Title

Bayesian Analysis in Problems with High Dimensional Data and Complex Dependence Structure

Author
Lee, Wayne Tai

Publication Date
2013

Peer reviewed|Thesis/dissertation

Bayesian Analysis in Problems with High Dimensional Data and Complex Dependence Structure

By

Wayne Tai Lee

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Assistant Professor Cari G. Kaufman, Chair

Associate Professor Haiyan Huang

Associate Professor John C. Chiang

Spring 2013

Bayesian Analysis in Problems with High Dimensional Data and Complex Dependence Structure

by

Wayne Tai Lee

Abstract

Bayesian Analysis in Problems with High Dimensional Data and Complex
Dependence Structure

by

Wayne Tai Lee

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Cari G. Kaufman, Chair

This dissertation is a compilation of three different applied statistical problems from the Bayesian perspective. Although the statistical question in each problem is different, a common challenge is the high dimensionality of the data and the complex dependence structure. These introduce challenges with standard statistical techniques and computational issues. For each problem, we address the statistical problem and resolve the computational issues in the implementation.

The first topic considers the problem of Bayesian inference for the location of the global extreme of a nonparametric regression function given noisy observations. We model the unknown function using a Gaussian Process (GP) prior. The unknown function may be high dimensional and sampling posterior realizations of the function can be computationally intensive. We introduce a novel algorithm that makes use of existing optimization routines to simultaneously sample and optimize the GP realizations in an efficient manner. We demonstrate our method on a spatial data sets with non-Gaussian observations as well as an application in astronomy in which the location of the extreme varies temporally.

The second topic constructs a Bayesian Hierarchical Model for surface wind fields over the globe. Surface winds are intrinsically multivariate with spatially heteroscedastic behavior over the globe. Our model is the first to model wind fields at the global scale over land and sea. Motivated by the geostrophic relationship, we fit a varying coefficient model to model wind fields using the pressure gradient. We apply our method on surface wind and sea level pressure products from a general circulation model. We will show that our model can produce realistic wind fields that resemble the wind fields from the climate model.

The third topic considers the problem of hierarchical multilabel classification (HMC) given existing single label classifier outputs. In our problem setting, multiple labels can be assigned to each subject and the assignments have to respect a given hierarchy. We want to utilize the existing local classifiers to give assignments consistent with the hierarchy. We rank each label assignment under a Bayesian framework by its probability of being positive given all local classifier outputs. We use this ranking to sequentially assign labels according to a cutoff. However, we also update the ranking after each assignment to ensure consistency. Our algorithm outperforms existing HMC methods in various simulation studies and on a disease diagnosis dataset with a large hierarchy with few independent observations.

I dedicate this dissertation to my family and people who enjoy statistics.

Contents

List of Figures

List of Tables

Acknowledgements:

I thank all the collaborators for their invaluable guidance and feedback.

Some portion of this work is based upon work supported by the National Science Foundation under Award No. DMS-0636667. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Some portion of this work was supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

CHAPTER 1

Introduction:

This dissertation is a compilation of three different applied statistical problems from the Bayesian perspective. Although the statistical question in each problem is different, the common challenge is the high dimensionality of the data and the complex dependence structure. These introduce challenges with standard statistical techniques along with various computational issues. For each problem, we address the statistical problem and resolve the computational issues in the implementation. Overall, each chapter is a self-contained topic that covers the respective motivations, methods, examples, and discussion.

This chapter first clarifies what is meant by the Bayesian perspective in this dissertation. Then we introduce common concepts and techniques used for inference in this dissertation. The chapter ends with an overview for each topic.

1.1. Bayesian Inference

In parametric statistics, the data Y is often assumed to be random variable from a distribution F governed by certain parameters θ. We then denote this as

$$Y \sim F_\theta$$

A classic example is the number of heads from a single coin toss where the chance of landing a head is unknown. The data is then $Y \in \{0, 1\}$ where F_θ is a Bernoulli distribution and $\theta \in [0, 1]$ is the chance the coin lands a head.

The difference between the Frequentist approach and the Bayesian approach comes from the interpretation of probability and the inference on θ. From the Frequentist perspective, the long-run frequency is the probability of an event. For the same coin, θ is an unknown fixed parameter and is not random in the stochastic sense. Therefore $P(\theta = a) = 0$ for any a except the true value. From the Bayesian perspective, however, uncertainty can be treated as a source of probability. The lack of knowledge about θ allows us to treat it as a random variable and attach probabilities to its possible values. Our lack of knowledge, however, is not always uninformative. For example, since most coins are made roughly symmetric and uniform in density, our prior belief about $P(\theta = a)$ is likely highest for $a = 0.5$ than any other value. Before observing any data, we can reflect our prior belief about the differnet possible values for θ using a $Beta(b_1, b_2)$ distribution where $b_1 = b_2 = 2$. This distribution is called the prior distribution for θ. Overall, the Bayesian probability reflects a *degree of belief* rather than the asymptotic frequency interpretation for Frequentist probabilities.

This difference allows Bayesians to place probability distributions on unknown but fixed quantities and update our beliefs as we observe data. After observing data, Y, our belief is updated according to Bayes rule

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{P(Y)} \tag{1.1.1}$$

where $P(Y|\theta)$ is called the likelihood and describes the chance of seeing the data with parameter θ, $P(\theta)$ is the prior and reflects our prior belief about θ before observing data, and lastly $P(Y) = \int P(Y|\theta)P(\theta)d\theta$ is a normalizing constant that ensures $P(\theta|Y)$ is a probability distribution. $P(\theta|Y)$ is called the posterior distribution which is our updated belief about θ after observing Y. With no data, Bayesian inference is completely based on the prior. However, even with limited data, Bayesian inference formally reflects the change in uncertainty in the posterior distribution after updating the prior. Frequentist analysis on the other hand often require ad hoc assumptions to perform meaningful analysis with limited data.

Overall, in this dissertation, the Baysian approach is appropriate since the goal is often to portray the uncertainty in an unknown but possibly fixed quantity. Chapter 2 performs inference on the global extreme for an unknown function, Chapter 3 characterizes the wind outputs from a deterministic numeric model, and Chapter 4 treats the unknown disease status for a patient as a random variable. Again, the probability here reflect the degree of belief and should not be mixed with the Frequentist probability interpretation. In the follow sections we lay out some useful concepts for the later chapters but do not attempt to cover the details of Bayesian analysis. *Gelman et al.* (2004) offers a good overview of practical Bayesian data analysis.

1.2. Modeling unknown functions through Gaussian process

For the problems in this dissertation, our data sometimes depend on an unknown continuous function. For example, we can imagine scallop catches to depend on the unknown scallop abundance (Chapter 2) and wind fields to depend on an underlying climate process (Chapter 3). In these cases, a reasonable prior distribution on continuous functions is the Gaussian process (GP). A GP, Z, is a stochastic process over domain X where for any finite locations $\{x_1, \ldots, x_n\} \subset X$

$$[Z(x_1), \ldots, Z(x_n)] \sim MVN\,(\tilde{m}\ ,\ \Sigma) \tag{1.2.1}$$

where MVN is the multivariate Gaussian distribution with mean vector $\tilde{m}^T = (m(x_1;\theta), \ldots m(x_n;\theta))$ and covariance matrix with entries $\Sigma_{ij} = K(x_i, x_j; \theta)$ for $i, j \in \{1, \ldots, n\}$. Here $m(\cdot;\theta)$ is the mean function and $K(\cdot,\cdot;\theta)$ is the covariance function that governs Z. We usually denoted GPs as

$$Z \sim GP\,(\ m\,(\cdot;\theta)\ \ ,\ K\,(\cdot,\cdot;\theta)\ )$$

where θ are parameters that govern the mean and covariance functions. The mean function dictates the placement of the realizations where the covariance function dictates the dependence between locations. The stronger the dependence, the smoother the functions implied by the GP. We show some realizations from a one dimension GP over a fine grid with different mean functions and covariance functions in Figure 1.2.1.

GPs are attractive because they define a distribution over an infinite collection of functions with the specification of the mean and covariance function. With these two functions, GPs can capture a wide range of continuous functions. Different

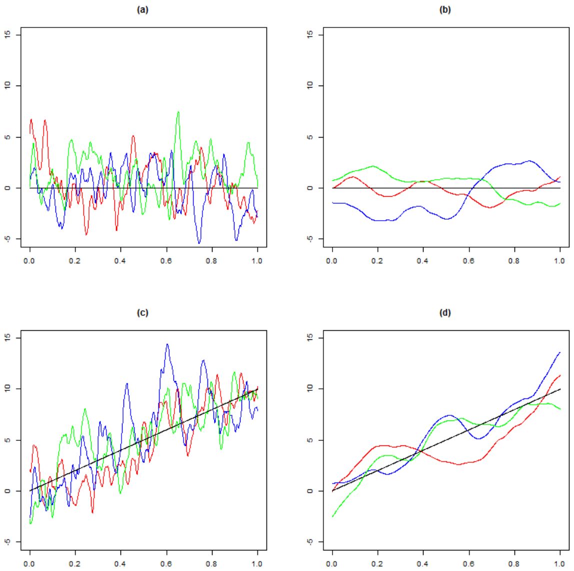

FIGURE 1.2.1. Independent GP Realizations under different mean and covariance functions over a fine grid where $x \in [0, 1]$. The black line indicates the mean functions where the colored lines are independent realizations over a fine grid. (a) and (b) have a constant mean function $m(x) = 0$ where (c) and (d) have mean function $m(x) = 10x$. (a) and (c) have a Matern covariance function with smoothness $\nu = 2$ and range $\rho = 0.01$ where (b) and (d) have a Matern covariance with smoothness $\nu = 2$ and range $\rho = 0.1$. A larger range parameter indicates distant values are more correlated to each other and therefore produces smoother curves.

types of covariance functions are discussed in *Gelfand et al.* (2010) but a popular choice is the Matern covariance family defined in Equation 2.2.4 in Chapter 2. The Matern covariance functions allow us to specify the degrees of differentiability of the covariance function which is often useful. We do not dive into the details for the Matern covariance family but details on the desirable properties of the Matern covariances are discussed in *Stein* (1999).

Now that we can describe our uncertainty over smooth functions with GP priors, we can also easily update our belief of possible functions when measurements are collected. As a consequence of the definition in Equation 1.2.1, for any finite collection of locations $x'_1, \dots, x'_k$

$$[Z(x_1), \dots, Z(x_n)]\,|Z(x'_1), \dots, Z(x'_k) \sim MVN\left(\tilde{m'}_k\ ,\ \Sigma'_k\right)$$

In other words, the conditional distribution when the measurements are exact (noise-free) remains Gaussian. The specific form for $\tilde{m}'$ and Σ' will be explained in Equation 2.2.3 in Chapter 2. Another special case that retains the Gaussian conditional distribution is when the measurement error follows a Gaussian distribution.

Figure 1.2.2 shows posterior realizations over a fine grid after making observations with and without Gaussian error for a fixed underlying function.

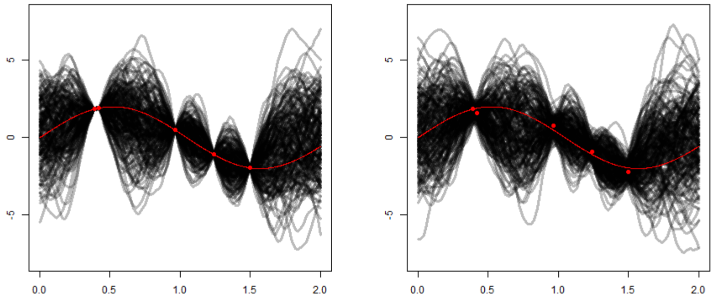

FIGURE 1.2.2. GP realizations over a fine grid after observing data. The red line denotes the true underlying function f where $f(x) = 2\sin(3x)$ for $x \in [0, 2]$ and the red dots are the observations. The left figure shows possible realizations of the GP over a fine grid when observations are made without error. The right figure shows possible realizations of the GP over a fine grid when observations are made with Gaussian error with a small variance. The possible realizations are tighter around the locations with observations. When there is no measurement error, the realizations pass through the data points.

Error-free observations are common when the unknown function is deterministic as is the case for many optimization problems or simulations from a numeric model. In this case, we expect the posterior realizations to pass through the observed data which matches the outcome in Figure 1.2.2. The topics in this dissertation have more complex cases. Each chapter describes the specific updates performed that will not be elaborated here. For further details, *Rasmussen and Williams* (2006) provides a good introduction and reviews various applications with GPs.

1.3. Inferring unknown status through local false discovery rate

Instead of modeling unknown functions, Chapter 4 handles the unknown status for a label. For example, the status for cancer is fixed but unknown before rigorous testing. The Bayesian approach allows us to attach uncertainties to the status, Q. Chapter 4 tackles this in the context of hierarhical multilabel classification but here we introduce some concepts for binary classification from the Bayesian perspective.

In the most basic classification setting, the goal is to assign a label $\hat{Q}_i \in \{0, 1\}$ based on a single feature $S_i \in \mathbb{R}$ for $i = 1, \ldots, m$ where m is the total number of label assignments possible. We assume the data S follows a mixture distribution

$$S_i | Q_i \sim (1 - Q_i)F_0 + Q_i F_1$$

where F_0 and F_1 are the respective distributions for S when $Q = 0$ and $Q = 1$. Under the Bayesian framework, a suitable classification quantity would be

$$\begin{aligned} P(Q_i = 1|S_i) &= \frac{P(S_i|Q_i = 1)P(Q_i = 1)}{P(S_i)} = \frac{\pi_1 f_1(S_i)}{f(S_i)} \\ &= \frac{\pi_1 f_1(S_i)}{\pi_1(1 - f_1(S_i)) + (1 - \pi_1)(1 - f_0(S_i))} \end{aligned}$$

where $\pi_1 = P(Q_i = 1)$ and f_1, f_0 respectively denote the density for S when $Q_i = 1$ and $Q_i = 0$. However, despite being a Bayesian quantity, the estimation for this term sometimes take an empirical approach without specifying priors for π_1. $1 - \pi_1$ is often estimated from the empirical distribution where f and f_0 are estimated through kernel density estimators. This occurs most commonly in the multiple hypothesis testing framework because $P(Q_i = 1|S_i)$ is equivalent to the complement of local false discovery rate ($lfdr$) studied in *Efron and Tibshirani* (2002); *Efron* (2010). This empirical estimation approach is mostly because $lfdr$ is derived from the False Discovery Rate (*Benjamini and Hochberg*, 1995), a Frequentist concept used to control the rate of false positives in multiple hypothesis testing. We do not elaborate on this connection here and refer the interested readers to *Efron and Tibshirani* (2002). This approach is also known as the empirical Bayes approach.

In other words, we use the Bayesian interpretation but do not implement the full Bayesian analysis through the posterior distribution. In Chapter 4, we estimate $P(Q_i = 1|S_i)$ from a third approach motivated by maximizing the pooled precision rate as shown in *Jiang et al.* (2013). However, we will show how the Bayesian framework will naturally extend this simple quantity to our sequential classifier in the hierarchical setting.

1.4. Sampling from the Posterior Distribution through Metropolis-Hastings Algorithm

As mentioned before, the full Bayesian approach derives the posterior distribution to reflect our uncertainties after observing data. However, obtaining the posterior distribution is arguably the most difficult task with Bayesian analysis. The difficulty mostly comes from the inability to derive the normalizing constant in Equation 1.1.1. Therefore we often perform analysis through Monte Carlo methods, i.e. empirically estimating the statistic of interest using posterior samples from the posterior distribution. To draw samples from the posterior distribution, a popular method relies on the convergence of Markov chains. The Monte Carlo methods following the Markov chain sampling is often referred as Markov Chain Monte Carlo (MCMC).

To draw posterior samples, MCMC methods set the stationary distribution $\pi(\theta)$ for a Markov chain to equal the posterior distribution $P(\theta|Y)$. To do so, we set up a Markov chain to satisfy the detailed balanced equation

$$P(\theta|Y)r(\theta, \theta') = P(\theta'|Y)r(\theta', \theta) \tag{1.4.1}$$

where $r(\theta, \theta')$ is the transition probability from θ to θ' for the Markov chain. If the chain is ergodic, i.e. aperiodic and positive recurrent, we know that the chain $\{\theta_t\}$ will converge to the unique stationary distribution which is set to be $P(\theta|Y)$ in Equation 1.4.1. At a high level, aperiodicity prevents periodic patterns for the chain. Positive recurrent ensures that the chain can always return to each possible state without getting stuck at certain states or diverging. Naturally, the construction of $r(\cdot,\cdot)$ is crucial and one popular construction is provided by the Metropolis-Hastings algorithm (*Brooks et al.*, 2011).

The Metropolis-Hastings algorithm has two stages: the proposal and the acceptance. Define $r(\theta,\theta') = prop(\theta,\theta')accept(\theta,\theta')$, so the probability to transition from θ to θ' involves the probability of proposing θ' starting at θ and the probability of accepting θ' starting at θ. With regularity conditions, we can rearrange Equation 1.4.1 to become

$$\frac{accept(\theta,\theta')}{accept(\theta',\theta)} = \frac{P(\theta'|Y)prop(\theta',\theta)}{P(\theta|Y)prop(\theta,\theta')}$$

This equation can be satisfied if we set the acceptance distribution as $accept(\theta,\theta') = \min\left(1, \frac{P(\theta'|Y)prop(\theta',\theta)}{P(\theta|Y)prop(\theta,\theta')}\right)$ and $prop(\theta,\theta')$ as a symmetric distribution that preserves the ergodic conditions. In this dissertation, we often convert our parameters to be defined over the real line so a reasonable proposal is the Gaussian distribution. Recall that we do not have the posterior distribution $P(\theta|Y)$ but the posterior is proportional to the product between the prior and likelihood, $P(\theta|Y) \propto P(\theta)P(Y|\theta)$. A consequence from the definition of the acceptance distribution then allows the normalizing constants to cancel. In the end, we only need to work with the prior and likelihood which we determine in the model specification.

For a multivariate dimensional parameter example, the algorithm then becomes

(1) Initiate the chain with θ_0 corresponding to high values of $P(\theta)P(Y|\theta)$
(2) Propose $\theta' \sim MVN(\theta_t, \Sigma)$ based on some Σ and the current θ_t
(3) Sample $U \sim Unif[0,1]$
(4) If $U < \frac{P(\theta')P(Y|\theta')}{P(\theta_t)P(Y|\theta_t)}$, set $\theta_{t+1} = \theta'$, otherwise $\theta_{t+1} = \theta_t$
(5) Increment t and repeat step 2 through step 4.

We repeat the algorithm until we collect enough samples after the chain converges. Convergence is often determined by visually examing θ_t over the iterations t to ensure there are no linear trends in the chain. After the chain converges to the stationary distribution, the sampled θ_t values are correlated draws from the desired posterior distribution. To obtain roughly independent samples, we perform thinning by only collecting every n_{th} sample from the converged chain. The value of n is usually determined by evaluating autocorrelation plots of the chain. The initiation in step 1 can be arbitrary but the convergence will be faster if we start at probable values of θ under $P(\theta|Y)$. Since the posterior is proportional to the likelihood and the prior, reasonable starting values are the maximum likelihood estimator (MLE) for θ, i.e. $\theta_{MLE} = \arg\max_\theta P(Y|\theta)$. With enough data, $P(Y|\theta)$ should dominate the posterior and our beliefs will be mostly based on the data instead of the prior.

By default, Σ is set to be a diagonal matrix where its variance is highly correlated with the acceptance rate of the chain. Large variance tends to yield lower acceptance with less correlation chains where small variances produces the opposite result. Low acceptance is not desirable because the number of different samples is small and the convergence for the chain is often slow. On the other hand, highly correlated chains are not desirable because we would need to perform more thinning to obtain roughly independent samples and the chain might take longer to converge if θ_t can only vary slightly from iteration to iteration. Some theoretical work on optimal accetance rates can be found in *Roberts and Rosenthal* (2001).

1.4.1. Adaptive Metropolis-Hastings Algorithm. In this dissertation we use a modified Metropolis-Hastings algorithm that adapts the proposal covariance structure to achieve desirable acceptance rates. Based on the algorithm specified in *Shaby and Wells* (2010), we first specifying a desired acceptance rate α_{opt}, some parameters $\gamma_0 = 1$ and $\gamma_1 = 0.8\gamma_0$, and initialize the proposal distribution $\Sigma_0 = I_d$ and $\sigma_0^2 = \frac{2.4^2}{d}$ where d is the dimension of θ. The specific value settings here are just suggestive based on the same suggestions in the technical report by *Shaby and Wells* (2010). With these specifications, for each adaptive Metropolis-Hastings step t

(1) Take k Metropolis-Hastings steps using σ_t^2 and Σ_t
(2) Calculate the empirical acceptance rate from the k steps, $\alpha_{emp_t} = \frac{\#accepted}{k}$
(3) Calculate the sample covariance matrix based on the k Metropolis-Hastings samples $\hat{\Sigma}_t = \frac{1}{k-1}\left(X_{d\times k} - \bar{X}_{(t)}\right)\left(X_{d\times k} - \bar{X}_{(t)}\right)^T$
(4) Update $\sigma_{t+1}^2 = \exp\left[\log \sigma_t^2 + \frac{\gamma_0}{t^{\gamma_1}}\left(\alpha_{emp_t} - \alpha_{opt}\right)\right]$
(5) Update $\Sigma_{t+1} = \Sigma_t + \frac{1}{t^{\gamma_1}}\left(\hat{\Sigma}_t - \Sigma_t\right)$

where the columns of $X_{d\times k}$ hold θ_i from the k standard Metropolis-Hastings steps and $\bar{X}_{(t)}$ is a matrix of the same size with the rows contain the row means of $X_{d\times k}$. This adaptive strategy inflates the proposal variance when the acceptance rate is too high and encourages the chain to explore farther values. The proposal variance shrinks when the acceptance rate is too low. The proposal covariance matrix also adapts according to the chain. If different components of θ are correlated in the posterior, then the proposal distribution should also pick up this correlation and propose correlated values for those components. In our experience, the adaptive Metropolis-Hastings algorithm works well for θ with $d < 200$ but the convergence is noticably slower for higher dimensional parameter spaces.

1.5. The Problems

Now that we have covered some concepts for Bayesian inference used in this dissertation, we quickly summarize each statistical problem here.

1.5.1. Bayesian inference for global extreme. Chapter 2 investigates efficient Bayesian inference for the location of the global extreme for an unknown function given noisy measurements. In other words, given noisy observations for an unknown function, we want the posterior distribution for the location of the global extreme. The posterior distribution is more desirable than traditional point estimates because it portrays the full uncertainty with limited data. In our example, it is possible that the posterior distribution for the location of the global extreme is multimodal with limited data. This information cannot be reflected in a typical point estimate which can be useful to researchers.

An important distinction to make is our focus is on inference for the location of the global extreme instead of developing a new optimization routine. Traditional optimization assumes there exists a fixed objective function that is expensive to evaluate so maximizing the information from each evaluation is essential. In other words, the question in optimization is often "where to search next?" for the next function evaluation or measurement. However, when no more function evaluations or measurements are possible, Bayesian inference portrays the full uncertainty in the location for the global extreme location given the limited data.

In general, we model the unknown function using a Gaussian Process prior then search for the global extreme in the posterior realizations. Sampling the location of the global extreme can be broken down into a two stage process. This involves drawing posterior sample of the unknown function then searching for the global extreme over a fine grid. However, sampling continuous functions over a fine grid is computationally demanding in multiple dimensions. Instead, we utilize existing optimization routines that dynamically sample and search for global extreme.

Several challenges arise when inferring the location of the global extreme through sampling methods. The first challenge is numerical stability. To obtain accuracy, the samples are necessarily close to one another which makes covariance matrices approach singularity in the sampling. The second is cohesively capturing the different sources of uncertainties from the noisy data and from the model for the unknown function. Most methods that use GPs in optimization fix the model parameters at the MLEs then perform sampling (*Forrester et al.*, 2006; *Villemonteix et al.*, 2008). This ignores the uncertainty in the unknown function which can alter the results significantly. We perform the full Bayesian approach that cohesively incorporate both sources of uncertainty in our inference.

1.5.2. Varying Coefficient Model for Global Surface Winds. Chapter 3 constructs a Bayesian hierarchical model for global surface wind fields. Surface winds models have important applications in wind energy and climate modeling.

Genton and Hering (2007) have argued that statistical models for wind is important for the future of wind energy. Wind energy is a viable source of sustainable electricity that can compete in the marketplace. Unfortunately, storing the energy is difficult so wind energy is only available when the wind is blowing. Moreover, the turbine plants that harvest wind energy require early notice to begin production. Therefore accurate forecasts are benefitial for efficient energy production. Overall, the intermittent nature of wind prevents energy suppliers from relying on wind energy. A good statistical model for wind can produce wind predictions that decrease the uncertainty with wind speeds and even identify potential wind farm locations (*Hering and Genton*, 2010).

Climate models, on the other hand, are complex numerical models that incorporate various physics, parameters, and approximations to help understand the climate system. However, quantifying the effects of each different specification on the output is difficult. Efforts such as the Program for Climate Model Diagnosis and Intercomparison (PCMDI) and the North American Regional Climate Change Assessment Program (NARCCAP) are responses to help understand the complex nature of these models. To quantify the uncertainty efficiently and cohesively, statistical techniques are often employed. For example, *Kaufman and Sain* (2010); *Sain et al.* (2011) have quantified the effects of downscaling for temperature and precipitation under a Bayesian framework. To implement similar analysis for surface winds, however, we need a statistical model that can capture the spatial heterscedastic behavior and intrinsically multivariate nature of surface winds.

Past work on surface wind modeling focused on wind fields strictly over the ocean or on wind records from a few weather stations. Our model is the first to model wind fields over the entire globe that includes land surfaces. Overall, we have surface wind fields at 8192 locations over 40 years for Winter and Summer. In additional to the large data size, wind fields over large regions exhibit heteroscedastic behavior that require flexible models. These models unfortunately introduce many computational challenges. We use the geostrophic relationship along with a varying coefficient model to construct a flexible surface wind model. We then utilize the Gaussian Random Markov Field methods in *Lindgren et al.* (2011) to resolve the computational issues. We will show that this model can simulate wind fields that resemble the wind fields from our data.

1.5.3. Hierarhical Multilabel Classification Through Local False Discovery Rate. Chapter 4 proposes a sequential classification method for hierarchical multilabel classification (HMC) based on a Bayesian framework. HMC is a type of classification where multiple labels can be assigned to each subject and the assignments have to respect a given hierarchy. For example, a patient may be diagnosed with multiple diseases but these diseases have to respect a hierarchical relationship – one with lung cancer must also have cancer. Our framework further focuses on the case when outputs from existing single label classifiers are given. Single label classifiers can be highly tailored to the subject matter but the collective assignments may not be consistent with respect to the hierarchy. Instead of discarding these past efforts, we want to utilize these outputs to create a consistent classifier.

Jiang et al. (2013) tackled multilabel classification under a similar situation where the single label classifiers are given but the labels were independent from one another. They developed an optimal ranking for all the label assignments where assigning the top k labels as positive would maximize the pooled precision rate. Unfortunately, the same method is unlikely produce consistent label assignments with respect to the hierarchy.

To obtain consistency, we propose a sequential classification method motivated under a Bayesian framework. More specifically, we first rank each label assignment by the probability of the label being positive given all local classifier outputs. We follow this ranking to assign labels according to a cutoff. We then update the ranking after each assignment. The updating naturally produces consistent label assignments where the ranking provides accurate classification results. Our method is more efficient than existing HMC methods without making any distributional assumptions on the single label classifier scores. We will show that our algorithm outperforms the existing HMC methods in various simulation studies and on a disease diagnosis dataset that has limited independent samples over 110 diseases.

CHAPTER 2

Efficient Bayesian Inference for Global Extreme Using Gaussian Processes

2.1. Introduction

This chapter considers the problem of Bayesian inference for the quantity $x^* = \arg\max_{x \in X} f(x)$ for a nonparametric regression function f defined on $X \subseteq R^d$ (or, equivalently, $\arg\min_{x \in X} f(x)$). Finding the location of the global maximum or minimum of f is of interest in a variety of applied problems. For example, this problem has arisen in the context of agriculture (?*Board and Modali*, 2005), public health (*Facer and Müller*, 2003), chemistry (*Box and Wilson*, 1951), and astronomy (*Williams et al.*, 1994). As another example, Figure 2.1.1 shows data on scallop catches off Long Island, New York. Given such data, one could infer the location of maximum abundance to guide future locations to target. We will see that the posterior distribution for x^* under our model is multi-modal which highlights the need for a statistical approach that can capture such features.

The problem of inferring x^* has been well studied using estimators derived from kernel estimators of f (*Müller*, 1985, 1989; *Müller et al.*, 1996; *Facer and Müller*, 2003), and these estimators can be shown to possess desirable asymptotic properties (*Müller*, 1985, 1989; *Facer and Müller*, 2003). However, we are interested in specifying a Bayesian model for f and then deriving the posterior distribution for x^*. If we are given a fixed number of noisy observations, as in the applications above, it is quite possible that the posterior distribution for x^* is widely dispersed, perhaps even multi-modal, and this kind of information is not reflected in a point estimate.

Various authors, e.g. *Jones et al.* (1998); *Booker et al.* (1999); *Forrester et al.* (2006); *Villemonteix et al.* (2008), have taken up a related problem from a Bayesian perspective: stochastic algorithms for global function optimization, in which f is observed without noise and the goal is to determine where to next evaluate f in an iterative optimization routine. Our motivating examples, and the resulting statistical concerns, are different. Rather than using Bayesian models to guide an optimization routine to converge to x^*, we instead want to accurately portray our uncertainty about x^* after observing f with error at a fixed number of locations.

From a conceptual standpoint, this problem is straightforward. One simply specifies a prior distribution for f and a likelihood for the observations, then derives the posterior for f and hence the posterior for x^*. However, achieving this in practice may be difficult. In particular, we will consider using Gaussian process (GP) models for f. As a simple example, suppose we take $Y_i = f(x_i) + \epsilon_i$ for $i = 1, \ldots, n$, with $\epsilon_1, \ldots, \epsilon_n | \tau^2 \overset{iid}{\sim} N(0, \tau^2)$ and f to have a GP prior distribution governed by parameters θ. Conditioning on θ, τ, and observations $Y_1, \ldots, Y_n$, f has another GP distribution, whose mean and covariance function may be calculated

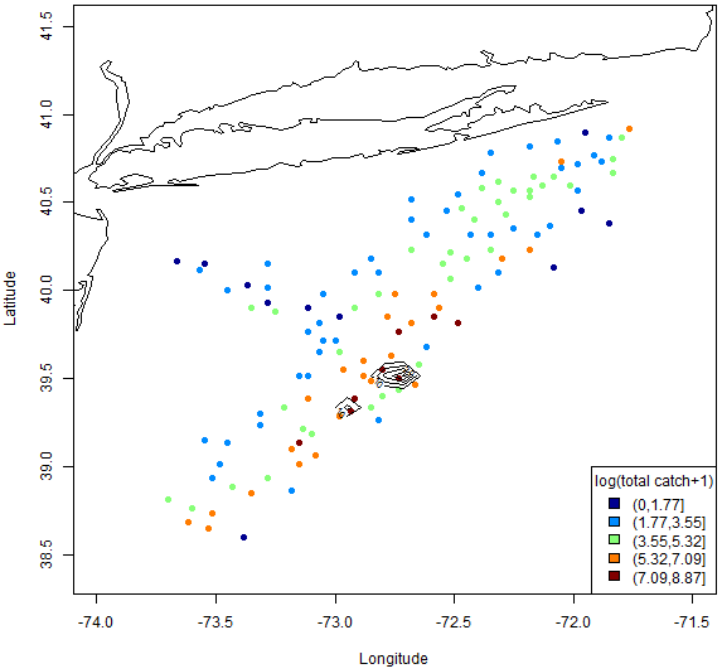

FIGURE 2.1.1. Scallop catches (log transformed) based on a 1990 survey cruise in the Atlantic continental shelf off Long Island, New York, U.S.A (*Ecker and Heltshe*, 1994). The locations are projected from longitude and latitude to UTM coordinates. The contour plot reflects the estimated posterior distribution of peak locations for high scallop abundance. There are two modes in the posterior samples which is a feature that cannot be captured by a typical point estimate for the location of global extreme.

using the well-known kriging equations (see e.g. *Stein*, 1999). Note that collecting posterior samples of x^* is different from optimizing the posterior mean of the GP. In general

$$\arg\max_x E\left(f(x)|Y_1,\ldots Y_n,\tau,\theta\right) \quad \neq \quad E\left(\arg\max_x f(x)|Y_1,\ldots Y_n,\tau,\theta\right)$$

so optimizing the GP mean function is not necessarily a reasonable point estimate for x^* under the Bayesian framework. This estimate has been used by *Simpson et al.* (1998b) in optimizing computer models since the GP mean coincides with the kriging estimate for unknown functions. Even $E\left(\arg\max_x f(x)|Y_1,\ldots Y_n,\tau,\theta\right)$ may not be sensible if the posterior is multi-modal.

In this chapter, the posterior distribution will be empirically estimated using posterior samples. To generate a posterior sample of x^*, we could draw a posterior realization of f over a fine grid over X, then choose the value that maximizes the realization (*Villemonteix et al.*, 2008). This unfortunately introduces a trade-off between efficiency and accuracy in the sampling. Increasing the resolution greatly increases the computational burden, particularly in high dimensions, while also introducing numerical stability issues due to large near-singular covariance matrices.

In this chapter we provide a novel method that efficiently samples from the posterior of x^*. This method can utilize any existing optimization routine to simultaneously sample f and search for x^*. The basic concept is to replace joint sampling of f at fixed input values $x^{(1)}, \ldots, x^{(m)}$ with a sequential sampling scheme that allows the input values $x^{(1)}, \ldots, x^{(m)}$ to be dynamically determined. We also carry out the full Bayesian nonparametric analysis for x^*, whereas most previous work has ignored the variability that comes from the estimation of θ (*Jones et al.*, 1998; *Simpson et al.*, 1998a; *Forrester et al.*, 2006; *Villemonteix et al.*, 2008).

The chapter describes the algorithm in detail, then applies it to generate samples from the posterior for x^* conditional on two very different datasets. Section 2.2 describes the algorithm with implementation details along with an illustrative example. Section 2.3 showcases the efficiency of this algorithm relative to grid searches. In Section 2.4, we apply the method to the scallop data shown in Figure 2.1.1. This example is interesting because the data itself is not Gaussian. Code for the illustration and the scallop example are available at `http://www.stat.berkeley.edu/~lwtai/Waynes_Stat_Website/Tech_Reports.html`. Our last example in Section 2.5 is on the spectrophotometric time series of the Type Ia supernova SN 2011fe (*Pereira et al.*, 2013). This dataset is interesting because the goal of the analysis is to infer x^* over multiple time points. The last section discusses possible improvements and challenges of performing the full Bayesian analysis for this problem.

2.2. Methods and Illustration:

The goal is to derive the posterior distribution $P\left(\arg\max_{x \in X} f(x) | Y_1, \ldots, Y_n\right)$. To provide the Bayesian solution, we first need to specify the data generating process. We observe $Y_1, \ldots, Y_n$ at locations $x_1, \ldots, x_n \in X$. These observations are noisy measurements of the true objective function $f(x)$ that we want to optimize. We adopt the generalized linear model (*Diggle et al.*, 2002):

$$\begin{aligned} E\left(Y_i\right) &= f(x_i) \\ x^* &= \arg\max_x f(x) \\ h(f(x)) &= Z(x) \\ Z(\cdot) &\sim GP\left(m(\ \cdot\ ;\theta) \mid k(\ \cdot\ ,\ \cdot\ ;\theta)\ \right) \end{aligned} \tag{2.2.1}$$

where the prior for f is a transformed GP with mean and covariance function $m(\cdot;\theta)$ and $k(\cdot,\cdot;\theta)$ governed by parameters θ. $h(\cdot)$ is a link function and is assumed to be invertible and monotonic. These assumptions guarantee that optimizing over $Z(.)$ is the same as optimizing over $f(.)$.

Our task is now to approximate the posterior of x^* under this model. That is, we generate samples from

$$p\left(x^* | Y_1, \ldots, Y_n\right) = \int p\left(x^* | Y_1, \ldots, Y_n, \theta\right) p\left(\theta | Y_1, \ldots, Y_n\right) d\theta$$

by first sampling $\theta^i \sim P\left(\theta | Y_1, \ldots, Y_n\right)$ for $i = 1, \ldots B$ then sampling $x_i^* \sim P\left(x^* | \theta^i, Y_1, \ldots, Y_n\right)$. The samples from $P\left(\theta | Y_1, \ldots, Y_n\right)$ can be obtained through various standard Bayesian inference techniques, but in this chapter we use Adaptive Metropolis Hastings (*Shaby and Wells*, 2010).

Our main contribution is to propose an efficient method of sampling from $P\left(x^*|\theta^i, Y_1, \ldots, Y_n\right)$. A simple approach draws $f(.) \sim P\left(f(.)|\theta^i, Y_1, \ldots, Y_n\right)$ over a fine grid $x^{(1)}, \ldots, x^{(m)}$ then reports $\arg\max_{x \in \{x^{(1)}, \ldots, x^{(m)}\}} f(x)$ (*Villemonteix et al.*, 2008). However, most locations on the fine grid are not of interest, and the resolution of the grid is typically limited. Optimization routines, on the other hand, intelligently select the evaluation locations $x^{(1)}, \ldots, x^{(m)}$, but their sequential behavior prohibits jointly sampling the posterior realization of f.

Jointly sampling the realization, however, is just a convenience to guarantee consistency of the realization of f. Basic conditional probability suggests a natural transition from jointly sampling f to sequential sampling. In particular,

$$\begin{aligned} & p\left(f(x^{(1)}), \ldots, f(x^{(m)})|\theta, Y_1, \ldots, Y_n\right) \\ = \; & p(f(x^{(1)})|\theta, Y_1, \ldots, Y_n) \ldots p(f(x^{(m)})|f(x^{(1)}), \ldots, f(x^{(m-1)}), \theta, Y_1, \ldots, Y_n) \end{aligned}$$

In other words, to sample a consistent realization at arbitrary inputs, $x^{(1)}, \ldots, x^{(m)}$, we can sample sequentially, as long as we condition on the previous function evaluations for the same realization. By carefully constructing a self-updating objective function, the optimization routine will first draw from $p(f(x^{(1)})|\theta, Y_1, \ldots, Y_n)$ then draw from $p(f(x^{(2)})|f(x^{(1)}), \theta, Y_1, \ldots, Y_n)$ when evaluating f at $x^{(2)}$. Iterating this process allows the input sequence $x^{(1)}, \ldots, x^{(m)}$ to be sequentially and dynamically determined. This results in the optimization routine simultaneously sampling the posterior realization of f and finding the corresponding x^*. This method can be easily parallelized to generate multiple samples and works well with existing optimization routines.

One challenge is then to efficiently sample from needed conditional distributions. Since optimizing $f(\cdot)$ is equivalent to optimizing $Z(\cdot)$, we will instead construct the objective function by sampling $Z(.)$ which is modeled as a GP. More specifically, at the $k+1$ evaluation, the objective function should return a sample from

$$P(Z(x^{(k+1)})|\theta, Y_1, \ldots, Y_n, Z(x^{(1)}), \ldots, Z(x^{(k)})) \tag{2.2.2}$$

where $Z(x^{(1)}), \ldots, Z(x^{(k)})$ are the previous k evaluations of the same realization of $Z(\cdot)$. The mean and covariance functions are provided by the standard kriging equations

$$\begin{aligned} Z(x')|Z(x^{(1)}), \ldots, Z(x^{(k+1)}) \quad \sim \quad & MVN(\; m(x', \theta) + \Sigma(x', \tilde{x}) \left[\Sigma(\tilde{x}, \tilde{x})\right]^{-1} \left(\tilde{Z} - \tilde{m}\right), \\ & \Sigma(x', x') - \Sigma(x', \tilde{x}) \left[\Sigma(\tilde{x}, \tilde{x})\right]^{-1} \Sigma(\tilde{x}, x')) \end{aligned} \tag{2.2.3}$$

where $\tilde{x} = \{x_1, \ldots x_n\}$, $\tilde{Z} = (Z(x^{(1)}), \ldots, Z(x^{(k)}))^T$, and $\tilde{m} = (m(x_1; \theta), \ldots, m(x_n; \theta))^T$ (*Rasmussen and Williams*, 2006).

So again the basic steps are

(1) Obtain samples $\{\theta_1, \ldots, \theta_B\} \sim P(\theta|Y_1, \ldots, Y_n)$ via Bayesian inference techniques, e.g. MCMC or variational methods.

(2) Construct the self-updating objective function that samples from the conditional distribution in equation 2.2.2 if asked to evaluate $x^{(k+1)}$, given θ.

(3) Provide a sample of θ_i to the objective function and run the optimization routine to obtain x_i^* for a realization from the GP governed by θ_i.

(4) Repeat step (3) for $\theta_j \in \{\theta_1, \dots, \theta_B\}$

(5) Discard the parameter samples$\{\theta_1, \dots, \theta_B\}$ and report $\{x_1^*, \dots, x_B^*\} \sim P\left(x^* | Y_1, \dots, Y_n\right)$

2.2.1. Prior Choice: The Bayesian methodology requires specifications for the parameter priors. Throughout this chapter, we assume a constant mean function $m(x;\theta) = \beta$ for simplicity so $Z(\cdot) \sim GP\left(\beta, \sigma^2\Sigma(\rho)\right)$. Reference priors exist for this model (*Berger et al.*, 2001) but can be slow to compute. We take $\rho \sim Uniform[0, \rho^*]$, where $\rho*$ is chosen such that the $Corr_{\rho*}(Y(x), Y(x+D_{max})) \approx .9999$ where D_{max} is the largest possible distance in the respective input space. We take $p(\sigma^2, \beta) \propto \frac{1}{\sigma^2}$. As the prior for ρ is proper, so is the posterior (*Berger et al.*, 2001). For the illustrations and examples below, unless specified otherwise, the prior specifications are all set accordingly.

2.2.2. Numerical Implementation . To capture the uncertainty from the parameters, one should sample many realizations from $P(\theta|Y_1, \dots, Y_n)$. In this paper we obtain samples mostly using the adaptive Metropolis Hastings algorithm described by *Shaby and Wells* (2010). We determine the chain has reached stationarity when the trace plots show no obvious trends. After discarding the burn-in samples, we only use the parameter samples that have less than .1 autocorrelation to obtain uncorrelated samples from $P(x^*|\theta, Y_1, \dots, Y_n)$.

The MCMC will benefit from some tuning and good starting values. In general we recommend using the maximum likelihood estimates (MLE) for θ as starting values if possible. For the proposal distribution, running a short chain using the naive adaptive Metropolis Hastings algorithm then estimating the proposal covariance based on the short chain can improve the efficiency drastically. Lastly, although the adaptive Metropolis Hastings algorithm “auto-tunes” itself, it is often desirable to tune the algorithm so the starting acceptance rate is greater than zero. In general, we set the optimal acceptance rate to be between 0.25 and 0.3.

As mentioned before, sampling from equation 2.2.3 needs to be efficient. This can be difficult since we are drawing GP realizations at $x^{(k+1)}$ given the data and the k previous GP evaluations. However, we can introduce another computational trick here, which is to use the Cholesky method of sampling, noting that the Cholesky matrix can be recursively updated as we evaluate more locations simply by adding additional rows to the previous Cholesky matrix, $L_{\tilde{x}}$. If the new Cholesky matrix for the locations $(\tilde{x}, x')$ is $L_{(\tilde{x},x')}$ then we want Q and $L_{x'}$ such that

$$\begin{aligned} L_{(\tilde{x},x')} &= \begin{bmatrix} L_{\tilde{x}} & 0 \\ Q & L_{x'} \end{bmatrix} \\ \Rightarrow \begin{bmatrix} L_{\tilde{x}}L_{\tilde{x}}^T & L_{\tilde{x}}Q^T \\ QL_{\tilde{x}}^T & QQ^T + L_{x'}L_{x'}^T \end{bmatrix} &= \begin{bmatrix} \Sigma(\tilde{x},\tilde{x}) & \Sigma(\tilde{x},x') \\ \Sigma(\tilde{x},x') & \Sigma(x',x') \end{bmatrix} = \Sigma\left((\tilde{x},x'),(\tilde{x},x')\right) \end{aligned}$$

From the expression above, clearly $Q = \left[L_{\tilde{x}}^{-1}\Sigma(\tilde{x}, x')\right]^T$ and $L_{x'}$ is the Cholesky decomposition of $\Sigma(x', x') - QQ^T$. The term $\left[L_{\tilde{x}}^{-1}\Sigma(\tilde{x}, x^*)\right]^T$ can also be recycled to compute the GP mean and draw GP realizations to avoid unnecessary computation.

The reason the Cholesky factor helps computationally is because it can be recycled into many places throughout the process in place of the full covariance matrix Σ. For example, the posterior variance, $\Sigma(x^*, \tilde{x})\left[\Sigma(\tilde{x}, \tilde{x})\right]^{-1}\Sigma(\tilde{x}, x^*)$, can be expressed as $\Sigma(\tilde{x}, x^*)\left[L_{\tilde{x}}L_{\tilde{x}}^T\right]^{-1}\Sigma(x^*, \tilde{x}) = \left[L_{\tilde{x}}^{-1}\Sigma(\tilde{x}, x^*)\right]^T L_{\tilde{x}}^{-1}\Sigma(\tilde{x}, x^*)$. Instead of inverting $[\Sigma(\tilde{x}, \tilde{x})]$, we simply back-solve a triangular matrix into a vector which is much faster. The term $\left[L_{\tilde{x}}^{-1}\Sigma(\tilde{x}, x^*)\right]^T$ can also be used to compute the GP mean: $\beta + \Sigma(x^*, \tilde{x})\left[\Sigma(\tilde{x}, \tilde{x})\right]^{-1}\left(\tilde{Z} - \beta\right) = \beta + \left[L_{\tilde{x}}^{-1}\Sigma(\tilde{x}, x^*)\right]^T L_{\tilde{x}}^{-1}\left(\tilde{Z} - \beta\right)$. Overall, to efficiently draw a GP sample at finite locations we compute $\beta + \left[L_{\tilde{x}}^{-1}\Sigma(\tilde{x}, x^*)\right]^T\left[L_{\tilde{x}}^{-1}\left(\tilde{Z} - \beta\right) + U\right]$ where $U \sim MVN(0, I)$.

We also need to specify an optimization routine. For the demonstrations and applications below, we will use a hybrid of global and local optimizers, specifically simulated annealing followed by a quasi-Newton method as our default optimization routine. However, our algorithm is suitable for any optimization routine. Our choice of the hybrid optimizer is due to convenience and because it strikes a balance between speed and accuracy for our applications.

Since optimization routines converge, they evaluate points that are closer and closer to each other to produce a precise x^*. This, however, can result in a near-singular posterior covariance matrix. To resolve this issue, *Booker et al.* (1999) regularized the matrix by adding an ad hoc diagonal term. In this paper, when numerical singularity is reached (here defined as the conditional variance being less than the $Cov(x', x') * 10^{-8}$), we treat $Z(x')$ as known with zero uncertainty given the previous data and GP evaluations. In this case, the objective function reports the GP mean instead of drawing a sample from the posterior. However, the objective function is not updated, i.e. $Z(x')$ is not included in the list of previous GP evaluations since $Z(x')$ provides no additional information. This makes sense because singularity implies $Y(x')$ is known given the data and the existing GP evaluations. This also guarantees the updated Cholesky matrix is numerically positive definite without losing information.

We note that it is possible to obtain samples of x^* outside the convex hull of the data, which may or may not be appropriate in a given example. For example, this occurs for the scallop data in Figure 2.1.1. This makes sense because the GP realizations can fluctuate in the boundary beyond the extrema in the regions with data just by chance. This can be remedied by imposing restrictions on the optimization routine or the objective function. In our scallop example below, we made the objective function return $-\infty$ (without updating the objective function) when the point of evaluation is further from all data locations beyond some threshold. To set the threshold, for each data point, we find its distance with its nearest neighbor then choose the maximum over those distances. This works well for our examples but other restrictions can be imposed into the objective function if necessary.

2.2.3. Incorporating Derivatives. A benefit from GP modeling is that the derivative process is also a GP (assuming the original GP covariance function is twice differentiable at distance 0). The derivative process can be useful in optimization routines to speed up the search for the extrema. *Solak et al.* (2003) has used the derivative process to improve prediction but it can also be used for optimization routines that require gradient evaluations. More importantly, the analytical forms of the necessary covariance functions to sample the derivative process can be computed from the original covariance function (*Rasmussen and Williams*, 2006):

$$\begin{aligned} Cov\left(Z(x_i), Z(x_j)\right) &= k(x_i,\ x_j); \\ Cov\left(\frac{\partial Z(x_i)}{\partial x_{ih}}, Z(x_j)\right) &= \frac{\partial k(x_i,\ x_j)}{\partial x_{ih}}; \\ Cov\left(\frac{\partial Z(x_i)}{\partial x_{ih}}, \frac{\partial Z(x_j)}{\partial x_{jk}}\right) &= \frac{\partial^2 k(x_i, x_j)}{\partial x_{ih} \partial x_{jk}}; \end{aligned}$$

Here x_{jk} is the k element in the location x_j. Conveniently, the same parameters used for sampling the GP realizations are used for sampling realizations of the derivative process. For GPs with constant means the mean for the derivative process is 0. With the mean and covariance function defined, we have finished specifying the derivative process. If we denote realizations from the derivative process as $V(x)$ at inputs $x^{(w'+1)}$ then we can use equation 2.2.3 again to construct a self-updating and stochastic derivative function that returns a sample from

$$P\left(V(x^{(w'+1)})|\theta, Y_1, \ldots, Y_n, Z(x^{(1)}), \ldots, Y(x^{(k)}), V(x^{(1')}), \ldots, V(x^{(w')})\right)$$

where k is the number of GP samples and w' is the number of derivative samples drawn so far and the previous $\tilde{Z}$ in Equation 2.2.3 now contains data, GP realizations, and corresponding derivative realizations. From this expression, note that sampling from the derivative process also updates the original GP process. Although this is just a draw from a multivariate Gaussian distribution where the updating remains the same, careful indexing is required to differentiate the derivative realizations from the original process realizations.

2.2.4. Derivative field covariance functions with the Matern covariance function.

One common covariance function is the Matern covariance

$$\frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}} \left(\frac{d}{\rho}\right)^{\nu} K_\nu(\frac{d}{\rho}) \tag{2.2.4}$$

where σ^2 is the variance parameter, ρ is the range parameter, and ν specifies the differentiability of the process. $d = distance(x, x')$ is the distance where $K_\nu(.)$ is the modified bessel function of the second kind with degree ν. For the derivative process to be defined, we require $\nu \geq 1.5$. Using the identities that $\frac{\partial}{\partial z} K_\nu(z) = -\frac{1}{2}\left[K_{\nu-1}(z) + K_{\nu+1}(z)\right]$ and $K_\nu(z) = K_{\nu+2}(z) - \frac{2(\nu+1)}{z} K_{\nu+1}(z) = K_{\nu-2}(z) + \frac{2(\nu-1)}{z} K_{\nu-1}(z)$, the corresponding covariance functions are

$$\begin{aligned}
\frac{\partial k(x_i,\ x_j)}{\partial x_{ih}} &= \underbrace{\frac{-\sigma^2}{\rho\Gamma(\nu)2^{\nu-1}}}_{=A}\left(\frac{d}{\rho}\right)^{\nu} K_{\nu-1}(\frac{d}{\rho})\frac{\partial d}{\partial x_{ih}} \\
\frac{\partial^2 k(x_i, x_j)}{\partial x_{ih}\partial x_{jk}} &= A\left(\frac{d}{\rho}\right)^{\nu-1}\{\ \frac{1}{\rho}\frac{\partial d}{\partial x_{ih}}\frac{\partial d}{\partial x_{jk}}\left[K_{\nu-1}(\frac{d}{\rho}) - \frac{d}{\rho}K_{\nu-2}(\frac{d}{\rho})\right] \\
&\quad + \frac{d}{\rho}\frac{\partial^2 d}{\partial x_{ih}\partial x_{jk}}K_{\nu-1}(\frac{d}{\rho})\ \}
\end{aligned}$$

Using these expressions we can construct the stochastic gradient function to sample from the derivative process when the optimization routine requires it. In the algorithm, both the objective function and the gradient function are provided to the optimization routine but they update each other since the conditional distribution changes as the GP and derivative realizations are sampled.

2.2.5. 1D Illustration. We illustrate our method using a one dimensional example that is easily generalized to higher dimensions. The true objective function to minimize is $f(x) = \cos(2\pi x)\exp(-x)$ where $X = [0, 1]$. However, we only observe $Y_i = f(x_i) + \epsilon_i$ where $\epsilon_i \overset{i.i.d.}{\sim} N\left(0,\ \tau^2\right)$ and x_i are equally spaced over $[0, 1]$ for $i = 1, \ldots, 20$. Under the generalized linear model framework, $h(.)$ is the identify function and $f(x)$ is modeled as a GP with constant mean β and the Matern covariance function with smoothness parameter $\nu = \frac{3}{2}$. Here the parameters θ include β, σ, and ρ for the GP, and τ for the noise level. ν is assumed known.

Here we compute the necessary covariance functions to perform the sampling with derivatives. An alternative expression for the Matern $\nu = \frac{3}{2}$ is

$$Cov\left(Y(x_i),\ Y(x_j)\right) = \sigma^2(1 + \frac{\sqrt{3}\|x_i - x_j\|}{\rho})\exp(-\frac{\sqrt{3}\|x_i - x_j\|}{\rho})$$

σ^2 and ρ are parameters we need to estimate through the data. Since we chose a Matern covariance function that is twice differentiable, the covariance of the derivative process is (where $d = \|x_i - x_j\|$ is the Euclidean distance)

$$\begin{aligned}
Cov\left(\frac{\partial}{\partial x_{ih}}Y(x_i), Y(x_j)\right) &= -\sigma^2\frac{3(x_{ih} - x_{jh})}{\rho^2}exp(-\frac{\sqrt{3}d}{\rho}) \\
Cov\left(\frac{\partial Y(x_i)}{\partial x_{ih}}, \frac{\partial Y(x_j)}{\partial x_{jk}}\right) &= \frac{3\sigma^2}{\rho^2}\exp(-\frac{\sqrt{3}d}{\rho})\left[\delta_{hk} - \frac{\sqrt{3}(x_{ih} - x_{jh})(x_{ik} - x_{jk})}{\rho d}\right]
\end{aligned}$$

where $\delta_{hk} = \begin{cases} 1 & if\ k = h \\ 0 & otherwise \end{cases}$ but here $\delta_{hk} = 1$ since we are working in one dimension only.

To obtain samples from $P(\theta|Y_1, \ldots, Y_n)$, we first find the MLEs for β, σ, ρ , and τ as starting values. Then we run the Metropolis Hastings algorithm. The proposal distribution for all parameters is a Gaussian distribution with variance chosen to have roungly 0.25 acceptance rate for each parameter. We run the algorithm until all 4 trace plots do not have obvious trends.

Figure 2.2.1 illustrates one run from the algorithm. We picked a random θ from the MCMC chain then ran the hybrid optimization routine, allowing the routine to determine the sampling locations for the realization. Notice that the curve is only one realization from the GP so other realizations would produce different minimum locations.

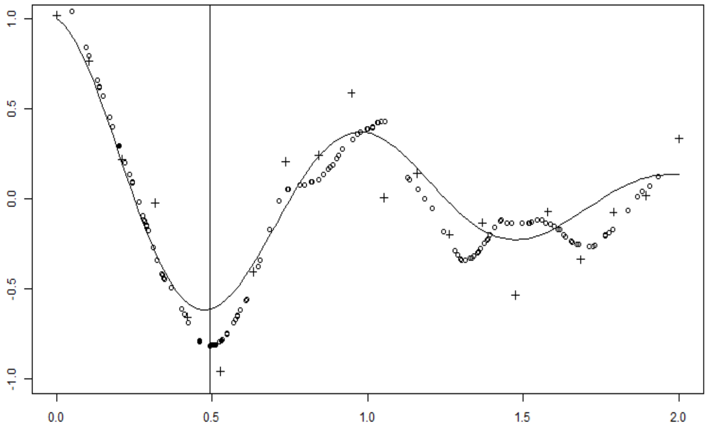

FIGURE 2.2.1. One dimension Illustration of Jointly Sampling and Optimizing GP realization. The solid curve is the true $f(\cdot)$. $+$ denotes the observed noisy data points. $\circ$ denotes evaluations from the global optimization algorithm, and $\bullet$ denotes evaluations from the local optimization algorithm. The vertical line is the output from the hybrid optimization: the minimizing value, x^*, for this sampled realization.

As mentioned, one run of the optimization routine returns only one x^*, whereas the goal is to obtain samples from the posterior distribution of x^*. To get another sample, we run the optimization again using the same data but a different sample of θ from the MCMC. Since our objective function is stochastic, the optima produced will be different. Figure 2.2.2 shows two examples after running the optimization 500 times for data generated either with $\tau = 0.2$ or $\tau = 0.3$.

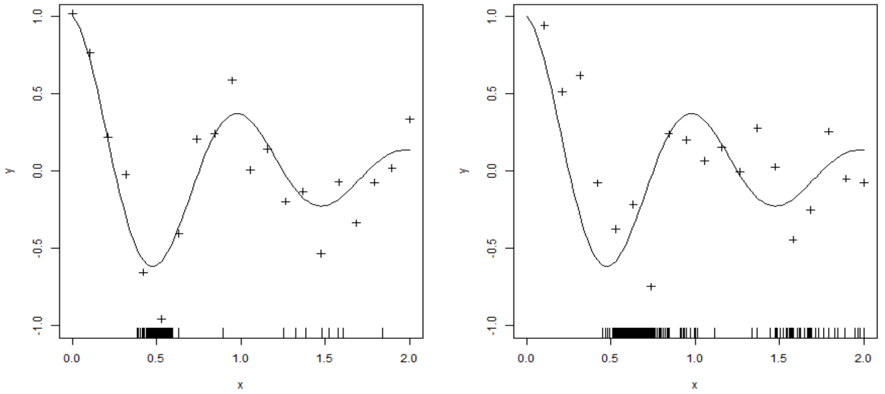

FIGURE 2.2.2. One dimensional illustration of x^* under different noise levels. Left: Low noise level $\tau = 0.2$. Right: High noise level $\tau = 0.3$. The solid curve is the true function $f(\cdot)$. $+$ denote the observed data points. The rug plot reflect the posterior samples of x^*. The higher noise level reflects more uncertainty in x^* since the samples are more dispersed.

Intuitively, the posterior samples concentrate around regions with small observation values. The comparison in Figure 2.2.2 shows that higher noise levels lead to more uncertainty in the posterior distribution and indeed a more noticable second mode. This is entirely appropriate for this data and would not be captured by simply minimizing $E(Z(\cdot) \mid Y_1, \ldots, Y_n)$.

2.3. Efficiency Gains

Intuitively, we expect most optimization routines to outperform a grid search for finding the extrema locations. We will show in our examples that the efficiency gains are higher in higher dimensions. Unfortunately, to create a fair comparison between grid searches and optimization routines is not trivial. The difficulty comes from the fact that grid searches are tuned by the granularity between locations while optimization routines are tuned by the level of improvement between function evaluations. More stringent tuning parameters in both cases will result in more function evaluations. To connect the two dimensions, our examples will work with Lipschitz functions that have the property

$$|f(x_1) - f(x_2)| \leq Lip_X * \|x_1 - x_2\|, \qquad \forall x_1, x_2 \in \Omega$$

for some constant Lip_X over the domain X. With this assumption, we can infer the granularity necessary between locations to achieve any level of improvement in the function. Note that any function with a bounded first derivative is naturally Lipschitz where $Lip_X = \sup_{x \in X} |f'(x)|$.

The demonstration is a simple multimodal function $f(x) = \sum_{i=1}^{p} x_i^2$ where $x = (x_1, \ldots x_p)$ over the domain $[-0.5, 0.5]^p$. The true minimum is at the origin and the gradient is $f'(x) = [2x_1, \ \ldots, \ 2x_p]$. Given our domain, the derivative is bounded by $Lip_X = 1$. We observe 10^p points over an evenly spaced grid throughout the domain with some noise from $N(0, \tau^2)$ as shown in Figure 2.3.1 (τ is set to 0.03 that is unknown to the algorithm). We run the adaptive Metropolis Hastings algorithm to obtain the parameters and run the local optimization (quasi-Newton method) routine alone with the self-updating objective function.

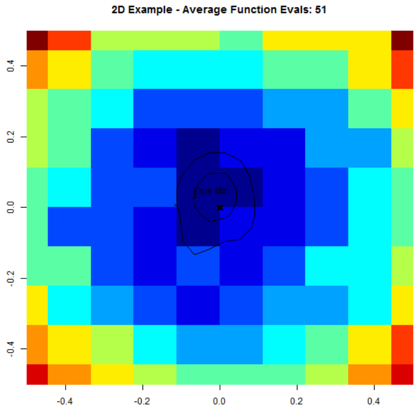

FIGURE 2.3.1. Extrema Location Density Estimate vs Data. The contour shows the density estimation for the posterior in the 2D case where the grid colors shows the data values. The contour centers the true minimum which is promising.

Table 2.3.1 reports the average number of function evaluations over 301 runs with different relative tolerance levels. A relative tolerance ϵ indicates that the function improvement must be less than $\epsilon(|y| + \epsilon)$ to conclude convergence. In other words, we conclude convergence if $\left|y^{t+1} - y^t\right| \leq \epsilon(|y^t| + \epsilon)$ when evaluating y^{t+1}. For our example, the least stringent threshhold is

$$10^{-4}\left(\underbrace{\sup_X f(x)}_{=0.5^2 p} + 10^{-4}\right) = z_p$$

To reach the same precision for a naive grid search, the Lipschitz condition suggests $z_p \leq Lip_X * \|t_1 - t_2\|$ so the granularity needs to be at least $\frac{z_p}{Lip_X}$ ($2.501 * 10^{-5}$ for 1D example and $5.001 * 10^{-5}$ for the 2D example). This implies at least $[\text{range}(X)/\left(z_p/Lip_X\right)]^p$ function evaluations over the grid (assuming the domain is a cube) for the same precision. Notice as z_P gets smaller and p gets larger, the number of function evaluations increases which agrees with our intuition that more stringent thresholds and large dimensions optimization is more challenging. Larger Lip_X values also have the same effect since the function can fluctuate more easily.

The number of function evaluations for the grid search only requires the Lipschitz number, the relative tolerance level, the domain size, and the maximum value within the domain. This allows us to estimate the resources needed for a grid search.

ϵ	$p=1$	$p=2$	$p=3$
10^{-3}	24.0 ± 9.0 *vs* $4 * 10^3$	52.0 ± 8 *vs* $4 * 10^6$	52.7 ± 8 *vs* $2.4 * 10^9$
10^{-4}	28.6 ± 10.7 *vs* $4 * 10^4$	50.99 ± 8.69 *vs* $4 * 10^8$	52.7 ± 7.8 *vs* $2.4 * 10^{12}$

Table 2.3.1. Function Evaluation Comparisons. p is the dimension of the input space where ϵ is the relative tolerance. Function counts from our stochastic method are reported as $mean \pm 1SD$ over 301 runs. The grid search evaluations are derived from the conservative estimates using Lip_X and the relative tolerance parameter ϵ. The function evaluations are much lower for our algorithm relative to the grid search. The difference is even more noticable in higher dimensions.

To make the comparison fair, we can assume the grid search would perform a coarse grid search before conducting a detailed grid search. This coarse search however can correspond to the number of iterations we allow the global optimizer to search the domain. In this example we assumed we have found the best grid where now a local search would suffice. Table 2.3.1 shows the number of GP evaluations using the local optimization algorithm is much fewer than the grid search. The savings is even greater as p increases.

2.4. Latent Gaussian Field: Harvesting Scallops

Our first data example is the scallop data set from *Ecker and Heltshe* (1994) shown in Figure 2.1.1. The dataset is publicly available in the R package SemiPar and our code is available online. The dataset contains longitude, latitude, and total catches of scallops (counts) $Y_1, \ldots, Y_n$ at locations $x_1, \ldots x_n$ where $n = 148$. This dataset is interesting because the data is not Gaussian, but we can still model the underlying intensity as a transformed GP.

We model the total catches as independent Poisson random variables given the underlying intensity. We will model the log transformation of the scallop abundance, $\log(\lambda(x))$ as a GP which we will denote as $Z(x)$. For this example, we project the latitude and longitude into UTM coordinates and treat the locations as if they were in $\mathbb{R}^2$ (all distances are in Euclidean distances). Given the small area this assumption should be reasonable. Our model for the scallop catches is

$$\begin{aligned} Y_i|\lambda(x_i) &\overset{independent}{\sim} Poisson(\lambda(x_i)) \qquad i = 1, \ldots, 148 \\ \log(\lambda(x)) = Z(x) &\sim GP\left(\beta, \sigma^2\Sigma(\rho)\right) \end{aligned} \tag{2.4.1}$$

where $\Sigma(\cdot)$ is the Matern covariance function in equation 2.2.4. In words, the scallop catches, Y_i, at location x_i is modeled as independent Poissons conditioning on the abundance level $f(x)$, i.e. $Y_i|f(x_i) \sim Poisson\left(f(x_i)\right)$. The abundance level is treated as the intensity function for the Poisson catches. We set $h(.) = \log(.)$, so the log of the intensity function is modeled as a GP. To infer the log abundance level $Z(x_{n+1})$ at any unobserved location x_{n+1}, we calculate $P\left(Z(x_{n+1})|Y_1, \ldots, Y_n, Z(x_1), \ldots, Z(x_n)\right)$. This completes the specification for the scallop catches and allows inference for scallop catches and abundence level at all other input values in X.

The goal is to find $P\left(\arg\min_x \lambda(x)|Y_1,\ldots,Y_n\right)$ which is the same as $P\left(\arg\min_x Z(x)|Y_1,\ldots,Y_n\right)$ since log is monotone. In contrast to the illustration in Section 2.2.5, the likelihood now involves the Poisson density and the intensity function $\lambda(x)$ needs to be estimated. To estimate $\lambda(x)$ through the generalized model, they will be inferred like the parameters θ (that govern the GP itself). In other words, we will first sample $P(\tilde{Z},\theta|\tilde{Y})$ then draw $P(x^*|\theta,\tilde{Z},\tilde{Y})$ where $\tilde{Z} = Z(x_1),\ldots,Z(x_n)$.

To fit this model, we run adaptive Metropolis Hastings on β, σ^2, ρ, and $\tilde{Z}$ at the 148 locations where we observe data (again assume $\nu = \frac{3}{2}$, the smallest value that allows derivative inference to be incorporated). This results in 151 parameters for the MCMC algorithm that we jointly propose. We initialize the MCMC algorithm by estimating $Z(x_i) \approx \log(Y_i + 1)$ then use these values to estimate β, σ^2, ρ via maximum likelihood for good starting values. Moreover, instead of initiating the proposal with the naive adaptive Metropolis Hasting algorithm, the proposal covariance matrix is made block diagonal composed of two blocks. The first block is simply a diagonal matrix with the square of the MLE coefficients where the second block is the Matern covariance for the abundance level using the MLE coefficients. We then run a short chain then adjust the proposal covariance and rerun the adaptive Metropolis Hastings algorithm. Convergence is determined when no clear trends appear in the long chain. The acceptance rate after burn-in is roughly .249. For this example, running an overall of 3,000,000 iterations took two days and the chain reached stationarity roughly around 1,000,000 iterations. Our processor is a Quad-Core AMD Opteron(tm) Processor 8384 with 2692.847 MHz.

Then we use samples from the MCMC, $\theta^i, Z^i(x_1),\ldots,Z^i(x_n)$ for $i = 1,\ldots B$ to obtain samples from $P(x^*|\theta,\tilde{Y},\tilde{Z})$ via the algorithm described in Section 2.2. The resulting density estimation from these samples are reflected in the contour in Figure 2.1.1. There is clear multimodality.

One major issue with this dataset is that the data is not evenly distributed around the domain. This allows the sampled x^* to be outside of the region with data. The posterior mean does not have this issue because it converge to the prior mean at regions without data and this is typically not extreme. Again, as mentioned in the methodology section, we take the domain X over which to maximize to be the set of locations $\{s : \|s - x_i\| < r,\ i = 1,\ldots n\}$, where r is the largest nearest neighbor distance among all data points.

2.5. Estimating Absorption Minima in Spectroscopy Data

Our last example data set is the public spectrophotometric time series of the Type Ia supernova SN 2011fe (*Pereira et al.*, 2013), obtained by the Nearby Supernova Factory (*Aldering et al.*, 2002). The data is available at `http://snfactory.lbl.gov/snf/data/index.html`. The time series itself is a sequence of spectra, each consisting of flux and flux error in units of erg s^{-1} cm^{-2} Å^{-1} tabulated as a function of wavelength in Å. Each spectrum was obtained on a different night. There are 25 nights of observation in the time series. We work with a subset of 438 flux (and flux error) measurements for each night in the region $wavelength \in (5625.134, 6667.498)$ in which we know the minima will be. This yields a sizable 10950 values in the dataset. The time coverage is not uniform but the wavelength grid is regularly spaced and the same from night to night. The flux values themselves have been calibrated so that differences in the brightness of the supernova from night to night and wavelength to wavelength are physically meaningful.

We are interested in obtaining the wavelengths of absorption feature minima in the flux of SN 2011fe as a function of phase (t, defined as days relative to the time of maximum luminosity) and wavelength, i.e. $x^*_{t_i} = \arg\min_{wavelength}\left[flux(wavelength,\ t_i)\right]$ for $i = 1, \ldots J$. The spectrum of a supernova contains broad absorption and emission features whose appearance is the result of physical processes and conditions in the expanding stellar ejecta. The widths, depths, and heights of such features change with time as the supernova expands and cools. The wavelengths of absorption feature minima are physically interesting quantities to extract from spectral time series as a function of time. These translate to a characteristic ejecta velocity that provides an estimate of the kinetic energy of the supernova explosion, something of great interest to those that study exploding stars.

Overall, estimating $\{x^*_{t_i}\}_{i=1,\ldots,J}$ provides an estimate for the ejecta velocity. We model the spectrum as a GP with a mean derived from a standard template Type Ia supernova spectral time series (*Hsiao et al.*, 2008). We take the correlation function to be a product of two Materns correlations, one for each dimension, each with smoothness parameter $\nu = 2$. We then introduce random effects for each phase (time point) to adjust for the systmatic deviation from the mean for each phase. Lastly, there is measurement error from photon noise.

The model we propose for the flux measurements is

$$\begin{aligned}
Y_i &= Z(t_i, wavelength_i) + \alpha_{t_i} + \epsilon_i \\
\alpha_1, \ldots \alpha_J &\overset{i.i.d.}{\sim} N(0, \tau^2) \\
Z &\sim GP\left(\mu(\cdot; \kappa, \lambda),\ \sigma^2 K(\cdot, \cdot; \rho_{phase}, \rho_{wavelength})\right) \\
\epsilon_i &\sim N\left(0, \xi_i^2\right) \text{ where } \epsilon_1, \ldots, \epsilon_n \text{ are independent} \\
\mu(wavelength,\ t_i; \kappa, \lambda) &= \kappa g(\frac{t_i}{\lambda})
\end{aligned}$$

Here $K(\rho_{phase}, \rho_{wavelength}) = K_1(\rho_{phase})K_2(\rho_{wavelength})$ where K_1 and K_2 are both Matern correlation functions with smoothness parameter $\nu = 2$, J is the total number of phases in the dataset, and n is the number of flux measurements. $\mu(wavelength,\ t_i; \kappa, \lambda)$ is a template based on aggregating many different spectral time series that are transformed to match observed luminosities (*Hsiao et al.*, 2008). α_{t_i} is a random effect shared among flux values in the same phase that represents a systematic deviation from the template. Z is the light curve that is modeled as a GP. ϵ is measurement error in the flux values and its standard errors ξ_i are derived through the image extraction process and are fixed and known. Priors for the mean parameters were chosen to be $\kappa \sim Uniform[0, 2]$ and $\lambda \sim Uniform[0, 3]$ based on conservative bounds given by our collaborator. The other priors are all specified as described in Section 2.2.1.

The goal is to derive the joint posterior for $P(x^*_{t_1}, \ldots, x^*_{t_J}|Y_1, \ldots, Y_n)$. To do this, we need to record the function evaluations for all previous k phases when sequentially searching for $x^*_{t_{k+1}}$. This quickly increases the computational burden and numerical stability issues for this example, but implementation is still feasible using the techniques described in Section 2.2.2.

We run the adaptive Metropolis Hastings algorithm to obtain posterior samples for $\theta = \{\sigma^2, \rho_{phase}, \rho_{wavelength}, \tau^2, \kappa, \lambda\}$. The algorithm is initiated at the MLEs and the convergence is judged by looking at the trace plots. Then we run the hybrid optimization routines using those parameter samples. One detail in the implementation is that we only implement simulated annealing in the search for $x^*_{t_1}$, and for $j > 1$, the search for $x^*_{t_j}$ begins at the converged value for $x^*_{t_{j-1}}$. This utilizes the smoothness over phases in the light curves and avoids the computational burden of the global optimizer. We derived $x^*_{t_i}$ at 0.5 intervals over the input space including times with and without observations.

The output from each run is the wavelengths corresponding to the minimum fluxes for each phase. To translate each wavelength value to the ejecta velocity, we calculate $v_{t_i} = c(\lambda_R/x^*_{t_i} - 1)$ where $\lambda_R = 6355$ is the rest wavelength of an important silicon ion transition and c is speed of light $3 * 10^8 \frac{m}{s}$.

To quantify the uncertainty, we generate a 95% credible band, for which 95% of the posterior samples are within the band for all phases (see Figure 2.5.1). There is no unique way to define this credible band. Some phases have asymmetrical posterior distributions, so we construct the credible band using phase-wise posterior sample percentiles. At each phase, the band ranges from the $100 * \frac{\alpha^*}{2}$ percentile to the $100 * (1 - \frac{\alpha^*}{2})$ percentile where $\alpha^* \in [0, 1]$ (same α^* for all phases). We pick the largest α^* that contains 95% of the runs within the credible band for all phases. This construction centers the credible band around the phase-wise median and accounts for the dependency over phases. Notice that the width of the interval is not constant over different phases. The width, a measure for the uncertainty, has an inverse relationship with the amount of data near each phase. Some $x^*_{t_i}$ are also easier to predict due to the concavity of the spectrum.

The method in the chapter provides the ability to extract not just the position of the absorption minimum in an isolated spectrum, but to follow the shift in the position of the absorption minimum with time. Modeling the data as a GP allows us to interpolate the absorption minimum position between observations in a principled way as well. This is in marked contrast to the standard technique, which is to measure the position of absorption minima in each spectrum independently (with some error estimate) then interpolate the results afterwards. Finding the posterior distribution of x^* instead of a point estimate at each spectrum fills in the spaces between the observations in a more principled way that uses the covariance structure instead of treating each spectrum as independent.

2.6. Discussion

In this chapter, we combined optimization with GP sampling by constructing a self-updating objective function. This creates an efficient method to obtain the posterior samples of x^*. Under the Bayesian framework, we incorportated both the uncertainty in the parameters and the data to obtain the posterior distribution of x^*. We demonstrated our method on non-Gasussian data and a large dataset with promising results.

When the input domain is large, drawing samples from $P(\theta|Y_1, \ldots, Y_n)$ or GP realizations are time consuming. However, if the entries in the covariance are small for distant points, it might be appropriate to taper the covariance matrix to speed up computations with sparse matrices algorithms as shown in *Kaufman et al.* (2008). For example, for phases or wavelengths that are far apart, we could

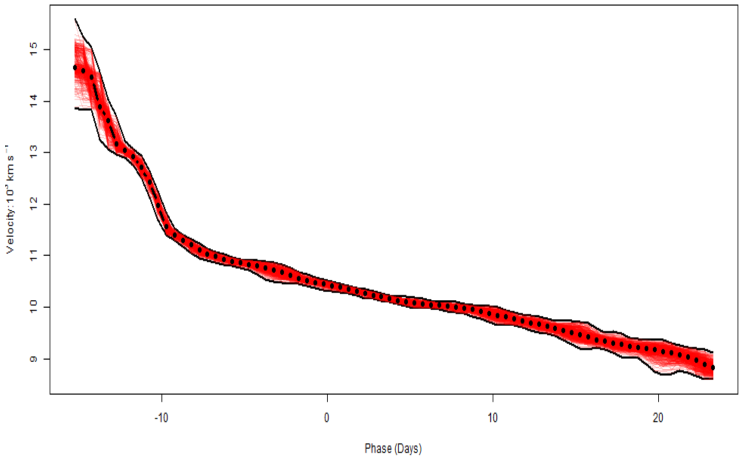

FIGURE 2.5.1. 95% Ejecta Velocity Credible Band Over Phases. We transform $\{x^*_{t_i}\}_{i=1,\ldots,J}$ into velocities using the transformation: $v_{t_i} = c(\lambda_R/x^*_{t_i} - 1)$ where $\lambda_R = 6355$ is the rest wavelength of an important silicon ion transition, and c is speed of light $3 * 10^8 \frac{m}{s}$. The center line mark the phase-wise medians.

taper the covariance entries to 0 and retain the differentiability of the GP. This allows for fast computation and efficient storage with the sparse matrix algorithms. However, the corresponding derivative process covariance function will be different after tapering and needs to be recalculated. Other similar strategies exist where covariance functions with compact support can be used as shown in *Kaufman et al.* (2011). In particular, the Bohman covariance function has compact support and is twice differentiable. We explored these strategies for the large SN2011 dataset but the estimated effective ranges were too large for these approaches to provide significant reduction in the computation without impacting the quality of the results.

Although drawing posterior parameters is time consuming, it is how uncertainty in the GP parameters are introduced in the results. It is tempting to simply fix the parameters at the MLE and perform the optimization. We believe this practice will ignore the variability of the unknown parameters and can lead to overly confident results. However, if there is enough data, the posterior distribution for the parameters might be concentrated enough that the resulting posterior will not differ markedly by using the MLE. In some of our examples, different parameters produced a wide variety of GP realizations so the variability should not be ignored in general.

One possibility to use the MLE parameters is to carry out a hypothesis test and test if the posterior samples of x^* using either parameters have a different distribution. *Bickel* (1969); *Hall* (2002) have proposed permutation based rank tests for high dimensional distributions with limited samples. To implement these methods, we need to first compute a statistic for each sample generated from the MLE and Bayesian parameters then perform a rank test on the sorted statistics. If the change is not detectable then one might be justified to use the MLE parameters.

The method we have proposed is conceptually applicable to most optimization algorithms although we have not systematically explored the choices of algorithms here. The choice of the hybrid optimization routine was purely due to convenience but worked well for our applications. The one concern about the hybrid optimization routine is whether the global optimizer sufficiently explores the domain. If few global optimization steps were allowed, it is possible to miss the global extreme and get stuck at a local extreme. This has not happened in our examples after examining the data and global optimizer behavior. Unfortunately, there is no quick way to determine whether unexpected extremes are due to insufficient exploration from the optimizers or simply volatile GP realizations.

Naturally, the choice of the optimization routine depends on the dataset. If one believes multiple peak locations exist then the global optimization routine might be necessary although it is often slow. If that is not a concern, then implementing the local optimization routine alone will be faster and sufficient. Different optimization routines excel in different situations so we leave this question open.

CHAPTER 3

Varying Coefficient Model for Global Surface Wind Fields over Land and Sea

3.1. Introduction

We propose a statistical model for surface wind fields over the globe. This model has many potential applications, such as prediction of wind energy and uncertainty quantification for climate models. The intermittent nature of wind prevents energy suppliers from relying on wind energy. A statistical model can produce wind predictions that identify potential wind farm locations and decrease the uncertainty with wind energy (*Hering and Genton*, 2010). Moreover, surface wind is a major driver for the ocean and critical for climate models (*Kerr*, 1998; *Milliff et al.*, 1999). Climate models each have different physics and parameters that produce surface winds with different behaviors. Quantifying the effects of these differences has been a major challenge in climate science that often relied on statistical techniques. For example, *Kaufman and Sain* (2010); *Sain et al.* (2011) have quantified the effects of downscaling for temperature and precipitation. To implement similar analysis for surface winds, however, we need a statistical model that can capture the spatially heterscedastic and intrinsically multivariate nature of surface winds.

To help elucidate the challenges in surface wind modeling, we show a sampled global wind field in Figure 3.1.1. Unlike temperature and precipitation, wind is intrinsically multivariate with varying speeds and directions. Modeling multivariate spatial fields is an active field in spatial statistics that can lead to complex models (*Gelfand et al.*, 2010). Figure 3.1.1 also shows that surface winds are spatially heteroscedastic. The heteroscedasticity is most apparent when comparing wind vectors over land and sea and different latitudes. The wind field over the ocean is often larger in magnitude and smoother where the opposite is true over land surfaces. There is also a negative correlation between wind speed and the distance from the equator. Capturing these features require a flexible statistical model which comes with many computational issues. The computational issues are worsened as the resolution for wind fields increase over different generations of climate models. In face of these challenges, some surface wind models have been applied to limited locations while others have been restricted to regions over the ocean. We review these approaches in Section 3.2.

In this chapter, we construct a novel statistical model for global surface winds that will addresses all of these challenges. The model is motivated by the geostrophic relationship between surface wind and pressure gradient (*Royle et al.*, 1998; *Milliff et al.*, 2011). This relationship attributes the dependency between eastward and northward wind velocities to a shared variable, pressure. To capture the heteroscedasticity, we introduce a spatially varying coefficient model (*Gelfand*

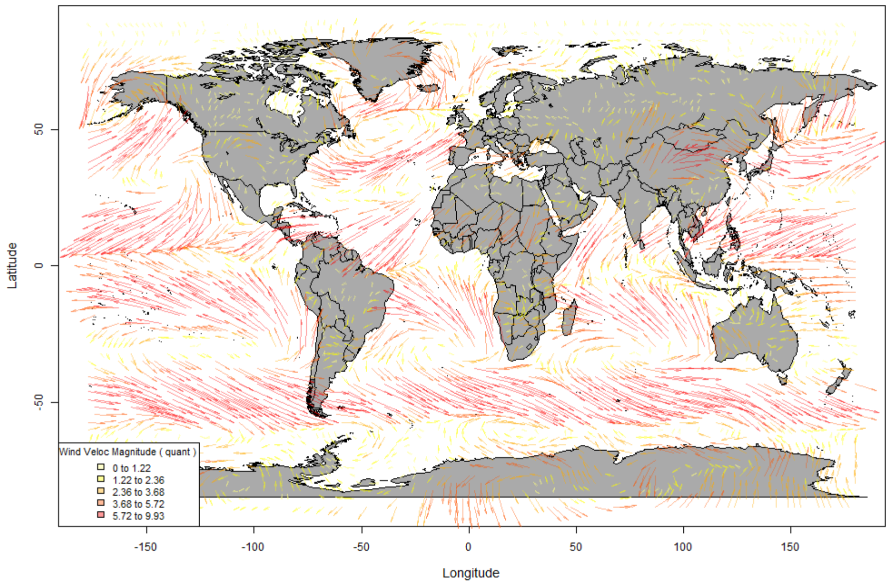

Figure 3.1.1. Sub-sampled Global Wind Field. The length and darkness of the arrows are positively correlated to the wind velocities. This example demonstrates the spatially heteroscedastic and multivariate nature of wind fields. The field is smoother over the ocean than land surfaces. There is also a negative correlation between speed and the distance from the equator.

et al., 2003) that allows the geostrophic relationship to vary spatially. We model the wind fields and varying coefficients with Gaussian Processes (GPs). Although both spatial fields produce dense covariance matrices, we resolve the computational demands by implementing the Gaussian Markov Random Field (GMRF) methods proposed by *Lindgren et al.* (2011). The idea is to construct sparse precision matrices (inverse of the covariance matrix) that correspond to the smooth Matern covariance matrices. This allows efficient computation with sparse matrix routines. We will show that our posterior sample wind fields can capture the behavior of the wind fields from the climate models.

Section 3.2 reviews the statistical literature on wind. Section 3.3 describes the data we use for this chapter. Section 3.4 details the model and distributional assumptions. Section 3.5 lays out the details of how we fitted the model since this is a major challenge for statistical models that involve GPs. Section 3.6 validates our model and finally Section 3.7 covers potential extensions and limitations for our model.

3.2. Literature Synthesis

Statistical models on surface winds have been motivated by applications in harvesting wind energy (see e.g. *Haslett and Raftery* (1989); *Hering and Genton* (2010); *Salameh et al.* (2009)). The data in these papers were wind records from weather stations for a few locations. These statistical models perform well locally but in general cannot extend to global wind fields. The coefficients are often not constant over space at the global scale. Moreover, the dependence structure for wind fields is high dimensional and requires different modeling approaches.

Cornford (1997, 1998) modeled the high dimensional dependence in wind fields using GPs. The key was to decompose the wind into uncorrelated components (stream function and velocity potential) then model each component using GPs in a hierarchical Bayesian framework. This avoids the dependency structure between the different wind components and gives physically meaningful interpretations to the model. Later work also tackled the dependency structure using a similar hierarchical Bayesian model (BHM) framework but attributing the wind dependencies to a shared variable between the components such as pressure. An issue with modeling the stream function and velocity potential is the resulting wind fields can be quite unstable. In our attempts, simulating wind fields from this method can yield unreasonably large variances. The use of GPs to reflect the smooth patterns in wind fields was also widely adapted in later work. The computational challenges from modeling the smooth wind fields however often restricts the size of possible datasets.

Royle et al. (1998); *Milliff et al.* (2011) constructed hierarchical models with GPs conditioning on pressure, which was unobserved and treated as a latent process. They used pressure to explain the dependency between the different components of wind. However, their models were restricted to the sea surface since the effects from the pressure gradients dominates surface wind behaviors over low friction regions. Their models also focused on a relatively small region where the effects of pressure gradient did not vary much within the region of interest. However, our explorations on global wind fields show that this assumption does not hold over larger regions. We will generalize their model using a varying coefficient model that can, surprisingly, capture the variability of surface winds over land as well.

To tackle the computational demands from GPs, *Wikle et al.* (2001) used basis functions to reduce the dimensionality of wind fields. The key is to project the wind fields onto a few basis functions then model the data on this low dimensional space. One issue with basis functions is the various ad hoc choices involved in selecting a basis function class, the number of basis functions, and whether each basis is modeling the covariance structure or the mean patterns. *Shi and Cressie* (2007) provides general guidance to these questions. Even so, the spatial basis in *Wikle et al.* (2001) were chosen for the tropical ocean and may not extend easily to the land surfaces. Poor selection of basis functions can lead to patterns in the posterior distribution as an artifact of the basis instead of the data as shown in Figure 5 in *Cressie and Johannesson* (2008).

Lastly, *Reich and Fuentes* (2007) proposed a semiparametric model using stick-breaking methods instead of GPs to model hurricane surface winds. In their results, their model could adapt to the asymmetries and complex hurricane patterns better than GP based wind models. Unfortunately, the flexibility in the stick breaking methods comes with great computational demands that are not suitable for global scale wind fields.

3.3. Data

We demonstrate our model using the data products from the PCMDI database over the globe. The data is defined over a regular grid with resolution roughly 2.8 by 2.8 degrees both in longitude and latitude. This yields 128 longitudes for each 64 latitudes with a total of 8192 locations. More specifically, we took the daily frequency sea level pressure and surface winds from the Japanese Model MIROC3.2 at medium resolution under the pre-industrial experiment scenario.

In this analysis, we work with average wind fields over each season for each year. Since the relationship with pressure gradients is linear (see Equation 3.4.2), the relationship should hold for the averaged fields. This yields 40 years of average surface wind fields for Winter and Summer. We treat the separate years as replicates since no forcings were introduced to the runs. We also have the 40 years of sea level pressure that was averaged over each season for each year. The sea level pressure gradient is approximated by taking the difference of sea level pressure in neighboring locations divided by the respective great circle distance (*Royle et al.*, 1998). Lastly, the locations of each observation is transformed to fit over a unit sphere to implement the GMRF algorithm built by *Lindgren et al.* (2011).

In climate science, wind fields are generally decomposed into U, the eastward winds (zonal winds), V, the northward winds (meridional winds), and W, the vertical wind. Most researchers (see e.g. *Royle et al.* (1998); *Wikle et al.* (2001); *Reich and Fuentes* (2007); *Milliff et al.* (2011)) have focused their efforts on modeling U and V and ignored W. Through our exploration on surface winds, the magnitude of W is insignificant relative to the magnitudes in U and V unless a storm system is formed. In other words, the velocity in the prevailing winds mostly come from U and V so we will ignore W as well for simplicity.

Finally, we withhold the latest 10 years of wind fields as a test set. We will fit the model using the first 30 years of data then predict the next 10 years of data for validation.

3.4. Methods

To tackle the dependency structure between U and V, we model U and V as functions of pressure gradient. This is useful because sea level pressure is usually considered a stationary univariate variable where its gradient can explain complex variations within the multivariate wind fields. This physical relationship was first used by *Royle et al.* (1998) to approximate the first order relationship between pressure gradient and the velocity of each wind component. The geostrophic relationship is

$$(3.4.1) \qquad -f_{lat}V = -\frac{1}{\rho}\frac{\partial P}{\partial x}; \qquad f_{lat}U = -\frac{1}{\rho}\frac{\partial P}{\partial y}$$

where P is the sea level pressure, ρ is the air density which depends on the local ratio between dry and wet air, f_{lat} is the Coriolis parameter that varies based on latitude (*Neelin*, 2010), and $\frac{\partial P}{\partial x}$ and $\frac{\partial P}{\partial y}$ are the pressure gradients. *Royle et al.* (1998), however, built a linear model where each wind component depended on both pressure gradient terms. This model was later justified by *Milliff et al.* (2011) through expanding the Rayleigh friction equations to introduce the effects of friction. Specifically, the resulting equation used in *Milliff et al.* (2011) was

$$\begin{aligned} U &= -\frac{\gamma}{\rho(f_{lat}^2+\gamma^2)}\frac{\partial P}{\partial x} - \frac{f_{lat}}{\rho(f_{lat}^2+\gamma^2)}\frac{\partial P}{\partial y} \\ V &= \frac{f_{lat}}{\rho(f_{lat}^2+\gamma^2)}\frac{\partial P}{\partial x} - \frac{\gamma}{\rho(f_{lat}^2+\gamma^2)}\frac{\partial P}{\partial y} \end{aligned} \tag{3.4.2}$$

where γ is the Rayleigh friction term which depends on the underlying surface. Notice that without friction, i.e. $\gamma = 0$, then we retrieve the geostrophic equation in Equation 3.4.1. Interestingly, *Chiang and Zebiak* (2000) have shown that γ has different magnitudes for U and V under a linear friction assumption. This suggests that the coefficients for the pressure gradients should differ between U and V.

With these physical motivations, the implied linear equation then becomes

$$\begin{aligned} V &= \beta_{v,1}(f_{lat},\rho,\gamma_v)\frac{\partial P}{\partial x} + \beta_{v,2}(f_{lat},\rho,\gamma_v)\frac{\partial P}{\partial y}; \\ U &= \beta_{u,2}(f_{lat},\rho,\gamma_u)\frac{\partial P}{\partial x} + \beta_{u,1}(f_{lat},\rho,\gamma_u)\frac{\partial P}{\partial y} \end{aligned} \tag{3.4.3}$$

The interpretations for the parameters, however, suggest that the coefficients for the linear relationship should change spatially by latitude, air density, and friction. Friction depends on the underlying surface which is stronger over mountainous areas and weaker over the ocean. Broadly speaking, air density is a function of temperature, pressure, and moisture which all varies spatially.

To handle the spatial heterscedasticity, we propose a spatially varying coefficient model where the coefficients in the linear relationship are allowed to vary spatially. This adjustment is often not done because most examples in the literature focus on a reasonably small region over the ocean. Our model covers winds at the global scale and extends over land surfaces which necessarily has varying parameters.

Spatially varying coeffieint models have typically been applied on smaller datasets (*Assunçao*, 2003) although *Gelfand et al.* (2003) have implemented this for more sizable problems with specific covariance structures. The idea is to allow the coefficients in a linear model to change smoothly over the spatial domain. With the spatial dependence, coefficients can be inferred using neighboring data while restricting the total degrees of freedom of the varying coefficient field. This results in flexible models that do not overfit the data. This method is straightforward to understand but its computational burden on large datasets makes it difficult to implement. We will discuss solutions for this in Section 3.5.

Through our data exploration, the pressure gradient is not the only source of variability for surface winds so we include an intercept and error terms to the final linear model. The intercept should capture consistent deviations from Equation 3.4.3 where the error term is the variability that cannot be captured otherwise.

$$\begin{aligned} U_t(s) &= \beta_{u,0}(s) + \beta_{u,2}(s)\frac{\partial P_t}{\partial x}(s) + \beta_{u,1}(s)\frac{\partial P_t}{\partial y}(s) + \epsilon_t(s) \\ V_t(s) &= \beta_{v,0}(s) + \beta_{v,1}(s)\frac{\partial P_t}{\partial x}(s) + \beta_{v,2}(s)\frac{\partial P_t}{\partial y}(s) + \epsilon_t(s) \end{aligned} \tag{3.4.4}$$

s indicates the locations and t are indices for different years. Notice that the coefficients depend on s so they vary spatially. Similar to *Cornford* (1997, 1998), we fit a BHM to facilitate inference. A graphical overview of the model is shown in Figure 3.4.1.

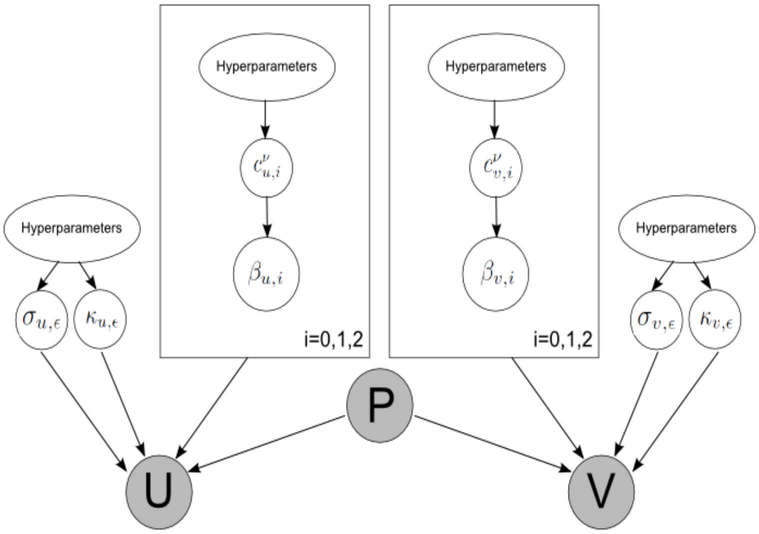

FIGURE 3.4.1. Graphical Model for Varying Coefficient Model For Surface Winds. The gray nodes are observed from the data. The other parameters are derived from fitting the model in Section 3.5. This graph details all of our conditional independence assumptions in our model.

Based on the model in Equation 3.4.4, we model U and V as conditionally independent given the sea level pressure, i.e. $p\left(U,V|P\right) = p\left(U|P\right)p\left(V|P\right)$. This choice was also made by *Milliff et al.* (2011) who commented that this decision did not affect their results and made their model much more efficient. We further validate this assumption in our exploratory analysis by fitting a naive version of the model in Equation 3.4.4. We perform a ordinary least squares regression separately for U and V with respect to the pressure gradient for each location treating different years as replicates. This completely ignores the spatial dependence between the coefficients and data but serves as a reasonable proxity for the final model. Figure 3.4.2 compares the residuals for U and V from this naive fit. No relationship is visible between the residuals which supports our conditional independence assumption.

Throughout we assume that pressure will be given and omit it from our notation. To detail our BHM, we break the data generating process into the data model, the process model, and the prior model.

The data model for U focuses on the observations given the underlying processes, i.e. the coefficient fields (V is similarly defined). We model the residuals from the varying coefficient model as a GP with variance that changes spatially.

$$\begin{aligned} U|\beta_u, \kappa_{u,\epsilon} &\sim GP\left(X(\cdot)^T\beta_u(\cdot),\ \sigma^2_{u,\epsilon}(\cdot)K_{\nu_\epsilon}(\cdot,\cdot;\kappa_{u,\epsilon})\right) \\ \log(\sigma_{u,\epsilon}) &\sim GP\left(0,\ b^2_\sigma K_{\nu_\sigma}(\cdot,\cdot;\kappa_\sigma)\right) \end{aligned} \tag{3.4.5}$$

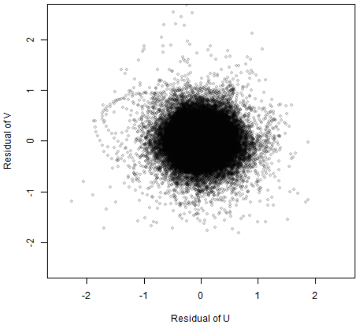

FIGURE 3.4.2. Graphical Examination of Conditional Independence between U and V given P. A least square regression was fitted separately for U and V for each location treating different years as replicates and ignoring any spatial dependence. If U and V were not independent given P, we should expect heterscedastic or linear relationships between these residuals.

For location s, let $X(s)^T$ be a feature vector $\left(1,\ \frac{\partial P}{\partial x}(s),\ \frac{\partial P}{\partial y}(s)\right)$ and $\beta_u(s)^T = (\beta_{u,0}(s), \beta_{u,1}(s), \beta_{u,2}(s))$. ν_ϵ and ν_σ are the smoothness parameters for the Matern correlation function K and will be assumed to be 1 based on our explorations with the data. $\kappa_{u,\epsilon}$ is the inverse of the range parameter in Matern covariance functions. The mean is the linear relationship as described in Equation 3.4.4. We also allow the variance of the residual field to varying spatially since our explorations and *Neelin* (2010) has suggested that the relationship with pressure gradient does not hold well beyond the mid-latitude regions. In response, we model the log transformed standard deviations, $\log(\sigma_{u,\epsilon})$, as a GP. In our analysis, κ_σ and b_σ^2 will be fixed to create a sensible weakly informative prior based on physical bounds. Details setting for these hyperparameters are detailed in Appendix A.1 and Appendix A.3.

The process model for the coefficient field is also modeled as a GP (coefficients for V are similarly defined).

$$\beta_{u,i}|\kappa_{u,i} \sim GP\left(m_{u,i}(\cdot),\ s_{u,i}^2 K_{\nu_{u,i}}(\cdot,\cdot;\ \kappa_{u,i})\right) \tag{3.4.6}$$

We assume that the $\beta_{u,i}$ for $i = 0, 1, 2$ are *a prior* all independent of one another. From our exploratory data analysis, our naive fit for Equation 3.4.4 shows that the intercept field $\beta_{u,0}$ is noticably coarser than $\beta_{u,1}$ and $\beta_{u,2}$. We fix $\nu_{u,0} = 1$ and $\nu_{u,1} = \nu_{u,2} = 2$ after visually comparing these fields to simulated Matern fields generated from the model by *Lindgren et al.* (2011). Similar to the residual fields, each $\kappa_{u,i}$ is the inverse of the range parameter in standard Matern Covariance specification. $s_{u,i}$ is the standard deviation for the coefficient field but will be fixed to be one half of the upper bound for the largest possible $\beta_{u,i}$ values (details in Appendix A.1). $m_{u,i}(\cdot)$ is the mean function which is a constant 0 for $i = 0,\ 2$. For $\beta_{u,1}$, $m_{u,1}(s) = \frac{f_{lat}(s)}{\left(\rho(f_{lat}^2(s)+\gamma^2\right)}$ based on Equation 3.4.2. The calculations for ρ, f_{lat}, and γ are explained in Appendix A.1. In our experience, the results are not sensitive to changes in $m_{u,i}(\cdot)$ and $s_{u,i}$ as long as $s_{u,i}$ is fixed sufficiently large.

Notice that β_0 and the residual field have the same Matern smoothness. To ensure identifiability, fixing $s_{u,0}$ is necessary for allowing $\sigma_{u,\epsilon}$ to vary spatially.

Finally, we detail our prior specifications to complete the BHM. The prior model for the parameters for the coefficient and residual field is

$$\begin{aligned} \log(s_{u,i}^2 \kappa_{u,i}^{2\nu_i}) = c_{u,i} &\sim N\left(a_{u,i},\ b_{u,i}^2\right) \\ \log(\kappa_{u,\epsilon}) &\sim N\left(a_\epsilon, b_\epsilon^2\right) \end{aligned} \tag{3.4.7}$$

Instead of modeling $\kappa_{u,i}$ directly, we model $c_{u,i}$. In our experience, $c_{u,i}$ is robust to different prior specifications and *Zhang* (2004) has shown that $c_{u,i}$ is consistently estimable for Matern fields. $\kappa_{u,\epsilon}$ was treated differently since the spatially varying $\sigma_{u,\epsilon}$ does not allow the same definition. The hyperparameters are again chosen to create weakly informative priors (details in the Appendix A.1) and our results are overall quite robust to these specifications.

The above relationships between V and pressure gradient is similarly defined by replacing the subscripts of u to v. One key here is although the covariance for the $\beta_{u,i}$ and $\beta_{v,i}$ fields are all *a priori* stationary Materns. Given the number of replicates and the size of the data, however, we expect the posterior to reflect the non-stationarity in the data even though our prior is stationary. Moreover, *Reich et al.* (2011) showed little prediction improvement by using flexible nonstationary priors. We discuss possible extentions for this in Section 3.7.

3.5. Fitting the Model

As mentioned in Section 3.4, fitting a spatially varying coefficient model is computationally demanding. This section details how we fit the model using the methods from *Lindgren et al.* (2011).

We use MCMC methods to perform our inference on this model. . Since U and V are assumed to be conditionally independent given P, we can obtain the posterior for $P(\sigma_{u,\epsilon}, c_{u,i}, \kappa_{u,\epsilon}, \beta_u | U, P)$ and $P(\sigma_{v,\epsilon}, c_{v,i}, \kappa_{v,\epsilon}, \beta_v | V, P)$ separately. For the implementation details we will focus on U alone and drop the subscript but the procedure is the same for V.

To clarify some notation, our model starts with the linear model

$$U_t = X_t^T \beta + \epsilon_t$$

for $t = 1, \ldots, N$ where N is the number of years in the data. Let n indicate the number of spatial locations then for each year t, U_t is a $n \times 1$ vector that contains the wind speed, ϵ_t is a $n \times 1$ vector that contains the residuals, and X_t^T is a $n \times (3n)$ matrix that contains the pressure gradients and intercept. More specifically, $X_t^T = \left[I_{n\times n}\ :\ diag(\frac{\partial P_t}{\partial x})\ :\ diag(\frac{\partial P_t}{\partial y}) \right]_{n\times(3n)}$ where $diag(\frac{\partial P_t}{\partial x})$ indicate a diagonal matrix where the diagonal holds the n pressure gradient values. Naturally $\beta^T = \left(\beta_0^T, \beta_1^T, \beta_2^T\right)$ is a vector of length $3n$ where $\beta_i = (\beta_i(s_1), \ldots, \beta_i(s_n))^T$ for $i = 0, 1, 2$ each contains the spatially varying coefficients.

Since our data only exists over finite locations, β is a multivariate Gaussian with mean m and covariance Σ_β. The $3n \times 1$ mean vector is $m^T = (m_0^T, m_1^T, m_2^T)$ where $m_i = (m_i(s_1), \ldots, m_i(s_n))^T$. Σ_β is a $3n \times 3n$ block diagonal matrix composed of Σ_{β_0}, Σ_{β_1}, and Σ_{β_2}. Each Σ_{β_i} is the prior covariance matrix for β_i based on the Matern covariance function specified in Section 3.4. Similarly we define Σ_ϵ to be the $n \times n$ covariance matrix for the residuals, ϵ. Lastly, we introduce the short hand notation $c = \{c_0, c_1, c_2\}$ for brevity.

To obtain the posterior samples from $P(\sigma_\epsilon, c, \kappa_\epsilon, \beta|U)$, we implement a Gibbs sampler by repeatedly sampling from $p\left(\sigma_\epsilon|c, \kappa_\epsilon, U, \beta\right)$, $p\left(c, \kappa_\epsilon|\sigma_\epsilon, U, \beta\right)$, and $p\left(\beta|\sigma_\epsilon, c, \kappa_\epsilon, U\right)$. The last term has an analytical distribution we can easily sample where the former two terms do not. The former two distribution will be sampled using the adaptive Metropolis Hastings algorithm (*Shaby and Wells*, 2010) in blocks.

The analytical term is just a multivariate Gaussian distribution

$$\begin{aligned} & p\left(\beta|\sigma_\epsilon, c, \kappa_\epsilon, U\right) \\ \propto\ & \underbrace{p\left(\beta|\sigma_\epsilon, c, \kappa_\epsilon\right)}_{MVN} \underbrace{p\left(U|\beta, \sigma_\epsilon, c, \kappa_\epsilon, U\right)}_{MVN} \end{aligned}$$

The full conditional mean for this MVN is

$$(\Sigma_\beta^{-1} + \sum_{t=1}^{N}\{X_t^T \Sigma_\epsilon^{-1} X_t\})^{-1}(\Sigma_\beta^{-1} m + \sum_{t=1}^{N} X_t^T \Sigma_\epsilon^{-1} U_t)$$

The full conditional covariance matrix is

$$(\Sigma_\beta^{-1} + \sum_{t=1}^{N}\{X_t^T \Sigma_\epsilon^{-1} X_t\})^{-1}$$

All the matrices here are of considerable size (Σ_β is 24576×24576!) but the computation involves only the precision matrices instead of the covariance matrices. This allows us to implement the Gaussian Markov Random Fields (GMRF) algorithms provided in *Lindgren et al.* (2011) where the exact sampling techniques can be found in *Rue and Held* (2005).

GMRF methods rely on the Markov property for Gaussians random variables that induces sparse precision matrices (*Rue and Held*, 2005; *Lindgren et al.*, 2011). This allows the use of sparse matrix algorithms to speed up computation. The biggest limitation for GMRF methods is the Gaussian or latent Gaussian assumption within the model. However, our smooth spatial fields are modeled as GPs which satisfies this assumption. The other issues with GMRF methods was its difficulty in producing precision matrices that corresponded to smooth spatial fields (*Wall*, 2004; *Rue and Held*, 2005). Fortunately, *Lindgren et al.* (2011) discovered the link between GMRF and smooth Gaussian fields via Stochastic Partial Differential Equations (SPDEs). This resolves the computational challenges while avoiding ad hoc choices with low rank methods. For details, many packages and examples are available at the R-INLA project `http://www.r-inla.org/`.

The next term in the Gibbs sampler is $p\left(\sigma_\epsilon|c, \kappa_\epsilon, U, \beta\right)$. Unfortunately, the prior log normal distribution for σ_ϵ does not have the same conjugate relationship with U so we implement the adaptive Metropolis Hastings algorithm instead. The density for the full conditional is

$$\begin{aligned} & p\left(\sigma_\epsilon | c, \kappa_\epsilon, U, \beta\right) \\ \propto \quad & p(\sigma_\epsilon | c, \kappa_\epsilon, \beta) p(U | \beta, \sigma_\epsilon, k_\epsilon) \\ = \quad & p(\sigma_\epsilon) p(U | \beta, \sigma_\epsilon, k_\epsilon) \\ = \quad & \left|\Sigma_\sigma^{-1}\right|^{\frac{1}{2}} \exp\left[-\frac{1}{2}\left(\log(\sigma_\epsilon)\right)^T \Sigma_\sigma^{-1}\left(\log(\sigma_\epsilon)\right)\right] \\ & \left|\Sigma_\epsilon^{-1}\right|^{\frac{N}{2}} \exp\left[-\frac{1}{2}\sum_{t=1}^{N}\left(U_t - X_t\beta\right)\Sigma_\epsilon^{-1}\left(U_t - X_t\beta\right)\right] \end{aligned}$$

Σ_σ is the covariance matrix based on the covariance function specifications in Equation 3.4.5. Notice that σ_ϵ is a high dimensional smooth field which is difficult to sample with adaptive Metropolis Hastings. The challenge comes from updating the proposal covariance matrix which is dense and high dimensional. Instead, we run the adaptive Metropolis Hastings algorithm by only updating the proposal variance parameter while fixing the proposal precision matrix. We discuss the details for this section in Appendix A.3.

The final term in the Gibbs sampler is $p\left(c, \kappa_\epsilon | \sigma_\epsilon, U, \beta\right)$. We again draw samples using the adaptive Metropolis Hasting algorithm since no analytical distribution is known. The full conditional can be partitioned into independent blocks for faster mixing

$$\begin{aligned} & p\left(c, \kappa_\epsilon | U, \beta, \sigma_\epsilon\right) \\ = \quad & p\left(\kappa_\epsilon | U, \beta, \sigma_\epsilon\right) \prod_i p\left(c_i | \beta_i\right) \end{aligned}$$

By assuming the different years are independent replicates of one another, the first block becomes

$$\begin{aligned} & p\left(\kappa_\epsilon | U, \beta, \sigma_\epsilon\right) \\ \propto \quad & p\left(\kappa_\epsilon | \beta, \sigma_\epsilon\right) \prod_{t=1}^{N} p\left(U_t | \beta, \sigma_\epsilon, \kappa_\epsilon\right) \\ = \quad & p\left(\kappa_\epsilon\right) \prod_{t=1}^{N} p\left(U_t | \beta, \theta_\epsilon, \kappa_\epsilon\right) \\ \propto \quad & \left|\Sigma_\epsilon^{-1}\right|^{\frac{N}{2}} \exp\left[-\frac{1}{2}\sum_{t=1}^{N}\left(U_t - X_t\beta\right)\Sigma_\epsilon^{-1}\left(U_t - X_t\beta\right)\right] p(\kappa_\epsilon) \end{aligned}$$

$p\left(c_i | \beta_i\right)$ for $i = 0, 1, 2$ breaks down similarly

$$\begin{aligned} & p\left(c_i | \beta_i\right) \\ \propto \quad & p\left(\beta_i | c_i\right) p\left(c_i\right) \\ \propto \quad & \left|\Sigma_{\beta_i}^{-1}\right|^{\frac{1}{2}} \exp\left[-\frac{1}{2}\left(\beta_i - m_i\right)^T \Sigma_{\beta_i}^{-1}\left(\beta_i - m_i\right)\right] p\left(c_i\right) \end{aligned}$$

The product form implies we can run the adaptive Metropolis Hasting algorithm separately in blocks for each parameter set.

Overall, we have two adaptive Metropolis Hastings routines embedded within a three step Gibb Sampler. The adaptive Metropolis Hastings algorithm is appealing because it adjusts its proporsal distribution to control the acceptance rate. This yields posterior samples that are not strongly correlated over different iterations which may occur in standard Gibb samplers such as those in *Milliff et al.* (2011). Moreover, the adaptive Metropolis Hasting algorithm is relatively straightforward to code in multidimensions than other algorithms such as slice samplers used *Gelfand et al.* (2003).

To initialize the algorithm, good starting values are important. In our experience, first running a short MCMC with reasonable starting values followed by a long chain produces promising results. The final iteration in the short chain yields good starting values where the covariance based on the short chain provides efficient proposal distributions for the long chain. We re-initiate the Gibb Sampler after stopping the short chain then determine convergence by evaluating the parameter trace plots. We discuss the selection of reasonable starting values in Appendix A.2.

3.6. Results and Evaluation

After obtaining posterior samples from the MCMC chain we now validate our model. Figure 3.6.1 and Figure 3.6.2 show the posterior means for β_u and β_v for the Winter wind fields (the β_u and β_v for Summer wind fields are shown in Appendix). One quick sanity check is the agreement with Equation 3.4.2. Given that the Coriolis force switches sign at the equator, the sign change for $\beta_{u,1}$ and $\beta_{v,1}$ at the equator is promising. Moreover, the Coriolis effect is inversely related to the coefficients so the higher magnitudes at the equator and decreasing magnitudes towards the polar regions is reassuring. Another promising aspect is the fact that the $\beta_{u,1}$ and $\beta_{v,1}$ are larger in magnitude over the ocean than land. This confirms our understanding about the effect of friction on wind velocity. The coeffieint process also shows a discontinuous behavior when the topography changes between land and sea which is a feature we wanted to capture in this model. Another assuring fact is that $\beta_{u,2}$ and $\beta_{v,2}$ are consistently negative if not zero. This also agrees qualitatively with the results in *Milliff et al.* (2011).

Besides the qualitative agreement with Equation 3.4.2, prediction accuracy is also useful to help evaluate models. Figure 3.6.3 shows the predicted wind velocities vs the actual wind velocities in the 10 years we set aside at the beginning. The predicted wind velocities is $E(X\beta + \epsilon|U) = XE(\beta|U)$. In other words, we average the posterior samples of β and apply them to the last 10 years of X_t to predict the wind velocities. Relative to the overall variability in the wind components, the prediction error is quite small.

Lastly, for each year, we construct a 95% credible band based on 374 posterior sample wind fields. The credible band will cover 95% of the posterior samples at all 8192 locations. This band summarizes the general center and spread of possible wind fields under the posterior distribution. Unfortunately this band is not uniquely defined. To construct a reasonable credible band, we center the band at the location-wise average over the posterior samples. We then compute the corresponding location-wise SD to create a symmetric credible band. We then expand this band by a factor of 1.04 iteratively until we obtain a credible band that covers 95% of all posterior samples. Figure 3.6.4 shows the width of this band is larger around regions where the land-sea surface changes, near the polar regions,

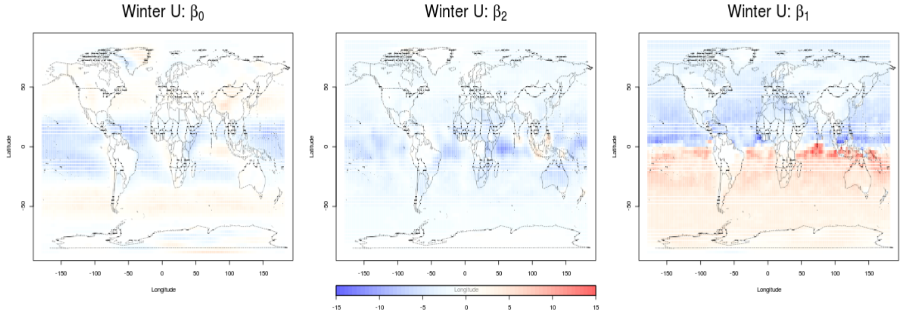

FIGURE 3.6.1. Posterior means of β_U terms for Winter. $\beta_{u,1}$ has a larger magnitude as we expected. Moreover, the sign change of the coefficients and decreasing magnitude away from the equator is consistent with Equation 3.4.2. The coefficients "jump" when the underlying surface changes between land and sea. This is consistent with our observations of surface wind behavior.

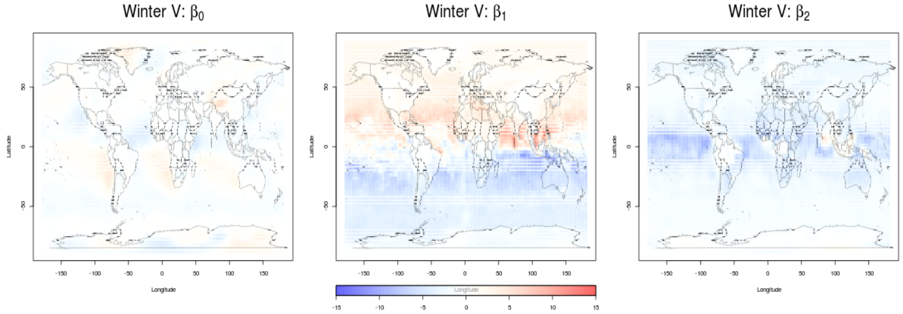

FIGURE 3.6.2. Posterior means of β_V terms for Winter. $\beta_{v,1}$ has a larger magnitude as we expected. Moreover, the sign change of the coefficients and decreasing magnitude away from the equator is consistent with Equation 3.4.2. The coefficients "jump" when the underlying surface changes between land and sea. This is consistent with our observations of surface wind behavior.

and Greenland. This is a desirable property since the wind velocities indeed become more sporadic around these locations. For each year, over 99% of the locations are all within this credible band. The coverage rate over 10 years for each location is quite high which suggests that our posterior samples resemble the actual wind fields from the climate model with a few exceptions.

Recall one of the goals for our model is to compare wind fields from different distributions or climate models. To detect these differences, we can check if the credible interval for U in the Winter covers the samples for U in the summer. Figure 3.6.5 is a strong contrast relative to Figure 3.6.4 where few locations are covered in the credible interval over 10 years. This shows that our high coverage rate in Figure 3.6.4 is not a result of overly inflated credible band widths.

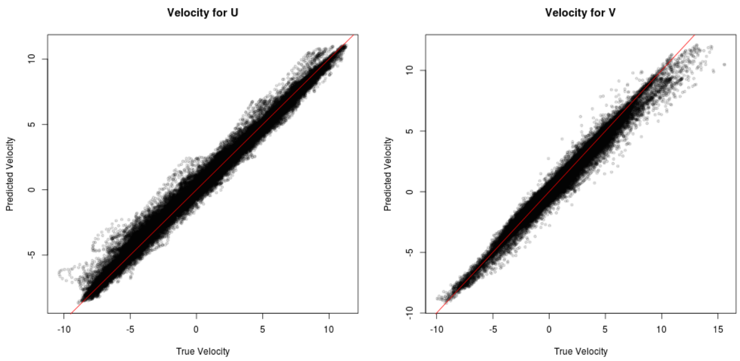

FIGURE 3.6.3. 10 Year Surface Prediction vs Wind Velocity. The vertical deviation from the 45 degree line is small relative to the overall variability in the data. This suggests that the predictions are quite good.

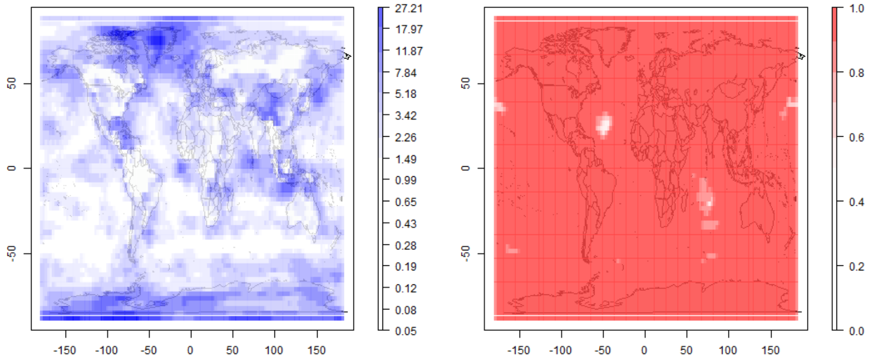

FIGURE 3.6.4. Credible Interval for Winter U and Location-wise Coverage Rate over 10 years. Left figure shows the width of the credible interval for Year 1 on the log scale. Right figure shows, for each location, the proportion of points covered in their respective credible intervals over all 10 years. Most locations have very narrow credible band widths which suggests the posterior samples have low uncertainty. Moreover, the coverage rate over 10 years is high over most locations with a few exceptions. This shows that our credible interval captures most of the variability in the data.

3.7. Discussion

Overall, we developed a statistical model for surface winds that extends over the entire globe. Motivated by the geostrophic relationship, our model efficiently handles the multivariate and the spatial heterscedastic nature of surface winds.

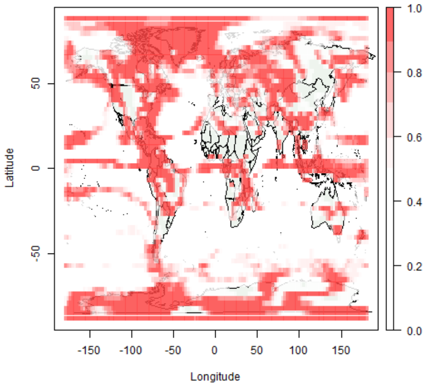

FIGURE 3.6.5. Credible Interval for U in Winter does not capture U from Summer. The low location specific coverage rate implies that our coverage rate in Figure 3.6.4 is not a result of overly inflated credible band widths.

However, there are many possible improvements for this model. First of all, throughout we have assumed that the sea level pressure, P, was given. This seems like a strong condition especially because we have relied on the pressure gradient to explain most of the variability within the wind components. However, a quick examination of the sea level pressure shows that an assumption of stationarity is reasonable. Thus modeling the sea level pressure should be straightforward to incorporate into the hierarchical model. Moreover, modeling the pressure gradient should provide better estimates for the pressure gradient since the derivative process for a GP is another GP that depends on the same parameters (*Rasmussen et al.*, 2006).

Surprisingly, fitting the varying coefficient model is more difficult over smaller regions on the globe with non-circular boundaries. These regions are commonly used in regional climate models (RCM) for higher resolution climate patterns. The challenge comes from specifying the boundary conditions when constructing the precision matrix on a finite region. By working over the globe, we exploit the circular boundary condition and avoided this question. One solution is to embed the boundary entries with the correct precision values obtained from inverting the corresponding covariance matrix (*Rue and Held*, 2005). Unfortunately, inverting the boundary regions for a dense covariance matrix is still often very computational for large datasets. Another proposal is to greatly extend the boundary and embed the region of interest on the globe. The model then reduces back to our global model with circular boundary conditions.

In this chapter we have employed a stationary prior for the β fields even though we know *a priori* that the fields are non-stationary based on lattitude and topography. This choice was mostly due to computational convenience. We relied on the data to inform the posterior distribution of β to be non-stationary. There are many ways to construct a non-stationary prior for the β field (see e.g. *Higdon* (1998); *Fuentes* (2001); *Paciorek and Schervish* (2006); *Reich et al.* (2011)), but most are not computationally feasible for datasets the size used in this chapter. With the GMRF representation, we recommend constructing nonstationary covariances using additive methods based on the topography. In other words, we could model $\Sigma_\beta = \Sigma_{globe} + 1_{land}\Sigma_{land}1_{land}$ where 1_{land} is a diagonal matrix with 0-1 entries along the diagonal indicating whether s_i is over land. The corresponding precision matrix can be expressed using the Sherman-Morrison-Woodbury formula that will remain sparse. This construction is theoretically a special case of the feature-dependent nonstationary covariance models by *Reich et al.* (2011). However, the GMRF methods with sparse matrix routines make these nonstationary models much more computationally feasible.

Throughout, we have ignored the temporal effects here even though we had wind fields from different seasons. In our experience, the β fields from year to year are stable but the season to season variation is noticable. *Gelfand et al.* (2010) compiled several approaches used to extend spatial models to incorpoate the temporal dimension. We again leave this for future developments.

CHAPTER 4

Ranking in Hierarchical Multilabel Classification using Local False Discovery Rate

4.1. Introduction:

This chapter considers the question of classification where each subject can be assigned multiple labels and the label assignments have to respect a given hierarchical relationship, also known as hierarchical multilabel classification (HMC). HMC has many applications in fields like astronomy (*Richards et al.*, 2011), biology (*Barutcuoglu et al.*, 2006; *Huang et al.*, 2010; *Dimitrovski et al.*, 2011), and computer science (*Sun and Lim*, 2001). However, often custom classifiers have been built to determine the status for certain topics, e.g. breast cancer (*Sotiriou et al.*, 2003) and Alzheimer's disease (*Klöppel et al.*, 2008). These single label classifiers are highly tailored to the subject matter but their label assignments may not be consistent with respect to the hierarchy. Instead of discarding these past efforts, we approach HMC from the view of constructing a consistent classifier based on outputs from existing single label classifiers.

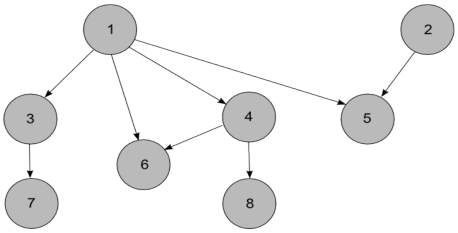

FIGURE 4.1.1. Example hierarchy for simulation: directed acyclic graph (DAG)

For example, Figure 4.1.1 shows a possible hierarchy between eight nodes (or classes). Under the hierarchical constraint, if node 1 is negative, then all its descendants (all nodes but node 2) must also be negative. On the other hand, if node 8 is positive, then its ancestors (node 1 and 4) must also be positive. If a single label classifier is trained for each node, then it is possible that the resulting assignments may not respect the hierarchical constraints.

HMC problems often have additional requirements beyond the hierarchical constraint. To understand the type of HMC we are targeting, it is convenient to put our problem in the context of disease diagnosis. Besides the hierarchical constraint, each patient can be diagnosed for diseases down different paths of the hierarchy since having cancer does not prevent one from having diabetes. Moreover, one may not have the terminal form of a disease so the diagnosis may not reach the leaf nodes in the hierarchy. Lastly, the hierarchy follows a directed acyclic graph (DAG) where each disease can have multiple parent diseases (Unified Medical Language System *Bodenreider*, 2004). This is in contrast to supernovae classification done by *Richards et al.* (2011). Supernovae are separated into exclusive classes down a single path on the hierarhcy. Each classification necessarily reaches the leaf node of the hierarchy and the hierarchy follows a tree structure where each node has at most one parent node.

One unique issue with HMC is the prevalence of imbalanced data sets. As we move down the hierarchy, the proportion of positive example drops quickly relative to negative examples. Classifiers are then trained on imbalanced data sets where most examples are negative cases and few positive cases are available. This often causes methods that minimize classification error to ignore the positive cases altogether but still achieve high classification performance. In response to this, most papers promote the use of precision and recall to evaluate HMC methods which focuses on the positive cases (*Vens et al.*, 2008; *Silla Jr and Freitas*, 2011). With this focus, we will show that our method can yield high precision performance relative to existing HMC methods.

The main contribution of this chapter introduces ranking into HMC under a statistical framework. Ranking is used in multilabel classification (MC) where the labels are independent from one another *Tsoumakas and Katakis*, 2007; *Jiang et al.*, 2013. The approach first ranks the label assignments by their likelihood of a positive status then assigns the top k ranked labels as positive. In HMC, ranking is difficult because of the constraint and dependencies between the labels. We propose to rank each label by its chance of being positive given all outputs, i.e. $p(Q_j = 1|S_1, \ldots, S_p)$ for $j = 1, \ldots, p$ (p is the number of classes, Q_j is the label for class j, and S_j is the single label classifier output for class j). For classification, however, instead of treating the top k ranked labels as positive, we classify only the highest ranked unassigned label by a cutoff, then update the rankings, and repeat. More specifically, after classifying the highest ranked label, e.g. $\hat{Q}_z = q$, then we update the ranking by $P(Q_j = 1|S_1, \ldots, S_p, Q_z = q)$ for $j \neq z$. The ranking avoids single label classifier outputs with weak signals where the updating naturally accounts for the dependency structure. We will show that this ranking also gives useful insights for understanding the difference between MC and HMC.

The structure of this chapter is summarized here. Section 4.2 reviews the HMC literature. Section 4.3 introduces our method along with estimation details. Section 4.4 compares the performance of our method to predictive clustering trees under a simulation study. Section 4.5 applies our method on the disease classification dataset shown in *Huang et al.* (2010). Finally, Section 4.6 discusses the limitations and improvements for our ranking framework and sequential classification method.

4.2. Literature Review

There are many existing HMC methods that tackle different problems with different approaches. *Silla Jr and Freitas* (2011) give a good overview of the different HMC methods which we summarize below.

There are three general approaches to HMC: flat classification, local classification, and global classification. Flat classification ignores the hierarchy and performs classification on all the leaf-nodes. Naturally, the ancestors of each leaf-node is assumed to be positive if the leaf node is positive. The appeal for this method is its simplicity but it is only appropriate for cases when the assigned label must reach the leaf nodes in the hierarchy.

Local classification trains separate classifiers locally then adjusts or propagates the classifications with respect to the hierarchy. The local classifiers can be a collection of single label classifiers, multi-label classifiers for a level in the hierarchy, or multi-label classifiers for nodes that share a parent. Each of these local classifiers will give assignments over the hierarchy for different nodes without accounting for the assignments from one another. To ensure consistency with the hierarchy, a second stage process is usually built into the local classification routines.

There are two main approaches for ensuring the hierarchical constraints for the label assignments. The heuristic approach is a top-down process where lower level classifications are only performed if all of their parent nodes are first classified as positive. This avoids inconsistencies in the hierarchy and saves computation for low level classes in the hierarchy (*Koller and Sahami*, 1997; *Wu et al.*, 2005). This, however, suffers from blocking issues where misclassifications at high level nodes drastically hinder the performance of the classifier. *Sun et al.* (2004) proposed several heuristics to remedy blocking but none are theoretically driven.

In contrast, *Barutcuoglu et al.* (2006) used a Bayesian framework to obtain the distribution of all possible label combinations given all the existing classifier outputs, i.e. $p(Q_1, \ldots, Q_p | S_1, \ldots, S_p)$. This resolves inconsistencies over the hierarchy and theoretically provides an optimal label assignment under a cohesive framework. The downside to their method is in the implementation. Numeric underflow issues and computational time both increase as p grows. Distributional assumptions were made to compute the necessary probabilities which are often unknown in practice. Moreover, in our experience, outputs from poor classifiers often introduce more error for neighboring label assignments. Thus using all classifier scores is sometimes sub-optimal. We will motivate our method under a similar statistical framework that avoids most of these issues.

Instead of adjusting for inconsistencies after the classifiers are built, global classifiers consider the entire hierarchy during the training phase of the classifier. *Vens et al.* (2008) argued that global classification methods are more efficient because they require fewer decision rules overall than local classification methods. The general approach is to find efficient decision rules that quickly split the training data into similar clusters with similar hierarchical labels. Since the labels respect the entire hierarchical structure, the classifier will build decisions implicitly respecting the hierarchy. The challenge for these methods comes from the computational demands that exhaustively searches for good decision rules.

Another way to approach HMC is to perform a hypothesis test for the status of each label. From the multiple hypothesis testing literature, *Yekutieli* (2008) proposed a top-down procedure that bounds the overall false discovery rate when testing hypotheses that respect a hierarchy. If we consider each null hypothesis to be the label is negative then this procedure is also applicable for classification. Unfortunately, the top-down procedure still suffers from blocking issues as the other top-down classification methods. Theoretically bounding the false discovery rate does not guarantee high precision and thus this may not be best for classification purposes.

Our method falls under the local classification methods and relies on the outputs from existing single label classifiers. We believe this is a reasonable approach since local classifiers can be more tailored to the data and it prevents duplicated efforts. Moreover, using local classifiers sequentially often avoids the influences of poor classifier outputs. Using the hierarchy, our method often allows informative outputs from quality classifiers to ignore the noisy outputs from poor classifiers. We will compare our sequential approach to the global classifier based on bagging predictive clustering trees. To make the algorithms comparable for our problem setting, the global classifier can only utilize the outputs from the same single label classifiers. The predictive clustering trees have shown great success in HMC and few methods in the literature have been able to outperform their results (*Silla Jr and Freitas*, 2011).

Our ranking approach is largely inspired by the work in *Jiang et al.*, 2013. They developed a ranking scheme for MC that optimizes precision for any recall value. The quantity to rank the label assignments was coined Local Precision Rate (LPR) which is theoretically equivalent to the complement of Local False Discovery Rate ($1-lfdr$) by *Efron and Tibshirani* (2002). $1-lfdr$ has an empirical Bayes interpretations: the chance of a positive status given the classifier output, i.e. $p(Q_j = 1|S_j)$. For brevity we will denote $1 - lfdr$ as $lfdr'$. The biggest challenge with $lfdr'$ is in its computation requires density estimations. The derivation of LPR offers a robust method of estimating the necessary density ratios directly while avoiding densities estimation with limited samples. In this chapter, we also extend the existing estimation methods for LPR which slightly improves the results from the original work. Overall, our ranking extends LPR ot $lfdr'$ to the HMC setting.

4.3. Methodology

We propose a sequential classification method by using outputs from existing single label classifiers. Single label classifiers often output a score that reflects the confidence or uncertainty for a label assignment of being positive (e.g. logit probabilities from a logistic regression). However, these are often not comparable between different classifiers so ranking according to these raw outputs will be sub-optimal. To achieve consistent ranking, we rank each label assignment according to the posterior probability of the label being positive given all classifier outputs. This ranking creates an order for the sequential classification method. We then classify the highest ranked label according to a cutoff, update the ranking according to all the previous classifications, and repeat until all labels are assigned. The ranking produces high precision assignments by avoiding outputs with uninformative signal where the updates will ensure the assignments are consistent with the hierarchy.

Here we first define some notation and the general problem setting. There are p nodes with n subjects. Without loss of generality, assume each class has one best classifier so there are p classifiers. The true status for each subject i for class k is denoted as $Q_{k,i} \in \{0, 1\}$ where 0 and 1 respectively indicate a negative and positive status. We will assume the classifier scores for subject i and node k, $S_{k,i}$, to be a mixture distribution depending on the true label status. Overall, we believe the data is generated as

$$\begin{aligned} Q_{k,i} &\sim \begin{cases} 0 & \text{if } Q_{j,i}=0 \text{ for any } j \in par(k) \\ Bernoulli(p_k) & \text{otherwise} \end{cases} \\ S_{k,i}|Q_{k,i} &\sim Q_{k,i}F_{k,1} + (1-Q_{k,i})F_{k,0} \end{aligned}$$

where $par(k)$ denote the parents of node k where $p_k,\ F_{k,1},\ F_{k,0}$ is unknown for all $k = 1, \ldots, p$. We further define $\pi_k = P(Q_{k,i} = 1)$ for $i = 1, \ldots, n$ which is equivalent to p_k if there were no hierarchy. Our calls/predictions/classifications for each status is denoted $\hat{Q}_{k,i} \in \{0, 1\}$. For simplicity below, we will omit the i subscript in the follow derivations.

4.3.1. Ranking Framework. The ranking quantity proposed by *Jiang et al.*, 2013 is $lfdr' = 1 - lfdr$. This has three different expressions

$$\begin{aligned} lfdr'(S_k) &= \frac{\pi_k f_{k,1}(S_k)}{\pi_k\ f_{k,1}(S_k) + (1-\pi_k)\ f_{k,0}(S_k)} \\ &= \left[1 + \left(\frac{1-\pi_k}{\pi_k}\right)\ \frac{f_{k,0}(S_k)}{f_{k,1}(S_k)}\right]^{-1} \qquad (4.3.1) \\ &= p(Q_k = 1 \mid S_k) \end{aligned}$$

This quantity is difficult to estimate in practice since HMC problems suffer from imbalanced data sets where the number of positive samples are extremely limited. In particular, the estimation of $f_{k,1}$ is very difficult and problematic. *Jiang et al.*, 2013 however proposed a robust method to estimate the $lfdr'$ directly without computing the individual densities. We will elaborate on this detail in the implementations in Section 4.3.3.

An important result from *Jiang et al.*, 2013 is that $lfdr'$ provides an optimal ranking when assigning labels using a uniform cutoff in the MC setting. Unfortunately, the assignments from a uniform cutoff may produce label assignments that are inconsistent with the hierarchy. To extend $lfdr'$ to the hierarchical setting, we could rank the hierarchical label assignments for $k = 1, \ldots, p$ using a similar quantity

$$\begin{aligned} & p\left(Q_k = 1 \mid S_1, \ldots, S_p\right) \\ = & \frac{p(Q_k = 1 \mid S_k)\ p(S_{-k} \mid S_k,\ Q_k = 1)}{p(S_{-k}|\ S_k)} \\ = & [lfdr'(S_k)]\,\frac{p(S_{-k} \mid S_k,\ Q_k = 1)}{p(S_{-k} \mid S_k)} \qquad (4.3.2) \end{aligned}$$

where $S_{-k} = \{S_1, \ldots, S_{k-1}, S_{k+1}, \ldots S_p\}$, i.e. all but the output from classifier k. Equation 4.3.2 shows that our ranking is just an adjusted value of the optimal ranking in the MC setting. For the adjustment term, notice if S_k is high for a quality classifier, conditioning on S_k essentially implies $Q_k = 1$ then the ranking in the MC setting will agree with the ranking in HMC setting.

Based on the mixture distribution assumption, we can expand the adjustment term as

$$\frac{p(S_{-k} \mid Q_k = 1)}{P(Q_k = 1|S_k)\; p(S_{-k} \mid Q_k = 1) + (1 - P(Q_k = 1|S_k))\; p(S_{-k} \mid Q_k = 0)}$$

$$\left[lfdr'(S_k) + (1 - lfdr'(S_k))\; \frac{p(S_{-k} \mid Q_k = 0)}{p(S_{-k} \mid Q_k = 1)} \right]^{-1} \tag{4.3.3}$$

$$\left[lfdr'(S_k) + (1 - lfdr'(S_k))\; (\frac{\pi_k}{1 - \pi_k})\; \frac{p(Q_k = 0 \mid S_{-k})}{p(Q_k = 1 \mid S_{-k})} \right]^{-1} \tag{4.3.4}$$

Equation 4.3.3 shows that the adjustment term is bounded in the interval $\left[0, [lfdr'(S_k)]^{-1}\right]$. So assignments that have high $lfdr'$ values will be affect by neighboring nodes less than nodes assignments with low $lfdr'$ values. This agrees with the intuition that the ranking in the HMC setting will be similar to the ranking in the MC setting for high values of $lfdr'(S_k)$.

For now, assume that we could compute Equation 4.3.2 (implementation details will be elaborated in Section 4.3.3). What should happen to the ranking if we were informed that Q_{k^*} was positive or negative? Without loss of generality, let's assume we were informed $Q_{k^*} = 1$. With this knowledge, we should update the ranking for $j \in \{1, \ldots, p\} \backslash k^*$ as

$$p\left(Q_j = 1 \mid S_1, \ldots, S_p,\; Q_{k^*} = 1\right)$$

Notice that if j is an ancestor node of k^* then this probability is 1. Thus the ranking for all the ancestor nodes of k^* jumps to the top of the ranking and will be immediately be classified as 1 as well. On the other hand, if $\tilde{Q}_{k^*} = 0$ then this probability is 0 for any j that is a descendant node of k^* which must be negative. This update naturally enforces the hierarchical constraints and prevents possible inconsistencies.

After classifying the ancestor nodes as positive, denote $D = k^* \cup ancestors\{k^*\}$ then for a non-ancestor node of k^*, the ranking quantity becomes

$$p\left(Q_j = 1|S_j, Q_D = 1\right) \frac{p\left(S_{-j}|S_j, Q_D = 1, Q_j = 1\right)}{p(S_{-j}|Q_D = 1, S_j)}$$

which is similar to Equation 4.3.2 factoring in the effects from knowing $Q_{k^*} = 1$. Under our generative model for the data, denote $\pi_{j|D} = p(Q_j = 1|Q_D = 1)$ then the conditional independence suggests that

$$\begin{aligned}
& p\left(Q_j = 1 \mid S_j,\; Q_D = 1\right) \\
= \; & \frac{\pi_{j|D} p(S_j \mid Q_j = 1,\; Q_D = 1)}{p(S_j \mid Q_D = 1)} \\
= \; & \frac{\pi_{j|D} p(S_j \mid Q_j = 1)}{\pi_{j|D} p(S_j \mid Q_j = 1) + (1 - \pi_{j|D}) p(S_j \mid Q_j = 0)} \\
= \; & \left[1 + \frac{(1 - \pi_{j|D})}{\pi_{j|D}} \frac{f_{j,0}(S_j)}{f_{j,1}(S_j)} \right]^{-1} \qquad (4.3.5)
\end{aligned}$$

Notice that Equation 4.3.5 is the same as Equation 4.3.1 except the marginal probability has changed. This is useful because updating the marginal probability is a simple counting problem with the training data. The mixture assumption ensures that the density ratio will remain unchanged after assigning the labels for D.

The updating and conditional independence yields further desirable properties for the adjustment term as well.

$$\frac{p\left(S_{-j} \mid S_j,\ Q_D = 1,\ Q_j = 1\right)}{p(S_{-j} \mid Q_D = 1,\ S_j)} \quad = \quad \frac{p\left(S_{-\{j,D\}} \mid S_j,\ Q_D = 1,\ Q_j = 1\right)}{p(S_{-\{j,D\}} \mid Q_D = 1,\ S_j)} \tag{4.3.6}$$

where $-\{j, D\} = \{1, \dots, p\} \backslash j \cup D$. This means that once we have the labels for D, S_D no longer affects the future rankings. This simplifies the computation necessary for the adjustment term considerably if $|D|$ is large. More importantly, if the ancestor of node k^* was a poor classifier, then our ranking method will not be affected by its classifier outputs in further calculations.

4.3.2. Classification Algorithm. Under this ranking framework, we now lay out a corresponding classification algorithm. With efficiency and implementation constraints in mind, we exploit the simplifications and approximations whenever possible. The general form for the ranking quantity is

$$lfdr^*_{k,i}(D_i) = p\left(Q_{k,i} = 1 \mid S_1, \dots, S_p, Q_{D_i}\right)$$

where $D_i = \{j\ :\ j \in \{1, \dots, p\}\ \&\ \hat{Q}_{j,i} \text{ is assigned}\}$. Naturally $D = \emptyset$ if no classification has been made.

First, our proposed method begins with the estimation of $lfdr^*_{k,i}(\emptyset)$ for all $i = 1, \dots, n_{test}$ and $k = 1, \dots, p$ where n_{test} is the number of the examples in the test set. This can be separated into estimating $lfdr'$ and the adjustment terms. The estimation for $lfdr'$ involves nonparametric curve fitting for each node k using the training data which will be elaborated in Section 4.3.3. The adjustment on the other hand depends on $lfdr'$, π_k, and $\frac{P(Q_k=0|S_{-k})}{P(Q_k=1|S_{-k})}$ according to Equation 4.3.4. π_k can be easily estimated from the training data and $lfdr'$ will be estimated. The ratio however is difficult to estimate in general so we simplify it to

$$\frac{P(Q_{k,i} = 0|S_{-k,i})}{P(Q_{k,i} = 1|S_{-k,i})} = \frac{P(Q_{k,i} = 0|S_{fam(k),i})}{P(Q_{k,i} = 1|S_{fam(k),i})}$$

where $fam(k)$ are all the parents and children of node k. We naively estimate this quantity with a logistic regression using $S_{fam(k),i}$ as regressors along with an intercept. If a node does not have any neighbors, the quantity trivially becomes $\frac{1-\pi_k}{\pi_k}$.

This yields $lfdr^*_{k,i}(D_i)$ for $i = 1, \dots, n_{test}$ and $k = 1, \dots, p$. To ensure high precision, we first classify the highest ranked label assignment, $k^*_i = \arg\max_k \{lfdr^*_{k,i}(D_i)\ :\ k \notin D_i\}$ for $i = 1, \dots, n_{test}$. By starting with k^*, our method often by-passes blocking issues from poor classifiers or outputs with poor signal.

Secondly, we will make a classification decision based on $lfdr^*_{k^*,i}(D_i)$. For a given cutoff value, $\alpha_{cutoff} \in (0, 1)$, we classify values above α_{cutoff} as positive and negative otherwise. So our classification rule for k^*_i for $i = 1, \dots, n_{test}$ is

$$\hat{Q}_{k^*,i} = \begin{cases} 1 & \text{if } lfdr^*_{k^*,i}(D_i) > \alpha_{cutoff} \\ 0 & \text{otherwise} \end{cases} \tag{4.3.7}$$

so α_{cutoff} is applied only to the highest ranked value among those without an assigned label. To evaluate our method we simply apply many different cutoffs. However, we give a some guidance in selecting a single cutoff for researchers later in this section.

After classifying $\hat{Q}_{k^*,i}$, immediately we have $D_i = \{k_i^*\}$ and we need to update our ranking for $j \in \{1,\ldots,p\}\backslash k^*$. This update is a two stage process: enforcing the hierarchical constraints then updating $lfdr^*_{\cdot,i}(D_i)$ for all i. Recall that ancestors or descendants of the classified node will be immediately classified as positive or negative based on the hierarchical relationship. In other words, the first step in the update process is

(1) If a $\hat{Q}_{k^*,i} = 1$, then $\hat{Q}_{ancestors(k^*),i} = 1$ and $D_i = \{k^*, ancestors(k^*)\}$
(2) Otherwise, $\hat{Q}_{k^*,i} = 0$, then $\hat{Q}_{descendants(k^*),i} = 0$ and $D_i = \{k^*, descendants(k^*)\}$

for $i = 1,\ldots,n_{test}$.

The second step updates $lfdr^*_{\cdot,i}(D_i)$ for all i for the unclassified cases according to the assigned labels. Recall that Equation 4.3.5 implies that we can express $lfdr'$ value as a function of the density ratio, $\frac{f_{k_0}(S_{k,i})}{f_{k_1}(S_{k,i})}$, and the updated marginal probability. Since the update only affects the marginal probabilities, we only need to estimate the new marginal probability using the training data. Specifically, find all the cases in the training set that match the current assigned labels for each i, then update the weight $\pi_{k,i}$ according to the proportions seen in the training set. In other words, for some $i = 1,\ldots,n_{test}$

$$\pi^{new}_{k,i} = \frac{\sum_{j \in train} I(Q_{k,j} = 1 \ \&\ Q_{D_i,j} = \hat{Q}_{D_i,i})}{\sum_{j \in train} I(Q_{D_i,j} = \hat{Q}_{D_i,i})} \tag{4.3.8}$$

where *train* is the examples in the training set. One note is that when many nodes are classified, $|D_i|$ is large and the number of matching cases in the training set decreases. We recommend to only update this quantity if the number of matching cases exceeds some threshold (set to be 10 for all of our examples below). Notice that we also update the $lfdr'$ values for nodes that do not have a hierarchical relationship with D. The reason for this choice was because in practice the DAG may not be perfect and hidden factors could affect the prevalence of 2 unrelated classes. If the classes were indeed independent, the update should not affect the marginal probabilities by much.

As for the adjustment term, this term should be recomputed since Equation 4.3.6 suggests that S_{D_i} no longer affect the adjustment. However, the computational burden will increase quickly if we recompute the ratio term in the adjustment after each classification. Therefore we keep the ratio term fixed without updating it but update $lfdr'$ and π_k within the adjustment term.

The updates for $lfdr'$ and π_k produce new $lfdr^*_{\cdot,i}(D_i)$ for all i which creates a new ranking. To complete the classification for all labels, we iterate the process from selecting k_i^* from $k \in \{1,\ldots,p\}\backslash D_i$ for $i = 1,\ldots n_{test}$ again until all assignments are determined.

We summarize the algorithm in 7 steps:

(1) Estimate each $lfdr^*_{\cdot,i}(D_i)$ for all i from $lfdr'$ and the adjustment term using the training data.
(2) Choose a α_{cutoff} (see Section 4.6).

(3) For each $i = 1, \dots, n_{test}$, among all labels without an assignment, choose the node with the highest ranking, i.e. $k_i^* = \arg\max_k \{lfdr^*_{k,i}(D_i) : k \notin D_i\}$

(4) Classify $\hat{Q}_{k^*,i}$ according to α_{cutoff} according Equation 4.3.7.

(5) Enforce the hierarchical constraints implied by the hierarchy.

(6) Update each $lfdr'$ accoring to Equation 4.3.8 and π_k according to Equation 4.3.8.

(7) Repeat (3) through (6) until all labels for the test set have been assigned.

For brevity we will denote $lfdr^* = \{lfdr^*_{k,i}(D_i)\}$ for $k = 1, \dots, p$ and $i = 1, \dots, n_{test}$.

We gave little guidance to the choice of the cutoff value when a single cutoff is necessary. In this case, the researcher should specify a desirable recall level enable to select a cutoff, $\alpha_{cutoff} \in (0, 1)$ for actual classification. We recommend using the training data to select α_{cutoff} via cross validation.

$$\alpha^{(z)}_{cutoff} = \arg \min_{\alpha \in [0,1]} \left\{ recall_{desired} = \frac{\sum_{k,i \in train^{(z)}} I(\hat{Q}_{k,i}(\alpha) = 1 \ and \ Q_{k,i} = 1)}{\sum_{k,i} I(Q_{k,i} = 1)} \right\}$$

Each cross validation will provide one $\alpha^{(z)}_{cutoff}$ based on the different training samples $train^{(z)}$. A simple average of these cutoffs should produce a sensible final cutoff value, α_{cutoff}. This cross validation method can be used to choose α_{cutoff} based on other criteria such as the F-measure (*Musicant et al.*, 2003) as well. The choice of the criteria should be based on the goal of the classification.

4.3.3. Estimation of $lfdr'$. *Jiang et al.*, 2013 has shown that $lfdr'$ is theoretically equivalent to their LPR in the MC setting. Intuitively, LPR measures the trade-off between precision and recall for different cutoffs applied to the classifier scores. Estimating this trade-off is much more robust than estimating densities under the local false discovery rate definition by *Efron and Tibshirani* (2002).

Here we first define some notation for their derivation. Define $u_k(\lambda_k)$ to be the probability that a random person will have a classifier score less than or equal to λ_k for class k, i.e. $u_k(\lambda_k) = P(S_{k,\cdot} \leq \lambda_k)$. Intuitively, $u_k(\lambda_k)$ places the different classifier scores on the same scale between $[0, 1]$. Then define the precision function for class k as $G_k(\lambda_k) = P(Q_{k,\cdot} = 1 | S_{k,\cdot} > \lambda_k)$ when we classify a positive status for scores above λ_k. Based on the one to one relationship between λ_k and u_k, however, we can denote the precision function as a function of the u_k as $G_k(u_k)$. This makes the precision functions comparable over the classifiers. Finally, LPR is defined as

$$lpr_k(u_k) \quad = \quad -\frac{d}{du_k}\left[(1 - u_k)G_k(u_k)\right] \tag{4.3.9}$$

Equation 4.3.9 is derived from maximizing the pooled precision rate $\frac{\sum_k (1-u_k)G_k(u_k)}{\sum_k (1-u_k)}$ after fixing $\sum_k (1 - u_k)$ (the negative sign is to simply make high LPR values correspond to higher rankings). Intuitively, when the classes are independent, LPR measures the gain in probability of a true positive calls for each unit decrease in u_k for $k = 1, \dots, p$. When the trade-offs of these gains are equal over each class, then there can be no more gains in the pooled precision rate with a fixed expected number of positive assignments, i.e. $\sum_k (1 - u_k)$.

For estimation, *Jiang et al.*, 2013 recommended re-expressing Equation 4.3.9 as

$$lpr_{k,i} = G_k(u_{k,i}) - (1 - u_{k,i}) \frac{d}{du_k} G_k(u_{k,i}) \tag{4.3.10}$$

then obtain lpr by estimating $G_k(\cdot)$ through kernel smoothing methods and obtaining the implied derivative $\frac{d}{du_k} G_k(\cdot)$. Here we propose an estimation for $G_k(\cdot)$ via splines (*Hastie and Tibshirani*, 1990) comebined with bagging to estimate LPR. Splines can fit complex curves quickly and have many existing algorithms with sensible default values. Moreover, the derivative value is easily obtainable through most spline fitting algorithms. Next, the empirical estimation of $G_k(\cdot)$ is heteroscedastic which can be factored into the spline fitting and not by kernel methods with a constant bandwidth.

Our contribution to the estimation is to introduce bagging and factoring the heteroscedasticity of the empirical $G_k(\cdot)$. Bagging makes the estimation more robust and accurate where adjusting for heteroscedasticity provides better estimates.

To implement bagging eventually, we first resample our data to create B batches of samples each with size n. We then follow *Jiang et al.*, 2013 and estimate u_k and $G_k(\cdot)$ empirically

$$\begin{aligned} \hat{u}_{k,i} &= \frac{1}{n} \sum_{j=1}^{n} I(S_{k,j} \leq S_{k,i}) \\ \hat{G}_k(\hat{u}_{k,i}) &= \sum_{j} I(Q_{k,j} = 1 \& S_{k,j} > S_{k,i}) / \sum_{j=1}^{n} I(S_{k,j} > S_{k,i}) \end{aligned} \tag{4.3.11}$$

Under Equation 4.3.11, large $S_{k,i}$ values have few samples to estimate $G_k(u_{k,i})$ and thus should exhibit larger variance. Figure 4.3.1 demonstrates that as $\hat{u}_{k,i}$ increases, the variability increases in $\hat{G}_k(\hat{u}_{k,i})$. This heterscedastic behavior is amplified when the positive cases is low since $\hat{G}_k(\hat{u}_{k,i})$ will vary greatly with the loss of one positive case when $\hat{u}_{k,i}$ increases.

To accomodate this heteroscedastic bahvior, we put weights to be inversely proportional to the sample size used to compute $\hat{u}_{k,i}$. To choose the smoothing parameter for the splines, we implement a 5 fold cross-validation where each turn we use $\frac{1}{5}$ of the data as training to predict the remaining $\frac{4}{5}$ of the data. This yields larger smoothing parameters which works better in our simulations. The reason to prefer larger smoothing parameters is because $\hat{G}_k(\hat{u}_{k,i})$ can decrease drastically due to the loss of a single positive case so more smoothing is often beneficial. Another reason is because the deviations of $\hat{G}_k(\hat{u}_{k,i})$ from $G_k(u_{k,i})$ is highly correlated. This smoothness is due to the sorted nature of the $S_{k,\cdot}$ and leads to underestimating the smoothing parameter. By using a small subsample of the training set, we artificially create less correlated data which leads to higher smoothing parameters.

The bagging provides the most improvement in the new estimation of LPR. Recall that the estimation for $\hat{G}_k$ and $\hat{u_k}$ came from the baggin samples so we have B estimates for $lpr_{k,i}$. Bagging simply averages over the B batches to create a robust estimation for $lpr_{k,i}$.

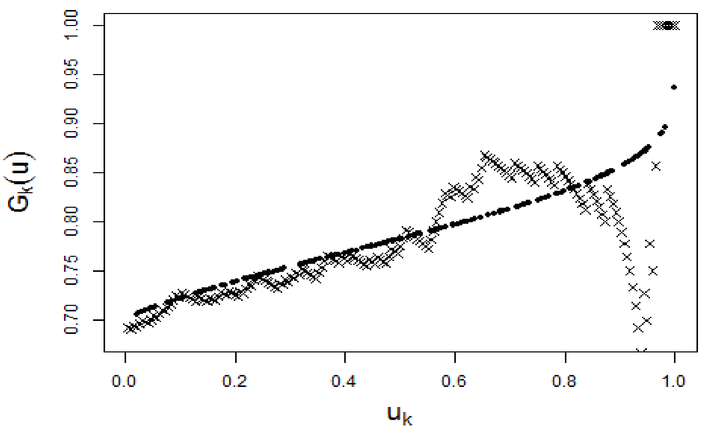

FIGURE 4.3.1. Empirical Estimation of $G_k(u_k)$ and u_k vs. the truth with 70% positive cases. × represent the values from $\hat{G}_k(\hat{u}_k)$ and $\hat{u}_k$ where • represent values from the true $G_k(u_k)$ and u_k knowing the underlying distribution. With fewer positive cases, the fluctuation is even worst with the loss of a single positive case.

$$\begin{aligned} \tilde{lpr}_{k,i,b} &= \hat{G}_k(\hat{u}_{k,i}) - (1 - \hat{u}_{k,i})\hat{G}'_k(\hat{u}_{k,i}) \\ \hat{lpr}_{k,i} &= \frac{1}{B}\sum_b \tilde{lpr}_{k,i,b} \end{aligned}$$

In our implementation, we use all n cases for the first bagging sample (i.e. no resampling) for determining the smoothing parameter. We then recycle the smoothing parameter from this first sample for all remaining $B - 1$ estimates of $\hat{G}_k(\cdot)$. This avoids unnecessary cross validation and greatly speeds up the estimation.

4.4. Simulation Study: Hierarchical Gaussian Mixture

We demonstrate our algorithm on a simulation study. Our code for this section is available at `http://www.stat.berkeley.edu/~lwtai/Waynes_Stat_Website/Tech_Reports.html`.

We first generate the true class statuses as specified in the methodology section with $p_k = 0.7$ for all k. The DAG structure that identifies the parent and child relationships is as illustrated in Figure 4.1.1. To match our disease diagnosis example below, the training set will have 196 samples only and the testing set will have 20 samples.

After generating the class statuses, we then produce classifier scores for each class. We assume that

$$\begin{aligned} S_{k,i}|Q_{k,i}=0 &\sim Gaussian(0,1) \\ S_{k,i}|Q_{k,i}=1 &\sim Gaussian(m_k,1) \end{aligned}$$

for $k = 1, \ldots p$ and $i = 1, \ldots, n$. Here, m_k determines the quality of the data or local classifier. If class k has poor data quality or a weak local classifier, then m_k is set to be 0.5. This makes the scores between the positive and negative cases more similar and harder to differentiate. Otherwise, a well-trained classifier will have $m_k = 1.5$.

To make the simulation more meaningful, we compare it to the global classifier based on predictive cluster trees. Specifically, predictive cluster trees with bagging proposed in *Dimitrovski et al.* (2011). The original algorithm builds a decision tree that maximizes the label similarities between the training samples (**?**). At each branch of the tree, the training samples are separated by the feature that creates groups that most improves some cluster-similarity measure on the labels. The tree stops growing when number of training samples in the branch is too small or when the label similarities increase less than a certain threshold after the split. The leaf nodes are formed when any of these stopping criteria are met then the proportion of positive labels for each class is computed. These proportions then become the classifier scores for all test examples that land in that leaf node after being passed down the decision tree. A cutoff is then chosen to provide an assignment for all labels. Notice that since all the labels in the training data respect the hierarchical constraints, the label assignments will also be consistent.

This algorithm was extended to DAGs by *Vens et al.* (2008) where *Dimitrovski et al.* (2011) added the bagging component to remedy errors early in the decision tree. Here we treat the local classifier outputs as features so all algorithms are provided with the same information. Intuitively, the predictive clustering tree should be able to recover the a decent solution with its exhaustive searches. For decent classification, the algorithm needs to identify the split at $\frac{m_k}{2}$ for each node k and start the branching with the node with the strongest signal, i.e. largest π_k and m_k. Even so, we will show that our algorithm is shows better performance. We will refer this algorihm as the ClusHMC algorithm from here on for convenience.

We can also treat the problem as a MC problem as in *Jiang et al.*, 2013 and apply a uniform cutoff to the $lfdr'$ values. This completely ignores the hierarchy but will serve as a good comparison to know the benefits from introducing the hierarchy. Naturally, we implement the sequential classificaiton method based on $lfdr^*$ in Section 4.3 using the estimated $lfdr^{'}$. To understand the effects of estimation $lfdr'$, we also implement the sequential classificaiton method knowing the true Gaussian densities and parameters.

Finally, to make the classifiers comparable, we restrict the number of bagging for ClsHMC and the $lfdr'$ estimation to be the same. Overall, our method can perform well with limited bagging where the tree based methods need significantly more bagging to produce consistent results.

To compare the performance, we calculate the precision and recall curve defined in Equation 4.4.1.

$$\begin{aligned} Precision_k &= \frac{\sum_i I(Q_{k,i} = 1 \& \hat{Q}_{k,i} = 1)}{\sum_i I(\hat{Q}_{k,i} = 1)} \\ Recall_k &= \frac{\sum_i I(Q_{k,i} = 1 \& \hat{Q}_{k,i} = 1)}{\sum_i I(Q_{k,i} = 1)} \end{aligned} \tag{4.4.1}$$

To represent the average performance, we repeat this simulation over 20 independent simulations and show the average precision and average recall curve in Figure 4.4.1.

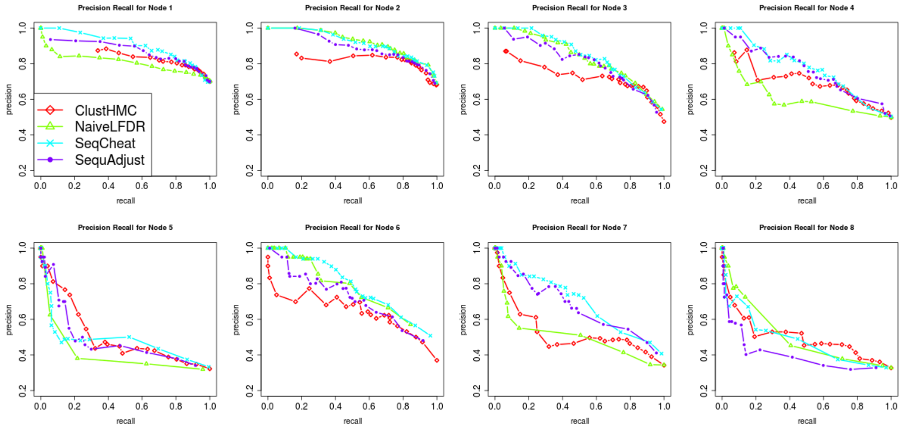

FIGURE 4.4.1. Precision and recall curve for each node with Gaussian distribution scores. In this example, $m_2 = m_3 = m_6 = 1.5$ and the rest all have $m_k = 0.5$. The curves are averages over 20 repetitions where each repetition has 196 training points and 20 testing points. ClusHMC is the predictive cluster trees method with bagging, NaiveLFDR is the MC method of using a uniform cutoff for all true $lfdr^{'}$ values (not estimated) ignoring the hierarchical structure, Sequential is our sequential method while we estimate $lfdr'$, SeqCheat is our sequential method knowing the mixture distribution of the scores.

Most algorithms perform very well when the signal is strong, i.e. the distribution between the negative and positive cases are very different. The comparison between the uniform cutoffs for $lfdr'$ and the sequential method with $lfdr^*$ shows that neighboring nodes can help drastically. For example, node 4 has a weak signals but most algorithms perform very well due to the signal from node 6. The performance for node 7 shows that strong signals from the parent node can also be beneficial. Node 5 and 8 on the other hand are difficult cases for most algorithms. This is expected since their neighboring nodes have weak signals, their own signal is weak, and there are limited positive cases in the training set in these leaf nodes. One surprise is that although node 5 has one parent node with a strong signal, its performance is still poor. This is likely due to the effects from node 1 that makes the estimation for $\frac{P(Q_k=0|S_{-k})}{P(Q_k=1|S_{-k})}$ difficult. We rule out the possibility of imbalanced dataset issues since node 7 is low in the hierarchy and also depends on two ancestors nodes. This raises questions about neighbor selection which we will not discuss in this chapter.

Overall, our sequential method achieves higher precision rates than ClusHMC for most recall values but the performance is comparable at high levels of recall. Lastly, the difference between the sequential classification based on the true $lfdr^*$ and estimated $lfdr^*$ shows that knowing the distribution can further improve the performance of our method.

We believe the difference in performance between ClusHMC and our sequential method is because the global classifier builds decisions for all outputs from the same local classifier at once where our algorithm works with the individual outputs separately. The global algorithm avoids poor outputs from poor classifiers but can suffer from uninformative outputs from quality classifiers. With poor data, quality classifiers can also produce uninformative outputs that do not help determine the label status. In this case, outputs from lesser quality classifiers might be more informative that can produce more accurate results. Our ranking captures this concept by ranking the individual label assignments instead of creating a single order for all outputs.

When the number of nodes increase, individually evaluating the precision and recall is no longer feasible. To accomodate this, *Vens et al.* (2008); *Silla Jr and Freitas* (2011); *Jiang et al.* (2013) all used a measure that considers all the nodes jointly as shown in Equation 4.4.2. Under this measure, the results for the same simulation is shown in Figure 4.4.2.

$$\begin{aligned} Precision^* &= \frac{\sum_i \left|Q_{\cdot,i} = 1 \& \hat{Q}_{\cdot,i} = 1\right|}{\sum_i \left|\hat{Q}_{\cdot,i} = 1\right|} = \frac{\sum_k \sum_i I(Q_{k,i} = 1 \& \hat{Q}_{k,i} = 1)}{\sum_k \sum_i I(\hat{Q}_{k,i} = 1)} \\ Recall^* &= \frac{\sum_i \left|Q_{\cdot,i} = 1 \& \hat{Q}_{\cdot,i} = 1\right|}{\sum_i \left|Q_{\cdot,i} = 1\right|} = \frac{\sum_k \sum_i I(Q_{k,i} = 1 \& \hat{Q}_{k,i} = 1)}{\sum_k \sum_i I(Q_{k,i} = 1)} \end{aligned} \tag{4.4.2}$$

When considering the classes jointly, the comparison is consistent with our observations before. Our sequential method still achieves higher precision values than the global classifier at most recall value. One interesting fact is that the precision performance for the uniform cutoff for $lfdr'$ is comparable at low recall regions with the sequential method. Again, the label assignments from this will not be consistent with the hierarchy so is not desirable for our HMC. The comparable results is likely a result of the similar rankings produced by $lfdr'$ and $lfdr^*$ when $lfdr'$ is high.

To demonstrate the flexibility of our algorithm, we can repeat the same simulation by replacing the Gaussian distributions with Beta distributions but maintaining the same DAG structure. Specifically, we create classifier scores as

$$\begin{aligned} S_{k,i}|Q_{k,i} = 0 &\sim Beta(1, 1.3) \\ S_{k,i}|Q_{k,i} = 1 &\sim Beta(1, m_k) \end{aligned}$$

This will produce classifier scores all between $[0, 1]$ which can be difficult for classifiers. The smaller values of m_k will put more mass on the higher end of the unit

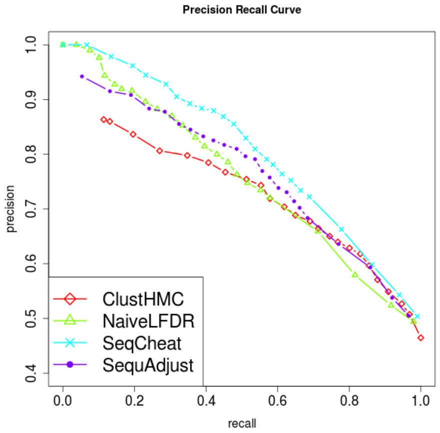

FIGURE 4.4.2. Joint Evaluation over all Nodes of Simulation using Recall and Precision with Gaussian distribution scores. Our algorithm outperforms ClusHMC at lower recall values but is comparable at high recall values.

interval and differentiate the positive and negative score distributions better. We perform the same evaluations and show the results in Figure 4.4.3 and Figure 4.4.4.

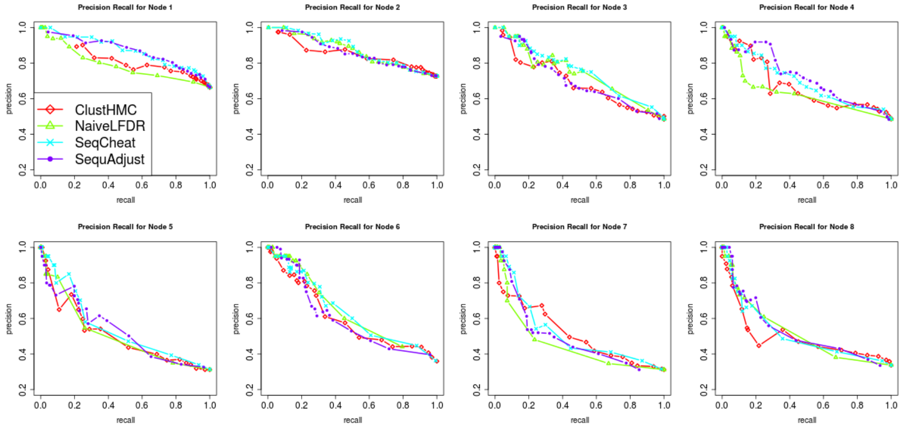

FIGURE 4.4.3. Precision and recall curve for each node with Beta distribution scores. In this example, $m_2 = m_3 = m_6 = 0.5$ and the rest all have $m_k = 0.9$. The curves are averages over 20 repetitions where each repetition has 196 training points and 20 testing points. ClusHMC is the predictive cluster trees method with bagging, NaiveLFDR is the MC method of using a uniform cutoff for all true $lfdr^{'}$ values (not estimated) ignoring the hierarchical structure, Sequential is our sequential method while we estimate $lfdr'$, SeqCheat is our sequential method knowing the mixture distribution of the scores.

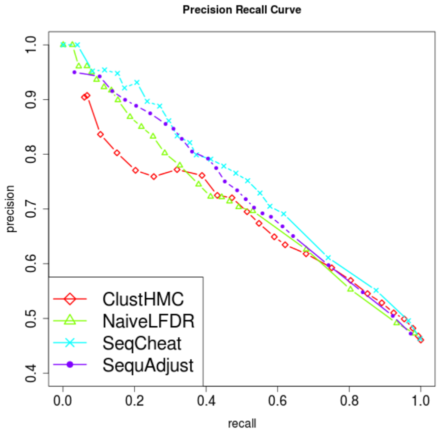

FIGURE 4.4.4. Joint Evaluation over all Nodes using Recall and Precision with Beta distribution scores.

With all the classifier scores in the unit interval, the performance is overall worse than the Gaussian score example. One difference from the Gaussian case is that node 7 did not show the same benefit from the parent node. However, the overall comparison is roughly the same which shows that our sequential method is quite adaptive to different distributions of data.

4.5. GEO Disease Database

We apply our algorithm to the National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO). The goal is to predict the disease statuses in each data sets based on the genetic information of its subjects. The annotation is positive for a dataset if at least one of the subjects has the particular disease. The initial classifier results are from *Huang et al.* (2010) along with the annotations derived from the documentation on the GEO database. The derived annotations are noisy and do not strictly follow the hierarchical structure specified in the UMLS. However, according to the authors, positive annotations should be accurate but certain negative annotations might be false due to poor documentation and possible text mining errors. This yields 196 datasets over 110 diseases, each with a particular classifier score.

To evaluate the methods, we performed a leave-one-out method over all 196 datasets where 195 were used for training and the remaining one data set was used as a test set. The final precision and recall curve is then based on all 196*110 label assignments. The results over the 110 diseases are summarized into one precision recall curve as calculated in Equation 4.4.2.

We implement similar comparisons as those mentioned in the simulation study. $lfdr*$ values are again estimated by calculating the adjustment term and estimating $lfdr'$ as specified in Section 4.3.3. But here we compare these to the $lfdr'$ values estimated in *Jiang et al.* (2013) using the same uniform cutoff. We also implement ClusHMC on this dataset. Each method is provided with the classifier scores from *Huang et al.* (2010) which improved the results from the local classifiers drastically. The results are shown in Figure 4.5.1.

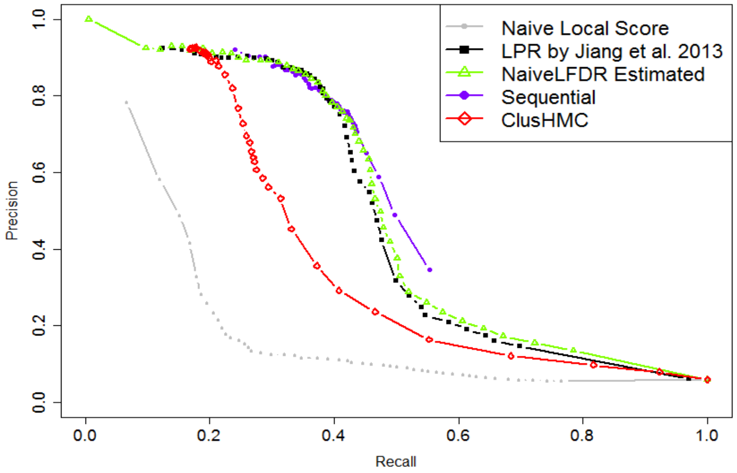

FIGURE 4.5.1. GEO Disease Classification Comparisons. Naive Local Score is the result from using the local classifier outputs with a uniform cutoff. LPR by *Jiang et al.* (2013) is the results from Jiang et al. 2013 for comparison. The rest is similar to the simulation notation. Our method again outperforms all other methods at most recall values. The improvement from LPR is mostly at mid-recall regions. The improvement is not huge but our label assignments are consistent with the UMLS hierarchy.

Our $lfdr'$ estimation method slightly improved the results from Jiang et al. 2013 for recall values beyond 0.4. Our sequential method further improved those results by producing higher precision values for most recall regions. The only exception is around 0.3 and 0.4 recall rate. We believe this is a consequence from the noisy annotations because the sequential method should perform at least as well as the naive $lfdr^*$ method as long as the $lfdr^*$ values were updated sensibly.

The global classifier surprisingly did not have comparable performance to our sequential classifier at any recall value. In addition to the reasons mentioned in the simulation, possible reasons for the poor performance for ClusHMC are due to the high dimension of the data set, the noisy annotation, and the limited sample and bagging size. Since ClusHMC does not explicitly use the hierarchy in their clustering, noisy annotations in the training set can lead to poor classifications.

4.6. Discussion

Overall, we introduce a Bayesian framework for ranking individual label assignments in the HMC setting that extends from the optimal ranking in the MC setting. We also provided a sequential classification algorithm that highlights the role of neighbors in HMC in our problem setting. The algorithm is greedy and does not consider the joint assignment of all labels but its sequential nature accomodates for these flaws. We also introduce new estimation methods for $lfdr'$ that is more robust than the existing method. Through simulation and real data, we have shown that our classification method requires little computational effort and provides robust results when compared against popular existing global classifiers.

The biggest limitation of our method is that it relies on classifier scores from existing local classifiers. While facing new classes, a single global classifier might be more simplistic and faster than building individual local classifiers and aggregating the results (*Vens et al.*, 2008). On the other hand, our framework allows researchers to construct customized local classifiers without needing to worry how to extend their method to other classes. Our framework also allows recycling the efforts from well documented work which can be beneficial.

A second minor limitation is that different cutoffs require re-running the algorithm since the method is sequential. MC methods can typically trivially apply different cutoffs to obtain different classification results where our classification each depends on the previous assignment. Fortunately the $lfdr^*$ values do not need to be re-estimated so the computational effort from this issue is quite small.

Many improvements can be implemented for this method. The first is a better estimation for $\frac{P(Q_k=0|S_{-k})}{P(Q_k=1|S_{-k})}$ for the adjustment term. The logistic regression limited to the family nodes is fast and convenient but the simulation study raised the question of neighbor selection. Updating this term efficiently is also challenging which we ignored in this chapter.

It is possible that the classifier scores do not follow a simple mixture distribution. More importantly, the conditional independence assumption of the classifier output given the label status is quite strong and provided many simplifications. ClusHMC on the other hand performs an exhaustive search over all the feature at each split to accomodate this possible dependency. Moreover, extending our method to the case where label assignments necessarily reach a leaf-node is not trivial. Ad-hoc constraints in the algorithm might produce sensible answers but we do not explore these possibilities here.

Lastly, if a disease could have multiple statuses instead of the binary status we assumed here, e.g. {negative, curable, hopeless}, it is not clear how our algorithm and estimation would be affected in this case. One possibility is to treat the multiple statuses as differernt stages of a disease, e.g. a new disease down the DAG. This, however, could greatly increase the dimensionality of the DAG and break the conditional independence assumption between the status and classifier scores. We also leave this for future extensions on this topic.

CHAPTER 5

Concluding Remarks

We tackled three different applied statistical problems from the Bayesian perspective: inference for the global extreme with limited data, a spatial model for global surface winds, and a sequential method for hierarchical multilabel classification on DAGs. Each had a complicated dependence structure where the data was often high dimensional. For the first two topics, the Bayesian interpretation allowed us to formally quantify our uncertainty given different prior beliefs and observations. This allows researchers to introduce their prior beliefs and knowledge to resolve issues with limited independent observations that are high dimensional. The multidimensional dependence was mostly handled by GPs. For the last topic, the Bayesian interpretation was used as a motivation and framework to design a consistent classifier on complicated DAGs. Overall, the Bayesian perspective on statistical problems is cohesive, intuitive, and flexible.

The biggest challenge we encountered in this dissertation is in the computational demands from the MCMC methods and the specification of priors. We specified many methods to decrease the computational burden. However, we did not explore other inference techniques such as variational Bayes which approximates the posterior distribution with great computational performances (*Ormerod and Wand*, 2010). Moreover, we have tested our sensitivity to many different prior specifications but this process was extremely taxing in terms of time and resources. Efficient guidance or theory on when the prior might influence the results would helpful. We leave these challenges for future explorations.

Bibliography

Aldering, G., G. Adam, P. Antilogus, P. Astier, R. Bacon, S. Bongard, C. Bonnaud, Y. Copin, D. Hardin, F. Henault, et al., Overview of the nearby supernova factory, in *Astronomical Telescopes and Instrumentation*, pp. 61–72, International Society for Optics and Photonics, 2002.

Assunçao, R., Space varying coefficient models for small area data, *Environmetrics*, *473*(April 2002), 453–473, doi:10.1002/env.599, 2003.

Barutcuoglu, Z., R. E. Schapire, and O. G. Troyanskaya, Hierarchical multi-label prediction of gene function, *Bioinformatics*, *22*(7), 830–836, 2006.

Benjamini, Y., and Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 289–300, 1995.

Berger, J. O., V. De Oliveira, and B. Sansó, Objective Bayesian analysis of spatially correlated data, *Journal of the American Statistical Association*, (December 2012), 37–41, 2001.

Bickel, P. J., A distribution free version of the smirnov two sample test in the p-variate case, *The Annals of Mathematical Statistics*, *40*(1), 1–23, 1969.

Bickel, P. J., and E. Levina, Regularized estimation of large covariance matrices, *The Annals of Statistics*, pp. 199–227, 2008.

Board, J. E., and H. Modali, Dry matter accumulation predictors for optimal yield in soybean, *Crop science*, *45*(5), 1790–1799, 2005.

Bodenreider, O., The unified medical language system (umls): integrating biomedical terminology, *Nucleic acids research*, *32*(suppl 1), D267–D270, 2004.

Booker, A. J., J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset, A rigorous framework for optimization of expensive functions by surrogates, *Structural and Multidisciplinary Optimization*, *17*(1), 1–13, 1999.

Box, G., and K. Wilson, On the experimental attainment of optimum conditions, *Journal of the Royal Statistical Society. Series B (. . .*, *13*(1), 1–45, 1951.

Brooks, S., A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, Chapman and Hall/CRC, 2011.

Chiang, J. C., and S. E. Zebiak, Surface wind over tropical oceans: Diagnosis of the momentum balance, and modeling the linear friction coefficient, *Journal of Climate*, *13*(10), 1733–1747, 2000.

Cornford, D., Surface wind fields (on Earth), *Tech. rep.*, Neural Computing Research Group, 1997.

Cornford, D., Flexible Gaussian process wind field models, *Tech. rep.*, Neural Computing Research Group, 1998.

Cressie, N., and G. Johannesson, Fixed rank kriging for very large spatial data sets, *Journal of the Royal Statistical Society: Series B*, *70*, 209–226, 2008.

Diggle, P. J., J. a. Tawn, and R. a. Moyeed, Model-based geostatistics, *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *47*(3), 299–350, doi:10.1111/1467-9876.00113, 2002.

Dimitrovski, I., D. Kocev, S. Loskovska, and S. Džeroski, Hierarchical annotation of medical images, *Pattern Recognition*, *44*(10), 2436–2449, 2011.

Ecker, M. D., and J. F. Heltshe, Geostatistical estimates of scallop abundance, *Case studies in biometry*, pp. 107–124, 1994.

Efron, B., *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, vol. 1, Cambridge University Press, 2010.

Efron, B., and R. Tibshirani, Empirical bayes methods and false discovery rates for microarrays, *Genetic epidemiology*, *23*(1), 70–86, 2002.

Facer, M. R., and H.-G. Müller, Nonparametric estimation of the location of a maximum in a response surface, *Journal of Multivariate Analysis*, *87*(1), 191–217, doi:10.1016/S0047-259X(03)00030-7, 2003.

Forrester, A. I. J., A. J. Keane, and N. W. Bressloff, Design and Analysis of "Noisy" Computer Experiments, *American Institute of Aeronautics and Astronautics Journal*, *44*(10), 2331–2339, doi:10.2514/1.20068, 2006.

Fuentes, M., A high frequency kriging approach for non-stationary environmental processes, *Environmetrics*, *483*(April 2000), 469–483, doi:10.1002/env.473, 2001.

Gelfand, A., P. J. Diggle, P. Guttorp, and M. Fuentes, *Handbook of spatial statistics*, CRC Press, 2010.

Gelfand, A. E., H.-J. Kim, C. F. Sirmans, and S. Banerjee, Spatial modeling with spatially varying coefficient processes, *Journal of the American Statistical Association*, *98*(462), doi:10.1198/016214503000170, 2003.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*, Chapman & Hall/CRC, 2004.

Genton, M., and A. Hering, Blowing in the wind, *Significance*, *4*(1), 11–14, 2007.

Hall, B. P., Permutation tests for equality of distributions in high-dimensional settings, *Biometrika*, pp. 359–374, 2002.

Haslett, J., and A. Raftery, Space-time modelling with long-memory dependence: Assessing Ireland's wind power resource, *Applied Statistics*, *38*(1), 1–50, 1989.

Hastie, T., and R. Tibshirani, *Generalized additive models*, vol. 43, Chapman & Hall/CRC, 1990.

Hering, A., and M. Genton, Powering up with space-time wind forecasting, *Journal of the American Statistical Association*, (2006), doi:10.1198/jasa.2009.ap08117, 2010.

Higdon, D., A process-convolution approach to modelling temperatures in the North Atlantic Ocean, *Environmental and Ecological Statistics*, *190*, 1998.

Hsiao, E., A. Conley, D. Howell, M. Sullivan, C. Pritchet, R. Carlberg, P. Nugent, and M. Phillips, K-corrections and spectral templates of type ia supernovae, *The Astrophysical Journal*, *663*(2), 1187, 2008.

Huang, H., C.-C. Liu, and X. J. Zhou, Bayesian approach to transforming public gene expression repositories into disease diagnosis databases, *Proceedings of the National Academy of Sciences*, *107*(15), 6823–6828, 2010.

Jiang, C.-R., C.-C. Liu, X. Zhou, and H. Huang, Optimal ranking in multilabel classification using local precision rate, *Accepted Under Revision*, 2013.

Jones, D. R., M. Schonlau, and W. J. Welch, Efficient global optimization of expensive black-box functions, *Journal of Global optimization*, *13*(4), 455–492, 1998.

Kaufman, C., M. Schervish, and D. Nychka, Covariance tapering for likelihood-based estimation in large spatial data sets, *Journal of the American Statistical Association*, *103*(484), 1545–1555, 2008.

Kaufman, C. G., and S. R. Sain, Bayesian functional {ANOVA} modeling using gaussian process prior distributions, *Bayesian Analysis*, *5*(1), 123–149, 2010.

Kaufman, C. G., D. Bingham, S. Habib, K. Heitmann, and J. a. Frieman, Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology, *The Annals of Applied Statistics*, *5*(4), 2470–2492, doi:10.1214/11-AOAS489, 2011.

Kerr, R., Models win big in forecasting el niño, *Science*, *280*(5363), 522–523, 1998.

Klöppel, S., C. M. Stonnington, C. Chu, B. Draganski, R. I. Scahill, J. D. Rohrer, N. C. Fox, C. R. Jack, J. Ashburner, and R. S. Frackowiak, Automatic classification of mr scans in alzheimer's disease, *Brain*, *131*(3), 681–689, 2008.

Koller, D., and M. Sahami, Hierarchically classifying documents using very few words, 1997.

Lindgren, F., H. v. Rue, and J. Lindström, An explicit link between Gaussian fields and Gaussian Markov random fields: The SPDE approach, *Journal of the Royal Statistical Society: Series B*, *73*, 423–498, 2011.

Milliff, R., W. Large, J. Morzel, G. Danabasoglu, and T. Chin, Ocean general circulation model sensitivity to forcing from scatterometer winds, *Journal of geophysical research*, *104*(C5), 11,337–11, 1999.

Milliff, R. F., A. Bonazzi, C. K. Wikle, N. Pinardi, and L. M. Berliner, Ocean ensemble forecasting. Part I: Ensemble Mediterranean winds from a Bayesian hierarchical model, *Quarterly Journal of the Royal Meteorological Society*, (November 2009), 858–878, doi:10.1002/qj.767, 2011.

Müller, H., Kernel estimators of zeros and of location and size of extrema of regression functions, *Scandinavian journal of statistics*, *12*(3), 221–232, 1985.

Müller, H., M. Facer, N. Bills, and A. Clifford, Statistical interaction model for exchangeability of food folates in a rat growth bioassay, *The Journal of nutrition*, (June), 2585–2592, 1996.

Müller, H.-G., Nonparametric Peak Estimation, *The Annals of Stastistics*, *17*(3), 1053–1069, 1989.

Musicant, D. R., V. Kumar, and A. Ozgur, Optimizing f-measure with support vector machines, in *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, pp. 356–360, 2003.

Neelin, J., *Climate Change and Climate Modeling*, Climate Change and Climate Modeling, Cambridge University Press, 2010.

Ormerod, J., and M. Wand, Explaining variational approximations, *The American Statistician*, *64*(2), 140–153, 2010.

Paciorek, C., and M. Schervish, Spatial modelling using a new class of nonstationary covariance functions, *Environmetrics*, (February), 483–506, doi:10.1002/env.785, 2006.

Pereira, R., R. Thomas, G. Aldering, P. Antilogus, C. Baltay, S. Benitez-Herrera, S. Bongard, C. Buton, A. Canto, F. Cellier-Holzem, et al., Spectrophotometric time series of sn 2011fe from the nearby supernova factory, *Astronomy and Astrophysics Accepted. arXiv preprint arXiv:1302.1292*, 2013.

Rasmussen, C., and C. Williams, *Gaussian processes for machine learning*, vol. 1, MIT press Cambridge, MA, 2006.

Rasmussen, C. E., C. K. I. Williams, G. Processes, M. I. T. Press, and M. I. Jordan, *Gaussian Processes for Machine Learning*, 2006.

Reich, B. J., and M. Fuentes, A multivariate semiparametric bayesian spatial modeling framework for hurricane surface wind fields, *The Annals of Applied Statistics*, *1*(1), 249–264, 2007.

Reich, B. J., J. Eidsvik, M. Guindani, A. J. Nail, and A. M. Schmidt, A class of covariate-dependent spatiotemporal covariance functions for the analysis of daily ozone concentration, *The Annals of Applied Statistics*, *5*(4), 2425–2447, doi:10.1214/11-AOAS482, 2011.

Richards, J. W., D. L. Starr, N. R. Butler, J. S. Bloom, J. M. Brewer, A. Crellin-Quick, J. Higgins, R. Kennedy, and M. Rischard, On machine-learned classification of variable stars with sparse and noisy time-series data, *The Astrophysical Journal*, *733*(1), 10, 2011.

Roberts, G. O., and J. S. Rosenthal, Optimal scaling for various metropolis-hastings algorithms, *Statistical Science*, *16*(4), 351–367, 2001.

Royle, J., L. Berliner, C. Wikle, and R. Milliff, A hierarchical spatial model for constructing wind fields from scatterometer data in the Labrador Sea, in *Case Studies in Bayesian Statistics IV*, edited by C. Gatsonis, R. Kass, A. Cariquiry, B. Carlin, C. A., A. Gelman, I. Verdinelli, and M. West, pp. 367–381, Springer - Verlag, 1998.

Rue, H., and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, *Monographs on Statistics and Applied Probability*, vol. 104, Chapman & Hall, London, 2005.

Sain, S. R., D. Nychka, and L. Mearns, Functional anova and regional climate experiments: a statistical analysis of dynamic downscaling, *Environmetrics*, *22*(6), 700–711, 2011.

Salameh, T., P. Drobinski, M. Vrac, and P. Naveau, Statistical downscaling of near-surface wind over complex terrain in southern france, *Meteorology and Atmospheric Physics*, *103*(1-4), 253–265, 2009.

Shaby, B., and M. T. Wells, Exploring an Adaptive Metropolis Algorithm, 2010.

Shi, T., and N. Cressie, Data Mining of MISR Aerosol Product using Spatial Statistics, *Computational Intelligence and Data Mining, . . .*, (Cidm), 712–719, 2007.

Silla Jr, C. N., and A. A. Freitas, A survey of hierarchical classification across different application domains, *Data Mining and Knowledge Discovery*, *22*(1-2), 31–72, 2011.

Simpson, T., T. Mauery, J. Korte, and F. Mistree, Comparison of response surface and kriging models for multidisciplinary design optimization, *AIAA paper 98*, pp. 1–11, 1998a.

Simpson, T., T. Mauery, J. Korte, and F. Mistree, Comparison of response surface and kriging models for multidisciplinary design optimization, *AIAA paper 98*, *4758*(7), 1998b.

Solak, E., R. Murray-Smith, W. Leithead, D. Leith, and C. Rasmussen, Derivative observations in gaussian process models of dynamic systems, 2003.

Sotiriou, C., S.-Y. Neo, L. M. McShane, E. L. Korn, P. M. Long, A. Jazaeri, P. Martiat, S. B. Fox, A. L. Harris, and E. T. Liu, Breast cancer classification and prognosis based on gene expression profiles from a population-based study, *Proceedings of the National Academy of Sciences*, *100*(18), 10,393–10,398, 2003.

Stein, M., *Interpolation of spatial data: some theory for kriging*, Springer Verlag, 1999.

Sun, A., and E.-P. Lim, Hierarchical text classification and evaluation, in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pp. 521–528, IEEE, 2001.

Sun, A., E.-P. Lim, W.-K. Ng, and J. Srivastava, Blocking reduction strategies in hierarchical text classification, *Knowledge and Data Engineering, IEEE Transactions on*, *16*(10), 1305–1308, 2004.

Tsoumakas, G., and I. Katakis, Multi-label classification: An overview, *International Journal of Data Warehousing and Mining (IJDWM)*, *3*(3), 1–13, 2007.

Vens, C., J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, Decision trees for hierarchical multi-label classification, *Machine Learning*, *73*(2), 185–214, 2008.

Villemonteix, J., E. Vazquez, and E. Walter, An informational approach to the global optimization of expensive-to-evaluate functions, *Journal of Global Optimization*, *44*(4), 509–534, doi:10.1007/s10898-008-9354-2, 2008.

Wall, M. M., A close look at the spatial structure implied by the CAR and SAR models, *Journal of Statistical Planning and Inference*, *121*, 311–324, doi:10.1016/S0378-3758(03)00111-3, 2004.

Wallace, J. M., J. M. Wallace, and P. V. Hobbs, *Atmospheric science: an introductory survey*, vol. 92, Academic press, 2006.

Wikle, C. K., R. F. Milliff, D. Nychka, and L. M. Berliner, Spatiotemporal hierarchical Bayesian modeling tropical ocean surface winds, *Journal of the American Statistical Association*, *96*(454), 382–397, 2001.

Williams, J. P., E. J. De Geus, and L. Blitz, Determining structure in molecular clouds, *The Astrophysical Journal*, *428*, 693–712, 1994.

Wu, F., J. Zhang, and V. Honavar, Learning classifiers using hierarchically structured class taxonomies, in *Abstraction, Reformulation and Approximation*, pp. 313–320, Springer, 2005.

Yekutieli, D., Hierarchical false discovery rate–controlling methodology, *Journal of the American Statistical Association*, *103*(481), 309–316, 2008.

Zhang, H., Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics, *Journal of the American Statistical Association*, *99*(465), 250–261, doi:10.1198/016214504000000241, 2004.

APPENDIX A

Global Surface Wind Model

A.1. Weakly Informative Priors

To create a weakly informative prior for β in Equation 3.4.6, we derive the upper bound for $|\beta_i|$ for all i. We then apply half of this upper bound to be $s_{u,i}$ and $s_{v,i}$ for all i. Based on the Rayleigh Equation approximations presented by *Milliff et al.* (2011), we know the upper bound will be determined $\beta_{\cdot,1}$

$$\frac{f_{lat}}{\rho(f_{lat}^2+\gamma^2)}$$

where the Coriolis parameter $f_{lat} = 2*7.2921*10^{-5}\sin(Latitude)$ which equals to 0 at the equator. Fortunately γ ensures the quantity is finite. The physical meaning of γ is the inverse of the damping time scale for winds in the boundary layer. The damping time is, in the absence of forces other than friction, the seconds taken for wind to reduce by a factor of $\frac{1}{e}$. To obtain a reasonable γ, we reverse engineer the Rayleigh friction value implied in the prior specifications by *Milliff et al.* (2011). In their work, they specified

$$\begin{aligned} 8690 &= \frac{f_{lat}}{\rho(f_{lat}^2+\gamma^2)} \\ 4380 &= \frac{\gamma}{\rho(f_{lat}^2+\gamma^2)} \end{aligned}$$

The implied γ value is then $\frac{4380}{8690}f_{lat}$. Since their region of interest is $Latitude \in [34, 46]$ over the ocean, the smallest possible γ is then $4.11 * 10^{-5}$.

Lastly, we find a lower bound for ρ value to be 0.6 using our data and extreme climate records. The bound for ρ was derived as a function $\frac{P}{Temp*R}$. The smallest P in our data was 92103.82, the highest temperature on earth from web searches was 330.95K, and R, the ideal gas constant for air was set to match water vapor at 461.5 $Jkg^{-1}K^{-1}$. Using the latitude values in our data, this yields an upper bound of 20236.1 so we set $s_{u,i} = s_{v,i} = 10118.05$ for $i = 0, 1, 2$.

On the other hand, the mean function for $\beta_{u,1}$ and $\beta_{v,1}$ are calculated similarly except ρ is replaced with 1.25 according to the average sea level air density by *Wallace et al.* (2006). $\beta_{\cdot,0}$ and $\beta_{\cdot,2}$ are assumed to have mean 0.

Now we discuss the construction for the weakly informative priors for κ. This is the inverse of the range parameter for the usual Matern covariance function. Recall that we scaled the data to fit on a unit sphere. The largest great circle distance between any two locations is then π. The smallest distance between any two distinct locations is calculated based on our data. This limits sensible $\log(\kappa)$ values to be in the interval $[-1.14473, 6.525477]$. For Equation 3.4.7, we then set $a_{u,\epsilon} = a_{v,\epsilon}$ to be the midpoint of this interval and $b_{u,\epsilon}^2 = b_{v,\epsilon}^2 = 42.58186$ which is the square of the upper bound. We use similar logic to create weakly informative priors for $c_{u,i}$ and $c_{v,i}$ in Equation 3.4.7 by using these upper bounds for $\log \kappa_{u,i}$ and $s^2_{u,i}$ to create $b^2_{u,i}$ and $b^2_{v,i}$ for $i = 0, 1, 2$. $a_{u,i}$ and $a_{v,i}$ on the other hand are set to 0.

Lastly, the prior for Equation 3.4.5 is found using upper bounds on the wind velocity data. In other words, b_σ was fixed to be the largest difference between wind velocities in the data.

κ_σ on the other hand is difficult to fix. To obtain a robust quantity, we instead fix $c_\sigma = \log(b^2_\sigma \kappa^{2\nu_\sigma}_\sigma)$ at a sensible value. To fix c_σ, we first place a prior $N(0, b_{c'})$ where $b_{c'}$ is set similarly as $b_{u,i}$. We then sample this in the same adaptive Metropolis Hasting step with κ_ϵ as specified in Section 3.5 during the short chain of our Gibbs sampler. We then calculate the implied κ_σ using c_σ at the final iteration value in the sampler when some form of convergence is reached. κ_σ is then fixed for the remaining inferences.

A.2. Starting Values for Surface Wind Model Gibb Sampler

To obtain reasonable starting values for β, we first ignore spatial dependence and regress U and V locally (using only the data from the same location) against the pressure gradients treating time as replicates. This naive fitting has produced reasonable coefficients for mid-latitude and high latitude regions but not for equatorial regions. To smooth out the high variance in the equatorial regions, we performed a naive smoothing by averaging the regression coefficients within 7 degrees latitude and 15 degrees longitude. We smooth more over longitude since the coefficients is expected to change more over latitude than longitude. Then we use these smoothed coefficients from the naive fit to obtain MLE for c. The estimated β values also provide initial $\hat{\epsilon}$ values. With these, we estimate σ_ϵ by calculating the standard deviation of $\hat{\epsilon}$ for each location. We then again derive the MLE for the necessary parameters for the residual field and variance field.

A.3. Sampling for σ_ϵ in Surface Wind Model

κ_σ is fixed using the methods mentioned in Section A.1 by running a short chain of the Gibb Sampler. While obtaining κ_σ, we also obtain samples for σ_ϵ. We then use these samples for σ_ϵ to estimate the proposal precision matrix for σ_ϵ. We fix this proposal precision matrix for the long chain but allow the proposal variance parameter to adapt as in the usual adaptive Metropolis Hasting algorithm. To estimate the proposal precision based on the limited samples, we use the banded sample precision matrices constructed in *Bickel and Levina* (2008). This method requires the users to specify the relevant neighbors to each location. These neighbors can be inferred using the GMRF representation of the precision matrix from *Lindgren et al.* (2011) which provide very sparse precision matrices.

A.4. Posterior Means for β_u and β_v for Summer Winds

Here we show the posterior means for β_u and β_v for Summer winds. These look qualitatively very similar to the coefficients from Winter. Overall, the difference between the spatially varying variances is more obvious between the seasons than the coefficient fields.

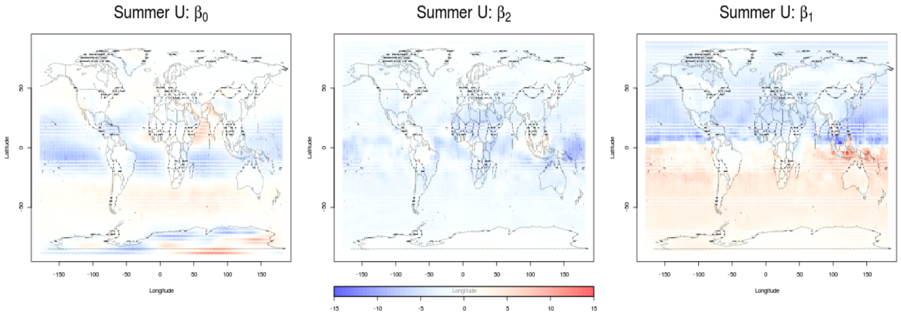

FIGURE A.4.1. Posterior means of β_U terms for Summer. $\beta_{u,1}$ has a larger magnitude as we expected. Moreover, the sign change of the coefficients and decreasing magnitude away from the equator is consistent with Equation 3.4.2. The coefficients "jump" when the underlying surface changes between land and sea. This is consistent with our observations of surface wind behavior.

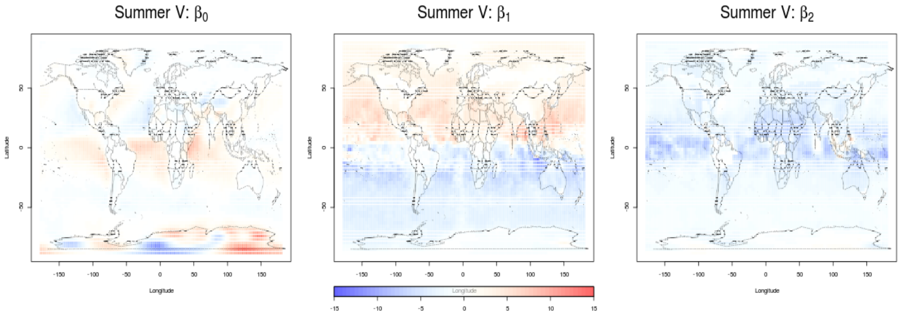

FIGURE A.4.2. Posterior means of β_V terms for Summer. $\beta_{v,1}$ has a larger magnitude as expected. Moreover, the sign change of the coefficients and decreasing magnitude away from the equator is consistent with Equation 3.4.2. The coefficients "jump" when the underlying surface changes between land and sea. This is consistent with our observations of surface wind behavior.