Robust Bayesian Anomaly Detection Methods for Large Scale Sensor Systems

Sierra N. Merkes

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Statistics

Scotland C. Leman, Chair
Eric P. Smith
David M. Higdon
Christopher T. Franck

Wednesday, June 22, 2022
Blacksburg, Virginia

Keywords: Anomaly Detection, Bayesian, Mixture Models, Process Control, Wind Tunnel

Robust Bayesian Anomaly Detection Methods for Large Scale Sensor Systems

Sierra N. Merkes

(ABSTRACT)

Sensor systems, such as modern wind tunnels, require continual monitoring to validate their quality, as corrupted data will increase both experimental downtime and budget and lead to inconclusive scientific and engineering results. One approach to validate sensor quality is monitoring individual sensor measurements' distribution. Although, in general settings, we do not know how to correct measurements should be distributed for each sensor system. Instead of monitoring sensors individually, our approach relies on monitoring the co-variation of the entire network of sensor measurements, both within and across sensor systems. That is, by monitoring how sensors behave, relative to each other, we can detect anomalies expeditiously. Previous monitoring methodologies, such as those based on Principal Component Analysis, can be heavily influenced by extremely outlying sensor anomalies. We propose two Bayesian mixture model approaches that utilize heavy-tailed Cauchy assumptions. First, we propose a Robust Bayesian Regression, which utilizes a scale-mixture model to induce a Cauchy regression. Second, we extend elements of the Robust Bayesian Regression methodology using additive mixture models that decompose the anomalous and non-anomalous sensor readings into two parametric compartments. Specifically, we use a non-local, heavy-tailed Cauchy component for isolating the anomalous sensor readings, which we refer to as the Modified Cauchy Net.

Robust Bayesian Anomaly Detection Methods for Large Scale Sensor Systems

Sierra N. Merkes

(GENERAL AUDIENCE ABSTRACT)

Sensor systems, such as modern wind tunnels, require continual monitoring to validate their quality, as corrupted data will increase both experimental downtime and budget and lead to inconclusive scientific and engineering results. One approach to validate sensor quality is monitoring individual sensor measurements' distribution. Although, in general settings, we do not know how to correct measurements should be distributed for each sensor system. Instead of monitoring sensors individually, our approach relies on monitoring the co-variation of the entire network of sensor measurements, both within and across sensor systems. That is, by monitoring how sensors behave, relative to each other, we can detect anomalies expeditiously. We proposed two Bayesian monitoring approaches called the Robust Bayesian Regression and Modified Cauchy Net, which provide flexible, tunable models for detecting anomalous sensors with the historical data containing anomalous observations.

Dedication

To my ride or die crew: Anne Merkes, Carl Merkes, Cody Merkes, Marina Koliopoulos, and Catie Deneen

Acknowledgments

Graduate school is an adventure of failures upon failures that results in a 100+ page book documenting your successes. Still, these successes would not have been as rewarding without the people that helped me enjoy the journey. My six-year journey as a graduate student started when my Radford University advisor, Anthony Dove, *heavily* pushed me to apply to the Jean D. Gibbons Graduate Program in Statistics at Virginia Tech. I am and forever will be grateful for your belief in what 18-year-old Sierra could come and for your family's support (Emily, Anna, and Owen Dove) throughout my graduate school career with the homecooked meals, entertaining conversations, and also starting my soccer coaching career.

During my Virginia Tech career, I worked with an unbelieve advisor, Scotland Leman, a.k.a "Boss Man", who always believed in my ability to succeed as a graduate student and as a future statistics professor. His guidance helped me become a leader within the department, in my collaborative research projects, and in my life. While I am beyond appreciative of his time and limitless patience when guiding me through learning new statistical concepts and helping me debug my code, I am more grateful for our friendship.

While my advisor helped guide me through my academic career, my family has dealt with me since day one. I would not have made it on this adventure without my mother, Anne Merkes, father, Carl Merkes, brother, Cody Merkes, and my "sisters", Marina Koliopoulos and Catie Deneen. These five beautiful, loving people have seen me through my best and worst days, and I would not be as successful without them. I love y'all!

A journey through graduate school would be mute without friends, so big shout-outs to "The Boys" group: Macaulay Soto, Hailey Akens, Rebekah Arrigo, Emily Whitaker, and Maddie Houlihan, who has helped keep me sane during this process, and for constantly reminding me to take a break and grab a drink! My statistics buddies: Matt Slikfo, Adeline Guthrie, John Smith, Nathan Wycoff, and Chris Grubb, for always wanting to talk about statistics.

I would also like to thank the Statistics department staff: Michele Strauss, Christina Dillon, and Betty Higginbotham, who helped guide me to the correct forms to fill out, understand various travel scholarships/refunds and figure out the other miscellaneous things that go into being a graduate student. Additionally, I would like to thank Steve Slaughter for his incredible patience in answering all my computer-related questions and teaching me how to use our STAT Linux computers.

Finally, thank you to Eric Smith, Dave Higdon, and Christopher Frank for their positions on my committee and invaluable advice through the thesis and job hunting process. Furthermore, I want to thank William Devenport, Nathan Alexander, and Aaron Defreitas in the Department of Aerospace and Ocean Engineering for tutoring me around the Virginia Tech Stability Wind Tunnel and providing valuable datasets to analyze.

Thank you to the Office of Naval Research, particularly Debbie Nalchajian, for their support under grant N00014-17-1-2944.

Contents

Chapter 1

Introduction

Wind tunnels provide engineers an environment to understand and test novel aeronautic innovations. To monitor and analyze their innovations, engineers design large-scale sensor systems to collect, store, and analyze the response of test instruments (or apparatuses). Unfortunately, experimental sensor systems, such as those found in modern wind tunnel experiments, are susceptible to different types of errors. Typically, errors transpire either in the collection process due to erroneous sensor readings or misreported sensor readings in the storage process. If undetected errors persist, scientific and engineering results and conclusions may be compromised or misleading and have financial costs and experimental downtime. Sometimes field experts rectify misreported data in the post-processing analysis phase; however, erroneous sensor readings will likely cripple an experiment's results. Regardless of the error type, we must detect errors early. Thus, the sensor monitoring system requires continual monitoring to ensure reliable scientific conclusions, productive experimental trials, and a cost-efficient budget. This thesis evaluates several anomaly monitoring sensor systems where we refer to various error types as anomalous, anomalies, or outliers.

In 1930, Walter A. Shewhart introduced the manufacturing world to the importance of reducing variation in the manufacturing process. His control charts identified what he called assignable-cause and chance-cause variation. Today, we would label "assignable-cause" variation as a disturbance to the expected pattern, such as an outlier or anomaly, while "chance-

cause" variation is the random background noise [82, 83, 84]. Shewhart's goal was to identify and quantify the uncertainty solely due to the "chance-cause" variation. He wanted to eliminate the effect of the "assignable-cause" observations in the process' variability estimation to aid in detecting future outlying observations.

The univariate Shewhart control charts ranged from monitoring metrics such as the process mean ($\bar{x}$-chart), standard deviation (s-chart), or range (R-chart) for single continuous variables. Additionally, Shewhart developed control charts to monitor discrete processes, such as monitoring the fraction of non-conforming products in p and np control charts. However, with the emergence of the digital age and technological advancements, univariate methodologies became less equipped to handle the increased number of variables and their correlation structure. For example, when correlated variables existed, the correct estimate of the multivariate in-control region was defined differently than the in-control region determined by the univariate charts [3]. Additionally, when solely examining univariate charts, it is difficult to determine the simultaneous error rates from the charts [2, 29, 43].

Researchers began developing methods to accurately and simultaneously monitor multiple variables using techniques such as Hotelling T^2, multivariate cumulative sum procedures (CUSUM), multivariate exponentially weighted moving average chart (EWMA), and principal components. Hotelling T^2 control chart is a multivariate extension of the univariate $\bar{x}$ chart that accounts for the covariance structure of a multivariate normal distribution [71]. A limitation of the Hotelling T^2 charts is that the charts are relatively insensitive to minor or moderate shifts in the mean vector. Multivariate CUSUM [21, 77] and EWMA [62] procedures provide a more sensitive approach to detecting small deviations from the mean compared to Hotelling.

The "classical" control charts are relatively effective when the monitoring system's dimensionality (P) is small. However, as the dimensionality increases, detecting deviations from the mean becomes harder. Several anomaly detection methodologies have been adopted to monitor and validate the quality of aeronautical apparatus, such as different variations of Principal Component Analysis [24, 25] and Gaussian Processes [19]. Gaussian Processes and Principal Component Analysis methodologies rely heavily on quantities of Gaussian-ly distributed data with no large perturbations from the process's intended signal. Researcher often utilize Gaussian Processes for monitoring spatial-temporal problems (e.g., [12, 20, 28]) and modeling computer simulations (e.g., [39, 44]) when the distance between sensors can adequately be defined and measured. However, we have not found Gaussian Processes practical for our wind tunnel experiments due to the continual need to re-measure the regularly changing sensor space; thus, we will not discuss Gaussian Processes in this thesis.

Principal Component Analysis (PCA) [53] and modifications of PCA [24] can provide a powerful modeling framework for inferring covariation in high dimensional datasets. Principal Components Analysis methods equivalently assumes that observations come from:

$$\underline{x}_i \overset{\text{i.i.d}}{\sim} \text{MVN}\left(\underline{0}, \Sigma_{P\times P}\right),$$

where i.i.d stands for independent and identical distributed and MVN denotes multivariate normal distribution with sampling density:

$$f_{\text{MV.Normal}}\left(\underline{x}_i|\underline{\mu}, \Sigma_{P\times P}\right) = |2\pi\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(\underline{x}_i-\underline{0})'\Sigma^{-1}(\underline{x}_i-\underline{0})},$$

where $\underline{x}_i$ is a $P \times 1$ vector denoting a single, observational run with P sensors for $i = 1, \ldots, N$ total observations. We use $\mathbf{X}_{N \times P}$ to denote the collection of all N observations. The PCA model assumption of $\underline{0}$ mean stems from centering the data. From this assumption, PCA utilize the spectral decomposition of an estimated covariance matrix to identify a reduced dimensional subspace that explains the majority of variation within the sensor systems. The standard spectral decomposition of a covariance structure, $\hat{\Sigma}$, is:

$$\hat{\Sigma}_{P \times P} = \mathbf{Q}_{P \times P} \Lambda_{P \times P} \mathbf{Q}'_{P \times P}, \tag{1.1}$$

where the p^{th} column in the orthonormal matrix, $\mathbf{Q}$, represents the eigenvector associated with the p^{th} eigenvalue in the diagonal matrix, Λ, which orders the eigenvalue magnitude from $\lambda_1 > \lambda_2 > \cdots > \lambda_p > \cdots > \lambda_P$. Under the multivariate normal assumption, the covariance estimator used in Principal Component Analysis is:

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\underline{x}_i - \underline{\bar{x}})(\underline{x}_i - \underline{\bar{x}})',$$

where $\underline{x}_i$ is the i^{th} row of $\mathbf{X}_{N \times P}$, and $\underline{\bar{x}}$ is a $P \times 1$ vector of the column means (i.e., sensor means). We refer to PCA as a k-modal analysis because the method bases its outlier detection on a reduced covariance matrix described by the k highest variance sub-dimensions of the sensor space. The reduced covariance structure is:

$$\hat{\Sigma}_k = \mathbf{Q} \Lambda_k \mathbf{Q}', \tag{1.2}$$

where $\mathbf{Q}$ is equivalent to that in Eq. 1.1, and Λ_k represents a diagonal matrix with the

first largest k eigenvalues from Λ, and the $P-k$ remaining diagonal elements are zeros. See Appendix B for the full matrix representation of Λ_k and Λ_k^{-1}. For choosing the number of modes, k, there are various variable selection techniques such as a scree plot or cross-validation methods [53]. We use an eigenvalue percent variation technique and choose the value of k such that the sum of the first k eigenvalues divided by the total sum of the eigenvalues is equivalent to a chosen percentage of variation. Using the k-modal reduced covariance, we calculated the expectation of a new observation, $\mathbb{E}[\underline{x}_{\text{new}}]$, using:

$$\mathbb{E}[\underline{x}_{\text{new}}] = \hat{\Sigma}(\mathbf{Q}\Lambda_k^{-1}\mathbf{Q}')\underline{x}_{\text{new}}, \tag{1.3}$$

where $\hat{\Sigma}$ denotes the estimated covariance for the data. The uncertainty of each new observation estimate is calculated by square rooting the diagonal elements of:

$$\text{Cov}[\underline{x}_{\text{new}}] = \mathbf{Q}(\Lambda - \Lambda_k)\mathbf{Q}'. \tag{1.4}$$

Given the mean estimates and uncertainty values for a new observation, we identify an observation as anomalous if the observed run falls outside the uncertainty threshold band. The uncertainty threshold band is determined by multiplying the uncertainty values, found in Eq. 1.4, by a threshold constant. Appendix C provides a simulation study that discusses the impact of the number of chosen modes based on the eigenvalues percent variation technique. For further detail on the derivation and deeper understanding of PCA, reference [53]. An adaptive of PCA is Independent Component Analysis (ICA) which identifies the basis where we want each vector as an independent component of your data rather than the basis that best explains the variability of your data. That is, PCA helps find a reduced-rank representation of the data while ICA helps to find independent sub-elements to represent the data

[18, 91]. Future work will investiage the utility of Independent Component Analysis for large scale sensor systems.

The premise of anomaly detection methodologies is to detect (or flag) observations that deviate beyond the expected process signal. Typically, a deviation from the signal (i.e., outliers) is defined as an observation that falls outside three standard deviations from the mean. Thus, we define a non-anomalous observation (or readings), $\underline{x}_i$, to follow a multivariate normal distribution with P dimensions, denoted by:

$$\underline{x}_i \sim \text{MVN}(\underline{\mu}, \Sigma_{P\times P}) \tag{1.5}$$

for $i = 1, \ldots, N$ with sampling density:

$$f(\underline{x}_i|\underline{\mu}, \Sigma_{P\times P}) = \frac{1}{|(2\pi)\Sigma|^{\frac{1}{2}}} e^{\frac{-1}{2}(\underline{x}_i-\underline{\mu})'\Sigma^{-1}(\underline{x}_i-\underline{\mu})}. \tag{1.6}$$

where $\underline{x}_i$ is a $P\times 1$ vector denoting a single, observational run with P sensors and $|\bullet|$ denotes the determinant. Using the historical data, $\mathbf{X}_{N\times P}$, we train monitoring methodologies to detect anomalous signals by focusing on the covariance estimation.

1.1 Modeling Variability of Covariance

Modeling variability of covariance structure, Σ, can help companies better understand and quantify their sensor systems' uncertainty, thus potentially detecting abnormal results quicker and aiding in their mission to produce quality products or reliable scientific results under

budget. While Principal Component Analysis is a valuable tool for aiding in explaining the variance in high-dimensional space, the method is relatively rigid and inflexible to breaks in model assumptions compared to a Bayesian approach when monitoring for anomalies. In the Bayesian paradigm, we utilize the likelihood function and prior distribution to help build a more flexible more for modeling the variability of the covariance. A common Bayesian approach to model variation in covariance structure, Σ, is through utilizing an inverse-Wishart prior distribution [6]:

$$p_{\text{inverse-Wishart}}(\Sigma|\psi,\Omega) = \frac{|\Omega|^{\frac{\psi}{2}}}{2^{\frac{\psi M}{2}}\Gamma_P(\frac{\psi}{2})}|\Sigma|^{-\frac{\psi+P+1}{2}}e^{-\frac{1}{2}\text{tr}(\Omega\Sigma^{-1})}$$

where $|\bullet|$ is the determinant and $\Gamma_P(\bullet)$ is the multivariate Gamma function. The inverse-Wishart parameters are positive, scalar degrees of freedom, ψ and a $P \times P$ positive-definite matrix, Ω, which represents the centering of the distribution.

There are numerous reasons for choosing the inverse-Wishart prior such as it can (1) aids in developing closed-form solutions, (2) ease the computational burden in our Bayesian inferential techniques. Additionally, based on the choice of hyper-parameters, the prior distribution can calibrate error rates and have an intuitive interpretation for non-statisticians. For instance, if companies had prior knowledge about the relationship between their systems, Bayesian can impose the expert knowledge into the model through Ω_o. On the other hand, if we want to assume no prior knowledge, we could incorporate the "lack of knowledge" into our inferences.

In the Bayesian paradigm, our choice in the prior distribution and its hyperparameters im-

pacts posterior inferences. For instance, several authors [35, 36, 95] provide alternative prior distribution for the covariance structure. These authors decided against using the inverse-Wishart because the ψ_o parameter controls the uncertainty for all variances, the marginal distribution of the variances have low density near zero, and the distribution assumes dependence between correlations and variances. Two proposed adaptions of inverse-Wishart prior developed are the scaled-inverse Wishart [76], and a hierarchical inverse Wishart [45]. In Chapter 2, we motivate and briefly describe the foundational elements of Bayesian statistics along with a discussion of prior distributions.

1.2 Incorporating Mixed Signals

When most large-scale sensor systems collect more data and operate under different conditions, we inherently incur the issue of accumulating mixed signals. For example, a typical mixed signal for process control applications is when we receive both in-control (non-anomalous) and out-of-control (anomalous) data. Under the Principal Component Analysis or other single model frameworks, we may produce inaccurate or unreliable covariance estimates when utilizing both the non-anomalous and anomalous data to construct covariance estimators. We extend our model assumptions, Eq. 1.6, using mixture model methodology to aid in handling various mixed signals and to produce more reliable covariance estimates.

Mixture models will enable us to decompartmentalize and separately characterize the anomalous and non-anomalous data using multiple probabilistic models. From a mathematical standpoint, we represent mixture models as:

$$\underline{x}_i \sim \sum_{k=1}^{K} \pi_k f_k(\theta_k)$$

for $i = 1, \ldots, N$ and $k = 1, \ldots, K$, where $\sum_{k=1}^{K} \pi_k = 1$. The mixture model assumes each observation ($\underline{x}_i$) is one of K populations represented by some probability distribution ($f_k(\theta_k)$). Section 2.4.2 provides more general details on K-finite mixture model methodology.Chapter 5 extends the general K-finite mixture model methodology to a two-component mixture model with a multivariate normal distribution for the non-anomalous and 'non-local,' heavy-tailed prior distribution for the anomalous data [51, 52]. We discuss the 'non-local' distribution in Section 5.1.

1.3 Robust Approaches Techniques

Most of the "classical" methodologies make a strong assumption that the training data is "in-control" and contains no anomalous or outlying readings when quantifying the process' variability. While it is ideal to have no outliers in the training (or historical) data, it is an impractical assumption. So, in practice, we want our monitoring methodologies to be robust to influential outliers. Robust methodologies limit the strong assumptional needs for the training data to be "in control." A common first attempt to develop robust methods is utilizing robust mean and covariance estimators to replace the typically mean and covariance estimator.

The first developments of the robust PCA methods involved applying PCA techniques to a robust covariance estimator such as the affine-equivariant M-Estimator or the Fast - Minimum Covariance Determinant (MCD) estimator. The motivation was to identify principal

components (i.e., the linear combinations) that are not influenced by the outlying observations. The MCD estimator is limited to cases where $N > 2P$ because the estimator is defined from a subset of h observations where $\frac{N+P+1}{2} < h < N$ and whose covariance matrix has the smallest determinant [80]. While this restriction may be practical in some applications (e.g. [46, 81]), it is not always an appropriate assumption for applications like Virginia Tech Stability Wind Tunnel experiments due to continual alterations in wind tunnel configurations. Huber [47] developed Robust Principal Component Analysis (ROBPCA) that merges a robust covariance estimate, the Fast-Minimum Covariance Determinant, with projection pursuit.

The inclusion of projection pursuit alleviates the $N > 2P$ limitation by bypassing the dimensionality issue. Projection pursuit applies a brute force process of evaluating low-dimensional projections of high-dimensional data to find "interesting" patterns [34]. Like PCA, ROBPCA uses the k-modal tuning parameter; however, ROBPCA has an additional tuning parameter, $\alpha \in [.5, 1]$, that influences the robustness of the method. After a simulation study where we analyzed ROBPCA at three levels (low: $\alpha = .5$, medium: $\alpha = .75$, high: $\alpha = 1$), we saw no "serious" difference between the α values and the results did not compete well against the robust Bayesian method to show the results. For more details about Robust Principal Component Anaylsis covariance estimator, see [47].

Through the Bayesian perspective, we propose two robust monitoring techniques that utilize heavy-tailed distribution to reduce the influence of outliers in the historical data. Chapter 4 describes a Bayesian scale-mixture approach to induce a Cauchy regression while Chapter 5 describes a two–component mixture model that compartmentalizes anomalous and non-anomalous observations utilizing a non-local, heavily tail distribution. Before discussing

our proposed monitoring methods, we briefly outlines the foundational elements of Bayesian statistics as well as the Monte Carlo algorithms need to perform the analysis.

Chapter 2

Bayesian Statistics

In 1763, Richard Price published Reverend Thomas Bayes' paper "An Essay Towards Solving a Problem in the Doctrine of Chances," providing the first detailed theoretical glimpse into probability theory now associated with Bayes' name [7]. Additionally, Price included an appendix to Bayes' paper that provided illustrative applications such as the Rising of the Sun problem to elevate Bayes' work and discussed forecasting [30, 88]. In the 1770s, Pierre Simon Laplace laid the groundwork for modern Bayesian statistics by elaborating on the work of Sir Thomas Bayes by modernizing the notation, stating their prior distribution decision explicitly, and developing the "Laplace's Rule of Succession" [55]. However, during the infancy of statistics, statisticians did not utilize terms like "frequentist" and "Bayesian" to describe their methods or perspectives. Rather "Bayesian" ideas were connected to the phrase "inverse probability" when De Morgan [73] wrote about the inverse probability methods and attributed its general form to Laplace's 1812 book [56] until the mid-1900s. The word "inverse" refers to the methods inferring the parameter of interest given the data.

Throughout the early 1900s, there were numerous advancements in the statistical field, starting with William Gosset's development of the t-distribution in 1908. In the 1920s, R.A. Fisher introduced key concepts and terminology like likelihood functions, maximum likelihood estimates, the statistical notions of sufficiency and efficiency, and using the label parameter, [31]. Additionally, Fisher provided a formalized approach for tests of significance

[32] which Jerzy Neyman and Egon Pearson extended to "complete" Fisher's work because they believed Fisher's work lacked mathematical detail. Neyman and Pearson's work introduced hypothesis testing and confidence intervals commonly connected with "classical" or "frequentist" statistics. In the 1940s, Richard von Mises criticizes Neyman's confidence interval method using a Bayesian argument [70, 99] which we utilize as a motivation to demonstrate the difference between the two paradigms in the following sections.

From a Bayesian standpoint, Harold Jeffreys published the Theory of Probability, which outlined the inverse probability approach of updating degrees of beliefs in inferences using, most notably, Bayes' theorem to learn from experience and data. Additionally, he discussed his objective prior approach to inducing a lack of knowledge known as Jeffreys priors [49]. Meanwhile, Bruno de Finetti provided a different justification for subjective probability, introducing the notion of exchangeability and the implicit role of prior distributions [23]. During World War II (the 1940s), I.J. Good, Alan Turing, and George Barnard worked on sequential data analysis methods, using the weight of evidence (log Bayes factors) [5, 38].

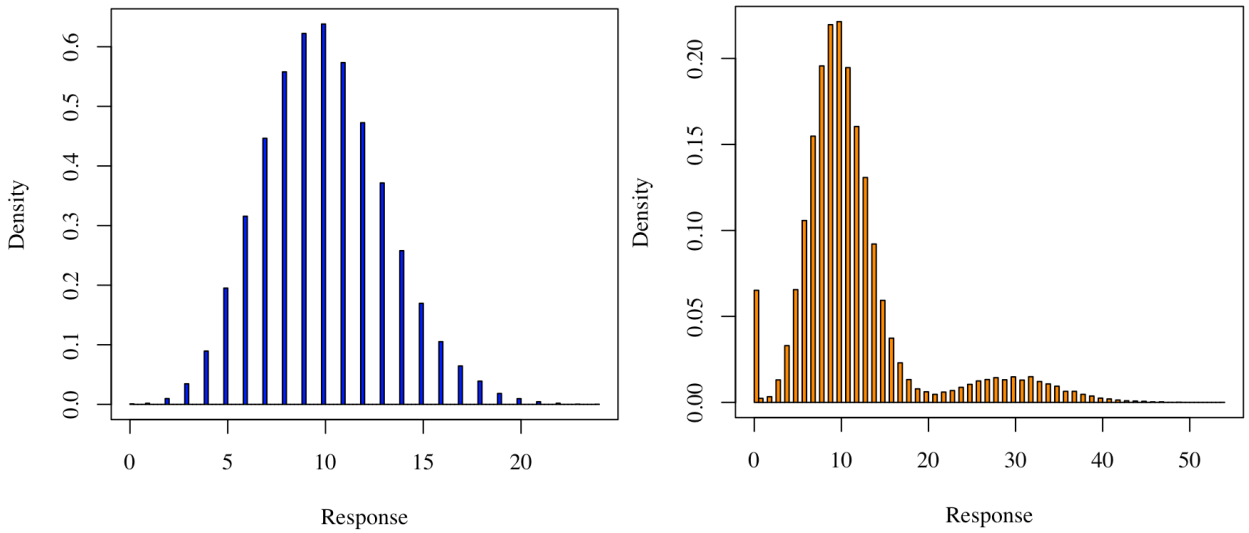
While statisticians constructed the foundational elements of Bayesian statistics throughout the 1900s, the Bayesian paradigm was not popularized until the 1990s, when computers became widely accessible to utilize Monte Carlo techniques to evaluate posterior distributions. As a result, most modern Bayesian statistical analyses involve a Monte Carlo (MC) technique or computational algorithm for performing inferences about our parameters of interest. Thus, the rise of Bayesian statistics coincides with the growth of Monte Carlo techniques. In this section, we briefly motivate Bayesian methods through a comparative analysis between confidence intervals and credible intervals, outline the construction of posterior distribution, and provide various examples to highlight the utility and flexibility of the Bayesian paradigm.

Finally, we dedicate Chapter 3 to the discussion of Monte Carlo theory and techniques.

2.1 Comparative Analysis between Classical and Bayesian Paradigms

Classical methods rely heavily on (1) the conceptualization of repeated realization of data given fixed parameters and (2) require large amounts of data to induce asymptotically normally distributed assumptions. For example, the Central Limit Theorem (CLT) is one of the famous theorems utilized in traditional statistics, which addresses the asymptotic properties of the sampling distribution of the sample mean regardless of the population probability distribution. The Central Limit Theorem states that if we have a population with mean μ and standard deviation σ and take sufficiently large sample sizes (n) from the population, then the distribution of the sample means will be approximately normally distributed. For instance, let us consider the two populations illustrated in Figure 2.1 where we generated Figure 2.1a data from a single Poisson model and generated Figure 2.1b from a mixture of Poisson, each with a population size of $N = 60,000$.

According to the Central Limit Theorem, if our sample size, n, is large, asymptotic frequentist results can be applied since the deviation between the theoretical and actual results closely agrees. We perform a simulation study to demonstrate the Central Limit Theorem holds by repeatedly collecting 100,000 samples of size 20,000 ($n = 20,000$) and calculating the associated sample mean for both populations. Figure 2.2 displays results from the simulation study using Quantile - Quantile (QQ) plots to visually illustrate whether the distribution of sample means came from a normal distribution. The black 45-degree line



(a) Population #1: Single Poisson.

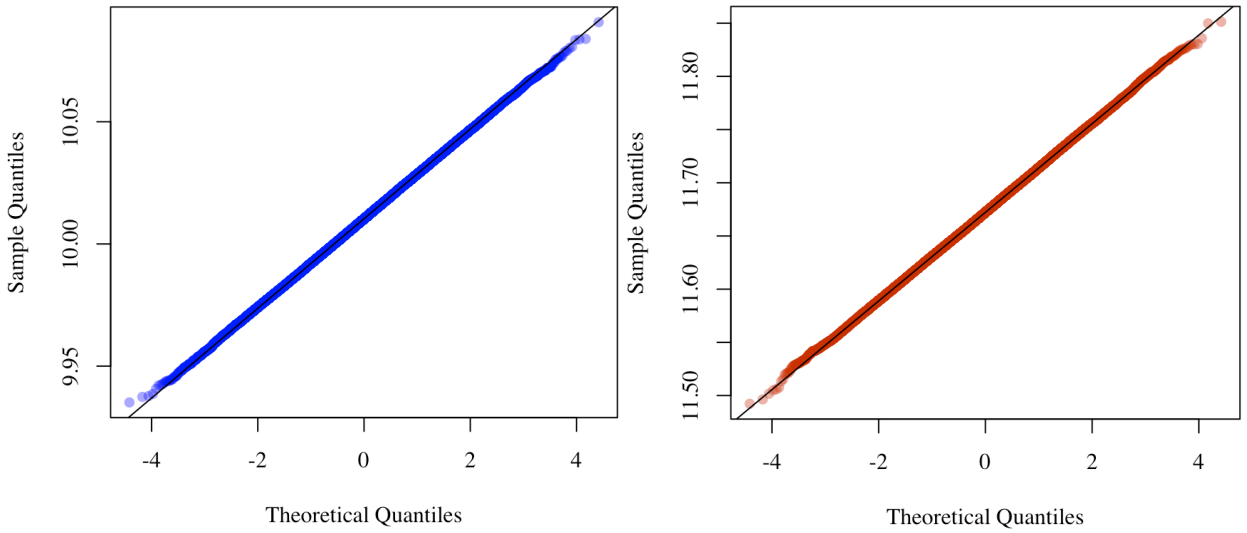
(b) Population #2: Mixture of Poisson.

Figure 2.1: Illustration of two populations with population size $N = 60,000$.

provides a visual reference to check the data's normality where if the points closely follow the line, we can assume the data came from a normal distribution.

Of course, the question for any analysis relying on asymptotic properties is "How large should the sample size be?". Commonly, in introductory statistics courses, professors teach students that the *rough* rule of thumb "large enough" is a sample size of thirty ($n = 30$). However, such generic rules do not always yield reliable probabilistic analysis. Figure 2.3 repeats our simulation study illustrated in Figure 2.2, but with a sample size of thirty-one ($n = 31$).

The key aspect to notice between the respective QQ plots in Figure 2.2 and 2.3 is the difference in tail behavior. For instance, Figure 2.3b, the deviation from the theoretical normal distribution quantiles (i.e., distance from the black line) is larger than in Figure 2.2b; thus demonstrating the Central Limit Theorem asymptotic property has yet to converge to normal distribution. We do not see as large of a difference in the QQ plot between Figure 2.2a



(a) QQ plot for population #2

(b) QQ plot for population #2

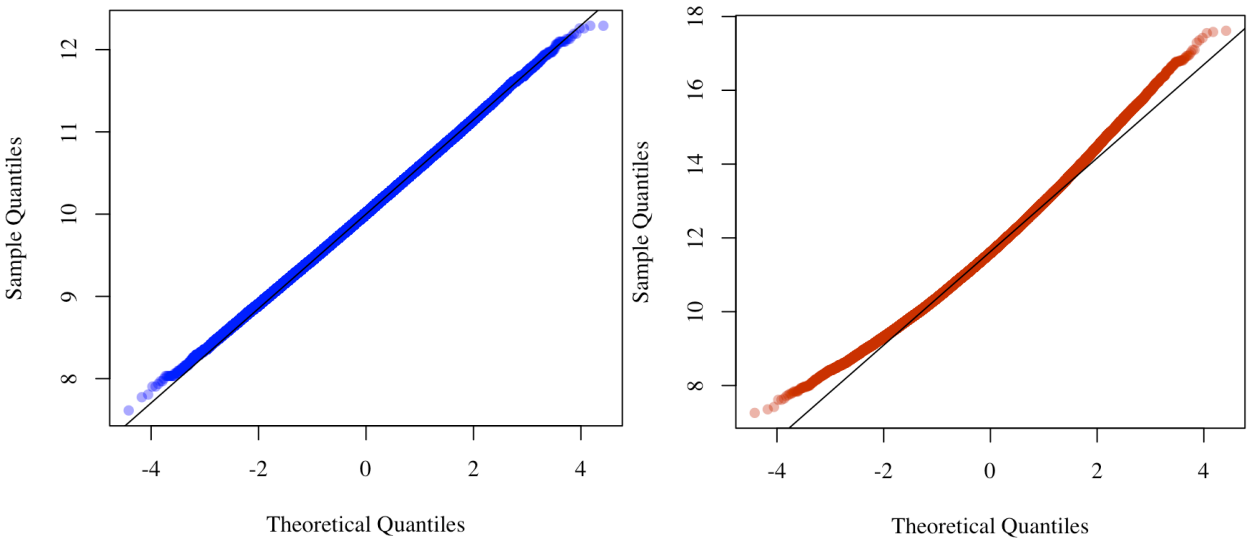
Figure 2.2: Demonstration of Central Limit Theorem when we collected 100,000 samples of size $n = 20,000$ to evaluate the distribution of the sample means for the populations in Figure 2.1.

and Figure 2.3a because population # 1 is already an approximately normal distribution based on theory. From an inferential standpoint (i.e., interval estimation), the lack of convergence will produce inaccurate results regarding the probability intervals. We provide a simulation study to demonstrate this result in Figure 2.5.

In the classical paradigm, the concept of repeated realization of data given fixed parameters is called a sampling distribution (or density) that we denote as:

$$\underline{x}_i \overset{\text{i.i.d}}{\sim} f\left(x|\theta\right); \tag{2.1}$$

for $i = 1, \ldots, N$ where $\underline{x}_i$ represent the i^{th} observation of the N total number of observation, i.i.d stands for independent and identical distributed, and θ represents the associated



(a) QQ plot for population #2

(b) QQ plot for population #2

Figure 2.3: Demonstration of Central Limit Theorem when we collected 100,000 samples of size $n = 31$ to evaluate the distribution of the sample means for the populations in Figure 2.1.

parameters with distribution $f(\bullet)$ where:

$$\int_{-\infty}^{\infty} f\left(x|\theta\right) dx = 1.$$

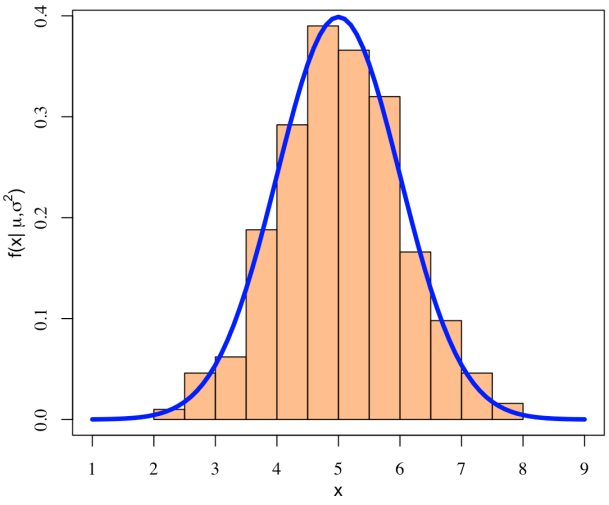
For instance, Figure 2.4a represents 1,000 repeated realizations from a univariate normal distribution with fixed parameters $\theta = \{\mu = 5, \sigma^2 = 1\}$ with sampling distribution:

$$f_{\text{Normal}}\left(x|\mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}},$$

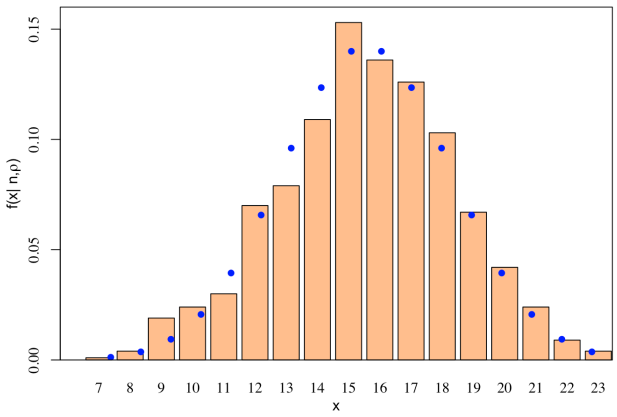
where μ and σ^2 are scalar values representing the centering and scale parameters, respectively. While Figure 2.4b demonstrates a binomial distribution $\theta = \{n = 31, \rho = 0.5\}$ with a sampling distribution:

$$f_{\text{Binomial}}\left(x|n,\rho\right) = \binom{n}{\rho}\rho^{x}(1-\rho)^{n-x}, \tag{2.2}$$

where n and ρ represent the sample size and probability of success, respectively. The blue curve/dots illustrates the true density values of the respective distribution.



(a) Normal sampling distribution



(b) Binomial sampling distribution

Figure 2.4: Illustration of sampling distribution of normal ($\mu = 5, \sigma^2 = 1$) and binomial distribution ($n = 31, \rho = 0.5$) for 1000 realizations. The blue curve/dots represents the true density of the respective distribution.

To further illustrate the limitation of relying on asymptotic assumption, we briefly consider the classic problem of interval estimation of a binomial proportion under the classicist and Bayesian perspectives. In this problem, we assume our data follows:

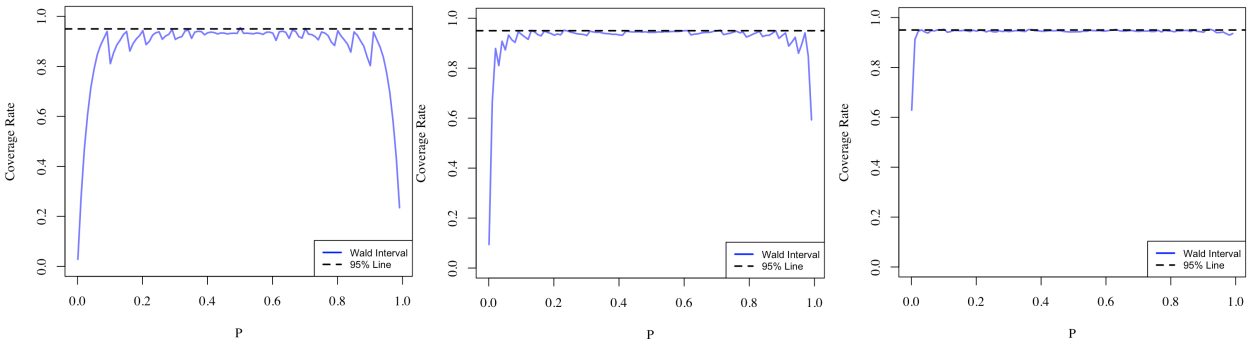
$$\underline{x_i} \sim \text{Binomial}\left(n,\rho\right),$$

with a sampling density, Eq. 2.2. The classical approach constructs the following Wald

confidence interval:

$$\hat{\rho} \pm z_{\alpha/2}\sqrt{\frac{(1-\hat{\rho}) \times \hat{\rho}}{n}},$$

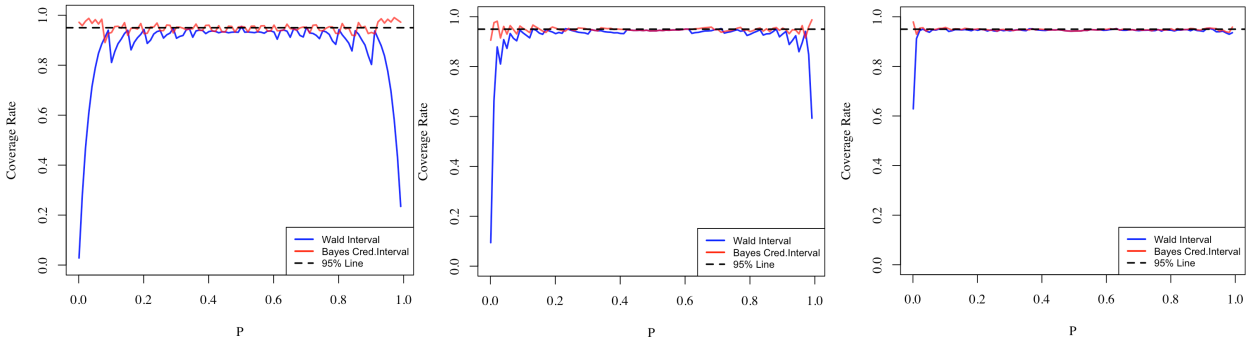
where $\hat{\rho} = \frac{x}{n}$ is the sample proportion of successes and $z_{\alpha/2}$ represents the $((1 - \frac{\alpha}{2}) \times 100)^{th}$ percentile of the standard normal distribution. Under this set-up, we would expect the confidence interval covers the true ρ value $((1 - \frac{\alpha}{2}) \times 100)^{th}\%$ of the time. Figure 2.5 illustrates the results of a simulation study that evaluates the coverage rates under the Wald Confidence Interval (blue lines) for a 95% confidence interval. We calculated the coverage rates by generating 10,000 binomial dataset with the respective ρ value and calculated the frequency of times the interval covered the true ρ value. We see that for small n values, our Wald coverage rates are not at the 95% threshold line (dotted line). Additionally, near the boundaries of ρ, the classical asymptotic assumptions break, even with larger n values. While we provide a quick simulation study, copious amounts of literature demonstrate the Wald Intervals poor behavior [14, 37, 98]. Adaptations to the Wald method, such as the "adjusted Wald" method [1], have been developed to account for situations when the parameter is on the boundary.



(a) n = 30 (b) n = 100 (c) n = 1000

Figure 2.5: Illustration of the coverage rates under the Wald Confidence Interval (blue lines) for a 95% confidence interval at various sample sizes (n ={ 30, 100, 1000}).

While implementing classical methods is simple, they require large amounts of data to induce asymptotically normally distributed assumptions as demonstrated in the confidence interval and CLT examples. In the Bayesian paradigm, we have more modeling flexibility through our choice in prior distribution and likelihood functions which aid in removing strict asymptotic assumptions (i.e., large n). Without all the details yet, in the binomial example, a Bayesian would perform a conditional analysis to estimate an credible interval on ρ by constructing a posterior distribution on ρ given the data. A credible interval is an interval such that the integral of the posterior distribution over the parameter is equal to a specified percentage, in this case 95%. In our simulation study, we calculate a 95% credible interval by chopping off 2.5% of the posterior distribution's tails. Figure 2.6 provides a comparative analysis of the coverage rates produced by the credible Bayesian interval (red line) and the Wald confidence interval (blue line). We see that the Bayesian coverage rates are closer to the 95% threshold compared to the Wald confidence interval. In section 2.4, we discuss in detail the construction of the posterior distribution and credible intervals.



(a) n = 30 (b) n = 100 (c) n = 1000

Figure 2.6: Illustration of the coverage rates under the Bayesian Credible Interval (red line) and Wald Confidence Interval (blue lines) for a 95% confidence interval at various sample sizes (n ={ 30, 100, 1000}).

Bayesian statistics focuses on making inferences (or conclusions) using probability statements about our unknown parameters (θ) conditioned on the observed data, ($\mathbf{X}$) through

constructing a joint, posterior distribution, denoted as $f(\theta|\mathbf{X})$. The posterior distribution represents the joint relationship (or probability distribution) between the parameters and the data. Through utilizing Bayes' Rules, we develop a posterior distribution by:

$$f(\theta|\mathbf{X}) = \frac{\mathcal{L}(\theta|\mathbf{X}) \times p(\theta)}{m(\mathbf{X})},$$

where $\mathcal{L}(\theta|\mathbf{X})$ represents the likelihood function, $p(\theta)$ represents the prior distributions for θ, and $m(\mathbf{X})$ represents the marginal distribution of $\mathbf{X}$ where:

$$m(\mathbf{X}) = \int_{\theta} \mathcal{L}(\theta|\mathbf{X}) \times f(\theta) d\theta.$$

We dedicate Section 2.2 and 2.3 to discuss likelihood functions and choices of prior distributions. The pivotal difference between the classical and Bayesian perspectives is the concept of conditioning on the data (i.e., treating the data as fixed) and developing a probability distribution around our unknown parameters. By constructing a probability distribution around our parameters, Bayesian's interpret probability as a measure of the relative plausibility of an event rather than as the long-run relative frequency of a repeatable event (frequentist/classicist perspective).

2.2 Likelihood Functions

The likelihood function enables us to evaluate the relative compatibility of data with the parameter(s) of interest. We derive the likelihood function by:

$$\mathcal{L}(\theta|\mathbf{X}) = \prod_{i=1}^{N} f(x_i|\theta), \tag{2.3}$$

where $\prod_{i=1}^{N} f(x_i|\theta)$ represents the joint distribution of $\mathbf{X}$ given θ. For instance, in the independent and identical distributed (i.i.d) case:

$$f(\mathbf{X}|\theta) = \prod_{i=1}^{N} f(x_i|\theta),$$

and necessarily:

$$\int_{-\infty}^{\infty} f(\mathbf{X}|\theta) d\mathbf{X} = 1.$$

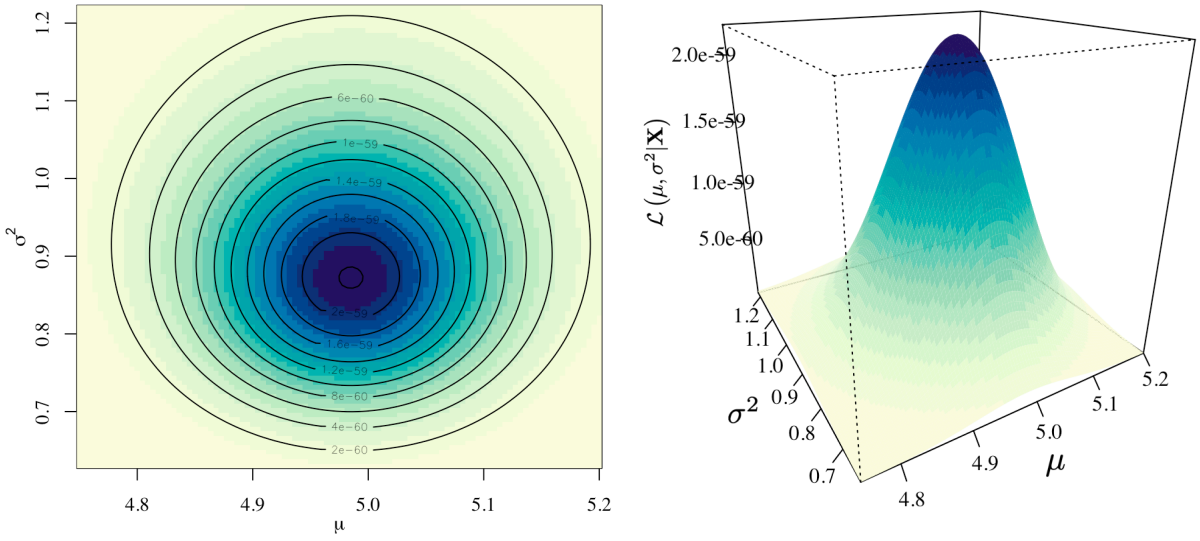
A likelihood function utilizes the same mathematical parametric relationship between $\mathbf{X} = \{x_1, \ldots, x_N\}$ and θ, but is a function over θ while $\mathbf{X}$ is fixed. We denote this by writing $\mathcal{L}(\theta|\mathbf{X}) \propto f(\mathbf{X}|\theta)$ which expresses the same relationship between $\mathbf{X}$ and θ, but the argument of the function has changed. While the sampling distribution describes probability distribution of the data, the likelihood function is not probability distribution, but instead describes the feasibility of jointly seeing specific value of our parameters given the observed data.

For instance, consider univariate normal sampling distributions, Eq. 2.1 from Figure 2.4, our likelihood for μ and σ^2 is:

$$\mathcal{L}\left(\mu, \sigma^2|\mathbf{X}\right) = \prod_{i=1}^{N} f(x|\mu, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(x-\mu)^2}{2\sigma^2}}. \tag{2.4}$$

Figure 2.7 illustrates the likelihood function using a 2D contour plot and 3D surface plot to

help demonstrate the difference between the sampling distribution of the data, Figure 2.4a, which is one dimensional of the data, $\mathbf{X}$, and the likelihood function of the parameters, μ and σ^2, which is two-dimensional plot. Note that if we generated a different set of data from the same sampling distribution, the likelihood contours also would change due to the new data.



(a) 2D contour plot

(b) 3D surface plot

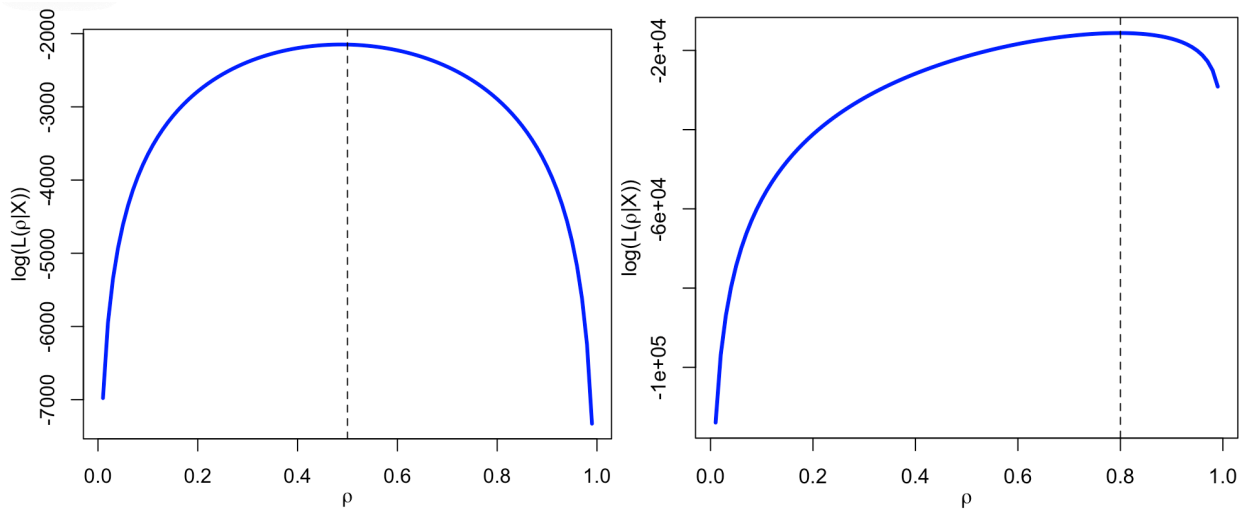
Figure 2.7: Illustration of normal likelihood function given 100 observation for data in Figure 2.4a

For the binomial sampling distribution, Eq. 2.2, our likelihood of ρ is:

$$\mathcal{L}(\rho|\mathbf{X}) = \prod_{i=1}^{N} f(x|\rho) = \prod_{i=1}^{N} \binom{n}{x_i} \rho^{x_i}(1-\rho)^{n-x_i}, \tag{2.5}$$

where n represents the number of trials in the experiment and N represent the number of times the experiments was ran. Figure 2.8 illustrates two cases of the log-likelihood function, log of Eq. 2.5, for a binomial distribution to further demonstrate the difference between

likelihood functions and sampling distribution. Notice the binomial sampling distribution, Figure 2.4b, ranges from zero to thirty-one in a discrete space, while the log-likelihood function of ρ, Figure 2.8, ranges from zero to one in a continuous space. Specifically, Figure 2.8a illustrates the log-likelihood of the data represented in Figure 2.4b where $\rho = 0.5$. Figure 2.8b demonstrates a log-likelihood where we generated data with $\rho = 0.8$. Often times, we use the log of the likelihood function for numerical and computational reasons.



(a) $\rho = 0.5$ (b) $\rho = 0.8$

Figure 2.8: Illustration of the log-Likelihood for the Binomial distribution for data in Figure 2.4b when $\rho = 0.5$ and for data generated with $\rho = 0.8$

While the likelihood function gives us the ability to understand parameters jointly, the prior distribution on the parameters, $p(\theta)$, allows us to apply Bayes Theorem and "elevates" our inference about θ onto a probability scale. Thus, while $\mathcal{L}(\theta|\mathbf{X})$ is not a probability distribution, the posterior distribution $f(\theta|\mathbf{X})$ is a probability distribution which provides a standard scale for providing inferences about θ. That is:

$$\int_{\theta \in \Omega} f(\theta|\mathbf{X}) d\theta = 1, \tag{2.6}$$

where $\theta \in \Omega$ represents values of θ that "best" explain the data, $\mathbf{X}$. The prior distribution's "job" is to help encode information, regularize the space, and produce "good" convergence properties. There are a multitude of priors to choose from based on your research problem of interest such as conjugate, reference, Jeffreys, proper and improper, or point-mass prior.

2.3 Prior Distributions

The prior distribution encodes our belief about our unknown parameters before seeing the data (i.e., *a-priori*). An advantage of encoding our beliefs is that if we have expert knowledge about our process (or experiment), we can utilize the knowledge to improve our inferences. Conversely, if we lack knowledge about or do not want to bias our results, we can use *non-informative* priors to invoke a 'neutral' inferential analysis. In Figure 2.6, we constructed the Bayesian credible interval using a Beta prior distribution with the sampling density:

$$f_{Beta}\left(x|\alpha,\beta\right) = \frac{\Gamma(\alpha_\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \tag{2.7}$$

where $\Gamma(\bullet)$ denotes the Gamma function:

$$\Gamma(c) = (c-1)!, \tag{2.8}$$

where c is a positive integer. We utilize with hyperparameters $\alpha = 1/2$ and $\beta = 1/2$ to

provide optimal coverage rates as demonstrated by [15]. However, we could have picked alternative settings for the hyperparameters subjected to researcher knowledge, such as $\alpha = 4$ and $\beta = 4$. Figure 2.9 demonstrates the difference in the weighting scheme placed on the parameter space between Beta(1/2,1/2) and Beta(4,4) before seeing the data. Our Beta(4,4) prior places more weight at $\rho = 0.5$ compare to the remaining parameter space to suggest that *a-priori* our researcher believes ρ exists near 0.5. The Beta(1/2, 1/2) prior acts as a Bayesian analogous to the Agresti-Coull confidence interval. The Agresti-Coull confidence interval takes the dataset and adds an equal amount of ones (1) and zeros (0), then computes the interval [1]. The intention behind the Agresti method is to penalize the endpoints in situations when we have a small sample size.

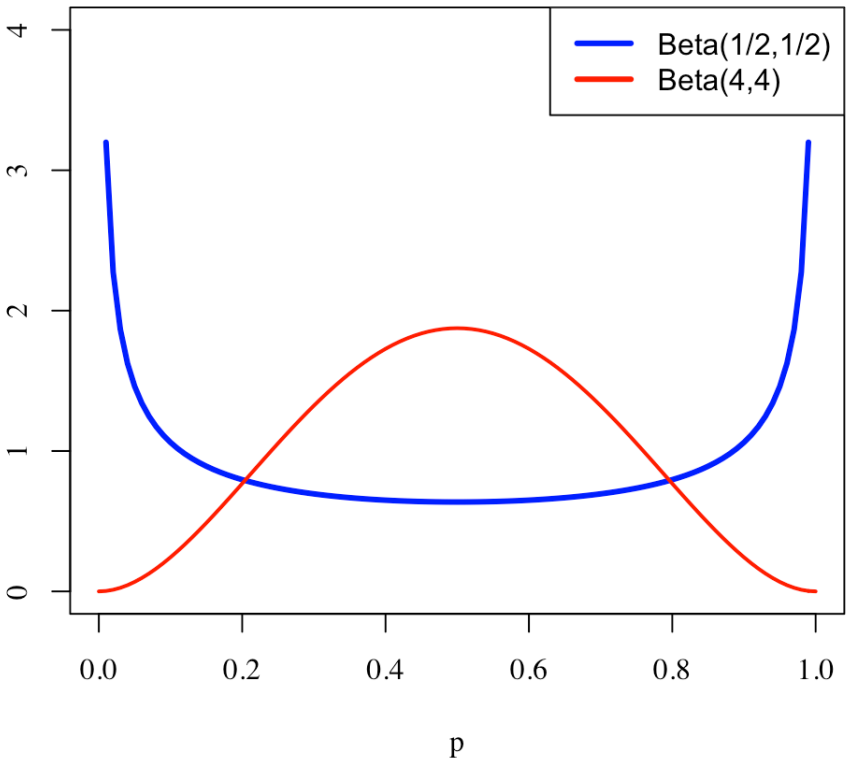
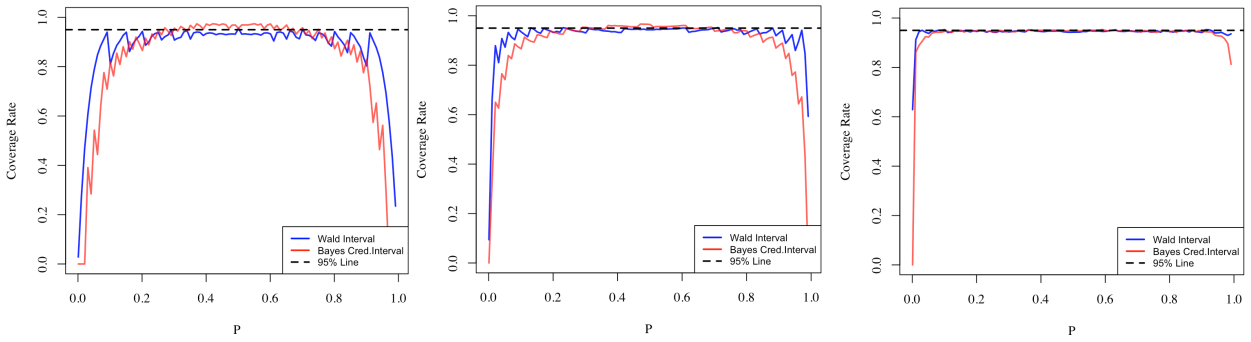


Figure 2.9: Illustration of difference between Beta($\frac{1}{2}$,$\frac{1}{2}$) (blue) and Beta(4,4) (red) prior distribution.

Figure 2.10 demonstrates the Beta(4,4) coverage rates relative to the Wald interval at our three varying sizes. Comparing the two Bayesian and Frequentist (Wald) analyses enables us to highlight the advantages and disadvantages of using various priors. In Figure 2.10a, the Beta(4,4) coverage rates for ρ between 0.35 and 0.65 are higher than the 95% threshold. Under this situation, our expert's *a-priori* knowledge helps improve our inference if the true ρ exists in the interval. However, if ρ lives closer to the bounds, the Wald and Beta(1/2, 1/2) intervals prior produce better coverage rates. When choosing a prior distribution and associated hyperparameters, we want to think about and assess our decisions because, in small sample sizes, the prior impacts the posterior distribution. Fortunately, as the sample size increase, the likelihood function becomes the driving force in the posterior analysis, thus decreasing the impact of the prior distributions. Figure 2.10c demonstrates this result aside from $\rho = 0.001$ due to some numerical instability.



(a) n = 30 (b) n = 100 (c) n = 1000

Figure 2.10: Illustration of the coverage rates under the Bayesian Credible Interval (red line) with prior distribution of Beta(4,4) and Wald Confidence Interval (blue lines) for a 95% confidence interval at various sample sizes (n ={ 30, 100, 1000}).

Analytically, the choice of prior distribution should allow our analysis to stabilize inferences with low sample sizes and regularize the space. Regularizing the space applies a penalization to our parameters space to improve our results. In classical statistics, standard penalization

techniques for model selection are Ridge Regression, Least Absolute Shrinkage and Selection Operator (LASSO), and Elastic-Net. The Bayesian analogy to these techniques requires using different priors such as Gaussian distribution, double-exponential, and a mixture between Gaussian and double-exponential prior. Additionally, researchers often choose priors because they induce "good" properties of the posterior and convergence properties. We use "good" to denote that what some researchers deem as good and valuable can be severely different from other researchers.

In this section, we discuss several prior distributions to enable the reader to grasp the different capabilities of prior distributions. Before discussing specific priors, we required the prior distribution to (1) account for all possible events, (2) assign prior probabilities to every event, and (3) be valid density, regardless of the prior chosen. For instance, if we consider a univariate normal distribution with mean, μ, and variance, σ^2 where we need a prior distribution on σ^2. We know the support of σ^2 is $[0, \infty)$; thus some reasonable prior distributions options are inverse-Gamma with sampling density of:

$$f_{\text{inverse-Gamma}}(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} e^{-\frac{\beta}{x}},$$

where α and β are the shape and scale parameter, respectively, or truncated normal on the interval $[0, \infty)$ with sampling density:

$$f_{\text{truncated normal}}(x|\mu, \sigma) = \frac{1}{\sigma \left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right]} \times \left[\frac{1}{\sqrt{2\pi}} e^{-\frac{\left(\frac{x-\mu}{\sigma}\right)^2}{2}}\right],$$

where Φ represents the cumulative density function for a normal distribution and $a = 0$ and $b = \infty$ representing the lower and upper bounds.

Jeffreys and Reference Priors

"Non-informative" priors attempt to pass on little to no information about the parameter of interest into the posterior analysis. The two commonly discussed "non-informative" priors are Jeffreys' priors and reference priors.

Sir Harold Jeffreys performed inference on parameters as a physicist, but he wanted a property that reduced arbitrary choices due to the selection of parameter space. As a result, Jeffreys' priors induce a property called transformation invariant [50]. Transformation invariant enables researchers to utilize different parameterizations of a variable and develop identical inferences. For instance, a typical example pertains to a statistician's choice in either using variance σ^2 or the inverse of variance, known as the precision ($\phi = \frac{1}{\sigma^2}$), when estimating the spread of a normal distribution. If both statisticians utilized the Jefferys prior distribution for their respective parameterization, they would make identical inferences.

We derived Jeffreys' priors using:

$$p_J(\theta) \propto I^{\frac{1}{2}}(\theta),$$

where $I(\theta)$ represents Fisher information and $p_J(\theta)$ denotes the Jeffreys' prior for the parameter θ. In general, the Fishers Information is:

$$I(\theta) = -\mathbb{E}_x \left[\frac{d^2 \log(\mathcal{L}(\theta|\mathbf{X}))}{d\theta^2} |\theta \right], \tag{2.9}$$

or

$$I(\theta) = \mathbb{E}_x \left[\left(\frac{d \log(\mathcal{L}(\theta|\mathbf{X}))}{d\theta} \right)^2 |\theta \right]. \tag{2.10}$$

Eq. 2.9 and Eq. 2.10 are equivalent if Fubini's theorem holds. Fubini's theorem gives us permission to swith the order of the integration when the double integral results in a finite answer. Jeffreys' priors are also known as reference priors in the one-dimensional case as demonstrated by Bernardo and Smith [11]; however, this is not the case in higher dimensions although we can utilize the marginally Jefferys priors. For more details, reference [8].

Reference priors are derived by maximuixing a measure of distance or divergence between the posterior and prior distribution [10]. Often, we utilized the Kullback - Lieber (KL) divergence which is:

$$K.L. = \int f\left(\theta|T(\mathbf{X})\right) \times \log \left(\frac{f\left(\theta|T(\mathbf{X})\right)}{p(\theta)} \right) d\theta, \tag{2.11}$$

where $T(\mathbf{X})$ represents a sufficient statistic, $p(\theta)$ represents the prior distribution, and $f\left(\theta|T(\mathbf{X})\right)$ denotes the posterior distribution.

By identifying the prior distribution with the largest distance from the posterior, it enables the likelihood function (i.e., the data) to impact the posterior distribution more than the

prior. To pick the reference prior, we take the expected value of the KL divergence with respect to the model distribution of the data (i.e., the sufficient statistic) as:

$$\mathbb{E}[K.L.] = \mathbb{E}_{T(\mathbf{X})}\left[\int f\left(\theta|T(\mathbf{X})\right) \times \log\left(\frac{f\left(\theta|T(\mathbf{X})\right)}{f(\theta)}\right) d\theta\right]. \tag{2.12}$$

Conjugate Priors

When the parametric form of the prior and posterior distributions are from the same family, the prior distribution is called conjugate. We typically invoke conjugate priors because they induce a closed-form analytic solution, computational reasons, ease interpretability, and they often have good approximations. In our Bayesian credible interval example, our posterior distribution is:

$$f(\rho|\mathbf{X}, \alpha, \beta) \propto \rho^{\alpha-1+\sum_{i=1}^{N} x_i}(1-\rho)^{\beta-1+\sum_{i=1}^{N} n-x_i}, \tag{2.13}$$

which we constructed by multiplying the likelihood function, Eq. 2.5, by the Beta conjugate prior :

$$p(\rho) \propto \rho^{\alpha-1}(1-\rho)^{\beta-1}. \tag{2.14}$$

Both posterior and prior are Beta distributions with different parameterization; thus, the conjugate prior for ρ is a Beta distribution. Within our proposed methodologies, a Robust Bayesian Regression and a Modified Cauchy Net, we utilize several conjugate priors such as the multivariate normal prior distribution for the mean, $\underline{\mu}$, denoted as:

$$p_{\text{MV. Normal}}(\underline{\mu}|\underline{m}, \mathbf{V}_{P\times P}) = |2\pi\mathbf{V}|^{-1/2} e^{\frac{-1}{2}(\mu-m)^T\mathbf{V}^{-1}(\mu-m)}, \tag{2.15}$$

where m is a $P \times 1$ location vector parameter and $\mathbf{V}$ is a $P \times P$ positive definite covariance matrix. For the covariance structure Σ, we use an inverse-Wishart distribution:

$$p_{\text{inverse-Wishart}}(\Sigma_{P\times P}|\Omega, \psi) = \frac{|\Omega|^{\frac{\psi}{2}}}{2^{\frac{\psi P}{2}}\Gamma_P(\frac{\psi}{2})}|\Sigma|^{-\frac{\psi+P+1}{2}} e^{-\frac{1}{2}\text{tr}(\Omega\Sigma^{-1})},$$

where $\psi > P - 1$ represents the degrees of freedom and Ω is a $P \times P$ positive definite scale matrix.

Outside of reducing computational burden, conjugate priors have the flexibility to induce subjective or objective *a-prior* beliefs about our model. For instance, we demonstrated a "subjective" prior, Beta(4,4), in the credible binomial example, where we could have also used Beta(1,1) prior, which results in uniform distribution from [0,1]. The uniform distribution would impose equal weighting across the parameter space and incorporates no additional knowledge about our process.

In a Bayesian analysis, we choose our hyperparameters for various reasons, including but not limited to creating "tunable" coverage rates, inducing frequentist properties, or imposing various degrees of prior belief about the data. Researchers can pick $\alpha = \beta$ to invoke unbiasedness in the binomial interval estimation problem. However, we adjusted the values ($\alpha = \beta = 4$) to fit our knowledge when we assumed prior information about ρ. For our coverage rate simulation in Figure 2.6, we choose $\alpha = \beta = \frac{1}{2}$ to induce optimal coverage

rates as illustrated by Brown [15].

2.4 Bayesian Analysis Examples

We used the binomial interval estimation to help motivate and detail the foundational elements of Bayesian statistics. Unfortunately, the example lacks the properties of real-world applications because it only considers a single parameter estimation. Frequently, in real-world applications, the models that describe our data have more complex and intricate connections a single distribution cannot explain. Thus, we need hierarchical models to help define the relationship between the various parameters. While there are several different forms of hierarchical models, we focus on scaled-mixture and additive mixture models to foreshadow concepts in this thesis.

2.4.1 Scaled - Mixture Models

The scaled mixture models provide a flexible technique to model heavy-tailed data (i.e., non-normal data), heteroscedasticity (i.e., non-constant variance), or autocorrelation data by using the hierarchal model framework. The first implementation of the scaled mixture of normal focused on sampling symmetric distributions that have a normal component [4]. We use scaled mixtures to invoke a Student t-distribution, specific Cauchy, to induce a robust methodology for detecting anomalous observations. Using the hierarchical model,

$$\begin{aligned} x_i &\sim \text{Normal}\left(\mu, \gamma_i^{-1}\sigma^2\right), \\ \gamma_i &\sim \text{Gamma}\left(\alpha, \beta\right), \end{aligned}$$

we produce a Student t-distribution when we integrate over the joint distribution $f(x_i, \gamma_i)$ to γ_i where the Gamma probability density is:

$$p_{Gamma}\left(\gamma_i|\alpha, \beta\right) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}}\gamma_i^{\alpha-1}e^{-\frac{\gamma_i}{\beta}},$$

where α and β represent the shape and scale hyperparameters, respectively.

While this seems like a long-winded approach to get a Cauchy distribution, we utilize the scaled-mixture model to avoid computational issues. For instance, the Cauchy sampling density is:

$$f_{Cauchy}(x|\mu, \sigma) = (\pi\sigma)^{-1}\left[1 + \left(\frac{x-\mu}{\sigma}\right)\right]^{-1},$$

where μ and σ represent the scalar location and scale parameters; thus, the resulting Cauchy likelihood function is:

$$\mathcal{L}\left(\mu, \sigma|\mathbf{X}\right) = \prod_{i=1}^{N}(\pi\sigma)^{-1}\left[1 + \left(\frac{x_i-\mu}{\sigma}\right)\right]^{-1}.$$

Unfortunately, as the sample size (N) increases, the likelihood function becomes a large product that may results in a numerical unstable outcome. Whereas, the likelihood of a normal distribution, Eq. 2.4, collapse into the summation term in the exponent term such as:

$$\begin{aligned}\mathcal{L}\left(\mu,\sigma^2\right) &= \prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{(x-\mu)^2}{2\sigma^2}} \\ &= \left(2\pi\sigma^2\right)^{-N/2}e^{\sum_{i=1}^{N}\frac{(x-\mu)^2}{2\sigma^2}},\end{aligned}$$

producing a computational easier problem. The hierarchical nature of the scale-mixture model leads to a simple implementation of Markov chain Monte Carlo known as Gibbs Sampling. We prove details on Gibbs sampler in Chapter 3 and go into depth about how we utilize scaled-mixture model to detect anomalies and perform inferences in Chapter 4.

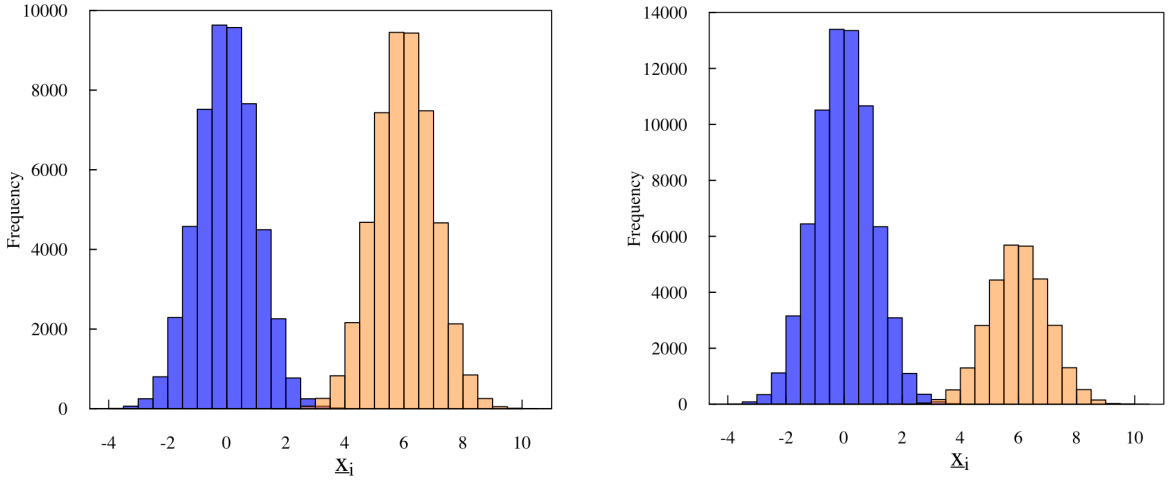
2.4.2 Additive Mixture Models

While the scaled - mixture model provide an extra component of flexibility and complexity to our model to aid in modeling more real-world application situation, the scaled - mixture models are limited to characterizing a single behavior of the process. In some experiments, it may be beneficial to compartmentalize several different types of signal or noise component, thus, additive mixture models enable us to break our space to better represent our process. A general k-finite mixture model assumes each observation comes from one of K component distributions. That is, an observation x_i:

$$\underline{x}_i \sim \sum_{k=1}^{K} \pi_k f(x|\theta_k), \tag{2.16}$$

for $i = 1, \ldots, N$ and $k = 1, \ldots, K$, where K represent the total number of components and $f(x|\theta_k)$ is some distribution with parameters θ_k for the k^{th} component. The π_k represents the mixing weight (i.e., proportion we sample from the kthcomponent) under the constraint $\sum_{k=1}^{K} \pi_k = 1$ [65].

A common variation of the generalized k-finite mixture model is the Gaussian mixture model in which all the k distributions, $f(x|\theta_k)$, are Gaussian distribution with different parameters, μ and σ^2,[79]. The parameters, θ_k, plays an important role distinguish between the different groups, thus the distance and overlay of the components will heavily influence the estimation procedure (discussed later on). Figure 2.11 illustrates a 1d, 2-component Gaussian mixture model with different location parameters, μ, and same variance to demonstrate the effect of the mixing weight. Figure 2.11a displays when the mixing weights are equal across each component (i.e., $\pi_1 = \pi_2 = 0.5$); whereas Figure 2.11b illustrate unequal mixing weight (i.e., $\pi_1 = 0.7, \pi_2 = 0.3$).



(a) Equally weighted components $\pi_1 = \pi_2 = .5$

(b) Unequally weighted components $\pi_1 = 0.7$ and $\pi_2 = .3$

Figure 2.11: Illustration of mixture weight, π_k ,in a 1d, 2-component Gaussian mixture model.

We can hierarchically represent a k-finite mixture model by:

$$\begin{aligned} x_i &\sim \text{Normal}(\mu_k, \sigma_k^2), \\ c_i = \{1, \ldots, K\} &\sim \text{Multinomial}(\pi_1, \ldots, \pi_K), \end{aligned}$$

where c_i represents the i^{th} observation comes from the k^{th} group. We utilize conjugate prior distributions:

$$\begin{aligned} \mu_k &\sim \text{Normal}(\mu_o, \sigma_o), \\ \sigma_k^2 &\sim \text{Inverse Gamma}(\alpha_o, \beta_o). \end{aligned}$$

Once we construct our hierarchical model, the next application of interest is performing

inferences.

k-Finite Mixture Model Inference

There are a multitude of inferential approaches for estimating the mixture model parameters, θ_k, and π_k, which vary from distance-based techniques, hierarchical clustering, and model-based approaches. Distance-based techniques, such as K-means [63] or K-medians, minimize a distance-based objective function between the k-centroids and each observation. K-means utilizes a mean centroid estimator, whereas K-medians implement a median estimator. Due to the simplicity of their algorithms, K-means and K-medians algorithms are common clustering techniques. However, there are some limitations, such as distance-based algorithms apply a hard clustering assignment. That is, each observation is assigned directly to one and only one component. Additionally, depending on initialization, we are not guaranteed to find a global solution. An approach around identifying a global solution is to have multiple initializations and compare the assignments based on a specific metric.

Hierarchical Clustering procedures are algorithms that merge or split groups based on a specified "similarity" or "dissimilarity" metric such as single linkage, complete linkage, or Ward's distance. The most known hierarchical clustering strategies are agglomerative and divisive. Agglomerative methods start by assigning each of the N observations to individual clusters and merges the clusters until there is a singular N-observation cluster. Divisive techniques employ an opposite scheme to agglomerative algorithms. These techniques start with a single cluster and repetitively split the data into a series of smaller ones until the algorithm produces a N-individual clusters. However, divisive strategies are computationally expensive because there are $2^{N-1} - 1$ possible partitions. Thus, divisive strategies are less commonly used.

Model-based approaches, such as the Expectation-Maximization Algorithm [26, 72] and Gibbs sampler [17, 93], employ a more structured approach than the largely heuristic K-means and hierarchical clustering strategies. Model-based techniques assume the observations are independent and identically distributed (i.i.d) realizations from a probability model. Specifically, model-based approaches assume the probability model is a k-finite mixture model expressed in Eq. 2.16. Under a k-finite mixture model, we need to infer the latent group assignments (c_i) and the respective parameters, θ_k, for the k-groups. While the EM algorithm and the Gibbs sampler take two different inferential approaches to estimate the component labels and parameters, the algorithms break down into the following two main steps:

1. Calculating the probability of each observation belong to the k-clusters
2. Updates the k-distributional parameters according to the new group assignments

The Expectation-Maximization (EM) algorithm is an iterative process that effectively replaces maximizes a difficult likelihood, where some variables are (or treated as) latent variables, to maximize a sequence of easier likelihoods. The EM algorithm's first step takes the expectation of the log-likelihood with respect to the latent variables (c_i), that is:

$$\mathbb{Q}(\underline{\theta}|\underline{\theta}^{(t)}) = \mathbb{E}_{\underline{c}|\underline{\theta}^{(t)},\underline{x}}[\log \mathcal{L}(\underline{\theta}, \underline{c}|\mathbf{X})]. \tag{2.17}$$

This step boils down to calculating the probability of $\underline{x}_i$ in each k-component given the data and the current parameters models. The probability the i^{th} observation is in the k-component is:

$$f(c_i = k|\theta_k) = \frac{f(\underline{x}_i|c_{ik} = k, \theta_k) \times f(c_{ik} = k)}{\sum_{k=1}^{K} f(\underline{x}_i|c_{ik} = k, \theta_k) \times f(c_{ik} = k)}, \tag{2.18}$$

where $f(\underline{x}_i|c_{ik} = k, \theta_k)$ represent evaluating the density of $\underline{x}$ given the k component. The $f(c_i = k)$ represents the probability of existing in the component k, which can be 1/K or a Dirichlet prior distribution. The second step of the EM algorithm, which maximizes Eq. 2.17 with respect to the parameters, utilizes these probability values to invoke a weighted averaging scheme.

A Bayesian approach for estimating the component labels and parameters utilizes Markov chain Monte Carlo techniques known as Metropolis-Hasting algorithm. We elaborate in detail about Monte Carlo and Markov chain Monte Carlo theory and techniques in the Chapter 3.

Chapter 3

Monte Carlo History and Techniques

Monte Carlo (MC) methods are computational, algorithmic approaches that rely on repeated random sampling to obtain a numerical answer in a finite amount of time. Historically Monte Carlo approaches date back to 1945 during World War II with the work of John von Neumann, Stanislaw Ulam, and Nicholas Metropolis. Metropolis, Ulam, and von Neumann modeled the chain reaction of highly enriched uranium in the Manhattan Project at Los Alamos National Laboratory [67, 69]. Specifically, from a statistical perspective, the spark of Monte Carlo methods stemmed from Stanislaw Ulam becoming aware of the speed and versatility of the first electronic computer (ENIAC) and recognizing the tedious calculations for statistical sampling techniques could be solved using ENIAC [66]. Von Neumann and Ulam outline the implementations of a statistical approach to solving the problem of neutron diffusion utilizing Monte Carlo techniques [69, 96]. In 1948, the von Neumanns' (John and Klara) and Metropolis ran the first computerized Monte Carlo simulation series of calculations on ENIAC focused on problems such as the initial distribution of neutrons and configurations of material [41, 96].

While the modernization of computers quickly advanced Monte Carlo techniques, mathematicians and statisticians used Monte Carlo approaches before the invention of computers to validate mathematical solutions and discover revolutionary statistical concepts. For instance, in 1777, George Buffon proposed the infamous Buffon's needle drop problem [22, 48],

which investigated,

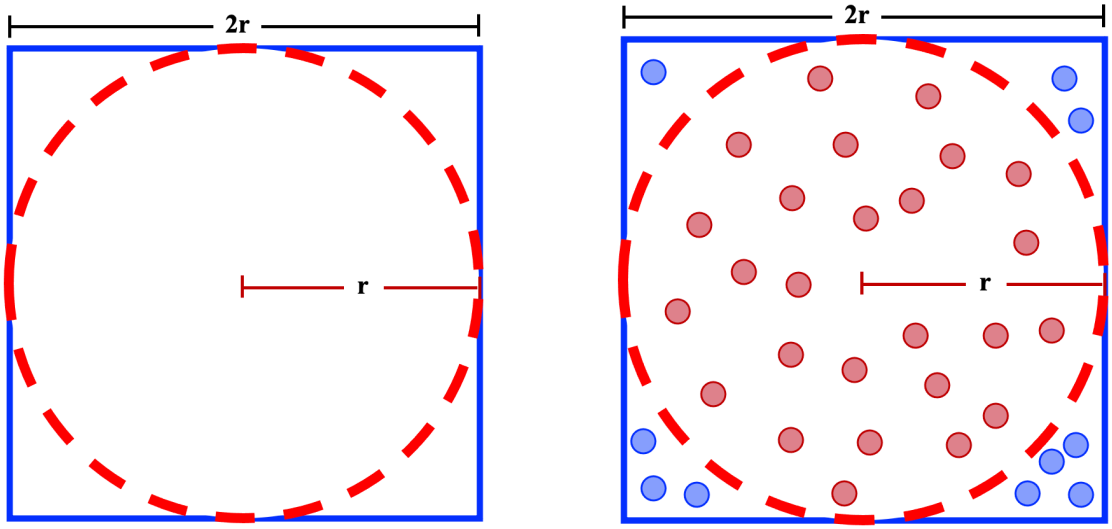
> "If we drop a short needle on ruled paper, what is the probability that the needle lies in a position where it crossed one of the lines?"

Buffon performed a Monte Carlo experiment where he repeatedly dropped a needle on lined paper and tracked whether the needle landed on a line or not to answer this question [22, 48]. In the 1770s, a limitation of Buffon's approach is the ability to repeat the experiment for a large number of trials to improve the estimation. In 1901, Mario Lazzarini extended Buffon's experiment by building a machine to increase the number of drops [57]. These experiments approximated the ratio of any circle's circumference to its diameter, i.e., π.

In today's society (the 2020s), Buffon and Lazzarini could replicate the Needle problem by utilizing a computer to improve their estimation accuracy and reduce their work time. A modern approach to Buffon's problem is drawing a circle with a radius, r, inside a square with a width of $2r$, as seen in Figure 3.1a, and uniformly sampling a large number of random points bounded by the square, illustrated by Figure 3.1b. We calculated our approximation by dividing the number of sampled points (red dots) within the circle by the total number of sampled points (blue + red dots), resulting in an approximation of:

$$\frac{\text{area of circle}}{\text{area of square}} = \frac{\pi r^2}{(2r) \times (2r)} = \frac{\pi}{4}. \tag{3.1}$$

Additionally, Erastus Lyman de Forest (the 1870s) and William Gosset (the 1900s) demonstrated the earliest (non-computer) implementation of Monte Carlo techniques to study smoothing times series and discover the distribution of the t-statistic [90] and the correlation coefficient [89], respectively. Both researchers utilized sampling experiments to discover



(a) Illustration of circle with radius r inside a square.

(b) Generating and tracking random points within the square and circle.

Figure 3.1: Computer simulation approach to Buffon's needle problem to approximate $\pi/4$ where w generate random points within the square (area $= 2r \times 2r$) and tracking frequency of landing in circle with radius r.

their statistical insight. For instance, Stephen Stiger describes De Forest's Monte Carlo simulation to study smoothing times series by simulating data using cards drawn from a box [87]. The experiments conducted by Buffon, de Forest, and Gosset demonstrate the basis of a Monte Carlo simulation which focuses on constantly repeating an experiment to understand the underlying properties and perform estimation calculations.

Repeated *random* sampling is the foundation of all Monte Carlo techniques. Thus, a "good" Monte Carlo algorithm decreases the random-ness around the true answer as we increase our fixed amount of time. By *random*, we refer to the statistical concept of a random variable and not producing any random, arbitrary numbers. From a technical standpoint, a random variable (denoted as R) is a function that associates a real number with each element in the sample space. Furthermore, each random variable comes with an associated probability density function, $f_R(r)$, (or probability distribution), which allows us to understand the

probability of any event within the sample space. In laypeople's terms, a random variable represents when our experiment can have more than one value, but we cannot predict the value in advance. We typically find the chance (or probability) of observing an event through a Monte Carlo simulation of repeated experiments. For an illustrative example of the concept of a random variable, reference Appendix A.

Since the 1990s, the implementation of Monte Carlo techniques, even the simplest of applications, is inherently connected with utilizing computers because Monte Carlo provides a relatively easy and efficient solution to solving optimization, estimation, and sampling problems often encountered by researchers. For example, in the optimization area, the random sampling of Monte Carlo techniques enables a stochastic algorithm to escape local minimums and maximums, allowing the algorithm to explore the area of interest, unlike a deterministic optimization approach. In addition, mathematicians utilized Monte Carlo techniques to evaluate multi-dimensional integrals because most numerical methods suffer from the curse of dimensionality (Riemann sum). Finally, in statistics, Monte Carlo techniques serve as a way to mimic the behavior of a real-life process. Specifically to this thesis, Bayesian statistics use Markov chain Monte Carlo (special class of Monte Carlo) techniques to sample from a posterior distribution.

This chapter briefly discusses Monte Carlo integration to motivate Monte Carlo sampling techniques, which segues into Markov chain Monte Carlo theory and algorithms. We focus on Metropolis-Hasting, Gibbs sampler, Multi-try Metropolis, and Multiset sampler as the Markov chain Monte Carlo algorithms of interest due to the proposed methodology in the following chapters.

3.1 Monte Carlo Integration

Research-based or calculus-driven problems often require computing an integral:

$$I = \int_{\Omega_f} f(x)dx, \tag{3.2}$$

where Ω_f represents a high-dimensional region and $f(x)$ denotes the interested target function. Early in our mathematical careers, we learn about a variety of numerical techniques to approximate I, Eq. 3.2 such as Riemann sum integration, trapezoid rule, Simpson's rule, Netwon-Cotes, or Gauss-Legendre rules. The most elementary Riemann sum integration utilizes a constant interpolation with uniform partitions, $\rho_o = \{x_o, x_1, \ldots, x_n\}$, of Ω_f to break the region into rectangles as illustrated by Figure 3.2. Figure 3.2 illustrates a left Riemann sum approximation of a general $f(x)$ over the $\Omega_f = [a, b]$, blue curve, where we calculate the height of the rectangle using $f(x_i)$.

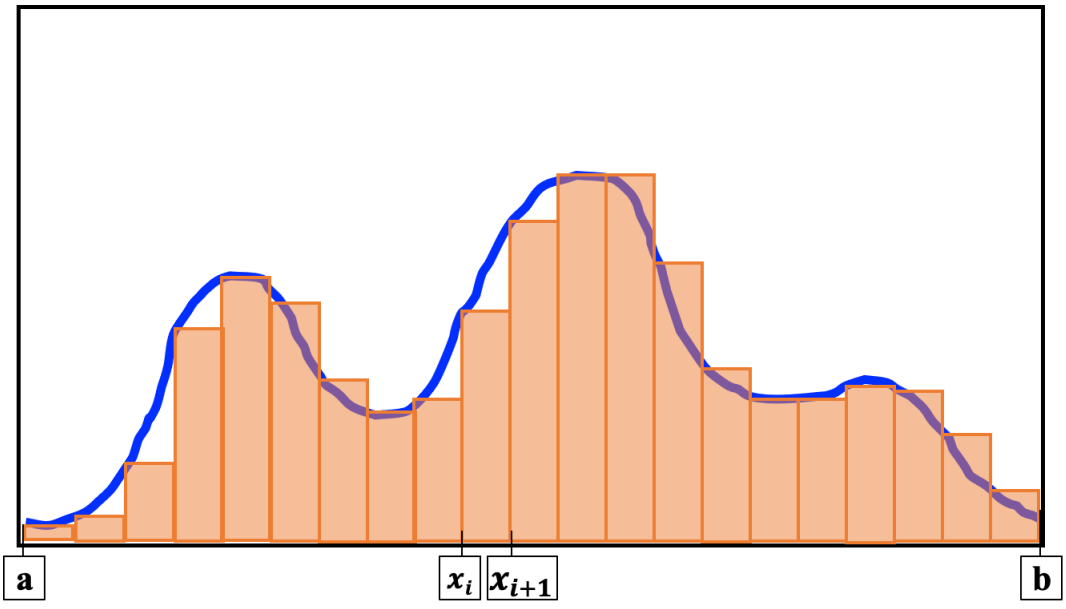


Figure 3.2: Illustration of Riemann sum integration.

We calculate and add the area of each rectangle to approximate I denoted as $\hat{I}$. Mathematically, we denote the Riemann sum approximation to I, Eq. 3.2, as:

$$\hat{I}_{Riemann} = \sum_{i=0}^{n} f(x_i)\Delta x, \tag{3.3}$$

where $\Delta x = (x_{i+1} - x_i)$ are chosen to be constant over a uniform partition where $a = x_0 \leq x_1 \leq x_2 \leq \cdots \leq x_n \leq b = x_{n+1}$, and $f(x_i)$ represents the evaluate of x_i given the function $f(x)$.

As the uniform partition width decreases towards zero, ($\Delta x \to 0$), the Riemann sum approximation of I, Eq. 3.3, converges to numerical result of Eq. 3.2. Note there are other Riemann summ approximation using constant interpolation such as using the Midpoint rule:

$$\hat{I}_{Midpoint} = \sum_{i=0}^{n} f\left(\frac{x_i + x_{i+1}}{2}\right)\Delta x,$$

and the right-Riemann sum approximation:

$$\hat{I}_{Right-Riemann} = \sum_{i=0}^{n} f(x_{i+1})\Delta x.$$

While there are a variety of numerical solutions and techniques to solve one-dimensional integration problems, the methods are limited by the curse of dimensionality, as demonstrated by Thisted [92]. Monte Carlo integration is an analogous, randomized approximation approach to Riemann sum integration that provides a straightforward, intuitive approach to

integrating complex, high-dimensional integrals. Since the basis of the Monte Carlo is generating a *random* answer in a finite amount of time, Monte Carlo integration "rewrite" Eq. 3.2 as the expectation of a random variable. The expectation of a random variable is the long-term weighted average of all possible events of X with weight associated with the probability of X occurring. Mathematically, we denoted the expectation of a random variable X as:

$$\mathbb{E}[x] = \sum x \times g(x)dx,$$

in the discrete case and:

$$\mathbb{E}[x] = \int_{\Omega_X} x \times g(x)dx,$$

in the continuous case where g(x) represent the probability distribution associated with our random variable, X. Probability theory tells us that if we have a function of our random variable, f(x), the expected value of the function is:

$$\mathbb{E}[f(x)] = \sum f(x) \times g(x)dx,$$

in the discrete case and:

$$\mathbb{E}[f(x)] = \int_{\Omega_X} f(x) \times g(x)dx,$$

in the continuous case.

The "simplest" Monte Carlo integration technique generates and evaluates random points from a uniform distribution on the interval [a,b] at the target distribution, $f(x)$. To incorporate the inclusion of uniform generated "random" points, we multiply Eq. 3.2 by a $\frac{1}{(b-a)}(b-a)$ resulting in:

$$I = \int_a^b f(x)dx = \int_a^b f(x) \times \frac{1}{(b-a)}(b-a)dx, \tag{3.4}$$

where we recognize $g(x) = \frac{1}{(b-a)}$ as the uniform density between a and b. The $g(x)$ distribution is referred to a the proposal distribution. After sliding the $(b-a)$ term outside the integral and substituting $g(x) = \frac{1}{(b-a)}$, we rewrite Eq. 3.4 as:

$$I = (b-a)\int_{\Omega_g} f(x)g(x)dx, \tag{3.5}$$

where Ω_g denotes the support of $g(x)$, the interval $[a, b]$ in this case. Following the definition of expectation, we express Eq. 3.5 as:

$$I = (b-a)\mathbb{E}_{g(x)}\left[f(x)\right], \tag{3.6}$$

represents the expectation of $f(x)$ with respect to $g(x)$ where our random variable is from a uniform distribution. We utilize a Monte Carlo approximation:

$$\mathbb{E}_{g(x)}\left[f(x)\right] \approx \frac{1}{n} \sum_{x_i \sim g(x) | i=1}^{n} f(x_i), \tag{3.7}$$

where $x_i \sim g(x)$ denoted sampling from $g(x)$ to evaluate the expectation in Eq 3.6 by sampling $(x_1, \ldots, x_n)$ from $g(x)$ which is a uniform density on the interval $[a, b]$ and computing the average [69]. Combining Eq. 3.6 and Eq. 3.7 results in:

$$\hat{I}_{MonteCarlo} = (b-a) \times \frac{1}{n} \sum_{x_i \sim g(x) | i=1}^{n} f(x_i), \tag{3.8}$$

illustrating the Monte Carlo integration under our uniform proposal distribution, which we refer to as naive Monte Carlo integration. By the Strong Law of Large Numbers, Eq. 3.8 converges almost surely to Eq. 3.2 as $n \to \infty$. Unlike Riemann sum integration, the Monte Carlo integration does not constrict the evaluated points to the uniform partition of ρ_o, but rather samples uniformly in the interval [a,b]. In this case, we were sampling uniformly from a uniform proposal distribution, but our choice of proposal distribution can differ. Thus, sampling uniformly does not imply sampling from a uniform distribution.

The objective of both Riemann sum and Monte Carlo integration is to approximate an integral such as Eq. 3.2. While both techniques take different approaches, the Riemann sum (Eq. 3.3) and Monte Carlo (Eq. 3.8) integration:

$$\hat{I}_{Riemann} = \sum_{i=0}^{n} f(x_i)\Delta x; \quad \hat{I}_{MonteCarlo} = (b-a) \times \frac{1}{n} \sum_{x_i \sim g(x) | i=1}^{n} f(x_i),$$

are inherently connected.

We replace our constant weights, $\Delta x = (x_{i+1} - x_i)$, in the Riemann sum integration with $\frac{1}{n}$ in Monte Carlo integration which denotes acquiring the samples uniformly from $g(x)$. Whereas, the additional scaling factor $(b - a)$ in the Monte Carlo approach accounts for introducing the proposal uniform distribution, $g(x) \equiv \text{Unif}(a, b)$.

3.2 Importance Sampling

While there are simple modifications to the constant interpolation Riemann sum, such as the right-Riemann or the Midpoint rule, we can improve the Riemann approximation to provide a better numerical approximation by adjusting the interpolation technique using quadrature techniques, such as the trapezoid rule and Simpson's rules. For instance, the trapezoid rule:

$$\hat{I}_{Trapezoid} = \sum_{i=0}^{n} \frac{1}{2} \left(f(x_i) + f(x_{i+1}) \right) \times \Delta x,$$

applies a linear interpolation obtained by averaging the left-hand $f(x_i)$ and right-hand $f(x_{i+1})$ Riemann integration schemes. Figure 3.3 demonstrates the trapezoid rule, using the same partitions and target function, $f(x)$ as Figure 3.2 which demonstrated a left-Riemann integration scheme. We adjusted the interpolation scheme to provide a "better" approximation of $f(x)$ with less evaluated points. We define better approximation as the decrease in the difference in area between the orange histogram and the function of interest (blue line).

Additionally, we could have applied Simpson's rule which utilizes a quadratic interpolation whose formula is:

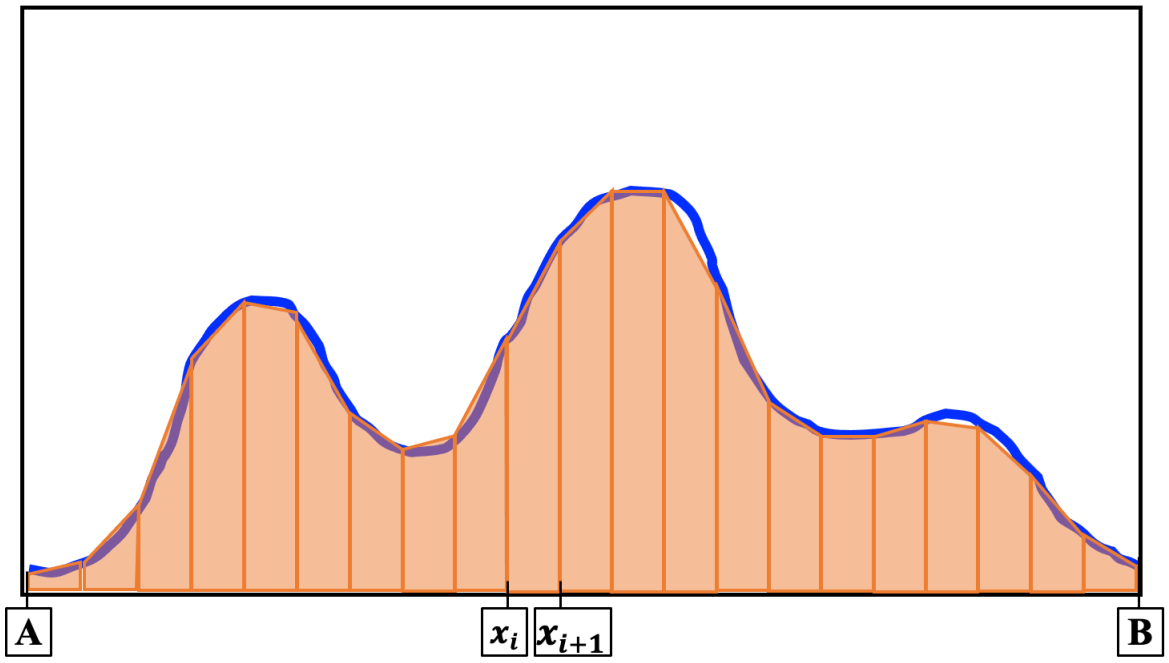


Figure 3.3: Illustration of trapezoid rule integration.

$$\hat{I}_{Simpsons} = \frac{\Delta x}{6} \times \left[f(a) + 4 \sum_{i=1}^{N} f(x_{2i-1} + 2 \sum_{i=1}^{N} f(x_{2i})_f(b) \right],$$

where we have equally spaced $\Delta x = (x_{i+1} - x_i)$. Note that there are a variety of approaches such as orthogonal polynomials (Legendre [13]), splines [40], or quasi-Monte Carlo techniques to approximate an interval. However, we use different interpolation schemes of Riemann sum integration to motivate Monte Carlo integration, as most researchers are familiar with quadrature techniques due to high school math classes.

As an aside, while 'Monte Carlo' is the name, quasi-Monte Carlo is more of a numerical solution than a simulation approach. The idea behind quasi-Monte Carlo is to replace the randomly generated uniform numbers ($X \sim \text{Uniform}(0, 1)$) with a deterministic sequence on [0,1] to minimize the distance between the empirical CDF and the uniform distribution

known as the Kolmogorov - Smirnov distance.

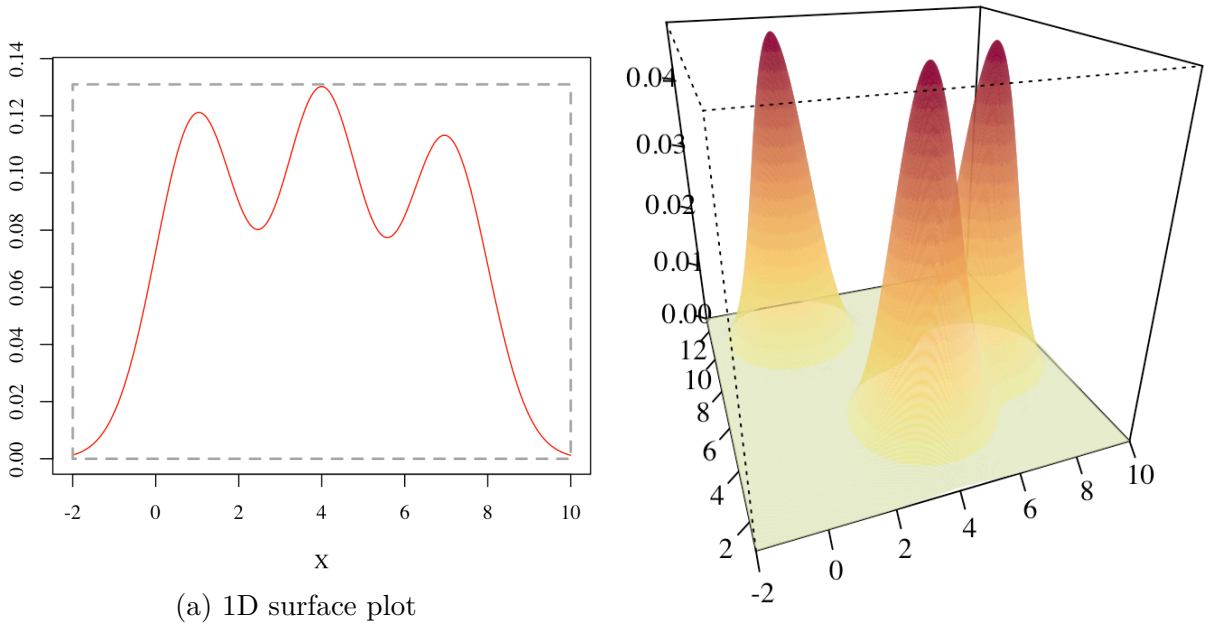
The naive Monte Carlo strategy in high-dimensional regions suffers similarly to its deterministic approaches because the algorithm wastes computational effort in evaluating random samples located in regions where the function is almost zero. For instance, consider Figure 3.4 that illustrates a one dimensional (1D) and two dimensional (2D) surface plot of interested functions. When we utilize a uniform proposal scheme (naive Monte Carlo) in Figure 3.4a, illustrated by the grey line, we collect samples from the high-density region. On the other hand, when we increase the dimensionality by one in Figure 3.4b and use a bi-variate uniform proposal scheme, we often sample from low-density regions and wastes computational resources. Naive Monte Carlo approaches, which uniformly sample from simple regions, are bound to fail in high-dimensional problems.

We utilized more sophisticated interpolation schemes for Riemann sum integration to better fit our function $f(x)$ and improve our approximation of our integral, Eq. 3.2. Importance sampling provides an analogous extension to the naive Monte Carlo integration, discussed in Section 3.1, by sampling from regions of "importance" [64]. Importance sampling performs a "weighted" sampling scheme where the weights depend on a ratio between the target function, $f(x)$ and a proposal distribution, $g(x)$. The importance sampling algorithm works by

1. sampling $(x_1, \ldots, x_n)$ from the proposal distribution, $g(\bullet)$,

2. calculating the importance weight,

$$I.W. = \frac{f(x)}{g(x)}, \tag{3.9}$$



(a) 1D surface plot

(b) 2D surface plot

Figure 3.4: Illustration of 1D and 2D distribution to demonstrate the limitation of naive Monte Carlo

3. resampling $(x_1, \ldots, x_n)$ based on importance weights.

Using the importance sampling process, we draw samples from our target distribution, $f(x)$; however, this does not yet integrate over $f(x)$ on some interval [a,b], Eq. 3.2. To adjust from naive Monte Carlo method "equally" weights sampled points to importance sampling, we generalize Eq. 3.4 as:

$$I = \int_a^b f(x)dx = \int_{-\infty}^{\infty} \frac{f(x)}{g(x)} \times g(x) \times \delta(a \leq x \leq b)dx, \tag{3.10}$$

where $\delta(a \leq x \leq b)$ denotes an indicator function of the range of x. Following the definition of expectation, we rewrite Eq. 3.10 as:

$$I = \mathbb{E}_{g(x)} \left[\frac{f(x)}{g(x)} \times \delta(a \leq x \leq b) \right], \tag{3.11}$$

which represents the expectation of $f(x)$ with respect to $g(x)$ as the random variable. We utilize a Monte Carlo approximation:

$$\hat{I}_{ImportanceSampling} \approx \frac{1}{N} \sum_{x_i \sim g(x) | i=1}^{N} \frac{f(x)}{g(x)} \times \delta(a \leq x \leq b),$$

where $\frac{f(x)}{g(x)}$ represent the importance ratio or "weights" to evaluate the expectation in Eq. 3.11 by sampling $(x_1, \ldots, x_n)$ from $g(x)$ which is a density on the interval $[a, b]$ and computing the average. The "weights" in importance sampling aid in sampling observations from the target distribution more frequently than those not in the target distribution. Additionally, we can improve the algorithm by choosing a proposal distribution closer to the target distribution. For instance, in Figure 3.4a, we utilized a uniform proposal distribution; however, we can improve our algorithm by using a proposal distribution that "fits" closer to our target function, such as the one demonstrated in Figure 3.5.

A key factor in developing a "good" sampling scheme is the choice of proposal distribution which is often a major limitation to Monte Carlo methods, especially in the high-dimensional cases. A reasonable choice in proposal distribution is picking a $g(X)$ with heavier tails than the target distribution, $f(X)$. Another Monte Carlo alternative approach, Markov chain Monte Carlo, utilizes dependent samples to draw samples from the target distribution.

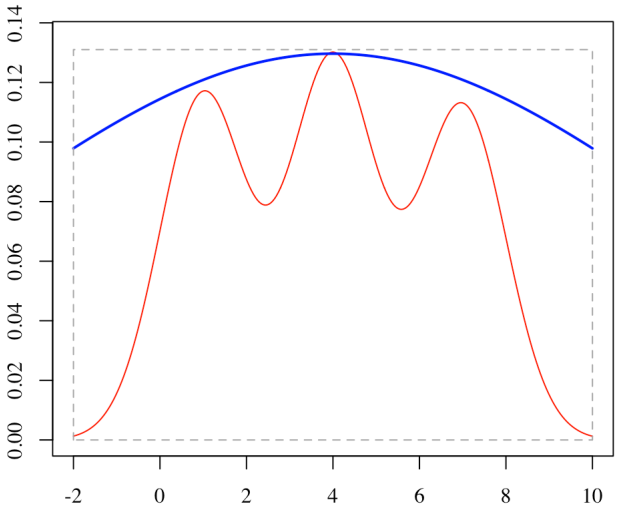


Figure 3.5: Illustration of using different proposal scheme for the importance sampling

3.3 Markov chain Monte Carlo

In importance sampling, we utilized weighted sample scheme from a proposal distribution, $g(X)$, which was different but close in shape to the target distribution. An alternative approach Monte Carlo, Markov chain Monte Carlo, utilizes correlated - or dependent - samples to draw samples from the target distribution. Bayesian statistic heavily utilizes Markov chain Monte Carlo methods to sample from the posterior distribution, $f(\theta|\mathbf{X})$. Note that a Bayesian's posterior distribution is the target distribution. In the following sections, we define and discuss Markov chain theory followed by discussing the foundational Markov chain Monte Carlo algorithm, Metropolis-Hasting and an extension known as the Gibbs sampler.

3.3.1 Markov chains

A Markov chain describes a sequence of possible events in which the probabilities of the next event only depends on the previous state. Mathematically, if we let x_t represent the event

at time t, then:

$$Pr(x_t|x_{t-1}, ..., x_1) = Pr(x_t|x_{t-1}).$$

When learning about Markov chains, it is often easier to first discuss discrete Markov chains before moving to continuous Markov chains. A discrete Markov chain utilizes a transition matrix to represent the probability of moving from state i to state j,

$$A = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1j} \\ \vdots & \ddots & \vdots \\ \alpha_{i1} & \dots & \alpha_{IJ} \end{bmatrix},$$

where the row index, $i = 1, \dots, I$, represent where we start and the column index, $j = 1, \dots, J$ represent where we are going. Additionally, the sum of the rows must sum to one and the elements (i.e., probabilities) must be between zero and one. That is,

$$\sum_{j=1}^{J} A_{ij} = 1; \text{ and } 0 \leq \alpha_{ij} \leq 1.$$

A common application of discrete Markov chains is sabermetrics which is the application of statistical analysis to baseball records, especially to evaluate and compare the performance of individual players or teams. Discrete Markov chains offer a quick computational solution for calculating the expected scores. Thus, we consider a softball game to conceptualize the utility and practicality of Markov chains.

In softball, an inning starts with zero outs and no runners on base, as highlighted in green in

Figure 3.6. From this initial state, an out (denoted as O) will cause a transition to the state with one out with probability P(O). Likewise, a walk (indicated as BB) or a single (denoted as 1B) cause a transition to the upper left state with a runner on first base and no change in score or outs with probability P(BB) + P(1B). Additionally, Figure 3.6 illustrates other types of transitions based on batting outcomes, such as doubles (2B), triples (3B), and home runs (HR). Often, sports analysts calculate the transition probabilities through the relative frequencies for each team based on their historical records.

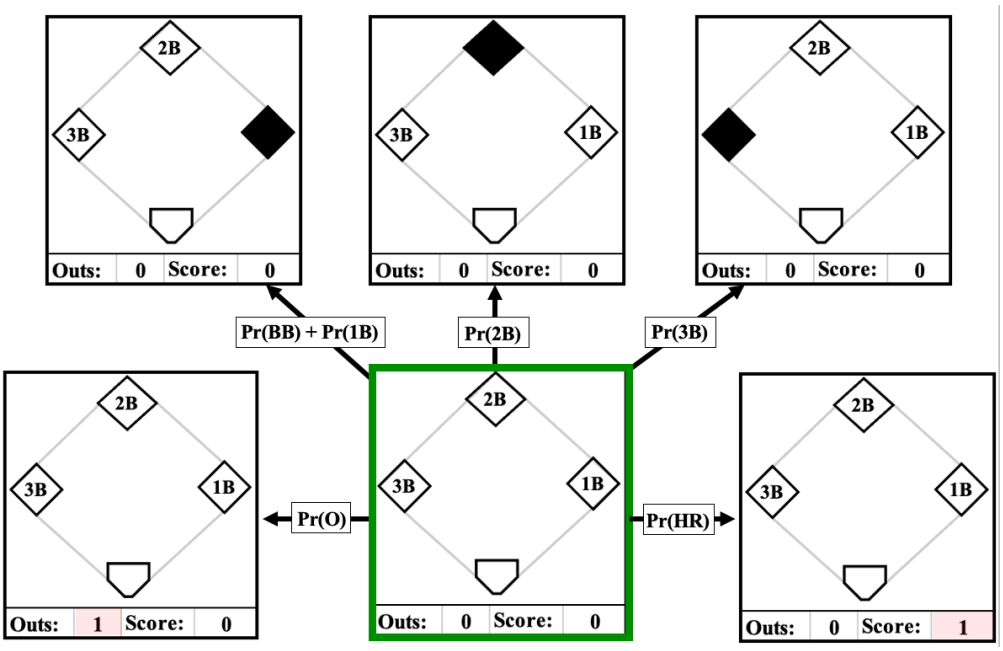


Figure 3.6: Softball illustration of discrete Markov chain.

Softball and baseball applications are prime example of Markov chains because each event in these games depends on the previous play. In the above example, we described a one-step transition matrix. However, a softball game would repeatly bring hitter to the plate until there was three outs. Thus, we need to a N-step transition matrix which is defined as:

$$A^N = \prod_{n=1}^{N} \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1j} \\ \vdots & \ddots & \vdots \\ \alpha_{i1} & \dots & \alpha_{IJ} \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1j} \\ \vdots & \ddots & \vdots \\ \alpha_{i1} & \dots & \alpha_{IJ} \end{bmatrix} \times \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1j} \\ \vdots & \ddots & \vdots \\ \alpha_{i1} & \dots & \alpha_{IJ} \end{bmatrix} \times \cdots \times \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1j} \\ \vdots & \ddots & \vdots \\ \alpha_{i1} & \dots & \alpha_{IJ} \end{bmatrix}.$$

If extend the N-step transition matrix to infinity (∞), then we would produce the limiting distribution. Mathematically, we express the limiting distribution as:

$$\lim_{N->\infty} [A]^N = \pi_j,$$

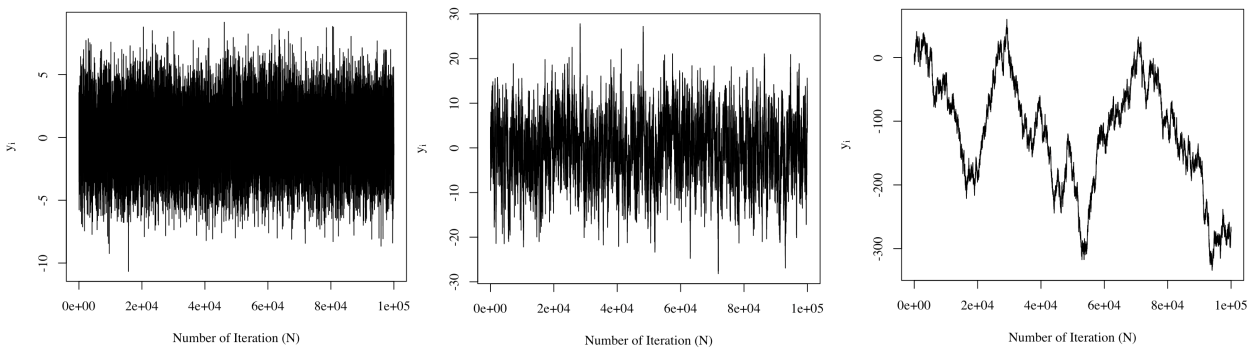
where π_j represents the probability of being in state j regardless of the starting position. That is, the limiting distribution is only concentrated on where we end. That is, the limiting distribution concentrates on where we end. However, the limiting distribution only exists if our transition matrix, A, is ergodic.

Ergodic Properties

An ergodic transition matrix has the following properties: (1) positive recurrence, (2) irreducibility, and (3) a-periodicity. A Markov chain is positive recurrent if the expected amount of time to return to the current state is finite. That is, we will return to our original state in some finite amount of time. Two counterexamples of positive recurrence is (1) our softball example and (2) an autoregression model where our coefficient is greater than one. The softball example does not hold the positive recurrence properties because once we get three outs, the inning is over, and we can not return to the original state of zero runners on base and zero outs. In an autoregression model with lag one, the model is:

$$y_i = \theta y_{i-1} + \epsilon_i,$$

for $i = 1, \ldots, N$ where y_0 represents an initial point and $\epsilon \sim N(0, \sigma^2)$ denotes the error structure. In most autoregressive models, $|\theta| < 1$ and the model would hold the positive recurrence property as demonstrated in Figure 3.7a and 3.7b where $\theta = 0.90, 0.99$, respectively. However, Figure 3.7c illustrates when $|\theta| \geq 1$, specifically $\theta = 1.00$ and the line trends away and we are not guaranteed that we will return to the original state in a finite amount of time.



(a) $\theta = 0.90$ (b) $\theta = 0.99$ (c) $\theta = 1.00$

Figure 3.7: Demonstration of θ value in an autoregression model lag 1 (AR(1)) when $\theta = \{0.90, 0.99, 1.00\}$

A Markov chain is irreducible if we can eventually get from every state to every other state with positive probability. For instance, Figure 3.8 provides an example of a non-irreducible Markov chain where we give the transition matrix with a corresponding visualization of the probability of moving from each state. In this example, once we get into state C, we are trapped and can not leave.

$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.1 & 0.7 & 0.2 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \end{array}$$

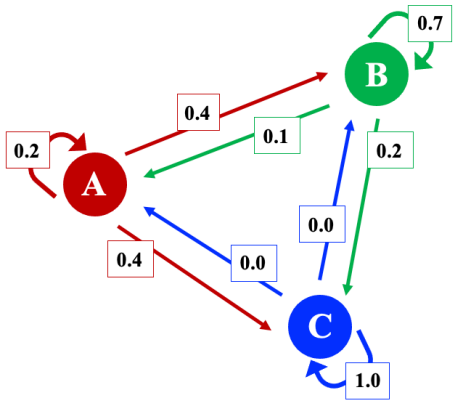


Figure 3.8: Example of a non-irreducible Markov chain.

We can adjust Figure 3.8 to be an irreducible Markov chain by adjusting the probabilities to transition out of state C and into the other states, as demonstrated in Figure 3.9. Note that we do not require that each element be a positive probability; we need every state to connect eventually.

$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.1 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array}$$

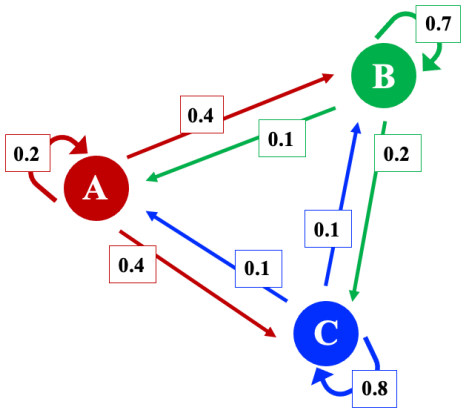


Figure 3.9: Example of an irreducible Markov chain.

The positive recurrence and irreducible properties tell us that the Markov chain can explore the entire space. Meanwhile, a-periodicity ensures that there are no funny or deterministic patterns when exploring the parameter space. For instance, in our softball example, we have

constraints that require us to follow a "specific" pattern that mimics a real softball game. Therefore, aperiodic Markov chains do not have a "deterministic" pattern. Mathematically, a Markov chain is a-periodic if all states of a Markov chain have a period of one. State "i" of a Markov chain is said to have period D, if:

$$A_{ij}^{N} = 0,$$

for N not divisible by D and D is the largest integer with this property. Often, if a Markov chain contains a self-loop as demonstrated in Figure 3.8 and 3.9, then our Markov chain has a period of one.

This section describes the ergodic properties for discrete Markov chains to help the reader conceptual understand these properties. In Section 3.3.2, 3.3.3 and 3.4, we discuss a variety of Markov chain Monte Carlo algorithms focused on sampling from continuous posterior distributions, where we require our proposal distribution $g(\bullet)$ to maintain the ergodic properties. The ergodic properties ensure that our Markov chain Monte Carlo algorithm eventually converges to sampling from the stationary distribution, which in the Bayesian framework is our posterior distribution. We validate that Markov chain Monte Carlo methods limit to the stationary distribution for different algorithms in Section 3.3.2, and 3.3.3. Before discussing specific algorithms, we must talk about stationary distribution and basic properties of Markov chain Monte Carlo algorithms.

Stationary Distribution

While the limiting distribution might not always exist, the stationary distribution will always exists. A stationary distribution is defined for N- state Markov chain in the discrete case as:

$$\pi(x=j) = \sum_{i=1}^{N} Pr(x=j|x=i) * \pi(x=i) \iff A^T\pi = \pi,$$

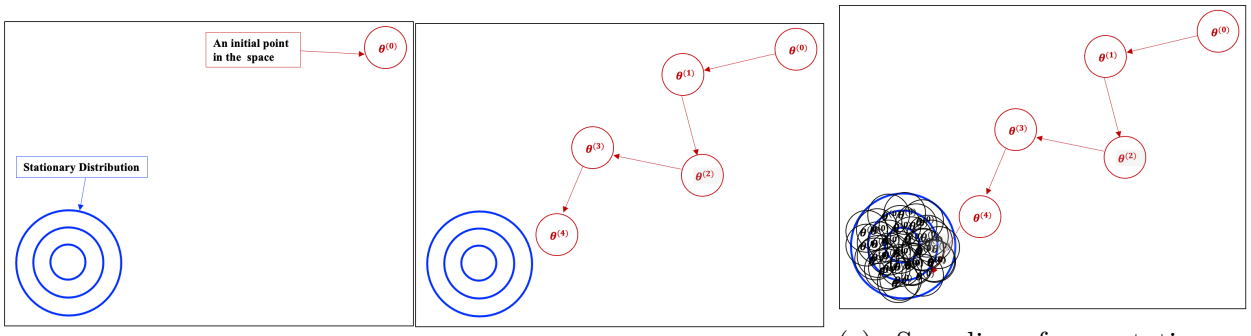
then π is the stationary distribution where $\sum_{j=1}^{J} \pi_j = 1$, $\pi(x=i)$ and $\pi(x=j)$ represent the probability of being in the i^{th} and j^{th} state respectively, and $Pr(x=j|x=i)$ denotes the transition probability given we are in state i and move to state j. In the continuous case, we denoted the stationary distribution as:

$$\pi\left(\theta^{(t)}\right) = \int \rho\left(\theta^{(t)}|\theta^{(t-1)}\right)\pi\left(\theta^{(t-1)}\right)d\theta^{(t-1)}, \tag{3.12}$$

where $\pi(\theta^{(t)})$ represent the stationary distribution and $\rho\left(\theta^{(t)}|\theta^{(t-1)}\right)$ represents the transition probability, i.e., continuous version of transition matrix, A. Note that $\pi\left(\theta^{(t-1)}\right)$ and $\pi\left(\theta^{(t)}\right)$ are the same function evaluated at different locations. In a Markov chain Monte Carlo, we aim to sample from the stationary distribution of interest. From a Bayesian perspective, the stationary distribution is the joint posterior distribution. Under Eq. 3.12 construction, we are guaranteed that if we sample out of the stationary distribution, our Markov process will eventually converge to the stationary distribution. Additionally, once we reach the stationary distribution, we do not leave the stationary distribution.

For instance, Figure 3.10 illustrates three important phases of any Markov chain Monte Carlo

algorithm as it explores the parameter space to find the stationary distribution represented by the blue circle. First, all MCMCs require an initialization point to kick start the algorithm, as demonstrated in Figure 3.10a. We present picking a random starting point within our parameter space; however, various ways exist to initialize MCMC algorithms such as using multiple start points or a deterministic algorithm to initial. Once we have our starting location, the algorithm explores the space based on some transition rate, $\rho\left(\theta^{(t)}|\theta^{(t-1)}\right)$, illustrated in Figure 3.10b, until it reaches the stationary distribution in Figure 3.10c. The θ values in Figure 3.10b, referred to as burn-in, have no impact on the samples and are cut off before any inferences. Additionally, notice that our samples do not leave the stationary distribution once we reach the stationary distribution.



(a) Initialization (b) Burn-in process (c) Sampling from stationary distribution

Figure 3.10: Illustration of the Markov chain Monte Carlo algorithm searching for stationary distribution

In Section 3.3.2, we motivate and outline the foundational Markov chain Monte Carlo algorithm known as the Metropolis-Hasting algorithm. The Metropolis-Hasting algorithm lays the groundwork for all other MCMC algorithms, which we utilize within our novel methodology in Chapters 4 and 5.

3.3.2 Markov chain Monte Carlo Algorithms

Markov chain Monte Carlo (MCMC) methods are wondering stochastic algorithms where our goal is to obtain values from our posterior distribution, $f(\theta|\mathbf{X})$. However, we cannot directly sample from $f(\theta|\mathbf{X})$, but we can write the distribution out to proportionality. The ability to sample and analytical write down a function are two inherently different concepts. For instance, a common beginner Bayesian example is the posterior distribution of the mean, μ, and precision $\phi = 1/\sigma^2$ where σ^2 variance with reference prior for μ and ϕ. The posterior distribution, known as a normal-Gamma distribution, is proportionally:

$$f_{\text{normal-Gamma}}(\mu, \phi|\mathbf{X}) \propto \phi^{\frac{N}{2}-1} e^{-\frac{\phi \sum_{i=1}^{N}(x_i-\mu)^2}{2}}.$$

Compare to the individual normal distribution or Gamma distribution which we sample from using direct computer software functions, the Normal-Gamma distribution does not have a direct sampler function. Thus, we can express the equation, but cannot sample from it. We could easily expand this example to a large hierarchical model with several parameters of interest when we would not be able to directly sample.

Now, we could an talk about marginal and conditional relationships, but we want to understand the relationship between the parameters and the only way to know this is through studying the joint distribution of the parameters. Markov chain Monte Carlo methods provide us the ability us to sample from relatively complex distribution. In Figure 3.11, we illustrate a conceptual example of wanting to sample from a complex distribution $f(\theta|\mathbf{X})$ represented by the blue line. When we implement a MCMC technique, we will eventually collect samples of θ from $f(\theta|\mathbf{X})$ represented by the histogram.

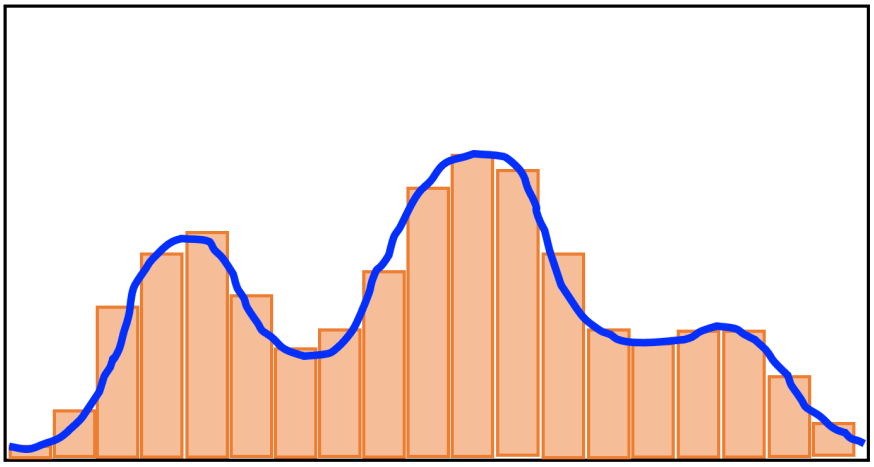


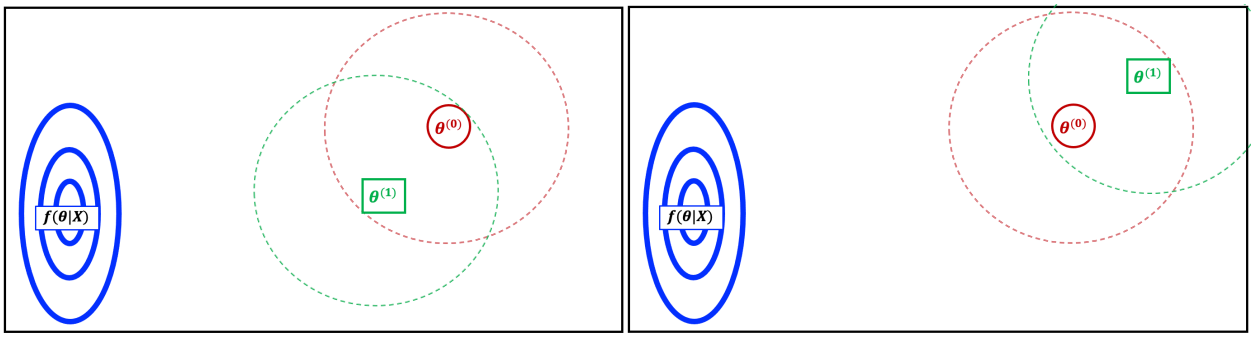
Figure 3.11: Illustrative example to motivation the end goal of Markov chain Monte Carlo method.

In 1953, Nicholas Metropolis introduced the foundation ground of all MCMC method by developing the Metropolis algorithm [68]. Keith Hastings generalized Metropolis' ideas in 1970 with the Metropolis-Hasting algorithm, which enabled the proposal distribution to have fewer restrictions [42]. Thus, we discuss the Metropolis-Hasting algorithm in depth and make connections to other Markov chain Monte Carlo variation such as the Metropolis Algorithm and Gibbs sampler. Additionally, we extend our discussion of Markov chain Monte Carlo algorithms to include ensemble techniques known as the Multi-try Metropolis and the Multiset sampler in Section 3.4.

Metropolis-Hasting Algorithm

The Metropolis-Hasting (M-H) Algorithm provides us with a wandering algorithm to jointly samples from posterior distribution, $f(\theta|\mathbf{X})$, using two steps. Figure 3.12 provides an illustrative perspective of the M-H algorithm where our goal is to sample from a high-mass

distribution $f(\theta|\mathbf{X})$ represented by the blue contour. With Markov chain Monte Carlo algorithms, we require initializing a start value denoted by $\theta^{(0)}$ before running the algorithm. Given an initial point, we propose a new location, $\theta^{(1)}$, by sampling from our proposal distribution, $g(\bullet)$. Figure 3.12a and 3.12b display two possible proposal moves where the dashed circle represents an tunable deviance of our current location.



(a) Proposal move towards the stationary distribution

(b) Proposal move away from the stationary distribution

Figure 3.12: Illustration of two proposal moves, denoted by $\theta^{(1)}$, given the current location, $\theta^{(0)}$, and its respective deviance (red dashed circle.

We decide to stay at the current location ($\theta^{(0)}$) or move to the new proposed location ($\theta^{(1)}$) based on an acceptance probability:

$$\alpha = \min\left(1, \frac{f(\theta^*|\mathbf{X})g\left(\theta^{(t-1)}|\theta^*\right)}{f(\theta^{(t-1)}|\mathbf{X})g\left(\theta^*|\theta^{(t-1)}\right)}\right). \tag{3.13}$$

If we closely investigate the acceptance probability, the probability consists of two ratios that balance exploration and exploitation. The ratio of our sampling densities (up to proportionality):

$$RSD = \frac{f(\theta^*|\mathbf{X})}{f(\theta^{(t-1)}|\mathbf{X})},$$

determines whether our density is higher for our proposed, θ^*, compared to our current value, $\theta^{(t-1)}$ signifying we should move to our proposed values. In Figure 3.12a, the $f(\theta^*|\mathbf{X}) > f(\theta^{(t-1)}|\mathbf{X})$ and we would accept the new proposal value. However, in Figure 3.12b, our proposal value is further away from $f(\theta|\mathbf{X})$. That is, $f(\theta^*|\mathbf{X}) < f(\theta^{(t-1)}|\mathbf{X})$, so we would expect stay at the current value.

The inverse ratio of the proposal distribution:

$$IRP = \frac{g\left(\theta^{(t-1)}|\theta^*\right)}{g\left(\theta^*|\theta^{(t-1)}\right)}, \tag{3.14}$$

balances the exploration of the Metropolis-Hasting algorithm by accounting for the probability the algorithm can return to the original parameter space. A well-constructed proposal distribution has ergodic properties such as the proposal distribution can (1) allows for exploration of the parameter space (irreducibility), (2) enables the ability to return to starting location in a finite amount of time (positive recurrence), and (3) there are no systematic patterns (a-periodic). Algorithm 3.1 outlines a general Metropolis-Hasting Algorithm where θ denotes the interested parameter value and θ^* denotes the proposed value.

Algorithm 3.1: General Metropolis-Hasting Algorithm

Initialize values for $\theta^{(t=0)}$

for $t \leftarrow 1$ **to** T

1. PROPOSAL A NEW VALUE $\theta^* \sim g\left(\theta^* | \theta^{(t-1)}\right)$

2. DECISION

$$\theta^{(t)} = \begin{cases} \theta^* & \text{with probability } \alpha = \min\left(1, \frac{f(\theta^*|\mathbf{X}) g\left(\theta^{(t-1)}|\theta^*\right)}{f(\theta^{(t-1)}|\mathbf{X}) g\left(\theta^*|\theta^{(t-1)}\right)}\right) \\ \theta^{(t-1)} & \text{with probability } 1-\alpha \end{cases}$$

end

In the Metropolis-Hasting algorithm, our choice of the proposal distribution, $g(\bullet)$, affects our Monte Carlo algorithms ability to move around the space and efficiently sample from the target distribution. Figure 3.13 illustrates a proposal distribution with a tunable deviance , ψ, changing between small, medium, and large values from the current value. When our deviance is small (left), the algorithm requires more iterations to properly and fully sample from $f(\theta|\mathbf{X})$. Conversely, when the proposal deviance is large (right), we obtain proposal values outside of our target distribution, thus the algorithm gets stuck at a currently location a lot.

In Figure 3.13, we utilized a local proposal distribution with a symmetric variance. A local proposal distribution is a distribution centered around the previous observation with a tunable parameter of deviance from on current location. There are a variety of other proposal such as symmetric, independent, mixture, or multi-try proposals schemes.

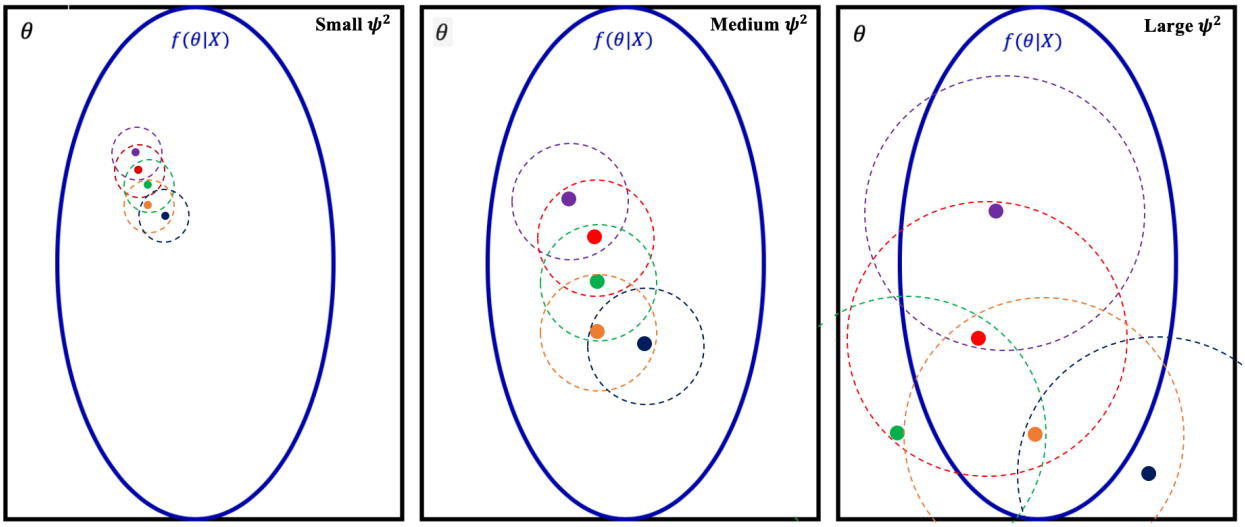


Figure 3.13: Demonstration of proposal's deviance in Metropolis-Hasting algorithm.

Proposal Schemes The original Metropolis algorithm [68] utilized symmetric proposal distributions most likely to avoid dealing with intractable densities because of the lack of computational power. A symmetric proposal is not any symmetric distribution, such as uniform or normal distribution or t-distribution; but, rather a distribution where the ratio of proposals, Eq. 3.14, in the acceptance probability cancels out. We characterize a symmetric proposal as:

$$g\left(\theta^{*}|\theta^{(t-1)}\right) = g\left(\theta^{(t-1)}|\theta^{*}\right).$$

For instance, a common symmetric proposal distribution for the mean is a normal distribution centered at the current location with some tunable variance denoted as:

$$\mu^* \sim \text{Normal}\left(\mu^{(t-1)}, \psi\right),$$

where $\mu^{(t-1)}$ represents the current state location and ψ is a tunable deviance parameter. However, if the normal distribution is not centered at the current location, then this is no longer a symmetric proposal and the ratio of proposal does not cancel out. An advantage of utilizing this proposal is it may save computing time and easy to implement. However, the limitation of a symmetric proposal is tuning the deviance parameter and the possibility of not exploring the space enough.

The Independent Metropolis-Hastings (or Metropolized independent sampling) algorithm utilize a proposal distribution that is independent of our current location [60, 94]. That is, our proposal distribution criterion is:

$$g\left(\theta^* | \theta^{(t-1)}\right) = g(\theta^*),$$

which reduces to down to an algorithm similar to importance sampling. The efficiency of this algorithm depends heavily on our proposal distribution being close to our target distribution.

Both the symmetric and the independent proposal schemes make local moves; therefore, they potentially will not explore the entire space. Thus, a practical thought is to sometimes randomly sample a value out in the tails of our distribution to ensure we do not get trapped in a local mode. One type of proposal scheme incorporates a two-component mixture model enabling the algorithm to propose local and non-local moves. Within our proposal schemes for

our Modified Cauchy Net, we utilize a multi-try procedure known as the Multi-try Metropolis, which suggests M proposal values rather than a single proposal value.

Detailed Balance

In Section 3.3.1, we discussed the basic property of Markov chains and the ergodic properties. The Metropolis-Hasting algorithm is a Markov chain where the transition rate of moving is:

$$\rho\left(\theta^{*}|\theta^{(t-1)}\right) = \alpha \times g(\bullet), \tag{3.15}$$

where α is the acceptance probability, Eq. 3.13, and $g(\bullet)$ is a proposal distribution. Thus, Metropolis-Hasting (Markov process) limits to sampling from our target distribution, $f(\theta|\mathbf{X})$ if our proposal distribution can explore the space and return in a finite amount of time, and the process is aperiodic. We know that the Markov process is aperiodic because the decide rule is a binomial random variable with a positive probability of staying in the same location. To prove the Metropolis-Hasting algorithm will eventually limit to the stationary distribution, $f(\theta|\mathbf{X})$, we need detailed balanced to hold which states:

$$\rho\left(\theta^{(t)}|\theta^{(t-1)}\right) f\left(\theta^{(t-1)}\right) = \rho\left(\theta^{(t-1)}|\theta^{(t)}\right) f\left(\theta^{(t)}\right), \tag{3.16}$$

where $\rho\left(\theta^{(t)}|\theta^{(t-1)}\right)$ represents Markov transition, $f\left(\theta^{(t)}\right)$ is shorthand notation for the stationary distribution $f(\theta|\mathbf{X})$, and $\rho\left(\theta^{(t-1)}|\theta^{(t)}\right)$ is the reverse process. When $\rho\left(\theta^{(t)}|\theta^{(t-1)}\right) = \rho\left(\theta^{(t-1)}|\theta^{(t)}\right)$, then our process is time reversible. Under most localize proposal strategies, the Markov process are time reversible, but not all Markov chains are time reversible.

Before proving that detailed balances holds for Metropolis-Hasting, we prove that $f\left(\theta^{(t)}\right)$ is stationary by integrating over Eq. 3.16 with respect to our current location:

$$\int \rho\left(\theta^{(t)}|\theta^{(t-1)}\right) f\left(\theta^{(t-1)}\right) d\theta^{(t-1)} = \int \rho\left(\theta^{(t-1)}|\theta^{(t)}\right) f\left(\theta^{(t)}\right) d\theta^{(t-1)}.$$

On the right-hand side, we slide the integrand over our function of $\theta^{(t)}$:

$$\int \rho\left(\theta^{(t)}|\theta^{(t-1)}\right) f\left(\theta^{(t-1)}\right) d\theta^{(t-1)} = f\left(\theta^{(t)}\right) \int \rho\left(\theta^{(t-1)}|\theta^{(t)}\right) d\theta^{(t-1)}$$

because $f\left(\theta^{(t)}\right)$ a constant with respect to the integral. Then, we know $\int \rho\left(\theta^{(t-1)}|\theta^{(t)}\right) d\theta^{(t-1)} = 1$ because it is proper distribution resulting in:

$$\int \rho\left(\theta^{(t)}|\theta^{(t-1)}\right) f\left(\theta^{(t-1)}\right) d\theta^{(t-1)} = f\left(\theta^{(t)}\right),$$

which is the definition of stationary, Eq. 3.12.

Metropolis-Hasting Detailed Balance We need the transition probability to validate that detailed balance holds for the Metropolis-Hasting algorithm. The forward transition rate, moving from $\theta^{(t-1)}$ to θ^{*}, is:

$$\begin{aligned}\rho\left(\theta^{*}|\theta^{(t-1)}\right) &= g\left(\theta^{*}|\theta^{(t-1)}\right)\times\alpha \\ &= g\left(\theta^{*}|\theta^{(t-1)}\right)\times\min\left\{1,\frac{f(\theta^{*}|\mathbf{X})g(\theta^{(t-1)}|\theta^{*})}{f(\theta^{(t-1)}|\mathbf{X})g(\theta^{*}|\theta^{(t-1)})}\right\}.\end{aligned}$$

If we assume, without loss of generality, that:

$$\frac{f(\theta^{*}|\mathbf{X})}{f(\theta^{(t-1)}|\mathbf{X})}\frac{g\left(\theta^{(t-1)}|\theta^{*}\right)}{g\left(\theta^{*}|\theta^{(t-1)}\right)}\geq 1, \tag{3.17}$$

then, this implies $\alpha = 1$ resulting in:

$$\rho\left(\theta^{*}|\theta^{(t-1)}\right) = g\left(\theta^{*}|\theta^{(t-1)}\right)\times 1.$$

The reverse transition rate, moving from $\theta^{(t-1)}$ to θ^{*}, is:

$$\begin{aligned}\rho\left(\theta^{(t-1)}|\theta^{*}\right) &= g\left(\theta^{(t-1)}|\theta^{*}\right)\times\alpha \\ &= g\left(\theta^{*}|\theta^{(t-1)}\right)\times\min\left\{1,\frac{f\left(\theta^{(t-1)}|\mathbf{X}\right)g\left(\theta^{*}|\theta^{(t-1)}\right)}{f\left(\theta^{*}|\mathbf{X}\right)g\left(\theta^{(t-1)}|\theta^{*}\right)}\right\}.\end{aligned}$$

With our assumption Eq. 3.17, then we know that:

$$\frac{f(\theta^{(t-1)}|\mathbf{X})}{f(\theta^{*}|\mathbf{X})}\frac{g\left(\theta^{*}|\theta^{(t-1)}\right)}{g\left(\theta^{(t-1)}|\theta^{*}\right)} < 1,$$

resulting in:

$$\begin{aligned}\rho\left(\theta^{(t-1)}|\theta^{*}\right) &= g\left(\theta^{(t-1)}|\theta^{*}\right) \times \frac{f(\theta^{(t-1)}|\mathbf{X})}{f(\theta^{*}|\mathbf{X})}\frac{g\left(\theta^{*}|\theta^{(t-1)}\right)}{g\left(\theta^{(t-1)}|\theta^{*}\right)} \\ &= \frac{f(\theta^{(t-1)}|\mathbf{X})g\left(\theta^{*}|\theta^{(t-1)}\right)}{f(\theta^{*}|\mathbf{X})}.\end{aligned}$$

When we substitute our forward and reverse transition rates into Eq. 3.16, we have:

$$g\left(\theta^{*}|\theta^{(t-1)}\right) f\left(\theta^{(t-1)}|\mathbf{X}\right) = \frac{f\left(\theta^{(t-1)}|\mathbf{X}\right)}{f(\theta^{*}|\mathbf{X})} g\left(\theta^{*}|\theta^{(t-1)}\right) f\left(\theta^{*}|\mathbf{X}\right),$$

which reduces to:

$$g\left(\theta^{*}|\theta^{(t-1)}\right) f\left(\theta^{(t-1)}|\mathbf{X}\right) = f\left(\theta^{(t-1)}|\mathbf{X}\right) g\left(\theta^{*}|\theta^{(t-1)}\right),$$

thus, detailed balance hold.

3.3.3 Gibbs sampler

A Gibbs sampler is specialized version of a Metropolis-Hasting algorithm where the proposal distributions are the full conditional distributions [17, 86, 94]. A full conditional distribution is a distribution of a single parameter of interest given the remaining parameters and the data. For instance, let's consider we have posterior distribution, $f(\underline{\theta} = \{\theta_1, \theta_2, \theta_3\}|\mathbf{X})$, where we can not directly sample from the joint posterior distribution, but we can sample from the full conditional distributions. Thus, we derive and sample from all the full conditional distribution as illustrated in Algorithm 3.2 where $f(\theta_1|\theta_2, \theta_3, \mathbf{X})$ represents the full conditional distribution of θ_1 given everything else.

Algorithm 3.2: Generalized Gibbs sampler Algorithm

Initialize values for $\theta_1^{(t=0)}, \theta_2^{(t=0)}, \theta_3^{(t=0)}$

for $t \leftarrow 1$ **to** T

1. Sample $\theta_1^{(t)} \sim f\left(\theta_1|\theta_2^{(t-1)}, \theta_3^{(t-1)}, \mathbf{X}\right)$
2. Sample $\theta_2^{(t)} \sim f\left(\theta_2|\theta_1^{(t)}, \theta_3^{(t-1)}, \mathbf{X}\right)$
3. Sample $\theta_3^{(t)} \sim f\left(\theta_3|\theta_1^{(t)}, \theta_2^{(t)}, \mathbf{X}\right)$

end

In a Gibb sampler, the acceptance probability is always one because our proposal distribution is the full conditional:

$$g\left(\theta_1|\theta_1^{(t-1)}, \theta_2^{(t-1)}\right) = f\left(\theta_1|\theta_2^{(t-1)}, \mathbf{X}\right),$$

which implies the previous $\theta_1^{(t-1)}$ is not used, thus:

$$g\left(\theta_1|\theta_2^{(t-1)}\right) = f\left(\theta_1|\theta_2^{(t-1)}, \mathbf{X}\right).$$

Therefore, resulting in the acceptance probability is:

$$\begin{aligned}
\alpha_{GIBBS} &= \frac{f\left(\theta_1^*|\theta_2^{(t-1)}, \mathbf{X}\right)}{f\left(\theta_1^{(t-1)}|\theta_2^{(t-1)}, \mathbf{X}\right)} \times \frac{g\left(\theta_1^{(t-1)}|\theta_2^{(t-1)}\right)}{g\left(\theta_1^*|\theta_2^{(t-1)}\right)} \\
&= \frac{f\left(\theta_1^*|\theta_2^{(t-1)}, \mathbf{X}\right)}{f\left(\theta_1^{(t-1)}|\theta_2^{(t-1)}, \mathbf{X}\right)} \times \frac{f\left(\theta_1^{(t-1)}|\theta_2^{(t-1)}, \mathbf{X}\right)}{f\left(\theta_1^*|\theta_2^{(t-1)}, \mathbf{X}\right)} \\
&= 1
\end{aligned}$$

Using the Gibbs sampler, we iteratively sampling though conditional distributions in order to sample from high p-dimensional function; thus converting a p-dimensional problem into p one dimensional problems. We verify that the Gibbs sampler, eventually, algorithm produces samples from the target distribution by demonstrating that detailed balance holds.

Detailed Balance for Gibbs sampler To validate detailed balance holds for the Gibbs sampler algorithm, let us assume our stationary distribution is $f(\theta_1, \theta_2|\mathbf{X})$ for simplicity. Our forward transition in the space follows as:

1. $\theta_1^{(t)} \sim f\left(\theta_1|\theta_2^{(t-1)}\right)$
2. $\theta_2^{(t)} \sim f\left(\theta_2|\theta_1^{(t)}\right)$

thus, our forward transition rate is:

$$\rho\left(\theta_1^{(t)}, \theta_2^{(t)} | \theta_1^{(t-1)}, \theta_2^{(t-1)}\right) = f\left(\theta_2^{(t)} | \theta_1^{(t)}\right) \times f\left(\theta_1^{(t)} | \theta_2^{(t-1)}\right). \tag{3.18}$$

Using detailed balance, Eq. 3.16, and the Gibbs forward transition rate, Eq. 3.18, we have:

$$\rho\left(\theta_1^{(t)}, \theta_2^{(t)} | \theta_1^{(t-1)}, \theta_2^{(t-1)}\right) \times f\left(\theta_1^{(t-1)}, \theta_2^{(t-1)} | \mathbf{X}\right) =$$
$$f\left(\theta_2^{(t)} | \theta_1^{(t)}\right) \times f\left(\theta_1^{(t)} | \theta_2^{(t-1)}\right) \times f\left(\theta_1^{(t-1)}, \theta_2^{(t-1)} | \mathbf{X}\right),$$

which can be rewritten utilizes the relationship between the full conditional, joint, and marginal distribution as:

$$f\left(\theta_2^{(t)} | \theta_1^{(t)}\right) \times f\left(\theta_1^{(t)} | \theta_2^{(t-1)}\right) \times f\left(\theta_1^{(t-1)}, \theta_2^{(t-1)} | \mathbf{X}\right) =$$
$$\frac{f\left(\theta_1^{(t)}, \theta_2^{(t)}\right)}{m\left(\theta_1^{(t)}\right)} \times \frac{f\left(\theta_1^{(t)}, \theta_2^{(t-1)}\right)}{m\left(\theta_2^{(t-1)}\right)} \times f\left(\theta_1^{(t-1)}, \theta_2^{(t-1)} | \mathbf{X}\right),$$

where $m(\bullet)$ represents the marginal distribution and assume the full conditional and marginal distribution are conditional on data, $\mathbf{X}$. After arranging the equation such that:

$$f\left(\theta_1^{(t)}, \theta_2^{(t)}\right) \times \frac{f\left(\theta_1^{(t)}, \theta_2^{(t-1)}\right)}{m\left(\theta_1^{(t)}\right)} \times \frac{f\left(\theta_1^{(t-1)}, \theta_2^{(t-1)}|\mathbf{X}\right)}{m\left(\theta_2^{(t-1)}\right)},$$

we show that:

$$\rho\left(\theta_1^{(t)}, \theta_2^{(t)}|\theta_1^{(t-1)}, \theta_2^{(t-1)}\right) \times f\left(\theta_1^{(t-1)}, \theta_2^{(t-1)}|\mathbf{X}\right) =$$
$$f\left(\theta_1^{(t)}, \theta_2^{(t)}\right) \times f\left(\theta_2^{(t-1)}|\theta_1^{(t)}\right) \times f\left(\theta_1^{(t-1)}|\theta_2^{(t-1)}\right).$$

where $f\left(\theta_1^{(t)}, \theta_2^{(t)}\right)$ represents the target distribution at $\theta_1^{(t)}$ and $\theta_2^{(t)}$, and $f\left(\theta_2^{(t-1)}|\theta_1^{(t)}\right) f\left(\theta_1^{(t-1)}|\theta_2^{(t-1)}\right)$ represents the reversal transition rate, $\rho\left(\theta_1^{(t-1)}, \theta_2^{(t-1)}|\theta_1^{(t)}, \theta_2^{(t)}\right)$.

We describe the reversal rate by flipping the sampling scheme that provide us the forward transition rate as follows:

1. $\theta_2^{(t-1)} \sim f\left(\theta_2|\theta_1^{(t)}\right)$

2. $\theta_1^{(t-1)} \sim f\left(\theta_1|\theta_2^{(t-1)}.\right)$

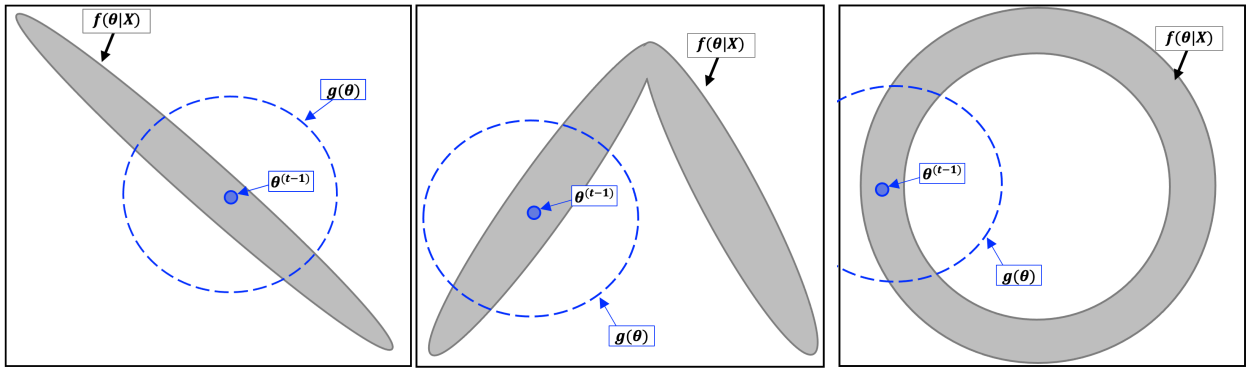
Thus, detailed balance holds, but Gibbs sampler is not a time reversible process because the forward transition rate and reversal transition rate are not equal.

3.4 Markov chain Monte Carlo Ensemble Techniques

Theoretically, we can apply the Metropolis-Hasting Algorithm, Section 3.3.2, to almost any target distribution, $f(\theta|\mathbf{X})$. However, it is rare to find a "good" proposal distribution, $g(\theta)$, that allows us to search the parameter space effectively. For researchers, a typical solution is to adjust a tunable deviance parameter, as demonstrated in Figure 3.13, where a small deviance parameter results in an exceeding slow search of the space and a large deviance results in a very low acceptance rate. In both cases, the algorithm's mixing rate can be prolonged. This section briefly discusses two types of Markov chain Monte Carlo ensemble techniques: the Multi-try Metropolis and the Multiset sampler. Ensemble techniques utilize two or more related but different "models" to improve the results of a method. The Multi-try Metropolis uses an ensemble proposal distribution technique where we propose M θ^* values rather than one. On the other hand, the Multiset sampler is a technique where we sample from an ensemble of posterior distributions based on a mult-set. Neal reviewed the Multi-try Metropolis and Multiset Sample in relation to ensemble theory [74]. In this section, we introduce both of these ensemble techniques before using the algorithms in our Modified Cauchy Net in Chapter 5.

3.4.1 Multi-try Metropolis

In the Metropolis-Hasting Algorithm, our first step is to propose a single new parameter value, θ^*, from a proposal distribution. However, in Figure 3.14, we illustrate several cases when proposing a singular value would result in our slow converges in the Metropolis-Hasting algorithm because we will propose several unsuitable values(outside the parameter space) before proposing a suitable candidate. Thus, we would not explore the parameter space well enough.



(a) Example #1 (b) Example #2 (c) Example #3

Figure 3.14: Three demonstration of highly correlated spaces where the Metropolis-Hasting proposal scheme is not efficient.

The Multi-try Metropolis (MTM) algorithm is a generalized approach to the Metropolis-Hasting Algorithm that generates M proposal values from our proposal distribution rather than a single value to help better explore the space [61, 78]. We initialize at the current location of our parameter of interest, $\theta^{(t-1)}$, as illustrated in Figure 3.15a. Given the current location, Figure 3.15b illustrates the Multi-try Metropolis algorithm proposing M new trials from the proposal distribution $g\left(\theta|\theta^{(t-1)}\right)$ denoted as:

$$\theta^*_{(1:M)} = \{\theta^*_1, \ldots, \theta^*_M\} \sim g\left(\theta^*|\theta^{(t-1)}\right).$$

Among the M trial proposals, we select a single $\theta^*_{(1:M)}$ value with probabilities proportional to

$$w\left(\theta^*_m|\theta^{(t-1)}\right) = f\left(\theta^*_m|\mathbf{X}\right) \times g\left(\theta^{(t-1)}|\theta^*_m\right) \times \lambda\left(\theta^*_m, \theta^{(t-1)}\right) \tag{3.19}$$

where $f\left(\theta_m^*|\mathbf{X}\right)$ represents the target distribution evaluated at each of the M proposals values, $g\left(\theta^{(t-1)}|\theta_m^*\right)$ denotes the transition rate from θ_m^* to $\theta^{(t-1)}$ and $\lambda\left(\theta_m^*, \theta^{(t-1)}\right)$ represents a non-negative symmetric function chosen by the researcher that requires $\lambda\left(\theta_m^*, \theta^{(t-1)}\right) > 0$ whenever $g\left(\theta^{(t-1)}|\theta_m^*\right) > 0$. For simplicity, we picked $\lambda\left(\theta_m^*, \theta^{(t-1)}\right) = 1$. Figure 3.15c illustrates selecting a specific $\theta_{(1:M)}^*$ as our proposal values which we denote as θ_j^*.

In the Metropolis-Hasting algorithm, we balance exploration by calculating the probability we can return to the current location. We hold this ergodic property for the Multi-try Metropolis algorithm by producing a reference set that draws $M-1$ reverse proposal value from our proposal distribution, given that the current location is θ_j^*. That is,

$$\theta_{(1:M-1)}^r \sim g\left(\theta^r|\theta_j^*\right),$$

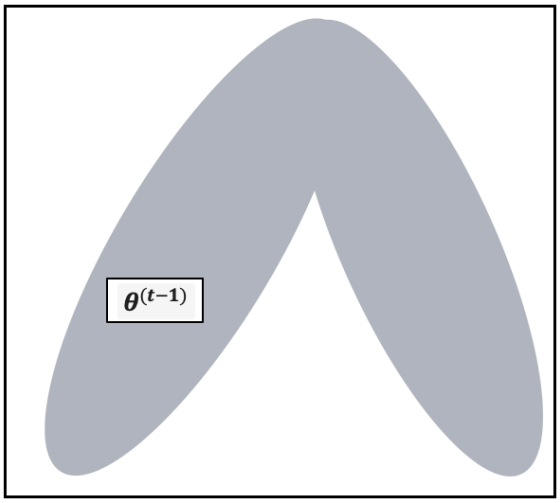
and the M^{th} reversal proposal value is $\theta_M^r = \theta^{(t-1)}$ as illustrated in Figure 3.15d. We accept our proposal value θ_j^* with probability:

$$\alpha_{MTM} = \min\left\{1, \frac{\sum_{m=1}^M w\left(\theta_m^*|\theta^{(t-1)}\right)}{\sum_{m=1}^M w\left(\theta_m^r|\theta_j^*\right)}\right\},$$

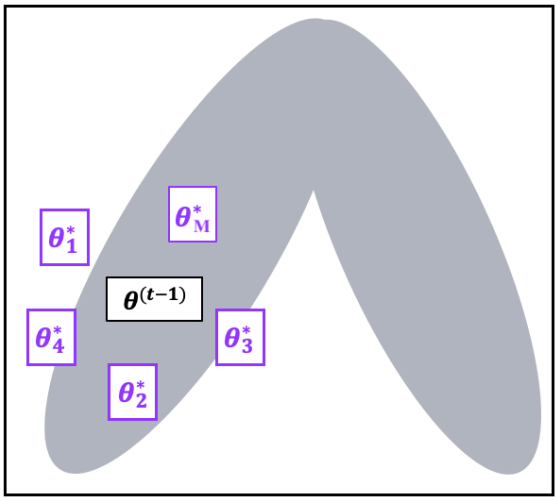
and reject with probability $1-\alpha_{MTM}$ where α_{MTM} denotes the generalized Metropolis-Hasting acceptance ratio. Noted that when M = 1, the Multi-try Metropolis reduces to the Metropolis-Hasting algorithm, and the acceptance ratio is:

$$\begin{aligned}\frac{\sum_{m}^{M=1} w\left(\theta_m^*|\theta^{(t-1)}\right)}{\sum_{m}^{M=1} w\left(\theta_m^r|\theta_j^*\right)} &= \frac{f\left(\theta_1^*|\mathbf{X}\right) g\left(\theta^{(t-1)}|\theta_1^*\right)}{f\left(\theta_1^r|\mathbf{X}\right) \times g\left(\theta_1^*|\theta_1^r\right)} \\ &= \frac{f\left(\theta^*|\mathbf{X}\right) g\left(\theta^{(t-1)}|\theta^*\right)}{f\left(\theta^{(t-1)}|\mathbf{X}\right) \times g\left(\theta^*|\theta^{(t-1)}\right)}\end{aligned}$$

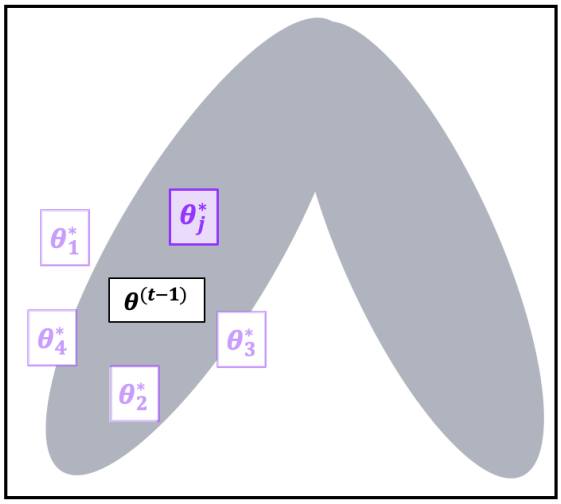
where $\theta_1^* = \theta^*$ because we only had one proposal value and $\theta_1^r = \theta^{(t-1)}$ because we require one of reference set, θ^r, to be $\theta^{(t-1)}$. Algorithm 3.3 summarizes the steps of Multi-try Metropolis.



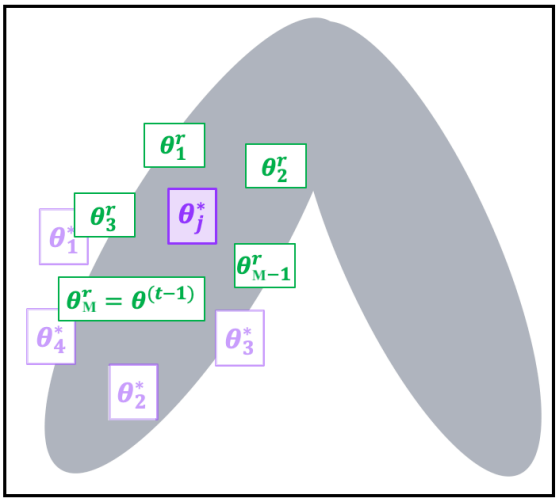
(a) Initialization of starting point.



(b) Sampling M observations from proposal distribution, $\theta^*_{(1:M)}$



(c) Choose θ^*_j from $\theta^*_{(1:M)}$ based on Eq. 3.19.



(d) Sampling reference set, θ^r.

Figure 3.15: Illustration of Multi-try Metropolis sampler.

Algorithm 3.3: Generalized Multi-try Metropolis Algorithm

Initialize values for $\theta^{(t=0)}$

for $t \leftarrow 1$ **to** T

1. PROPOSE M NEW TRIAL VALUES

$\theta^*_{(1:M)} = \{\theta^*_1, \theta^*_2, \ldots, \theta^*_M\} \sim g\left(\theta^*|\theta^{(t-1)}\right)$

2. SAMPLE A $\theta^*_{(1:M)}$ WITH PROBABILITY PROPORTIONAL TO $w(\theta^*_m|\theta^{(t-1)})$

$$w\left(\theta^*_m|\theta^{(t-1)}\right) = f(\theta^*_m|\mathbf{X}) \times g\left(\theta^{(t-1)}|\theta^*_m\right) \times \lambda\left(\theta^*_m, \theta^{(t-1)}\right)$$

where θ^*_j denotes the selected $\theta^*_{(1:M)}$.

3. SAMPLE $M-1$ REFERENCE VALUES

$$\theta^r_{(1:(M-1))} = \left\{\theta^r_1, \theta^r_2, \ldots, \theta^r_{M-1}\right\} \sim g\left(\theta^{(t-1)}|\theta^*_j\right)$$

where $\theta^r_{(M)} = \theta^{(t-1)}$

4. DECISION

$$\theta^{(t)} = \begin{cases} \theta^*_j & \text{with probability } \alpha_{MTM} = \min\left(1, \frac{\sum_{m=1}^M w\left(\theta^*_m|\theta^{(t-1)}\right)}{\sum_{m=1}^M w\left(\theta^r_m|\theta^*_j\right)}\right) \\ \theta^{(t-1)} & \text{with probability } 1-\alpha_{MTM} \end{cases}$$

end

Detailed Balance for Multi-try Metropolis

In Section 3.3.2, we validate Metropolis-Hasting algorithm is a valid sampling scheme by demonstrating that detailed balance. Now, we illustrate the Multi-try Metropolis, the generalized version of the Metropolis-Hasting algorithm, is a valid sampling scheme. Recall for detailed balance, we need Eq. 3.16:

$$\rho\left(\theta^{(t)}|\theta^{(t-1)}\right) f\left(\theta^{(t-1)}\right) = \rho\left(\theta^{(t-1)}|\theta^{(t)}\right) f\left(\theta^{(t)}\right),$$

to hold where:

$$\rho\left(\theta^{(t)}|\theta^{(t-1)}\right),$$

represents the transition rate as illustrated by Eq. 3.15 which constructed by acceptance rate, α_{MTM} multiplied by the proposal distribution, $g(\bullet)$. Without loss of generality, we assume $M = 2$ implying we have two proposal values, θ_1^* and θ_2^*, and have two reference values, θ_1^r and θ_2^r. We let $\theta_2^r = \theta^{(t-1)}$ and θ_2^* be the selected proposal values (θ_j^*) via the weighted probabilities. Thus, our acceptance ratio is:

$$\alpha_{MTM} = \min\left\{1, \frac{w\left(\theta_1^*|\theta^{(t-1)}\right) + w\left(\theta_2^*|\theta^{(t-1)}\right)}{w\left(\theta_1^r|\theta_2^*\right) + w\left(\theta_2^r|\theta_2^*\right)}\right\},$$

where we assume, without loss of generality, that:

$$w\left(\theta_1^*|\theta^{(t-1)}\right) + w\left(\theta_2^*|\theta^{(t-1)}\right) > w\left(\theta_1^r|\theta_2^*\right) + w\left(\theta_2^r|\theta_2^*\right).$$

This assumptions informs us that we move to the proposed value, $\theta_j^* = \theta_2^*$, and our transition rate $\alpha_{MTM} = 1$.

Evaluation of Left-Hand Side Transition Rate $\theta^{(t-1)} \rightarrow \theta_j^*$: The transition rate from $\theta^{(t-1)}$ to θ_j^* is:

$$\rho\left(\theta_j^*|\theta^{(t-1)}\right) = g\left(\theta_1^*|\theta^{(t-1)}\right) g\left(\theta_2^*|\theta^{(t-1)}\right) g\left(\theta_1^r|\theta_j^*\right) \times \left[\frac{w\left(\theta_j^*|\theta^{(t-1)}\right)}{w\left(\theta_1^*|\theta^{(t-1)}\right) + w\left(\theta_2^*|\theta^{(t-1)}\right)}\right] \times 1,$$

where $\left[\frac{w\left(\theta_j^*|\theta^{(t-1)}\right)}{w\left(\theta_1^*|\theta^{(t-1)}\right)+w\left(\theta_2^*|\theta^{(t-1)}\right)}\right]$ is the probability we selected $\theta_j^* = \theta_2^*$. When insert the definition of $w(\bullet)$, Eq. 3.19, and multiply by target distribution at $\theta^{(t-1)}$, we get:

$$\rho\left(\theta^{(t)}|\theta^{(t-1)}\right) f\left(\theta^{(t-1)}\right) = g\left(\theta_1^*|\theta^{(t-1)}\right) g\left(\theta_2^*|\theta^{(t-1)}\right) g\left(\theta_1^r|\theta_j^*\right) \times \left[\frac{f\left(\theta_j^*\right) g\left(\theta^{(t-1)}|\theta_j^*\right) \lambda\left(\theta^{(t-1)}, \theta_j^*\right)}{\sum_{m=1}^{M} f\left(\theta_m^*\right) g\left(\theta^{(t-1)}\right) \lambda\left(\theta^{(t-1)}, \theta_m^*\right)}\right] \times f\left(\theta^{(t-1)}\right) \tag{3.20}$$

Evaluation of Right-Hand Side Transition Rate $\theta_j^* \rightarrow \theta^{(t-1)}$ The transition rate of θ_j^* to $\theta^{(t-1)}$ is:

$$\rho\left(\theta^{(t-1)}|\theta_j^*\right) = g\left(\theta_1^r|\theta_j^*\right) g\left(\theta_2^r|\theta_j^*\right) g\left(\theta_1^*|\theta^{(t-1)}\right) \times \left[\frac{w\left(\theta_2^r|\theta_j^*\right)}{w\left(\theta_1^r|\theta_j^*\right) + w\left(\theta_2^r|\theta_j^*\right)}\right] \left[\frac{w\left(\theta_1^r|\theta_j^*\right) + w\left(\theta_2^r|\theta_j^*\right)}{w\left(\theta_1^*|\theta^{(t-1)}\right) + w\left(\theta_2^*|\theta^{(t-1)}\right)}\right].$$

When insert the definition of $w(\bullet)$, Eq. 3.19, and multiply by target distribution at θ_j^*, we get:

$$\rho\left(\theta^{(t-1)}|\theta_j^*\right) f\left(\theta^{(t)}\right) = g\left(\theta_1^r|\theta_j^*\right) g\left(\theta_2^r|\theta_j^*\right) g\left(\theta_1^*|\theta^{(t-1)}\right) \times \left[\frac{f\left(\theta^{(t-1)}\right) g\left(\theta_j^*|\theta^{(t-1)}\right) \lambda\left(\theta_j^*, \theta^{(t-1)}\right)}{\sum_{m=1}^{M} f(\theta_m^*) g\left(\theta^{(t-1)}|\theta_m^*\right) \lambda\left(\theta^{(t-1)}, \theta_m^*\right)}\right] \times f\left(\theta_j^*\right). \tag{3.21}$$

To compare Eq. 3.20 and 3.21 we rearrange the respective side along with remove the denominator which does not impact our validation of this sampling scheme results in:

$$g\left(\theta_1^*|\theta^{(t-1)}\right) g\left(\theta_2^*|\theta^{(t-1)}\right) g\left(\theta_1^r|\theta_j^*\right) g\left(\theta^{(t-1)}|\theta_j^*\right) f\left(\theta^{(t-1)}\right) \lambda\left(\theta^{(t-1)}, \theta_j^*\right) f\left(\theta_j^*\right) =$$
$$g\left(\theta_1^*|\theta^{(t-1)}\right) g\left(\theta_j^*|\theta^{(t-1)}\right) g\left(\theta_1^r|\theta_j^*\right) g\left(\theta_2^r|\theta_j^*\right) f\left(\theta^{(t-1)}\right) \lambda\left(\theta_j^*, \theta^{(t-1)}\right) f\left(\theta_j^*\right).$$

Recall that $\theta_2^* = \theta_j^*$ and $\theta_2^r = \theta^{(t-1)}$ and $\lambda(\bullet)$ is a symmetric function, thus resulting detailed balance holding for the Multi-try Metropolis.

3.4.2 Multiset sampler

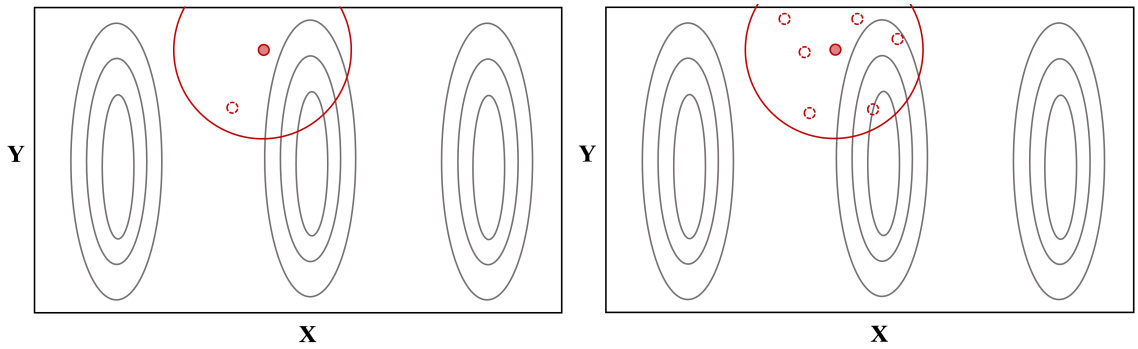
In Section 3.4.1, we briefly discussed the Multi-try Metropolis algorithm to aid in efficiently sampling from the highly-correlated posterior distributions. However, the Multi-try sampling scheme, like the other Markov chain Monte Carlo algorithm we have discussed, can get caught in local modes of multimodal distribution or be limited in exploring high-dimensional discrete spaces. The evolutionary forest algorithm [59] introduced the Multiset sampler in the context of coalescence processes for evolutionary analysis to improve the mixing rate of an algorithm by avoiding getting stuck in local modes. Leman et. al. [58] expanded and described the Multiset sampler to establish a general, multipurpose algorithm for high-dimensional discrete setting and lower dimensional multi-modal bounded examples. Kim and MacEachern [54] extended the Multiset sampler to high-dimensional continuous setting by intertwining the Multiset sampler and importance sampling scheme to make inferences about the posterior distribution using the multiset samples. We dedicate the following section to briefly highlighting the Multiset sampler algorithm since we utilize the algorithm in our proposed Modified Cauchy Net methodology in Chapter 5.

Markov chain Monte Carlo algorithms aim to draw samples from the target distribution to

make inferences. However, if Markov chain Monte Carlo algorithms get stuck in local modes, then the algorithms ineffectively sample and produce biased results. For instance, consider the bivariate, multimodal target distribution, $f(x, y)$, with large valleys between each mode in Figure 3.16. Figures 3.16a and 3.16b demonstrate instances where the Metropolis-Hasting and the Multi-try Metropolis algorithms, respectively, try to sample from the bivariate multimodal target distribution. In both scenarios, we get stuck in the middle local mode and become restricted from exploring the other modes. This characteristic would hold if we started the algorithms in a different region.

Markov chain Monte Carlo algorithms aim to draw samples from the target distribution to make inferences. However, if Markov chain Monte Carlo algorithms get stuck in local modes, then the algorithms ineffectively sample and produce biased results. For instance, consider the bivariate, multimodal target distribution, $f(x, y)$, with large valleys between each mode in Figure 3.16. Figures 3.16a and 3.16b demonstrate instances where the Metropolis-Hasting and the Multi-try Metropolis algorithms, respectively, try to sample from the bivariate multimodal target distribution. In both scenarios, we get stuck in the middle local mode and become restricted from exploring the other modes. This characteristic would hold if we started the algorithms in a different region.

Another option to explore the parameter space is to increase the deviance of the proposal distribution, as demonstrated in Figure 3.17. However, this adjustment does not necessarily help the mixing rate of the algorithm and will cause the algorithm to explore too often and not exploit the promising high-dimensional modes. Therefore, we need an algorithm to exploit the high-density regions to ensure our samples accurately represent our target distribution while simultaneously exploring the remainder of the parameter space.



(a) A demonstration of the MH proposal scheme.

(b) A demonstration of the MTM proposal scheme.

Figure 3.16: An illustration of the limitations of the Metropolis-Hasting and Multi-try Metropolis algorithm in a multimodal target distribution.

The Multiset sampler provides the capability to explore and exploit the parameter space by defining a "new" target distribution containing an ensemble of the "old" target distribution. For example, in Figure 3.16, our goal is to sample from the target distribution, $f(x, y)$; however, we demonstrated that the current Markov chain Monte Carlo schemes are ineffective. Therefore, instead of sampling from $f(x, y)$, the Multiset sampler samples from Y and S, a multiset of K values of the X. That is, the Multiset sampler supports a state vector $(s = \{x_1, \ldots, x_K\}, y)$ consisting of K values of X and one value of Y to define the "new" target distribution as:

$$f_{MSS}\left(\{x_1, \ldots, x_K\}, y\right) = C \sum_{k=1}^{K} f(x_k, y), \tag{3.22}$$

for some normalizing constant C. Multisetting the X space provides the dual capability of exploring and exploiting simultaneously. For instance, consider a multiset of two on the X space, denoted by $s = \{x_1, x_2\}$, with the target distribution of:

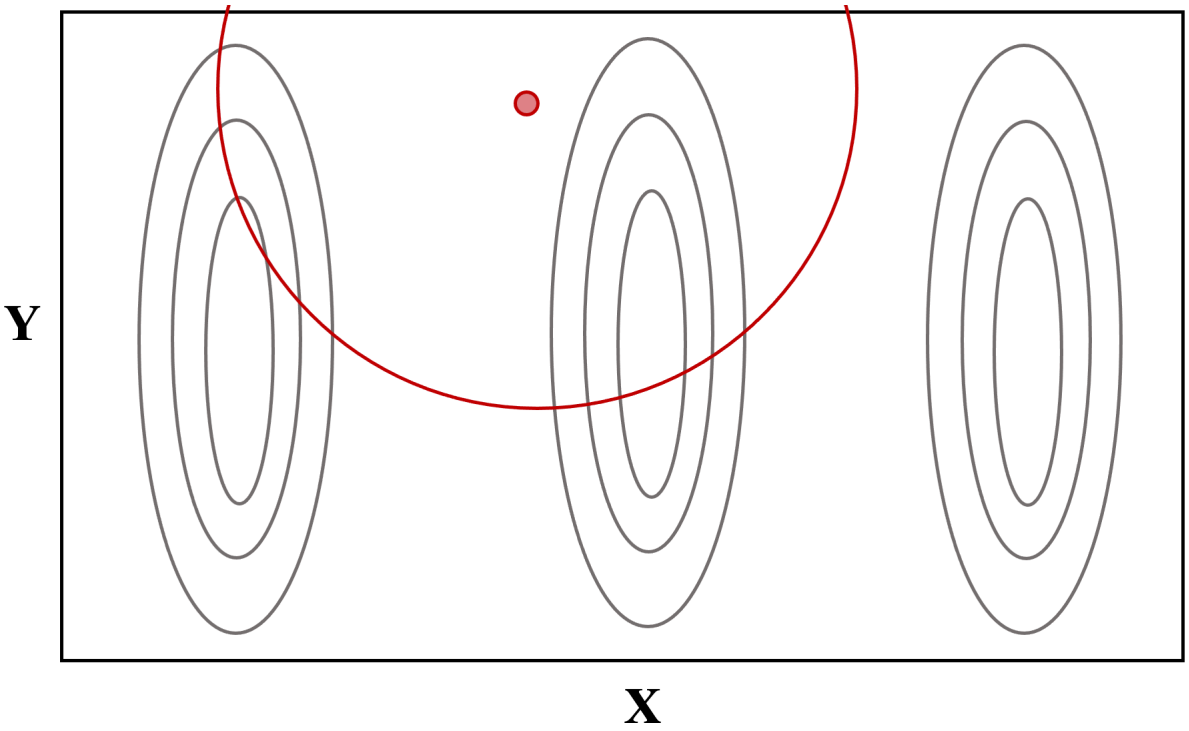
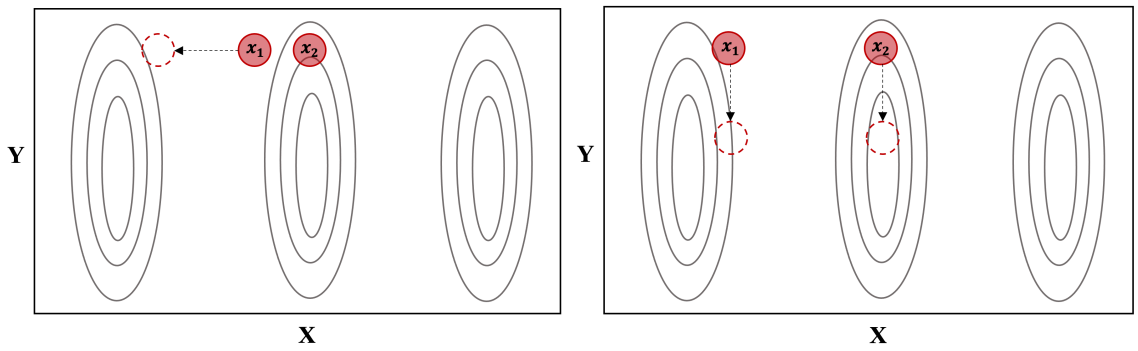


Figure 3.17: Visualization of adjusting the devaince in the proposal distribution to aid the exploration of the Metropolis-Hasting and Multi-try Metropolis algorithm.

$$f_{MSS}\left(\{x_1, x_2\}, y\right) \propto f(x_1, y) + f(x_2, y).$$

to evaluate $f(x, y)$. Figure 3.18a represents our sampler is in a current state marked by the closed circles where we propose moving one of the x values, the open, dashed circle. The algorithm may accept the proposal x value because $f_{MSS}\left(\{x^*, x_2\}, y\right)$ is smaller than $f_{MSS}\left(\{x_1, x_2\}, y\right)$ by a factor of about a half. Figure 3.18b illustrates accepting the proposal value, x^*, and proposing a values in the Y space.

In addition to the Multiset sampler's ability to explore the space, the algorithm is relatively easy to implement without careful tuning of choosing the proposal distribution. Note that the Multiset sampler is a Metropolis-within-Gibbs sampler on (S, Y) and not (X, Y), which enables the Multiset sampler to work because one of the x in s can move across the valleys



(a) Current state of Multiset sampler and proposal value, x^*.

(b) Acceptance of proposal value and proposing move in Y space.

Figure 3.18: An illustration of the Multiset sampler algorithm steps a multimodal target distribution.

while the other x stays in a local mode.

For proposing a new multiset state, s^* given a current multiset, $s = \{x_1, \ldots, x_j, \ldots, x_K\}$, the Multiset sampler algorithm randomly selects, with equal probability, $j \in \{1, \ldots, K\}$. Next, the algorithm generates a x_j^* using a proposal distribution, $g(x_j^*|x_j)$, to propose a new multiset state:

$$s^* = \{x_1, \ldots, x_j^*, \ldots, x_K\}.$$

We utilized the Metropolis-Hasting acceptance rule, Eq. 3.13, where the target distribution is $f_{MSS}(S, Y) \propto \sum_{k=1}^{K} f(x_k, y)$. That is, for our example in Figure 3.18 we accept our proposal value x^* with probability:

$$\alpha_{MSS} = \min\left(1, \frac{[f(x^*, y) + f(x_2, y)] \times g\left(x_1|x^*\right)}{[f(x_1, y) + f(x_2, y)] \times g\left(x^*|x_1\right)}\right). \tag{3.23}$$

We provide the general Multiset sampler algorithm in Algorithm 3.4. After accepting or rejecting s^* value, the algorithm continues by proposing y^* from its respective proposal distribution and deciding with probability:

$$\alpha = \min\left(1, \frac{[f(x_1, y^*) + f(x_2, y^*)] \times g\left(y|y^*\right)}{[f(x_1, y) + f(x_2, y)] \times g\left(y^*|y\right)}\right).$$

to accept or reject the proposed y^*.

Algorithm 3.4: Generalized Multiset sampler Algorithm for $f(X, Y)$ with a multiset of size K on X.

Initialize values for $s^{(t-1)} = \{x_1, \ldots, x_j, \ldots, x_K\}$ where x_j represents the j^{th} observation for the parameter interest.

for $t \leftarrow 1$ **to** T

1. SAMPLE $j \in \{1, \ldots, K\}$ WITH EQUAL PROBABILITY
2. SET $s_j^* = s^{(t-1)}$ WHERE $x_j = x_j^*$ AND

$$x_j^* \sim g\left(x_j^* | x_j\right)$$

3. DECISION

$$s^{(t)} = \begin{cases} s^* & \text{with probability } \alpha_{MSS} = \min\left(1, \frac{\left[\sum_{x \in s^*} f(x,y)\right] \times g\left(x_j | x_j^*\right)}{\left[\sum_{x \in s} f(x,y)\right] \times g\left(x_j^* | x_j\right)}\right) \\ s^{(t-1)} & \text{with probability } 1 - \alpha_{MSS} \end{cases}$$

end

Chapter 4

Robust Bayesian Regression (RBR)

Principal Component Analysis operates under normality assumptions, which assumes exponentially weighted ("light") tails for explaining uncertainties. Robust Bayesian Regression (RBR) is a Bayesian approach to relax the normality assumption and decreases the influence of the anomalous observations by using the model's tunable hyper-parameter. The Robust Bayesian Regression model follows from a multivariate normal scale mixture model [4] denoted as:

$$\underline{x}_i \sim \text{MVN}\left(\underline{\mu}, \gamma_i^{-1}\Sigma_{PxP}\right), \tag{4.1}$$

for $i = 1, \ldots, N$ with the conjugate multivariate normal distribution, Eq. 2.15, on $\underline{\mu}$ and the conjugate inverse-Wishart prior [6, 36] on Σ with degrees of freedom ψ and scale matrix Ω:

$$\Sigma_{P\times P} \sim \text{IW}(\psi, \Omega^* = (\psi - P - 1)\Omega), \tag{4.2}$$

where $\Omega = \mathbf{I}_P$. For each individual γ_i value, we place the following gamma prior:

$$\gamma_i \sim \text{Gamma}(\eta/2, \eta/2), \tag{4.3}$$

where η tunes the tail behavior of the resulting error structure. For example, integer values ($\eta = \{1, 2, 3, ...\}$) produce t-distributed errors with various η degrees of freedom [33, 36]. We select $\eta = 1$ to invoke a Cauchy distribution [97]. Cauchy distributions are often used because of the naturally heavier-tail behavior, compared to a normal or Laplace distribution, which allows for robust parameter estimation [16, 51]. Figure 4.1 demonstrates the difference between the tail behavior of the normal, Eq. 1.5, and the Cauchy, Eq. 4.1, model assumptions. We illustrate the difference in tail behavior (green line) through computing the log of the absolute relative distance (AD) which is defined as:

$$\text{AD} = \log\left(\left|\frac{f_{Normal}(X|\mu = 0, \sigma^2 = 1) - f_{Cauchy}(X|0, 1)}{f_{Normal}(X|\mu = 0, \sigma^2 = 1)}\right|\right). \tag{4.4}$$

Under this formulation, the Bayesian estimator adjusts the likelihood based estimator using a regularization (smoothing) penalty through the hyper-parameters ψ and Ω and induces a heavy-tailed Bayesian estimator through the multiplicity term, γ_i, to Σ which describes the "outlierness" of an observation. The remainder of this chapter discusses the impact of the penalty parameters, implementation of the Gibbs sampler, and prediction of a new observation.

4.1 Penalization Parameters

As the number of modes, k, impacts Principal Component Analysis, the hyper-parameters in Eq. 4.2 and 4.3 (ψ, Σ, η) impact the Robust Bayesian Regression. Under the Bayesian paradigm, the prior distribution plays a significant role in constructing the posterior distribution. The prior distribution choices enable "good" posterior properties for estimation and regularization of the parameter space. By "good" posterior properties, we mean that

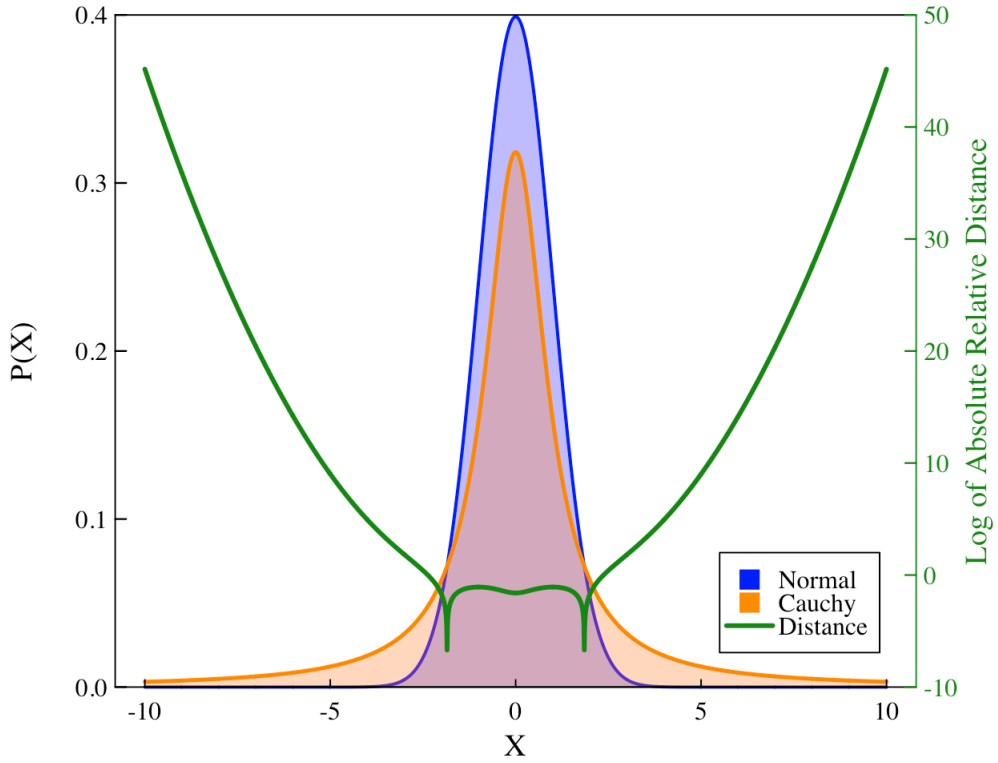


Figure 4.1: Illustrating the difference in tail behavior between a normal and Cauchy distribution. P(X), left axis, represents the respective probability density whereas the right axis illustrates the large difference in tail behavior between the two distribution.

our inference techniques should be able to utilize the posterior to estimate the parameters. A common prior choice of Bayesians analysis is using a conjugate prior distribution to induce these properties and analytical/computational reasons. A prior distribution is conjugate prior if the posterior and prior distribution follows the same distributional family. For instance, if the posterior and prior distributions follow a Normal distribution where the parameterizations may differ. Bayesian methods inherently adopt regularization because the prior distribution allows us to penalize different parameter space areas. The inherent regularization helps Bayesian techniques perform better compared to classical methods in low sample size scenarios.

For all of our proposed methods in this thesis, we use conjugate priors unless otherwise stated. In the remainder of this section, we demonstrate the impact of the hyper-parameters in the inverse-Wishart prior distribution on covariance and the Gamma prior distribution place on the multiplicity term.

4.1.1 Understanding the Inverse Wishart Hyper-parameters

The conjugate prior for covariance, Σ, is an inverse-Wishart distribution as denoted in Eq. 4.2 with sampling density:

$$p_{inverse-Wishart}\left(\Sigma_{P\times P}|\Omega^*,\psi\right) = \frac{\left|\Omega^*\right|^{\frac{\psi}{2}}}{2^{\frac{\psi M}{2}}\Gamma_P(\frac{\psi}{2})}|\Sigma|^{-\frac{\psi+P+1}{2}}e^{-\frac{1}{2}\text{tr}(\Omega^*\Sigma^{-1})},$$

where Γ_p represents a multivariate Gamma function. Under of specified inverse-Wishart prior distribution, the *a-priori* expected value of of the sensors relationship is

$$\mathbb{E}\left[\Sigma\right] = \Omega, \text{if } \psi > P+2.$$

Setting the scale matrix to $\Omega^* = (\psi - P - 1)\Omega$ enables our method to interpret the parameter more intuitive. For a generalized formula of the inverse-Wishart distribution, see Appendix D.1 for more details. The positive-definite scale matrix, Ω, has the same dimensionality of Σ and provides a "centering" of our prior beliefs about the relationship between the sensors. By setting Ω to the identity matrix, we impose an *a-priori* idea that the sensors are independent of each other to induce estimated covariance matrices less sensitive to outlying tendencies. In case study applications, such as the Virginia Tech Wind Tunnel, the Ω can represent specified uncertainty values provided by the engineers about their sensor systems.

The positive degree of freedom scalar, ψ, represents the degree to which the likelihood function (as a function of sensor quantity and associated true signals) overwhelms our centered beliefs, Ω. For example, when ψ is large, compared to the number of sensors, we require strong empirical signals (i.e., more data) to outweigh the prior assumptions of sensor independence. On the other hand, when ψ is small, there is a less "smoothed" behavior of the RBR covariance estimate, and it is potentially more sensitive to data anomalies. Thus, small ψ values heavily rely on the historical data for the covariance estimation, while larger ψ values favor the prior belief, Ω. Figure 4.2 illustrates the effect ψ has on the covariance estimate when we have a small case of fifty (50) observations for three (3) sensors. Notice that as ψ becomes larger the resulting Robust Bayesian Regression Σ estimate converges towards Ω, the identity matrix. For the *a-priori* expected value to exist, we need $\psi \geq P + 2$.

4.1.2 Understanding the γ_i parameter

Under our model assumption, Eq. 4.3, we utilize the prior distribution for γ_i as a Gamma distribution with a sampling density:

$$f_{\text{Gamma}}\left(\gamma_i | \frac{\eta}{2}, \frac{\eta}{2}\right) = \frac{\left(\frac{\eta}{2}\right)^{\eta/2}}{\Gamma\left(\frac{\eta}{2}\right)} \gamma_i^{\frac{\eta}{2}-1} e^{-\frac{\eta\gamma_i}{2}},$$

where the expectation and variance of γ_i before seeing data is:

$$\mathbb{E}\left[\gamma_i\right] = \frac{\frac{\eta}{2}}{\frac{\eta}{2}} = 1; \quad \mathbb{V}\left[\gamma_i\right] = \frac{\frac{\eta}{2}}{\left(\frac{\eta}{2}\right)^2} = \frac{2}{\eta},$$

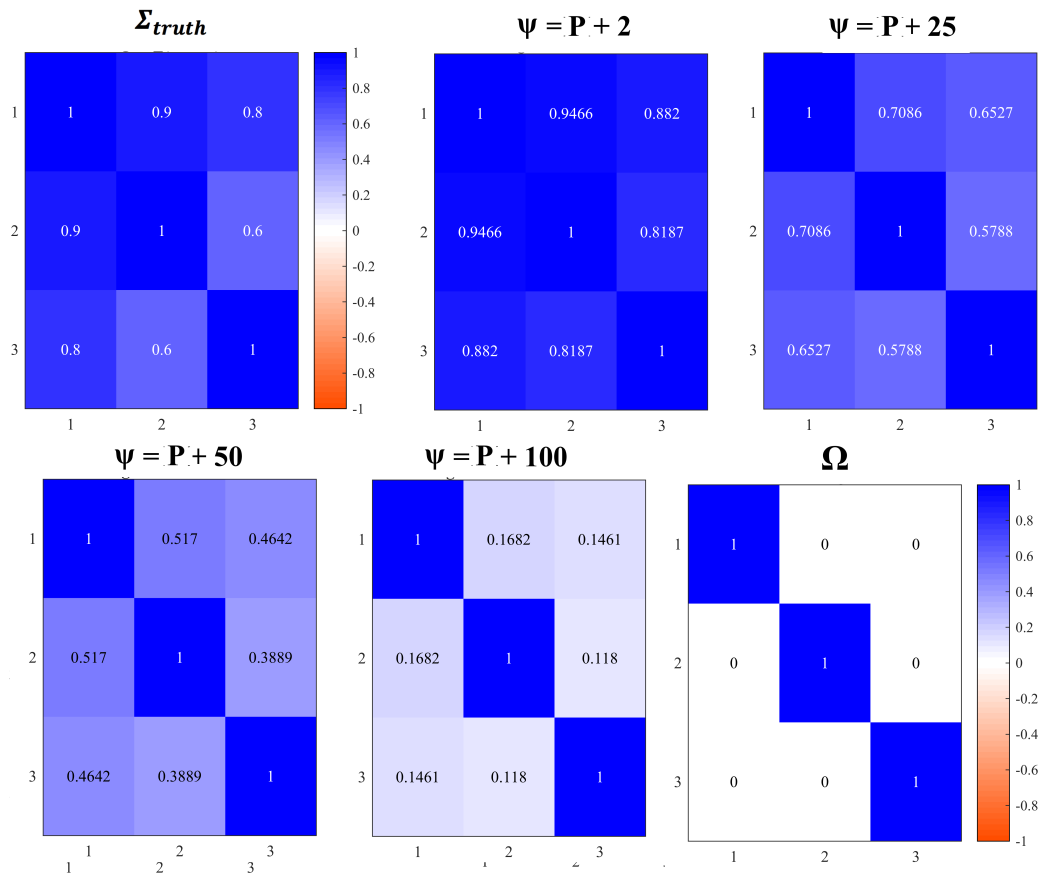


Figure 4.2: Illustrates the effect of ψ on Robust Bayesian Regression estimator. The size of ψ increases from left-top to right-bottom.

respectively. As previously statement, we choose $\eta = 1$ to induce a Cauchy distribution; thus, setting the hyper-parameters of Eq. 4.3 to $\frac{1}{2}$. Under our parameterization, the $\mathbb{E}[\gamma_i] = 1$ signifies that on average, prior to seeing data, we would expect $\gamma_i = 1$ which translates our Robust Bayesian Regression model assumption, Eq. 4.1 , to the normal data assumption, Eq. 1.5. This assumption is practically reasonable because we do not automatically assume that our data contains many anomalous observations.

Writing out the expected variance, $\mathbb{V}[\gamma_i] = \frac{2}{\eta}$, helps to establish that smaller η values induce

heavier-tailed t-distributions compared to large η values which produce more "Gaussian"-like distributions. Figure 4.3 aids in visualizing this concept by plotting the sample mean of 100 random Gamma value under two different parameterization ($\eta = 1, \eta = 10$) for 10,000 datasets. The key aspect to notice is that the large η value (blue-shaded histogram) produces a tighter distribution around one (1) compared to the small η value (orange-shaded histogram). With the tighter distribution around one (1), we are more likely to sample from a γ_i closer to one, thus more likely to sample from our normal data assumption, Eq. 1.5. In contrast, the smaller η values vary more from the expected mean, thus enabling our Robust Bayesian Regression model to produce a heavier-tailed distribution such as the Cauchy.

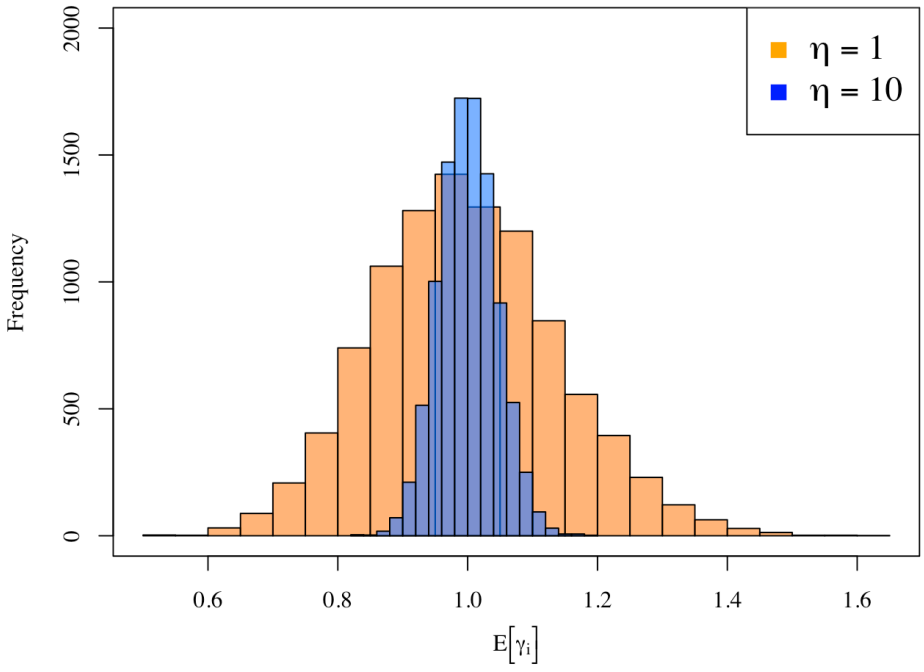
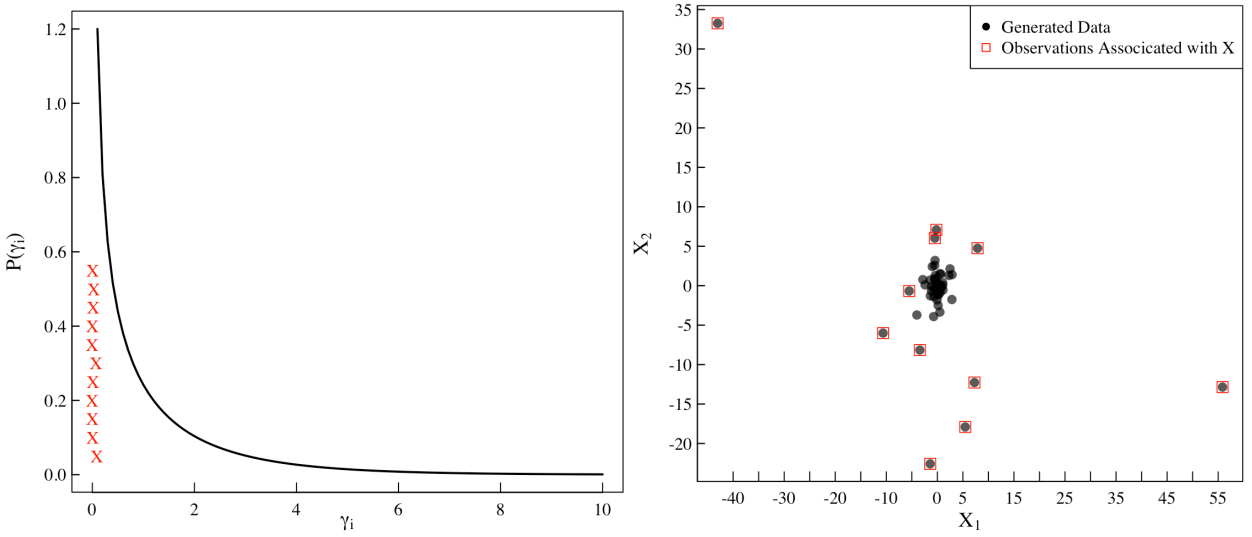


Figure 4.3: Plot of the sample mean of 100 random Gamma value under two different parameterization ($\eta = 1$ (orange), $\eta = 10$ (blue)) for 10,000 datasets to demonstrate the effect η has on inducing heavy-tailed distributions.

Thus far, we discuss the prior distribution's impact and our a-priori belief for understanding the multiplicity parameter, γ_i. Now, we segway into understanding how the γ_i^{-1} in the RBR

model assumption, Eq. 4.1, generates data from Cauchy distribution under our specific model assumptions.

Figure 4.4a illustrates the density for γ_i under the parameterization $\eta = 1$. When we generated $\gamma_i \sim \text{Gamma}\,(1/2, 1/2)$, from Figure 4.4a, we see there is a higher probability of selecting small values of γ_i such as those between the interval of $(0, 1]$. Due to the inversion of gi small values correspond to generating observations, $\underline{x}$, from large covariance structures. For instance, let's consider we know the true covariance structure is $\Sigma = \mathbf{I}$ and we sample $\gamma_i = 0.25$. Then, the resulting $\underline{x}_i$ would follow a normal distribution with a covariance structure of $4 * \mathbf{I}$; thus giving the $\underline{x}_i$ a chance to be sampled the tails of the distribution. To conceptually visualize this concept, Figure 4.4b portrays a generated dataset from the Robust Bayesian Regression model with fifty (50) observations where $\underline{\mu} = \underline{0}$ and $\Sigma = I_2$.The red boxes in Figure 4.4b correspond to their respective generated γ_i value denoted by X in Figure 4.4a. It is important to note that not all small values of γ_i become observations in the tail.



(a) Illustration of the $\gamma_i \sim \text{Gamma}\left(\frac{1}{2}, \frac{1}{2}\right)$ prior where the X correspond to red boxes in Figure 4.4b

(b) A 2D toy example of fifty (50) observations generated from Eq. 4.1 with $\underline{\mu} = \underline{0}$ and $\Sigma = I_2$. The red boxes correspond to X in Figure 4.4a.

Figure 4.4: Demonstrating how the γ_i induce Cauchy distribution.

4.2 Algorithm for Robust Bayesian Regression Inference

In the Bayesian paradigm, we develop hierarchical models through defining the sampling distribution and prior belief. Subsequently, we construct the joint posterior distribution of the model's parameters which serves as the foundation for inference. Under the RBR model assumption, Eq. 4.1, and priors, Eq. 4.2 and 4.3, the resulting joint posterior distribution is

$$f(\underline{\mu}, \Sigma, \gamma_i | \mathbf{X}, \psi, \Omega) \propto \left[\prod_{i=1}^{N} \gamma_i^{\frac{P+1}{2}-1} \right] |\Sigma|^{-\left(\frac{N+\psi+P+1}{2}\right)} e^{\frac{-1}{2}\text{tr}(\Omega^* \Sigma^{-1})} \times e^{-\frac{\gamma_i}{2} - \frac{1}{2}\sum_{i=1}^{N} (\underline{x}_i - \underline{\mu})' \gamma_i \Sigma^{-1} (\underline{x}_i - \underline{\mu})}. \tag{4.5}$$

Since we cannot directly sample from Eq. 4.5, we utilize a particular Markov chain Monte Carlo algorithm to jointly estimate the parameters called a Gibbs sampler [17, 93]. Recall the Gibbs sampler is an iterative procedure that draws samples from the parameter's full conditional distributions to jointly sample from the posterior distribution. The full conditional distribution is the distribution of a single parameter of interest given the remaining parameters and the data. For the Robust Bayesian Regression algorithm we must derive and sample from all the full conditional distribution as illustrated in Algorithm 3.1.

Since the Gibbs sampler is an Markov chain Monte Carlo approach, we must iterate for some large number, T, and remove the first (B) iterations that represent the burn-in effect. In the Robust Bayesian Regression model, we can directly sample from our full conditional distribution, displayed in Algorithm 4.1, because we used conjugate priors, which induce good posterior properties for estimation. However, if we could not sample from the full conditionals, we could have implement the Metropolis-Hasting Algorithm.

Algorithm 4.1: Gibbs sampler for Robust Bayesian Regression

Initialize values for $\Sigma_{(t=0)}, \underline{\mu}^{(t=0)}, \gamma_i^{(t=0)}$

for $t \leftarrow 1$ **to** T

1. Sample

for $i \leftarrow 1$ **to** N

$$f(\gamma_i^{(t)}|\Sigma_{(t-1)}, \underline{\mu}^{(t-1)}) \sim \text{Gamma}\left(\frac{P+1}{2}, \frac{1}{2}\left[1 + (\underline{x}_i - \underline{\mu}^{(t-1)})'\Sigma_{(t-1)}^{-1}(\underline{x}_i - \underline{\mu}^{(t-1)})\right]\right)$$

end

2. Sample

$$f(\Sigma_{(t)}|\underline{\gamma}^{(t)}, \underline{\mu}^{(t-1)}) \sim \text{inverse-Wishart}\left(N + \psi, \Omega^* + \sum_{i=1}^{N}\left(\underline{x}_i - \underline{\mu}^{(t-1)}\right)\left(\underline{x}_i - \underline{\mu}^{(t-1)}\right)'\gamma_i^{(t)}\right)$$

3. Sample

$$f(\underline{\mu}^{(t)}|\Sigma_{(t)}, \underline{\gamma}^{(t)}) \sim \text{MV. Normal}\left(\left[\sum_{i=1}^{N}\gamma_i^{(t)}\Sigma_{(t)}^{-1}\right]^{-1}\left[\sum_{i=1}^{N}\gamma_i^{(t)}\Sigma_{(t)}^{-1}\underline{x}_i\right], \left[\sum_{i=1}^{N}\gamma_i^{(t)}\Sigma_{(t)}^{-1}\right]^{-1}\right)$$

END

4.3 Prediction

Thus far, we have created a posterior distribution, Eq. 4.5, and used a Gibb sampler approach to estimates the parameters, $\underline{\mu}, \Sigma$, and γ_i. Using the RBR method, we now address predicting sensor estimates and their uncertainty values to evaluate if the next (or incoming) observation, $\underline{x}_{new}$, is anomalous or not. Using Eq. 4.5, we create a posterior predictive distribution that describes the probability of observing a new sensor measurement and its uncertainty, given the observed training data, $\mathbf{X}_{N\times P}$. Under our model assumptions, $\underline{x}_{new}$ follows a multivariate Cauchy distribution. Unlike multivariate normal distributions that have expected means and expected covariance matrices, multivariate Cauchy distributions do not have expected means and expected covariance matrices. Therefore, we utilize the posterior predictive median for predicting sensor measurements and the 90% Highest Posterior Density (HPD) region to represent measurement uncertainty.

Chapter 5

Modified Cauchy Net (MCN)

The Robust Bayesian Regression monitoring system constructs weighted average estimators for the mean and covariance structure by introducing a multiplicity term, γ_i, on the Σ to adjust for the influence of anomalies in the training dataset. However, while introducing γ_i benefits our objective to develop a robust sensor monitoring system, it has limitations. Since γ_i controls whether an entire observed run is anomalous or not, we cannot create reliable covariance structures because our estimation scheme down weighting the whole run instead of just the anomalous sensors. We propose a k-finite mixture model [79] to decompose the anomalous and non-anomalous sensors' observations into their respective groups to alleviate the RBR limitation. We extend the mixture model methodology to incorporate a non-local distribution with Cauchy-like tails as the component to induce robustness and reduce classification errors [51, 52]. In Section 2.4.2, we discussed generalized Gaussian finite mixture models which we now utilize with non-local distribution, Section 5.1, to motivate our proposed Modified Cauchy Net (MCN) in Section 5.2.

5.1 Non-Local Distribution

Our proposed Modified Cauchy Net utilizes a two-component mixture model to compartmentalize the signal (non-anomalous sensors) component or noise (anomalous sensors and extraneous noise) component and develop more reliable estimators. In Chapter 1, we estab-

lished a non-anomalous observations follow a multivariate normal distribution, thus Eq. 1.5 represent the signal components. In this section, we discuss the reasoning behind choosing a non-local heavy-tailed distribution to represent the anomalous component over a Cauchy distribution.

To model the anomalous observations, we utilize a non-local distribution with heavy tails introduced by [52]. In [51, 52], the authors introduced the non-local distribution for hypothesis testing purposes; however, we utilize these distributions in a mixture modeling framework to classify anomalous sensor readings. We use a non-local distribution rather than a Cauchy distribution because Cauchy and normal densities significantly overlap in their central regions. This large overlap in densities results in tangled inferences between non-anomalous and anomalous distributions. We could not distinguish an anomalous observation from non-anomalous because the anomalous behavior acts similarly to non-anomalous sensor reading. In Figure 5.1, we illustrate the differences between a Cauchy density (orange) and non-local density (blue) compared to standard normal density (black). For instance, observations between negative one (-1) and positive one (+1) would be challenging to classify as anomalous or non-anomalous when differentiating between Cauchy and normal due to both distributions place a significant amount of mass in that area. In contrast, it is easier to classify observation as anomalous or non-anomalous when differentiating between non-local and normal.The non-local distribution enables us to place little to no mass in the central region of normal distribution under specific parameterizations.

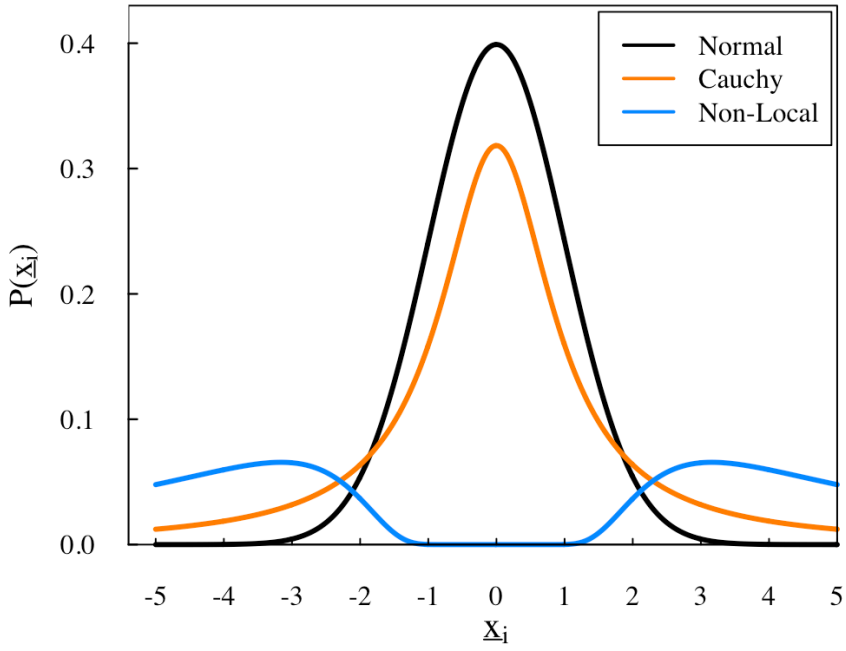


Figure 5.1: Comparison of densities for mixture model components.

The authors in [51, 52] refer to the particular non-local distribution illustrated in Figure 5.1 as the inverse moment distribution with parameter, τ, where τ is a tunable parameter that controls the width of the central trough. If the data are standardized, Johnson et. al [51, 52] recommend $\tau = 1$. However, they [51, 52] introduced the non-local distributions for hypothesis testing purposes; however, we utilize these distributions in a mixture modeling framework to classify anomalous sensor readings. Thus, after standardizing our data, we choose the τ parameter such that we place little to no mass in the middle of the normal (non-anomalous) distribution, as seen in Figure 5.1.

5.1.1 Generalized Mixture Model with Non-Local

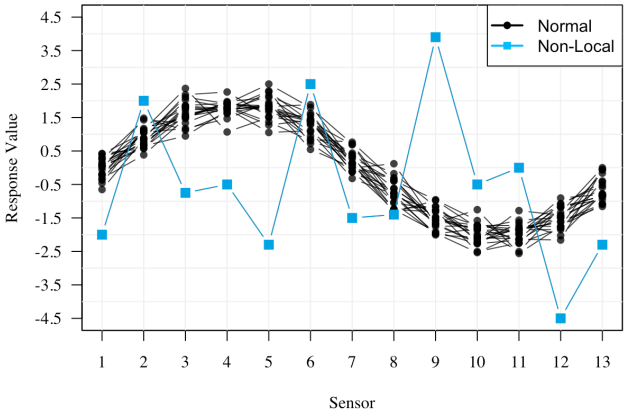
If we directly applied the non-local distribution to our generalize mixture model, Eq. 2.16, it would result in the following model:

$$\underline{x}_i \sim \pi_1 f_{\text{MV. Normal}}(\underline{\mu}, \Sigma) + (1 - \pi_1) f_{\text{Non-Local}}(\tau), \tag{5.1}$$

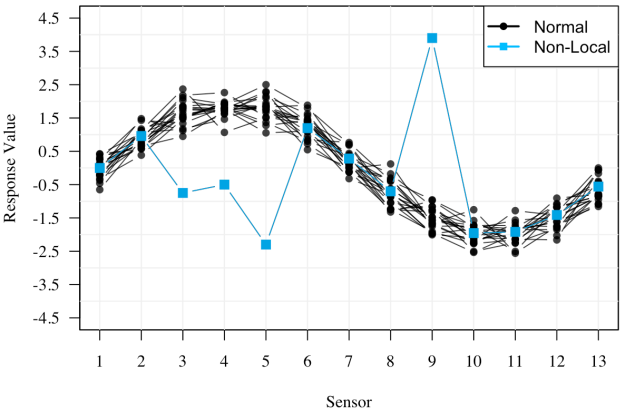
where $f_{\text{MV. Normal}}(\underline{\mu}, \Sigma)$ denotes the multivariate normal sampling distribution 1.5 and $f_{\text{Non-Local}}(\tau)$ denotes the non-local, inverse moment distribution presented by [51] with sampling distribution:

$$f_{\text{Non-local}}\left(x|\mu, \tau, \nu, \kappa\right) = \frac{\kappa \tau^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2\kappa}\right)} \left[(x-\mu)^2\right]^{-\frac{\nu+1}{2}} \times e^{-\left(\frac{(x-\mu)^2}{\tau}\right)^{-k}} \tag{5.2}$$

for $\tau, \nu, \kappa > 0$ and m represents the centering parameter and τ controls the trough size. The disadvantage under this model is similar to that of the Robust Bayesian Regression in that an entire observation (or run) is either entirely classified as anomalous or non-anomalous, as seen in Figure 5.2a. While it can be practical for entire experimental runs to result in erroneous readings, it is more common for a single sensor or a group of sensors to be erroneous in an experimental run as demonstrated in Figure 5.2b. Thus, Eq. 5.1 represents a particular anomalous case and not a generalize anomalous case. We propose our Modified Cauchy Net model to account for intermittent mixed signals.



(a) An illustration of generalized mixture model with non-local distribution under Eq. 5.1.



(b) An illustration of Modified Cauchy Net.

Figure 5.2: Illustration between the generalized mixture model and our Modified Cuachy Net.

5.2 Modified Cauchy Net model

To induce behaviors demonstrated in Figure 5.2b, we develop our proposed Modified Cauchy Net model (MCN) under the following sampling scheme. We classify each sensor as anomalous or non-anomalous under a Bernoulli distribution with the probability of a non-anomalous sensor is ρ. That is, s_{ij} is an indicator of the j^{th} sensor in the i^{th} run, such that:

$$s_{ij} = \begin{cases} 1 & \text{non-anomalous with probability } \rho \\ 0 & \text{anomalous with probability } 1-\rho \end{cases}. \tag{5.3}$$

Given the sensor's label classification, Eq. 5.3, we sample the non-anomalous sensors, (i.e.,$\{s_{i\bullet} = 1\}$) from the following multivariate normal:

$$\underline{x}_{i,j\subset\{s_{i\bullet}=1\}} \sim \text{MVN}\left(\underline{\mu}_{\{s_{i\bullet}=1\}}, \Sigma_{\{s_{i\bullet}=1\}}\right), \tag{5.4}$$

where $\underline{\mu}_{\{s_{i\bullet}=1\}}$ and $\Sigma_{\{s_{i\bullet}=1\}}$ represent a subset of the full mean vector and covariance structure corresponding with the non-anomalous sensor's labels. By utilizing the covariance subset, our model preserves the sensor systems relationship. We independently sample the anomalous sensors (i.e.,$\{s_{i\bullet}=0\}$) from a non-local distribution:

$$\underline{x}_{i,j\subset\{s_{i\bullet}=0\}} \sim NL(\tau), \tag{5.5}$$

where NL denotes the non-local inverse moment distribution. For an illustration of sampling scheme, reference Appendix D.2. Through understanding the sampling scheme, we construct the Modify Cauchy Net likelihood as:

$$\mathcal{L}(\underline{\mu}, \Sigma, \mathbf{S}|\mathbf{X}) \propto \prod_{i=1}^{N}\left[\rho^{\sum_{j=1}^{P} s_{ij}=1}\left|\Sigma_{\{s_{i\bullet}=1\}}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\left[\left([\underline{x}_{i\bullet}-\underline{\mu}]\circ s_{i\bullet}\right)'\Sigma^{-1}\left([\underline{x}_{i\bullet}-\underline{\mu}]\circ s_{i\bullet}\right)\right]}\right] \times \prod_{i=1}^{N}\left[(1-\rho)f_{NL}(\tau)\right]^{\sum_{j=1}^{P}[1-s_{ij}]}, \tag{5.6}$$

where $\circ$ represents the hadamard product. We shorthand the non-local distribution, $f_{NL}(\tau)$ because we are not interested in estimating any of the non-local parameters. The hadamard product produces a $P \times 1$ vector with zeros for anomalous observations. For instance, if we consider a four-dimensional sensor structure:

$$[x_{i\bullet} - \underline{\mu}] = \begin{bmatrix} x_{i1} - \mu_1 \\ x_{i2} - \mu_2 \\ x_{i3} - \mu_3 \\ x_{i4} - \mu_4 \end{bmatrix}; \quad s_{i\bullet} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}; \quad [x_{i\bullet} - \underline{\mu}] \circ s_{i\bullet} = \begin{bmatrix} x_{i1} - \mu_1 \\ 0 \\ x_{i3} - \mu_3 \\ 0 \end{bmatrix},$$

the hadamard product allows us to retain the dimensionality which aids in constructing conditional posterior distributions for the Metropolis-Hasting algorithm. By carrying out the product over the observations $\left(\text{i.e.}, \prod_{i=1}^{N}\right)$, we simplify the Modified Cauchy Net likelihood in Eq. 5.6 to:

$$\mathcal{L}(\underline{\mu}, \Sigma, S|\mathbf{X}) \propto \rho^{\sum_{i=1}^{N}\sum_{j=1}^{P} s_{ij}} \left[\prod_{i=1}^{N} |\Sigma_{\{s_{i\bullet}=1\}}|^{-\frac{1}{2}}\right] e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([\underline{x}_{i\bullet}-\underline{\mu}]\circ s_{i\bullet}\right)'\Sigma^{-1}\left([\underline{x}_{i\bullet}-\underline{\mu}]\circ s_{i\bullet}\right)\right]} \times \prod_{i=1}^{N} [(1-\rho) f_{NL}(\tau)]^{\sum_{j=1}^{P}[1-s_{ij}]}. \tag{5.7}$$

To perform Bayesian inference, we utilize the same conjugate prior for $\underline{\mu}$, Eq. 2.15, and Σ, Eq. 4.2, as in the Robust Bayesian Regression inference to preform inferences for the Modified Cauchy Net. The posterior distribution under this framework is:

$$f\left(\underline{\mu}, \Sigma, \rho|\mathbf{X}\right) \propto \left[\rho^{\sum_{i=1}^{N}\sum_{j=1}^{P} s_{ij}} \left[\prod_{i=1}^{N} |\Sigma_{\{s_{i\bullet}=1\}}|^{-\frac{1}{2}}\right] e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([\underline{x}_{i\bullet}-\underline{\mu}]\circ s_{i\bullet}\right)'\Sigma^{-1}\left([\underline{x}_{i\bullet}-\underline{\mu}]\circ s_{i\bullet}\right)\right]} \times \prod_{i=1}^{N}\left[(1-\rho)f_{NL}(\tau)\right]^{\sum_{j=1}^{P}[1-s_{ij}]}\right] \times |\mathbf{V}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\underline{\mu}-\underline{m})^T\mathbf{V}^{-1}(\underline{\mu}-\underline{m})} \times \cdot \left|\Omega^*\right|^{\frac{\psi}{2}} \left|\Sigma\right|^{-\frac{-(\psi+P+1)}{2}} e^{-\frac{1}{2}\mathrm{tr}\left(\Omega^*\Sigma^{-1}\right)} \tag{5.8}$$

To predict the expected behavior of a new experimental run, we estimate $\underline{\mu}$ and Σ of the multivariate normal component, and none of the parameters in the non-local distribution.

5.3 Modified Cauchy Net Inference

We implemented a Gibbs sampler to sample from the posterior distribution with the Robust Bayesian Regression since direct sampling from each full conditional distribution is feasible. However, since the Modified Cauchy Net model has a more complex joint posterior distribution due to the **S** matrix (indicates anomalies), we cannot directly sample from the full conditional distributions. Thus, we implement multiple steps of the Metropolis-Hasting algorithm (Algorithm 3.1) to help us sample the joint posterior distribution. Algorithm 5.1 displays a general procedural outline of our Metropolis-Hasting algorithm implemented for the Modified Cauchy Net with the full conditional and proposal distributions.

Within our Modified Cauchy Net Metropolis-Hasting algorithm, we implemented the Multi-try Metropolis algorithm to sample from the posterior full conditional on Σ to efficient

sample from the potentially highly-correlated parameter space. Section 5.4 outlines the implementation details for the Multi-try Metropolis algorithm for our Modified Cauchy Net. Additionally, to increase the computational efficiency of our Modified Cauchy Net, we utilized the Multiset sampler for $\mathbf{S}$ matrix due to the discrete, high-dimensional nature of the problem. We provide details of the Multiset algorithm specific to the Modified Cauchy Net in Section 5.5.

Algorithm 5.1: Metropolis-Hasting for Modified Cauchy Net

INITIALIZE VALUES FOR $\Sigma_{(t=0)}, \underline{\mu}^{(t=0)}, \mathbf{S}^{(t=0)}$

for $t \leftarrow 1$ **to** T

1. SAMPLE FROM $\underline{\mu}$ FULL CONDITIONAL (UP TO PROPORTIONALITY)

$$f\left(\underline{\mu}^{(t)}|\Sigma_{(t-1)}, s_{ij}^{(t-1)}\right) \propto \left[e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}^{(t-1)}\right)'\Sigma_{(t-1)}^{-1}\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}^{(t-1)}\right)\right]}\right] \times |\mathbf{V}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\underline{\mu}-\underline{m})'\mathbf{V}^{-1}(\underline{\mu}-\underline{m})},$$

WITH $\underline{\mu}$ PROPOSAL DISTRIBUTION: $g(\underline{\mu}^*) \sim \text{MVN}\left(\underline{\mu}^{(t-1)}, \mathbf{I}_P\right)$

2. SAMPLE FROM Σ FULL CONDITIONAL (UP TO PROPORTIONALITY)

$$f\left(\Sigma_{(t)}|\underline{\mu}^{(t)}, s_{ij}^{(t-1)}\right) \propto \left[\left[\prod_{i=1}^{N}|\Sigma_{\{s_{i\cdot}^{(t-1)}=1\}}|^{-\frac{1}{2}}\right] e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([x_{i\cdot}-\underline{\mu}^{(t)}]\circ s_{i\cdot}^{(t-1)}\right)'\Sigma^{-1}\left([x_{i\cdot}-\underline{\mu}^{(t)}]\circ s_{i\cdot}^{(t-1)}\right)\right]}\right] \times$$
$$|\Omega^*|^{\frac{\psi}{2}}|\Sigma|^{\frac{-(\psi+P+1)}{2}} e^{-\frac{1}{2}\text{tr}\left(\Omega^*\Sigma^{-1}\right)},$$

WITH Σ PROPOSAL DISTRIBUTION: $g(\Sigma^*) \sim \text{Wishart}\left(d, \frac{1}{d}\Sigma_{(t-1)}\right)$ where we set d to be large to search the high-dimensional space of covariance.

3. SAMPLE FROM $s_{i,}$ FULL CONDITIONAL (UP TO PROPORTIONALITY)

for $i \leftarrow 1$ **to** N

$$f\left(s_{ij}^{(t)}|\underline{\mu}^{(t)}, \Sigma_{(t)}\right) \propto \left[\rho^{\sum_{i=1}^{N}\sum_{j=1}^{P}s_{ij}}\prod_{i=1}^{N}|\Sigma_{\{s_{i\cdot}=1\}}|^{-\frac{1}{2}}\right] e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([x_{i\cdot}-\underline{\mu}^{(t)}]\circ s_{i\cdot}\right)'\Sigma_{(t)}^{-1}\left([x_{i\cdot}-\underline{\mu}^{(t)}]\circ s_{i\cdot}\right)\right]}$$
$$\times\prod_{i=1}^{N}\left[(1-\rho)f_{NL}(\tau)\right]^{\sum_{j=1}^{P}[1-s_{ij}]},$$

end

WITH s_{ij} PROPOSAL DISTRIBUTION: $g\left(s_{ij^*}\right) \sim \text{Bernoulli}(\pi)$ where j^* represents the randomly choosing sensor.

end

5.4 Multi-try algorithm for Modified Cauchy Net

In the Multi-try sampling scheme, we propose M trial values; specifically, for the Modified Cauchy Net, we suggest M values of Σ, denoted as Σ_m for $m = 1, \ldots, M$. Once given the set of $\Sigma_{(1:M)}$, we sample one of the trial values based on the probability proportional to:

$$w\left(\Sigma^*_{(1:M)}|\Sigma^{(t-1)}\right) = f(\Sigma^*_m|\mathbf{X}) \times g\left(\Sigma^{(t-1)}|\Sigma^*_m\right) \times \lambda\left(\Sigma^*_m, \Sigma^{(t-1)}\right). \tag{5.9}$$

Our full conditional distribution for Σ_m given values current values of $\underline{\mu}$ and $\mathbf{S}$, up to proportionality, is:

$$f\left(\Sigma^*_m|\underline{\mu}, \mathbf{S}\right) \propto \left[\left[\prod_{i=1}^{N} |\Sigma^*_{m,\{s_{i\cdot}=1\}}|^{-\frac{1}{2}}\right] e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}\right)^T (\Sigma^*_m)^{-1}\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}\right)\right]}\right] \times$$
$$|\Omega^*|^{\frac{\psi}{2}} |\Sigma^*_m|^{-\frac{-(\psi+P+1)}{2}} e^{-\frac{1}{2}\mathrm{tr}\left(\Omega^*(\Sigma^*_m)^{-1}\right)}, \tag{5.10}$$

where $\Omega^* = (\psi - P - 1) \times \Omega$. The proposal distribution, $g(\Sigma^{(t-1)}|\Sigma^*_m)$ is a Wishart distribution with degree of freedom, d, and scale matrix, $\frac{1}{d}\Sigma^{(t-1)}$, with density (up to proportionality):

$$g\left(\Sigma^{(t-1)}|\Sigma^*_m\right) \propto |\Sigma^*|^{\frac{d-P-1}{2}} e^{\frac{-1}{2}\mathrm{tr}\left[\left(\frac{1}{d}\Sigma^{(t-1)}\right)^{-1}\Sigma^*\right]}. \tag{5.11}$$

We set the symmetric function, $\lambda(\bullet) = 1$. Algorithm 5.2 intertwines Algorithm 3.3 along with the Modified Cauchy Net specific equations such as our proposal distribution and our weight function, $w(\bullet)$.

Algorithm 5.2: Multi-try Metropolis Algorithm for Modified Cauchy Net

INITIALIZE VALUES FOR $\Sigma^{(t=0)}$

for $t \leftarrow 1$ **to** T

1. PROPOSE M NEW TRIAL VALUES

$\theta^*_{(1:M)} = \{\theta^*_1, \theta^*_2, \ldots, \theta^*_M\} \sim \text{Wishart}\left(d, \frac{1}{d}\Sigma^{(t-1)}\right)$

2. SAMPLE A $\Sigma^*_{(1:M)}$ WITH PROBABILITY PROPORTIONAL TO $w(\Sigma^*_m|\Sigma^{(t-1)})$

$$w\left(\Sigma^*_m|\theta^{(t-1)}\right) = \left[\left[\prod_{i=1}^{N}|\Sigma^*_{m,\{s_{i\cdot}=1\}}|^{-\frac{1}{2}}\right] e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}\right)^T(\Sigma^*_m)^{-1}\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}\right)\right]}\right] \times$$
$$|\Omega^*|^{\frac{\psi}{2}}|\Sigma^*_m|^{-\frac{-(\psi+P+1)}{2}} e^{-\frac{1}{2}\text{tr}\left(\Omega^*(\Sigma^*_m)^{-1}\right)} \times$$
$$|\Sigma^*|^{\frac{c-P-1}{2}} e^{\frac{-1}{2}\text{tr}\left[\left(\frac{1}{c}\Sigma^{(t-1)}\right)^{-1}\Sigma^*\right]}$$

where Σ^*_j denotes the selected $\Sigma^*_{(1:M)}$.

3. SAMPLE $M-1$ REFERENCE VALUES

for $m \leftarrow 1$ **to** $M-1$

$$\Sigma^r_m \sim \text{Wishart}\left(d, \frac{1}{d}\Sigma^*_{(j)}\right)$$

end where $\Sigma^r_{(M)} = \Sigma^{(t-1)}$

4. DECISION

$$\Sigma^{(t)} = \begin{cases} \Sigma^*_j & \text{with prob. } \alpha_{MTM} = \min\left(1, \frac{\sum_{m=1}^{M} w\left(\Sigma^*_m|\Sigma^{(t-1)}\right)}{\sum_{m=1}^{M} w\left(\Sigma^r_m|\Sigma^*_j\right)}\right) \\ \Sigma^{(t-1)} & \text{with prob. } 1-\alpha_{MTM} \end{cases}$$

end

5.5 Multiset sampler for Modified Cauchy Net

While the Multi-try sampler solely impacts the the Σ full conditional, implementing the Multiset sampler requires us to slightly adjust acceptance ratio for each Metropolis-Hasting step to incorporate the inclusion of the K multisets of $\mathbf{S}$. We first discuss the Multiset sampler implementation of $\mathbf{S}$, then briefly describe the minor adjustments for $\underline{\mu}$ and Σ. Algorithm 3.4 outlined a generalized version of the Multiset sampler, Algorithm 5.3 outline the Multiset sampler specifically for the Modified Cauchy Net.

In Algorithm 5.1, we describe the full conditional of a s_{ij} given all the other parameters. When utilizing the Multiset sampler, we utilize the full conditional distribution of the $\mathbf{S}$ matrix:

$$f(\mathbf{S}|\underline{\mu},\Sigma) \propto \prod_{i=1}^{N}\Bigg[\rho^{\sum_{j=1}^{P} s_{ij}}(1-\rho)^{P-\sum_{j=1}^{P} s_{ij}} \times f_{MVN}\left(x_i \circ s_i|\underline{\mu}_{s_{i\bullet}=1}, \Sigma_{\{s_{i\bullet}=1\}}\right) \times \prod_{j|s_{ij}=0} f_{NL}\left(x_i \circ s_{ij=0}|\tau\right)\Bigg] \tag{5.12}$$

where $f_{MVN}(\bullet)$ denotes calculating the density of $x_i \circ s_i$ with the associated $\underline{\mu}$ and Σ. For instance, if we let:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{bmatrix}; \quad s_i = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix};$$

with the multivariate parameters of:

$$\underline{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix};$$

then,

$$f_{MVN}\left(x \circ s_i | \underline{\mu}_{s_{i\bullet}=1}, \Sigma s_{i\bullet} = 1\right) = f_{MVN}\left(\begin{bmatrix} x_{i2} \\ x_{i3} \end{bmatrix} | \underline{\mu}_{s_{i\bullet}=1} = \begin{bmatrix} \mu_2 \\ \mu_3 \end{bmatrix}, \Sigma s_{i\bullet} = 1 = \begin{bmatrix} \sigma_2^2 & \sigma_{23} \\ \sigma_{32} & \sigma_3^2 \end{bmatrix}\right).$$

We reduce Eq. 5.12 to:

$$f(\mathbf{S}|\underline{\mu}, \Sigma) \propto \rho^{\sum_{i=1}^{N}\sum_{j=1}^{P} s_{ij}}(1-\rho)^{NP-\sum_{i=1}^{N}\sum_{j=1}^{P} s_{ij}} \times f_{MVN}\left(x_i \circ s_i | \underline{\mu}_{s_{i\bullet}=1}, \Sigma_{s_{i\bullet}=1}\right) \times \prod_{i}^{N} \prod_{j \in \{s_{ij}=0\}} f_{NL}(x_i \circ s_{ij=0} | \tau) \tag{5.13}$$

by bring the product sign to help with computational ease and utilize Eq. 5.13 within the

acceptance ratio in Algorithm 5.3.

Algorithm 5.3: Multiset sampler for $\mathbf{S}$ in Modified Cauchy Net

Initialize the K multisets of $\mathrm{S}^{(k)}$. Without loss of generality, we assume $K = 2$.

1. Sample one of the K multisets

$$\mathbf{S}^* \sim \text{Multinomial}\left(\mathbf{S}^{(k)}, \underline{\rho} = \frac{1}{K}\right)$$

Without loss of generality, assume $\mathrm{S}^* = \mathrm{S}^1$

2. Sample a row and column indices

$$i^* \sim \text{Discrete Uniform}\,(1, N)\,; \quad j^* \sim \text{Discrete Uniform}\,(1, P)$$

3. Propose a new value given i^* and j^*

$$\mathbf{S}^* = \mathbf{S}^*_{i^*,j^*} \sim \text{Binomial}\,(\rho_o)$$

4. Decide

$$\mathbf{S}_{(t)} = \begin{cases} \mathbf{S}^* & \text{with probability } \alpha_{MSS} = \min\left\{1, \frac{f(\mathbf{S}^*|\underline{\mu},\Sigma)+f(\mathbf{S}^2|\underline{\mu},\Sigma)}{f(\mathbf{S}^1|\underline{\mu},\Sigma)+f(\mathbf{S}^2|\underline{\mu},\Sigma)}\right\} \\ \mathbf{S}_{(t-1)} & \text{with probability } 1 - \alpha_{MSS} \end{cases} \tag{5.14}$$

5.5.1 Multiset sampler adjustments for $\underline{\mu}$ and Σ full posterior distribution

Our Metropolis-Hasting acceptance ratio for $\underline{\mu}$ without the Multiset implementation is:

$$\alpha = \min\left(1, \frac{f(\underline{\mu}^*|\Sigma, \mathbf{S})g\left(\underline{\mu}^{(t-1)}|\underline{\mu}^*\right)}{f(\underline{\mu}^{(t-1)}|\Sigma, \mathbf{S})g\left(\underline{\mu}^*|\underline{\mu}^{(t-1)}\right)}\right).$$

However, the Multiset sampler flattens our $\mathbf{S}$ space through the K multiset and adjusts our target distribution of $\underline{\mu}$ to account for the K multiset. Therefore, the new Metropolis-Hasting acceptance ratio for $\underline{\mu}$ with the Multiset implementation ($K = 2$) is:

$$\alpha = \min\left(1, \frac{f\left(\underline{\mu}^*|\Sigma, \mathbf{S}^{(1)}\right) + f\left(\underline{\mu}^*|\Sigma, \mathbf{S}^{(2)}\right)}{f\left(\underline{\mu}^{(t-1)}|\Sigma, \mathbf{S}^{(1)}\right) + f\left(\underline{\mu}^{(t-1)}|\Sigma, \mathbf{S}^{(2)}\right)} \times \frac{g\left(\underline{\mu}^{(t-1)}|\underline{\mu}^*\right)}{g\left(\underline{\mu}^*|\underline{\mu}^{(t-1)}\right)}\right),$$

where:

$$f\left(\underline{\mu}|\Sigma, \mathbf{S}^{(k)}\right) \propto \left[e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}^k\right)^T\Sigma^{-1}\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}^k\right)\right]}\right] \times |\mathbf{V}|^{-\frac{1}{2}}e^{-\frac{1}{2}(\underline{\mu}-\underline{m})^T\mathbf{V}^{-1}(\underline{\mu}-\underline{m})}.$$

In general, the acceptance ratio is:

$$\alpha = \min\left(1, \frac{\sum_{k=1}^{K} f\left(\underline{\mu}^*|\Sigma, \mathbf{S}^{(k)}\right)}{\sum_{k=1}^{K} f\left(\underline{\mu}^{(t-1)}|\Sigma, \mathbf{S}^{(k)}\right)} \times \frac{g\left(\underline{\mu}^{(t-1)}|\underline{\mu}^*\right)}{g\left(\underline{\mu}^*|\underline{\mu}^{(t-1)}\right)}\right). \tag{5.15}$$

In our algorithm, we utilize a symmetric proposal; therefore, our ratio of inverse proposals cancels out and ease some computational burden.

In the Multi-try algorithm, the Multiset scheme only impacts the formulation of the weight function, $w(\bullet)$, through $f(\Sigma|\mathbf{X}, \underline{\mu}, \mathbf{S})$, Eq. 5.16, to incorporate the K multisets. The update

target distribution is:

$$f\left(\Sigma_m^*|\underline{\mu},\mathbf{S}^k\right) \propto \left[\left[\prod_{i=1}^{N}|\Sigma^*_{m,\{s_{i\cdot}^k=1\}}|^{-\frac{1}{2}}\right] e^{-\frac{1}{2}\sum_{i=1}^{N}\left[\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}^k\right)^T(\Sigma_m^*)^{-1}\left([x_{i\cdot}-\underline{\mu}]\circ s_{i\cdot}^k\right)\right]}\right]\times \\ |\Omega^*|^{\frac{\psi}{2}}|\Sigma_m^*|^{-\frac{-(\psi+P+1)}{2}}e^{-\frac{1}{2}\mathrm{tr}\left(\Omega^*(\Sigma_m^*)^{-1}\right)}, \tag{5.16}$$

where the impact of the multisets comes through the vector s^k of $\mathbf{S}$ matrix. The acceptance ratio for the Multi-try Metropolis with the Multiset implementation is:

$$\alpha_{MTM}=\min\left(1,\frac{\sum_{m=1}^{M}w\left(\Sigma_m^*|\Sigma^{(t-1)}\right)}{\sum_{m=1}^{M}w\left(\Sigma_m^r|\Sigma_j^*\right)}\right),$$

where the ratio of the weight functions expands to:

$$\frac{\sum_{m=1}^{M}w\left(\Sigma_m^*|\Sigma^{(t-1)}\right)}{\sum_{m=1}^{M}w\left(\Sigma_m^r|\Sigma_j^*\right)}= \\ \frac{\sum_{m=1}^{M}\left(\left[\sum_{k=1}^{K}f\left(\Sigma_m^*|\underline{\mu},\mathbf{S}^k\right)\right]\times g\left(\Sigma^{(t-1)}|\Sigma_m^*\right)\right)}{\sum_{m=1}^{M}\left[f\left(\Sigma_m^r|\underline{\mu},\mathbf{S}^k\right)\right]\times g\left(\Sigma_j^*|\Sigma_m^r\right)}.$$

Chapter 6

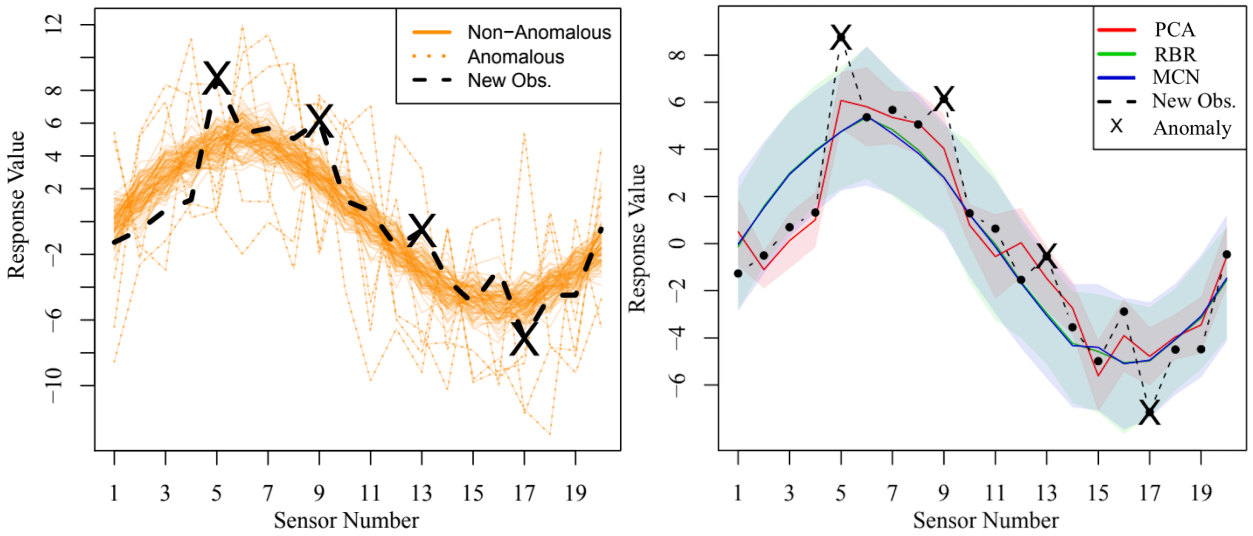
Simulation results

In large-scale multi-type sensor systems, such as the Virginia Tech Stability wind tunnel, the individual sensors or sensor systems seldomly act independently. While the apparatuses are inherently different in regards to their output information, the apparatuses have a relationship through their physical proximity, the experimental set-up, or the environmental elements. In our simulation study, we evaluate and compare the ability to detect anomalous sensors for the three monitoring systems under different known sensor covariance structures. Additionally, since real-experimental data containing potential anomalies, we analyze the detection methods work under various levels of anomalous sensor readings in the training data. The following simulation experiments consider the cases where the number of runs is greater than the number of sensors, $N > P$. We describe the basis of our simulations in and present results in Section 6.2.

6.1 Simulation Set-Up

We design our simulation studies to mimic the properties of the large-scale sensor systems, specifically the Virginia Tech Stability Wind Tunnel, on a smaller scale. In our simulations, we perform a Monte Carlo simulation by generating a training dataset, creating an independent observation vector with known anomalous sensors, and evaluating the anomaly detection methods. Figure 6.1 illustrates the Monte Carlo simulation process of generating a

training dataset, Figure 6.1a, and apply the three anomaly detection methods, Figure 6.1b. In Figure 6.1a, a new observation contains a particular amount of anomalous sensors, denoted by a dashed black line with black Xs. We apply the three anomaly detection methods to generate the predicted response surfaces and uncertainty bands given the training dataset and new run as illustrated Figure 6.1b. If a sensor exists outside the uncertainty bands, we flag the sensor as an anomaly. We compare our methods by assessing the type I and type II error rates of each sensor. A type I error (also known as a false positive) occurs when the method flags non-anomalous sensor reading as anomalous readings. In contrast, a type II error (also known as a false negative) occurs when the method detects an anomalous sensor reading as non-anomalous. We repeat the process of generating the datasets and a new observation, implementing the three monitoring systems, and evaluating the errors fifty times to calculate an average type 1 and type 2 error rate per sensor over the various anomalous case.



(a) An illustration of generating training dataset with anomalous observation with the new observation.

(b) Demonstration of predicted response surface with uncertainty calculated based on Figure 6.1a data.

Figure 6.1: Illustrating the Monte Carlo simulation process of generating a training dataset and apply the three anomaly detection methods.

Within this preliminary manuscript, we perform a simulation studies that evaluates the three methods based on various training sets.

6.2 Generalized Simulation Study

We generated numerous training datasets by varying three types of sensor covariance structure and three levels of "noisy" training data. In Figure 6.2, we provide examples of the correlation matrices to allow for a standardized comparison of the increasing sensor relationship for the respective covariance matrices used.

Eq. 5.4 and Eq. 5.5 generates and classifies the training observations as non-anomalous and anomalous, respectively. We parameterize the signal component, i.e., multivariate normal, based on one of the three covariance structures in Figure . While the noise component, i.e., non-local distribution, generates anomalous observations that are at least three (3) standard deviations away from the mean. Figure 6.1a illustrates a single dataset generated under correlation structure #3 with 5% of anomalous data. We characterize new, independent observations into four (4) categories. The categories are the sensors are: (1) all anomalous, (2) all non-anomalous, (3) 20% of the sensors are anomalous, and (3) 30 % of the sensors are anomalous.

The relationship between sensors (i.e., the covariance structure) and the amount of noisy data in the historical dataset are often two aspect that impact the monitoring methodologies. Thus, in our simulation study, we focus on these two aspects to evaluate and compare of the three monitoring methodologies. Figure 6.3 illustrates the average impact of increase noisy historical (i.e., anomalous sensor readings) while Figure 6.4 displays the impact of different

sensor correlation structures on type I and type II sensor error rates.

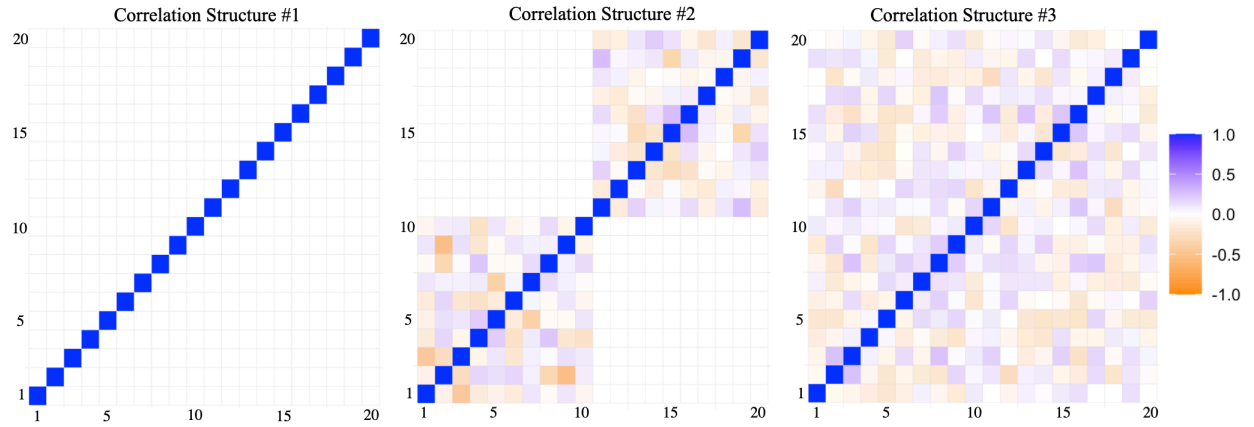


Figure 6.2: Examples of the three correlation structures. Correlation structure #1 (left) represents the uncorrelated sensors case. Correlation Structure #2 (middle) represents correlation within sensors system, but not between sensor systems. Correlation Structure #3 (right) represent a more generalized correlation structure.

In Figure 6.3, when the amount of contaminated data increases, the overall trend is that the average type I error rates decreases and the average type II error rates increase. Additionally, we see that RBR has a higher type I error rate compared to the other methods. In the RBR model assumption, the method places a weight (γ_i) on each observation to influences whether an entire observational run is anomalous or not. However, in our simulation studies, we generate historical datasets to represent more realistic anomalous observations, as in Fig 5.2b. Thus, the RBR results may be impacted by the more realistic data scheme. From Figure 6.3 we see, on average, the MCN method produces lower type I and type II error rates. Across the various correlation structures, our simulation study illustrates that the average type I and type II error rates are consistent over different covariance relationships and that Modified Cauchy Net method has lower error rates as seen in Figure 6.4.

The type I and type II errors are influenced by choice of tuning parameters, i.e., the k-modes or the hyper-parameters in RBR (Ω or ψ) and MCN (τ). The difficulty of PCA

lies within identifying a k which balances the sensor estimates and their uncertainty values. For the RBR and MCN methodology, the choice of Ω, ψ, τ impacts the anomaly detection results. The advantage that RBR and MCN have over PCA is that the sensor estimates and uncertainty are not correlated. That is when k increases, the predicted estimates go towards the observed measurements, and the uncertainty estimates are driven towards zero. An advantage MCN has over both RBR and PCA is the tunable τ parameter that allows the methodology to incorporate expert knowledge of anomalous behavior.

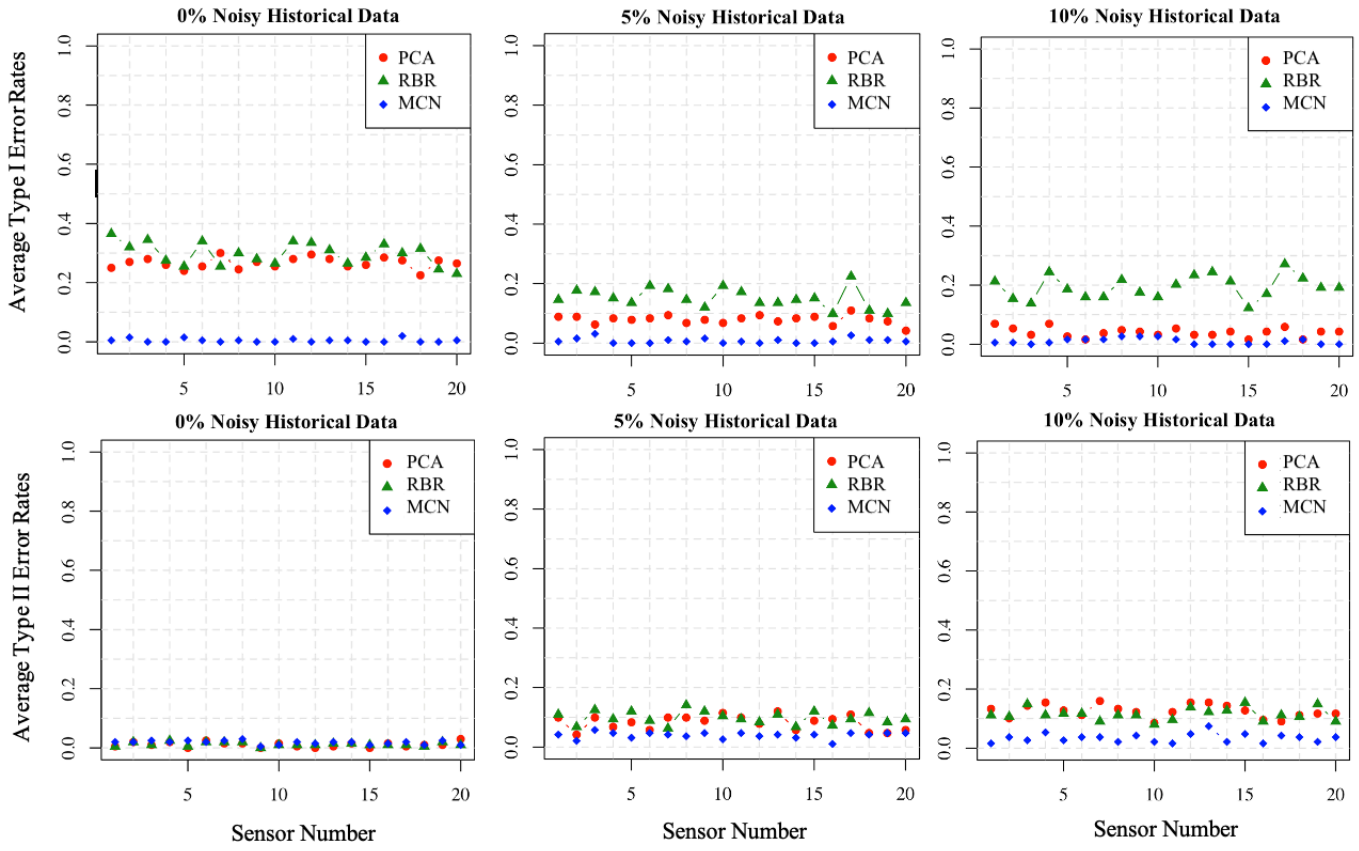


Figure 6.3: Illustration of type I error rate (top row) and type II error rate (bottom row) for correlation structure #2 across various levels of the historical data contaminated with anomalous sensors.

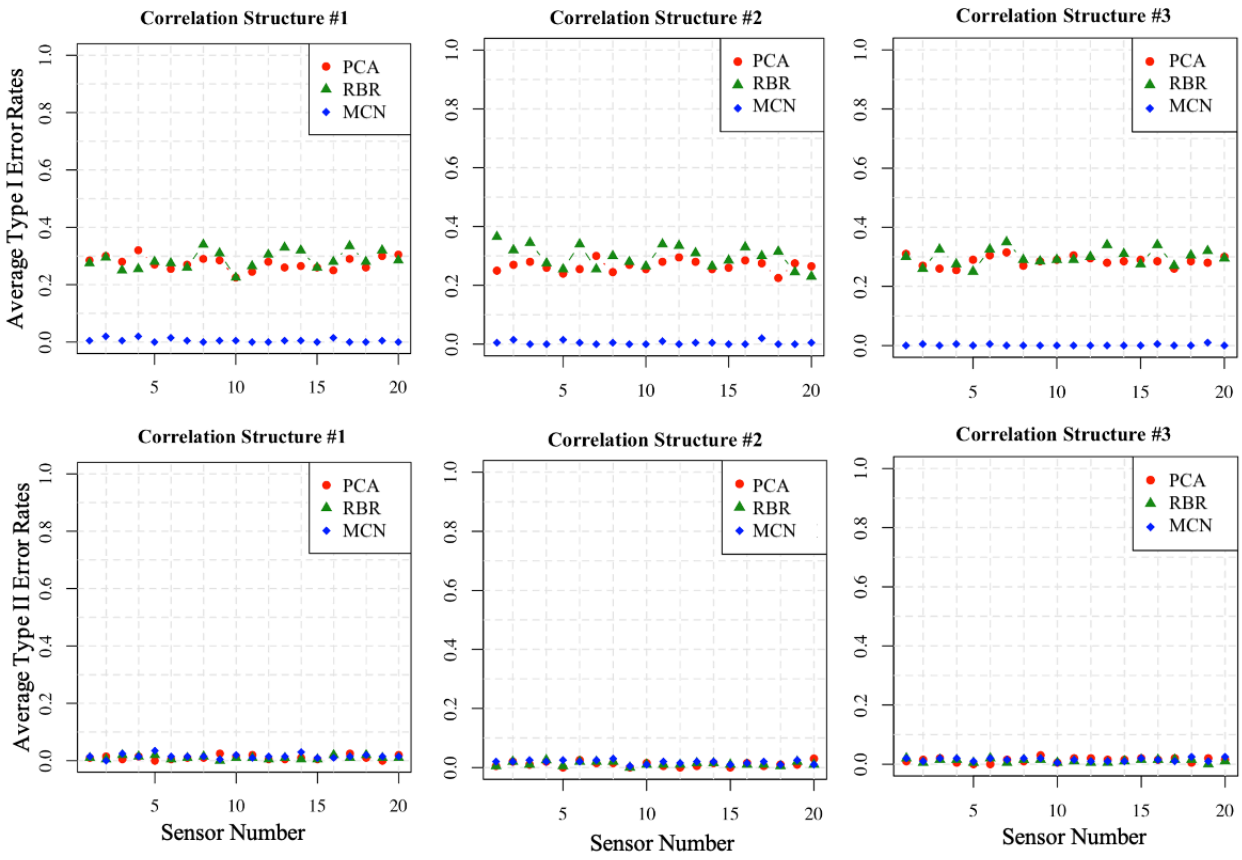


Figure 6.4: Illustration of type I error rate (top row) and type II error rate (bottom row) for across various covariance structures.

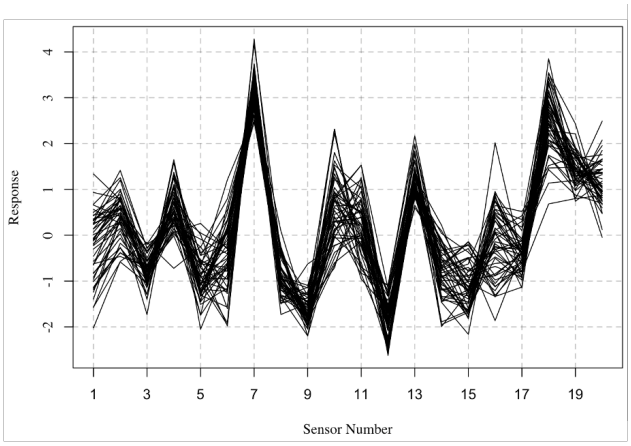
6.3 Comparative Analysis under the different model assumptions

The Principal Component Analysis, the Robust Bayesian Regression, and the Modified Cauchy Net model assumptions are three different scenarios that can stem from experimental data. That is, we could expect the data to (1) be all non-anomalous (PCA), (2) have entire runs anomalous, or (3) have anomalous sensors intermittently mixed within runs. Thus, we decide to explore the on-average type 1 and on-average type 2 errors for the three

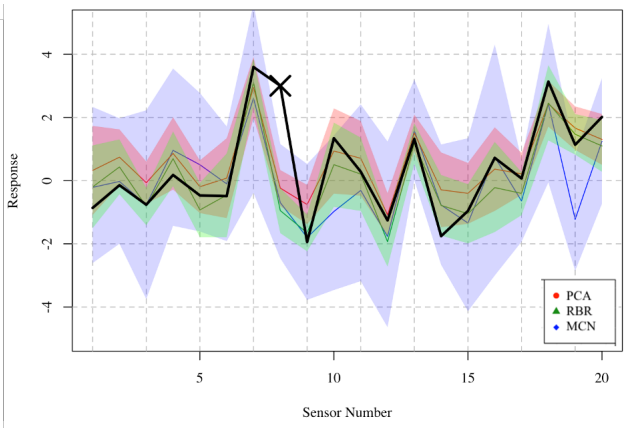
methodologies under the various data generation scheme. In this simulation study, we either generated data from a multivariate normal distribution, i.e., assuming all clean data, or from the Modified Cauchy sampling scheme with $\rho = 0.02$. We did not generate data from the Robust Bayesian Regression model because we can not accurately track the anomalous sensors or the percentage of entirely anomalous runs within the Robust Bayesian Regression model. After generating the data, we evaluate all three methods to construct each response surface, as illustrated in Figures 6.5b, 6.6b, 6.7b, and 6.8b, and assess the overall on-average type 1 and type 2 error rates for a new observations. We generated new observations that contain between zero and two anomalous sensors. We repeat this process for several iterations.

Table #1 and Table #2 represent the overall average Type 1 error and Type 2 error rates for data generated from the Principal Component Analysis and Modified Cauchy Net model assumptions, respectively. In addition, we provide two accompanying examples for each assumption to help demonstrate the various properties of each method. In Table #1, we see that the PCA Type 1 error rates are significantly higher; this artifact stems from the construction of the response surface. The RBR and MCN have relatively similar Type 1 errors with 0.07 and 0.06, respectively, but the difference lies in the Type 2 error rates. Recall that when we flag an anomalous sensor as non-anomalous, we refer to this as a Type 2 error. That is, our method's uncertainty bounds are wide enough to capture the anomalous observation within the uncertainty bounds when it should lay outside the bounds. By observing Figure 6.5b and 6.6b, we see that the Modified Cauchy Net is producing larger uncertainty bands than the Robust Bayesian Regression, which help illustrates the difference in Type 2 error results.

Table #1: Error Rates for when PCA generated data.			
Method	PCA	RBR	MCN
Average Type 1 Error	0.25	0.077	0.061
Average Type 2 Error	0.005	0.00	0.021

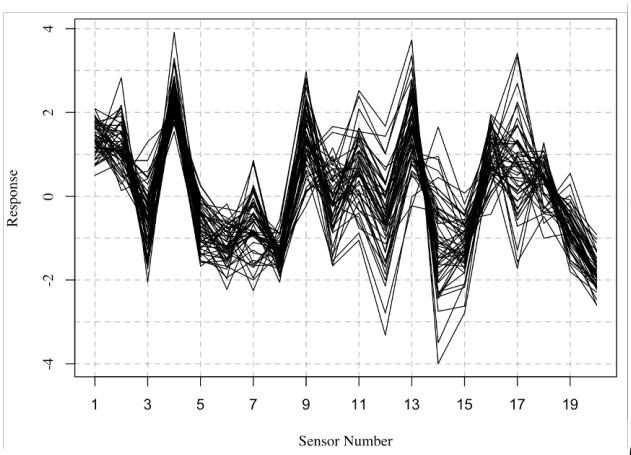


(a) Visualization of PCA generated data.

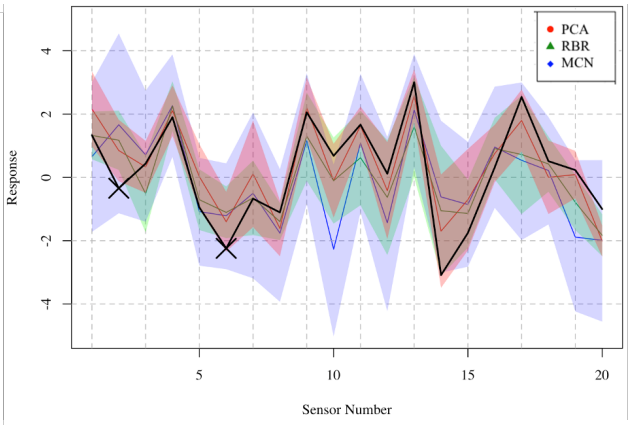


(b) Response surface for all three methods given the data in Figure 6.5a.

Figure 6.5: Example #1 of illustrating of PCA generated data with response surface results.



(a) Visualization of PCA generated data.



(b) Response surface for all three methods given the data in Figure 6.6a.

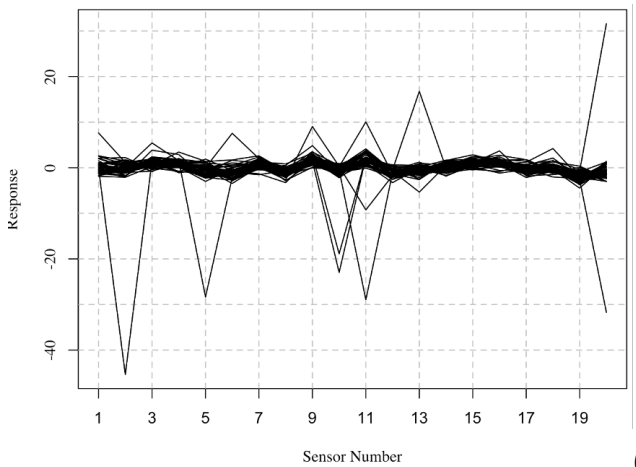
Figure 6.6: Example #2 of illustrating of PCA generated data with response surface results.

Table #2 illustrates the overall error rates under the Modified Cauchy Net assumption.

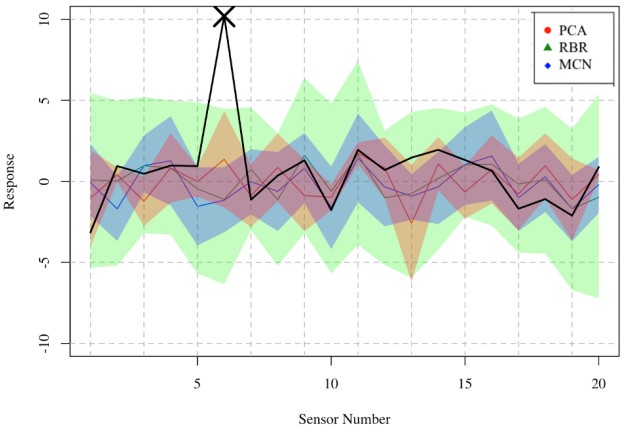
Compared to when the data was generated from under a multivariate normal assumption, the Robust Bayesian Regression Type 1 error decreased, but the Type 2 error increased displaying a similar pattern as the MCN under the PCA model assumption. It should be no surprised that the MCN methodology excels under the Modified Cauchy Net generation scheme.

Future work will consist of increasing the number of datasets evaluated and extending to different scenarios to aid in demonstrating the strengths of the MCN and the RBR and comparing computational time.

Table #2: Error Rates under generating data from MCN model.			
Method	PCA	RBR	MCN
Average Type 1 Error	0.08	0.00	0.025
Average Type 2 Error	0.014	0.014	0.00

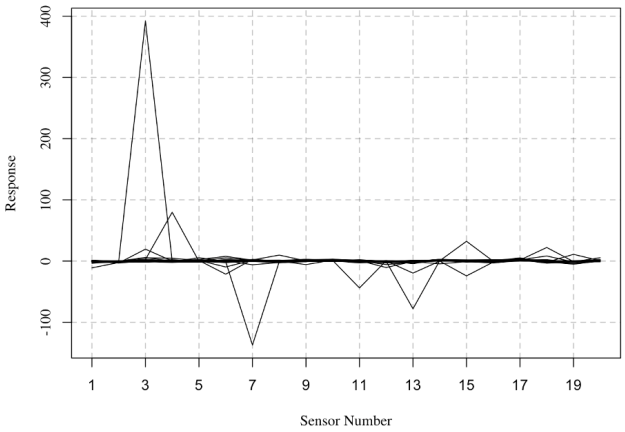


(a) Visualization of MCN generated data.

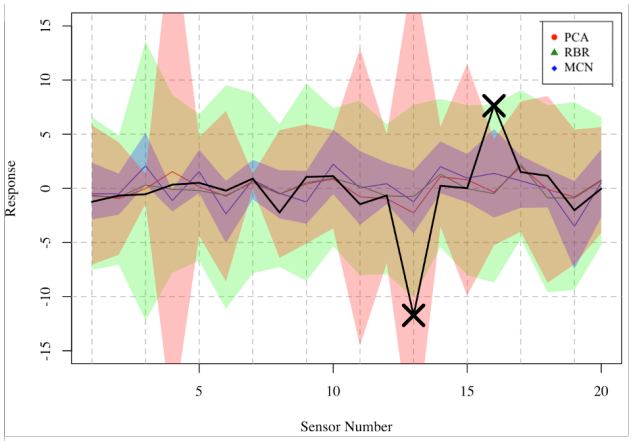


(b) Response surface for all three methods given the data in Figure 6.7a.

Figure 6.7: Example #1 of illustrating of MCN generated data with response surface results.



(a) Visualization of MCN generated data.



(b) Response surface for all three methods given the data in Figure 6.8a.

Figure 6.8: Example #2 of illustrating of MCN generated data with response surface results.

Chapter 7

Wind tunnel case studies

Experimental sensor systems that collect, store, and analyze large amounts of high-dimensional data, such as those found in modern wind tunnel experiments, are susceptible to different quantities of errors. Typically, errors transpire either in the collection process due to erroneous sensor readings or misreported sensor readings in the storage process. If undetected errors persist, scientific and engineering results and conclusions may be compromised or misleading. Furthermore, if these errors are detected late, there will be financial costs and experimental downtime. Sometimes field experts rectify misreported data in the post-processing analysis phase; however, erroneous sensor readings will likely cripple an experiment's results. Regardless of the error type, we need to detect errors early. In this chapter, we refer to the various error types as anomalous or anomalies, intending to evaluate three anomaly monitoring sensor systems collected data from Virginia Tech Stability Wind Tunnel.

While the chapter's purpose is to compare anomaly detection methodologies, we briefly discuss the experimental setup and data collection of the Virginia Tech Stability Wind Tunnel facilities. For further details that surpass the scope of our experimental set-up reference [24, 27].

7.1 Wind Tunnel Facilities and Data Collection

The Virginia Tech Stability Wind Tunnel, blueprinted in Figure 7.1, is a continuous, single return, subsonic, anechoic wind tunnel. That is, the wind tunnel engineers designed an "echo-free" environment to reduce the reflection of sound that operates at wind speeds under subsonic speeds (< 180 mph) with a continuous wind current. Additionally, the Virginia Tech Stability Wind Tunnel has an interchangeable test section, which is 7.32 meters stream-wise with a 1.85-meter square cross-section, enabling the ability to change various experimental apparatus such as different types of airfoils. The interchangeable test section for our experimental case study includes a wind turbine airfoil, a traversing wake rake, and tensioned kevlar walls serving acoustic windows for microphone arrays. Within this work, we focus on the airfoil and wake rake sensors and leave incorporating the microphone array sensors for future work. While there are a variety of wind turbine airfoils utilized in the Virginia Tech Stability Wind Tunnel, as illustrated in Figure 7.2 which differ in thickness, shape, and amount of sensors, our case studies use data from a DU91 airfoil. The wake rake moves vertically within the wind tunnel, collecting individual pressure readings at each location and providing an average pressure response.

Each experimental run (e.g., observation) contains a variety of pressure coefficient (C_p) measurements, including the average airfoil sensor measurements and the average wake rake sensor measurements, along with meta-data, such as angle of attack (AOA), temperature (T), flow speed (U), and ambient air pressure (p_{atm}). The angle of attack refers to the position of the airfoil in the test section, which results in different pressure distributions of the wake rake response. Within our experimental set-up, we have sixteen different angles of attack total. In addition, all the experimental observations are stored in a large MATLAB

database file with additional wind tunnel descriptive data that we reference as the historical dataset. Finally, we use a selected historical dataset for anomaly detection methodologies to create the training dataset.

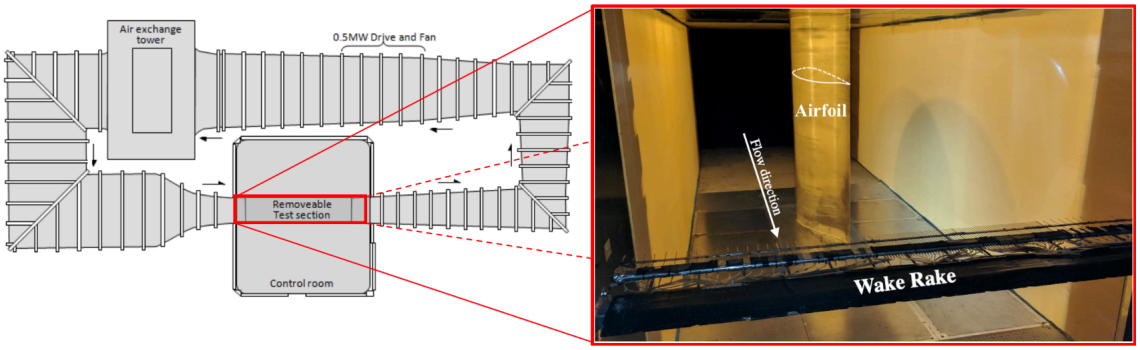


Figure 7.1: Illustration of the Virginia Tech Stability Wind Tunnel

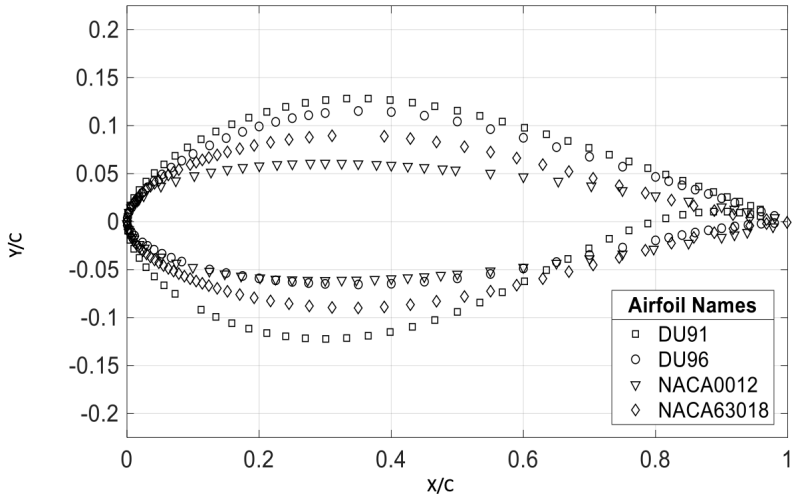
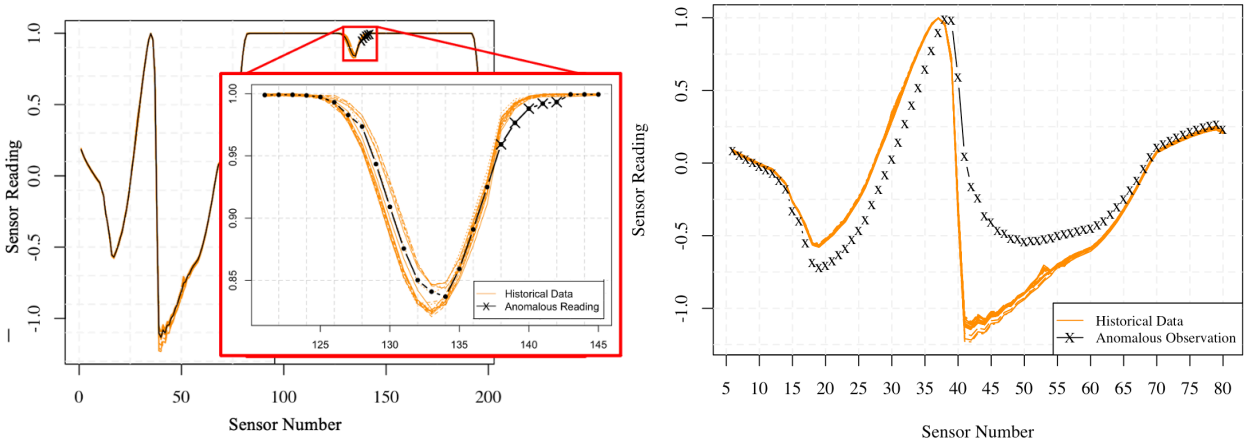


Figure 7.2: Illustration of four various airfoil

7.2 Wind Tunnel Results

Using real wind tunnel sensor readings, we evaluate the three sensor anomaly monitoring systems' effectiveness (PCA, RBR, MCN) by evaluating two case studies. Since errors typ-

ically occur either in the collection process due to erroneous sensor readings or the storage process due to misreported readings, we consider these two cases. Figure 7.3a shows the collected historical data (orange lines) for a DU96 airfoil at an angle of attack of -4 with five (5) anomalous sensors on the wake rake (sensors 139-143). Our second case study investigates misreported sensor readings, illustrated in Figure 7.3b. Figure 7.3b utilizes the same historical data display in Figure 7.3a; however, now the anomalous sensor readings are due to a misreported angle of attack for the airfoil. Figure 7.4 shows the regression and uncertainty bands for the anomaly detection methods.

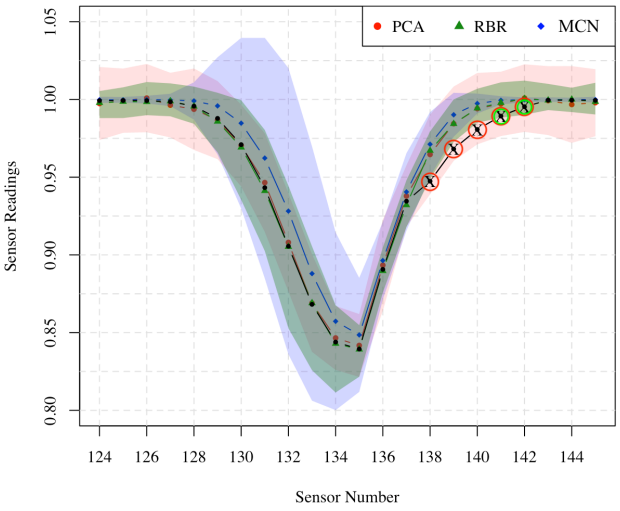


(a) Illustration of historical data and anomalous observation for collection process error.

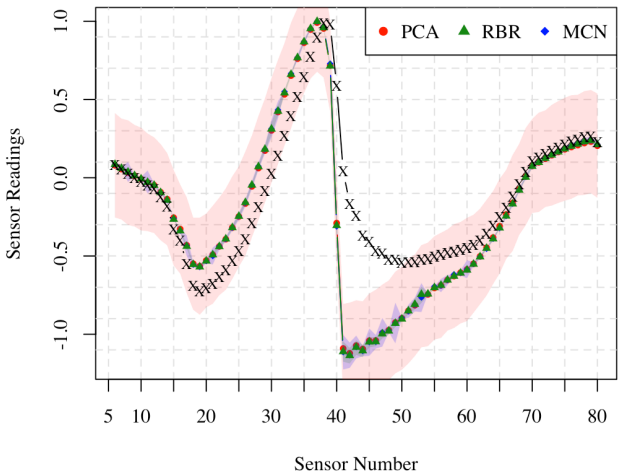
(b) Illustration of historical data and anomalous observation for storage process error.

Figure 7.3: Illustration of two anomalous observations. The orange lines represent the collected historical data, and the black line represents the experimental run with erroneous sensor reading.

In Figure 7.4a, we see that our Modified Cauchy Net method correctly classifies all five anomalous observation as anomalous. Meanwhile, RBR misclassifies two (2) sensors, and PCA misclassifies all five (5) sensors. In Figure 7.4b, we see that our Modified Cauchy Net method and RBR correctly classifies the majority of the anomalous sensor readings and have smaller uncertainty bands than PCA. The interesting aspect to notice is the predicted variance around the response surface for RBR and MCN. The RBR method has a rela-



(a) Illustration of the anomaly detection response surfaces for collection process error. The circles represent misclassified sensors for the respective method color.



(b) Illustration of the anomaly detection response surfaces for storage process error

Figure 7.4: Illustration of the anomaly detection response surfaces.

tively stable uncertainty band across all sensor readings, whereas the Modified Cauchy Net method's uncertainty band adjusts based on the historical. That is, in Figure 7.3a we see that sensor 130-135 have more variability compared to sensors 136-144, and the Modified Cauchy Net method captures the difference in the sensors' variability. While it is harder to see, this also occurs in the storage process error case study, Figure 7.4b.

Chapter 8

Conclusions and Future Work

Given potential anomalous behaviors in wind tunnel experiments, attention to robust techniques is necessary. Anomalous sensor readings can influence the covariance estimation in single model-based methodologies, such as PCA. We introduced two Bayesian methodologies that utilize heavy-tailed distribution, specifically the Cauchy distribution and the "non-local" distribution, to help reduce the influence of anomalous observations. Our generalized approach, the Modified Cauchy-Net, enables us to decompartmentalize anomalous from reliable sensor readings, which allows us to estimate reliable covariance structures within the non-anomalous readings. The two new main components to the Modified Cauchy Net are (1) incorporating the non-local distribution to describe anomalous behaviors and (2) utilizing a partial anomaly structure (i.e., $\mathbf{S}_{N\times P}$ matrix) to classify anomalous sensors. Ultimately, this increases statistical power, reduces variances within the non-anomalous readings, and conclusively reduces the error rates. While the Modified Cauchy Net method outperforms PCA and RBR, in our examples, the Modified Cauchy Net method original had a significantly longer computational running time due to estimating the sensor classification parameter s_{ni}.

In this thesis, we presented the partial anomaly structure (i.e., $\mathbf{S}_{N\times P}$ matrix) which allows sensors within a single run to be intermittently anomalous. Furthermore, another type of Modified Cauchy Net could classify entire observational runs as anomalous or not. In the future, we will aim to provide a full investigation into analyzing the computational efficiency

and error rate of the Modified Cauchy Net under different anomaly structures. Additionally, we will investigate combining the Cauchy Net for functional analysis (i.e., Modified Cauchy Net) with the Cauchy Net developed with the Dirichlet Process [85]. Lastly, within this work, we fixed the choice of the prior distribution hyper-parameters (namely τ) at previously mentioned values; however, other suitable choices remain based on the problem at hand. Following the literature for intrinsic [9] or fractional Bayes factors [75], we might develop a framework for estimating τ with a subset of the data.

Bibliography

[1] A. Agresti and B. A. Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998.

[2] F. B. Alt. Multivariate quality control. *The Encyclopedia of Statistical Sciences*, 1985.

[3] F. B. Alt and N. D. Smith. Multivariate process control. *Handbook of statistics*, 7:333–351, 1988.

[4] D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(1):99–102, 1974.

[5] D. L. Banks. A conversation with IJ good. *Statistical Science*, 11(1):1–19, 1996.

[6] J. Barnard, R. McCulloch, and X. L. Meng. Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica*, pages 1281–1311, 2000.

[7] T. Bayes. LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. *Philosophical transactions of the Royal Society of London*, (53):370–418, 1763.

[8] J. O. Berger, J. M. Bernardo, and D. Sun. The formal definition of reference priors. *The Annals of Statistics*, 37(2):905–938, 2009.

[9] J. O. Berger and L. R. Pericchi. The intrinsic bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122, 1996.

[10] J. M. Bernardo. Reference posterior distributions for Bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):113–128, 1979.

[11] J. M. Bernardo and A. F. Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.

[12] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974.

[13] J. L. Blue. A Legendre polynomial integral. *Mathematics of Computation*, 33(146):739–741, 1979.

[14] L. D. Brown, T. T. Cai, and A. DasGupta. Confidence intervals for a binomial proportion and Edgeworth expansions. Technical report, Purdue University, Statistics Department, 1999.

[15] L. D. Brown, T. T. Cai, and A. DasGupta. Confidence intervals for a binomial proportion and asymptotic expansions. *The Annals of Statistics*, 30(1):160–201, 2002.

[16] C. M. Carvalho, N. G. Polson, and J. G. Scott. The horseshoe estimator for sparse signals. Discussion Paper 2008-31, Duke University Department of Statistical Science, 2008.

[17] G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[18] P. Comon. Independent component analysis, 1992.

[19] I. Crandell, A. J. Millican, S. Leman, E. Smith, W. N. Alexander, W. J. Devenport, R. Vasta, R. Gramacy, and M. Binois. Anomaly detection in large-scale wind tunnel tests using Gaussian processes. In *33rd AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, page 4131, 2017.

[20] N. Cressie. The origins of kriging. *Mathematical Geology*, 22(3):239–252, 1990.

[21] R. B. Crosier. Multivariate generalizations of cumulative sum quality-control schemes. *Technometrics*, 30(3):291–303, 1988.

[22] G. L. L. de Buffon. Essai d'arithmétique morale. *Euvres philosophiques*, 1777.

[23] B. De Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l'institut Henri Poincaré*, volume 7, pages 1–68, 1937.

[24] A. Defreitas, W. N. Alexander, W. J. Devenport, S. Merkes, S. Leman, E. Smith, and A. Borgoltz. Anomaly detection in wind tunnel experiments by principal component analysis. In *AIAA Scitech 2019 Forum*, page 2380, 2019.

[25] A. Defreitas, W. N. Alexander, W. J. Devenport, S. N. Merkes, S. Leman, E. Smith, and A. Borgoltz. Improved anomaly detection in experimental wind tunnel data using PCA. In *AIAA Scitech 2020 Forum*, page 1198, 2020.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[27] W. J. Devenport, R. A. Burdisso, A. Borgoltz, P. A. Ravetta, M. F. Barone, K. A. Brown, and M. A. Morton. The kevlar-walled anechoic wind tunnel. *Journal of Sound and Vibration*, 332(17):3971–3991, 2013.

[28] P. J. Diggle, J. Tawn, and R. Moyeed. Model-based geostatistics. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47(3):299–350, 1998.

[29] N. Doganaksoy, F. W. Faltin, and W. T. Tucker. Identification of out of control quality characteristics in a multivariate manufacturing environment. *Communications in Statistics-Theory and Methods*, 20(9):2775–2790, 1991.

[30] S. E. Fienberg. When did Bayesian inference become "Bayesian"? *Bayesian analysis*, 1(1):1–40, 2006.

[31] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368, 1922.

[32] R. A. Fisher. Statistical methods for research workers. In *Breakthroughs in statistics*, pages 66–70. Springer, 1992.

[33] C. Fraley and A. E. Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification*, 24(2):155–181, 2007.

[34] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 100(9):881–890, 1974.

[35] A. Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian analysis*, 1(3):515–534, 2006.

[36] A. Gelman, H. S. Stern, J. B. Carlin, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2013.

[37] B. Ghosh. A comparison of some approximate confidence intervals for the binomial parameter. *Journal of the American Statistical Association*, 74(368):894–900, 1979.

[38] I. J. Good. Probability and the weighing of evidence. 1950.

[39] R. B. Gramacy, D. Bingham, J. P. Holloway, M. J. Grosskopf, C. C. Kuranz, E. Rutter, M. Trantham, R. P. Drake, et al. Calibrating a large computer experiment simulating radiative shock hydrodynamics. *The Annals of Applied Statistics*, 9(3):1141–1168, 2015.

[40] T. Greville. Spline functions, interpolation and numerical quadrature. *Mathematical method of digital computers*, pages 156–168, 1967.

[41] T. Haigh, M. Priestley, and C. Rope. Los Alamos bets on ENIAC: Nuclear monte carlo simulations, 1947-1948. *IEEE Annals of the History of Computing*, 36(3):42–63, 2014.

[42] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

[43] A. J. Hayter and K.-L. Tsui. Identification and quantification in multivariate quality control problems. *Journal of quality technology*, 26(3):197–208, 1994.

[44] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004.

[45] A. Huang and M. P. Wand. Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, 8(2):439–452, 2013.

[46] M. Hubert and M. Debruyne. Minimum covariance determinant. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):36–43, 2010.

[47] M. Hubert, P. J. Rousseeuw, and K. Vanden Branden. ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

[48] F. James. Monte Carlo theory and practice. *Reports on progress in Physics*, 43(9):1145, 1980.

[49] H. Jeffreys. *The theory of probability*. Oxford University Press., 1939.

[50] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[51] V. E. Johnson and D. Rossell. On the use of non-local prior densities in Bayesian hypothesis tests. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(2):143–170, 2010.

[52] V. E. Johnson and D. Rossell. Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(498):649–660, 2012.

[53] I. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.

[54] H. J. Kim and S. N. MacEachern. The generalized multiset sampler. *Journal of Computational and Graphical Statistics*, 24(4):1134–1154, 2015.

[55] P. S. Laplace. Mémoire sur la probabilité de causes par les évenements. *Mémoire de l'académie royale des sciences*, 1774.

[56] P. S. Laplace. *Théorie analytique des probabilités*, volume 7. Courcier, 1820.

[57] M. Lazzarini. Un'applicazione del calcolo della probabilitá alla ricerca sperimentale di un valore approssimato di π. *Periodico di Matematica*, 4:140–143, 1901.

[58] S. C. Leman, Y. Chen, and M. Lavine. The multiset sampler. *Journal of the American Statistical Association*, 104(487):1029–1041, 2009.

[59] S. C. Leman, M. K. Uyenoyama, M. Lavine, and Y. Chen. The evolutionary forest algorithm. *Bioinformatics*, 23(15):1962–1968, 2007.

[60] J. S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and computing*, 6(2):113–119, 1996.

[61] J. S. Liu, F. Liang, and W. H. Wong. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

[62] C. A. Lowry, W. H. Woodall, C. W. Champ, and S. E. Rigdon. A multivariate exponentially weighted moving average control chart. *Technometrics*, 34(1):46–53, 1992.

[63] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[64] A. W. Marshall. The use of multi-stage sampling schemes in Monte Carlo computations. Technical report, RAND CORP SANTA MONICA CALIF, 1954.

[65] G. J. McLachlan and K. E. Basford. *Mixture models: Inference and applications to clustering*, volume 38. M. Dekker New York, 1988.

[66] N. Metropolis. The beginning of Monte Carlo. *Los Alamos Science*, 15:125–130, 1987.

[67] N. Metropolis and E. C. Nelson. Early computing at Los Alamos. *IEEE Annals of the History of Computing*, 4(04):348–357, 1982.

[68] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[69] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[70] R. v. Mises. On the correct use of Bayes' formula. *The Annals of Mathematical Statistics*, 13(2):156–165, 1942.

[71] D. C. Montgomery. *Introduction to statistical quality control*. John Wiley & Sons, 2013.

[72] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[73] A. D. Morgan. Review of Laplace's theorie analytique des probabilites. *Dublin Review*, 2(3):338–354, 1837.

[74] R. M. Neal. MCMC using ensembles of states for problems with fast and slow variables such as Gaussian process regression. *arXiv preprint arXiv:1101.0387*, 2011.

[75] A. O'Hagan. Fractional bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):99–118, 1995.

[76] A. J. O'Malley and A. M. Zaslavsky. Domain-level covariance analysis for multilevel survey data with structured nonresponse. *Journal of the American Statistical Association*, 103(484):1405–1418, 2008.

[77] J. J. Pignatiello Jr and G. C. Runger. Comparisons of multivariate CUSUM charts. *Journal of quality technology*, 22(3):173–186, 1990.

[78] Z. S. Qin and J. S. Liu. Multipoint Metropolis method with application to hybrid Monte Carlo. *Journal of Computational Physics*, 172(2):827–840, 2001.

[79] D. A. Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741, 2009.

[80] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[81] P. J. Rousseeuw and K. V. Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[82] W. A. Shewhart. Quality control charts. *The Bell System Technical Journal*, 5(4):593–603, 1926.

[83] W. A. Shewhart. Quality control. *The Bell System Technical Journal*, 6(4):722–735, 1927.

[84] W. A. Shewhart. *Economic control of quality of manufactured product*. Macmillan And Co Ltd, London, 1931.

[85] M. D. Slifko. *The Cauchy-Net Mixture Model for Clustering with Anomalous Data*. PhD thesis, Virginia Tech, 2019.

[86] A. F. Smith and G. O. Roberts. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(1):3–23, 1993.

[87] S. M. Stigler. Mathematical statistics in the early states. *The Annals of Statistics*, pages 239–265, 1978.

[88] S. M. Stigler. Richard Price, the first Bayesian. *Statistical Science*, 33(1):117–125, 2018.

[89] Student. Probable error of a correlation coefficient. *Biometrika*, pages 302–310, 1908.

[90] Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

[91] A. Tharwat. Independent component analysis: An introduction. *Applied Computing and Informatics*, 2020.

[92] R. Thisted. Elements of statistical computing. chapman and hall, new york. 1988.

[93] L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, pages 1701–1728, 1994.

[94] L. Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

[95] T. Tokuda, B. Goodrich, I. Van Mechelen, A. Gelman, and F. Tuerlinckx. Visualizing distributions of covariance matrices. *Columbia Univ., New York, USA, Tech. Rep*, pages 18–18, 2011.

[96] S. Ulam, R. D. Richtmyer, and J. von Neumann. Statistical methods in neutron diffusion. *LAMS-551, Los Alamos National Laboratory*, pages 1–22, 1947.

[97] C. Velasco-Cruz, S. C. Leman, M. Hudy, and E. P. Smith. Assessing the risk of rising temperature on brook trout: a spatial dynamic linear risk model. *Journal of Agricultural, Biological, and Environmental Statistics*, 17(2):246–264, 2012.

[98] S. E. Vollset. Confidence intervals for a binomial proportion. *Statistics in medicine*, 12(9):809–824, 1993.

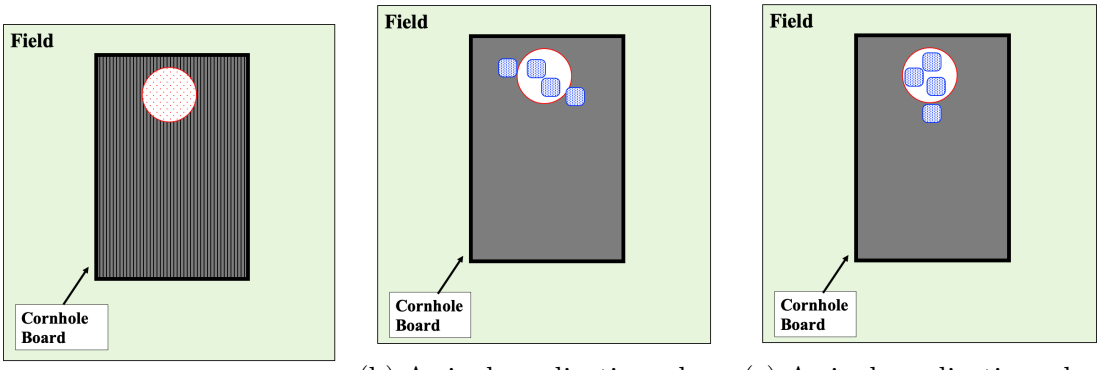
[99] R. Von Mises. On the foundations of probability and statistics. *The Annals of Mathematical Statistics*, 12(2):191–205, 1941.

Appendices

Appendix A

Random Variable Example

For example, let us consider a game of cornhole with a professional player, Jerry, throwing bags. Our reasonable assumption with a professional player is their throwing form is consistent. If you are unfamiliar with the game of cornhole, Figure A.1a illustrates the anatomy of a cornhole board where the objective is to throw all four cornhole bags (blue squares) through the hole (red circle) at the top of the board. However, when we watch Jerry throw the bags, they do not always make it into the hole due to unknown factors.. Figure A.1b and Figure A.1c represent two potential outcomes of a single cornhole game.



(a) An illustration of cornhole board.

(b) A single realization where two bags are and are not in the hole.

(c) A single realization where three bags are in the hole, and one is not.

Figure A.1: Outline of cornhole board and two potential realizations of cornhole game.

In this example, we could define our random variable, R, to be the number of cornhole bags that Jerry makes into the hole (i.e.,$R = \{0, 1, 2, 3, 4\}$) [discrete example]. With a Monte

Carlo simulation, we could calculate several aspect of our random variable such the probability distribution of R, the expected value of R (long-run average), or the variance of R. In order to calculate any of these statistics, we would continually throw four bags and track the number of bags in the hole per game. Then, after a long and finite period of time, we calculate the respective statistics of interest. For instance, we calculate the probability distribution of seeing a specific event by counting the frequency of observing zero, one, two, three, and four bags and dividing each total by the number of games played. Additionally, we could have compute the expected value of our random variable by averaging all the games results. In this example, we illustrate the concept of a random variable and a modern hands-on Monte Carlo approach to sampling from a probability distribution.

Appendix B

Principal Component Analysis Matrix Representation

This section illustrates the full matrix representation for $\Lambda\Lambda_k$, and Λ_k^{-1} where $\lambda_1 > \lambda_2 > \cdots > \lambda_k > \cdots > \lambda_P$ from Eq. 1.2.

$$\Lambda_{PxP} = \Lambda_P = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_k & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_P \end{bmatrix} \quad \Lambda_{P-k} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{k+1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_P \end{bmatrix}$$

$$\Lambda_k = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_k & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \Lambda_k^{-1} = \begin{bmatrix} 1/\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/\lambda_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\lambda_k & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Appendix C

Principal Component Analysis investigation into the number of k-modes

In Chapter 1, we briefly discussed utilizing Principal Component Analysis as anomaly detection methodology where we construct the response surface and its uncertainty bands utilizing Eq. 1.3:

$$\mathbb{E}[\underline{x}_{\text{new}}] = \hat{\Sigma}(\mathbf{Q}\Lambda_k^{-1}\mathbf{Q}')\underline{x}_{\text{new}},$$

and Eq. 1.4:

$$\text{Cov}[\underline{x}_{\text{new}}] = \mathbf{Q}(\Lambda - \Lambda_k)\mathbf{Q}',$$

where $\hat{\Sigma}$ denotes the estimated covariance for the data, $\mathbf{Q}$, represents the orthonormal matrix of eigenvectors associated with the Λ matrix which orders the eigenvalue magnitude from $\lambda_1 > \lambda_2 > \cdots > \lambda_p > \cdots > \lambda_P$.

In our simulation and case studies, we picked the number of modes using an eigenvalue percent variation technique. This technique chooses the value of k such that the sum of

the first k eigenvalues divided by the total sum of the eigenvalues is equivalent to a desired percentage of variation.

In the following simulation study, we generate 200 datasets from the Modified Cauchy Net (described in Section 5.2) at various observation sizes ($N = \{50, 100\}$), dimensionalities ($P = \{10, 20\}$) with probability a sensor is anomalous is 0.05 (i.e., $1 - \rho$). We evaluate the Principal Component Analysis at three eigenvalues percent set values of 85%, 90%, and 95% to investigate the effect of adjusting this tunable parameter. We use the average Type 1 and Type 2 errors of the 200 generated datasets to compare the three different settings. Table 3 and 4 demonstrate two scenarios when $N = 100$ and $P = 10$ and $P = 20$, respectively, illustrating the overall trend in the other simulation case studies. We notice the overall trend that as the eigen-percentage decreased (95 to 85), the average Type 1 and Type 2 error rates decreased. We utilize Figure C.1 and C.2 to show and explain this trend visually.

Table #3: Evaluation of overall error rates for when generated data comes from Modified Cauchy Net with N = 100 and P = 10.

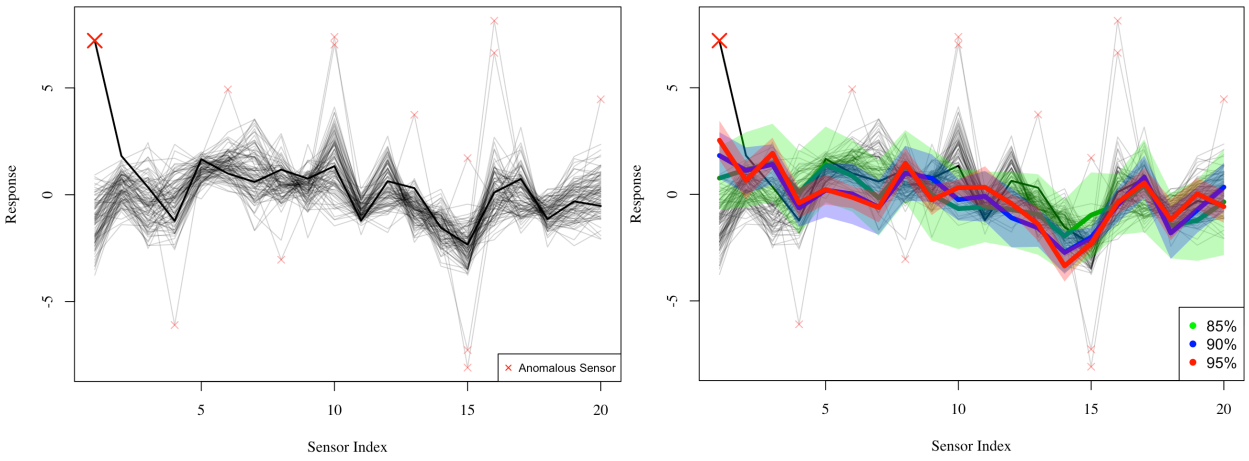
Method	PCA -85%	PCA - 90%	PCA - 95%
Average Type 1 Error	0.098	0.126	0.171
Average Type 2 Error	0.011	0.016	0.025

Table #4: Evaluation of overall error rates for when generated data comes from Modified Cauchy Net with N = 100 and P = 20.

Method	PCA -85%	PCA - 90%	PCA - 95%
Average Type 1 Error	0.062	0.089	0.14
Average Type 2 Error	0.007	0.011	0.017

In Figure C.1a and C.2a, we generated our training data and new observation with N = 100; P

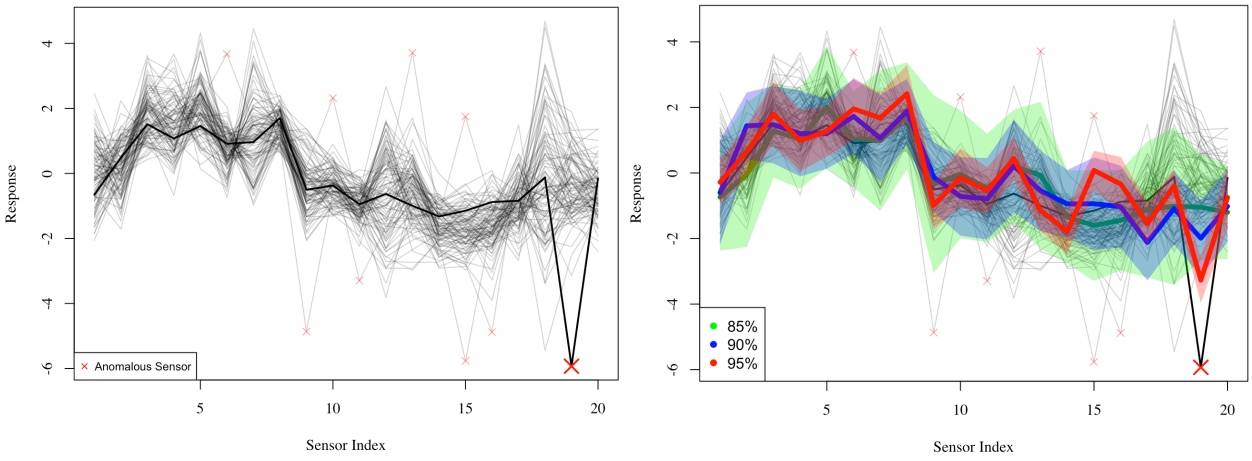
$= 20$ and $1 - \rho = 0.005$ where we use the bolded black line to represent the new observation and the faded lines to represent the training data. Figure C.2b and C.2b illustrates the expected mean and uncertainty bands for each of the percentages 85% (green), 90% (blue), and 95% (red) given its respective data. As the eigenpercentage decreases (95 to 85), we notice that the expected mean gets less perturbed by the anomalous observation and the uncertainty bounds increase. In the higher eigenpercentage response surfaces, the expected mean is effected by the anomalous observation because of the formulation of the expected mean of new observation given the training data, i.e., Eq. 1.3.



(a) Generated data

(b) Response surface

Figure C.1: An example of generated data and its corresponding PCA response surfaces under different eigenpercentages values.



(a) Generated data

(b) Response surface

Figure C.2: An example of generated data and its corresponding PCA response surfaces under different eigenpercentages values.

Appendix D

Distributional properties

D.1 Wishart and Inverse Wishart Distribution Properties

Let $\mathbf{Z}_P \sim \text{Wishart}(\nu, \mathbf{C})$ where ν represents the degrees of freedom and $\nu > P-1$ and $\mathbf{C}$ is a positive-definite PxP scale matrix.

$$\mathbb{E}[\mathbf{Z}] = \nu\mathbf{C} \quad ; \quad \mathbb{V}[\mathbf{Z}_{ij}] = \nu\left(c_{ij}^2 + c_{ii}c_{jj}\right) \tag{D.1}$$

Let $\mathbf{Q}_P \sim \text{Inverse Wishart}(\psi, \Omega)$ where ψ represents the degrees of freedom and $\psi > P-1$ and Ω is a positive-definite PxP scale matrix.

$$\mathbb{E}[\mathbf{Q}] = \frac{1}{\psi - P - 1}\Omega \tag{D.2}$$

D.2 Modified Cauchy Net Sampling Scheme

Consider the problem where we have 4 sensors (i.e., $P = 4$) with the following mean and covariance structure to represent the process's signal

$$\underline{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{bmatrix}$$

Now, for simplicity, let us say we have three observations (i.e, $N = 3$) and the following $\mathbf{S}$ matrix represents the sensor classification for each run.

$$\mathbf{S} = \begin{bmatrix} s_{1\bullet}^T \\ s_{2\bullet}^T \\ s_{3\bullet}^T \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Our first observation, $\underline{x}_{1p}$, has $s_{1\bullet}^T = \langle 1, 1, 1, 1 \rangle$ to represent all non-anomalous sensors. Thus, we generate the data from

$$\underline{x}_{1p} \sim \text{MVN}\left(\underline{\mu}, \Sigma\right)$$

The second observation, $\underline{x}_{2p}$, has $s_{2\bullet}^T = \langle 1, 1, 0, 1 \rangle$ which represent the third sensor is anomalous. Thus, our sampling scheme for the non-anomalous observations would follow from Eq. 5.4 as:

$$\underline{x}_{2,p\subset\{1,2,4\}} \sim \text{MVN}\left(\underline{\mu}_{\{1,2,4\}}, \Sigma_{\{1,2,4\}}\right)$$

$$x_{2,p\subset\{1,2,4\}} \sim \text{MVN}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_4 \end{bmatrix}, \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{24} \\ \sigma_{41} & \sigma_{42} & \sigma_{44} \end{bmatrix}\right)$$

While the anomalous observation follows Eq. 5.5 as

$$x_{2,p\subset\{3\}} \sim \text{Non-Local}$$

As another example, the third observation, $\underline{x}_{3p}$, has $s_{3\bullet}^T = \langle 1, 0, 0, 1\rangle$, thus the observation would follow similar sampling scheme as $\underline{x}_{2p}$.

$$x_{3,p\subset\{1,4\}} \sim \text{MVN}\left(\begin{bmatrix}\mu_1\\ \mu_4\end{bmatrix}, \begin{bmatrix}\sigma_{11} & \sigma_{14}\\ \sigma_{41} & \sigma_{44}\end{bmatrix}\right)$$

$$x_{3,p\subset\{2,3\}} \sim_{\text{iid}} \text{Non-Local}$$

where iid stands for independently and identically distributed.