THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF MATHEMATICS

FITTING A BAYESIAN MODEL BASED ON MONTE CARLO MARKOV CHAINS FOR SPARSE DATA

HELAL EL-ZAATARI
SPRING 2019

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Mathematics and Physics
with honors in Mathematics

Reviewed and approved* by the following:

Matthew Reimherr
Professor of Statistics
Thesis Supervisor

Sergei Tabachnikov
Professor of Mathematics
Honors Adviser

Abstract

We are currently trying to fit a functional regression model on a collection of functions. However, these functions are very erratically observed and are thus sparse. The approach we used was trying to fit a Bayesian model based on Monte Carlo Markov Chains. Bayesian models are a lot better at handling sparse or missing variable problems such as this one. Specifically for this thesis, we used the Monte Carlo Markov algorithm called Gibbs Sampling. This method updates its parameters based on the target density. This allows us to break complex problems into a series of easier but interrelated problems. Our goal was to estimate a function $f(t) = 20(t - 2t^2 + t^3)$.

Table of Contents

Acknowledgements

I would like to thank my family and my thesis adviser Matthew Reimherr.

Chapter 1

Introduction

Functional data analysis is concerned with the statistical analysis of data where one or more variables of interest is a function. This field has seen rapid progression over the last two decades. Functional Data Analysis has been applied to a myriad of different and broad fields such as Medicine, Business, Finance and Engineering [1]. For example consider the heights of 10 girls measured at a set of 31 ages in the Berkeley Growth Study.[2]. Below is a graph of their respective heights:

In this example, the data consists of 10 functional observations of the the patient's height. As with any measurement procedure there will be random errors which shall be referred to as noise. Note that the raw data is discrete and they are indicated above by circles. We wish to display features which can not be directly inferred from the graphs above. Say for example we want to view the acceleration curves from the data above. In order to achieve this we take the second derivative of the 10 height functions shown above and the following is obtained [2] :

From the graphs above one can deduce that pubertal growth spurt shows up as a sharp maximum at around 8 to 12 years old and is then followed by a sharp decrease. This illustration

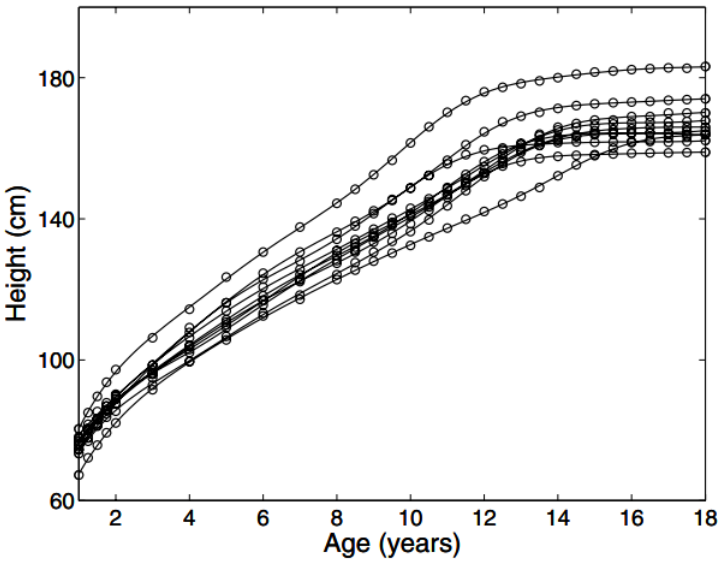


Figure 1.1: The heights of 10 girls measured at 31 ages. Circles indicate the unequally spaced ages of measurements [2].

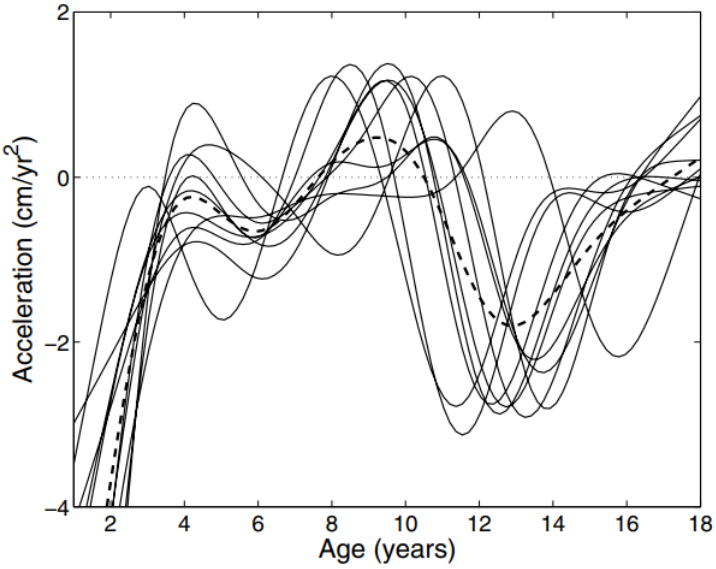


Figure 1.2: Estimated accelerations of height for 10 girls. Dashed line represents the cross-sectional mean [2].

scratches the surface of the applications functional data analysis has to offer. In fact, this thesis was a simulated data study motivated by a longitudinal study on Macrocephaly [3]. Macrocephaly is a medical condition in which children have particularly large head circumference.

The model in this thesis was a Scalar-on-function regression model. Linear regression is one of the most fundamental tools in statistics. In linear regression, the purpose is to better understand the functional dependence of one variable on another. In it's simplest form we have a relationship of the form [4]:

$$Y_i = \alpha + \beta x_i + \epsilon_i \tag{1.1}$$

Here Y_i is a random variable and x_i is a variable which is observed. The quantities α and β are respectively called the intercept and slope. Finally, ϵ_i is the error function and is a random variable [4]. The main goal of regression is to predict the random variable Y_i from the observed variable x_i. Our model differs from by having the observed variable x_i and intercept as functions instead of scalars. Thus the scalar-on-function model is [1]:

$$Y_i = \int \beta(s) X_i(s) ds + \epsilon_i \tag{1.2}$$

In order to predict Y_i we will use a form of statistical inference called Bayesian inference. Bayesian inference is based on Bayes Theorem which states:

Theorem 1 *Consider two events A and B with $P(B) \neq 0$. Bayes theorem states that $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ where $P(A|B)$ is the likelihood of event A occurring given that B is true [5].*

Bayesian inference derives the posterior probability and does so by relying on both the "prior probability" and the "likelihood function". For example, in the theorem above $P(A)$ is our prior probability and $P(B|A)$ is the likelihood which indicates the compatibility of the evidence with the given hypothesis. Finally, the posterior probability $P(A|B)$ is simple the probability of A after B has been observed. [6].

Bayesian inference is usually used as a form of "updating" rule. In which one puts down a form of prior distribution before any data is observed. This prior reflects the experimenters belief

about the nature of the data. These priors themselves may have hyper-prior distributions which in turn express beliefs about their values. These models which contain more than one level of prior are called a hierarchical Bayes model. The model in this thesis was hierarchical. Bayesian inference then returns a distribution over possible points. Thus Bayesian inference is a theory of how to interpret observed data. It is possible to combined Bayesian Inference with another independent discipline called Markov Chain Monte Carlo (MCMC). This method is mainly used to sample from a distribution. When these two methods are combined we can answer questions such as "Given the outcome of an experiment what is the probability of a cause as compared to some other cause?" [7]. Markov Chain Monte Carlo methods are used in order to help solve a problem set up by Bayesian inference. Indeed suppose we have a probability distribution function of some set of parameters θ given the data. The distribution is written as $p(\theta|D)$ where D is the data. Usually such distributions do not have analytical solutions thus numerical methods are needed in order to find solutions. Thus, we use MCMC methods to provide an easy and efficient way to sample points from this distribution [7]. This is the reason why Bayesian-MCMC methods have been used in a myriad of different fields.

Throughout the rest of this thesis, more mathematical theory will be provided in the next section. Rigorous definitions and descriptions of Markov Chain Monte Carlo, Scalar-on-function regression, Bayesian inference, Metropolis Hastings and Gibbs algorithm will be provided in the next section. Then, in the section after that, we will discuss the computational details of our simulation. Finally, the results obtained will be discussed and analyzed.

Chapter 2

Mathematical Framework

In this section we will rigorously describe two Markov Chain Monte Carlo methods for obtaining a sequence of random samples from a probability distribution. These two methods are the Metropolis-Hastings Algorithm and Gibbs sampling. Additionally, a rigorous description and definition of Bayesian inference is provided as well as a rigorous description of the model used in this thesis is provided.

2.1 Markov Chain Monte Carlo Methods

A Markov chain is a stochastic model which describes a sequence of possible events in which each event depends only on the state observed in the previous event [8]. A stochastic model is a collection of random variables defined on the same probability space $(\chi, \Re, P)$. Here, χ is our state space which is simply a measure space, and $\Re$ is our σ-algebra and finally P is our measure [9]. Thus, it is a sequence of dependent random variables:

$$\{X^{(t)}\} = X^{(0)}, \ldots, X^{(t)}, \ldots \tag{2.1}$$

The defining characteristic of Markov Chains is that the probability distribution of $X^{(t)}$ only depends the previous probability distribution that is $X^{(t-1)}$ Specifically,

$$X^{(t+1)}|X^{(0)}, ..., X^{(t)} \sim K(X^{(t)}, X^{(t+1)}) \tag{2.2}$$

Here $K(X^{(i)}, X^{(t)})$ is a probability distribution that depends only on $X^{(t)}$. It is a time homogeneous chain. It is called the Markov kernel. Monte Carlo Markov chains contain a strong stability property [10]. This means that, by construction there exists a probability distribution f such that if $X^{(t)} \sim f$ then we have that $X^{(t+1)} \sim f$. Thus the Markov kernel K and the stationary distribution f satisfies the following:

$$\int_{\chi} K(x, y) f(x) dx = f(y) \tag{2.3}$$

Here χ is the state space. The existence of the stationary distribution f implies that the Markov Kernel K has a positive probability of eventually reaching any region in χ if it has positive measure [10].

Markov Chain Monte Carlo is a class of algorithms used in order to sample from a probability distribution. The most general Markov Chain Monte Carlo algorithm is the Metropolis-Hastings algorithm [7]. In order to sample a distribution $f(x)$ on a state space E, with $x \in E$ we construct a transition kernel $K(x, y)$ which goes from x to y. The Metropolis-Hastings algorithm is then a two step process [7]:

The algorithm starts with a starting point x_1 and a target density $f(x)$ and an associated conditional density q, called the candidate distribution, which then produces a Markov chain $X^{(t)}$ by first generating $Y_t \sim q(y|x^{(t)})$. Here $x^{(t)}$ is the value of $X^{(t)}$ at a given time t. Then we update the Markov chain based on probability function

$$\rho(x, y) = min[\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}, 1] \tag{2.4}$$

such that,

$$X^{(t+1)} = Y_t \text{ with probability } \rho(x^{(t)}, Y_t) \tag{2.5}$$

$$X^{(t+1)} = x^{(t)} \text{ with probability } 1 - \rho(x^{(t)}, Y_t). \tag{2.6}$$

The performance of this algorithm is strongly dependent on our choice of q [10].

Gibbs sampling was introduced by Geman in 1984 [7]. It is another Marko Chain Monte Carlo algorithm for obtaining a sequence of observations which are approximated from a multivariate probability distribution when direct sampling is difficult. This algorithm is commonly used with Bayesian inference; which we will be discussed more rigorously in the next subsection.

Suppose we want to sample a distribution $f(x)$ where $x \in \chi$. χ is our state space. Gibbs sampler works by splitting the Markov Kernel $K(x, y)$ into multiple steps.First, suppose that the random variable $X \in \chi$ can be written as $X = (X_1, X_2, ..., X_p)$ where $p > 1$ and χ is our state space. Here each X_i can be either one-dimensional or multi-dimensional. We then simulate the corresponding conditional densities $f_1, ...f_p$. That is we have

$$X_i|x_1, x_2..., x_i, ..., x_p \sim f_i(x_i|x_1, x_2, ..., x_i, ..., x_p) \tag{2.7}$$

The densities $f_1, f_2, ..., f_p$ are called full conditionals. The algorithm for Gibbs Sampler is as follows. At iteration $t = 1, 2, ...$ given the observations $\mathbf{x}^{(t)} = (x_1^{(t)}, ..., x_p^{(t)})$ we generate the following algorithm [10]:

1. $X_1^{(t+1)} \sim f_1(x_1|x_2^{(t)}, ..., x_p^{(t)})$

.

.

.

p. $X_p^{(t+1)} \sim f_p(x_p|x_1^{(t+1)}, ..., x_{p-1}^{(t+1)})$

2.2 Bayesian Regression

The fundamental principle of Bayesian inference is that probability theory is the correct way to describe uncertainty which is describable by a probability distribution. In Bayesian Inference, there is a ”true” unknown parameter value.

First we let x be a data point or a vector of values. This data point has a distribution namely $x \sim p(x|\theta)$. Here θ is the parameter and can be a vector. Additionally this θ may have a distribution. Namely $\theta \sim p(\theta|\alpha)$. We call α a hyper-parameter (or simply prior) which means it is the parameter of a prior distribution. Now let $\mathbf{X}$ be a sample of size equal to n. Thus $\mathbf{X} = x_1, \ldots x_n$, where each

x_i is an observed data point. Finally, let $\tilde{x}$ be the predicted data point. The central idea behind Bayesian inference is the following equation determined by Bayes's Theorem [4]:

$$p(\theta|\mathbf{X},\alpha) = \frac{p(\mathbf{X}|\theta,\alpha)p(\theta|\alpha)}{p(\mathbf{X}|\alpha)} \propto p(\mathbf{X}|\theta,\alpha) \tag{2.8}$$

Here $p(\theta|\mathbf{X},\alpha)$ is called the posterior distribution. In essence it means that the posterior distribution is directly proportional to the likelihood multiplied by the prior respectively.

When methods of Bayesian inference is used in the statistical analysis of regression we obtain an approach called Bayesian regression. The model of this thesis is a scalar-on-function regression model recall that our model is of the form:

$$Y_i = \int \beta(s)X_i(s)ds + \epsilon_i \tag{2.9}$$

First Bayesian regression is used when the regression model of interest has a normally distributed error function. Indeed we have that $\epsilon_i \sim N(0,\sigma_\epsilon^2)$ where 0 is the mean and σ_ϵ^2 is the variance. When the prior distribution is a Gaussian process we refer to this Bayesian regression model as a Gaussian process regression model. Which is simply an ordinary Bayesian regression with infinite dimensional parameter space of unknown nonlinear regression functions [9].

A Gaussian Process is a stochastic process. Recall that a stochastic process is a collection of random variables defined on the same state space $(\chi,\Re,P)$ Where χ is our measure space, $\Re$ is our σ-algebra and P is our probability measure. Formally, a stochastic process can be written as $\{Y_t(\omega) : t \in T\}$ where t is our index and T is the set of indices. For fixed t, $Y_t(\omega)$ is a random variable [9]. Our index set T can be multi-dimensional. With this in mind a Gaussian process is a stochastic process parmaterized by its mean function and its co-variance function[9]. Respectively:

$$\mu(.) : T \to \mathbb{R}, \mu(\mathbf{x}) = E[Y(\mathbf{x})] \tag{2.10}$$

$$k(.,.) : T^2 \to \mathbb{R}, k(\mathbf{x}, \mathbf{x}') = Cov(Y(\mathbf{x}), Y(\mathbf{x}')) \tag{2.11}$$

Now Kolmogorov's extension theorem states that [11]:

Theorem 2 *Let T be an index set and consider $n, k \in \mathbb{N}$ and a finite sequence $t_1, \ldots, t_k \in T$ and let $\nu_{t_1}, \ldots, \nu_{t_k}$ be probability measures on $(\mathbb{R}^n)^k$. Additionally, we require that these measures satisfy two properties:*

1. For any permutation π of $\{1,\ldots, k\}$ and measurable sets $F_i \subseteq (\mathbb{R}^n)$ we have that $\nu_{t_{\pi(1)},\ldots t_{\pi(k)}}(F_{\pi(1)} \times \cdots \times F_{\pi(k)}) = \nu_{t_1,\ldots,t_k}(F_1 \times \cdots \times F_k)$

2. $\forall F_i \subseteq (\mathbb{R}^n), m \in \mathbb{N}$ we have that $\nu_{t_1,\ldots,t_k}(F_{\pi(1)} \times \cdots \times F_{\pi(k)}) = \nu_{t_1,\ldots t_k,t_{k+1},\ldots,t_{k+m}}\Big(F_1 \times \cdots \times F_k \times \underbrace{\mathbb{R}^n \times \ldots \mathbb{R}^n}_{m}\Big)$

If the measures satisfy these two properties then there exists a probability space $(\chi, \Re, P)$ and a stochastic process $\mathbf{X} : T \times \chi \to \mathbb{R}^n$ which is finite dimensional relative to the measures $\nu_{t_1 \ldots t_k}$. More precisely we have:

$$\nu_{t_1,\ldots t_k,t_{k+1}}(F_1 \times \cdots \times F_k) = P((X)_{t_1} \in F_1, \ldots, \mathbf{X_{t_k}} \in F_k$$

Thus due to Kolmogorov's extension theorem we can characterize our Gaussian Process by finite-dimensional distributions this leads us to the following definition

Definition : We say that $Z(t) : t \in \mathbb{R}$ is a Gaussian process with mean function $\mu(t)$ and co-variance function $C(t, s)$ if for any $t_1, t_2, ..., t_n \in \mathbb{R}$ we have that each $Z(t_1), ..., Z(t_n)$ is a multivariate normal with mean and co-variance matrix respectively found below:

$$\begin{bmatrix} \mu(t_1) \\ \vdots \\ \mu(t_n) \end{bmatrix}, \begin{bmatrix} C(t_1,t_1) & \dots & C(t_1,t_n) \\ \vdots & \ddots & \vdots \\ C(t_n,t_1) & \dots & C(t_n,t_n) \end{bmatrix}$$

The co-variance function $k(.,.)$ posses an intrinsic property that it is a non-negative definite function [11] which means that $\forall n \in \mathbb{N}$ every $\mathbf{x_1}, \dots, \mathbf{x_n} \in T$ and coefficients $a_1, \dots, a_n \in \mathbb{R}$ we have

$$\sum_{i,j=1}^{n} a_i a_j k(\mathbf{x_i}, \mathbf{x_j}) \geq 0 \tag{2.12}$$

There is a wide number of different co-variance function choices, for example the Matern covariance function. The Matern covariance can be written as

$$C_\nu = \sigma^2 (\frac{2^{1-\nu}}{\Gamma(\nu)})(\sqrt{2\nu}\frac{d}{\rho})^\nu K_\nu(\sqrt{2\nu}\frac{d}{\rho}) \tag{2.13}$$

Here d is the distance separated by two points. Γ is the usual gamma function, K_ν is the modified Bessel function. Our parameters are ν and ρ, these parameters are non-negative. Finally, σ^2 is the variance. [12].

In estimating the unknown regression function $f(x)$, the co-variance function $k(.,.)$ plays a central role in determining the smoothness of the data as well as the sample paths of the Gaussian process. Choosing a suitable co-variance function as a prior distribution for the regression function is thus vital in order to achieve posterior consistency [11].

The main goal of Bayesian inference is to treat all the unobservable parameters as random variables and then use Bayes's Theorem and conditional probabilities to express these forms of uncertainty. Before the data is even observed, we quantify our uncertainty about the parameters by assigning them prior distributions. We then update our opinions about the unknown parameters through the observed data and a conditional distribution of the parameter given the data. This con-

ditional distribution is called the posterior distribution [11].

Let $\mathbf{D} = \{(y_i, \mathbf{x_i})\}$ be our data set. Here $i = 1, \ldots, n$ and $\mathbf{x_i} \in T \subset \mathbb{R}^Q$. Consider this regression model with known variance and hyper-parameters θ for $k(\mathbf{x}, \mathbf{x}'; \theta)$ [11]

$$y_i = f(\mathbf{x_i}) + \epsilon_i, i = 1, \ldots, n \tag{2.14}$$

$$\epsilon_i \sim N(0, \sigma^2) \text{ with } \sigma^2 \text{ known} \tag{2.15}$$

$$f(.) \sim GP(\mu(.), k(.,.)) \text{ and } Cov(f(\mathbf{x_i}), f(\mathbf{x_j})) = k(\mathbf{x_i}, \mathbf{x_j}) \tag{2.16}$$

In order to determine the posterior distribution of $\mathbf{f} = (f(x_1), \ldots, f(x_n))^T$ we note that the observed responses $\mathbf{y} = (y_1, \ldots, y_n)$ then note that

$$(y_1, \ldots, y_n | \mathbf{f}, \sigma^2) \sim N(\mathbf{f}, \sigma^2) \tag{2.17}$$

$$\mathbf{f} \sim N_n(\mathbf{0}_n, \mathbf{K}) \tag{2.18}$$

Here K denotes the $n \times n$ covariance matrix evaluated at all pairs of the n points. Thus the posterior distribution of $\mathbf{f}$ will be proportional to the product of two n-variate normal distributions $\psi_n(.|\mu, \mathbf{\Sigma})$. Here μ is the mean vector and Σ is the covariance matrix [11].

$$p(\mathbf{f}|\mathbf{D}, \sigma^2) \propto \psi_n(\mathbf{y}|\mathbf{f}, \sigma^2\mathbf{I})\psi_n(\mathbf{f}, \mathbf{0}, \mathbf{K}) \tag{2.19}$$

Now when both the variance and the hyper-parameters θ we can obtain the posterior distribution through analytically via the multivariate normal distributions described above. However, the variance and hyper-parameters are not known we implement a Markov Chain Monte Carlo method, specifically Gibbs sampling [11]. We can calculate the posterior distributions of $\mathbf{f_n}$ and σ^2 by assigning suitable priors. For example a typical prior for σ^2 is the inverse gamma distribution.

An inverse gamma distribution has two hyper-parameters α and β. It's probability density function defined over a support $x > 0$ is as follows:

$$IG(x, \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}(\frac{1}{x})^{\alpha+1}e^{\frac{-\beta}{x}} \tag{2.20}$$

Chapter 3

Computational Details of Simulation

3.1 Preparing the Simulation

Our model is based on the following a Scalar-on-function regression model. It has the general form of

$$Y_i = \alpha + \int \beta(s) X_i(s) ds + \epsilon_i \tag{3.1}$$

Here $i = 1, .., N$ and the regressors are curves, however the dependent variable Y_i are scalars. The parameter in this model function $\beta(s)$. Finally, ϵ_i is an error function which has a normal distribution with mean zero and standard deviation of the error function is estimated by the sample variance. [1]

We then transform the integral in the above equation into a sum since we observe these functions on a discrete set of points. We then have

$$Y_i = \alpha + \frac{1}{m} \sum_{j=1}^{m} X_{ij} \beta_j + \epsilon_i \tag{3.2}$$

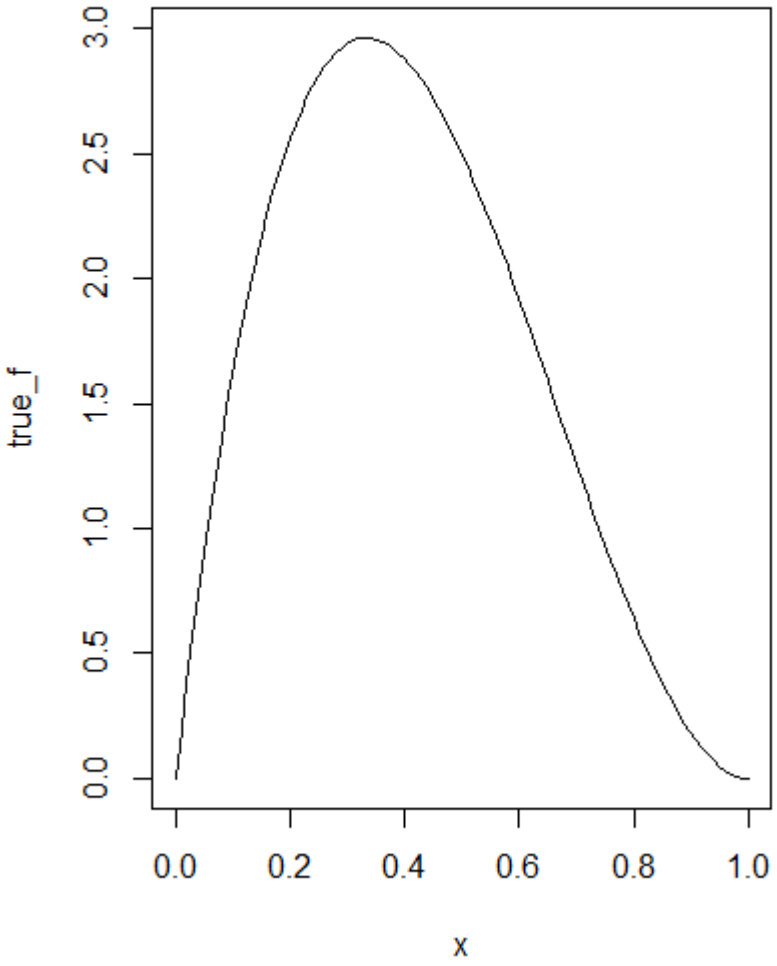
Now α is simply the intercept and is normally distributed function $\alpha \sim N(\mu_0, \sigma_\epsilon^2)$. Additionally the error function is also a normally distributed function such that

$$\epsilon_i \sim N(0, \sigma_\epsilon^2) \tag{3.3}$$

This thesis is a simulated data study, thus our true target density function is

$$f(t) = 20(t - 2t^2 + t^3) \tag{3.4}$$

Our true function $f(t)$ is simply the β_j term in equation (3.4) and it takes values t from 0 to 1. Our function $f(t)$ is plotted below:



Furthermore, let intercept have a value of $\alpha = 0.1$ and let the true value of the mean in our error function be zero with variance $\sigma_\epsilon^2 = 0.1$.

In order to simulate the observed points Y_i in equation (3.2). We take a sequence of $\{t_i\}$ where each $t_i = \frac{i}{20}$ and $i = 0, \ldots, 20$ and then randomly generate a Gaussian process for our function X_{ij} with its co variance being a Matern process with $\nu = \frac{1}{2}$ and variance 1. Thus our Guassian process will be of the form

$$C_{\frac{1}{2}}(d) = \sigma^2 e^{\frac{-d}{\rho}} \tag{3.5}$$

Then to simplify our code we combine the X_{ij} matrix and the intercept α . Then we proceed to simulate the Y_i scalars based on equation (3.2) above. Find below a plot of the points Y_i, our sample size was $n = 1000$. Thus we obtain 10,000 randomly generated points for our scalars Y_i as shown below.

3.2 Priors and Posteriors for Simulation

Now the prior distribution for the variance σ_ϵ^2 was chosen to be an inverse gamma distribution with hyper-parameters α_ϵ and β_ϵ. These two hyper parameters are respectively called the shape parameter and scale parameter [4]. Indeed we have that

$$\sigma_\epsilon^2 \sim IG(\alpha_\epsilon, \beta_\epsilon) \tag{3.6}$$

This inverse gamma distribution requires two priors one for α and one for β. The values for both hyper parameters α and β are chosen to be 1.

The priors for our Gaussian process $\beta(s) \sim N(\mu_0, \Sigma_0)$ are required. Here μ_0 and Σ_0 are the prior mean and prior variance respectively. The prior chosen for the mean is the zero matrix with $n = 1000$. Now for the prior of the variance is a Matern Covariance function was chosen with a range parameter of 1 for smoothing purposes. Our prior will be of the form

$$(1 + \gamma d)exp(-\gamma|d|) \tag{3.7}$$

Here γ is the range parameter and d is the difference between our points t_i.

The update for the hyper-parameter α in the variance of ϵ_i is

$$\alpha \to \alpha + \frac{n}{2} \tag{3.8}$$

The update for the hyper-parameter β is

$$\beta \to \beta + \frac{1}{2}\sum(\frac{1}{i} - \alpha - \frac{1}{m}\sum X_{ij}\beta(s)_i)^2 \tag{3.9}$$

Updates for the mean and variance of our $\beta(s)$ are required. The update for the mean will be

$$\mu \to (X^T X + (\sigma_\epsilon)^2\Sigma_0^{-1})^{-1}(X^T Y) \tag{3.10}$$

Now the update for the variance will be

$$\Sigma \to (\sigma_\epsilon)^2(X^T X + (\sigma_\epsilon)^2\Sigma_0^{-1})^{-1} \tag{3.11}$$

Now these parameters will update simultaneously after a certain number of repetitions according to the Multi-stage Gibbs sampler algorithm.

Chapter 4

Results

We vary the sample size in order to compare the effects on the estimation of the true function $f(t)$. Below we analyze the results for $n = 10$, $n = 100$ and $n = 1,000$

Let $n = 1,000$, we will analyze our estimation for the true function$f(t) = 20(t - 2t^2 + t^3)$. Thus we obtained 1,000 estimates for f(t). In one of these draws we will obtain 20 points, call the estimated function $f_r(t)$ where $r = 1, ..., n$. Thus a sequence of estimated functions is obtained. In order to properly compare our results visually with the true function $f(t)$, we have plugged in 20 points $t_i = \frac{i}{20}$ for $i = 0, ...20$ into our true function $f(t)$. Then we plotted these 20 points $f(t_i)$ below.

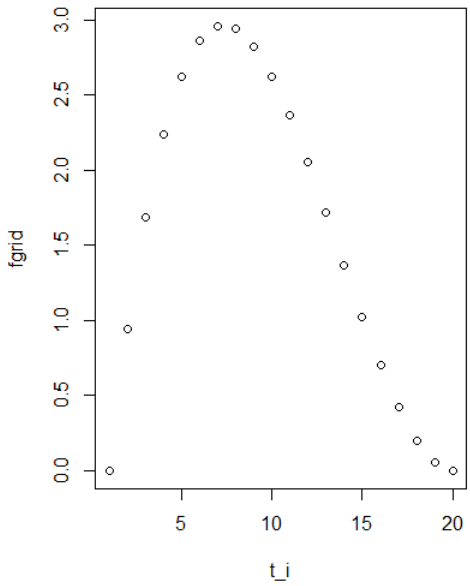


Figure 4.1: Plot of the true function

We will also include the correlation between our generated points Y_i and the 20 points $f(t_i)$ plotted above. The graph below will be linear as expected. We strive for our estimated 20 points to also have a linear correlation and to be as close to possible as the graph shown below.

The 20 points of $f_{994}(t_i)$ are plotted below in Figure 4.3. These are the twenty points resulting from draw 994 in our sequence of estimated functions. As we can see the points are noisy as ex-

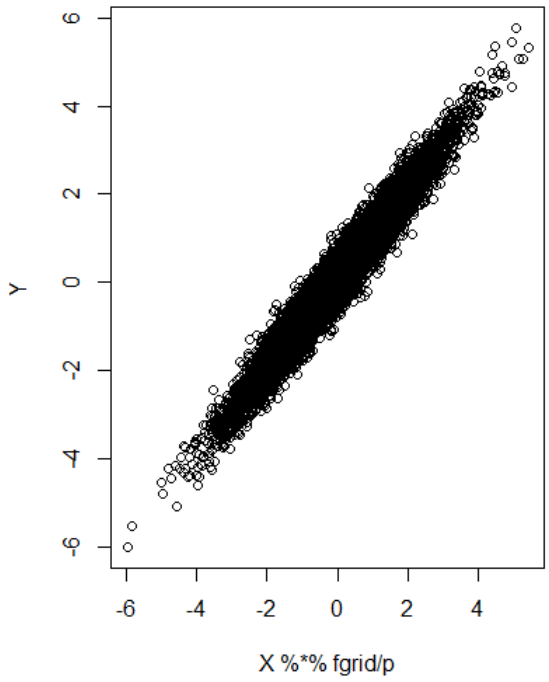


Figure 4.2: Correlation between observed data Y_i and points from the true function $f(t)$

pected. However, we can begin to see the general shape of our true function $f(t)$ forming.

In order to obtain a more accurate estimation we will take the point-wise mean of our sequence of estimated functions. We will fix each i for $i = 1, ..., 20$ and then take the average of $f(t_i)$ for a fixed i. Thus we have that

$$\overline{f(t_i)} = \frac{\sum_1^n f(t_i)}{n} \tag{4.1}$$

This gives us one point. We will thus repeat the above formula for each $i = 1, ...20$ in order to obtain the point-wise mean function $\overline{f(t)}$ of our sequences of functions f_r. Below, in figure 4.4 we have plotted the 20 points of the point-wise mean function $\overline{f(t_i)}$ for $n = 1,000$.

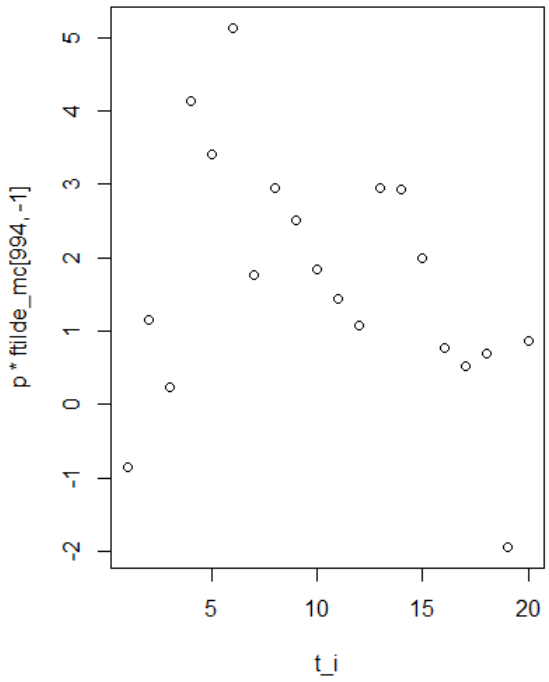


Figure 4.3: A sample draw in order to estimate f(t)

Visually our point-wise mean function $\overline{f(t)}$ is very similar to our true function $f(t)$. We will now plot the correlation between the points $\overline{f(t_i)}$ and our simulated points Y_i. As we can see below there is a clear linear trend and the correlation graph is very similar to the previous correlation graph above, thus confirming that we have adequately estimated our true function $f(t)$.

Below in figure 4.6 was the estimate for f shown for a small sample size of $n = 10$ and $n = 100$ in figure 4.7. One can see that the estimate for $n = 10$ is close to true function $f(t)$ Bayesian inference with Markov Chain Monte Carlo Method methods, specifically Gibbs sampling obtains accurate estimates even with a sample of size of $n = 10$ which is considered sparse. This shows the power of this method when dealing with a small sample size.

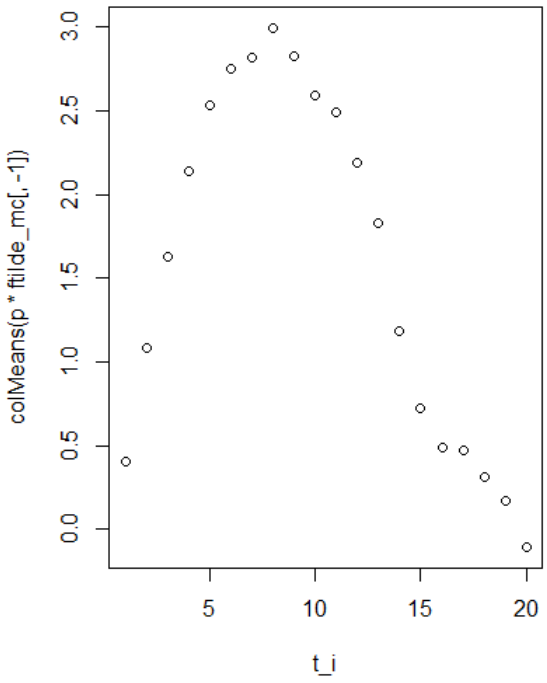


Figure 4.4: Estimation of true function for sample size $n = 1000$

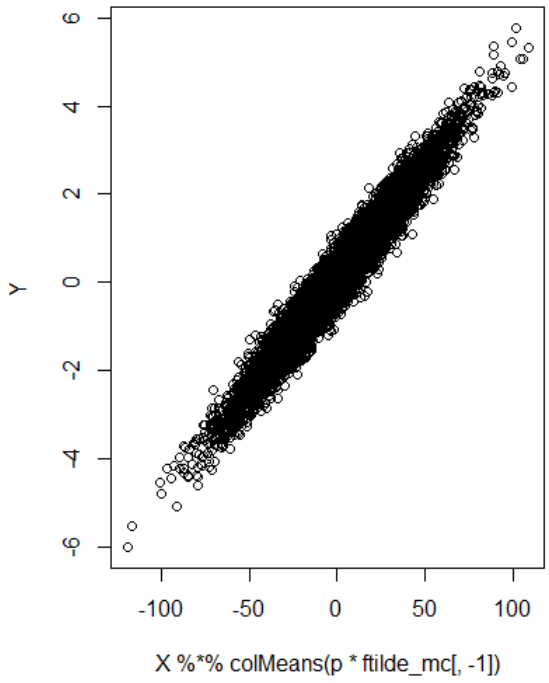


Figure 4.5: Correlation between point-wise mean function and the observed data points Y_i

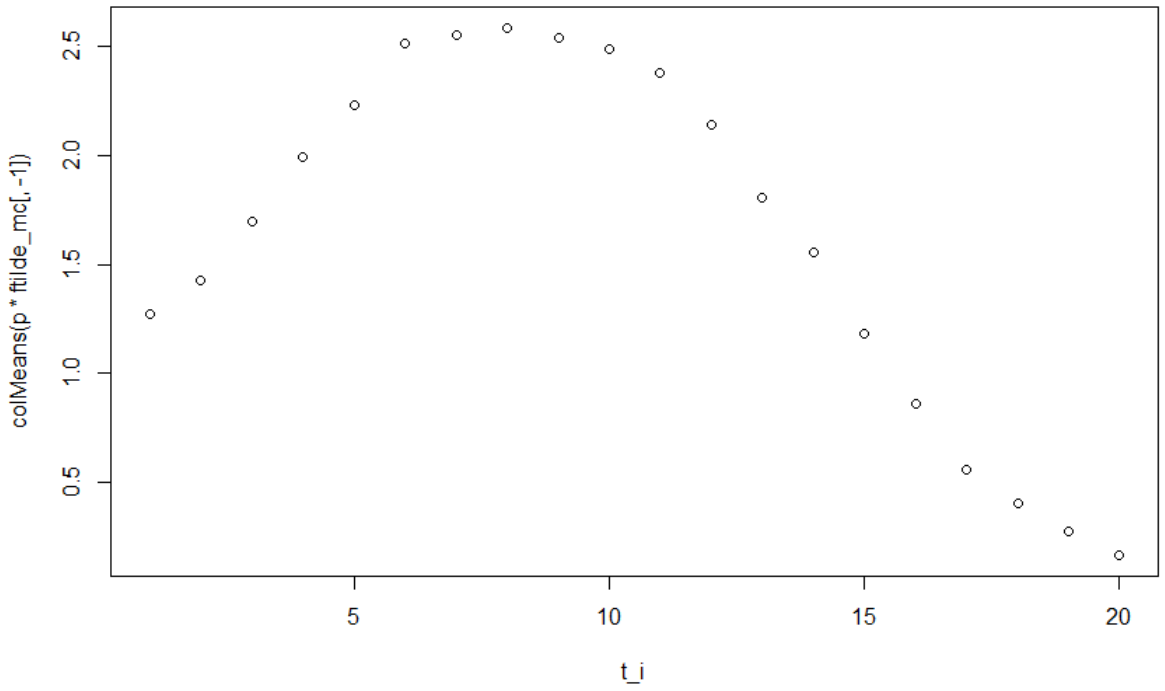


Figure 4.6: Estimation of true function for sample size $n = 10$

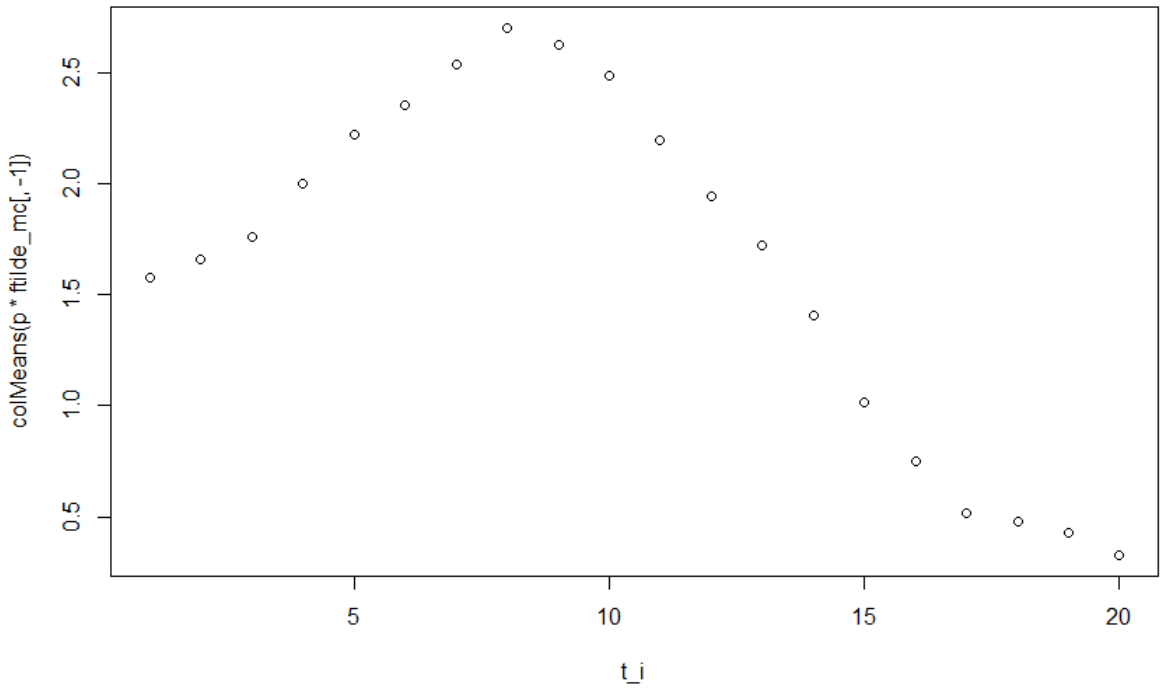


Figure 4.7: Estimation of true function for sample size $n = 100$

Bibliography

[1] P. Kokoszka and M. Reimherr. *Introduction to Functional Data Analysis*. Chapman & Hall / CRC numerical analysis and scientific computing. CRC Press, 2017.

[2] J. Ramsay and B. Silverman. Functional data analysis, 1997.

[3] Justin Petrovich, Matthew Reimherr, and Carrie Daymont. Functional regression models with highly irregular designs, 2018.

[4] G. Casella and R.L. Berger. *Statistical Inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning, 2002.

[5] M. G. Kendall, A. Stuart, and J. K. Ord, editors. *Kendall's Advanced Theory of Statistics*. Oxford University Press, Inc., New York, NY, USA, 1987.

[6] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian data analysis*. Texts in Statistical Science Series. Chapman & Hall/CRC, Boca Raton, FL, second edition, 2004.

[7] Sanjib Sharma. Markov chain monte carlo methods for bayesian data analysis in astronomy. 2017.

[8] Chungman Seo, Bernard P. Zeigler, and Doohwan Kim. Devs markov modeling and simulation: Formal definition and implementation. In *Proceedings of the 4th ACM International Conference of Computing for Engineering and Sciences*, ICCES'18, pages 1:1–1:12, New York, NY, USA, 2018. ACM.

[9] J.Q. Shi and T. Choi. *Gaussian Process Regression Analysis for Functional Data*. Taylor & Francis, 2011.

[10] Christian P. Robert and George Casella. *Introducing Monte Carlo Methods with R (Use R)*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2009.

[11] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Hochschultext / Universitext. Springer, 2003.

[12] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

```
library(MASS)

#Defining our true function and parameters which we wish to estimate

true_f<-function(t){20*(t - 2*t^2 + t^3)}
true_alpha <- 0.1
true_sig2<-0.1

# sample size is n and number of points is p
n<-10000
p<-20
t<-seq(0,1,length=p)
fgrid <- numeric(p)
for ( i in 1:p){
  fgrid[i] = true_f(t[i])
}
ftilde<-c(true_alpha,fgrid)
Y<- rep(0, times=n)
error <- rnorm(n, mean=0, sd=sqrt(true_sig2))

# generating gausssprocess function
library(RandomFields)
model<-RMmatern(nu=1/2,var=1,scale=1)
X<-RFsimulate(model=model,x=t,n=n)
X<-t(as.matrix(X))

#simulate the data Y
```

```
Xtilde<-cbind(1,X/p)
Y<-Xtilde%*%ftilde+error
plot(X%*%fgrid/p, Y)

##########################
#
# STARTING GIBBS PROCESS
#
##########################
reps<-1000
sig2_mc<-numeric(reps)
ftilde_mc<-matrix(nrow=reps,ncol=p+1)
sig2_mc[1] = 1
ftilde_mc[1,] = 0

## Hyperparameters for sig2

a <- 1
b <- 1
#### Setting up the prior for the mean and ####
####  variance of our guassian function    ####

rng_par<-1
d<-outer(t,t,FUN = "-")
Sig_prior<-(1+rng_par*d)*exp(-rng_par*abs(d))
f_post = rep(0,times=n+1)
```

```
mu_prior = rep(0, times=p+1)
fix_sig_prior <- matrix(0,nrow=p+1, ncol=p+1)
fix_sig_prior[1,1] <- 1
fix_sig_prior[2:(p+1),2:(p+1)] <- Sig_prior
fix_sig_prior <- rng*fix_sig_prior
Sig_p_inv<-solve(fix_sig_prior)
X_with_alpha = cbind(1,X)
X_transpose <- t(X_with_alpha)

#initializing the chain ofr sig2
sig2[1] = 1/rgamma(1, shape=a, rate=b)

# we begin the chain

for (i in 2:reps){

  #sig2 update

  a_post<-a + n/2
  b_post<-b + 0.5*sum((Y - Xtilde%*%ftilde_mc[i-1,])^2)
  sig2_mc[i] = 1/rgamma(1, shape=a_post, rate=b_post)


  #mu and variance update

  mu_update = (solve(X_transpose%*%X_with_alpha + sig2_mc[i]*Sig_p_inv))%*%
  variance_update = sig2_mc[i]*(solve(X_transpose%*%X_with_alpha + sig2_mc[
```

```
  ftilde_mc[i,] = mvrnorm(1, mu=mu_update, Sigma=variance_update)
}
```

ACADEMIC VITA

Helal El-Zaatari ⋄ (302)-257-9567 ⋄ hme5072@psu.edu

EDUCATION

Pennsylvania State University
B.S Mathematics (May 2019)
B.S Physics (May 2019)

RESEARCH

Functional Data Analysis
Project involves fitting a funcitonal regression model with erratically observed functions. Fitting a Bayesian model based on Monte Carlo Markov Chains.

Pulsar Search Col-laboratory
Searched for pulsars by analyzing integrated pulse profiles, time domain plots, sub-band plots and dispersion measure plots. Obtain certificate from Green Bank Observatory.

WORK EXPERIENCE

Penn State Learning August 2018
Math Tutor

· Tutor for Penn State Learning Center in Mathematics, taught calculus I,II III, Linear Algebra and Differential equations.

ACADEMIC ACHIEVEMENTS

Schreyer Honor's College and Dean's List all semesters.

Elsbach Honor's Scholarship in Physics and Leonard Euler Memorial Scholarship in Mathematics.

Inducted Junior year into Phi Beta Kappa

EXTRA-CIRRUCULAR

Coding Languages: R, Python, Matlab and Latex

Languages: 1- English (proficient), 2- Arabic (proficient), 3- German (working proficiency) 4- French (elementary)

Piano: played since age 4.