-wise approaches, a robustness to non-localized effects and a robustness to the inclusion of unnecessary foci. These benefits come at the expense of the loss of power and interpretability over analyses considering only a small set of tightly specified edge-wise hypotheses. Our method can also be extended to seed-based connectivity and voxel-by-voxel connectivity without any modification. However, the assumption of complete node invariance discards a potential sizable amount of relevant localization information. Therefore, we believe that the method would be primarily useful as an easy and simple early stage omnibus test, or after light localization efforts, such as considering connectivity densities between voxels within sets of regions of interest. To further emphasize the ease and simplicity of the method, we stress that that density regression can be coded from scratch in only a few lines of code in any modern scripting environment with PCA and GLM functions.

Density regression, as a prediction model, can be view as a generalization of connectome-based predictive modeling (Shen et al., 2017). Connectome-based

predictive modeling (CPM) uses individual connectivity matrices to predict behavioral measures. The method first selects location-pairs that are most significantly correlated with the outcome, then summarizes the matrix by adding up connectivity measures in selected pairs, and this sum is used as a predictor in a regression model. In CPM, there is no localized effect and CPM can be viewed as a regression on connectivity density using only a constant basis. Here we generalize it by utilizing more distributional information.

An interesting direction to pursue with connectivity density methods is to consider potential robustness to spatial registration (Oliveira and Tavares, 2014). The connectivity density can relatively easily be shown to be invariant to relabeling and affine transformations (see Theorem 1 in the appendix). In contrast, localization methods heavily rely on both accurate registration and biological functional localization across subjects. Therefore, it is interesting to posit that density regression could be used after only mild affine registration efforts prior to the more time-consuming non-linear registration.

However, to reiterate, ignoring potentially useful localization information can reduce power and sensitivity. Surely, the optimal strategy removes subject-specific artifacts and reduces the search space with - correct - strong a priori hypotheses and then tests only those edges. However, in the absence of this ideal case, one is often confronted with a massive unstructured search problem with localization analyses. In contrast, density regression is more akin to an omnibus F-test, looking over a large range of edges, dramatically mitigating multiple comparisons issues in the favor of testing one overview hypothesis, rather than a large collection of highly specific ones. Therefore, we suggest the method as an early stage tool in a neuroimaging data analyst's toolbox.

We used functional data analysis to relate connection densities to outcomes. Functional data analysis tools (Ramsay and Silverman, 2007) have grown to be quite flexible. Thus, density regression approaches can be relatively easily generalized to handle different settings, such as any typical statistical outcome model and longitudinal data. Also, density estimates may naturally make adjustments for missing data, in the form of missing foci, since the density can remain the same in some contexts. This has potential broad implications for the study of stroke and other diseases with abnormal brain pathology. Localization methods are not available if the region of interest is damaged or missing. In contrast, density based methods are easy to apply. In addition, we used PCA on the log quantile densities as the basis for functional regression. The result is that the method can be applied using standard software without modification. Other bases and penalization strategies may improve the approach. In fact, the utility and application of functional regression in neuroimaging has been greatly improved via recent research efforts (see Goldsmith et al., 2011; Goldsmith et al., 2012; Goldsmith, Wand, and Crainiceanu, 2011; Reiss et al., 2017, for examples).

Utilizing functional regression also has the benefit of producing more interpretable models as compared with machine learning approaches. However, this is achieved at a likely cost of prediction performance. It is possible that ML approaches could navigate the trade offs between localization and exchangeability non-parametrically and possibly achieve better prediction performance. Thus, we view density regression as a parsimonious modeling choice rather than a method to optimize prediction performance.

Statistically, we assumed independence between subjects and relied on the randomization to invert the predictor / response relationship using logit models.

This borrows techniques from case referent sampling from epidemiology dating back to the seminal work of Cornfield (see Breslow, 1996; Greenhouse, 1982, for overviews). A benefit of doing this is that it is generally easier to have the more complex variable as the predictor rather than a response. To elaborate, to have a density as an outcome, predictions from the model must be functions that are both positive and integrate to one. Most existing functional approaches, especially point-wise ones, would satisfy neither criteria and modeling distributional outcomes is an active area of statistical research. The probability space containing the outcome is necessarily a probability distribution on distributions, such as a Dirichlet process. While this is not a problem per se, it makes inference more technically challenging. In contrast, by conditioning on the density, as we have done, its distribution does not need to be modeled and the fitting and inference requires little more than well known generalized linear model techniques. In Appendix 2.6.3 we further the discussion in comparison with function on scalar regression. It is seen that, with almost no effort, one obtains the use of easier models (GLMs) and appropriate inferences by reversing the relationship and the resulting estimates are similar to those of function on scalar regression. However, because the constraints are not accounted for in the function on scalar model, inferences remain in question.

Nonetheless, we reiterate that the use of connectivity density as a regressor remains useful, even if one prefers not to flip the predictor / response relationship. For example, in our tDCS example, connecting the connectivity density to behavioral outcomes would be relevant, where the natural predictor would be functional connectivity.

Independence between subjects was used for inference. We also used density

estimates for connection densities, techniques that implicitly require sampling assumptions for theoretical convergence. However, we contend that connectivity densities are intrinsically of interest, and therefore no appeals to super-population inference and sampling assumptions are needed for estimation. This is analogous to spatial group ICA, where productive estimates are obtained via independence assumptions on voxels over space, without a true sampling or super-population model for inference (Calhoun et al., 2001). An interesting future direction of research would investigate dependencies between foci correlations.

Our recommended approach uses log quantile densities as the functional predictor, rather than the density, distribution function or quantile function directly (Petersen, Müller, et al., 2016). This approach has convenient theoretical properties, but also the practical benefit of focusing attention on tail behavior, where effects are most likely to be seen. Utilizing the quantile density also creates robustness to irrelevant foci pairs being included in the analysis.

Our simulations and data results focus on settings that highlight the benefits of an omnibus density regression approach. In the simulations, we investigated a non-localized caricature of typical effects. Similarly, in our data analysis, we performed no filtering of regions prior to analysis (thus magnifying multiple comparison concerns). It was shown in the simulation, that functional density regression approaches can find real non-localized effects, whereas, as expected, edgewise methods do not find any. It should be emphasized that the performance of the density regression approach is invariant to the distribution of effects across subjects, whereas edgewise approaches become viable as the degree of localization increases.

In addition, the flexibility of the approach finds effects in the real data, even

though there are a great deal of irrelevant connections (i.e. unnecessarily included region pairs) being studied. Edgewise and other regression approaches are highly sensitive to unnecessary null connections being included in the analysis. A benefit of the data being considered is the likely existence of an effect related to the stimulation. However, we emphasize that a single omnibus approach does not represent a full analysis of the data. We recommend this approach as a global analysis to be performed prior to edgewise or other localization methods. This mirrors the classic ANOVA (analysis of variance) approach of performing an overall F test before investigating pairs of explanatory factor levels. It would be most useful in exploratory model building where foci selection is not restrictive. In cases of tightly coupled statistical hypotheses involving relatively few regions or foci, density regression would not be needed or particularly helpful.

This methodology raises many avenues for future research. For example, one the idea of non-localized effects in dynamic connectivity (Hutchison et al., 2013) via stochastic processes of connectivity densities (by time). In addition, there are multiple alternatives for densities estimated from correlation of each region pair for contralateral regions. Here, it should be acknowledged that there is strong homotopic correlations from symmetric regions. One should then deal with multivariate densities estimated from pairs of correlations. This same logic could be applied to geographically close regions and for instances with longitudinal scans. The connectivity density of spectral information (Haan et al., 2012), like leading principal component scores, should also be studied to potentially extract relevant brain graph properties.

Finally, there's the role that connecvity density methods could play in fMRI analysis of subjects with missing brain tissue, such as studies of stroke or surgical

interventions. Connectivity density methods may be resilient to the missing data impact of differential brain structure in a way that localization methods are not. In fact, it is interesting to conjecture what localization methods even mean in these settings where a subset of subjects are missing areas of localization. In contrast, density methods may provide a more robust and well defined methodology. It is worthy of note that components of graph methodology (Sporns, 2010; Bullmore and Sporns, 2009) often considers summary metrics that do not require or assume localization. Density regression can be considered a subset of weighted graph metric analysis.

2.5 Acknowledgments

We would like to thank our participants and referring physicians for their dedication, helpful comments and interest in our study. All the data reported here were collected through grants from the Science of Learning Institute at Johns Hopkins University and NIH (National Institute of Deafness and Communication Disorders) through award R01 DC014475 to KT. A.V. was supported by National Science Foundation CRCNS award 1822575 and National Science Foundation CAREER award 1845430. M.A.L. was supported by NIH grants R01 EB016061, EB029977 and EB026549 from the National Institute of Biomedical Imaging and Bioengineering. J.P was supported by NIH/NIBIB grants P41EB015909 and P41EB031771, both to P. van Zijl. The Authors declare that there is no conflict of interest.

2.6 Appendix

2.6.1 Invariance properties

Here we discuss some invariance properties of the connectivity density. Consider C a connectivity measure where $C(x, y)$ is measuring the connectivity between location $x \in \mathcal{D}$ and $y \in \mathcal{D}$. The connectivity density can be defined as the density of random variables $C(U, V)$, where (U, V) follows a sampling distribution on $\mathcal{D} \times \mathcal{D}$. Denote f_{sample} as the density of that sampling distribution. It is easy to see that the connectivity density, f, defined in Section 2.2.3 also follows such a definition while using the uniform distribution as f_{sample}. We prove that f is invariant to re-labeling in discrete cases (e.g. connectivity between ROIs) and to affine transformation in continuous cases (e.g. interpolation of connectivity between voxels). Denote $supp(C)$ be the support of connectivity measure C. After any invertible transformation, $\mathcal{F}$, the connectivity measure $C_{\mathcal{F}}$ will be naturally defined as $C_{\mathcal{F}}(x, y) = C(\mathcal{F}^{-1}(x), \mathcal{F}^{-1}(y))$. Then we have the following Theorem 1

Theorem 1. *Let (U^C, V^C) follow the uniform distribution on $supp(C)$. Then, the density of $C(U^C, V^C)$ has the same distribution with $C_{\mathcal{F}}(U^{C_{\mathcal{F}}}, V^{C_{\mathcal{F}}})$, where $\mathcal{F}$ is any permutation map if $supp(C)$ is a finite discrete set and $\mathcal{F}$ is any affine transformation if $supp(C)$ is a closure of some open set in $\mathcal{R}^3$. Therefore, connectivity densities are invariant to these transformations.*

Proof. By simple change of variable calculus of random variables, we know that under sampling distribution f_{sample}, $C(U, V)$ has the the sample distribution with

$C_{\mathcal{F}}(U', V')$ if $(U', V') \sim f'_{sample}$ where

$$f'_{sample}(x, y) = |\mathcal{F}|^2 f_{sample}(\mathcal{F}^{-1}(x), \mathcal{F}^{-1}(y))$$

.

In our uniform cases, the Jacobian $|\mathcal{F}|$ and sampling distribution $f_{U^C,V^C}, f_{U^{C_\mathcal{F}},V^{C_\mathcal{F}}}$ will always be a constant. Therefore the condition above always holds and $C(U^C, V^C)$ must follow the sample distribution with $C_{\mathcal{F}}(U^{C_\mathcal{F}}, V^{C_\mathcal{F}})$. □

Since the uniform distribution is the only distribution invariant to all affine transformations / permutations, we know that the connectivity density defined in Section 2.2.3 is also the only possible distributional summary that has such an invariance property for arbitrary connectivity measures.

2.6.2 Additional Simulation

Here we describe another intuitive simulation setting and show that a similar pattern is observed. Specifically, it shows that our methods can detect non-localized effects, while edgewise method or dimension reduction methods, like PCA, can not, although the best transformation of densities might change for different signal distributions.

Again we consider connectivity matrix of 20 regions $R_1, \cdots, R_{20}$. A no-stimulation connectivity matrix, C, is sampled uniformly from 50 pre-intervention scans in our data and its 20 rows and columns are also uniformly sampled from an original 78 by 78 connectivity matrix. Now consider a localized stimulation as additive Gaussian signals to the Fisher-z transformed correlation for specific region pairs.

It then gives post-stimulation connectivity matrix C^{sti} differs with C only on

$$\tanh^{-1} C^{sti}_{i_k j_k} = \tanh^{-1} C_{i_k j_k} + \varepsilon_k \tag{2.4}$$

for $\{(i_k, j_k)|k = 1, 2 \cdots K\}$ some specific region pairs and ε_i i.n.d. follows $\mathcal{N}(\mu_k, \sigma_k^2)$. Notice that this formulation corresponds with the underlying effect pattern in some common edgewise analysis of change in connectivity, for example Ficek et al., 2018.

In the simulation for localized analysis, the locations (i_k, j_k) are uniformly randomly selected from all 190 region pairs and then fixed for all samples. Naturally a stimulation with non-localized effect would also follows Equation (2.4). But every time it is performed, $\{(i_k, j_k)\}$ becomes another independent sample from the 190 regions. In the experiment, we choose $K = 10, \mu_k = 0.5, \sigma_k = 0.5$ for all k. We also observed similar patterns for a variety of parameters settings. We ran the experiment for 10,000 independent simulations. For every run we sampled 100 no-stimulation connectivity maps with another 100 each for localized stimulation, non-localized stimulation and no stimulation. We studied how different methods work in these situations as described in Section 2.2.6. The results for the simulation can be found in Table 2.2. We observe a similar pattern as Table 2.3c that connectivity density based methods can detect non-localized effect while edgewise analysis and principal component analysis cannot. It also shows that the optimal transformation might be different for different patterns of the effect, as the log transformation is the best in this situation while the log-density-quantile transformation is the best in Table 2.3c.

	Bonferroni	FDR	PC	T_0	T_l	T_{ldq}
Non-Localized	0.060	0.065	0.089	0.618	1	0.870
Localized	0.938	0.953	0.994	0.620	1	0.862
No-Stimulation	0.048	0.051	0.082	0.060	0.056	0.053

Table 2.2: The table shows the ratio of significant positive findings over 10,000 runs. T_0, T_l, T_{ldq} are density regressions with the identity, logarithm and log density-quantile transformations described in Section 2.2.3. Bonferroni, FDR (Benjamini and Hochberg, 1995) refer to edgewise regression with those associated multiplicity correction procedures. PC refers to principal component regression with the top 20 components.

2.6.3 Connectivity Density as Outcome

In this section we detail why we reversing the predictor/response relationship is a compelling idea and thus compare the results with a typical function-on-scalar regression with connectivity densities as outcomes.

Excepting the convenience, as discussed in section 2.4, the main reason for reversing the predictor/response is that typical function-on-scalar regression methods can not satisfy the integral constraints on the outcome, which are densities or isomorphic transformation of densities. Therefore, the specified distribution is not correct, creating concern regarding inferences.

Consider the following typical linear functional model with outcome function y and features $\boldsymbol{x}$.

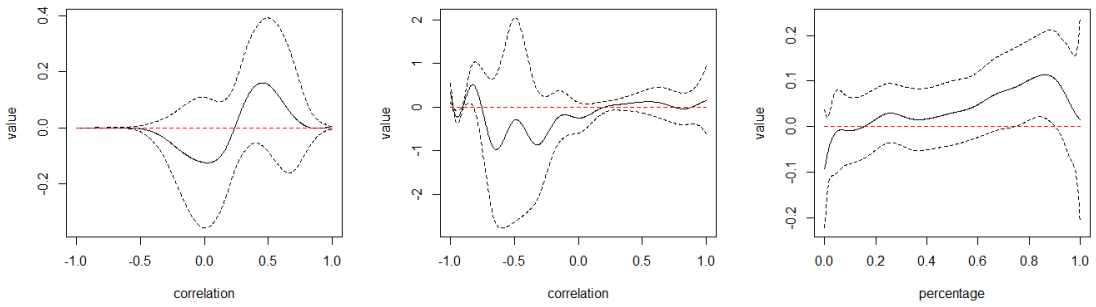
$$y(t) = f_0(t) + \boldsymbol{x} \cdot \boldsymbol{f}(t) + \varepsilon(t) \tag{2.5}$$

where y is a density functions, log density function, or the log-density-quantile transformations. Recall, density functions must be both positive and integrate to 1. Log densities require integral of their exponential to be 1 and Log density quantiles require their corresponding quantiles be supported within [-1, 1], because they are quantiles of correlations. It is easy to see that within the linear functional

framework 2.5, all these constraints cannot be translated into individual constraints on estimation of $f_0, \boldsymbol{f}$. Therefore, the model is specifying an easily demonstrably false distribution, resulting in possibly incorrect inferences even if estimation remains viable. Other methods exist to correct this problem, for example Szabó et al., 2016; Chen, Lin, and Müller, 2021, but this is an active area of research and is thus challenging to implement for most practitioners.

In Figure 2.6, we show the estimation results of model 2.5 on our data as a reference. These are the slope functions of the treatment assignment variable, the estimated differences before and after tDCS stimulation. We used the regression methods described in Reiss, Huang, and Mennes, 2010 to solve the problem 2.5 and the penalty parameters selected by generalized cross validation. There is, as expected a high degree of similarity between the corresponding curves and those in Figures 2.4a-2.4d. But, as we explained above, the distributional assumptions are questionable in this context and the confidence bands remain in question, and therefore we do not report such results in the main paper. We also note the distinction in convenience, whereby we obtain similar estimates using only a GLM, perhaps the most standard statistical model.

Figure 2.7 shows one sample outcome function from the fitted model. We have checked that it breaks the positive constraints on both tails and its integral is $0.99 < 1$. Also it is clear that the confidence band from the model does not make sense because all densities should be non-negative. Therefore the inference results from the model 2.5 are wrong.



(a) original density **(b)** log transform **(c)** log density quantiles

Figure 2.6: Estimated difference function of the transformed neural densities between treatment and control groups, holding all other variables the same. Similar patterns could be found compared to Figures 2.4a-2.4d but their confidence bands are biased because no constaints on the outcome function are satisfied.

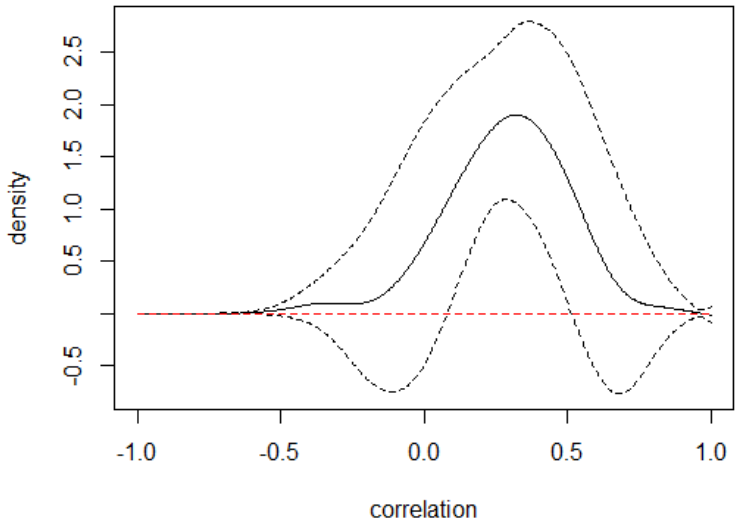


Figure 2.7: A sample outcome function from the fitted model. It breaks the positive constraints on both tails and its integral is $0.99 < 1$. Also the confidence band from the model does not make sense because all densities are non-negative.

References

Friston, Karl J (2011). "Functional and effective connectivity: a review". In: *Brain connectivity* 1.1, pp. 13–36.

Damoiseaux, Jessica S and Michael D Greicius (2009). "Greater than the sum of its parts: a review of studies combining structural connectivity and resting-state functional connectivity". In: *Brain Structure and Function* 213.6, pp. 525–533.

Bastos, André M and Jan-Mathijs Schoffelen (2016). "A tutorial review of functional connectivity analysis methods and their interpretational pitfalls". In: *Frontiers in systems neuroscience* 9, p. 175.

Kong, Ru, Jingwei Li, Csaba Orban, Mert R Sabuncu, Hesheng Liu, Alexander Schaefer, Nanbo Sun, Xi-Nian Zuo, Avram J Holmes, Simon B Eickhoff, et al. (2019). "Spatial topography of individual-specific cortical networks predicts human cognition, personality, and emotion". In: *Cerebral cortex* 29.6, pp. 2533–2551.

Haxby, James V, J Swaroop Guntupalli, Samuel A Nastase, and Ma Feilong (2020). "Hyperalignment: Modeling shared information encoded in idiosyncratic cortical topographies". In: *Elife* 9, e56601.

Shen, Xilin, Emily S Finn, Dustin Scheinost, Monica D Rosenberg, Marvin M Chun, Xenophon Papademetris, and R Todd Constable (2017). "Using connectome-based predictive modeling to predict individual behavior from brain connectivity". In: *nature protocols* 12.3, pp. 506–518.

Koutra, Danai, Joshua T Vogelstein, and Christos Faloutsos (2013). "Deltacon: A principled massive-graph similarity function". In: *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, pp. 162–170.

Vogelstein, Joshua T, William Gray Roncal, R Jacob Vogelstein, and Carey E Priebe (2012). "Graph classification using signal-subgraphs: Applications in statistical connectomics". In: *IEEE transactions on pattern analysis and machine intelligence* 35.7, pp. 1539–1551.

Broca, Paul (1861). "Remarques sur le siège de la faculté du langage articulé, suivies d'une observation d'aphémie (perte de la parole)". In: *Bulletin et Memoires de la Societe anatomique de Paris* 6, pp. 330–357.

Wernicke, Carl (1874). *Der aphasische Symptomencomplex: eine psychologische Studie auf anatomischer Basis*. Cohn.

Finger, Stanley and C Robert Almli (1985). "Brain damage and neuroplasticity: mechanisms of recovery or development?" In: *Brain Research Reviews* 10.3, pp. 177–186.

Petersen, Alexander, Hans-Georg Müller, et al. (2016). "Functional data analysis for density functions by transformation to a Hilbert space". In: *The Annals of Statistics* 44.1, pp. 183–218.

Scheinost, Dustin, J Benjamin, CM Lacadie, B Vohr, Karen C Schneider, Laura R Ment, Xenios Papademetris, and R Todd Constable (2012). "The intrinsic connectivity distribution: a novel contrast measure reflecting voxel level functional connectivity". In: *Neuroimage* 62.3, pp. 1510–1519.

Nitsche, Michael A, Leonardo G Cohen, Eric M Wassermann, Alberto Priori, Nicolas Lang, Andrea Antal, Walter Paulus, Friedhelm Hummel, Paulo S Boggio, Felipe Fregni, et al. (2008). "Transcranial direct current stimulation: state of the art 2008". In: *Brain stimulation* 1.3, pp. 206–223.

Gorno-Tempini, Maria Luisa, Argye E Hillis, Sandra Weintraub, Andrew Kertesz, Mario Mendez, Stefano Francesco Cappa, Jennifer M Ogar, JD Rohrer, Steven Black, Bradley F Boeve, et al. (2011). "Classification of primary progressive aphasia and its variants". In: *Neurology* 76.11, pp. 1006–1014.

Knopman, David S, Joel H Kramer, Bradley F Boeve, Richard J Caselli, Neill R Graff-Radford, Mario F Mendez, Bruce L Miller, and Nathaniel Mercaldo (2008). "Development of methodology for conducting clinical trials in frontotemporal lobar degeneration". In: *Brain* 131.11, pp. 2957–2968.

Tsapkini, Kyrana, Kimberly T Webster, Bronte N Ficek, John E Desmond, Chiadi U Onyike, Brenda Rapp, Constantine E Frangakis, and Argye E Hillis (2018). "Electrical brain stimulation in different variants of primary progressive aphasia: A randomized clinical trial". In: *Alzheimer's & Dementia: Translational Research & Clinical Interventions* 4, pp. 461–472.

Gandiga, Prateek C, Friedhelm C Hummel, and Leonardo G Cohen (2006). "Transcranial DC stimulation (tDCS): a tool for double-blind sham-controlled clinical studies in brain stimulation". In: *Clinical neurophysiology* 117.4, pp. 845–850.

Ficek, Bronte N, Zeyi Wang, Yi Zhao, Kimberly T Webster, John E Desmond, Argye E Hillis, Constantine Frangakis, Andreia Vasconcellos Faria, Brian Caffo, and Kyrana Tsapkini (2018). "The effect of tDCS on functional connectivity in primary progressive aphasia". In: *NeuroImage: Clinical* 19, pp. 703–715.

Goodman, RA and A Caramazza (1985). "The Johns Hopkins university dysgraphia battery". In: *Baltimore, MD: Johns Hopkins University*.

Mori, Susumu, Dan Wu, Can Ceritoglu, Yue Li, Anthony Kolasny, Marc A Vaillant, Andreia V Faria, Kenichi Oishi, and Michael I Miller (2016). "MRICloud: delivering high-throughput MRI neuroinformatics as cloud-based software as a service". In: *Computing in Science & Engineering* 18.5, pp. 21–35.

Ceritoglu, Can, Xiaoying Tang, Margaret Chow, Darian Hadjiabadi, Damish Shah, Timothy Brown, Muhammad H Burhanullah, Huong Trinh, John Hsu, Katarina A Ament, et al. (2013). "Computational analysis of LDDMM for brain mapping". In: *Frontiers in neuroscience* 7, p. 151.

Miller, Michael I, M Faisal Beg, Can Ceritoglu, and Craig Stark (2005). "Increasing the power of functional maps of the medial temporal lobe by using large deformation diffeomorphic metric mapping". In: *Proceedings of the National Academy of Sciences* 102.27, pp. 9685–9690.

Tang, Xiaoying, Kenichi Oishi, Andreia V Faria, Argye E Hillis, Marilyn S Albert, Susumu Mori, and Michael I Miller (2013). "Bayesian parameter estimation and segmentation in the multi-atlas random orbit model". In: *PloS one* 8.6, e65591.

Faria, Andreia V, Suresh E Joel, Yajing Zhang, Kenichi Oishi, Peter CM van Zjil, Michael I Miller, James J Pekar, and Susumu Mori (2012). "Atlas-based analysis of resting-state functional connectivity: Evaluation for reproducibility and multi-modal anatomy–function correlation studies". In: *Neuroimage* 61.3, pp. 613–621.

Behzadi, Yashar, Khaled Restom, Joy Liau, and Thomas T Liu (2007). "A component based noise correction method (CompCor) for BOLD and perfusion based fMRI". In: *Neuroimage* 37.1, pp. 90–101.

Gu, Chong and Chunfu Qiu (1993). "Smoothing spline density estimation: Theory". In: *The Annals of Statistics*, pp. 217–234.

Silverman, Bernard W (1986). *Density estimation for statistics and data analysis*. Vol. 26. CRC press.

Ramsay, James O (2004). "Functional data analysis". In: *Encyclopedia of Statistical Sciences* 4.

Ramsay, James O and Bernard W Silverman (2007). *Applied functional data analysis: methods and case studies*. Springer.

McLean, Mathew W, Giles Hooker, Ana-Maria Staicu, Fabian Scheipl, and David Ruppert (2014). "Functional generalized additive models". In: *Journal of Computational and Graphical Statistics* 23.1, pp. 249–269.

Prentice, Ross L and Ronald Pyke (1979). "Logistic disease incidence models and case-control studies". In: *Biometrika* 66.3, pp. 403–411.

Rothman, Kenneth J, Sander Greenland, and Timothy L Lash (2008). "Case–control studies". In: *Encyclopedia of Quantitative Risk Analysis and Assessment* 1.

Egozcue, Juan José, José Luis Díaz-Barrero, and Vera Pawlowsky-Glahn (2006). "Hilbert space of probability density functions based on Aitchison geometry". In: *Acta Mathematica Sinica* 22.4, pp. 1175–1182.

Reiss, Philip T and R Todd Ogden (2007). "Functional principal component regression and functional partial least squares". In: *Journal of the American Statistical Association* 102.479, pp. 984–996.

Ghanem, Roger G and Pol D Spanos (2003). *Stochastic finite elements: a spectral approach*. Courier Corporation.

Di, Chong-Zhi, Ciprian M Crainiceanu, Brian S Caffo, and Naresh M Punjabi (2009). "Multilevel functional principal component analysis". In: *The annals of applied statistics* 3.1, p. 458.

Wood, Simon N (2004). "Stable and efficient multiple smoothing parameter estimation for generalized additive models". In: *Journal of the American Statistical Association* 99.467, pp. 673–686.

Wahba, Grace (1983). "Bayesian "confidence intervals" for the cross-validated smoothing spline". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 45.1, pp. 133–150.

Nychka, Douglas (1988). "Bayesian confidence intervals for smoothing splines". In: *Journal of the American Statistical Association* 83.404, pp. 1134–1143.

Benjamini, Yoav and Yosef Hochberg (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the Royal statistical society: series B (Methodological)* 57.1, pp. 289–300.

Dezeure, Ruben, Peter Bühlmann, Lukas Meier, and Nicolai Meinshausen (2015). "High-dimensional inference: Confidence intervals, p-values and r-software hdi". In: *Statistical science*, pp. 533–558.

Gu, Chong and Jingyuan Wang (2003). "Penalized likelihood density estimation: Direct cross-validation and scalable approximation". In: *Statistica Sinica*, pp. 811–826.

Oliveira, Francisco PM and Joao Manuel RS Tavares (2014). "Medical image registration: a review". In: *Computer methods in biomechanics and biomedical engineering* 17.2, pp. 73–93.

Goldsmith, Jeff, Ciprian M Crainiceanu, Brian S Caffo, and Daniel S Reich (2011). "Penalized functional regression analysis of white-matter tract profiles in multiple sclerosis". In: *NeuroImage* 57.2, pp. 431–439.

Goldsmith, Jeff, Ciprian M Crainiceanu, Brian Caffo, and Daniel Reich (2012). "Longitudinal penalized functional regression for cognitive outcomes on neuronal tract measurements". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 61.3, pp. 453–469.

Goldsmith, Jeff, Matt P Wand, and Ciprian Crainiceanu (2011). "Functional regression via variational Bayes". In: *Electronic journal of statistics* 5, p. 572.

Reiss, Philip T, Jeff Goldsmith, Han Lin Shang, and R Todd Ogden (2017). "Methods for scalar-on-function regression". In: *International Statistical Review* 85.2, pp. 228–249.

Breslow, Norman E (1996). "Statistics in epidemiology: the case-control study". In: *Journal of the American Statistical Association* 91.433, pp. 14–28.

Greenhouse, Samuel W (1982). "Jerome Cornfield's contributions to epidemiology". In: *Biometrics*, pp. 33–45.

Calhoun, Vince D, Tulay Adali, Godfrey D Pearlson, and James J Pekar (2001). "A method for making group inferences from functional MRI data using independent component analysis". In: *Human brain mapping* 14.3, pp. 140–151.

Hutchison, R Matthew, Thilo Womelsdorf, Elena A Allen, Peter A Bandettini, Vince D Calhoun, Maurizio Corbetta, Stefania Della Penna, Jeff H Duyn, Gary H Glover, Javier Gonzalez-Castillo, et al. (2013). "Dynamic functional connectivity: promise, issues, and interpretations". In: *Neuroimage* 80, pp. 360–378.

Haan, Willem de, Wiesje M van der Flier, Huijuan Wang, Piet FA Van Mieghem, Philip Scheltens, and Cornelis J Stam (2012). "Disruption of functional brain networks in Alzheimer's disease: what can we learn from graph spectral analysis of resting-state magnetoencephalography?" In: *Brain connectivity* 2.2, pp. 45–55.

Sporns, Olaf (2010). *Networks of the Brain*. MIT press.

Bullmore, Ed and Olaf Sporns (2009). "Complex brain networks: graph theoretical analysis of structural and functional systems". In: *Nature reviews neuroscience* 10.3, pp. 186–198.

Szabó, Zoltán, Bharath K Sriperumbudur, Barnabás Póczos, and Arthur Gretton (2016). "Learning theory for distribution regression". In: *The Journal of Machine Learning Research* 17.1, pp. 5272–5311.

Chen, Yaqing, Zhenhua Lin, and Hans-Georg Müller (2021). "Wasserstein regression". In: *Journal of the American Statistical Association*, pp. 1–14.

Reiss, Philip T, Lei Huang, and Maarten Mennes (2010). "Fast function-on-scalar regression with penalized basis expansions". In: *The international journal of biostatistics* 6.1.

Chapter 3

Information Rates of Bayesian Distributional Regression

Bohao Tang [1], *Abhirup Datta*[1], *Yi Zhao* [2], *Brian Caffo*[1]

3.1 Introduction

This chapter considers the estimation error bounds for a regression setting $y = f(x) + \varepsilon$, where the outcome, y, is a scalar and the regressor, x, is an unknown probability distribution of which a finite sample is observed. Regressing with distributions, e.g. distributions as outcomes, predictors or both, is sometimes called "distribution regression" (Szabó et al., 2016; Oliva, Póczos, and Schneider, 2013; Fang, Guo, and Zhou, 2020; Law et al., 2018) with the associated terms: "scalar on distribution", "distribution on distribution" and "distribution on scalar" used for the specific outcome and predictor settings. Standard methods for the scalar outcome case is to estimate the density function through a kernel density estimate, or other equivalent convergent estimator, and utilize typical functional regression

[1] Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health
[2] Department of Biostatistics, Indiana University School of Medicine

techniques. Póczos et al., 2013 and Oliva et al., 2014 prove the consistency of this method. It should be emphasized that this approach is only applicable if the outcome is scalar. Another well known method is embedding the distributions in a reproducing kernel Hilbert space (RKHS) via a kernel mean embedding and then applying RKHS regression. For example, a multivariate probability distribution, P, on $\mathbb{R}^d$ would be embedded as:

$$\mu_p(t) = \int_{\mathbb{R}^d} K(t,s)P(ds), \tag{3.1}$$

where K is a Mercer kernel (i.e. symmetric and positive semidefinite). Changing P to the empirical distribution, $\hat{P}$, gives one the empirical kernel mean embedding, $\mu_{\hat{p}}$, that one can use with observed data. Under typical assumptions, Szabó et al., 2016 showed this method can achieve optimal bounds with $\hat{P}$ for excess risk, both for scalar outcome and distributional outcomes.

Gaussian process regression is a popular Bayesian non-parametric method used for scalar on vector regression $y = f(\boldsymbol{x}) + \varepsilon$, for scalar y and vector, x. Gaussian process regression has a close connection to RKHS regression in that the posterior mean of Gaussian process regression is the same as the minimizer of a RKHS regression with the same kernel and suitable regularization parameters. Asymptotics for such methods are also well established (Vaart and Zanten, 2008; Van Der Vaart and Van Zanten, 2011; Sniekers and Vaart, 2015; Choi, 2007). For example, Van Der Vaart and Van Zanten, 2011 proved that the posterior risk for the L_2 norm:

$$\mathbb{E}_{f_0} \int \|f - f_0\|_2^2 \, d\Pi(f|X_{1:n}, Y_{1:n}) \tag{3.2}$$

are bounded by an optimal rate for f_0 in *Hölder space*, where f_0 is the true function and the expectations are under the true data generating process. Notably, here one

can not only bound the prediction error, but also the estimation error, $\|f - f_0\|_2$.

We would like to explore whether similar results hold when $\boldsymbol{x}$ is generalized to be a multivariate Gaussian process. Consider a linear case generalization as follows:

$$y = \mathbb{E}_{\boldsymbol{x}} f + \varepsilon,$$

where $\mathbb{E}_{\boldsymbol{x}}$ represents expectation under the distribution associated with $\boldsymbol{x}$, i.e. $\mathbb{E}_{\boldsymbol{x}} f = \int_{-\infty}^{\infty} f(x) P_{\boldsymbol{x}}(dx)$. Here, note that f references a functional transformation of interest, not the density associated with $\boldsymbol{x}$. Specifically, as $P_{\boldsymbol{x}}$ limits to full probability at a single point, or equivalently that the associated covariate density is a Dirac mass, the problem degenerates to the typical scalar on vector regression, and hence this distributional formulation is a generalization. Furthermore, if all $\boldsymbol{x}$ have known associated densities, the problem becomes a weighted form of scalar on functional linear function regression, $y = \int f(x) P_{\boldsymbol{x}}(dx) + \varepsilon$, where it is known that optimal estimation error bounds can only be achieved under restrictive assumptions. Specifically, the alignment of the eigenspace of the RKHS that f_0 lies in and the covariance kernel $\mathrm{Cov}(p(\boldsymbol{x}_1), p(\boldsymbol{x}_2))$ of the densities p possesses (Yuan and Cai, 2010).

In practice, one only observes finite samples from the $\boldsymbol{x}$ distribution. That is, there is a two stage sampling procedure, first of individuals, and secondly of samples from the realized covariate density. Specifically, for every subject, one observes a finite empirical sample, $\{\boldsymbol{x}_{ij}\}_j$, from its distributional covariate, $\boldsymbol{x}_i$. We propose a unified method that directly utilizes the covariate density samples, $\boldsymbol{x}_{ij}$, and in the process obtain optimal estimation error bounds for regular functions, f, under assumptions on the distributional process of $\boldsymbol{x}$ that are practically acceptable

and empirically verifiable.

To compare our contributions with existing related work, Póczos et al., 2013 and Oliva et al., 2014 use kernel density estimators as an intermediate step and thus only consistency of the regression is shown. Szabó et al., 2016 can be regard as a fully non-linear version of distribution regression. However, that work focuses on prediction error bounds and requires the outcome, y, to be bounded, or have bounded support if y is a distribution. Our work relieves the boundedness assumption of the outcome and we focus on how well the regressor f can be estimated. In addition, we quantify uncertainty through full Bayesian modeling. Augustin et al., 2017 uses a similar model which can directly work on samples of individual distributions. However, they don't consider the underlying data generation process, therefore no theoretical guarantee is presented. Law et al., 2018 is the only Bayesian distribution regression method we observed in the literature. However, in this work, fiducial landmark points, rather than full samples, are used and priors are put on the kernel mean embeddings of those landmarks points. Also through these landmark points they only consider a restrictive set of regressors and no theoretical properties are shown. We instead consider a fully non-parameteric regression functions, f, only assuming certain regularities. Furthermore, we put priors directly on the functions.

3.2 Gaussian Process Distributional Regression

Consider estimating f from a model, $y_i = \mathbb{E}_{z_i} f + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2), i \in \{1, 2, \cdots, n\}$ and z_i are the individual distributions. [Recall, $E_z f = \int_{-\infty}^{\infty} f(x) P_z(dx)$.] Here, we change the distributional covariate notation from x to z, because we want

to clearly distinguish the difference between individual distributions and their observed samples. We will use z_i for distributions and x_{ij} for samples. Typically, one does not observe z, or P_z. Instead, one observes sample points, $x_i = \{x_{ij}\}_{j=1}^{m_i}$, and thus it is necessary to approximate the expectation with sample means. We further add a Gaussian process prior for f to perform non-parametric regression:

$$y_i \sim \mathcal{N}\left(\frac{1}{m_i}\sum_j f(x_{ij}), \sigma^2\right) \tag{3.3}$$

$$f \sim GP(0, K) \tag{3.4}$$

where K is a covariance kernel for the Gaussian process. Conveniently, $\frac{1}{m_i}\sum_j f(x_{ij})$ are jointly Gaussian distributed, since f follows a Gaussian process prior and the posterior process for f given the observed data is also a Gaussian process. Specifically, for given σ^2 we have:

$$\mathbb{E}[f(s) \mid X, Y] = \boldsymbol{l}^T(s)(M + \sigma^2)^{-1}Y \tag{3.5}$$

$$\mathrm{Cov}[f(s) \mid X, Y] = K(s, s) - \boldsymbol{l}^T(s)(M + \sigma^2)^{-1}\boldsymbol{l}(s) \tag{3.6}$$

where $\boldsymbol{l}_i(s) = \frac{1}{m_i}\sum_j K(x_{ij}, s)$ and $M_{ij} = \frac{1}{m_i m_j}\sum_{uv} K(x_{iu}, x_{jv})$.

Assume the samples x_{ij} are i.i.d samples from some distribution. This can be either a well motivated assumption or simply a useful working model for the assumption that samples are exchangeable. Therefore, for any regression model, $y_i = G(x_i) + \varepsilon_i$, where G is a mapping from the set $\{x_{ij}\}_j$ to a real number we minimally require G to be invariant to any permutation of the second index (j). From Zaheer et al., 2017 it is known that $G(\{x_{ij}\}_j) = \rho_i\left(\sum_j \phi_i(x_{ij})\right)$ for some functions ρ_i and ϕ_i. Our model then comes naturally with $\rho_i(x) = x$ and $\phi_i = \frac{1}{m_i}f$

shared across all subjects. The normalizing factor, $\frac{1}{m_i}$, is essential in statistical settings, since it is desirable to have different number of samples to contribute in the same magnitude.

Our model is also invariant to any invertible transformations of the observed x_{ij}. If only $w_{ij} = h(x_{ij})$ are observed, where h is invertible, Model 3.3 still holds for $\{w_i\}$ with $y_i \sim \mathcal{N}\left(\frac{1}{m}\sum_j f \circ h^{-1}(w_{ij}), \sigma^2\right)$. Therefore, the model is robust to fixed systematic bias in the measurements of the covariates.

3.3 Low Rank Approximation

Posterior inferences for Gaussian process regression, like 3.3, is typically slow when one has a large sample size, since it is $O(n^3)$ in complexity with n being number of subjects. Distributional regression settings can be an even greater computational challenge, since every subject involves many samples. When using native posterior inference algorithms with Equations 3.5, 3.6 the overall time complexity is $O(n^3 + n^2m^2)$, where m is the number of samples for each subject.

In the practical settings where an approximate posterior inference is acceptable, one can perform a low rank approximation technique to accelerate the algorithm. From Equations 3.5,3.6 we consider representing f as:

$$f(s) = \sum_i w_i l_i(s), \quad \text{with } \boldsymbol{w} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{M}^{-1}\right) \tag{3.7}$$

with basis $l_i(s) = \frac{1}{m_i}\sum_j K(x_{ij}, s)$ and covariance matrix $\boldsymbol{M}_{ij} = \frac{1}{m_i m_j}\sum_{uv} K(x_{iu}, x_{jv})$. The predicted $\hat{y}_i = \frac{1}{m_i}\sum_j f(x_{ij}) = \sum_j \boldsymbol{M}_{ij} w_j$ and we have the matrix representation $\hat{\boldsymbol{y}} = \boldsymbol{M}\boldsymbol{w}$. Therefore, the posterior mode for $\hat{\boldsymbol{w}}$ could be found by minimizing

$$\|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{w}\|^2 / \sigma^2 + \boldsymbol{w}^\top \boldsymbol{M}\boldsymbol{w} \tag{3.8}$$

Typical low rank method produced by the eigen approximation could then be easily performed. Let $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$ be the eigen decomposition of $\boldsymbol{M}$. Fixing a number, k, one would approximate $\boldsymbol{M}$ as $\boldsymbol{U}_k\boldsymbol{D}_k\boldsymbol{U}_k^\top$ where $\boldsymbol{U}_k$ consists of the first k eigenvectors and $\boldsymbol{D}_k$ is the diagonal matrix of first k eigenvalues. Restricting $\boldsymbol{w}$ be the column space of $\boldsymbol{U}_k$ with $\boldsymbol{w} = \boldsymbol{U}_k\boldsymbol{w}_k$, the optimization 3.8 becomes

$$\min_{\boldsymbol{w}_k} \|\boldsymbol{y} - \boldsymbol{U}_k\boldsymbol{D}_k\boldsymbol{w}_k\|^2 / \sigma^2 + \boldsymbol{w}_k^\top \boldsymbol{D}_k \boldsymbol{w}_k \tag{3.9}$$

Such optimization and its low rank approximation 3.9 is very similar to thin plate splines Wood, 2003. Therefore, the entire process could be efficiently implemented using typical spline regression packages, such as *mgcv* in the R language. Through such approximations we could reduce the time complexity of the regression part to $O(n^2k)$ using a suitable Lanczos algorithm. Unfortunately, the $O(n^2m^2)$ part of computing $\boldsymbol{M}$ cannot be avoided. Therefore, the algorithm is still costly with large m. However, since $\boldsymbol{M}_{ij}$ is just performing averages, the computation could be fully paralleled. Also one could down-sample the observations, x_{ij}, or bin them into tractable sizes.

3.4 Main Results

3.4.1 Notation and Assumptions

Suppose one has n observations, (z_i, y_i), where z_i is the subject related distributions, which are assumed to be i.i.d., following some distributional process, $\mathcal{Z}$, and y_i is the scalar outcome. In reality one cannot directly observe z_i. Instead, one has m_i i.i.d. samples, x_{ij}, where $x_{ij} \in \mathbb{R}^d$. For simplicity we will assume $m_i = m$ for all i. However, it is easy to see that results do not change if one relaxes to $m_i \geq m$ for

all i. We will denote $\mathbb{D}_n = \{\{x_{ij}\}_j, y_i\}_{i=1}^n$ and $\mathbb{Z}_n = \{z_i, y_i\}_{i=1}^n$. Further assume z_i are n random i.i.d. distributions that follow a distributional process, $\mathcal{Z}$. Further y_i follows the model:

$$y_i = \mathbb{E}_{z_i} f_0 + \varepsilon_i \tag{3.10}$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ mutually independent with each other and all z_i. We would like to fit f_0 from following model:

$$f \sim GP(0, K) \tag{3.11}$$

$$y_i \overset{i.n.d.}{\sim} \mathcal{N}\left(\frac{1}{m}\sum_{j=1}^{m} f(x_{ij}), \sigma^2\right) \tag{3.12}$$

We now show that the the posterior risk:

$$R_n = \mathbb{E}_{f_0} \int ||f - f_0||^2 d\Pi_n(f|\mathbb{D}_n) \tag{3.13}$$

contracts at optimal rate, where Π_n is the posterior distribution, $\mathbb{E}_{f_0}$ is relative to the distribution of $\mathbb{D}_n$ and norm $\|\cdot\|$ can be either an empirical norm, $\|\cdot\|_n$, or L_2 norm $\|\cdot\|_2$. The empirical norm is naturally defined as

$$\|f\|_n^2 = \frac{1}{n}\sum_{i=1}^{n} (\mathbb{E}_{z_i} f)^2 \tag{3.14}$$

Notice the expectation in this norm is respective to underlying distribution, z_i, which one can't directly observe.

To achieve estimation error bound we need the true function, f_0, to be regular in some way. That is, $f_0 \in \mathcal{F}$ for some functional space $\mathcal{F}$ with good properties. Herein, we focus on0 the *Hölder space* $C^\alpha[0,1]^d$ for $\alpha > 0$. When writing $\alpha = k + \eta$, the space $C^\alpha[0,1]^d$ is the space of all functions supported in $[0,1]^d$, whose partial derivatives of orders $(l_1, \cdots, l_d)$ exist for all nonnegative integers $l_1, \cdots, l_d$ such

that $l_1 + \cdots + l_d \leq k$ and for which the highest order partial derivatives are Hölder continuous with order η. (f being Hölder continuous with order η if $|f(x) - f(y)| \leq C \|x - y\|^\eta$ for all x, y and some constant C.)

Another functional space worth studying is the *Sobolev space*, $H^\alpha[0,1]^d$, which contains all $[0,1]^d \to \mathbb{R}$ functions f such that

$$\int_{\mathbb{R}^d} \left(1 + \|\lambda\|^2\right)^\alpha \left|\hat{f}(\lambda)\right|^2 d\lambda < \infty \tag{3.15}$$

where $\hat{f}$ is the Fourier transformation of f: $\hat{f}(\lambda) = (2\pi)^{-d} \int \exp(i\lambda^T t) f(t) dt$. We will call a function, f, to be α-regular in $[0,1]^d$ if $f \in C^\alpha[0,1]^d \cap H^\alpha[0,1]^d$. α-regular class will be the main functional class considered in our results.

As explained in the Introduction, one also needs constraints on the distributional process, $z_i \sim \mathcal{Z}$. Denote $\mathcal{S}$ to be the support of $\mathcal{Z}$. We assume the mean measure $\mu : \mu(A) = \mathbb{E}_{z_i \sim \mathcal{Z}}[z_i(A)]$ has density $\mu(x)$ whose support its $[0,1]^d$ and is bounded away from 0.

Notice that our model is invariant to any invertiable transformation of z_i. Therefore, one can always map the support of μ to $[0,1]^d$. Without loss of generality, we regard $\mu(x) = 1$. And more importantly we will need properties that make $\mathcal{Z}$ be able to separate regular functions:

Definition 3.4.1. We call a distributional process $\mathcal{Z}$ **weakly separates** a functional vector space $\mathcal{F}$ if and only if $\forall f_1, f_2 \in \mathcal{F} :\ \mathbb{P}_{z \sim \mathcal{Z}}[\mathbb{E}_z f_1 = \mathbb{E}_z f_2] = 1 \Leftrightarrow f_1 = f_2$. And we call $\mathcal{Z}$ **strongly separates** $\mathcal{F}$ if and only if there exists constant C such that $\mathbb{E}_\mu f^2 \leq C \mathbb{E}_{z \sim \mathcal{Z}}[(\mathbb{E}_z f)^2]$ for all $f \in \mathcal{F}$. Here μ is the expectation of $\mathcal{Z}$.

One can easily check that strong separability contains weak separability if $\mathbb{E}_\mu f^2 = 0 \Rightarrow f = 0$ for $f \in \mathcal{F}$. And using stick breaking representation, we can

show that Dirichlet process $\mathrm{DP}(\mu, \alpha)$ strongly separates bounded functions with constant $C = 1 + \alpha$. Proof is shown in Lemma 2.

Lemma 2. *Dirichlet process $DP(\mu, \alpha)$ strongly separates the space of bounded functions on $[0,1]^d$ for any measure μ supported within it.*

Proof. Using stick breaking representation, we know that the sample probability mass function $p(x)$ has the form

$$p(x) = \sum_{k=1}^{\infty} \beta_k \cdot \delta_{x_k}(x), \tag{3.16}$$

where $\beta_k = \beta'_k \prod_{i=1}^{k-1}(1 - \beta'_i)$ for β'_k i.i.d follows $\mathrm{Beta}(1, \alpha)$ and x_k i.i.d follows μ with δ_x the point mass at x. Also it is clear that the mean measure for $\mathrm{DP}(\mu, \alpha)$ is just μ. Therefore for bounded f, we can directly calculate $\mathbb{E}_{z\sim\mathrm{DP}(\mu,\alpha)}[(\mathbb{E}_z f)^2]$. We have:

$$\begin{aligned}
\mathbb{E}_{z\sim\mathrm{DP}(\mu,\alpha)}[(\mathbb{E}_z f)^2] &= \mathbb{E}_{z\sim\mathrm{DP}(\mu,\alpha)}\left[\left(\sum_{k=1}^{\infty} \beta_k f(x_k)\right)^2\right] \\
&= \mathbb{E}_{z\sim\mathrm{DP}(\mu,\alpha)}\left[\sum_{ij}^{\infty} f(x_i) f(x_j) \beta_i \beta_j\right] \\
&= \mathbb{E}_\mu f^2 \cdot \sum_i \frac{2}{\alpha(1+\alpha)} \left(\frac{\alpha}{2+\alpha}\right)^i + (\mathbb{E}_\mu f)^2 \sum_{i\neq j} \mathbb{E}[\beta_i \beta_j] \\
&\geq \frac{1}{1+\alpha} \mathbb{E}_\mu f^2
\end{aligned}$$

Therefore $\mathrm{DP}(\mu, \alpha)$ strongly separates the bounded functions with constant $C = 1 + \alpha$. □

In the following results we will use a special covariance kernel K for the

Gaussian process prior, which is called Matérn kernel and is widely used in spatial statistics and non-parameteric regression (and the corresponding Gaussian process is called Matérn process). An order α Matérn kernel for d dimensional process has the form:

$$K(s,t) = \int_{\mathbb{R}^d} \frac{e^{i\lambda^T(s-t)}}{(1+\|\lambda\|^2)^{\alpha+d/2}} d\lambda \tag{3.17}$$

From Van Der Vaart and Van Zanten, 2011 we know that the sample paths of order α Matérn process is α-regular in the sense that it belongs to $C^\beta[0,1]^d \cap H^\alpha[0,1]^d$ for any $\beta < \alpha$.

In the following sections we will use $\mathcal{H}_K$ to denote the reproducing kernel Hilbert space (RKHS) with kernel K and when it is not misleading $\mathcal{H}$ represents the RKHS of the Gaussian process prior (RKHS of the Gaussian process covariance kernel). And correspondingly $\|\cdot\|_{\mathcal{H}}$ would be the RKHS-norm. We will assume $K(s,t) \leq \kappa$ for all s,t. Such κ clearly exists for all Matérn kernel.

3.4.2 Fixed Design

Follow the notation and assumption in section 3.4.1. Given n data $y_1, \cdots, y_n$ generated from model 3.10 and i.i.d samples $x_{ij} \sim z_i$. If f_0 is continuous and bounded within $[0,1]^d$, we can bound the posterior risk 3.13 by the so called concentration function Vaart and Zanten, 2008:

$$\phi_{f_0}(\varepsilon) = \inf_{h \in \mathcal{H}: \|h - f_0\|_\infty < \varepsilon} \|h\|_{\mathcal{H}}^2 - \log \mathbb{P}\left(\|f\|_\infty \leq \varepsilon\right) \tag{3.18}$$

where the probability $\mathbb{P}$ is relative to the Gaussian process prior of f and the associated function:

$$\Psi_{f_0}(\varepsilon) = \frac{\phi_{f_0}(\varepsilon)}{\varepsilon^2} \tag{3.19}$$

whose order of magnitude has been well studied in Van Der Vaart and Van Zanten, 2011.

Theorem 3. *If the distributional process $\mathcal{Z}$ **weakly separates** $C^\beta[0,1]^d \cap H^\beta[0,1]^d$. Then for any β-regular function f_0 and model 3.10 with order α Matérn kernel. There exists a constant C independent of n such that the posterior risk is controlled as*

$$\mathbb{E}_{f_0} \int ||f - f_0||_n^2 \, d\Pi_n(f|\mathbb{D}_n) \leq C n^{-\frac{2\min(\beta,\alpha)}{2\alpha+d}} \tag{3.20}$$

given $m = \Omega(n^{C_{\alpha,\beta}})$, where $C_{\alpha,\beta}$ is a complex constant depend only on α and β. More interestingly optimal rate can be achieved when $\alpha = \beta$ and $m = \Omega\left(n^{2+\frac{d}{\beta}+\frac{4\beta}{2\beta+d}}\right)$.

We will discuss the order $C_{\alpha,\beta}$ of m in detail in section 3.6.2.

3.4.3 Random Design

For estimation error bound, one could consider L_2 norm correspond to the mean measure μ of the distributional process $\mathcal{Z}$. For any f that is μ-measurable with following integral finite, we define:

$$\|f\|_2^2 = \int f^2 d\mu \tag{3.21}$$

Theorem 4. *If the distributional process $\mathcal{Z}$ **strongly separates** $C^\beta[0,1]^d \cap H^\beta[0,1]^d$. Then for any β-regular function f_0 and model 3.10 with order α Matérn kernel. There exists a constant C independent with n such that the posterior risk is controlled as*

$$\mathbb{E}_{f_0} \int ||f - f_0||_2^2 \, d\Pi_n(f|\mathbb{D}_n) \leq C n^{-\frac{2\min(\beta,\alpha)}{2\alpha+d}} \tag{3.22}$$

given $m = \Omega(n^{C_{\alpha,\beta}})$ and $\min(\alpha,\beta) > d/2$, where $C_{\alpha,\beta}$ is a complex constant depend only on α and β. More interestingly optimal rate can be achieved when $\alpha = \beta$ and

$m = \Omega\left(n^{2+\frac{d}{\beta}+\frac{4\beta}{2\beta+d}}\right)$.

When $\alpha = \beta$ we get the best rate as $n^{-\beta/(2\beta+d)}$ (after square root). This is already the optimal rate if we don't make any other assumptions on the distributional process $\mathcal{Z}$. One can easily check that when z equals delta measure almost surely, $\mathcal{Z}$ strongly separates all function space and our model degenerate to typical Gaussian process regression. Therefore the optimal rate we can get is the rate for estimating β-regular functions in typical nonparametric regression case, which is $n^{-\beta/(2\beta+d)}$.

Notice that although we require $\alpha = \beta$ to achieve the optimal rate. We can actually require smaller m with smaller α if the only goal is consistency. For example, from section 3.6.2 we know that if $\beta > \alpha + 1/2$, consistency only requires $m/n^2 \to \infty$, instead with $m/n^{2+d/\beta} \to \infty$ if $\alpha = \beta$.

3.5 Simulations

One difference of our method comparing to existing ones, like Póczos et al., 2013, is that our expectation model does not require us to estimate the underlying densities. Because the sample mean is the best estimator of the expectation in non-parametric sense, we would also expect that our model to do better than functional linear models with estimated densities. Also, by using full samples, our model should have better asymptotic performance compared to the model with only fixed amount landmark points. In this section we conduct a simulation study to show that our method converges and have better rate compared to its estimated densities alternatives and the Bayesian model in Law et al., 2018.

We simulate our data as

$$z_i \sim DP\left(\text{Unif}[0,1], 25\right) \tag{3.23}$$

$$x_{ij} \sim z_i; \quad \varepsilon_i \sim \mathcal{N}(0, 0.01) \tag{3.24}$$

$$y_i = \mathbb{E}_{z_i}(f) + \varepsilon_i \tag{3.25}$$

where $f(x) = 10x \cdot \exp(-5x)$ is infinitely smooth within $[0,1]$.

We draw n by m x_{ij} samples from the distribution 3.24 and corresponding n y_i samples from 3.25, where n ranges from within $\{50, 100, 200, 300, 400\}$ and m ranges from within $\{50, 100, 250, 500, 1000, 2000\}$. We perform the exact posterior inference using Equations 3.5, and 3.6, not the low rank approximations, and subsequently compare the empirical risk $\int \left\| f - \hat{f} \right\|_2^2 d\Pi(\hat{f} \mid \{X, Y\})$ estimated with 100 samples from the posterior process $\Pi(\hat{f} \mid \{X, Y\})$ for each combination of n and m. For the *Bayesian density regression* model introduced in Law et al., 2018, we use 10 and 50 evenly spaced landmark points in $[0,1]$ and set all other hyperparameters as default ones. We also compare our method with a direct density estimation alternative, that is to replace empirical expectation $\sum_j f(x_{ij})/m$ with $\mathbb{E}_{\hat{z}_i} f$ in model 3.3, where $\hat{z}_i$ is a density estimated from a kernel density estimator.

$$y_i \sim \mathcal{N}\left(\mathbb{E}_{\hat{z}_i}(f), \sigma^2\right) \tag{3.26}$$

$$f \sim GP(0, K) \tag{3.27}$$

It is easy to see that this alternative is a functional linear regression in a Reproducing Kernel Hilbert Space with the same kernel K as in model 3.3.

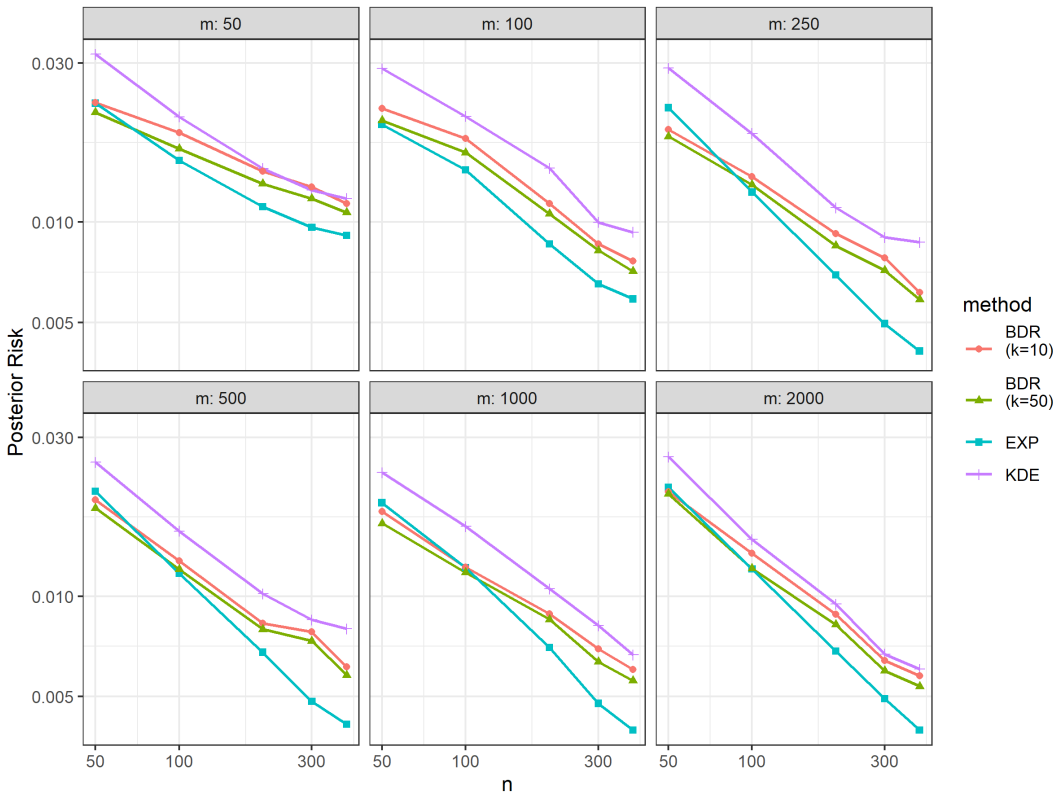


Figure 3.1: The empirical posterior risk for every combination of n and m under 100 runs, where n stands for the number of subjects and m is the number of samples for each subject. **EXP** label in *method* column stands for our model 3.3 using empirical expectation, and **KDE** stands for its direct alternative 3.26 using kernel density estimated expectations. **BDR** stands for the Bayes distribution regression method suggested in Law et al., 2018 where k is the number of landmark points.

We use a typical Gaussian kernel for all models, $K(s,t) = \exp\left(-(s-t)^2/2l\right)$, where l is set to 0.25. We run every setting 100 independent times and report the final mean empirical risk with confidence intervals. The results are shown in Figure 3.1. It is clear that the algorithm converges at a polynomial rate when n is increased for sufficiently large m. Furthermore, our method is significantly better and have sharper rates than the Bayesian density regression model (Law et al., 2018) and the kernel density estimation alternative.

3.6 Proofs

3.6.1 Risk Decomposition

The way we prove the theorems is to do risk decomposition. Consider the Gaussian process model with unknown true distributions z_i:

$$f \sim GP(0, K) \tag{3.28}$$

$$y_i \sim \mathcal{N}\left(\mathbb{E}_{z_i} f, \sigma^2\right) \tag{3.29}$$

the posterior of which is denoted as $\Pi_n(f|\mathbb{Z}_n)$. Then the risk term 3.13 can be decomposed into:

$$R_n = R_n^0 + R_n^1 = \mathbb{E}_{f_0} \int ||f - f_0||^2 d\Pi_n(f|\mathbb{Z}_n) + \left(\mathbb{E}_{f_0} \int ||f - f_0||^2 (d\Pi_n(f|\mathbb{D}_n) - d\Pi_n(f|\mathbb{Z}_n))\right) \tag{3.30}$$

where $\|\cdot\|$ can be empirical norm $\|\cdot\|_n$ and L_2 norm $\|\cdot\|_2$. We can bound $\mathbb{R}_n^0$ using similar method as in Van Der Vaart and Van Zanten, 2011 and bound $\mathbb{R}_n^1$ through a direct computation.

3.6.2 Bound R_n^1 for L_2 norm $\|\cdot\|_2$

A naive bound for R_n^1 is to directly calculate it out:

$$
\begin{aligned}
R_n^1 &= \mathbb{E}_{f_0} \int ||f - f_0||^2 (d\Pi_n(f|\mathbb{D}_n) - d\Pi_n(f|\mathbb{Z}_n)) \\
&= \mathbb{E}_{f_0} \int_{[0,1]^d} \left(\mathbb{E}_{f|\mathbb{D}_n}(f - f_0)^2(s) - \mathbb{E}_{f|\mathbb{Z}_n}(f - f_0)^2(s) \right) ds \\
&= \mathbb{E}_{f_0} \int_{[0,1]^d} \left(\mathbb{E}_{f|\mathbb{D}_n}(f^2(s) - 2f(s)f_0(s)) - \mathbb{E}_{f|\mathbb{Z}_n}(f^2(s) - 2f(s)f_0(s)) \right) ds \\
&= \mathbb{E}_{f_0} \int_{[0,1]^d} \mathrm{Var}(f(s)|\mathbb{D}_n) - \mathrm{Var}(f(s)|\mathbb{Z}_n) + \mathbb{E}^2(f(s)|\mathbb{D}_n) - \mathbb{E}^2(f(s)|\mathbb{Z}_n) \\
&\qquad\qquad - 2f_0(s) \left(\mathbb{E}(f(s)|\mathbb{D}_n) - \mathbb{E}(f(s)|\mathbb{Z}_n) \right) ds \\
&= \mathbb{E}_{f_0} \int_{[0,1]^d} V(s) + E_2(s) - 2f_0(s)E_1(s) ds
\end{aligned}
$$

where $V(s) = \mathrm{Var}(f(s)|\mathbb{D}_n) - \mathrm{Var}(f(s)|\mathbb{Z}_n)$, $E_1(s) = \mathbb{E}_{\boldsymbol{\varepsilon}} \left[\mathbb{E}(f(s)|\mathbb{D}_n) - \mathbb{E}(f(s)|\mathbb{Z}_n) \right]$ and $E_2(s) = \mathbb{E}_{\boldsymbol{\varepsilon}} \left[\mathbb{E}^2(f(s)|\mathbb{D}_n) - \mathbb{E}^2(f(s)|\mathbb{Z}_n) \right]$. All exchange of integral and expectation above should be legal because all functions are clearly bounded by a constant if fixing n.

Borrowing similar notation as in Szabó et al., 2016; Caponnetto and De Vito, 2007. Let $\mathcal{H}$ be the RKHS of the GP kernel K with inner product $\langle \cdot, \cdot \rangle$. Let

$$
\mu_z(s) = \int K(s,t) dz(t)
$$

be the kernel mean embedding of distribution z. Denote:

$$
T_{\boldsymbol{z}} = \frac{1}{n} \sum_{i=1}^{n} \mu_{z_i} \langle \mu_{z_i}, \cdot \rangle, \qquad g_{\boldsymbol{z}} = \frac{1}{n} \sum_{i=1}^{n} y_i \mu_{z_i}, \qquad \phi_{\boldsymbol{z}} = \frac{1}{n} \sum_{i=1}^{n} \mu_{z_i} \otimes \mu_{z_i}
$$

where $(f_1 \otimes f_2)(s,t) = f_1(s) f_2(t)$ and z_i, y_i are the true density and response for

subject i. Naturally, $\mathcal{H} \otimes \mathcal{H}$ would be the closure of set $\{f_1 \otimes f_2 : f_1 \in \mathcal{H}, f_2 \in \mathcal{H}\}$, equipped with inner product as continuous extension of $\langle f_1 \otimes f_2, g_1 \otimes g_2 \rangle = \langle f_1, g_1 \rangle \langle f_2, g_2 \rangle$. Then, we know that $g_z \in \mathcal{H}$ and $T_z : \mathcal{H} \to \mathcal{H}$ is Hermitian. Similarly, one can define $T_{\hat{z}}, g_{\hat{z}}, \phi_{\hat{z}}$ by changing the true densities, z_i, to empirical densities from samples, $\{x_{ij}\}$, and they have the same properties. Following this notation, Gaussian processes has following posterior Caponnetto and De Vito, 2007:

$$f \mid \mathbb{Z}_n \sim GP\left((T_z + \sigma_n^2)^{-1} g_z,\ K - \left[(T_z + \sigma_n^2)^{-1} \otimes \mathrm{Id}\right] \phi_z\right) \tag{3.31}$$

$$f \mid \mathbb{D}_n \sim GP\left((T_{\hat{z}} + \sigma_n^2)^{-1} g_{\hat{z}},\ K - \left[(T_{\hat{z}} + \sigma_n^2)^{-1} \otimes \mathrm{Id}\right] \phi_{\hat{z}}\right) \tag{3.32}$$

where $\sigma_n^2 = \sigma^2/n$ and Id is the identity operator. The operator $T_1 \otimes T_2 : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$ is naturally defined through continuous extension of $(T_1 \otimes T_2)(f_1 \otimes f_2) = T_1(f_1) \otimes T_2(f_2)$. Denote $\hat{z}_i$ as the empirical distribution from samples x_{ij}, $f_z = (T_z + \sigma_n^2)^{-1} g_z$ and $f_{\hat{z}} = (T_{\hat{z}} + \sigma_n^2)^{-1} g_{\hat{z}}$. Also $M_z = \left[(T_z + \sigma_n^2)^{-1} \otimes \mathrm{Id}\right] \phi_z$ and $M_{\hat{z}} = \left[(T_{\hat{z}} + \sigma_n^2)^{-1} \otimes \mathrm{Id}\right] \phi_{\hat{z}}$. Therefore, $M_z, M_{\hat{z}} \in \mathcal{H} \otimes \mathcal{H}$. Now we can bound R_n^1 step by step. In the following sections, we denote K_s as the function $K_s(t) = K(s,t)$.

3.6.2.1 Step 1: Bound $V(s)$

We have $V(s) = M_z(s,s) - M_{\hat{z}}(s,s)$. It can be observed that

$$M_z(s,s) = \langle M_z, K_s \otimes K_s \rangle = \left\langle \frac{1}{n} \sum_{i=1}^{n} (T_z + \sigma_n^2)^{-1} \mu_{z_i} \langle \mu_{z_i}, K_s \rangle, K_s \right\rangle$$

$$= \left\langle (T_z + \sigma_n^2)^{-1} T_z K_s, K_s \right\rangle$$

The second equation comes from the definition of $\mathcal{H} \otimes \mathcal{H}$. Therefore,

$$
\begin{aligned}
|V(s)| &= \left| \left\langle ((T_z + \sigma_n^2)^{-1} T_z - (T_{\hat{z}} + \sigma_n^2)^{-1} T_{\hat{z}}) K_s, K_s \right\rangle \right| \\
&\leq \kappa \left\| (T_z + \sigma_n^2)^{-1} T_z - (T_{\hat{z}} + \sigma_n^2)^{-1} T_{\hat{z}} \right\|_{\mathcal{L}(\mathcal{H})} \\
&= \kappa \sigma_n^2 \left\| (T_{\hat{z}} + \sigma_n^2)^{-1} - (T_z + \sigma_n^2)^{-1} \right\|_{\mathcal{L}(\mathcal{H})} \\
&= \kappa \sigma_n^2 \left\| (T_{\hat{z}} + \sigma_n^2)^{-1} (T_z - T_{\hat{z}}) (T_z + \sigma_n^2)^{-1} \right\|_{\mathcal{L}(\mathcal{H})} \leq \frac{\kappa}{\sigma_n^2} \left\| (T_z - T_{\hat{z}}) \right\|_{\mathcal{L}(\mathcal{H})}
\end{aligned}
$$

for operator norm $\|\cdot\|_{\mathcal{L}(\mathcal{H})}$. Recall that κ is the upper bound for the kernel K. To bound $\|(T_z - T_{\hat{z}})\|_{\mathcal{L}(\mathcal{H})}$, we have:

$$
\|T_z - T_{\hat{z}}\|_{\mathcal{L}(\mathcal{H})}^2 \leq \frac{1}{n} \sum_{i=1}^{n} \left\| \mu_{z_i} \langle \mu_{z_i}, \cdot \rangle - \mu_{\hat{z}_i} \langle \mu_{\hat{z}_i}, \cdot \rangle \right\|_{\mathcal{L}(\mathcal{H})}^2
$$

And for every $\left\| \mu_{z_i} \langle \mu_{z_i}, \cdot \rangle - \mu_{\hat{z}_i} \langle \mu_{\hat{z}_i}, \cdot \rangle \right\|$, apply it to arbitrary function f.

$$
\begin{aligned}
\left\| \mu_{z_i} \langle \mu_{z_i}, f \rangle - \mu_{\hat{z}_i} \langle \mu_{\hat{z}_i}, f \rangle \right\|_{\mathcal{H}}^2 &= \left\| (\mu_{z_i} - \mu_{\hat{z}_i}) \langle \mu_{z_i}, f \rangle + \mu_{\hat{z}_i} \langle \mu_{z_i} - \mu_{\hat{z}_i}, f \rangle \right\|_{\mathcal{H}}^2 \\
&\leq \|f\|_{\mathcal{H}}^2 \left(\left(\|\mu_{z_i}\|_{\mathcal{H}}^2 + \|\mu_{\hat{z}_i}\|_{\mathcal{H}}^2 \right) \|\mu_{z_i} - \mu_{\hat{z}_i}\|_{\mathcal{H}}^2 \right) \\
&\leq 2\kappa \|f\|_{\mathcal{H}}^2 \|\mu_{z_i} - \mu_{\hat{z}_i}\|_{\mathcal{H}}^2
\end{aligned}
$$

Therefore, $\|T_z - T_{\hat{z}}\|^2_{\mathcal{L}(\mathcal{H})} \leq \frac{2\kappa}{n}\sum_i \left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|^2_{\mathcal{H}}$. And,

$$\begin{aligned}\left|\mathbb{E}\int_{[0,1]^d} V(s)ds\right| &\leq \frac{\kappa}{\sigma_n^2}\mathbb{E}\left\|(T_z - T_{\hat{z}})\right\|_{\mathcal{L}(\mathcal{H})} \leq \frac{\sqrt{2}\kappa^{3/2}}{\sigma_n^2}\sqrt{\mathbb{E}\left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|^2_{\mathcal{H}}} \\ &= \frac{\sqrt{2}\kappa^{3/2}}{\sigma_n^2}\sqrt{\mathbb{E}_{z_i}\mathbb{E}_{x_{ij}}\left\|\frac{1}{m}\sum_j K_{x_{ij}} - \mu_{z_i}\right\|^2_{\mathcal{H}}} \\ &= \frac{\sqrt{2}\kappa^{3/2}}{\sigma_n^2}\sqrt{\mathbb{E}_{z_i}\left[\frac{1}{m}\mathbb{E}_{x_{ij}}\left\|K_{x_{ij}} - \mu_{z_i}\right\|^2_{\mathcal{H}}\right]} \leq \frac{2\sqrt{2}\kappa^2}{\sigma_n^2\sqrt{m}} \end{aligned} \tag{3.33}$$

3.6.2.2 Step 2: Bound $E_1(s)$

One main difficulty for the bound here is that it is not assumed f_0 lies in the RKHS of kernel K. In fact, when the optimal rate is achieved with $\beta = \alpha$, the RKHS $\mathcal{H}_K$ contains all $(\beta + 1/2)$-regular functions, implying $f_0 \notin \mathcal{H}_K$.

In the situation where $f_0 \notin \mathcal{H}_K$, from Lemma 4 of Van Der Vaart and Van Zanten, 2011 it is known that for an order α Matérn kernel K and β-regular f_0 with $\beta \leq \alpha$ one can always find $h \in \mathcal{H}_K$ such that:

$$\inf_{\|h - f_0\|_\infty < \varepsilon} \|h\|^2_{\mathcal{H}} \leq C_\alpha \left(\frac{1}{\varepsilon}\right)^{(2\alpha - 2\beta + d)/\beta} \tag{3.34}$$

for arbitrary small ε and constant C_α depending only on α and f_0. Now, one can use Equation 3.34 to find a $f_0^\gamma \in \mathcal{H}$ such that $\left\|f_0^\gamma - f_0\right\|_\infty \leq n^{-\gamma}$ and $\left\|f_0^\gamma\right\|_{\mathcal{H}} \leq 2C_\alpha n^{(2\alpha - 2\beta + d)\gamma/2\beta}$. We determine γ at the end of the proof. Next, we have $\mathbb{E}_{\varepsilon_i} y_i = \mathbb{E}_{z_i} f_0 = \mathbb{E}_{z_i} f_0^\gamma + \mathbb{E}_{z_i}(f_0 - f_0^\gamma) = \langle \mu_{z_i}, f_0^\gamma \rangle + \mathbb{E}_{z_i}(f_0 - f_0^\gamma)$. Denote $r_i = \mathbb{E}_{z_i}(f_0 - f_0^\gamma)$,

$|r_i| \leq n^{-\gamma}$ and:

$$\mathbb{E}_\varepsilon (T_z + \sigma_n^2)^{-1} g_z = (T_z + \sigma_n^2)^{-1} \left(\frac{1}{n} \sum_i (\langle \mu_{z_i}, f_0^\gamma \rangle \mu_{z_i} + r_i \mu_{z_i}) \right)$$

$$= (T_z + \sigma_n^2)^{-1} T_z f_0^\gamma + \frac{1}{n} \sum_i r_i (T_z + \sigma_n^2)^{-1} \mu_{z_i}$$

Similarly:

$$\mathbb{E}_\varepsilon (T_{\hat{z}} + \sigma_n^2)^{-1} g_{\hat{z}} = (T_{\hat{z}} + \sigma_n^2)^{-1} \left(\frac{1}{n} \sum_i (\langle \mu_{z_i}, f_0^\gamma \rangle \mu_{\hat{z}_i} + r_i \mu_{\hat{z}_i}) \right)$$

$$= (T_{\hat{z}} + \sigma_n^2)^{-1} T_{\hat{z}} f_0^\gamma + \frac{1}{n} \sum_i (r_i + d_i) (T_{\hat{z}} + \sigma_n^2)^{-1} \mu_{\hat{z}_i}$$

where $d_i = \langle \mu_{z_i} - \mu_{\hat{z}_i}, f_0^\gamma \rangle$. Therefore,

$$|E_1(s)| = |\langle E_1, K_s \rangle| \leq \sqrt{\kappa} \, \|E_1\|_{\mathcal{H}}$$

$$\leq \sqrt{\kappa} \left\| (T_z + \sigma_n^2)^{-1} T_z - (T_{\hat{z}} + \sigma_n^2)^{-1} T_{\hat{z}} \right\|_{\mathcal{L}(\mathcal{H})} \left\| f_0^\gamma \right\|_{\mathcal{H}}$$

$$+ \sqrt{\kappa} \left\| \frac{1}{n} \sum_i d_i (T_{\hat{z}} + \sigma_n^2)^{-1} \mu_{\hat{z}_i} \right\|_{\mathcal{H}} + \sqrt{\kappa} \left\| \frac{1}{n} \sum_i r_i (T_z + \sigma_n^2)^{-1} (\mu_{z_i} - \mu_{\hat{z}_i}) \right\|_{\mathcal{H}}$$

$$+ \sqrt{\kappa} \left\| \frac{1}{n} \sum_i r_i ((T_z + \sigma_n^2)^{-1} - (T_{\hat{z}} + \sigma_n^2)^{-1}) \mu_{\hat{z}_i} \right\|_{\mathcal{H}}$$

$$\leq \left(\frac{2 C_\alpha \sqrt{\kappa}}{\sigma_n^2} \left\| T_z - T_{\hat{z}} \right\|_{\mathcal{L}(\mathcal{H})} + \frac{2 C_\alpha \kappa}{\sigma_n^2 n} \sum_i \left\| \mu_{z_i} - \mu_{\hat{z}_i} \right\|_{\mathcal{H}} \right) n^{\frac{(2\alpha - 2\beta + d)\gamma}{2\beta}}$$

$$+ \left(\frac{\sqrt{\kappa}}{\sigma_n^2 n} \sum_i \left\| \mu_{z_i} - \mu_{\hat{z}_i} \right\|_{\mathcal{H}} + \frac{\kappa}{\sigma_n^4} \left\| T_z - T_{\hat{z}} \right\|_{\mathcal{L}(\mathcal{H})} \right) n^{-\gamma}$$

Using the bounds derived in the previous step, it follows that:

$$\left|\mathbb{E}\int_{[0,1]^d} 2f_0(s)E_1(s)ds\right| \leq \frac{16C_0C_\alpha\kappa^{3/2}}{\sigma_n^2\sqrt{m}}n^{\frac{(2\alpha-2\beta+d)\gamma}{2\beta}} + \frac{4C_0\kappa}{\sigma_n^2\sqrt{m}}n^{-\gamma} + \frac{4\sqrt{2}\kappa^2}{\sigma_n^4\sqrt{m}}n^{-\gamma}$$

where C_0 is the upper bound of $|f_0|$, which must exists, since it is assumed f_0 is regular (hence continuous) in a compact set. The rate can be improved when $f_0 \in \mathcal{H}_K$ ($\beta \geq \alpha + 1/2$), in which case f_0^γ can be f_0, making $r_i = 0$ and $\left\|f_0^\gamma\right\|$ constant. Finally:

$$\left|\mathbb{E}\int_{[0,1]^d} 2f_0(s)E_1(s)ds\right| \leq \frac{16C_0C_\alpha\kappa^{3/2}}{\sigma_n^2\sqrt{m}}$$

3.6.2.3 Step 3: Bound $E_2(s)$

Using the same notation as in Step 2, in the situation $f_0 \notin \mathcal{H}_K$, denote:

$$A = (T_z + \sigma_n^2)^{-1}T_z f_0^\gamma, \quad B = \frac{1}{n}\sum_i \varepsilon_i(T_z + \sigma_n^2)^{-1}\mu_{z_i}, \quad C = \frac{1}{n}\sum_i r_i(T_z + \sigma_n^2)^{-1}\mu_{z_i}$$

$$\hat{A} = (T_{\hat{z}} + \sigma_n^2)^{-1}T_{\hat{z}} f_0^\gamma, \quad \hat{B} = \frac{1}{n}\sum_i \varepsilon_i(T_{\hat{z}} + \sigma_n^2)^{-1}\mu_{\hat{z}_i}, \quad \hat{C} = \frac{1}{n}\sum_i (r_i + d_i)(T_{\hat{z}} + \sigma_n^2)^{-1}\mu_{\hat{z}_i}$$

Then we have $E_2 = \mathbb{E}_\varepsilon\left[(A+B+C)^2 - (\hat{A}+\hat{B}+\hat{C})^2\right]$. It can be seen that the cross term $\mathbb{E}_\varepsilon\left[B(A+C) + \hat{B}(\hat{A}+\hat{C})\right] = 0$. Because ε_i is mean 0 and independent of any z_i and x_{ij}, therefore,

$$E_2 = (A + C + \hat{A} + \hat{C})(A - \hat{A} + C - \hat{C}) + \mathbb{E}_\varepsilon\left[B^2 - \hat{B}^2\right]$$

For the first part

$$|(A+C+\hat{A}+\hat{C})(A-\hat{A}+C-\hat{C})|(s) = |\langle A+C+\hat{A}+\hat{C}, K_s\rangle\langle A-\hat{A}+C-\hat{C}, K_s\rangle|$$

$$\leq \kappa \left(\|A\|_{\mathcal{H}} + \left\|\hat{A}\right\|_{\mathcal{H}} + \|C\|_{\mathcal{H}} + \left\|\hat{C}\right\|_{\mathcal{H}}\right) \|E_1\|_{\mathcal{H}}$$

$$\leq \kappa \|E_1\|_{\mathcal{H}} \left(2\left\|f_0^{\gamma}\right\|_{\mathcal{H}} + \frac{2\sqrt{\kappa}}{\sigma_n^2} n^{-\gamma} + \frac{\sqrt{\kappa}\left\|f_0^{\gamma}\right\|_{\mathcal{H}}}{\sigma_n^2 n} \sum_i \left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|_{\mathcal{H}}\right)$$

One can use the same bound for E_1 and notice that:

$$\mathbb{E}\left[\frac{\|T_{\mathbf{z}} - T_{\hat{\mathbf{z}}}\|_{\mathcal{L}(\mathcal{H})}}{n} \sum_i \left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|_{\mathcal{H}}\right] \leq \sqrt{\mathbb{E}\,\|T_{\mathbf{z}} - T_{\hat{\mathbf{z}}}\|^2_{\mathcal{L}(\mathcal{H})}\,\mathbb{E}\left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|^2_{\mathcal{H}}} = O\left(\frac{1}{m}\right)$$

$$\mathbb{E}\left[\left(\frac{1}{n}\sum_i \left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|_{\mathcal{H}}\right)\left(\frac{1}{n}\sum_i \left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|_{\mathcal{H}}\right)\right] \leq \sqrt{\mathbb{E}\left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|^2_{\mathcal{H}}\,\mathbb{E}\left\|\mu_{z_i} - \mu_{\hat{z}_i}\right\|^2_{\mathcal{H}}} = O\left(\frac{1}{m}\right)$$

For the second part, we have (notice that $(T_{\mathbf{z}} + \sigma_n^2)^{-1}$ is Hermitian):

$$\mathbb{E}_{\boldsymbol{\varepsilon}}[B^2](s) = \frac{1}{n^2}\sum_i \sigma^2 \langle (T_{\mathbf{z}} + \sigma_n^2)^{-1}\mu_{z_i}, K_s\rangle\langle \mu_{z_i}, (T_{\mathbf{z}} + \sigma_n^2)^{-1}K_s\rangle$$

$$= \frac{\sigma^2}{n}\langle (T_{\mathbf{z}} + \sigma_n^2)^{-2}T_{\mathbf{z}}K_s, K_s\rangle$$

and

$$|\mathbb{E}_{\boldsymbol{\varepsilon}}[B^2 - \hat{B}^2](s)| \leq \frac{\sigma^2\kappa}{n}\left\|(T_{\mathbf{z}} + \sigma_n^2)^{-2}T_{\mathbf{z}} - (T_{\hat{\mathbf{z}}} + \sigma_n^2)^{-2}T_{\hat{\mathbf{z}}}\right\|_{\mathcal{L}(\mathcal{H})}$$

$$\leq \kappa\sigma_n^2\left\|((T_{\mathbf{z}} + \sigma_n^2)^{-1} - (T_{\hat{\mathbf{z}}} + \sigma_n^2)^{-1})(T_{\mathbf{z}} + \sigma_n^2)^{-1}T_{\mathbf{z}}\right\|_{\mathcal{L}(\mathcal{H})}$$

$$+ \kappa\sigma_n^2\left\|(T_{\mathbf{z}} + \sigma_n^2)^{-1}((T_{\mathbf{z}} + \sigma_n^2)^{-1}T_{\mathbf{z}} - (T_{\hat{\mathbf{z}}} + \sigma_n^2)^{-1}T_{\hat{\mathbf{z}}})\right\|_{\mathcal{L}(\mathcal{H})}$$

$$\leq \frac{2\kappa}{\sigma_n^2}\|T_{\mathbf{z}} - T_{\hat{\mathbf{z}}}\|_{\mathcal{L}(\mathcal{H})}$$

Therefore, using the bound in Equation 3.33 and ignoring the constant term, we have (recall that $\left\|f_0^{\gamma}\right\|_{\mathcal{H}} = O(n^{(2\alpha-2\beta+d)\gamma/2\beta})$):

$$\left|\mathbb{E}\int_{[0,1]^d} E_2(s)ds\right| \leq C(\alpha,\kappa,f_0)\left(\frac{1}{\sigma_n^2\sqrt{m}}n^{\frac{(2\alpha-2\beta+d)\gamma}{\beta}} + \frac{1}{\sigma_n^4\sqrt{m}}n^{-\gamma+\frac{(2\alpha-2\beta+d)\gamma}{2\beta}}\right. \tag{3.35}$$

$$\left. + \frac{1}{\sigma_n^6\sqrt{m}}n^{-2\gamma} + \frac{1}{\sigma_n^4 m}n^{\frac{(2\alpha-2\beta+d)\gamma}{\beta}} + \frac{1}{\sigma_n^6 m}n^{-\gamma+\frac{(2\alpha-2\beta+d)\gamma}{2\beta}}\right) \tag{3.36}$$

for some constant $C(\alpha,\kappa,f_0)$ depending only on α,κ,f_0.

Similarly, the rate can be improved if $f_0 \in \mathcal{H}_K$, where f_0^{γ} can directly be f_0, making $r_i = 0$ and $\left\|f_0^{\gamma}\right\|$ constant. Thus:

$$\left|\mathbb{E}\int_{[0,1]^d} E_2(s)ds\right| \leq \frac{C(\kappa,f_0)}{\sigma_n^2\sqrt{m}} \tag{3.37}$$

given $\sigma_n^2\sqrt{m} \to \infty$. Notice again that $\sigma_n^2 = \sigma^2/n$.

3.6.2.4 Combine Together

Combining all 3 steps above it follows that:

$$R_n^1 = O\left(\frac{n}{\sqrt{m}}\right) \tag{3.38}$$

if $f_0 \in \mathcal{H}_K$ ($\beta \geq \alpha + 1/2$). If $f_0 \notin \mathcal{H}_K$, one gets a more complex, and worse, rate as:

$$R_n^1 = O\left(\frac{n^{1+\frac{(2\alpha-2\beta+d)\gamma}{2\beta}}}{\sqrt{m}} + \frac{n^{2-\gamma+\frac{(2\alpha-2\beta+d)\gamma}{2\beta}}}{\sqrt{m}} + \frac{n^{3-2\gamma}}{\sqrt{m}}\right) \tag{3.39}$$

given $n/\sqrt{m} \to 0$. Therefore, when $\alpha = \beta$, setting $\gamma = 1$ we get that $R_n^1 = O(n^{-2\beta/(2\beta+d)})$ if $m = \Omega\left(n^{2+\frac{d}{\beta}+\frac{4\beta}{2\beta+d}}\right)$.

3.6.3 Bound R_n^1 for empirical norm $\|\cdot\|_n$

Denote $\hat{\mathcal{Z}}$ as the empirical distributional process support only on n points of z_i. We have:

$$
\begin{aligned}
R_n^1 &= \mathbb{E}_{f_0} \int \|f - f_0\|_n^2 \left(d\Pi_n(f|\mathbb{D}_n) - d\Pi_n(f|\mathbb{Z}_n)\right) \\
&= \mathbb{E}_{f_0} \int \mathbb{E}_{z\sim\hat{\mathcal{Z}}}[\mathbb{E}_z(f - f_0)\mathbb{E}_z(f - f_0)](d\Pi_n(f|\mathbb{D}_n) - d\Pi_n(f|\mathbb{Z}_n)) \\
&= \mathbb{E}_{f_0}\mathbb{E}_{z\sim\hat{\mathcal{Z}}} \int \left[\int (f - f_0)(s)(f - f_0)(t)dz(s)dz(t)\right] (d\Pi_n(f|\mathbb{D}_n) - d\Pi_n(f|\mathbb{Z}_n)) \\
&= \mathbb{E}_{f_0}\mathbb{E}_{z\sim\hat{\mathcal{Z}}} \int (\mathbb{E}_{f|\mathbb{D}_n} - \mathbb{E}_{f|\mathbb{Z}_n})[f(s)f(t)] - 2f_0(s)(\mathbb{E}_{f|\mathbb{D}_n} - \mathbb{E}_{f|\mathbb{Z}_n})[f(t)]\, dz(s)dz(t)
\end{aligned}
$$

It can then be seen that one can use exactly the same method as in Section 3.6.2 to get the same bound as for term $|(\mathbb{E}_{f|\mathbb{D}_n} - \mathbb{E}_{f|\mathbb{Z}_n})[f(s)f(t)] - 2f_0(s)(\mathbb{E}_{f|\mathbb{D}_n} - \mathbb{E}_{f|\mathbb{Z}_n})[f(t)]|$ with every s, t pair and therefore we can get the same overall bounds as for $\|\cdot\|_2$.

3.6.4 Bound R_n^0

We use the method described in Van Der Vaart and Van Zanten, 2011 to bound R_n^0, by extending it to distributional covariates settings. To do that we need to rewrite the model 3.28-3.29 into a typical Gaussian process regression in metric space. Consider:

$$
y_i = F_0(z_i) + \varepsilon_i \tag{3.40}
$$

where z_i is the individual distribution for subject i and F_0 is an element in the linear space $\mathcal{B}$. If there exists an bijection π from $\mathcal{B}$ to $\mathcal{C}_0$: the bounded continuous function on $[0,1]^d$ and $\forall F \in \mathcal{B}, z \in \mathcal{S}: \; F(z) = \mathbb{E}_z \pi(F)$. Naturally we give $\mathcal{B}$

a norm that $\|F\|_{\mathcal{B}} = \|\pi(F)\|_\infty$ and it is clear that $|F(z)| \leq \|F\|_{\mathcal{B}}$ for all z. Note. when it is not misleading, we will also denote $\pi(F)$ simply as f. Similarly, f_0 for $\pi(F_0)$ and any other super-sub-script.

It is clear that if $\mathcal{Z}$ **weakly separates** $\mathcal{C}_0$, π would be an isomorphism between $\mathcal{B}$ and $\mathcal{C}_0$. Because π would be a bijection with $\|F\|_{\mathcal{B}} = \|\pi(F)\|_\infty$. Hence, $\mathcal{B}$ is a separable Banach space.

Now consider a kernel $\mathbb{K}$ on set $\mathcal{S}$ such that $\mathbb{K}(z_1, z_2) = \langle \mu_{z_1}, \mu_{z_2} \rangle_{\mathcal{H}}$ where $\mu_z = \int K(\cdot, s) dz(s)$ is the kernel mean embedding of z. Clearly $\mathbb{K}$ satisfies the kernel property. Denote the RKHS of $\mathbb{K}$ to be $\mathcal{H}_{\mathbb{K}}$. Then, $\mathcal{H}_{\mathbb{K}}$ is a subspace of $\mathcal{B}$ when using a Matérn kernel for K. From Lemma 5, we also have $\pi(\mathcal{H}_{\mathbb{K}}) = \mathcal{H}$, with $\|h\|_{\mathcal{H}_{\mathbb{K}}} = \|\pi(h)\|_{\mathcal{H}}$.

Lemma 5. *Using the kernel $\mathbb{K}$ and projection π defined above, and $\mathcal{H}_{\mathbb{K}}$ be the RKHS from $\mathbb{K}$, we have $\pi(\mathcal{H}_{\mathbb{K}}) = \mathcal{H}$ with $\|h\|_{\mathcal{H}_{\mathbb{K}}} = \|\pi(h)\|_{\mathcal{H}}$ if $\mathcal{Z}$ weakly separates $\mathcal{C}_0$.*

Proof. First, $\mathcal{H}_{\mathbb{K}}$ is the Hilbert space spanned by $\mathbb{K}(z, \cdot) = \langle \mu_z, \mu_{(\cdot)} \rangle_{\mathcal{H}}$ and it is easy to see $\pi(\mathbb{K}(z, \cdot)) = \mu_z = \int K(s, \cdot) dz(s)$. Because $\langle \mu_w, g \rangle_{\mathcal{H}} = \mathbb{E}_w f$ for any distribution w and function $g \in \mathcal{H}$. Also $\|\mathbb{K}(z, \cdot)\|_{\mathcal{H}_{\mathbb{K}}} = \mathbb{K}(z, z) = \int K(s, t) dz \otimes dz = \|\pi(\mathbb{K}(z, \cdot))\|_{\mathcal{H}}$. Since π is an isomorphism between $\mathcal{B}$ and $\mathcal{C}_0$. We know that $\pi(\mathcal{H}_{\mathbb{K}})$ is a subspace in $\mathcal{H}$ spanned by $\{\mu_z : z \in \mathcal{S}\}$ with $\|\pi(h)\|_{\mathcal{H}} = \|h\|_{\mathcal{H}_{\mathbb{K}}}$.

Now decompose $\mathcal{H} = \pi(\mathcal{H}_{\mathbb{K}}) \oplus \pi(\mathcal{H}_{\mathbb{K}})^\perp$, for any $f \in \pi(\mathcal{H}_{\mathbb{K}})^\perp$ we have:

$$\forall z \in \mathcal{S} : \langle \mu_z, f \rangle_{\mathcal{H}} = \mathbb{E}_z f = 0$$

but by our assumption on the richness of $\mathcal{S}$, this can happen only when $f = 0$. Therefore $\pi(\mathcal{H}_{\mathbb{K}})^\perp = \mathbf{0}$ hence $\mathcal{H} = \pi(\mathcal{H}_{\mathbb{K}})$.

□

We now show that our model 3.28-3.29, with known individual distributions, is equivalent to Gaussian process regression:

$$F \sim GP(0, \mathbb{K}) \tag{3.41}$$

$$y_i \sim \mathcal{N}(F(z_i), \sigma^2) \tag{3.42}$$

We show that $\pi^{-1}(f)|\mathbb{Z}_n$ is the same distribution as $F|\mathbb{Z}_n$. Consider that $f \sim GP(0, K) \Rightarrow \pi^{-1}(f) \sim GP(0, \mathbb{K})$. Furthermore, by definition, $\mathbb{E}_{z_i} f = \pi^{-1}(f)(z_i)$ for all z_i. Therefore, from the uniqueness of Gaussian processes it follows that $\pi^{-1}(f)|\mathbb{Z}_n \sim F|\mathbb{Z}_n$.

Now, define the square operator, $\mathcal{B} \to \mathcal{B}$, as $F^2 = \pi^{-1}(\pi(F)^2)$, where square in $\mathcal{C}_0$ is the typical point-wise square. Since the square of bounded continuous functions is still bounded and continuous, this operator in $\mathcal{B}$ is well defined. Obviously, $F^2(z) = \mathbb{E}_z \pi(F)^2$. Define the L_2 norm in $\mathcal{B}$ as $\|F\|_2^2 = F^2(\mu)$, where μ is the mean measure of z_i (assumed to be Unif(0, 1)). It is clear that $\|F\|_2 \leq \|F\|_{\mathcal{B}}$. Similarly define the empirical norm in $\mathcal{B}$ as $\|F\|_n^2 = \frac{1}{n}\sum_i^n (F(z_i))^2$. It can be shown that the risk, R_n^0, agrees for both models under $\|\cdot\|_2$ and $\|\cdot\|_n$. For example, for $\|\cdot\|_2$:

$$R_n^0 = \mathbb{E}_{f_0} \int \|f - f_0\|_2^2 \, d\Pi_n(f|\mathbb{Z}_n) = \mathbb{E}_{f_0} \int \mathbb{E}_\mu (f - f_0)^2 d\Pi_n(f|\mathbb{Z}_n)$$

$$= \mathbb{E}_{f_0} \int \left\| \pi^{-1}(f) - F_0 \right\|_2^2 d\Pi_n(f|\mathbb{Z}_n) = \mathbb{E}_{f_0} \int \|F - F_0\|_2^2 \, d\Pi_n(F|\mathbb{Z}_n)$$

Using Theorem 1 in Van Der Vaart and Van Zanten, 2011 we can bound the risk term R_n^0 in the fixed design setting with $\Psi_{F_0}^{-1}(n)^2$, where $F \sim GP(0, \mathbb{K})$ and:

$$\varepsilon^2 \Psi_{F_0}(\varepsilon) = \inf_{h \in \mathcal{H}_{\mathbb{K}} : ||h - F_0||_{\mathcal{B}} < \varepsilon} ||h||_{\mathcal{H}_{\mathbb{K}}}^2 - \log \mathbb{P}(||F||_{\mathcal{B}} < \varepsilon) \tag{3.43}$$

Recall the definition of $\mathcal{H}_{\mathbb{K}}$ and $\mathcal{B}$, and consider $f \sim GP(0, K)$ we have:

$$\mathbb{P}(\|f\|_\infty \leq \varepsilon) = \mathbb{P}\left(\left\|\pi^{-1}(f)\right\|_{\mathcal{B}} \leq \varepsilon\right) = \mathbb{P}(||F||_{\mathcal{B}} < \varepsilon)$$

$$\inf_{h\in\mathcal{H}:||h-f_0||_\infty<\varepsilon} ||h||^2_{\mathcal{H}} = \inf_{h\in\mathcal{H}:||\pi^{-1}(h)-F_0||_{\mathcal{B}}<\varepsilon} ||\pi^{-1}(h)||^2_{\mathcal{H}_{\mathbb{K}}} = \inf_{h\in\mathcal{H}_{\mathbb{K}}:||h-F_0||_{\mathcal{B}}<\varepsilon} ||h||^2_{\mathcal{H}_{\mathbb{K}}}$$

Therefore, $\Psi_{F_0}(\varepsilon) = \Psi_{f_0}(\varepsilon)$ with:

$$\varepsilon^2\Psi_{f_0}(\varepsilon) = \inf_{h\in\mathcal{H}:||h-f_0||_\infty<\varepsilon} ||h||^2_{\mathcal{H}} - \log \mathbb{P}(||f||_\infty < \varepsilon) \tag{3.44}$$

which implies one obtains the same rates as in Theorem 5 of Van Der Vaart and Van Zanten, 2011, which is $n^{-2\min(\alpha,\beta)/(2\alpha+d)}$.

Strong separation is needed for the estimation bound because the empirical norm, $\|f\|_n^2 = \frac{1}{n}\sum_i(\mathbb{E}_{z_i}f)^2$, converges to $\mathbb{E}_{z\sim\mathcal{Z}}[(\mathbb{E}_z f)^2]$, not to $\|f\|_2^2$, introducing a gap between the empirical bound and estimation bound. However, if $\mathcal{Z}$ strongly separates $\pi(\mathcal{B})$ with constant C, then $\|f\|_2^2$ becomes equivalent with $\mathbb{E}_{z\sim\mathcal{Z}}[(\mathbb{E}_z f)^2]$. Furthermore, the proof for Theorem 2 in Van Der Vaart and Van Zanten, 2011 can be continued by observing:

$$P(\|f - f_\varepsilon\|_2 \geq 2C\|f - f_\varepsilon\|_n) \leq P(\|f - f_\varepsilon\|_{\mathcal{Z}} \geq 2\|f - f_\varepsilon\|_n)$$

$$\leq e^{-(n/5)\|f-f_\varepsilon\|^2_{\mathcal{Z}}/\|f-f_\varepsilon\|^2_\infty} \leq e^{-(n/5C^2)\|f-f_\varepsilon\|^2_2/\|f-f_\varepsilon\|^2_\infty}$$

for any f and f_ε. In such case one obtains the same rate as $n^{-2\min(\alpha,\beta)/(2\alpha+d)}$, given $\min(\alpha, \beta) > d/2$.

References

Szabó, Zoltán, Bharath K Sriperumbudur, Barnabás Póczos, and Arthur Gretton (2016). "Learning theory for distribution regression". In: *The Journal of Machine Learning Research* 17.1, pp. 5272–5311.

Oliva, Junier, Barnabás Póczos, and Jeff Schneider (2013). "Distribution to distribution regression". In: *International Conference on Machine Learning*. PMLR, pp. 1049–1057.

Fang, Zhiying, Zheng-Chu Guo, and Ding-Xuan Zhou (2020). "Optimal learning rates for distribution regression". In: *Journal of complexity* 56, p. 101426.

Law, Ho Chung Leon, Danica J Sutherland, Dino Sejdinovic, and Seth Flaxman (2018). "Bayesian approaches to distribution regression". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1167–1176.

Póczos, Barnabás, Aarti Singh, Alessandro Rinaldo, and Larry Wasserman (2013). "Distribution-free distribution regression". In: *Artificial Intelligence and Statistics*. PMLR, pp. 507–515.

Oliva, Junier, Willie Neiswanger, Barnabás Póczos, Jeff Schneider, and Eric Xing (2014). "Fast distribution to real regression". In: *Artificial Intelligence and Statistics*. PMLR, pp. 706–714.

Vaart, Aad W van der and J Harry van Zanten (2008). "Rates of contraction of posterior distributions based on Gaussian process priors". In: *The Annals of Statistics* 36.3, pp. 1435–1463.

Van Der Vaart, Aad and Harry Van Zanten (2011). "Information Rates of Nonparametric Gaussian Process Methods." In: *Journal of Machine Learning Research* 12.6.

Sniekers, Suzanne and Aad van der Vaart (2015). "Adaptive Bayesian credible sets in regression with a Gaussian process prior". In: *Electronic Journal of Statistics* 9.2, pp. 2475–2527.

Choi, Taeryon (2007). "Alternative posterior consistency results in nonparametric binary regression using Gaussian process priors". In: *Journal of statistical planning and inference* 137.9, pp. 2975–2983.

Yuan, Ming and T Tony Cai (2010). "A reproducing kernel Hilbert space approach to functional linear regression". In: *The Annals of Statistics* 38.6, pp. 3412–3444.

Augustin, Nicole H, Calum Mattocks, Julian J Faraway, Sonja Greven, and Andy R Ness (2017). "Modelling a response as a function of high-frequency count data: the association between physical activity and fat mass". In: *Statistical methods in medical research* 26.5, pp. 2210–2226.

Zaheer, Manzil, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola (2017). "Deep sets". In: *Advances in neural information processing systems* 30.

Wood, Simon N (2003). "Thin plate regression splines". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.1, pp. 95–114.

Caponnetto, Andrea and Ernesto De Vito (2007). "Optimal rates for the regularized least-squares algorithm". In: *Foundations of Computational Mathematics* 7.3, pp. 331–368.

Chapter 4

Accelerating fixed-point algorithms in statistics and data science: A state-of-art review

Bohao Tang [1], *Nicholas Henderson*[2], *Ravi Varadhan*[3]

4.1 Introduction

Computational problems in science and mathematics are often solved using iterative algorithms, which produce a sequence of real-valued vectors converging to the solution of interest. Examples include solving systems of linear and nonlinear equations, numerical solutions of differential equations, approximation of integrals, and minimization of multivariate functions. Parameter estimation in many practical problems in statistics and data science can be ultimately reduced to a specific optimization problem often involving parameter constraints. To solve such optimization problems, various iterative algorithms have been developed

[1] Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health
[2] Department of Biostatistics, University of Michigan
[3] Quantitative Sciences Division, Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University

including the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977), the majorization-minimization (MM) algorithm (Hunter and Lange, 2004), and gradient based methods like gradient descent (GD) and proximal gradient descent (Boyd, Boyd, and Vandenberghe, 2004). These methods are general and easy to use, and they can all be regarded as fixed-point iteration algorithms. A major appeal of these algorithms is their stability and their ability to readily handle high-dimensional problems which is a main reason for their surge in popularity for modern applications. However, a characteristic weakness of these algorithms is their potential slow convergence, i.e., the vector sequence produced by the fixed-point iterative algorithm, $x_{n+1} = F(x_n)$, may converge very slowly (if it converges) to the solution x^*, severely limiting their effective use in solving real problems. Hence, it is desirable to have tools available that can accelerate the convergence of the sequence $\{x_n\}$. Please refer to Supplementary Material for a general, theoretical discussion of the rate of convergence of fixed-point iterations.

As highlighted in Varadhan and Roland, 2008, an acceleration scheme should possess certain key properties in order to be an effective and practical tool for high-dimensional optimization problems. It should accelerate the convergence of the original iterative algorithm (fast local convergence); it should converge to the solution from any reasonable starting value (robust global convergence), provided, of course, that the base algorithm itself is convergent; it should have minimal storage/memory requirements (applicability to high-dimensional problems); and it should require minimal problem-specific tuning (off-the-shelf usability). The minimal storage requirement eliminates Newton-type algorithms which make use of full second-order information. In this paper, we examine several recently

developed acceleration schemes that satisfy these listed requirements. The acceleration schemes discussed include SQUAREM (Varadhan and Roland, 2008), Anderson acceleration and DAAREM (Henderson and Varadhan, 2019), Quasi-Newton (Zhou, Alexander, and Lange, 2011), Nesterov acceleration with restarts (O'donoghue and Candes, 2015), and Parabolic-EM (Berlinet and Roland, 2009).

The paper is organized as follows. First, in the next section, we describe several well-known fixed-point iterations and discuss their theoretical convergence properties. In Section 4.3, we introduce and describe the accelerating methods to be studied. In Section 4.5, we test the performance of each of these acceleration methods in a range of practical problems. In Section 4.4, implementation details and available R packages are described, and in last Section, we discuss the results and give strategies for choosing an acceleration scheme for a given problem at hand.

4.2 Popular iterative algorithms and their convergence

4.2.1 MM algorithm

The MM in the MM algorithm stands for "Majorization-Minimization" or "Minorization-Maximization", depending on whether the particular optimization problem is a minimization or maximization problem. The MM algorithm actually describes a family of algorithms that are implemented by creating a surrogate function that majorizes (minorizes) the objective function of interest and optimizing this surrogate function in each iteration. A key feature of MM algorithms is that the objective function will increase (decrease)in every iteration. See Hunter and Lange, 2004 for a general description of MM algorithms.

A function $g(x|x_k)$ is called a minorized version of the objective function f at x_k if it satisfies the following two conditions

$$\forall x : \; g(x|x_k) \leq f(x)$$

$$g(x_k|x_k) = f(x_k).$$

Similarly, $g(x|x_k)$ will be called a majorized version of f at x_k if $-g(x|x_k)$ is a minorized version of $-f$. An MM maximization algorithm updates the current iterate x_k by maximizing the minorizing function $g(x|x_k)$. If we define F to be the argmax operator for $g(x|x_k)$, then we can express the MM iteration as

$$x_{k+1} = \operatorname*{argmax}_x g(x|x_k) =: F(x_k). \tag{4.1}$$

The fixed-point iteration (4.1) generates a sequence which is monotone with respect to the objective function f; that is, we are guaranteed to have $f(x_0) \leq f(x_1) \leq f(x_2) \leq ....$ This is due to the fact that $f(x_{k+1}) \geq g(x_{k+1}|x_k) \geq g(x_k|x_k) = f(x_k)$, and hence, one will get a strict increase in the objective function whenever $g(x_{k+1}|x_k) \neq g(x_k|x_k)$.

If x^* denotes an optimal point of f, then for x_k close to x^*, we have the following local approximation

$$x_{k+1} - x^* \approx dF(x^*)(x_k - x^*),$$

where $dF(x^*)$ is the Jacobian of F at x^*. It can be shown that $dF(x^*)$ is given by

$$dF(x^*) = I - [d^2 g(x^*|x^*)]^{-1} d^2 f(x^*), \tag{4.2}$$

where $d^2 f(x^*)$ and $d^2 g(x^*|x^*)$ denote the Hessian matrices of $f(x)$ and $g(x|x)$ respectively (with the derivatives in $d^2 g(x^*|x^*)$ being taken with respect to the

first argument of $g(x|x)$). Therefore, an MM algorithm has linear convergence with a rate related to the largest eigenvalue of the Jacobian in (4.2), and the value of this Jacobian depends on both the objective function and choice of surrogate function. Globally, if the objective function f is strictly convex or concave, an MM algorithm will converge to the unique optimal point, assuming it exists. Otherwise, the MM algorithm will converge to one of the stationary points.

4.2.1.1 The EM algorithm as a special case of MM

EM algorithms are used to find the value of a parameter vector x which maximizes a log-likelihood function $\ell(x) = \log p(\mathbf{Y}|x)$ of interest, where $\mathbf{Y}$ denotes the observed data vector and $p(\cdot|x)$ is the probability distribution for the observed data that is parameterized by x. To develop an EM algorithm for maximizing $\ell(x)$, one introduces a vector of unobserved latent data $\mathbf{U}$ and a probability distribution $p(\mathbf{Y}, \mathbf{U}|x)$ for $(\mathbf{Y}, \mathbf{U})$ which is also parameterized by x. Because $\ell(x)$ can be decomposed as $\log p(\mathbf{Y}|x) = \log p(\mathbf{Y}, \mathbf{U}|x) - \log p(\mathbf{U}|\mathbf{Y}, x)$ and $\log p(\mathbf{Y}|x)$ does not depend on $\mathbf{U}$, if we take the expectation of $\log p(\mathbf{Y}|x)$ with respect to the conditional distribution $[\mathbf{U}|\mathbf{Y}, x_k]$ where x_k is the current iterate of the EM algorithm, we obtain

$$\begin{aligned} \log p(\mathbf{Y}|x) &= \mathbb{E}_{\mathbf{U}|\mathbf{Y},x_k}\{\log p(\mathbf{Y}, \mathbf{U}|x)\} - \mathbb{E}_{\mathbf{U}|\mathbf{Y},x_k}\{\log p(\mathbf{U}|\mathbf{Y}, x)\} \\ &= Q(x|x_k) + H(x|x_k). \end{aligned} \tag{4.3}$$

In (4.3), $Q(x|x_k)$ is often referred to as the "Q-function", and computing it is referred to as the "E-step" of the EM algorithm. The term $H(x|x_k)$ is the cross entropy of the conditional distribution $[\mathbf{U}|\mathbf{Y}, x]$ relative to the conditional distribution

$[\mathbf{U}|\mathbf{Y}, x_k]$.

After completing the "E-step", x_{k+1} is found by maximizing the Q-function $Q(x|x_k)$ with respect to x, namely,

$$x_{k+1} = \arg\max_x Q(x|x_k) =: F(x_k).$$

Computing x_{k+1} by maximizing $Q(x|x_k)$ is usually referred to as the "M-step" of an EM algorithm.

To see why the EM algorithm is a special case of the MM algorithm, note first that it directly follows from Jensen's inequality that

$$\begin{aligned} H(x|x_k) - H(x_k|x_k) &= \mathbb{E}_{\mathbf{U}|\mathbf{Y},x_k}\Big[\log\{p(\mathbf{U}|\mathbf{Y},\theta)/p(\mathbf{U}|\mathbf{Y},x_k)\}\Big] \\ &\leq \log\Big[\mathbb{E}_{\mathbf{U}|\mathbf{Y},x_k}\{p(\mathbf{U}|\mathbf{Y},\theta)/p(\mathbf{U}|\mathbf{Y},x_k)\}\Big] \\ &= 0. \end{aligned}$$

and hence $Q(x|x_k) + H(x_k|x_k) \leq \log p(\mathbf{Y}|x)$ for any value of x. In other words, $Q(x|x_k) + H(x_k|x_k)$ is a minorized version of the log-likelihood $\log p(\mathbf{Y}|x)$. Since $H(x_k|x_k)$ is a positive constant that does not depend on x, maximizing $Q(x|x_k) + H(x_k|x_k)$ is equivalent to maximizing the Q-function. Hence, we can regard the EM algorithm as an MM algorithm with minorization function $Q(x|x_k) + H(x_k|x_k)$.

4.2.2 Gradient based algorithms

4.2.2.1 Gradient descent

Consider the following optimization problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{4.4}$$

for a smooth function f that has all first order derivatives with $\nabla f(\mathbf{x}) = (\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_p})$ denoting the gradient of f at $\mathbf{x}$. Gradient descent is an iterative algorithm that always updates the current iterate $\mathbf{x}_k$ linearly in the direction where f decrease the fastest, namely, the negative gradient $-\nabla f(\mathbf{x}_k)$. In particular, for a given choice of step size or learning rate t_k, the gradient descent update of $\mathbf{x}_k$ is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k \nabla f(\mathbf{x}_k). \tag{4.5}$$

Gradient descent may also be interpreted in the following way. At each step, we do not directly minimize the original function f, but instead, we minimize its first order approximation $f_k(\mathbf{x})$ around $\mathbf{x}_k$ which is given by

$$f_k(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2t_k}||\mathbf{x} - \mathbf{x}_k||^2, \tag{4.6}$$

where $||\cdot||$ is the Euclidean norm. One can directly check that the minimizer of the function f_k is equal to $\mathbf{x}_{k+1}$ in (4.5).

4.2.2.2 Proximal gradient descent

Optimization problem (4.4) is not general enough to handle optimization problems that have non-smooth terms. In such cases, one might consider the following generalization of (4.4)

$$\min_{\mathbf{x}} f(\mathbf{x}) + h(\mathbf{x}), \tag{4.7}$$

where, again, f is assumed to be smooth up to first order but h is instead a non-smooth function. As an example of (4.7), the objective function used in LASSO regression (Tibshirani, 1996) can be expressed as a sum of a smooth function and the non-smooth L_1 norm term.

Using the same reasoning used to obtain approximation (4.6), we can approximate the target $f(\mathbf{x}) + h(\mathbf{x})$ at each step by

$$(f+h)_k(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2t_k}||\mathbf{x}-\mathbf{x}_k||^2 + h(\mathbf{x})$$

$$= \frac{1}{2t_k}||\mathbf{x}-\mathbf{x}_k + t_k\nabla f(\mathbf{x}_k)||^2 + h(\mathbf{x}) + Const, \tag{4.8}$$

where *Const* is a constant term that does not depend on $\mathbf{x}$. In step k, the proximal gradient descent update $\mathbf{x}_{k+1}$ is defined as the minimizer of the approximation $(f+h)_k(\mathbf{x})$ shown in (4.8). The minimizer $\mathbf{x}_{k+1}$ of (4.8) is typically expressed in terms of the proximal operator $\text{prox}_h(\cdot)$ of a function h which is defined as

$$\text{prox}_h(\mathbf{x}) = \arg\min_{\mathbf{z}} \Big\{ \frac{1}{2}||\mathbf{z}-\mathbf{x}||^2 + h(\mathbf{z}) \Big\}.$$

It follows from (4.8) that the proximal gradient descent update can be expressed in terms of the proximal operator of the function $t_k h$ as

$$\mathbf{x}_{k+1} = \text{prox}_{t_k h}(\mathbf{x}_k - t_k\nabla f(\mathbf{x}_k)) \tag{4.9}$$

The proximal gradient descent algorithm is most useful when the proximal operator has a closed form or is, at least, very easy to compute. For example, consider the case of LASSO regression where the non-smooth component $h(\mathbf{x})$ of the objective function is equal to the L_1 norm multiplied by a tuning parameter λ, i.e., $h(\mathbf{x}) = \lambda||\mathbf{x}||_1$ as L_1 norm. The proximal mapping of h can be expressed as

$$\text{prox}_h(\mathbf{x}) = \arg\min_{\mathbf{z}} \Big\{ \frac{1}{2}||\mathbf{z}-\mathbf{x}||^2 + \lambda||\mathbf{z}||_1 \Big\} = S_\lambda(\mathbf{x}),$$

where the j^{th} component of $S_\lambda(\mathbf{x})$ is given by

$$[S_\lambda(\mathbf{x})]_j = \begin{cases} x_j - \lambda & \text{if } x_j > \lambda \\ 0 & \text{if } -\lambda \leq x_j \leq \lambda \\ x_j + \lambda & \text{if } x_j < -\lambda \end{cases}, \quad j = 1, \ldots, p, \tag{4.10}$$

where x_j is the j^{th} component of $\mathbf{x}$.

As shown in Boyd, Boyd, and Vandenberghe, 2004, both gradient descent and proximal gradient descent are globally convergent with the same convergence rate in convex problems. Assuming f and h are both convex and ∇f is Lipschitz continuous with Lipschitz constant $L_f > 0$ (i.e., $\forall x, y : |\nabla f(x) - \nabla f(y)| \leq L_f$), then, for some constant C, the following inequality holds when a constant step size $t_k = t_f < 1/L_f$ is used

$$f(\mathbf{x}_k) + h(\mathbf{x}_k) - f(\mathbf{x}_\infty) - h(\mathbf{x}_\infty) \leq C\frac{||\mathbf{x}_0 - \mathbf{x}_\infty||^2}{kt_f}, \tag{4.11}$$

where $\mathbf{x}_\infty$ is the optimal point and $\mathbf{x}_0$ the initial point. Therefore, to obtain a precision level which is within at least ε of the optimal value of the objective function, we will need $O(1/\epsilon)$ proximal gradient or gradient descent iterations.

4.3 Acceleration techniques

4.3.1 Anderson acceleration and DAAREM

Anderson acceleration (AA), also known as Anderson mixing, was originally introduced by D.G. Anderson in 1965 to accelerate the rate of convergence of fixed-point iterations in the context of integral equations (Anderson, 1965), and this acceleration technique has turned out to be useful in a range of other applications. Recent examples include computing the nearest correlation matrix (Higham and

Strabić, 2016), reinforcement learning (Geist and Scherrer, 2018), EM acceleration (Henderson and Varadhan, 2019), and electronic structure computations (Fang and Saad, 2009).

The Anderson acceleration algorithm with order m applied to solving the fixed-point problem $f(x) = x$ is shown in Algorithm 1.

Algorithm 1: Anderson acceleration and DAAREM. In the description of the algorithm, $x_{k+1} = f(x_k)$ is the base fixed-point iteration, and m is the order of the acceleration scheme.

Initialize $x_0 \in X$
Set $x_1 = f(x_0)$
for $k = 1, 2, 3, \cdots$ **do**
  Set $m_k = \min\{k, m\}$
  Find the $\{\alpha_j^{k+1}\}$ to solve the following the minimization problem:

$$\min_{\sum_{j=k-m_k}^{k} \alpha_j^{k+1}=1} \left\| \sum_{j=k-m_k}^{k} \alpha_j^{k+1} (f(x_j) - x_j) \right\|^2 + \lambda_k \left\| \boldsymbol{\alpha}_{-k}^{k+1} \right\|^2 \tag{4.12}$$

  Update $x_{k+1} = (1 - \beta_k) \sum_{j=k-m_k}^{k} \alpha_j^{k+1} x_j + \beta_k \sum_{j=k-m_k}^{k} \alpha_j^{k+1} f(x_j)$
  if *meets restart criteria* **then**
    Restart m_k from 1
  end
end

In Algorithm 1, $\boldsymbol{\alpha}_{-k}^{k+1}$ denotes the vector of length m_k containing the values α_j^{k+1} for $j = k - m_k, \ldots, k - 1$, and β_k is the relaxation factor used in Walker and Ni, 2011 and Evans et al., 2020. The non-negative scalar $\lambda_k \geq 0$ is an optional damping factor used in Henderson and Varadhan, 2019, and if $\lambda_k = 0$ for all k, then the update x_{k+1} in Algorithm 1 reduces to the more typical formulation of Anderson acceleration shown, in, for example, Walker and Ni, 2011. Often, the minimization problem (4.12) in Algorithm 1 is stated as an equivalent unconstrained minimization problem with respect to m_k unconstrained parameters

rather than the $m_k + 1$ constrained parameters $\{\alpha_j^{k+1}\}$. This formulation is used, for example, in Higham and Strabić, 2016, and in Henderson and Varadhan, 2019, where the unconstrained version of the minimization problem allows a more direct comparison with so-called multisecant quasi-Newton methods (Fang and Saad, 2009).

The convergence of Anderson acceleration for a general, nonlinear fixed-point iteration has been shown in Toth and Kelley, 2015. A recent work (Evans et al., 2020) proved that Anderson acceleration can improve the convergence rate in a scenario with linear convergence but is not guaranteed to improve the convergence rate in cases of quadratic convergence.

As shown in Algorithm 1, Anderson acceleration can be modified to include *restarts*, where the order m_k is sometimes reset to 1 and all previous memory are dropped. Different restart schema have been proposed for Anderson acceleration. Henderson and Varadhan, 2019 implemented a direct, periodic restart scheme where the algorithm restarts whenever the m_k reaches the value m. Zhang, O'Donoghue, and Boyd, 2020 proposed an adaptive restart where the algorithm only restarts only when the algorithm shows signs of stagnation. In many practical examples, using restarts markedly improves the performance of Anderson acceleration and can reduce the occurrence of algorithm stagnation.

In all of the numerical experiments shown in Section 4.5, we use the version of Anderson acceleration described in Henderson and Varadhan, 2019 which they refer to as the damped Anderson acceleration with restarts and epsilon monotonicity (DAAREM) algorithm. The first component which distinguish DAAREM from many other implementations of Anderson acceleration is the addition of the damping terms $\lambda_k \geq 0$. This L_2 regularization term generates an

update which is a compromise between a pure fixed-point update and a pure Anderson acceleration update. Having large values of λ_k in early iterations and allowing λ_k to decrease in later iterations allows the procedure to bridge the robustness of the original fixed-point iteration with the fast local convergence of Anderson acceleration. Another key component of DAAREM is the use of systematic restarts rather than adaptive restarts, which as mentioned before, is implemented by restarting whenever the value of m_k reaches m. Finally, DAAREM includes some degree of monotonicity control where the fixed-point iteration update is used if the proposed Anderson acceleration increases the objective function (in a minimization problem) by more than a small, pre-specified amount.

4.3.2 SQUAREM

SQUAREM (Varadhan and Roland, 2008) is a technique originally designed to accelerate EM algorithms, but it has also been shown to be useful in accelerating a range of other fixed-point iteration problems. SQUAREM has been acknowledged as a useful, general-purpose acceleration scheme by Lange and others: (Zhou, Alexander, and Lange, 2011),

> Unfortunately, most acceleration techniques are ill-suited to complicated models involving large number of parameters. The squared iterative methods (SQUAREM), recently proposed by Varadhan and Roland, constitute a notable exception.

SQUAREM was motivated by an interesting and highly original modification of the Barzilai-Borwein type spectral gradient algorithm for optimization (Raydan and Svaiter, 2002). SQUAREM readily scales to high-dimensional settings and

is very simple to implement. Hence it has been used in numerous applications to accelerate convergence of underlying iterative algorithm. Examples include: large-scale genome-wide enrichment analysis (Zhu and Stephens, 2018); analysis of human movement (Raket et al., 2016); non-negative matrix factorization across multiple applications (Hobolth et al., 2020); analysis of differential expression in RNAseq data (Jin et al., 2015)); inferring and visualizing cancer mutation signatures (Shiraishi et al., 2015); and signal processing techniques using MM algorithms (Song, Babu, and Palomar, 2016). Convergence of SQUAREM was proved in Varadhan and Roland, 2004 under certain restrictive assumptions. In Varadhan and Roland, 2008, the global convergence was shown for the monotonic version of SQUAREM using the notion of Lyapunov function, which we describe in more detail in the supplementary material. To date, there is no proof that provides an insight on the improved convergence rate from using SQUAREM.

Algorithm 2 describes the SQUAREM acceleration technique for finding a solution of the fixed-point problem $f(x) = x$.

Algorithm 2: SQUAREM. In the description of the algorithm, $x_{k+1} = f(x_k)$ is the base fixed-point iteration.

1 Initialize $x_0 \in X$
2 **for** $k = 1, 2, 3, \cdots$ **do**
3 Set $y_k = f(x_{k-1})$ and $z_k = f(y_k)$
4 Set $r = y_k - x_{k-1}$ and $v = z_k - y_k - r$
5 Compute step length $\alpha = \alpha(r, v)$
6 Update $x_k = x_{k-1} + 2\alpha r + \alpha^2 v$
7 Stabilize $x_k = f(x_k)$
8 **end**

There are different versions of SQUAREM which only differ according to how the step length in step 5 of Algorithm 2 is computed. The three main choices

of the step length are: SqS1 which chooses $\alpha(r,v) = \frac{\langle r,v\rangle}{\langle v,v\rangle}$, SqS2 which chooses $\alpha(r,v) = \frac{\langle r,r\rangle}{\langle r,v\rangle}$, and SqS3 which chooses $\alpha(r,v) = -\frac{||r||}{||v||}$.

One can also relate SQUAREM to an order 1 Anderson acceleration update when the previous iterate has the form $x_k = f(x_{k-1})$. To see why this is the case, note that when $\lambda_k = 0$ and $x_k = f(x_{k-1})$ the solution of the minimization problem (4.12) in Algorithm 1 yields the following update

$$x_{k+1} = x_{k-1} + (\alpha + \beta_k)r + \alpha\beta_k v,$$

where $r = f(x_{k-1}) - x_k$, $v = f(f(x_{k-1})) - 2f(x_{k-1}) + x_{k-1}$ and $\alpha = \frac{\langle r,v\rangle}{\langle v,v\rangle}$. Therefore, a single SqS1-SQUAREM update of an iterate x_{k-1} is equivalent to the following procedure: define $\tilde{x}_k = f(x_{k-1})$ and find x_k by applying an order-1 Anderson acceleration update with $\beta_k = \alpha$ and where x_{k-1} and $\tilde{x}_k = f(x_{k-1})$ are considered to be the previous two iterates. Notice that α in SQUAREM does not necessarily belong to $(0,1]$ and typically it can be much larger than 1, indicating that SQUAREM can be viewed as an over-relaxed version of order-1 Anderson acceleration where β_k is not restricted to the interval $(0,1]$.

4.3.3 Parabolic-EM

Parabolic EM (Berlinet and Roland, 2009) is another extrapolation scheme designed to accelerate the EM algorithm. At each step, parabolic finds new iterate by extrapolating along a Bézier curve $M(t)$ controlled by the most recent three

iterations x_{k-2}, x_{k-1}, x_k. Specifically, $M(t)$ is given by

$$M(t) = (1-t)^2 x_{k-2} + 2t(1-t)x_{k-1} + t^2 x_k$$

$$= x_{k-2} + 2t(x_{k-1} - x_{k-2}) + t^2(x_k - 2x_{k-1} + x_{k-2}).$$

A direct calculation shows that, when recent iterations are obtained from the base EM iterations (i.e., $x_{k-1} = f(x_{k-2})$ and $x_k = f(x_{k-1})$ where f denotes the fixed-point iteration), all three forms of the SQUAREM update $x_{new} = x_{k-2} + 2\alpha(f(x_{k-2}) - x_{k-2}) + \alpha^2(f \circ f(x_{k-2}) - 2f(x_{k-2}) + x_{k-2})$ lie on the curve $M(t)$.

Parabolic EM applies a line search to find t by increasing t from 1 and stopping once the likelihood decreases. If no values of t in the line search are found to increase the likelihood, the algorithm will restart using the original fixed point iteration. Parabolic EM has two sub-types called *arithmetic search* and *geometric search* version which differ only in the way they perform the line search across values of t. Given a step size $h > 0$, arithmetic search evaluates the likelihood at $M(t)$ for $t = 1 + h, 1 + 2h, \ldots$ until the likelihood function decreases at which point the line search stops. Similarly, given both a step size $h > 0$ and exponent $a > 1$, geometric search evaluates the likelihood at $M(t)$ for $t = 1 + a, 1 + a^2 h, \ldots$ and stops whenever the likelihood function decreases. Algorithm 3 describes both the arithmetic and geometric search versions of parabolic EM.

Note that parabolic EM can also be applied to a general fixed-point iteration as long as the fixed-point iteration has an associated loss function to minimize. In that case, one could directly implement Algorithm 3 by replacing the likelihood evaluations in Algorithm 3 with evaluations of the negative of the loss function of interest.

Algorithm 3: Parabolic EM. In the description of the algorithm, $x_{k+1} = f(x_k)$ is the base fixed-point iteration.

Initialize $x_0 \in X$, $x_1 = f(x_0)$, $x_2 = f(x_1)$
for $k = 3, 4, 5, \cdots$ **do**
    $L_2 = \text{Likelihood}(x_{k-1})$
    $i = 0, t = 1 + a^i h$ (geometric) ; $i = 1, t = 1 + ih$ (arithmetic)
    $x_{new} = (1-t)^2 x_{k-3} + 2t(1-t)x_{k-2} + t^2 x_{k-1}$
    $L_{new} = \text{Likelihood}(x_{new})$
    if $L_{new} < L_2$ **then**
        $x_{k-2} = x_{k-1}$; $x_{k-1} = f(x_{k-2})$; $x_k = f(x_{k-1})$
    end
    else
        while $L_{new} \geq L_2$ **do**
            $x_{old} = x_{new}$; $L_2 = L_{new}$
            $i = i + 1$; $t = 1 + a^i h$ (geometric), $t = 1 + ih$ (arithmetic)
            $x_{new} = (1-t)^2 x_{k-3} + 2t(1-t)x_{k-2} + t^2 x_{k-1}$
            $L_{new} = \text{Likelihood}(x_{new})$
        end
        $x_k = f(f(x_{old}))$
    end
end

4.3.4 Quasi-Newton

Zhou, Alexander, and Lange, 2011 proposed a Quasi-Newton method that can applied to accelerating fixed-point iterations. Consider a map $f : X \subset \mathbb{R}^d \to \mathbb{R}^d$ from which we want to find its fixed point x such that $f(x) = x$. This is equivalent as finding the root of function $g(x) = x - f(x)$. If f is assumed to be differentiable with Jacobian df, then Newton's method for finding the root of $g(x)$ yields the following iteration

$$x_{k+1} = x_k - [I - df(x_k)]^{-1} g(x_k). \tag{4.13}$$

The goal in a quasi-Newton approach is to use an approximation of $df(x_k)$ in iteration (4.13) rather than the true $df(x_k)$. The secant method is a well-known root-finding algorithm for a function with scalar inputs that can also be thought of as a quasi-Newton algorithm.

Zhou, Alexander, and Lange, 2011 proposed Quasi-Newton method is based on the linear approximation $f \circ f(x_k) - f(x_k) \approx M(f(x_k) - x_k)$, where x_∞ denotes the fixed point of the iteration and $M = df(x_\infty)$ denotes the Jacobian of f at x_∞. If one sets $v_k = f \circ f(x_k) - f(x_k)$ and $u_k = f(x_k) - x_k$, then the secant requirement for a proposed approximate Jacobian M_k at iteration k would be that $M_k u_k = v_k$. For an improved approximation M_k, Zhou, Alexander, and Lange, 2011 require further that the following q secant conditions $M_k u_{k-j} = v_{k-j}$, $j = 1, \dots q-1$ hold. The matrix M_k with the smallest Frobenius norm among all matrices satisfying these q secant conditions is given by $M_k = V_k(U_k^T U_k)^{-1} U_k^T$, where U_k is the matrix with the q columns $\{u_{k-q+1}, \dots, u_k\}$ and V_k is the matrix with columns $\{u_{k-q+1}, \dots, u_k\}$. Using this approximate Jacobian in iteration (4.13) leads to the order q Quasi-Newton scheme described in Algorithm 4.

Algorithm 4: Quasi-Newton acceleration. In the description of the algorithm, x_k is a vector of length p, q is the order of the acceleration scheme, and $x_{k+1} = f(x_k)$ is the base fixed-point iteration.

Initialize $\beta_0 \in X$. Create an empty $p \times q$ matrix U
for $i = 1, 2, \cdots, q+1$ **do**
    $\beta_1 = f(\beta_0)$
    if $i > 1$ **then**
        Add new column $\beta_1 - \beta_0$ to the right of matrix U
    end
    $\beta_0 = \beta_1$
end
Set $\beta_2 = f(\beta_1)$
Create matrix $V = U$
Remove the first column of V and add column $\beta_2 - \beta_1$ to the right of V
Set $x_0 = \beta_0$
for $k = 1, 2, \cdots$ **do**
    Compute QN-updates
    $x_k = f(x_{k-1}) - V(U^T U - U^T V)^{-1} U^T (x_{k-1} - f(x_{k-1}))$
    Remove the leftmost columns of matrices U and V
    Add column $f(x_k) - x_k$ to the right of U
    Add column $f(f(x_k)) - f(x_k)$ to the right of V
    Check for convergence
end

4.3.5 Restarted Nesterov

Nesterov accelerated gradient descent (Nesterov, 2013; Tseng, 2009) is a popular technique for accelerating first order optimization methods. Using the same notation as in Section 4.2.2.2, Algorithm 5 outlines Nesterov acceleration applied to the composite optimization problem (4.7).

Algorithm 5: Nesterov accelerated proximal gradient descent. In the description of the algorithm, the objective function to be minimized is $f(\mathbf{x}) + h(\mathbf{x})$, where $f(\mathbf{x})$ is assumed to be a smooth function.

Initialize $\mathbf{x}_0 = \mathbf{x}_{-1}$ and $\theta_{-1} = 1, k = 0$
Find the Lipschitz constant L_f of f and set $t = \frac{1}{L_f}$
while *not converge* **do**
    $\theta_k = \frac{\sqrt{\theta_{k-1}^4 + 4\theta_{k-1}^2} - \theta_{k-1}^2}{2}$; $\alpha_k = \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}$
    $\mathbf{y}_k = \mathbf{x}_k + \alpha_k(\mathbf{x}_k - \mathbf{x}_{k-1})$
    $\mathbf{x}_{k+1} = \text{prox}_{th}(\mathbf{y}_k - t\nabla f(\mathbf{y}_k))$
    $k = k + 1$
end

It can be proved that Algorithm 5 has an error rate of $O(\frac{1}{k^2})$, where k is the iteration number. This is a substantial improvement over the $O(\frac{1}{k})$ error rate shown in Equation (4.11) for the fixed step length proximal gradient descent algorithm. Readers can consult Tseng, 2009 for a proof of this result.

Unlike most implementations of gradient descent, Algorithm 5 does not guarantee or check for monotonicity of the objective function. In practice, if you trace the objective value when running Algorithm 5, it is often the case that you see ripples or bumps in the objective function across iterations, which reduces the efficiency of the algorithm. To address this, O'donoghue and Candes, 2015 introduce a heuristic adaptive restart technique to Nesterov acceleration that can dramatically

improve the convergence rate. The basic idea is to reset θ_k to 1 whenever you see an increase of objective function $f(\mathbf{x}_k) + h(\mathbf{x}_k) > f(\mathbf{x}_{k-1}) + h(\mathbf{x}_{k-1})$. Setting $\theta_k = 1$ reduces the momentum term $\alpha_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k)$ to 0 and the accleration algorithm will degenerate to ordinary proximal gradient descent in the following step.

Notice that in Algorithm 5, the momentum coefficients α_k do not depend on the proximal gradient descent updates in any way. Therefore, it is possible to just replace the step $\mathbf{x}_{k+1} = \text{prox}_t(\mathbf{y}_k - t\nabla f(\mathbf{y}_k))$ with any fixed-point iteration $\mathbf{x}_{k+1} = F(\mathbf{y}_k)$ and obtain a Nesterov-like acceleration method for a general fixed-point iteration problem. In Section 4.5, we show that this strategy can accelerate convergence in problems where gradient descent is not the base fixed-point iteration. To the best of our knowledge, this is the first work to examine Nesterov acceleration for general fixed-point iteration problems.

4.4 Implementation

R packages are available to facilitate application of the acceleration schemes described above. For example, the `squarem` package (Du and Varadhan, 2020) implements the SQUAREM algorithm, and the `daarem` package (Henderson and Varadhan, 2020) implements the DAAREM algorithm described in Section 4.3.1. The `turboEM` package (Bobb and Varadhan, 2021) provides a unified API for SQUAREM, Parabolic-EM, and Quasi-Newton acceleration. Currently, `turboEM` does not implement DAAREM, although this should be available in the near future. In `turboEM`, when the objective function value for a proposed update increases the objective value by more than 0.1, we replace this update with one iteration of the base fixed-point iteration. This can dramatically increase the stability of the

Quasi-Newton method and, in many cases, can improve the ultimate convergence speed. Monotonicity control is also a default in the implementations of DAAREM, SQUAREM and Parabolic-EM. The use of restarts in Nesterov acceleration also plays a similar role to monotonicity control as the algorithm is restarted whenever a monotonicity violation occurs. It is worth mentioning that the implementation of the Quasi-Newton in `turboEM` includes the option of monotonicity control even though monotonicity control was not originally implemented in (Zhou, Alexander, and Lange, 2011).

Our package `AccelBenchmark` is available from Github which can be used to easily benchmark all of the methods described in this paper, and this is the package that we actually used for all the experiments described in this paper.

There is an R package called `FixedPoint` which contains various acceleration methods for fixed-point problems, including Anderson acceleration and several vector extrapolation algorithms. A fundamental difference between that package and our software packages is that `FixedPoint` doesn't contain safeguards such as steplength and monotonicity controls, damping, and restarts. Consequently, the algorithms are less reliable for general purpose use.

4.5 Experiments

4.5.1 Settings for the experiments

In all of the experiments in this paper we have used the default control parameters implemented in each of the acceleration packages (`turboEM` and `daarem`). We did not optimize the algorithmic settings for each problem. This is an important point because we would like to explore the performance of these methods when they are

used directly off-the-shelf. Also, unless otherwise stated, convergence is defined as the first iteration where the norm of the parameter difference $||x_{k+1} - x_k||$ is less than 10^{-7}. We evaluated the performance of each acceleration algorithm in terms of the number of fixed-point iterations (*fpevals*) and the elapsed time in seconds (*elapsed*). We report the mean $\pm$ standard deviation of *fpevals* and *elapsed* across a certain number of simulated experiments. Some of the performance metrics have distributions with a heavy tail making the standard deviation bigger than the mean, which is a sign of instability of that algorithm. We also report the number of failures (# *failures*), where failure is defined as not achieving convergence within an allotted number of iterations, which varied for each problem. We also plotted the convergence trajectories of algorithms in terms of the objective function (e.g., log-likelihood). The panels of Figure 1 in the Supplementary Material display the objective function values versus fixed-point iteration for each of the five main acceleration methods. The *Loss* item in the figures are normalized by subtracting the minimum value of the objective function. For every experiment, we only plot results for the most difficult setting (e.g., $\nu = 25$ in the multivariate t distribution).

4.5.2 Multivariate t-distribution

A d-dimensional Student-t distribution $T_\nu(\mu, \Sigma)$ with $\nu > 0$ degrees of freedom, location parameter $\mu \in \mathbb{R}^d$, and positive definite scatter matrix Σ has the density function:

$$p(x|\nu, \mu, \Sigma) = \frac{\Gamma(\frac{d+\nu}{2})}{\Gamma(\frac{\nu}{2})\nu^{\frac{d}{2}}\pi^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \frac{1}{(1 + \frac{1}{\nu}(x-\mu)^T\Sigma^{-1}(x-\mu))^{\frac{d+\nu}{2}}},$$

where $\Gamma(s)$ denotes the Gamma function $\Gamma(s) = \int_0^\infty t^{s-1}e^{-t}dt$. For observations $x_1, \ldots, x_n$ arising from a d-dimensional Student-t distribution, setting the

derivative of the associated log likelihood function with user-specified weights $\omega_1, \ldots, \omega_n$ to zero results in the following system of equations

$$0 = \sum_{i=1}^{n} \omega_i \frac{x_i - \mu}{\nu + (x_i - \mu)^T \Sigma^{-1} (x_i - \mu)}, \tag{4.14}$$

$$I = (d + \nu) \sum_{i=1}^{n} \omega_i \frac{\Sigma^{-\frac{1}{2}} (x_i - \mu)(x_i - \mu)^T \Sigma^{-\frac{1}{2}}}{\nu + (x_i - \mu)^T \Sigma^{-1} (x_i - \mu)}, \tag{4.15}$$

$$\begin{aligned} 0 =& \phi\left(\frac{\nu}{2}\right) - \phi\left(\frac{\nu + d}{2}\right) \\ &+ \sum_{i=1}^{n} \omega_i \left(\frac{\nu + d}{\nu + (x_i - \mu)^T \Sigma^{-1} (x_i - \mu)} - \log\left(\frac{\nu + d}{\nu + (x_i - \mu)^T \Sigma^{-1} (x_i - \mu)} \right) - 1 \right) \end{aligned} \tag{4.16}$$

where $\phi(x) = \frac{1}{\Gamma(x)} \frac{d\Gamma(x)}{dx} - \log(x)$ and where the weights ω_i are assumed to satisfy $\omega_i \geq 0$, $\sum_{i=1}^{n} \omega_i = 1$.

Here, we are interested in maximizing the weighted log-likelihood function under the assumption that all parameters, including the degrees of freedom ν, are unknown. Hasannasab et al., 2021 show that, under certain conditions, either a minimizer of the negative weighted log-likelihood exists, or the maximum likelihood estimator corresponds to the case $\nu \to \infty$, for which the Student-t distribution approaches the Gaussian distribution.

Because one can represent a $T_\nu(\mu, \Sigma)$ random variable as

$$\mu + \Sigma^{\frac{1}{2}} Z / \sqrt{Y} \sim T_\nu(\mu, \Sigma), \tag{4.17}$$

where $Z \sim N(0, I)$ and $Y \sim \Gamma(\nu/2, \nu/2)$, one can develop an EM algorithm for parameter estimation by augmenting the observed data $x_1, \ldots, x_n$ with latent data $y_1, \ldots, y_n$ where it is assumed that $y_i \sim \Gamma(\nu/2, \nu/2)$ independently and use the

fact that the conditional distribution $x_i|y_i$ is a multivariate normal distribution with mean vector μ and covariance matrix Σ/y_i. This data augmentation leads to the conventional EM algorithm (Liu and Rubin, 1995) for solving equations (4.14) - (4.16). The conventional EM algorithm for the multivariate t-distribution is described in Algorithm 6.

Algorithm 6: EM for estimating multivariate t-distribution parameters.

Initialize ν_0, μ_0, Σ_0.
for $k = 1, 2, \cdots$ **do**
  E-Step:
    $z_{i,k} = (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)$
    $\gamma_{i,k} = \frac{\nu_k + d}{\nu_k + z_{i,k}}$
  M-Step:
    $\mu_{k+1} = \frac{\sum_i \omega_i \gamma_{i,k} x_i}{\sum_i \omega_i \gamma_{i,k}}$; $\Sigma_{k+1} = \sum_i \omega_i \gamma_{i,k} (x_i - \mu_{k+1})(x_i - \mu_{k+1})^T$
    $\nu_{k+1} =$ **zero** of $\phi\left(\frac{\nu}{2}\right) - \phi\left(\frac{\nu_k + d}{2}\right) + \sum_i \omega_i (\gamma_{i,k} - \log(\gamma_{i,k}) - 1)$
end

Recently, Hasannasab et al., 2021 suggested a number of alternative algorithms for such problem that accelerated the naive EM algorithm. Among them, the Multivariate Myriad Filter (MMF) algorithm shows excellent overall performance. The MMF algorithm is the same as the EM algorithm (Algorithm 6) except that the updates for Σ_k and ν_k are changed to

$$\Sigma_{k+1} = \sum_{i=1}^{n} \frac{\omega_i \gamma_{i,k} (x_i - \mu_{k+1})(x_i - \mu_{k+1})^T}{\sum_{i=1}^{n} \omega_i \gamma_{i,k}}$$

$$\nu_{k+1} = \textbf{zero of } \phi\left(\frac{\nu}{2}\right) - \phi\left(\frac{\nu + d}{2}\right) + \sum_{i=1}^{n} \omega_i \left(\frac{\nu_k + d}{\nu_k + z_{i,k+1}} - \log\left(\frac{\nu_k + d}{\nu_k + z_{i,k+1}}\right) - 1 \right).$$

We conducted a simulation study to determine whether or not our black-box acceleration schemes can further accelerate the fast MMF algorithm. In this simulation study, we set $\mu = (0, 0)$ and $\Sigma = \text{diag}\{0.1, 1\}$, and we considered

the three choices for the degrees of freedom ν from $\{3, 10, 25\}$. For each choice of ν, we simulated 1000 observations from the corresponding t-distribution and ran the MMF algorithm together with all acceleration methods to estimate the parameters μ, Σ, ν, and this procedure was repeated across 200 simulation runs. In each run, we initialized μ_0 to be the value of the sample mean, Σ_0 be the sample covariance matrix, and we initialized the degrees of freedom ν_0 by sampling with equal probability from the set $\{2, 3, 4\}$. The hyperparameters of the acceleration algorithms and the convergence criteria are set to their defaults. These default settings are discussed in detail in Section 4.4.

Results from this simulation study can be found in Table 4.1. From this table, we observe that SQUAREM, DAAREM and parabolic-EM (pEM) provide consistent acceleration when compared to the original MMF algorithm, and the factor of speedup from these methods increases as ν increases. Quasi-Newton and Nesterov acceleration also accelerated the MMF algorithm in some simulation settings, but the improvement over MMF was not as consistent as SQUAREM, DAAREM, and pEM.

It is interesting to note that the MMF algorithm already gives a huge speed advantage when compared to the original EM algorithm. For example, in the $\nu = 25$ case, the EM algorithm required, on average, 1235 fixed point iterations before convergence which is 9 times more than that of the MMF algorithm. Despite the fact MMF is much faster than EM, we can still further accelerate MMF using the methods described in this manuscript.

Metric	MMF	SQUAREM	DAAREM	pEM	Quasi-Newton	Nesterov
fpevals (ν=3)	34.3 ± 2.94	15.2 ± 1.31	$\mathbf{14.4 \pm 1.45}$	21.1 ± 2.4	17.8 ± 17.9	29.5 ± 2.5
elapsed (ν=3)	1.12 ± 0.097	$\mathbf{0.398 \pm 0.035}$	0.501 ± 0.056	0.693 ± 0.088	0.678 ± 0.718	0.929 ± 0.085
# failures (ν=3)	0	0	0	0	0	0
fpevals (ν=10)	67.7 ± 12.7	20.4 ± 3.67	$\mathbf{16.0 \pm 1.42}$	22.5 ± 1.65	51.4 ± 39.3	50.2 ± 14.2
elapsed (ν=10)	2.20 ± 0.419	$\mathbf{0.54 \pm 0.103}$	0.561 ± 0.056	0.761 ± 0.063	1.38 ± 1.08	1.58 ± 0.451
# failures (ν=10)	0	0	0	0	0	0
fpevals (ν=25)	128 ± 42.1	22.9 ± 4.61	$\mathbf{17.3 \pm 3.37}$	24.9 ± 1.88	30.7 ± 12.8	77.8 ± 150
elapsed (ν=25)	4.11 ± 1.36	$\mathbf{0.61 \pm 0.127}$	0.615 ± 0.14	0.859 ± 0.08	1.22 ± 0.52	2.46 ± 4.81
# failures (ν=25)	0	0	0	0	0	1

Table 4.1: Simulation results for the multivariate t-distribution from 200 independent runs. MMF represents the original MMF algorithm described above, and other columns show results for different accelerated version of it. If an algorithm failed to converge or if it converged to a negative log-likelihood more than 1% larger than that of the original MMF algorithm, then we called it a failure. As a measure of robustness, we also recorded the number of failures for each method.

4.5.3 Poisson Mixtures

A finite mixture of Poisson distributions with C components has the following discrete probability distribution

$$f(y \mid \mathbf{p}, \boldsymbol{\lambda}) = \sum_{c=1}^{C} p_c f_{\mathcal{P}}(y \mid \lambda_c), \tag{4.18}$$

where $f_{\mathcal{P}}(y \mid \lambda_c) = e^{-\lambda_c}\lambda_c^y / y!$ denotes the probability distribution of y conditional on belonging to the c^{th} cluster of the mixture distribution. For observations $y_1, \ldots, y_n$, we can develop an EM algorithm for estimating the parameters in (4.18) by introducing latent variables $z_1, \ldots, z_n$ defined as $z_i = c$ if y_i belongs to the c^{th} cluster. This particular data augmentation scheme generates the following EM algorithm updates for the parameters of interest $(p_1, \ldots, p_C)$ and $(\lambda_1, \ldots, \lambda_C)$:

$$\hat{\pi}_{ic}^{(k)} = p_c^{(k)} \left(\lambda_c^{(k)}\right)^{y_i} e^{-\lambda_c^{(k)}} \Big/ \sum_{l=1}^{C} p_l^{(k)} \left(\lambda_l^{(k)}\right)^{y_i} e^{-\lambda_l^{(k)}}$$

$$p_c^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n} \hat{\pi}_{ic}^{(k)}; \qquad \lambda_c^{(k+1)} = \sum_{i=1}^{n} \hat{\pi}_{ic}^{(k)} y_i \Big/ \sum_{i=1}^{n} \hat{\pi}_{ic}^{(k)}.$$

We explored our acceleration methods using the real count data from Hasselblad, 1966 which contains 1096 observations with each observation representing

a day of survival. In this dataset, the observations range from 0 to 9 and the frequencies for these values are 162, 267, 271, 185, 111, 61, 27, 8, 3, 1 respectively, no censoring is presented. Using this dataset, we fit a finite mixture of Poisson distributions with 2 components so the parameters of interest are the mixture probability p_1 and the cluster-specific Poisson rates λ_1 and λ_2.

To study the performance of each acceleration procedure, we ran the original EM algorithm and all acceleration schemes 500 times. In each run, the mixture probability p_1 was drawn from a uniform distribution over $(0, 1)$, and the Poisson rates λ_c were independently drawn from a uniform distribution over $(0, 4)$. In the Quasi-Newton algorithm, the order was set to 2 since we only have 3 parameters in this problem. The maximum number of fixed-point iteration evaluations is set to 3000. Other hyperparameters of the acceleration algorithm and convergence criteria are set to their defaults, which will be discussed in more detail in Section 4.4. Results can be found in Table 4.2. In this experiment, all of the methods listed in Table 4.2 dramatically accelerated the original EM algorithm with an up to 11-fold reduction in execution time.

Metric	EM	SQUAREM	DAAREM	pEM	Quasi-Newton	Nesterov
fpevals	2238 ± 273	77 ± 19.1	$\mathbf{53 \pm 14.3}$	195 ± 101	131 ± 144	166 ± 23.5
elapsed	32.8 ± 6.34	$\mathbf{2.82 \pm 4.62}$	5.4 ± 4.28	5.49 ± 4.96	5.01 ± 5.82	3.82 ± 3.45
# failures	0	0	0	0	1	0

Table 4.2: Simulation results for estimating Poisson mixture parameters from 500 independent runs. Elapsed time are reporting in millisecond. If an algorithm failed to converge or if it converged to a negative log-likelihood more than 1% larger than that of the original EM algorithm, then we called that run a failure. As a measure of robustness, we also recorded the number of failures for each acceleration method.

4.5.4 LASSO

The least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996) is a widely used technique for high dimensional inference due to its ability to perform simultaneous feature selection and coefficient estimation. However, the additional L_1 penalty in the LASSO objective function also makes it impossible to obtain a general, closed-form solution for the regression coefficient estimates, and therefore, iterative algorithms are needed for optimization. A simple but effective iterative algorithm is proximal gradient descent which uses the iteration (4.9) described in Section 4.2.2.2.

In this experiment, we use the Madelon data (Guyon et al., 2004) to study the performance of difference acceleration methods applied to the proximal gradient descent algorithm. The Madelon data is artificially constructed to illustrate a particular difficulty for feature selection. It contains $n = 2600$ binary outcomes $y_1, \ldots, y_n$, and for each y_i, we have a predictor vector $\mathbf{x}_i$ of length $p = 500$. We use the logistic regression version of LASSO where the objective function $\ell(\boldsymbol{\beta})$ to be minimized is given by

$$\ell(\boldsymbol{\beta}) = f(\boldsymbol{\beta}) + h(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(\log\left(1 + e^{\mathbf{x}_i^T \boldsymbol{\beta}}\right) - y_i \mathbf{x}_i^T \boldsymbol{\beta} \right) + \lambda \sum_{p=1}^{p} |\beta_p|,$$

where λ is a parameter that controls the regularization level and where $h(\boldsymbol{\beta}) = \lambda \sum_{p=1}^{p} |\beta_p|$. Following (4.9) - (4.10) yields the following proximal gradient descent iteration

$$\begin{aligned} \boldsymbol{\beta}_{k+1} &= \text{prox}_{t_k h}\left(\boldsymbol{\beta}_k - t_k \nabla f(\boldsymbol{\beta}_k)\right) \\ &= S_{\lambda t_k}\left(\boldsymbol{\beta}_k - t_k \mathbf{X}^T \{\mathbf{y} - \mu(\mathbf{X}\boldsymbol{\beta}_k)\}\right), \end{aligned} \tag{4.19}$$

where $\mathbf{X}$ is the $n \times p$ matrix whose i^{th} row is $\mathbf{x}_i^T$ and where $\mu(\mathbf{X}\boldsymbol{\beta}_k)$ is the n-dimensional vector whose i^{th} element is $1/\{1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta}_k)\}$.

In each simulation run, we initialized each coefficient β_j independently and uniformly within $(-1, 1)$, and we then applied the proximal gradient descent algorithm (4.19) and each of the acceleration versions of it. The values of the tuning parameter λ where set to one of three values $\lambda \in \{0.1, 1, 10\}$. The maximum number of fixed-point iterations for each method was set to $20,000$. Other hyperparameters of the acceleration methods and convergence criteria were set to their defaults. For each choice of λ, we evaluated all of the methods using 200 independent simulation runs.

Results from this simulation study can be found in Table 4.3. On average, all of the methods provided consistent acceleration of the original proximal gradient descent algorithm. Among the different acceleration methods, DAAREM consistently gave the greatest acceleration with a more than 20 fold improvement in execution time across different choices of λ. The failure of gradient descent for smaller values of λ is due to slow convergence, i.e. it did not converge with a maximum number of iterations.

Metric	pGD	SQUAREM	DAAREM	pEM	Quasi-Newton	Nesterov
fpevals ($\lambda = 10$)	11125 ± 1721	351 ± 53.2	$\mathbf{214 \pm 61.2}$	1894 ± 126	617 ± 170	404 ± 53.7
elapsed ($\lambda = 10$)	31.3 ± 4.85	0.782 ± 0.13	$\mathbf{0.711 \pm 0.22}$	5.47 ± 0.678	1.68 ± 0.49	1.23 ± 0.22
# failures ($\lambda = 10$)	0	0	0	0	0	0
fpevals ($\lambda = 1$)	19292 ± 720	1057 ± 104	$\mathbf{603 \pm 481}$	3094 ± 597	2947 ± 1874	676 ± 44.4
elapsed ($\lambda = 1$)	56.3 ± 5.84	2.46 ± 0.645	$\mathbf{2.01 \pm 1.69}$	9.09 ± 2.32	7.78 ± 4.88	2.17 ± 0.84
# failures ($\lambda = 1$)	54	0	10	0	1	0
fpevals ($\lambda = 0.1$)	20000+	1063 ± 156	$\mathbf{618 \pm 429}$	2603 ± 80.8	3112 ± 1741	658 ± 77.6
elapsed ($\lambda = 0.1$)	58 ± 2.46	2.52 ± 0.717	$\mathbf{1.96 \pm 1.34}$	7.55 ± 0.55	8.74 ± 4.85	2.08 ± 0.33
# failures ($\lambda = 0.1$)	200	0	0	0	0	0

Table 4.3: Simulation results for estimating regression coefficients using LASSO logistic regression with 200 independent runs. *pGD* represents the original proximal gradient descent algorithm, and the other columns represent different acceleration methods. If an algorithm failed to converge or if it converged to a loss more than 1% larger than the optimal loss, we considered it to be a failure. As a measure of robustness, we also recorded the number of failures for each acceleration method.

4.5.5 Variational Inference in Bayesian Variable Selection

Bayesian variable selection methods for a regression model (e.g. George and McCulloch, 1997, Chipman et al., 2001) with binary outcomes often consider the following model:

$$\text{logit}\{p(y_i = 1|\mathbf{X}, \beta)\} = \beta_0 + \sum_{j=1}^{p} X_{ij}\beta_j; \quad \beta_j = \gamma_j Z_j;$$

$$\gamma_j \sim \text{Bern}(\pi); \quad Z_j \sim \mathcal{N}(0, \sigma_\beta^2); \quad \theta = (\pi, \sigma_\beta^2) \sim p_\theta(\cdot),$$

where $\text{Bern}(p)$ denotes the Bernoulli distribution with success probability p and $p_\theta(\cdot)$ denotes the prior distribution for the vector of hyperparameters $\theta = (\pi, \sigma_\beta^2)$. The values of the variables γ_j drive the model selection as $\beta_j = 0$ whenever $\gamma_j = 0$. For simplicity, in this section we set $\beta_0 = 1$ as a known constant.

In Bayesian variable selection, calculation of the marginal posterior inclusion probabilities is a primary interest. The marginal posterior inclusion probability for variable j, defined as $PIP(j) = p(\gamma_j = 1|\mathbf{X}, y)$, can be expressed as

$$PIP(j) = \int p(\gamma_j = 1|\mathbf{X}, y, \theta) p(\theta|\mathbf{X}, y) d\theta, \tag{4.20}$$

where $\mathbf{X}$ is the $n \times p$ design matrix whose (i, j) element is X_{ij} and $y = (y_1, \ldots, y_n)$.

The quantity $p(\gamma_j = 1|\mathbf{X}, y, \theta)$ does not have a closed-form and is often calculated using Markov Chain Monte Carlo (MCMC) methods, for example Bottolo, Richardson, et al., 2010 and Clyde, Ghosh, and Littman, 2011. However, in high-dimensional applications, MCMC can be very computationally inefficient and often requires days or even weeks to run. To address this, Carbonetto, Stephens, et al., 2012 propose using variational inference to approximate $p(\gamma_j = 1|\mathbf{X}, y, \theta)$

and then approximate the integral in (4.20) with importance sampling.

To approximate $p(\gamma_j = 1|\mathbf{X}, y, \theta)$, one first approximates the posterior density $p(\beta, \gamma|\mathbf{X}, y, \theta)$ with a density function of the form $q(\beta, \gamma|\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{s}^2) = \prod_{j=1}^{p} q(\beta_j, \gamma_j|\alpha_j, \mu_j, s_j^2)$, where the components of this product are assumed to be a mixture of a normal density and a point mass at 0. Specifically,

$$q(\beta_j, \gamma_j|\alpha_j, \mu_j, s_j^2) = \begin{cases} \alpha_j N(\beta_j|\mu_j, s_j^2) & \text{if } \gamma_j = 1 \\ (1-\alpha_j)\delta_0(\beta_j) & \text{otherwise,} \end{cases} \tag{4.21}$$

where $N(\cdot|\mu, \sigma^2)$ stands for a normal density with mean μ and variance σ^2 and $\delta_0(\cdot)$ for a delta mass centered at 0. To find the best set of parameters, Carbonetto, Stephens, et al., 2012 derives an EM-type algorithm that can maximize the evidence lower bound (ELBO):

$$\begin{aligned} s_j^2 &= \frac{1}{(\mathbf{X}^T\hat{U}\mathbf{X})_{jj} + 1/\sigma_\beta^2} \\ \mu_j &= s_j^2\left((X^T\hat{y})_j - \sum_{h\neq j}(\mathbf{X}^T\hat{U}\mathbf{X})_{hj}\alpha_h\mu_h\right) \\ \frac{\alpha_j}{1-\alpha_j} &= \frac{\pi}{1-\pi} \times \frac{s_j}{\sigma_\beta} \times \exp\left(\frac{\mu_j^2}{2s_j^2}\right) \end{aligned} \tag{4.22}$$

where $(\mathbf{X}^T\mathbf{X})_{hj}$ denotes the (h, j) element of the matrix $\mathbf{X}^T\mathbf{X}$ and $(\mathbf{X}^T y)_j$ denotes the j^{th} element of the vector $\mathbf{X}^T y$. Also, $\hat{U}$ and $\hat{y}$ are defined as $\hat{U} = \text{diag}\{\boldsymbol{u}\} - \boldsymbol{u}\boldsymbol{u}^T/\bar{\boldsymbol{u}}$; $\hat{y} = y - \frac{1}{2} - \boldsymbol{u}$, where $\boldsymbol{u} = (u_1, \ldots, u_n)$ and $\bar{\boldsymbol{u}} = \sum_{i=1}^{n} u_i$. The terms $u_i = \frac{1}{\eta_i}\big(\psi(\eta_i) - 1/2\big)$ are updated from

$$\eta_i^2 = \left(\sum_{j=1}^{p} X_{ij}\mathbb{E}[\beta_j]\right)^2 + \sum_{j=1}^{p} X_{ij}^2\text{Var}[\beta_j],$$

where $\psi(x) = 1/(1+e^{-x})$. From the variational approximation (4.21), we have $\mathbb{E}[\beta_j] = \alpha_j\mu_j$ and $\text{Var}[\beta_j] = \alpha_j(s_j^2 + \mu_j^2) - (\alpha_j\mu_j)^2$.

In this experiment, we examined whether the acceleration schemes described above can effectively accelerate the coordinate descent algorithm for Bayesian variable selection in the case of logistic regression. For this simulation study, we use a setting similar to that described in Carbonetto, Stephens, et al., 2012. Specifically, we assume that $\text{logit}\{p(y_i = 1|\beta, \mathbf{X})\} = -1 - Z_{i1} + Z_{i2} + \sum_{j=1}^{p} X_{ij}\beta_j$, where Z_{i1} and Z_{i2} are independent standard normal distributions. In these simulations, $\beta = (\beta_1, \ldots, \beta_p)$ has length 2000, and only the first m components of β are assumed to be nonzero. The nonzero components of β were sampled independently from a $N(0, 0.25)$ distribution, and the remaining components were set to 0. The elements X_{ij} of $\mathbf{X}$ were drawn independently from a Binomial$(2, p_{ij})$, where p_{ij} is drawn from a uniform distribution over $(0.05, 0.5)$. Values of $m = 100$ and $m = 200$ were considered, and we performed 200 simulation runs for each setting of m. In each run, we set the sample size and number of covariates to $n = 200$ and $p = 2000$ respectively. We initialized the α_j by drawing independently from a uniform distribution over $(0, 1)$, the μ_j were initialized by drawing from a normal distribution with mean 0 and variance 0.1, and the s_j do not need to be initialized as they are first updated using equation (4.22). The model hyperparameters (π, σ_β^2) were set to $\pi = m/2000$ and $\sigma_\beta = 0.5$. The maximum number of fixed-point iterations was set to 1000. The hyperparameters of the acceleration algorithms and convergence criteria were set to their default values (see Section 4.4).

Results from this simulation study are shown in Table 4.4. We do not observe a huge acceleration in this experiment, but SQUAREM still provides a consistent speedup with a roughly 4 fold improvement for both settings of m. DAAREM, on

the other hand, failed frequently partly due to it convergence to a solution with a slightly higher than the optimal value of the loss function.

Metric	EM	SQUAREM	DAAREM	pEM	Quasi-Newton	Nesterov
fpevals ($m = 100$)	207 ± 11.7	115 ± 21.5	883 ± 230	158 ± 22	349 ± 205	$\mathbf{77.5 \pm 10.3}$
elapsed ($m = 100$)	4.71 ± 0.54	$\mathbf{1.3 \pm 0.29}$	20.4 ± 5.49	3.65 ± 0.64	5.06 ± 3.06	1.76 ± 0.31
# failures ($m = 100$)	0	0	150	0	0	4
fpevals ($m = 200$)	194 ± 7.42	102 ± 13.5	157 ± 37.3	132 ± 22.6	603 ± 310	$\mathbf{81.3 \pm 6.88}$
elapsed ($m = 200$)	4.19 ± 0.26	$\mathbf{1.1 \pm 0.15}$	3.6 ± 0.85	2.89 ± 0.47	8.29 ± 4.23	1.75 ± 0.21
# failures ($m = 200$)	0	0	0	0	51	0

Table 4.4: Simulation results for Bayesian variable selection using 200 independent runs. *EM* represents the original algorithm with the various acceleration methods in other columns. If an algorithm failed to converge or if it converged to a loss more than 1% larger than the optimal loss, we called it a failure. We also recorded the number of failures for each method as a measure of robustness.

4.5.6 Sinkhorn Scaling

Given a matrix A, the problem of re-scaling its rows and columns to form a doubly stochastic matrix $\Gamma = DAE$, where D and E are diagonal matrices, is called a matrix balancing problem. A more constrained version of the matrix balancing problem is to find diagonal scaling matrices D, E such that $\Gamma = DAE$ and that Γ has specified row and column sums, that is, $\Gamma\mathbf{1} = \mathbf{a}$; $\Gamma^T\mathbf{1} = \mathbf{b}$, where $\mathbf{1}$ is a vector whose entries are all equal to 1. This problem has a vast array of applications including, for example, ranking web pages (Knight, 2008), learning permutation matrices from data (Mena et al., 2018), solving optimal transport problems (Altschuler, Weed, and Rigollet, 2017), etc.

A naive algorithm for the constrained matrix balancing problem is the Sinkhorn–Knopp algorithm (Sinkhorn and Knopp, 1967). The algorithm simply scales the matrix iteratively by rows and columns. Given an initialization $\mathbf{u}_0, \mathbf{v}_0$, the algorithm finds the next updates by

$$\mathbf{u}_{k+1} = \frac{\mathbf{a}}{A\mathbf{v}_k}; \quad \mathbf{v}_{k+1} = \frac{\mathbf{b}}{A^T\mathbf{u}_{k+1}}, \tag{4.23}$$

where the division of vectors in (4.23) is done element-wise and A is the matrix

with entries A_{ij}.

In this numerical experiment, we tested the performance of the acceleration methods on the Sinkhorn-Knopp algorithm for a constrained matrix balancing problem where we used certain ill-conditioned matrices known as Marshall-Olkin and Hessenberg matrices (Parlett and Landis, 1982). A Marshall-Olkin matrix $\boldsymbol{M}_3$ is a 3×3 matrix with columns $(100, 100, 0)^T$, $(100, 10000, 1)^T$, $(0, 1, 100)^T$. We choose the order n Hessenberg matrix $\boldsymbol{H}_n$ to be the $n \times n$ matrix such that $\boldsymbol{H}_n(i, i) = 100$ for all i and $\boldsymbol{H}_n(i, j) = 1$, for all (i, j) such that $j > i - 1, j \neq i$. All of the other elements in $\boldsymbol{H}_n$ are equal to 0. For simplicity, **a** and **b** in iteration (4.23) will be set to all 1s.

We run such experiment for 200 independent times. In each run, we used Hessenberg matrices of order 10 and 50. We initialized parameter $\boldsymbol{v}$ as i.i.d draws from a uniform distribution between $[0.5, 2]$. Notice that $\boldsymbol{u}$ can be calculate from algorithm (4.23), therefore we do not treat it as part of the parameter vector in the acceleration algorithms. The maximum number of fixed-point iterations was set to 50,000. We measured the performance of the scaling algorithm by calculating the mean absolute differences (MAD) between row/column sums and 1. If the two MAD values were, at any time, both smaller than 10^{-10} we terminated the algorithm and regarded it to be converged. Other hyperparameters of the acceleration algorithms are set to their defaults (see Section 4.4).

Results from this numerical experiment can be found in Table 4.5. The results in Table 4.5 show that we gain substantial and consistent acceleration by using either Nesterov with restarts or SQUAREM.

It is also possible to use a different approach to matrix scaling, which uses

a different fixed-point iteration (see Supplementary section 3). By using the intermediate scaled matrix $\text{diag}\{\boldsymbol{u}_k\}\ A\ \text{diag}\{\boldsymbol{v}_k\}$ as parameter vector rather than $\boldsymbol{u}, \boldsymbol{v}$. The acceleration schemes perform much better with this approach. Although the final output from this method is not guaranteed to be feasible, we confirmed that the relative difference is small. For example, in $\boldsymbol{H}_{50}$ case, the DAAREM algorithm, which originally failed, converged in a few hundreds of iterations. It also gave a reasonably accurate answer, the discrepancy from true result being smaller than 10^{-5}.

Metric	SK	SQUAREM	DAAREM	pEM	Quasi-Newton	Nesterov
fpevals ($\boldsymbol{M}_3$)	5764 ± 121	68 ± 10.1	29.4 ± 0.83	1243 ± 89.9	$\mathbf{27.5 \pm 10.8}$	218 ± 1.45
elapsed ($\boldsymbol{M}_3$)	185 ± 13.9	2.36 ± 0.69	3.88 ± 0.65	39.5 ± 3.91	$\mathbf{1.55 \pm 1.91}$	6.76 ± 4.51
# failures ($\boldsymbol{M}_3$)	0	0	0	0	0	0
fpevals ($\boldsymbol{H}_{10}$)	2671 ± 2.5	$\mathbf{146 \pm 53.6}$	3216 ± 5888	804 ± 17.1	7139 ± 1933	246 ± 10.1
elapsed ($\boldsymbol{H}_{10}$)	83.2 ± 4.26	$\mathbf{4.88 \pm 3.51}$	388 ± 721	29.3 ± 1.89	475 ± 148	8.14 ± 4.46
# failures ($\boldsymbol{H}_{10}$)	0	0	1	0	0	0
fpevals ($\boldsymbol{H}_{50}$)	50000+	4857 ± 10503	50000+	17853 ± 15.2	33307 ± 1910	$\mathbf{1336 \pm 735}$
elapsed ($\boldsymbol{H}_{50}$)	4700+	218 ± 516	7610+	1130 ± 37.2	3139 ± 168	$\mathbf{71.1 \pm 38.9}$
# failures ($\boldsymbol{H}_{50}$)	200	10	200	0	0	0

Table 4.5: Experimental results for matrix scaling from 200 independent runs. *SK* represents the original Sinkhorn-Knopp algorithm, and the other columns are different accelerated versions of it. Elapsed time are reported in milliseconds. The number of failures (failure to converge) is also recorded to capture the robustness of each algorithm.

4.5.7 Manifold Embedding

Manifold learning is a useful approach for performing both dimension reduction and data visualization of high-dimensional data. However, many manifold learning approaches can potentially be influenced by the exploding distance in high-dimensional space, and make points with moderate distance in the original space become too crowded in the embedded low-dimensional space. This phenomenon is of called the "crowding problem" (Cook et al., 2007). To address this problem, Maaten and Hinton, 2008 extended the Stochastic Neighborhood Embedding (SNE) method by using a t-distributed neighborhood probability for

points in the embedded space. The corresponding method is called t-SNE, and this method achieved great success in visualizing handwritten digits data and a variety of other higher-dimensional images and text data.

The main input into the t-SNE procedure, is a collection of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from which one can compute nonnegative similarity scores P_{ij} between pairs $\mathbf{x}_i$ and $\mathbf{x}_j$ normalized so that $\sum_{ij} P_{ij} = 1$. Given the matrix P containing the values of P_{ij}, t-SNE seeks to find an embedding matrix $Y = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n)^T$ with each $\mathbf{y}_k$ in a lower-dimensional space by minimizing the following KL divergence-based loss function with respect to Y

$$L(Y; P) = \sum_{ij} P_{ij} \log \frac{P_{ij}}{Q_{ij}} \tag{4.24}$$

$$Q_{ij} = \frac{\left(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2\right)^{-1}}{\sum_{ab} \left(1 + ||\mathbf{y}_a - \mathbf{y}_b||^2\right)^{-1}}. \tag{4.25}$$

Yang, Peltonen, and Kaski, 2015 derived an alternative algorithm for solving a range of manifold embedding problems and claimed it can be more efficient than many existing algorithms. The algorithm proposed by Yang, Peltonen, and Kaski, 2015 is essentially an MM algorithm that iteratively majorizes the complex loss function of interest by a specially designed quadratic form and then minimizing it with a closed-form solution. In the context of t-SNE where we want to minimize (4.24), this algorithm results in the following updating scheme for a current embedding matrix Y_k

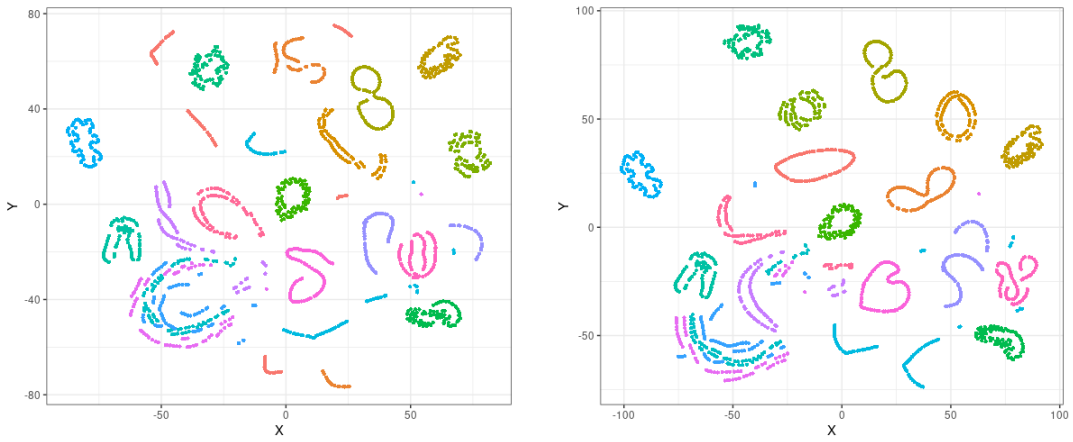
$$Y_{k+1} = \left(\mathcal{L}_{P \circ q} + \frac{\rho}{4} I\right)^{-1} \left(\mathcal{L}_{Q \circ q} Y_k + \frac{\rho}{4} Y_k\right), \tag{4.26}$$

where q is the matrix with elements $q_{ij} = \left(1 + ||\mathbf{y}_i - \mathbf{y}_j||^2\right)^{-1}$, the operation $\circ$

denotes the Hadamard product, Q is the matrix whose elements are defined in (4.25), and ρ is a positive scalar. Also, $\mathcal{L}_M$ is the Laplacian of the matrix M which is defined as $\mathcal{L}_M = \Lambda - M$, where Λ is a diagonal matrix with diagonal entries $\Lambda_{ii} = \sum_j M_{ij}$.

In this experiment, we used the COIL20 data to study the MM algorithm (4.26) and its acceleration using the methods described in Section 4.3. The COIL20 dataset has 1440 images of 20 objects with resolution 128×128. Each object has 72 images which were taken by capturing an image at every 5 degrees along a 360 degree viewing circle. To evaluate the different acceleration methods, we ran each method 50 times using the MM algorithm (4.26) as the base iteration. In each of the 50 runs, we initialized the elements of the embedding matrix Y by sampling from a Normal distribution with mean zero and standard deviation 10^{-3}. The value of ρ in iteration (4.26) is found by using an initial value of 10^{-5} in each iteration and using a backtracking search to maintain monotonicity. Since the parameter values are not identified in embedding problems, we guided convergence using values of the objective function. Specifically, an algorithm is terminated whenever 1000 fixed point iterations have been reached or when the relative change of the objective function is less than 10^{-4}. All other hyperparameters were set to their default values (see Section 4.4).

Results from this numerical experiment can be found in Table 4.6. We can see that SQUAREM, DAAREM, parabolic-EM, and Quasi-Newton all achieved a substantially better objective value upon convergence when compared to the original MM algorithm. Moreover, DAAREM achieved this improved objective value with an even shorter computation time than the MM algorithm. Figure 4.1 visualizes and compares the embedding results between MM and SQUAREM,



(a) MM **(b)** SQUAREM

Figure 4.1: Visualizing one run of experiments. (a) Embedding from the original MM algorithm with objective value 0.343. (b) Embedding from the SQUAREM algorithm with objective value 0.264. Different colors are used for different objects. We can see that SQUAREM, which obtained a lower value of the objective function, does provide better separation quality across different objects.

indicating that better values of the objective function achieved by SQUAREM do result in a better quality embedding.

Metric	MM	SQUAREM	DAAREM	pEM	Quasi-Newton	Nesterov
fpevals	62.3 ± 24	162 ± 84.2	$\mathbf{50 \pm 10.6}$	100 ± 45.8	114 ± 57.1	127 ± 128
elapsed	11.8 ± 4.41	22.1 ± 11.4	$\mathbf{9.6 \pm 1.98}$	18.1 ± 8.4	17.9 ± 9.17	26.9 ± 27.2
objval	0.328 ± 0.022	$\mathbf{0.279 \pm 0.017}$	0.289 ± 0.024	0.309 ± 0.016	0.299 ± 0.023	0.347 ± 0.054

Table 4.6: Experiment results for $t-$SNE from 50 independent runs. *MM* represents the original MM algorithm, and the other columns are different acceleration versions of it. *objval* is the final Kullback–Leibler divergence obtained by the acceleration method. Smaller values of *objval* correspond to better embeddings.

4.6 Discussion

In Section 4.5, we tested all the acceleration schemes described in Section 4.3 on six different practical problems. Although no single method is guaranteed to always work, at least one of SQUAREM and DAAREM effectively accelerated the original algorithm in every setting of the six examples. Moreover, SQUAREM

is particularly robust in providing improved performance in every scenario of our numerical experiments. Nesterov algorithm is most popular for accelerating proximal gradient. Here we have shown that SQUAREM and DAAREM are competitive when compared to Nesterov and hence deserving of greater attention.

We summarize the results from the six main simulation studies. SQUAREM effectively accelerated the original algorithm in 5 of the 6 problems, giving up to 78 fold improvement in elapsed time with a mean reduction of roughly 18 fold. In the *t*-SNE problem, SQUAREM did not accelerate the convergence of the original algorithm, but it did consistently converge to a better solution than the original MM algorithm. Nesterov with restarts accelerates convergence in 4 of the 6 problems, with a speedup of up to 68 fold and a mean of 16 fold improvement. DAAREM can also accelerate 4 of the 6 tasks, gaining a factor of up to 48 fold improvement with a mean of a 13 fold reduction in computation time. Quasi-Newton also accelerates 4 of the 6 problems with mean of a 13 fold improvement in convergence time. Lastly, *Parabolic-EM* accelerated convergence in 4 of the 6 tasks with a mean 4 fold improvement in convergence. Since the relative performance of the acceleration methods can vary across specific problems, we suggest trying SQUAREM, DAAREM, and Nesterov with restarts when acceleration is needed for a specific problem. We refer the reader to the our `AccelBenchmark` Github package, which allows the user to easily identify the **best** acceleration scheme for their specific problem. To use `AccelBenchmark`, the user only needs to supply the data, fixed-point mapping function, and a loss function if one is available, and the package will then automatically benchmark the performance of the original algorithm and all acceleration methods. For details, one can consult the package vignette.

We would like to draw attention to the importance of monotonicity control implemented in all our acceleration algorithms. The base algorithms such as EM and MM are intrinsically monotone. Hence they have guaranteed global convergence. Acceleration schemes are based on extrapolation methods. They are fast in the neighborhood of the solution. However, they are non-monotone and are not globally convergent. Therefore, we implement safeguards such as controlling the steplength and ensuring monotonicity to ensure reliable convergence from any starting value (note that in the default settings monotonicity is relaxed as the solution is approached). Performance is not guaranteed without such safeguards. More importantly, adding the safeguards seldom degrades the speed of convergence, while providing a better guarantee of convergence that is essential for general purpose implementation.

It is worth noting that the fixed-point iterations used in some of the examples discussed here are already faster versions of the original fixed-point schemes. Specifically, the MMF procedure for the multivariate-t distribution is a faster version of the original EM algorithm; proximal gradient is a faster version of the subgradient method for the LASSO problem; and the MM algorithm for t-SNE is a faster procedure than gradient descent. Nevertheless, as we have demonstrated in our simulation studies, we can further accelerate these faster schemes using the described acceleration schemes and achieve more significant speedups in many cases. Therefore, the acceleration schemes listed here are worth trying even in problems where there is a relatively fast fixed-point iterative algorithm already available.

We have shown the described acceleration methods can be very effective in a

wide range of applications where fixed-point iteration algorithms are used. However, the use of these or similar acceleration methods in other contexts are still lacking in development. For example, adapting these methods to handle nondeterministic iterations such as stochastic gradient descent or Markov chain Monte Carlo procedures would be an interesting topic for future research. Adapting these acceleration techniques to infinite-dimensional parameter settings (e.g., solutions of integral equations (Atkinson, 1976)) would also be an interesting direction for future study. The main issue here is that the parameter dimension can change across iterations, which may require the introduction of operators like an inner product in a certain Hilbert space to compute the acceleration method updates. For example, the Picard iteration (Junkins et al., 2013) and gradient tree boosting (Friedman, 2001) involve update functions where the number of parameters increases across iterations.

References

Dempster, Arthur P, Nan M Laird, and Donald B Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22.

Hunter, David R and Kenneth Lange (2004). "A tutorial on MM algorithms". In: *The American Statistician* 58.1, pp. 30–37.

Boyd, Stephen, Stephen P Boyd, and Lieven Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.

Varadhan, Ravi and Christophe Roland (2008). "Simple and globally convergent methods for accelerating the convergence of any EM algorithm". In: *Scandinavian Journal of Statistics* 35.2, pp. 335–353.

Henderson, Nicholas C and Ravi Varadhan (2019). "Damped Anderson acceleration with restarts and monotonicity control for accelerating EM and EM-like algorithms". In: *Journal of Computational and Graphical Statistics* 28.4, pp. 834–846.

Zhou, Hua, David Alexander, and Kenneth Lange (2011). "A quasi-Newton acceleration for high-dimensional optimization algorithms". In: *Statistics and Computing* 21.2, pp. 261–273.

O'donoghue, Brendan and Emmanuel Candes (2015). "Adaptive restart for accelerated gradient schemes". In: *Foundations of Computational Mathematics* 15.3, pp. 715–732.

Berlinet, Alain and Christophe Roland (2009). "Parabolic acceleration of the EM algorithm". In: *Statistics and Computing* 19.1, pp. 35–47.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Anderson, Donald G (1965). "Iterative procedures for nonlinear integral equations". In: *Journal of the ACM (JACM)* 12.4, pp. 547–560.

Higham, Nicholas J and Nataša Strabić (2016). "Anderson acceleration of the alternating projections method for computing the nearest correlation matrix". In: *Numerical Algorithms* 72.4, pp. 1021–1042.

Geist, Matthieu and Bruno Scherrer (2018). "Anderson acceleration for reinforcement learning". In: *arXiv preprint arXiv:1809.09501*.

Fang, Haw-ren and Yousef Saad (2009). "Two classes of multisecant methods for nonlinear acceleration". In: *Numerical Linear Algebra with Applications* 16.3, pp. 197–221.

Walker, Homer F and Peng Ni (2011). "Anderson acceleration for fixed-point iterations". In: *SIAM Journal on Numerical Analysis* 49.4, pp. 1715–1735.

Evans, Claire, Sara Pollock, Leo G Rebholz, and Mengying Xiao (2020). "A proof that Anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically)". In: *SIAM Journal on Numerical Analysis* 58.1, pp. 788–810.

Toth, Alex and C T Kelley (2015). "Convergence analysis for Anderson acceleration". In: *SIAM Journal on Numerical Analysis* 53.2, pp. 805–819.

Zhang, Junzi, Brendan O'Donoghue, and Stephen Boyd (2020). "Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations". In: *SIAM Journal on Optimization* 30.4, pp. 3170–3197.

Raydan, M and B F Svaiter (2002). "Relaxed Steepest Descent and Cauchy-Barzilai-Borwein Method". In: *Computational Optimization and Applications*, pp. 155–167.

Zhu, Xiang and Matthew Stephens (2018). "Large-scale genome-wide enrichment analyses identify new trait-associated genes and pathways across 31 human phenotypes". In: *Nature Communications* 9.1, pp. 1–14.

Raket, Lars Lau, Britta Grimme, Gregor Schöner, Christian Igel, and Bo Markussen (2016). "Separating timing, movement conditions and individual differences in the analysis of human movement". In: *PLoS Computational Biology* 12.9, e1005092.

Hobolth, Asger, Qianyun Guo, Astrid Kousholt, and Jens Ledet Jensen (2020). "A Unifying Framework and Comparison of Algorithms for Non-negative Matrix Factorisation". In: *International Statistical Review* 88.1, pp. 29–53.

Jin, Ying, Oliver H Tam, Eric Paniagua, and Molly Hammell (2015). "TEtranscripts: a package for including transposable elements in differential expression analysis of RNA-seq datasets". In: *Bioinformatics* 31.22, pp. 3593–3599.

Shiraishi, Yuichi, Georg Tremmel, Satoru Miyano, and Matthew Stephens (2015). "A simple model-based approach to inferring and visualizing cancer mutation signatures". In: *PLoS genetics* 11.12, e1005657.

Song, Junxiao, Prabhu Babu, and Daniel P Palomar (2016). "Sequence set design with good correlation properties via majorization-minimization". In: *IEEE Transactions on Signal Processing* 64.11, pp. 2866–2879.

Varadhan, Ravi and Ch Roland (2004). "Squared extrapolation methods (SQUAREM): A new class of simple and efficient numerical schemes for accelerating the convergence of the EM algorithm". In.

Nesterov, Yu (2013). "Gradient methods for minimizing composite functions". In: *Mathematical Programming* 140.1, pp. 125–161.

Tseng, Paul (2009). "On accelerated proximal gradient methods for convex-concave optimization. 2008". In: *URL http://www. math. washington. edu/~ tseng/papers/apgm. pdf*.

Du, Yu and Ravi Varadhan (2020). "SQUAREM: An R Package for Off-the-Shelf Acceleration of EM, MM and Other EM-Like Monotone Algorithms". In: *Journal of Statistical Software* 92, pp. 1–41.

Henderson, Nicholas and Ravi Varadhan (2020). *daarem: Damped Anderson Acceleration with Epsilon Monotonicity for Accelerating EM-Like Monotone Algorithms*. URL: https://doi.org/10.1080/10618600.2019.1594835.

Bobb, Jennifer F. and Ravi Varadhan (2021). *turboEM: A Suite of Convergence Acceleration Schemes for EM, MM and Other Fixed-Point Algorithms*. URL: https://CRAN.R-project.org/package=turboEM.

Hasannasab, Marzieh, Johannes Hertrich, Friederike Laus, and Gabriele Steidl (2021). "Alternatives to the EM algorithm for ML estimation of location, scatter matrix, and degree of freedom of the Student t distribution". In: *Numerical Algorithms* 87.1, pp. 77–118.

Liu, C and D B Rubin (1995). "ML estimation of the multivariate t distribution with unknown degrees of freedom". In: *Statistica Sinica* 5, pp. 19–39.

Hasselblad, Victor (1966). "Estimation of parameters for a mixture of normal distributions". In: *Technometrics* 8.3, pp. 431–444.

Guyon, Isabelle, Steve R Gunn, Asa Ben-Hur, and Gideon Dror (2004). "Result Analysis of the NIPS 2003 Feature Selection Challenge." In: *NIPS*. Vol. 4, pp. 545–552.

George, Edward I and Robert E McCulloch (1997). "Approaches for Bayesian variable selection". In: *Statistica Sinica*, pp. 339–373.

Chipman, Hugh, Edward I George, Robert E McCulloch, Merlise Clyde, Dean P Foster, and Robert A Stine (2001). "The practical implementation of Bayesian model selection". In: *Lecture Notes-Monograph Series*, pp. 65–134.

Bottolo, Leonard, Sylvia Richardson, et al. (2010). "Evolutionary stochastic search for Bayesian model exploration". In: *Bayesian Analysis* 5.3, pp. 583–618.

Clyde, Merlise A, Joyee Ghosh, and Michael L Littman (2011). "Bayesian adaptive sampling for variable selection and model averaging". In: *Journal of Computational and Graphical Statistics* 20.1, pp. 80–101.

Carbonetto, Peter, Matthew Stephens, et al. (2012). "Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies". In: *Bayesian Analysis* 7.1, pp. 73–108.

Knight, Philip A (2008). "The Sinkhorn–Knopp algorithm: convergence and applications". In: *SIAM Journal on Matrix Analysis and Applications* 30.1, pp. 261–275.

Mena, Gonzalo, David Belanger, Scott Linderman, and Jasper Snoek (2018). "Learning latent permutations with gumbel-sinkhorn networks". In: *arXiv preprint arXiv:1802.08665*.

Altschuler, Jason, Jonathan Weed, and Philippe Rigollet (2017). "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration". In: *arXiv preprint arXiv:1705.09634*.

Sinkhorn, Richard and Paul Knopp (1967). "Concerning nonnegative matrices and doubly stochastic matrices". In: *Pacific Journal of Mathematics* 21.2, pp. 343–348.

Parlett, BN and T L Landis (1982). "Methods for scaling to doubly stochastic form". In: *Linear Algebra and its Applications* 48, pp. 53–79.

Cook, James, Ilya Sutskever, Andriy Mnih, and Geoffrey Hinton (2007). "Visualizing Similarity Data with a Mixture of Maps". In: *Artificial Intelligence and Statistics*. PMLR, pp. 67–74.

Maaten, Laurens Van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE." In: *Journal of Machine Learning Research* 9.11.

Yang, Zhirong, Jaakko Peltonen, and Samuel Kaski (2015). "Majorization-Minimization for Manifold Embedding". In: *Artificial Intelligence and Statistics*. PMLR, pp. 1088–1097.

Atkinson, Kendall E (1976). *A Survey of Numerical Methods for the Solution of Fredholm Integral Equations of the Second Kind*. Vol. 16. Society for Industrial and Applied Mathematics Philadelphia.

Junkins, John L, Ahmad Bani Younes, Robyn M Woollands, and Xiaoli Bai (2013). "Picard iteration, chebyshev polynomials and chebyshev-picard methods: Application in astrodynamics". In: *The Journal of the Astronautical Sciences* 60.3, pp. 623–653.

Friedman, Jerome H (2001). "Greedy Function Approximation: a Gradient Boosting Machine". In: *Annals of Statistics*, pp. 1189–1232.

Chapter 5

Conclusion

It is shown in this dissertation that distributional regression can be used as a new framework for analysing functional connectivity. It has promising applications in fields like functional MRI when one wants to consider effects that are not geometrically localized. In such cases, by considering connections to be exchangeable, we could get a drastic reduction of the multiplicity, which is of great importance in connectivity analysis. We also show that transformations can be the key to enhance the regression fit. The recommended approach in the work of Chapter 2 uses a log quantile density as the functional predictor, rather than the density, distribution function or quantile function directly. It has been shown that it conveys practical benefit of focusing attention on tail behavior, where effects are likely to be seen. Utilizing the quantile density also creates robustness to irrelevant foci pairs being included in this kind of analysis.

However, we should emphasis that by making the exchangeability assumption, one could ignore potentially useful localization information that possibly exists and may reduce power and sensitivity. Obviously, an optimal strategy removes subject-specific artifacts and reduces the search space with - correct - strong a

priori hypotheses and then tests only those. However, in the absence of this ideal case, one is often confronted with a massive unstructured search problem with localization analyses. In contrast, distribution regression is more akin to an omnibus F-test, looking over a large range of edges, dramatically mitigating multiple comparisons issues in the favor of testing one overview hypothesis, rather than a large collection of highly specific ones. Therefore, we suggest the method as an early stage tool in a neuroimaging data analyst's toolbox.

The resulting methodology raises many avenues for future research. For example, one could consider non-localized effects in dynamic connectivity via stochastic processes of connectivity densities (by time). In addition, there are multiple alternatives for densities estimated from correlation of each region pair for contralateral regions. Here, it should be acknowledged that there is strong homotopic correlations from symmetric regions. One should then deal with multivariate densities estimated from pairs of correlations.

In Chapter 3, our theoretical results also suggest that estimating individual densities is not required if no further transformation on the distribution is required. Optimal and closed form inference can be performed fully non-parametrically with only observed samples if one puts a Gaussian process prior on the regression function. The posterior estimation error would contract in the same rate, in terms of the function of the number of subjects, as that of typical Gaussian process regression with vector predictors if our distribution process follow certain richness conditions. For example we show that Dirichlet process with any parameter falls into that category. However, the family of processes satisfying the condition is rather modest. In addition, we can at most access the optimal prediction error bound for other processes. For future research directions, one could extend the

linear expectation model to the generalized linear case by adding a link function to relate the expectation term and the means of outcomes. Also, one could study the nonparametric single index model extension such that $y = g(\mathbb{E}_x f)$ with some general function to be estimated. This could drastically increase the model capacity by expand the possible functions (which should be permutation invariant for the samples) that we can express.

In both of these chapters, a stark line is created between purely exchangeable settings and purely localized settings. Another potential avenue for future research is semi-localized settings. For example, the assumption that voxels near to one and another are more likely to have stronger connectivity than those further apart, i.e. the universal law of geography, would be a kind of semi-localized hypothesis. Other aspects of connectivity one could consider is the previously mentioned homotopic correlations and network or functional group membership. These could be implemented with variations on density regression or Gaussian process regression, yet would not require a common functional or structural geometry across subjects.

In Chapter 4, we evaluate several acceleration schemes on six separate practical problems. Although no single method is guaranteed to always work, at least one of SQUAREM and DAAREM effectively accelerated the original algorithm in every one of the six settings. Moreover, SQUAREM is particularly robust in providing improved performance in every scenario of our numerical experiments. We also show that the modified restart-Nesterov algorithm can be used to accelerate the general fixed-point iteration problem, where we improve the performance of 4 out of 6 settings and get a mean 16-fold speed up. Since the relative performance of the acceleration methods can vary across specific problems, we suggest trying

SQUAREM, DAAREM, and Nesterov with restarts when acceleration is needed for a specific problem. We refer the reader to the our `AccelBenchmark` Github package, which allows the user to easily identify the **best** acceleration scheme for their specific problem.

It is worth noting that the fixed-point iterations used in some of the examples discussed in Chapter 4 are already faster versions of the original fixed-point schemes. Specifically, the MMF procedure for the multivariate-t distribution is a faster version of the original EM algorithm; proximal gradient is a faster version of the subgradient method for the LASSO problem; and the MM algorithm for t-SNE is a faster procedure than gradient descent. Nevertheless, as we have demonstrated in our simulation studies, we can further accelerate these faster schemes or get better results using the described acceleration schemes and achieve practically significant speedups in many cases. Therefore, the acceleration schemes listed here are worth trying even, in problems where there is a relatively fast fixed-point iterative algorithm already available.

We have shown the described acceleration methods can be very effective in a wide range of applications where fixed-point iteration algorithms are used. However, the use of these or similar acceleration methods in other contexts are still lacking in development. For example, adapting these methods to handle nondeterministic iterations, such as stochastic gradient descent or Markov chain Monte Carlo procedures, is an interesting topic for future research. Adapting these acceleration techniques to infinite-dimensional parameter settings (e.g., solutions of integral equations) would also be an interesting direction for future study. A main issue in that case is that the parameter dimension can change across iterations, which may require the introduction of operators, like a Hilbert space inner product,

to compute the acceleration method updates. For example, the Picard iterations (Junkins et al., 2013) and gradient tree boosting (Friedman, 2001) involve update functions where the number of parameters increases across iterations.

Bohao Tang

EDUCATION

Johns Hopkins University — Baltimore, US
Ph.D. in Biostatistics — Expected 01/2023

Johns Hopkins University — Baltimore, US
MS.E. in Computer Science — 02/2019 - 11/2021

Fudan University — Shanghai, China
B.S. in Pure and Applied Mathematics — 09/2013 - 07/2017

PROFESSIONAL EXPERIENCE

Two Sigma Investments — New York, NY
Quantitative Researcher Intern — 05/2022 - 08/2022

Amazon.com, Inc — Seattle, WA
Applied Scientist Intern - Alexa Local Search — 06/2020 - 08/2020
Applied Scientist Intern - Payment Group — 05/2019 - 08/2019

Johns Hopkins University — Baltimore, MD
Teaching Assistant — 08/2018 - present

RESEARCH

- **Tang, B.**, …, Caffo, B. S., & Ewen, J. (In Progress). Causal Psychological Models in ASD: A Study of Communication Theories of Autism Using a Large Sample.
- **Tang, B.**, Zhao, Y., Datta, A. & Caffo, B. S. (In Progress). Information Rates of Bayesian Distributional Regression
- **Tang, B.**, Varadhan, R., Tsai, H. L., & Imus, P. (In Progress). Predicting Survival After Bone Marrow Transplant Using Time Series of Routine Laboratory Biomarkers.
- **Tang, B.**, Zhao, Y., … & Caffo, B. (2022). Differences in functional connectivity distribution after transcranial direct-current stimulation: a connectivity density point of view. Human Brain Mapping. doi: 10.1002/hbm.26112. **Presented in ENAR 2022: Distributional Data Analysis section.**
- **Tang, B.,** Henderson, N. C., & Varadhan, R. (2022). Accelerating Fixed-Point Algorithms in Statistics and Data Science: A State-of-Art Review. Journal of Data Science, 1-26. doi:10.6339/22-JDS1051. **Presented in NESS 2022.**
- Sheppard, S., … **Tang, B.**, ... & Hillis, A. E. (2022). Neural correlates of syntactic comprehension: A longitudinal study. Brain and Language, 225, 105068.
- Huang, Q., **Tang, B.**, ... & Gracias, D. H. (2022). Shell microelectrode arrays (MEAs) for brain organoids. *Science Advances*, *8*(33), eabq5031.
- Chen, X., **Tang, B.**, Fan, J., & Guo, X. (2022). Online gradient descent algorithms for functional data learning. Journal of Complexity, 70, 101635.
- Gill, R. E., **Tang, B.**, ... & Ewen, J. B. (2021). Quantitative EEG improves prediction of Sturge-Weber syndrome in infants with port-wine birthmark. Clinical Neurophysiology, 132(10), 2440-2446.
- Choe, A. S., **Tang, B.**, … & Pekar, J. J. (2021). Phase-locking of resting-state brain networks with the gastric basal electrical rhythm. PloS one, 16(1), e0244756.
- Choe, A. S., **Tang, B.**, … & Pekar, J. J. (2021). Methodological considerations in analyzing synchronization of resting-state brain networks with the intrinsic electrical rhythm of the stomach: Advantages of weighted phase-locking. bioRxiv.
- Akshintala, V. S., **Tang, B.**, ... & Khashab, M. A. (2019). Sa1470 Risk Estimation, Machine Learning Based ERCP Decision-Making Tool for Suspected Choledocholithiasis. Gastrointestinal Endoscopy, 89(6), AB246-AB247.