ABSTRACT

HOLLIS, ANDREW NATHAN. Statistical Methods for Nonproliferation and Counterproliferation. (Under the direction of Alyson Wilson and Ralph Smith).

With the advent of nuclear weapons in the middle of the 20th century, the United States and other foreign powers have pursued nuclear nonproliferation and nuclear counterproliferation policies. Nuclear nonproliferation and counterproliferation efforts focus on a wide variety of issues including preventing the spread of nuclear materials and nuclear weapons technology to countries that do not yet have them and neutralizing or mitigating threats posed by those who have illegitimately obtained or are in the process of obtaining nuclear materials or a nuclear device. The aim of this dissertation is to propose statistical methods that can address technical challenges in implementing nonproliferation and counterproliferation policy. We focus on the problems of urban nuclear source localization, arms-control monitoring, and proliferation detection in dual-use scientific research (research the can be used for military or civilian applications).

Effective urban source localization requires constructing a configuration of radiation detectors that is optimal for estimating the source location. We propose constructing a Gaussian process surrogate model of the mutual information design criterion used for configuration selection. We demonstrate that optimizing this inexpensive surrogate model leads to effective source localization over the space of all possible configurations for a simulated source localization scenario in downtown Washington, DC. For the arms control monitoring problem, we propose a novel Bayesian hidden Markov model that represents a concealed arms-development process being monitored as well observable data emitted by the process. Our model reparameterizes the HMM transition probabilities in terms of activity duration and allows us to answer key questions about the monitored process. We also propose a novel framework for incorporating subject matter expertise about the process and potential noise into the inference. We demonstrate the proposed approach on an underground explosives test case study. In addressing the dual-use technology problem, we propose two new variational inference loss functions for fitting a deep learning text classification model with uncertainty quantification. The framework is computationally tractable for large models and meets important uncertainty quantification objectives including producing predicted class probabilities that reflect our prior conception of how different classes are related.

Statistical Methods for Nonproliferation and Counterproliferation

by
Andrew Nathan Hollis

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina
2023

APPROVED BY:

Alyson Wilson Co-chair of Advisory Committee	Ralph Smith Co-chair of Advisory Committee
Brian Reich	John Mattingly

Karl Pazdernik

DEDICATION

To my parents, Kendall and Kris Hollis

BIOGRAPHY

Andrew Hollis was born raised in Los Alamos, New Mexico. He attended the University of New Mexico as a Regents' Scholar and received his bachelor's degree in statistics with minors in computer science and mathematics in spring 2018. During his undergraduate studies, he also completed four summer internships at Los Alamos National Laboratory in the Global Security Directorate. He began the PhD program in Statistics at North Carolina State University in August of 2018, and received his Masters of Statistics in December of 2020. While at NC State, he has conducted research in collaboration with the Laboratory for Analytical Sciences, a research lab focused on building analytical tools for the intelligence community, the Consortium for Nonproliferation Enabling Capabilities, and West Point. He has had opportunities to complete two internships with the Department of Defense including an internship with the Air Force at the Pentagon in the summer of 2022.

ACKNOWLEDGEMENTS

I know that having good advisors can make all the difference in a student's experience of a PhD program. I want to give very special thanks to both my advisors Dr. Alyson Wilson and Dr. Ralph Smith for striking an excellent balance between giving me independence and giving me guidance. I am thankful for their patience, thoroughness, mentorship, encouragement, the many opportunities they have given me to pursue exciting work for my dissertation and as an RA, and the way in which they challenged me to be a better researcher and writer. I have grown a lot through the experience of working on a PhD, and I know much of that is due to how they have guided me through the process. I want to thank my committee members, Dr. John Mattingly, Dr. Brian Reich, and Dr. Jonathan Stallrich, for their time and for the many comments, questions, and suggestions they have given me as I have been working on this dissertation. I want to thank colleagues at Pacific Northwest National Laboratory and Los Alamos National Laboratory especially Dr. Aaron Luttman, Dr. Boian Alexandrov, and Dr. Erik Skau for their feedback and conversations. I want to give special thanks to Dr. Karl Pazdernik of Pacific Northwest National Laboratory, who has served as a committee member and advisor for my last project, for his guidance and feedback and for giving me an opportunity to work with him on a really cool project! I also want to thank Dr. George Tompkins of Los Alamos National Laboratory who has been a mentor of mine since I was in high school and has given me opportunities to learn about and work in the defense and statistics world since I was a college student. I want to thank him for giving me the opportunity to work on another really cool project for my dissertation and for helping to make my interaction with Los Alamos a smooth one. I also want to thank some other statisticians whose mentorship and advice has been so valuable as I've been a grad student: Dr. Scott Vander Wiel and Dr. Isaac Michaud of Los Alamos National Laboratory and Dr. Stephen Vardeman of Iowa State University.

I want to thank the whole North Carolina State University Statistics department for contributing to such a great place to learn and develop as a statistician. I want to thank all the professors who taught me and challenged me, all the staff who answered my questions and made everything run smoothly, and all my classmates who helped create a collegial and supportive environment to learn in. I especially want to thank Nicholas Larsen, Joe Zhao, Chris Buccheri, and Jimmy Hickey for their friendship and encouragement going through this challenging time!

I want to thank all my friends in Raleigh who made life outside of PhD work a joy: Richard

Bethencourt, Hawes Collier, Will Vandorp, John Gartrell, John Feather, John Lee, Emmett LaHay, Kyle Phillips, Lily Harris, Ethan Poppe, Levi and Autumn Manring, and many others. Thanks to all my friends at Christ the King Presbyterian Church especially the McKinney family, the Manschot family, the Rench family, and the Hall family. Thanks especially to Dr. Steve Hall for his advice on PhD life in general.

I owe a big debt of gratitude to my family, my brother Jacob and sister-in-law Anna Kat, and all the extended family who prayed for me and encouraged me while I've been on this journey. I owe a lot to my parents who encouraged me and prayed for me so consistently and were a sounding board and a useful source of advice when I was facing important decisions or felt like I wanted to throw in the towel. Thank you for loving me and supporting me through this PhD process.

Finally, I give thanks to the all-knowing and almighty God, who the Reverend Thomas Bayes acknowledges as "the object of our most delightful contemplation as well as our highest reverence"[1]. Without his grace and mercy on my life, this accomplishment would not have been possible or worthwhile.

Note: This work has been reviewed and approved for unlimited release by Pacific Northwest National Laboratory and has been assigned the release number PNNL-34077.

[1] Bayes, Thomas. Divine Benevolence: Or, An Attempt to Prove that the Principal End of the Divine Providence and Government is the Happiness of His Creatures: Being an Answer to a Pamphlet, Entitled, Divine Rectitude [by John Balguy]... With a Refutation of the Notions Therein Advanced Concerning Beauty and Order, the Reason of Punishment, and the Necessity of a State of Trial Antecedent to Perfect Happiness. J. Noon, 1731.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

1.1 Deterence, Nonproliferation, and Counterproliferation

Early on the morning of July 16, 1945, the desert of southern New Mexico was lit up with a sudden flash of light that one observer described as being "brighter than 20 suns and the most spectacular sunrise ever seen."[1] This was no sunrise however. The sudden flash of light was the result of the world's first successful atomic weapons test, ushering in the Atomic Age. In the decades following, nuclear weapons would become one of the primary tension points of the Cold War arms race. Today, nuclear weapons are still one of the most important global security challenges facing our world.

For a nuclear weapons power like the United States, the strategy for addressing the nuclear weapons challenge can be split into two major components: strategic deterrence and nuclear nonproliferation/counterproliferation. In general, deterrence refers to a strategy aimed at discouraging other entities from crossing certain red lines in the geopolitical, technological, or military spaces. Deterrence is often primarily enforced by the military, but it can also involve other government agencies as well as partners and allies. Nuclear

[1] BURROUGHS, Chris. "Trinity Test 'Brighter than 20 Suns and the Most Spectacular Sunrise Ever Seen,' Says Ben Benjamin." LabNews, Sandia National Laboratories, 3 Nov. 2000

weapons play an important role in deterrence through strategic deterrence. Strategic deterrence efforts focus on ensuring that a nation's nuclear weapons stockpile poses a credible threat to its adversaries, thus deterring the adversary from using weapons. Deterrence efforts include modernizing weapon capabilities and delivery systems, developing robust anti-weapon defense systems, extending nuclear protections to non-nuclear allies, and ensuring the reliability and security of the nuclear weapons stockpile.

Anderson et al. (2014) define nonproliferation as "the means and methods for preventing the acquisition, transfer, discovery, or development of materials, technology, knowledge, munitions/devices or delivery systems related to weapons of mass destruction." Weapons of mass destruction (WMD) include chemical, biological, radiological, and nuclear weapons. Nuclear nonproliferation policy is established by laws and international treaties, such as the Treaty on the Non-Proliferation of Nuclear Weapons, signed by the United States and other foreign powers in 1968. For these treaties to be effective, compliance with nonproliferation agreements must be enforced. Individual nations like the US develop compliance verification technologies as part of their nuclear security strategy. Nonproliferation policy is also enforced internationally by agencies like the International Atomic Energy Agency (IAEA), who develop safeguards to ensure that nuclear technology and materials meant for civilian applications are not used for weapons applications.

Anderson et al. (2014) define counterproliferation as "efforts and initiatives aimed at (1) directly forestalling, rolling back, or eliminating efforts to proliferate WMD, and (2) preventing a WMD-armed actor from realizing any benefit from owning or employing these weapons."Nuclear counterproliferation efforts include building systems for intercepting nuclear smugglers, developing capabilities for detecting and locating potentially threatening nuclear devices, preventing acts of terrorism involving nuclear devices, and preparing for nuclear incidents.

Implementing and enforcing nuclear nonproliferation and counterproliferation policies raises a variety of technical challenges. This dissertation addresses some specific statistical problems that arise in nonproliferation and counterproliferation settings.

1.2 Challenges of Nuclear Nonproliferation and Counterproliferation Implementation

Both nonproliferation and counterprolifertion require the ability to detect, monitor, and neutralize potential threats related to the use or development of nuclear weapons. Doing this successfully often involves gathering data from a set of detectors and sensors and then properly analyzing this data to assess the threat. The gathered data could be very diverse and might include radiation detector readings, seismographic sensors, inspection data, wide-area environmental samples, images, videos, texts, and any other source of data that may be useful in assessing a proliferation threat.

There has been significant technical research effort focused on developing better sensors and data gathering technology to ensure that we obtain the most high quality data to feed into threat analysis (Runkle 2011; Satkowiak 2014). But, as the data we collect has become larger, more complex, and more diverse, agencies responsible for nonproliferation and counterproliferation implementation have recognized the need for more rigorous technical approaches to analyzing the data.

In many cases, the collected data may be contaminated with noise, may contain measurement errors due to sensor imperfections, or may be intentionally spoofed by the adversary so that the signal that may indicate threatening activity is obscured. In addition, many of the indicators of proliferation activity, the presence of special nuclear material, scientific interest in nuclear energy science, nuclear fuel refinement capabilities, etc., are also common in civilian nuclear applications. This is known as the dual-use problem. These issues introduce uncertainty into the conclusions we draw from the data. Many authors have already recognized the importance of uncertainty quantification (UQ) when making nonproliferation/counterproliferation decisions from data (Bonner et al. 2016; Gauld et al. 2016; Unwin and Fecht 2009).

Another challenge in making data-based decisions for nonproliferation or counterproliferation is that the data often comes in many forms from many sources (Hruby 2020; Gastelum et al. 2019). Thus, approaches to analysis must accommodate many different kinds of data including environmental readings, images, texts, seismographic readings, human observations, etc. For some scenarios, we receive data from multiple sources of the same type (data from a network of several identical sensors) or from multiple sources of different types (mixture of image data and seismic sensor data). This data must then be combined in a principled way (Hruby 2020). In many cases, the data can come in the form of subject matter expertise. This is especially common in scenarios where more traditional data types are scarce. Thus, analysis approaches may need to accommodate the incorporation of such subject matter expertise (White et al. 2014; Gillen 2014).

1.3 Applying Statistics to Nonproliferation and Counterproliferation

Statistical science provides methodology for rigorous data analysis and can address many of the challenges mentioned in Section 1.2. For this reason, many government agencies responsible for nonproliferation and counterproliferation analysis and decision making have an interest in developing sophisticated data analysis techniques. In 2018, the National Nuclear Security Administration (NNSA), the nuclear security arm of the Department of Energy (DOE), launched the Advanced Data Analytics for Proliferation Detection (ADAPD) project which tasked five DOE laboratories with developing data science solutions to address key nonproliferation challenges. The research discussed in Chapters 3 and 4 of this dissertation was inspired by work being performed under this project. NNSA has also sponsored multiple university consortia to examine scientific issues related to nuclear science and nuclear security to benefit nonproliferation and counterproliferation efforts. The research discussed in the second chapter of this dissertation was part of work done by the Consortium for Nonproliferation Enabling Capabilities (CNEC), an NNSA consortium led by North Carolina State University.

This dissertation develops statistical methodology that addresses some of the challegnes that arise in three nuclear nonproliferation/counterproliferation problems. We introduce each of these problems and summarize our contributions in the following sections.

1.3.1 Urban Source Localization for Counterproliferation

A key nuclear counterproliferation task is detecting and locating potentially threatening nuclear devices so that these devices can be captured and disarmed. This task is especially important and especially challenging in urban environments where background noise and objects can obscure the radiation coming from the device. Radiation source localization focuses on estimating a radiation emitting source's location and is accomplished by constructing a configuration of multiple radiation detectors located throughout the urban environment, gathering data from these detectors, and then using statistical methods to combine the collected data and infer the source's location. A Bayesian inference framework is often employed for performing uncertainty quantification for the source location (Hite et al. 2019; Ştefănescu et al. 2017).

Source localization performance is sensitive to the location of the detectors in the detector configuration. There are criteria that can be used to select an optimal detector configuration for source localization (Michaud 2019; Ştefănescu et al. 2019), but these criteria are generally difficult to optimize because of the compute time associated with evaluating the criteria for each configuration. In Chapter 2, we propose constructing a surrogate model of the design criterion used for configuration selection and show that optimizing this inexpensive surrogate model leads to effective source localization and allows for tractable optimization over the contiuous detector configuration space. We demonstrate our approach on a simulated source localization scenario in downtown Washington, DC.

1.3.2 Arms-Control Monitoring with Bayesian Hidden Markov Models

An important task in nonproliferation enforcement is monitoring processes which may be part of illicit arms development efforts. Accomplishing this monitoring can be very challenging as it is rare to have direct knowledge of the development activities being conducted. Most often, analysts have access to one or several streams of data emitted by the process being monitored. However, these data streams often contain gaps, are noisy, and can be complex to interpret. Analysts also have access to subject matter experts whose knowledge of the process is valuable in making inferences from the data.

In Chapter 3, we propose a novel modelling framework based on Bayesian hidden Markov models, which facilitates the incorporation of subject matter expertise, models the relationship between emitted data and unobserved process activities, and is parameterized in such a way as to facilitate answering key process-related questions of interest to national security decision-makers. We demonstrate how the proposed approach can be used to answer process modelling questions with well-calibrated uncertainty quantification using a real process modeling case study developed by Los Alamos National Laboratory.

1.3.3 Uncertain Text Classification for Proliferation Detection

An important indicator of potential proliferation activity is the research being performed by scientists. By analyzing the work published by scientists and by examining the collaborators in a scientist's network, we can gain insight into whether or not the research work may indicate proliferation activity. Because a significant portion of research work performed in the nuclear energy field can be dual-use, it is important that we have well-calibrated uncertainty quantification for the conclusions we draw from our analysis.

One approach to building an inference system designed for the above task involves using a statistical model that can classify scientific documents to one of several scientific research categories, some of which may raise some proliferation concerns. Typically, the classifier employed is a large deep learning model. To effectively incorporate the results of the classification into downstream analysis, we must have uncertainty quantification for the classification. Unfortunately, the size of most state-of-the-art document classification models makes uncertainty quantification difficult (Yu et al. 2022). In addition, many state-of-the-art uncertainty quantification approaches for deep learning do not always produce uncertainty quantification that are well-calibrated and understandable by decision-makers (Tong et al. 2019; Gustafsson et al. 2020). In Chapter 4, we propose two new variational inference loss functions for fitting a deep learning models with uncertainty quantification. The framework is computationally tractable for large models and meets important uncertainty quantification objectives including producing predicted class probabilities that reflect our prior conception of how different classes are related.

CHAPTER

2

SURROGATE BASED MUTUAL INFORMATION APPROXIMATION AND OPTIMIZATION FOR URBAN SOURCE LOCALIZATION

2.1 Introduction

Efforts to intercept smuggled nuclear materials, prevent nuclear terrorism, and recover hazardous radiological sources often rely on accurate and efficient methods for radiation source detection and localization. Hence, the development, implementation, and experimental validation of mathematical and statistical algorithms for detecting and locating nuclear devices in urban environments constitutes a critical nuclear security and nonproliferation problem.

Radiation source detection is the process of determining the presence or absence of a radiation source in a given area, whereas radiation source localization is the estimation of the location and intensity of one or more sources that are assumed to be present. Certain approaches to localization and detection require computing the response induced at a detector by a source. Because of the computational cost associated with computing detector responses, surrogate models are employed to approximate detector responses or functions of detector responses (Cook et al. 2019; Miles et al. 2021). In this chapter, we present an approach that optimizes surrogate models of design criteria to find optimal configurations of radiation detectors that will yield precise localization results for sources in urban environments. Whereas some authors address both detection and localization (Morelande et al. 2007; Rao et al. 2008; Vilim and Klann 2009), we will focus on the localization problem assuming a radiation source is present. We consider a single, fixed-location source, and refer readers to Chandy et al. (2008) and Vilim and Klann (2009) for moving sources and Morelande et al. (2007) for multiple sources.

Radiation source localization employs a configuration of radiation detectors to collect data used to estimate a source's location and intensity parameters. If there are no obstructions between the source and the detector and no noise or error in detector readings, a simple four detector method suffices to localize the source (Cox and Partensky 2007). Obstructions or sources of random noise in the environment induce error and variation in detector measurements making it necessary to fuse data collected from several detectors to perform localization accurately.

In the absence of obstructions, Chin et al. (2008) introduce methods that cluster the candidate source locations obtained using the ratios of squared distances computed from several three detector subsets to identify the true source location. Another approach is to treat detector measurements as random variables generated by a statistical model parameterized by the source parameters. Maximum likelihood estimation (MLE) can be used to estimate the source parameters (Baidoo-Williams et al. 2013; Gunatilaka et al. 2007; Rao et al. 2008). While MLE methods provide point estimates for the source parameters, it can be difficult to obtain exact or asymptotic variances for these estimates, making uncertainty quantification of these estimates challenging. This has led many investigators to employ Bayesian inference methods, which provide a posterior distribution for source parameters that can be used to easily compute point estimates for the source parameters and quantify the uncertainty of these estimates (Hite et al. 2019; Jarman et al. 2011; Ştefănescu et al. 2017). We will employ a Bayesian formulation for source localization.

Radiation source localization is most important, and often most challenging, when the source is in an urban environment. The detonation of a nuclear device in a densely populated urban environment could cause major loss of life, serious infrastructural damage, and severe health hazards. Lost or stolen medical radiation sources or misplaced radioactive waste can also pose health threats to an urban population. Various characteristics of the urban environment, however, make source localization particularly difficult. In contrast to the noise free, obstruction free scenario, buildings with varying materials as well as moving objects such as vehicles can obstruct the source. The presence of background radiation, which may vary in time and space within a city, also introduces noise that may be difficult to separate from radiation emitted by the source of interest.

To account for fixed position attenuating objects, like buildings, that may block the path between the detector and the source, Hite et al. (2019) and Ştefănescu et al. (2017) employ a simplified version of the Boltzmann transport model for modeling detector response. To account for background radiation, Penny et al. (2015) show that the environment can be partitioned so that each region has approximately uniform background. Alternatively, by employing a hierarchical Bayesian model, background radiation estimation and source localization can be accomplished simultaneously (Michaud et al. 2021). For simplicity, we treat background radiation observed at each detector as a Gaussian random variable with fixed parameters.

Another challenge of urban source localization is the sensitivity of the localization results to the employed configuration of the radiation detectors. In the Bayesian localization context, different detector configurations will produce different posterior distributions for the source parameters. An optimal configuration will produce a posterior distribution with minimal uncertainty. To find a detector configuration that minimizes uncertainty for frequentist estimates of the source location, Ştefănescu et al. (2019) propose finding a configuration that is D-optimal. Alternatively, in a Bayesian context, Michaud (2019) and Schmidt (2016) use mutual information to quantify the reduction in uncertainty for the source parameters achieved by knowing the responses observed at a detector configuration. They find an optimal detector configuration by maximizing mutual information. A challenge for both these approaches is the computational expense associated with finding an optimal configuration. To address this challenge, these previous works only optimized the design criterion over a finite set of candidate detector locations. Ştefănescu et al. (2019) also employed a surrogate model of the detector response model proposed in Ştefănescu et al. (2017) to further reduce computational costs. The primary disadvantage of this approach is that the ideal configuration may not exist as a combination of some fixed set of possible detector locations.

Instead of computing mutual information explicitly for each detector configuration, we propose an approach for maximizing mutual information over all possible detector configurations by constructing a surrogate model for the mutual information design criterion, rather than a surrogate model for detector response. We can find an optimal detector configuration by optimizing this surrogate model. This approach is computationally efficient and considers general detector configurations instead of only a finite subset of possible configurations. The generality of this approach is particularly important in larger urban environments or 3-dimensional space where small subsets of candidate detector locations may not be representative of the space.

The chapter is organized as follows: in Section 2.2, we discuss the Bayesian framework used for source localization, mutual information estimation, and the combinatorial optimization of mutual information demonstrated in Michaud (2019) and Schmidt (2016). In Section 2.3, we develop a continuous optimization approach using surrogate models, which, in Section 2.4, we compare to the performance of the combinatorial optimization approach for a localization problem with movable detector networks. We present concluding remarks in Section 2.5.

2.2 Bayesian Source Localization and Detector Configuration Selection

2.2.1 Bayesian Framework for Constructing Posterior Distributions of Source Parameters

Our objective is to estimate the source parameter vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$, where θ_1 and θ_2 are the two dimensional location coordinates of the source, and θ_3 is the source intensity. We assume the source is located in a bounded search space. We use the block of downtown Washington, D.C. described in Ştefănescu et al. (2017) as our search space.

To obtain estimates and perform uncertainty quantification for $\boldsymbol{\theta}$, we treat $\boldsymbol{\theta}$ as a random vector and compute the posterior distribution, $p(\boldsymbol{\theta}|\mathbf{Y})$, given the data $\mathbf{Y}$ where $\mathbf{Y} = (Y_1, ..., Y_n)$ are the radiation count measurements taken from a configuration of n detectors, $\mathbf{x} = \{d_{1j}, d_{2j}\}_{j=1}^{n}$, where d_{1j} and d_{2j} are the location coordinates of the j-th detector. To compute $p(\boldsymbol{\theta}|\mathbf{Y})$, we require the probability of the data given $\boldsymbol{\theta}$, $p(\mathbf{Y}|\boldsymbol{\theta})$, and a prior distribution for $\boldsymbol{\theta}$, $p(\boldsymbol{\theta})$. We employ delayed rejection adaptive Metropolis (DRAM) (Haario et al. 2006) Markov Chain Monte Carlo (MCMC) methods to sample $p(\boldsymbol{\theta}|\mathbf{Y})$.

In specifying the $p(\mathbf{Y}|\boldsymbol{\theta})$ and $p(\boldsymbol{\theta})$, we will describe the process we use to generate detector response data $\mathbf{Y}$ for a set of radiation sources. In practice, $\mathbf{Y}$ would be collected from real detectors. Instead of using real detector data as in Hite et al. (2019) or expensive, high-fidelity radiation transport codes (Briesmeister 2000), we have simulated detector response using a statistical framework employing a simplification of the Boltzmann transport model. All detector response data used in this chapter will be generated using this framework. The employed framework is very similar to the one used in Cook et al. (2019), Michaud (2019), and Ştefănescu et al. (2017).

We begin by drawing N parameter vectors $\boldsymbol{\theta}$ from the source parameter prior distribution, $p(\boldsymbol{\theta})$,

$$p(\theta_s) = \frac{I(\theta_{s,\min} < \theta_{is} < \theta_{s,\max})}{\theta_{s,\max} - \theta_{s,\min}}, s = 1,2,3 \tag{2.1}$$

$$p(\boldsymbol{\theta}) = p(\theta_1) \cdot p(\theta_2) \cdot p(\theta_3). \tag{2.2}$$

Here $\theta_{s,\min}$ and $\theta_{s,\max}$ are respectively the smallest and largest values that θ_s can take for $s = 1,2,3$. $I(\cdot)$ is an indicator function. Notice that the source parameters are assumed to be independent a priori. Because our search space is 246.615 m × 176.333 m, we set $\theta_{1,min} = 0$, $\theta_{1,max} = 246.615$, $\theta_{2,min} = 0$, $\theta_{2,max} = 176.333$. We set $\theta_{3,min} = 10^9$ counts per second (cps), and $\theta_{3,max} = 5 \times 10^9$ (cps).

After generating N source parameter vectors, we evaluate the expected detector response due to the radiation source for each of the N sources at n detectors. The expected detector response γ_i due to a source located at (θ_1, θ_2) with intensity θ_3 for detector i located at $\mathbf{d}_i = \{d_{i1}, d_{i2}\}$ is calculated using an approximation to the Boltzmann transport model (Ştefănescu et al. 2017),

$$\gamma_i = \theta_3 \cdot \Delta t_i \cdot \epsilon_i \cdot \frac{A_i}{4\pi||\mathbf{d}_i - (\theta_1, \theta_2)||_2^2} \exp(-\sum_{p=1}^{N_\ell} \ell_p \cdot \Sigma_T^{(p)}). \tag{2.3}$$

Here ϵ_i is the efficiency of the detector, and A_i is the face area of the detector. We assume all the detectors are the same and have the same dwell time, Δt_i, so ϵ_i, A_i, and Δt_i, will not vary with i. Here N_ℓ is the number of buildings in the bounded search space intersected by a ray from the source to the detector, ℓ_p is the length of the portion of the ray intersecting building p, and $\Sigma_T^{(p)}$ is the macroscopic cross-section of building p. We note that γ_i quantifies uncollided flux, the flux of photons on the path from the source to the detector that are not absorbed or scattered.

To generate the observed radiation count at detector i, we draw B_i from a Gaussian distribution with mean 140.8 and variance 9.61, multiply it by Δt_i, and add this to γ_i to get

$$\lambda_i = \gamma_i + B_i \cdot \Delta t_i,\ B_i \sim N(140.8, 9.61).$$

Here B_i simulates the background radiation in the urban environment in counts per second. The mean and variance are based on radiation concentration measurements collected from a site replicating a small urban environment at Fort Indiantown Gap National Guard Training Center (Swinney et al. 2016). The concentration measurements were used to compute detector responses with the Gamma Detector Response and Analysis Software Detector Response Function (GADRAS-DRF) (Horne et al. 2014). These mean and variance parameters were also used in Michaud (2019).

Observed radiation counts are often treated as Poisson random variables (Knoll 2010), with the Poisson accounting for detector measurement error. Thus, after computing λ_i, the expected radiation count due to the radiation source and the random radiation count due to background, we draw the radiation count observed at detector i from a Poisson distribution with λ_i as the mean parameter. The probability of the data observed at n detectors, given θ and B_i is

$$p(\mathbf{Y}|\boldsymbol{\theta}, \mathbf{B}) = \frac{e^{-\sum_{i=1}^{n} \lambda_i} \prod_{i=1}^{n} \lambda_i^{y_i}}{\prod_{i=1}^{n} y_i!}. \tag{2.4}$$

2.2.2 Mutual Information Estimation

To select an optimal configuration $\mathbf{x}^{\text{opt}}$, at which to collect the detector data, we maximize the mutual information between the source parameters $\boldsymbol{\theta}$ and the data collected at a particular detector configuration $\mathbf{Y}$. We denote this quantity as $MI(\boldsymbol{\theta}, \mathbf{Y})$.

The mutual information between continuous random vectors $\mathbf{Y}$ and $\boldsymbol{\theta}$ is

$$MI(\boldsymbol{\theta}, \mathbf{Y}) = \int_{\boldsymbol{\theta}} \int_{\mathbf{Y}} p(\boldsymbol{\theta}, \mathbf{Y}) \log\left(\frac{p(\boldsymbol{\theta}, \mathbf{Y})}{p(\boldsymbol{\theta})p(\mathbf{Y})}\right) d\boldsymbol{\theta}\, d\mathbf{Y}, \tag{2.5}$$

or equivalently,

$$MI(\boldsymbol{\theta}, \mathbf{Y}) = H(\boldsymbol{\theta}) - H(\boldsymbol{\theta}|\mathbf{Y}),\ H(\boldsymbol{\theta}) = -\int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}) \log p(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{2.6}$$

Here $H(\boldsymbol{\theta})$ is the Shannon entropy of $\boldsymbol{\theta}$, a measure of the uncertainty of $\boldsymbol{\theta}$. Thus, mutual information between $\boldsymbol{\theta}$ and $\mathbf{Y}$ is the total uncertainty of $\boldsymbol{\theta}$ minus the uncertainty of $\boldsymbol{\theta}$ given we know $\mathbf{Y}$.

In general, (2.5) cannot be evaluated analytically. Consequently, many estimators of mutual information have been proposed to estimate this quantity (Belghazi et al. 2018; Gao et al. 2015a,b; Kraskov et al. 2004). We use the Kraskov-Stogbauer-Grassberger (KSG) k-Nearest Neighbors estimator (Kraskov et al. 2004). Although biased, the KSG estimator is straightforward and efficient to compute and has been shown to perform well in selecting optimal configurations based on the maximization of the mutual information criterion (Michaud 2019).

The KSG mutual information estimator, employing N draws from $p(\boldsymbol{\theta})$, $\{\boldsymbol{\theta}_i\}_{i=1}^N$, and the corresponding realizations of data, $\{\mathbf{Y}_i\}_{i=1}^N$, from some detector configuration is

$$\widehat{MI}(\boldsymbol{\theta},\mathbf{Y}) = \psi(N)+\psi(k)-\frac{1}{N}\sum_{i=1}^{N}\psi(n_{\boldsymbol{\theta}}(\epsilon_{\boldsymbol{\theta}}(i)))-\frac{1}{N}\sum_{i=1}^{N}\psi(n_{\mathbf{Y}}(\epsilon_{\mathbf{Y}}(i)))-\frac{1}{k} \tag{2.7}$$

where

$$n_{\boldsymbol{\theta}}(\epsilon_{\boldsymbol{\theta}}(i)) = \sum_{j=1}^{N} I(||\boldsymbol{\theta}_j-\boldsymbol{\theta}_i||_\infty \leq \frac{\epsilon_{\boldsymbol{\theta}}(i)}{2}) \text{ and } n_{\mathbf{Y}}(\epsilon_{\mathbf{Y}}(i)) = \sum_{j=1}^{N} I(||\mathbf{Y}_i-\mathbf{Y}_j||_\infty \leq \frac{\epsilon_{\mathbf{Y}}(i)}{2}) \tag{2.8}$$

and

$$\frac{\epsilon_{\boldsymbol{\theta}}(i)}{2} = \max_{1\leq j\leq k}||\boldsymbol{\theta}_i^j-\boldsymbol{\theta}_i||_\infty \text{ and } \frac{\epsilon_{\mathbf{Y}}(i)}{2} = \max_{1\leq j\leq k}||\mathbf{Y}_i^j-\mathbf{Y}_i||_\infty. \tag{2.9}$$

Here $(\boldsymbol{\theta}_i^j, \mathbf{Y}_i^j)$ is the j-th nearest neighbor to $(\boldsymbol{\theta}_i, \mathbf{Y}_i)$ in the joint space for $\boldsymbol{\theta}$ and $\mathbf{Y}$, N is the number of $(\boldsymbol{\theta}_i, \mathbf{Y}_i)$ pairs in the data, and $\psi(\cdot)$ is the digamma function. The estimator uses the form of mutual information in (2.6). The average count of points in a ball with radius equal to a point's distance to its k-th nearest neighbor is employed to estimate entropy. We direct the reader to Kraskov et al. (2004) for a full derivation of this estimator and its connection to entropy.

$\widehat{MI}(\mathbf{Y},\boldsymbol{\theta})$ is a noisy function of detector configuration $\mathbf{x}$, requiring $N \times n$ evaluations of (2.3). The noise in $\widehat{MI}(\mathbf{Y},\boldsymbol{\theta})$ is due to the random background radiation detector readings, B_i, and the measurement error in $\mathbf{Y}_i$. To emphasize the dependence on $\mathbf{x}$, we denote $\widehat{MI}(\mathbf{Y},\boldsymbol{\theta})$ as $\widehat{MI}(\mathbf{Y}(\mathbf{x}),\boldsymbol{\theta})$. Our objective is to determine the detector configuration that maximizes $E(\widehat{MI}(\mathbf{Y}(\mathbf{x}),\boldsymbol{\theta}))$, which we denote as $g(\mathbf{x})$, a function of $\mathbf{x}$ because the expectation taken over $\mathbf{Y}$ eliminates the dependence on $\mathbf{Y}$. Thus our objective is

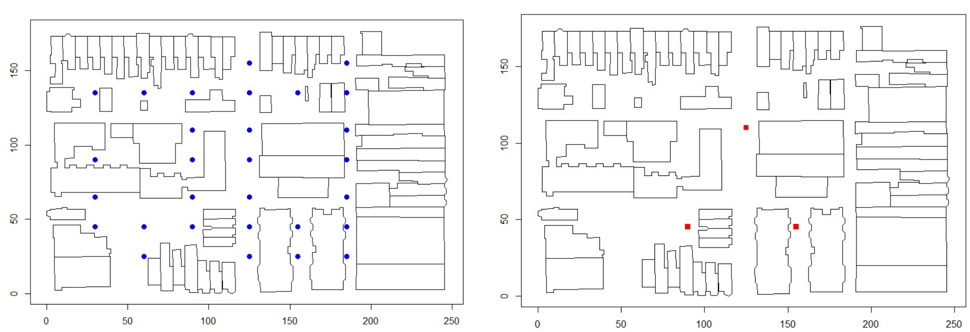
$$\mathbf{x}^{\text{opt}} = \underset{\mathbf{x}\in\mathscr{D}}{\operatorname{argmax}}\, g(\mathbf{x}). \tag{2.10}$$

Here $\mathbf{x} \in \mathscr{D}$ if and only if $\theta_{1,\min} \leq d_{1j} \leq \theta_{1,\max}$ and $\theta_{2,\min} \leq d_{2j} \leq \theta_{2,\max}$ for all $j = 1, ..., n$.

In general, solving (2.10) requires K evaluations of $\widehat{MI}(\mathbf{Y}(\mathbf{x}), \boldsymbol{\theta})$ and $N \times n \times K$ evaluations of (2.3), for K on the order of 10^4 or more to insure the space over $\mathbf{x}$ is thoroughly searched. Because of the cost of evaluating (2.3), (2.10) can be prohibitively expensive to solve over all $\mathscr{D}$.

2.2.3 Combinatorial Optimization of Detector Configurations

To facilitate the solution of (2.10), Michaud (2019) and Schmidt (2016) proposed maximizing $\widehat{MI}(\mathbf{Y}(\mathbf{x}), \boldsymbol{\theta})$ over a finite 2-D set $\{d_{1m}, d_{2m}\}_{m=1}^{p}$ instead of $\mathscr{D}$. This strategy is illustrated for our scenario in Figure 2.1. For this strategy, there are only p possible detector locations, and we only need to evaluate (2.3) $N \times p$ times. The data for any K configurations we wish to consider will be some subset of these $N \times p$ computations. Instead of using this combinatorial optimization approach, we will propose a new surrogate-model based approach that allows us to efficiently perform continuous optimization over the whole detector configuration space.



(a) Finite set of p detector locations

(b) Optimal 3-detector configuration

Figure 2.1: Optimization of detector configurations over finite subset

2.3 Continuous Optimization of Detector Configurations

2.3.1 Surrogate Based Mutual Information Approximation and Optimization

Rather than constrain $\mathscr{D}$ to some finite set, we could find an approximate solution to (2.10) by creating a surrogate model that approximates $g(\mathbf{x})$ and is less expensive to compute. We first describe a method for constructing sets of input-output pairs that can be used to fit these models. We then review a variety of popular surrogate models that can be used for our application and compare the performance of these models in capturing the relationship between mutual information and detector configuration.

For some fixed $\{\boldsymbol{\theta}_j\}_{j=1}^N$, let $z_i = \widehat{MI}(\mathbf{Y}(\mathbf{x}_i), \boldsymbol{\theta})$ be the KSG estimated mutual information corresponding to the ith n-length detector configuration, $\mathbf{x}_i$. We assume that

$$z_i = g(\mathbf{x}_i) + e_i. \tag{2.11}$$

Here $e_i \overset{iid}{\sim} N(0, \sigma^2)$ is noise due to random background radiation and the stochastic nature of the detector readings.

It is difficult to optimize $g(\mathbf{x})$ directly. We never observe it nor compute it directly; instead we only compute the noisy z_i estimates. Each z_i is expensive to generate. A single z_i estimate from 1000 data points for a 5-detector configuration can take 8-10 minutes to generate on a standard laptop computer. Rather than optimizing $g(\mathbf{x})$ directly, we will generate a set of outputs $\{z_i\}_{i=1}^M$ at a set of inputs $\{\mathbf{x}_i\}_{i=1}^M$, called design points, and use these design points to build a surrogate model $\hat{g}(\mathbf{x})$ that approximates $g(\mathbf{x})$. Surrogate models can be used to efficiently find approximate optima of expensive objective functions (Han et al. 2010; Queipo et al. 2005).

In this surrogate-based optimization framework, we replace $g(\mathbf{x})$ with $\hat{g}(\mathbf{x})$ to obtain the new optimization problem,

$$\mathbf{x}^{\text{opt}} = \underset{\mathbf{x} \in \mathscr{D}}{\operatorname{argmax}}\, \hat{g}(\mathbf{x}). \tag{2.12}$$

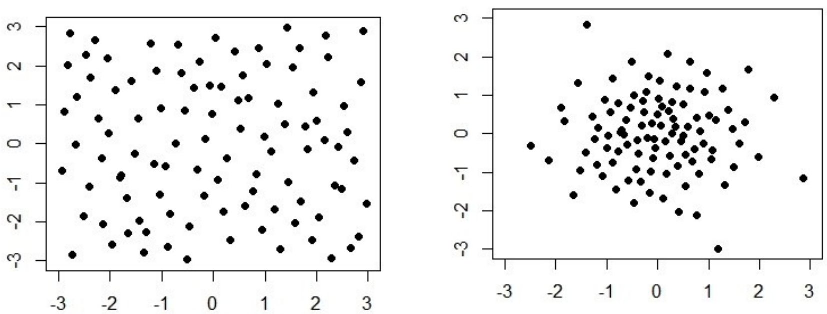
When the surrogate model is built only once for a fixed set of design points prior to optimization, the surrogate-based optimization is called the one-shot solution (Queipo et al. 2005). This is the solution method we will employ to perform detector configuration optimization.

2.3.2 Design Point Selection

To construct $\hat{g}(\mathbf{x})$, we require a set of M input-output pairs, $\{z_i, \mathbf{x}_i\}_{i=1}^{M}$, from (2.11). The set $\{\mathbf{x}_i\}_{i=1}^{M}$, called the design points set, should be space-filling, so that the design points are spread throughout the input space we are investigating to insure the surrogate model is accurate for the entire space. One popular space-filling design is the Latin Hypercube Design (LHD) (McKay et al. 1979). A LHD of size M is generated by treating each input variable as an independent random variable and then dividing each variable's range into M intervals having equal probability density. In other words, if U is an input variable with assumed distribution function $F_U(u)$, then a LHD selects interval boundaries $t_1 < \ldots < t_M$ in the range of U_k such that $F_U(t_1) = F_U(t_2) - F_U(t_1) = \ldots = F_U(t_M) - F_U(t_{M-1})$. After performing this procedure for all variables in the input, design points are selected so that every interval for each variable is represented in the design by exactly one design point.

A key advantage of using a LHD is that it allows us to manipulate the resolution at which different regions of the input space will be represented. If there are parts of the input space that we are less interested in, we can select distribution functions for the input variables that assign low probability to those regions and those regions will be sparsely sampled. Figure 2.2 shows the LHD from (a) two independent Uniform(-3,3) input variables and (b) two independent standard Gaussian random variables. Note that because the standard Gaussian has more probability mass concentrated near its mean of zero, the majority of the points in the Gaussian LHD are clustered in the box $[-1,1] \times [-1,1]$. The uniform LHD, however, distributes points evenly across the range of the two input variables. For our application, we wish to have more configuration design points with detectors in regions of the search space where the source location has high probability. To achieve this using the LHD method, we will use the marginal distribution functions of the source location coordinates as described in Stein (1987) for independent input variables. We denote the generated matrix of design points as $\mathbf{X}$.

Computing the mutual information for M design points, each with n detectors, requires evaluating (2.3) $N \times n \times M$ times, where N is the number of source locations used to approximate mutual information. We need $n \times M$ evaluations for each of the N source locations because each of the n detectors in the M design points are distinct and have distinct responses for each source. Instead of computing the response at the $n \times M$ distinct detectors in the design, we could reduce the computational cost by reducing the number of distinct detectors in the M point design to some $p < n \times M$, and then use the p distinct detectors to construct a design that approximates the original $n \times M$ design.



(a) LHD from two independent Uniform(-3,3) variables

(b) LHD from two independent Gaussian(0,1) variables

Figure 2.2: Uniform versus Gaussian Latin hypercube design

Specifically, we can generate a p point LHD across the space of possible detector locations: $P = \{(d'_{11}, d'_{21}), (d'_{12}, d'_{22}), ..., (d'_{1p}, d'_{2p})\}$ and compute the detector response at these locations in $p \times N$ model evaluations. We then approximate $\mathbf{X}$ by replacing each $(d_{1ij}, d_{2ij}) \in \mathbf{X}$ with $\underset{(d'_{1k}, d'_{2k}) \in P}{\arg\min} ||(d'_{1k}, d'_{2k}) - (d_{1ij}, d_{2ij})||_2$. This constructs the best element-wise ℓ_2-norm approximation to $\mathbf{X}$ with the elements of P. We denote this approximation as $\tilde{\mathbf{X}}$. Simulation studies show that while using $\tilde{\mathbf{X}}$ instead of $\mathbf{X}$ has a small impact on localization precision, it does not have a significant impact on localization accuracy, see Appendix A.

2.3.3 Surrogate Model Selection

We consider three surrogate models for $g(\mathbf{x})$: Gaussian processes (GP) (aka kriging), Bayesian Multivariate Adaptive Regression Splines (BMARS), and multi-quadratic radial basis functions (RBF). Each method is trained using M estimated mutual information values, $\{z_i\}_{i=1}^M = \mathbf{z}$, and a full LHD, $\mathbf{X}$, or an approximated LHD, $\tilde{\mathbf{X}}$.

Gaussian Processes

Gaussian processes are a class of non-parametric, nonlinear models. Consider an $M \times 1$ vector of KSG estimates, $\mathbf{z}$, and an $M \times 2n$ design matrix of detector configuration coordinates, $\mathbf{X}$. We assume that

$$z_i = g(\mathbf{x}_i) + e_i,\ e_i \overset{iid}{\sim} N(0, \sigma_e^2). \tag{2.13}$$

Gaussian processes approximate $g(\mathbf{x})$ for new configuration points $\mathbf{x}_i^*$ by constructing a conditional distribution for unobserved function values $z_i^* = g(\mathbf{x}_i^*)$ given $\mathbf{z}$ and $\mathbf{X}$, allowing us to make predictions for the underlying function and assess the uncertainty of these predictions.

Let $g(\mathbf{X})$ be the M-length vector of $g(\cdot)$ evaluated for each row of $\mathbf{X}$. A prior for $g(\mathbf{X})$ is

$$g(\mathbf{X}) \sim MVN(\beta_0 \mathbf{1}_M, \sigma^2 \mathbf{R}). \tag{2.14}$$

Here $\mathbf{1}_M$ is an M-length vector of ones, I is the $M \times M$ identity matrix, and $\mathbf{R}$ is an $M \times M$ matrix with $R_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, for kernel function $K(\cdot,\cdot)$.

In predicting M^* new function values $\mathbf{z}^*$ for a set of new inputs $\mathbf{X}^*$ we derive the conditional expected value, which we use as the predicted value for the underlying function, and the conditional variance as

$$E[\mathbf{z}^*|\mathbf{z},\mathbf{X}] = \beta_0 + \sigma^2 \mathbf{r}^{*T}(\sigma_e^2 I + \sigma^2 \mathbf{R})^{-1}(\mathbf{z} - \beta_0 \mathbf{1}_M) \tag{2.15}$$

$$Var[\mathbf{z}^*|\mathbf{z},\mathbf{X}] = \sigma^2 \mathbf{R}^* - \sigma^4 \mathbf{r}^{*T}(\sigma_e^2 I + \sigma^2 \mathbf{R})^{-1} \mathbf{r}^* \tag{2.16}$$

where $\mathbf{r}^*$ is an $M \times M^*$ matrix with $\mathbf{r}^*_{ij} = K(\mathbf{x}_i, \mathbf{x}_j^*)$, and $\mathbf{R}^*$ is an $M^* \times M^*$ matrix with $\mathbf{R}^*_{ij} = K(\mathbf{x}_i^*, \mathbf{x}_j^*)$.

The parameters β_0 and σ^2 are fit using maximum likelihood estimation. The noise variance, σ_e^2, is estimated by computing the sample variance of mutual information for several replications of the mutual information estimator at the same detector configuration.

Due to the presence of buildings and other obstacles in our input space, mutual information is expected to be a nonsmooth function of detector configuration. The kernel function $K(\cdot,\cdot)$ can be used to encode smoothness of the underlying function $g(\mathbf{x})$ being modeled (Santner et al. 2018). The Matern kernels are often used for modeling non-smooth functions. The Matern 3/2 kernel, which we found worked best for our problem, is

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \sqrt{3} h(\mathbf{x}_i, \mathbf{x}_j)\right) \exp\left\{-\sqrt{3} h(\mathbf{x}_i, \mathbf{x}_j)\right\}, \tag{2.17}$$

where $h(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{2n} \frac{(x_{ik} - x_{jk})^2}{\ell_k^2}}$, where each ℓ_k is a scaling factor that is fit using maximum likelihood. We use the DiceKriging package in R for fitting GP models (Roustant et al. 2012).

Bayesian Multivariate Adaptive Regression Splines

One common alternative to GPs is the multivariate adaptive regression spline (MARS) (Friedman 1991). MARS models are flexible, non-parametric models. They often scale better than GP models to training sets with high dimension or a large number of design points because they do not require the inversion and multiplication of large matrices (Banerjee et al. 2013a; Østergård et al. 2018). As with GP models, we assume the observed mutual information values $\mathbf{z}$ represent noisy observations of some function of the input configuration coordinates $\mathbf{x}_i$

$$z_i = g(\mathbf{x}_i) + e_i, e_i \overset{iid}{\sim} N(0, \sigma^2). \tag{2.18}$$

MARS approximates the underlying function $g(\cdot)$ with a basis function expansion,

$$\hat{g}(\mathbf{x}_i) = a_0 + \sum_{q=1}^{Q} a_q B_q(\mathbf{x}_i), \tag{2.19}$$

where the basis functions, $B_q(\mathbf{x}_i)$, are

$$B_q(\mathbf{x}_i) = \prod_{k=1}^{K_q} [s_{kq}(x_{v_{kq}} - t_{kq})]_+. \tag{2.20}$$

Here K_q is the maximum degree of interaction between terms, v_{kq} indexes a variable selected from the set of all input variables, t_{kq} is a knot location selected from the design points for $x_{v_{kq}}$, and s_{kq} is -1 or 1. The truncation function is $[x]_+ = \max\{0, x\}$. In Friedman (1991), MARS models are fit using a two-stage process. In the first stage, the forward pass, the algorithm generates a large pool of candidate basis functions by adding to the model the basis function which most reduces the model fit error given the other basis functions already in the model. In the second stage, the backward pass, candidate basis functions are sequentially eliminated using the generalized cross-validation criterion until some stopping criterion is met. The forward pass insures that all relevant basis functions are considered as candidates for the model, the backward pass insures the model is not overfit to the training data.

The Bayesian formulation of MARS, first proposed in Denison et al. (1998), samples the posterior distribution of the parameter vector $\{\sigma^2, Q, \mathbf{a}, \mathbf{K}, \mathbf{s}, \mathbf{v}, \mathbf{t}\}$ given the observed data $\mathbf{z}$. Here $\mathbf{a}$ is the vector of the basis function coefficients $(a_0, ..., a_Q)$, $\mathbf{K}$ is the vector of degrees of interaction for each basis function, $\mathbf{s}$,$\mathbf{v}$, and $\mathbf{t}$ are the vectors of signs, variables, and knots used for each basis function respectively. Similar to GP surrogates, BMARS constructs a posterior distribution over approximations of $g(\mathbf{x})$. Instead of a closed form posterior, however, we have samples from the posterior for $\hat{g}(\mathbf{x})$. The details for the parameter prior specification and the posterior sampling strategy used are found in Francom et al. (2018).

Each Monte Carlo sample generated by BMARS represents a MARS model for $g(\mathbf{x})$. To predict response values $\mathbf{z}^*$ for new design points $\mathbf{X}^*$ we use the MCMC samples to evaluate the posterior mean for $g(\mathbf{x})$. We use the BASS package in R for fitting the models (Francom and Sansó 2020).

Radial Basis Functions

The final surrogate model we employ is the radial basis function model. As with the previous surrogate models, the relationship between the mutual information estimates z_i and the detector configuration coordinate design points $\mathbf{x}_i$ is

$$z_i = g(\mathbf{x}_i) + e_i, e_i \overset{iid}{\sim} N(0, \sigma^2). \tag{2.21}$$

As with the MARS models, $g(\mathbf{x}_i)$ is modeled as a basis function expansion,

$$\hat{g}(\mathbf{x}_i) = a_0 + \sum_{q=1}^{Q} a_q \phi(||\mathbf{x}_i - \mathbf{x}^q||_2). \tag{2.22}$$

The M design points $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$ are formed into Q clusters using the K-means algorithm. The point $\mathbf{x}^q$ is the cluster center of the q-th cluster. There are several common choices for the basis function, $\phi(\cdot)$; for our analysis, we found the multi-quadratic basis function performed the best. The form of the multi-quadratic basis function is

$$\phi(||\mathbf{x}_i - \mathbf{x}^q||_2) = \sqrt{||\mathbf{x}_i - \mathbf{x}^q||_2^2 + s^2}. \tag{2.23}$$

Here s is a hyperparamter that must be selected by the user; we select this parameter using maximum likelihood. The model for $\hat{g}(\mathbf{x}_i)$ is fit by estimating the basis function coefficients $\mathbf{a} = \{a_0, ..., a_Q\}$ using ordinary least squares. We employ an implementation based on the radial basis function implementation by Neto (2013).

Surrogate Model Comparisons

To compare the accuracy of these three surrogate models, we computed a training set of 200 LHD design points, $\{\mathbf{x}_i\}_{i=1}^{M=200}$. Using the data generating process described in Section 2.2.1, we compute the detector response y_{ij}, for detector j in configuration i, $i = 1, ..., M = 200$, $j = 1, ..., n$, given $N = 1000$ source parameters drawn from $p(\boldsymbol{\theta})$. We do this for $n = 3$ and $n = 5$, corresponding to 3 and 5-detector configurations. The computed response vectors $\mathbf{y}_i$ as well as the 1000 source parameter vectors are used to compute KSG estimates z_i

for each configuration. The set $\{z_i, \mathbf{x}_i\}_{i=1}^{M=200}$ is a training set for fitting $\hat{g}(\mathbf{x})$ using each of the surrogate models discussed earlier in Section 2.3.3. We repeat this process to form a test set of 200 input-output points. We randomly divide the set of 200 test points into 20 subsets of 10 points each. For each test set, we compute predictions from the three fitted surrogate models compute the error between the surrogate model response and the test set response. For each test set, we fit the model and compute the error 15 times to account for the stochastic nature of each surrogate model-fitting process. We do this for all three surrogate models.

We measure surrogate model prediction error using root mean squared error (RMSE),

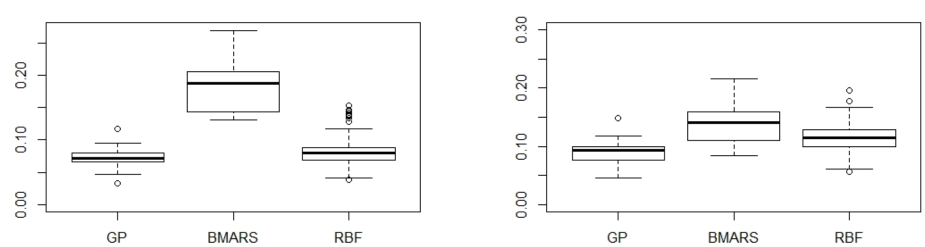
$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}(z_i - \hat{z}_i)^2}{M}} \tag{2.24}$$

where z_i is the test set mutual information output and $\hat{z}_i$ is the surrogate prediction for z_i.

Figure 2.3 presents a boxplot comparison of RMSE across the three methods for the 3 and 5-detector problem. For this application, the RBF and GP models appear to perform better than the BMARS model in both the 3-detector and 5-detector case. For the 20 test sets, we computed the median RMSE of the 15 repeated fits for each of the surrogate model types.

For the 3-detector case, we found that for all test sets, GP and RBF both had a lower median RMSE than BMARS. For 70% of the test sets, GP had a lower median RMSE than RBF. Thus for this problem in the 3-detector case, GP fits the best.

For the 5-detector case, we found that for 80% of the test sets, RBF had a lower median RMSE than BMARS and for 90% of the test sets, GP had a lower median RMSE than BMARS.



(a) RMSE comparison for 3 detectors (b) RMSE comparison for 5 detectors

Figure 2.3: RMSE comparison for surrogate methods

For 85% of the test sets GP had a lower RMSE than RBF. Thus, the GP model seems to also be the best choice for the 5-detector case in this application.

While the GP model is best for this particular example, this may not be the case when the method is applied in different urban settings. For different applications, we encourage the comparison of different surrogate models to compare accuracy, precision, and computational performance.

2.4 Movable Detector Placement Experiment

To compare the performance of the surrogate-based configuration optimization method described in Section 2.3 and the combinatorial optimization method described in Section 2.2.3, we compare the results of these two methods in localizing a source in a block of Washington, D.C. using movable detector placement. Movable detector networks are networks of radiation detectors where each detector can be moved so that the network is easily re-configurable. We describe how to construct and execute movable detector experiments and metrics that are useful for analyzing the results of these experiments. We will compare the source localization performance of the continuous and combinatorial configuration optimization methods in both the 3-detector and 5-detector cases.

2.4.1 Movable Detector Placement

Because of the difficulty in localizing a source due to varying background and obstructions, obtaining data from only a single detector configuration may be insufficient for precisely localizing the source. One strategy to improve source localization is movable detector placement. After we obtain data from an initial detector configuration and then employ the data along with the framework in Section 2.2.1 to compute $p(\boldsymbol{\theta}|\mathbf{Y})$, we can employ sequential optimization to find a subsequent detector configuration based on $p(\boldsymbol{\theta}|\mathbf{Y})$ to collect more data. This iterative process of constructing detector configurations, collecting data, and updating the source parameter distribution, can be repeated until we obtain an estimate of the source location with sufficient precision.

In this section, we compare the performance of the combinatorial optimization method and the continuous surrogate-based optimization method in the movable detector placement problem. For both methods, we start by drawing $N = 1000$ source parameter vectors from the source prior $p(\boldsymbol{\theta})$.

The combinatorial method generates detector response data $\mathbf{y}$ at each of 29 prespecified detector locations for each of the N source parameter vectors. These prespecified locations are depicted in Figure 2.1(a) and are the same detector locations used in Michaud (2019) for combinatorial optimization. We use a genetic algorithm (Wolters 2015) with the KSG mutual information estimator as the fitness function to perform global sequential optimization over the set of all possible 3- and 5-detector configurations composed of the 29 detector locations. We compute the detector response for each of the detectors in the selected optimal configuration using the data generating process in Section 2.2.1 for a true radiation source with location coordinates $\theta_1 = 118.073$ m and $\theta_2 = 134.388$ m and intensity parameter $\theta_3 = 3.472 \times 10^9$. We use this generated data as well as the Bayesian framework in Section 2.2.1 to draw MCMC samples from $p(\boldsymbol{\theta}|\mathbf{Y})$ using DRAM MCMC as implemented in the Python library pymcmcstat (Miles 2019). We find a subsequent configuration by drawing $N = 1000$ MCMC samples from $p(\boldsymbol{\theta}|\mathbf{Y})$, and then repeating the process of computing detector responses and performing sequential optimziaiton to find a subsequent optimal configuration. We draw samples from the updated posterior distribution using the data generated from the subsequent detector deployment using the same true source. We obtain an initial and subsequent deployment for the 3 and 5-detector scenarios.

For surrogate-based continuous optimization, we generate a LHD $\mathbf{X}$ with $M = 200$ over the configuration space. This design is large enough to fill the configuration space, but is small enough to be approximated by $\tilde{\mathbf{X}}$. We approximate $\mathbf{X}$ with $\tilde{\mathbf{X}}$ where $p = 30$. To generate detector response data, we only generate detector responses at the 30 LHD detectors for each of the 1000 source parameter vectors. Thus, the time required to generate detector response data is comparable to the combinatorial case with 29 detector locations. We employ the 1000 source parameter vectors and the generated detector data to compute the set of KSG mutual information estimators $\mathbf{z}$ for each detector configuration in $\tilde{\mathbf{X}}$. We employ $\mathbf{z}$ and $\tilde{\mathbf{X}}$ to fit a GP surrogate model for both the 3-detector and 5-detector cases. We then employ a genetic algorithm (Scrucca 2013) using the surrogate model as the fitness function to find a detector configuration that approximately maximizes mutual information. We use data generated from this optimal configuration and the same fixed source used in the combinatorial optimization procedure to draw MCMC samples from $p(\boldsymbol{\theta}|\mathbf{Y})$.

In the subsequent deployment, we sample $N = 1000$ source parameter vectors from $p(\boldsymbol{\theta}|\mathbf{Y})$. For the initial deployment, we used the marginal prior distributions of θ_1 and θ_2 to generate $\mathbf{X}$. In order to generate a LHD that adapts to our updated beliefs about source location, we should generate a LHD using the marginal posterior distributions of θ_1 and θ_2. We do not have these in closed form, but we obtain approximate closed form expressions by fitting 2-component Gaussian mixture models to the MCMC samples from the posterior. Under the posterior, θ_1 and θ_2 may no longer be independent, an assumption for the procedure described in Section 3.2. We employ the method described in Stein (1987) for generating LHDs using dependent random variables. Since we are primarily interested in configurations containing detector locations that are close to the likely location of the source, we only need to accurately model the reduced portion of the configuration space with detector locations in this small, high density region. Accurately modeling this reduced space requires fewer generated design points. Thus, we generate an $\mathbf{X}$ with $M = 50$ and $\tilde{\mathbf{X}}$ with $p = 30$. The rest of the procedure is the same way as for the initial deployment except the search space for the genetic optimization algorithm is restricted to the box whose boundary is formed by the most extreme coordinates from the MCMC sample from the posterior distribution.

Both the combinatorial and continuous surrogate-based optimization strategies yield a posterior distribution for the source parameters. In comparing the two approaches for configuration optimization, we compare the quality of the produced posterior distributions. We employ two metrics that quantify accuracy and precision. Maximum a posteriori (MAP) error compares the posterior mode of source location to the true location. The 95% highest density region (HDR) is the smallest region containing 95% of the probability density of the posterior. For our application a higher quality posterior distribution has a lower MAP error and a smaller HDR area.

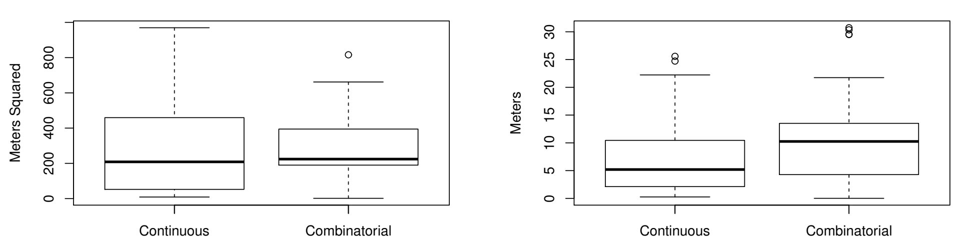
2.4.2 Simulation Results: Initial Deployment

We compute an initial detector configuration deployment for both 3- and 5-detector configurations using both the combinatorial and continuous optimization approaches and generate the resulting posterior distributions using data generated from the optimal configurations as described in Section 2.4.1. We repeat this 30 times for each combination of optimization method and number of detectors to account for variation in the optimal configuration found using genetic algorithm optimization. We compute the MAP error and HDR area metrics for all 30 generated posterior samples.

We compare the distributions of the MAP error and HDR area metrics for the 30 runs by ranking the 30 metrics for the continuous and combinatorial approaches and then counting the number of times the metric for the combinatorial approach exceeds the metric of the same rank for the continuous approach.

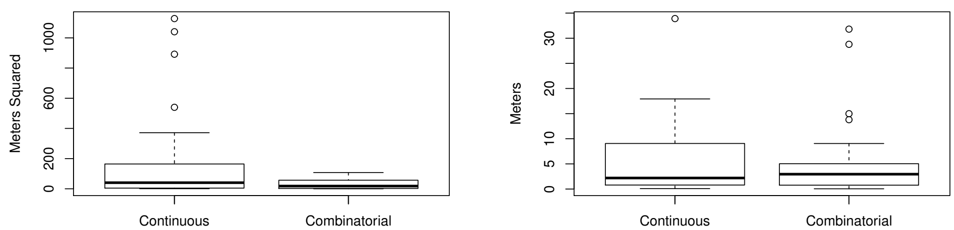
Figure 2.4 shows the boxplots of the HDR area and MAP error for the combinatorial and continuous optimization methods in the 3-detector case. For 29 of the 30 ranked MAP errors, the combinatorial method produced a higher error. In 16 of the 30 ranked HDR area metrics, the combinatorial method produced a posterior with higher HDR area. Thus, the continuous method is more accurate than the combinatorial approach and they have comparable precision. The superior accuracy of the continuous surrogate-based optimization method is likely due to its greater flexibility in selecting optimal detector configurations.

The same analysis is carried out for the 5-detector case as depicted in Figure 2.5. We found that for 17 of the 30 ranked MAP errors, the combinatorial method produced a higher error. Only 3 of the 30 ranked HDR area metrics, however, were higher for the combinatorial method than for the continuous method. Thus, while the continuous method still tends to be more accurate, it is less precise than the combinatorial metric in the 5-detector case. This is may be due to the greater flexibility gained by both methods in selecting 5 detectors instead of 3. This would indicate that the continuous method has a greater advantage in situations where we have a limited number of detectors to deploy relative to our search space. Because we are dealing with a higher dimensional input in the 5-detector case but using the same training set size as in the 3-detector case, it could be that the surrogate models are less accurate relative to those used in the 3-detector case. We did find that the surrogates for the 5-detector case had higher MSE for all surrogate modeling methods.



(a) HDR Area Comparison (b) MAP Error Comparison

Figure 2.4: Comparison of Initial Posterior Distributions in 3-Detector Case



(a) HDR Area Comparison (b) MAP Error Comparison

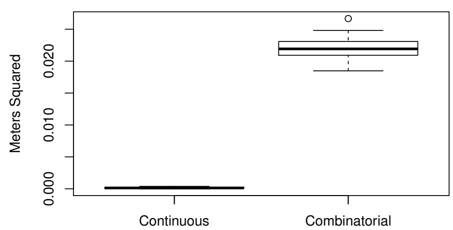
Figure 2.5: Comparison of Initial Posterior Distributions in 5-Detector Case

2.4.3 Simulation Results: Subsequent Deployment

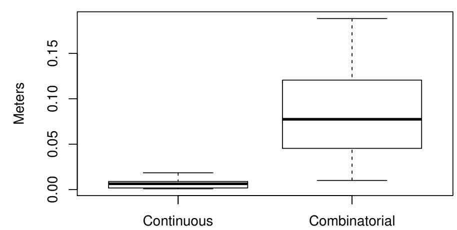
After comparing the results of the two optimization methods for an initial deployment of the detectors, we find subsequent deployments of 3 and 5-detector configurations using the strategy described in Section 2.4.1. To find a subsequent deployment, we require a posterior distribution for source location from which to draw N source parameter vectors. From the initial deployment results, we have 30 posterior distributions for each combination of optimization method and configuration size. For each optimization combination, we choose a posterior distribution whose MAP error and HDR area is close to the median MAP error and median HDR area for the 30 posterior distributions so that the posteriors chosen are comparable relative to the optimization method that produced them. Once subsequent optimal configurations have been found, we generate the detector response data for each configuration using the same fixed source as in the initial deployment. We use this generated data to draw MCMC samples from the updated posterior distribution. We repeat this process 30 times for each combination of configuration size and optimization method and compute the associated MAP error and HDR area metrics.

Figure 2.6 shows the boxplots for the MAP error and HDR area for the two methods in the 3-detector case. For all 30 runs, the ranked MAP errors and ranked HDR area metrics were larger for the combinatorial method than for the continuous method. This in conjunction with Figure 2.6 indicates the superiority across all metrics of the continuous optimization method for subsequent deployments of 3 detectors.

The strong performance of the continuous method in the subsequent deployment scenario is illustrated in Figure 2.7. The combinatorial method (black dots) can only place



(a) HDR Area Comparison



(b) MAP Error Comparison

Figure 2.6: Comparison of Subsequent Posterior Distributions in 3-Detector Case

detectors at prespecified candidate locations, and thus can only place detectors as close as the closest candidate locations. As illustrated in Figure 2.7, the continuous method (blue dots) can select configurations with detectors that are arbitrarily close to the true source location (red dot).

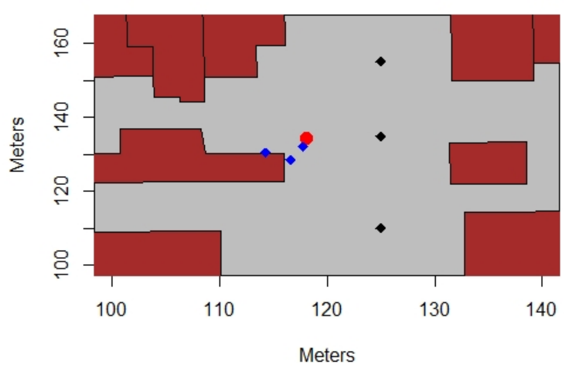
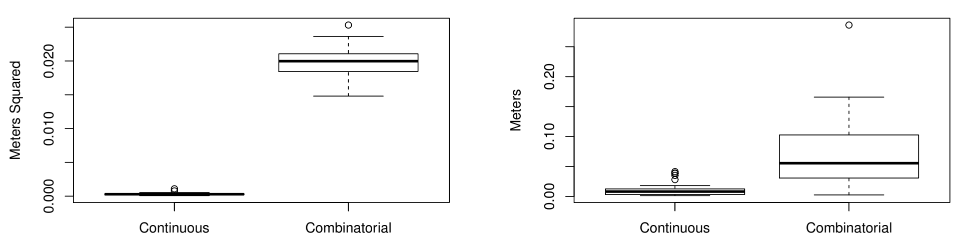


Figure 2.7: Subsequent Detector Deployments for Using Continuous Optimization and Combinatorial Optimization

In the initial deployment, the search space is the entire city block because the source location prior is uniform across this whole region. In the subsequent deployment, the search space is reduced by the posterior distribution constructed in the initial deployment. Since the combinatorial method employs a set of candidate detector locations fixed for all deployments, the density of candidate detector locations available in the subsequent search space is substantially reduced relative to the initial search space. Thus, the flexibility of the combinatorial optimization approach in constructing configurations is even more limited than it was in the initial deployment. The continuous optimization approach, because it searches the continuous configuration space, automatically adjusts itself to the smaller search space. Resampling the design points using the posterior distribution and rebuilding the surrogate models insures the surrogates are still accurate in this smaller search space. Thus, the continuous optimization approach automatically adapts itself to search spaces of different resolutions.

Figure 2.8 displays analogous results for the 5-detector case. For all 30 runs, the ranked HDR area metrics and the ranked MAP errors were larger for the combinatorial method than for the continuous method. Again, the continuous optimization method clearly outperforms the combinatorial method. Unlike in the initial deployment, in the subsequent deployment, the continuous optimization method for the 5-detector case performs significantly better. Even with 5 detectors, the candidate location grid used by the combinatorial optimization approach is too coarse to produce the same quality results as the continuous method.



(a) HDR Area Comparison (b) MAP Error Comparison

Figure 2.8: Comparison of Subsequent Posterior Distributions in 5-Detector Case

2.5 Conclusion and Future Work

Localizing nuclear radiation sources in an urban environment is important for securing urban populations against nuclear hazards resulting from negligence or hostility. This task is particularly challenging in urban environments where obstructions and background radiation make localization difficult. In this chapter, we have presented a framework for constructing configurations of radiation detectors that are optimal for performing radiation source localization. This framework optimizes mutual information to select the detector configuration that yields the greatest reduction in uncertainty about source location.

This framework constructs surrogate models of mutual information using Latin hypercube designs. We found that these surrogates are sufficiently accurate for identifying detector configurations that produce data and posterior distributions that are comparable or superior to the combinatorial optimization method. The advantage of this method over a combinatorial optimization method is that it optimizes over the continuous space of detector configurations while remaining computationally tractable because of the efficiency of surrogate models. The greater generality of this surrogate-based optimization allows for greater flexibility when selecting optimal detector configurations. This greater flexibility is critically important in large search spaces or domains represented in 3 dimensions, where sets of candidate detector locations may not be representative of the space of possible detector configurations.

In future work, we may apply different techniques for determining optimal detector placement such as Bayesian optimization. This technique simultaneously estimates and optimizes an objective function for which we only have input-output pairs. In our case, mutual information between source parameters and detector configuration data is an expensive unknown objective function we wish to estimate and optimize. The Bayesian optimization framework is thus a natural choice for solving this optimization problem.

Future work will also focus on extending this framework to 3-dimensional urban environments. In this case, the efficiency and flexibility of the continuous optimization method should produce significant advantages over the combinatorial optimization approach without incurring greater computational costs. To this end, we have created a software package that uses the powerful NVIDIA OptiX ray tracing engine to efficiently compute the expected detector response using (2.3). We describe the implementation and performance of this software package in Appendix B.

In future studies, it would also be worthwhile to consider larger urban environments for our search space. In our experiments with the block of downtown Washington, D.C., we were able to get a fairly tight posterior distribution for source location after the initial deployment, and the subsequent deployment for both methods produced a posterior with almost all its mass right on top of the source. While we did show that the surrogate-based approach did produce smaller errors in the subsequent deployment, both the surrogate approach and the combinatorial approach produced source location estimates that were less than a meter away from the true source location. Practically speaking, then, the two approaches essentially produced the same results. The value of the proposed surrogate-based approach in the context of sequential optimization and movable detectors would be clearer in a larger urban environment where the source is more challenging to localize.

CHAPTER

3

BAYESIAN HIDDEN MARKOV MODELS FOR PROCESS MONITORING WITH ARMS CONTROL APPLICATIONS

3.1 Introduction

As weapons systems have become more sophisticated and destructive, many nations have entered into arms control agreements to prevent the proliferation and development of especially destructive weapons such as nuclear, chemical, or biological weapons.

Effective arms control depends on a robust capability to verify that potential proliferators are not engaged in illicit arms development. This often involves monitoring the activities of state and non-state entities who are suspected of illicit arms development. In many cases, however, we do not have direct knowledge of what activities the potential proliferators are engaged in because the monitored activity is hidden from us intentionally or unintentionally. In these cases, the multi-modal data collected during the monitoring process, including satellite imagery, remote and on-site sensor readings, and in-person inspector observations, is provided to analysts with expertise in arms development to infer what the current activity is and describe the uncertainty for this inference.

It is challenging, however, for analysts to correctly and consistently interpret large and noisy sets of collected sensor data to make inferences about the monitored process. It is even more challenging to provide well-calibrated uncertainty estimates for these inferences. As a result, some authors suggest using statistical modeling and formal uncertainty quantification methods in support of national security decision-making (Stracuzzi et al. 2018).

In this chapter, we develop a Bayesian model based on hidden Markov models (HMM) for monitoring arms development processes. We consider processes that can be described as a sequence of M activities, which do not repeat, as depicted in Figure 3.1. We assume that we observe at discrete time points the use of certain pieces of equipment. This data may come from satellite imagery or an on-site observer. An example of this data is depicted in Figure 3.2. At any given time we may see a single piece of equipment in use (a crane) or multiple pieces of equipment in use (a crane and a truck). Observing certain patterns of equipment usage enables inferences about the corresponding hidden process because certain activities require particular pieces of equipment.

Given an observed data stream, like the one in Figure 3.2, we consider the following process monitoring questions:

1. What hidden process activities were underway at what times during the monitoring period?
2. When did this process start?
3. When will the process be finished?

Answers to these questions provide decision-makers with the information they need to make an informed evaluation of the threat posed by the process, but answering these questions poses several challenges. As already noted, what activity is underway at any

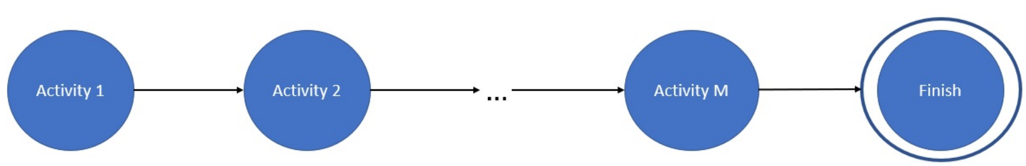


Figure 3.1: Process Model with M States

given time is hidden from us, and the data we observe does not indicate the corresponding hidden activity with perfect certainty. This introduces uncertainty into our answers for all the above questions.

The hidden Markov model is a natural choice for modeling the probabilistic relationship between process activities and observations, but standard HMMs do not incorporate activity duration into the model. Activity duration is crucial for answering the three process monitoring questions of interest. While there exist extensions of HMMs that do incorporate duration (Johnson 2005; Liu et al. 2015), inference with these approaches requires substantial algorithimic modifications to the standard HMM, significant increases to the size of the state space, and/or computationally expensive approximations of expressions that cannot be written in closed form. We introduce a reparameterization of the standard HMM that incorporates activity duration, reduces the size of the parameter set, requires no change to the state space, and involves only closed-form expressions. We also demonstrate how this reparameterization allows us to answer the questions of interest.

In most arms control applications, we do not have access to large amounts of process data with which to fit a model, but we do have access to analysts and experts whose knowledge about the process is useful for inference. This is common in national security analysis problems (Beaumont et al. 2015; Stracuzzi et al. 2018; Wilson et al. 2007). To ensure we are making the most of available information about the process, it is crucial that we incorporate expert knowledge of the process activities and potential noise in the data. For example, a weather event might temporarily prevent us from observing the process, or equipment might be used during the process to repair a facility; such equipment usage would not be directly related to the process and would introduce noise into the data. We propose a

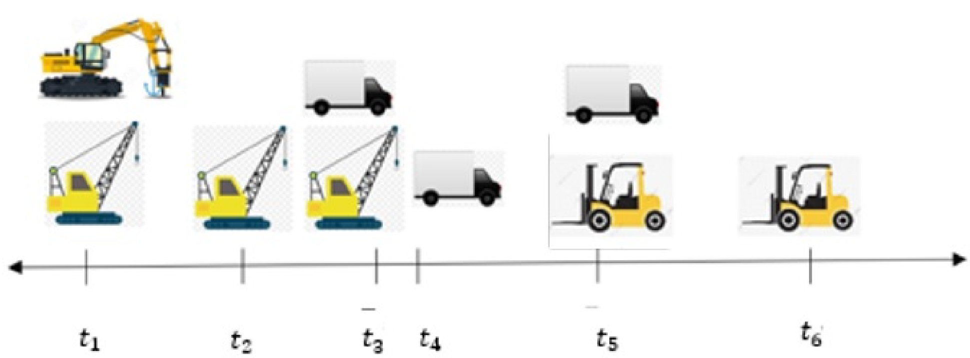


Figure 3.2: Sequence of Observed Data

novel framework for incorporating subject matter expertise, including information about noise, into our model framework. We also derive a Metropolis-within Gibbs algorithm for sampling from the posterior of the proposed model.

3.2 Related Work

3.2.1 Arms Control Process Monitoring

To foster more consistent, rigorous analysis, significant work has focused on constructing models of arms development processes and related data. Yue et al. (2008) develop a Markov model of the operations of peaceful nuclear energy systems, safeguards associated with the system, and potential diversion of special nuclear material by proliferators. The model simulates proliferation scenarios and calculates various metrics used to assess the proliferation resistance of a process. The authors develop the model from subject matter expertise and do not incorporate data observed from a real process. Rathore and Chirayath (2019) develop a stochastic Petri net model, discussed in the next section, to generate time-dependent simulations of proliferation pathways by Iran in order to determine Iran's ability to produce a weapon within a given time window. As in Yue et al. (2008), the Petri net model is developed from subject matter expertise and does not incorporate observed data from a process of interest.

Garcia et al. (2012) propose constructing a model of a nuclear facility and use data to determine if activities at the facility pose a proliferation risk. The authors discuss the challenge of incomplete or deceptive data as well as the difficulty of producing consistent, timely analytical conclusions when employing only human analysts. Their approach represents an attempt at statistical modeling of a proliferation process using observed data. Their approach, however, focuses on modeling a facility rather than a process. Garcia et al. (2017) develop a discrete event process model for a nuclear pyroprocessing system (a nuclear fuel processing procedure that is is vulnerable to exploitation by proliferators) that facilitates anomaly detection using data collected from multiple sensors. Although the approach does build a model describing the process being monitored and the relationship between observations and process states, the focus is on detecting anomalies in a process with observable states rather than determining the hidden activities of a process from observed data.

3.2.2 Process Models

As in the works discussed in Section 3.2.1, we construct a process model to represent the arms development process we are monitoring. The idea of using mathematical and probabilistic models to represent a real-life process is well-established. One of the oldest and most popular process modeling frameworks is the Petri net (Peterson 1977). A Petri net represents the process of interest as a directed graph where there are two types of nodes: places and transitions. A distribution of tokens to places denotes a state of the process. The transition nodes represent the activities that cause the process to move between states. Petri nets are used for a wide range of applications including business process management (Van Der Aalst 2015), system reliability (Volovoi 2004), and biological process modeling (Hardy and Robillard 2004), and, as mentioned, nuclear weapons proliferation modeling (Rathore and Chirayath 2019).

In traditional Petri nets, the activities of the process have a fixed duration. The stochastic Petri net models variability in the duration times with a probability distribution (Marsan 1988). Typically, the exponential distribution is used to model activity duration. In this case, Ciardo et al. (1994) show it is possible to derive a continuous time Markov chain (CTMC) model that is equivalent to the stochastic Petri net. This has many convenient mathematical properties especially when modeling processes with multiple concurrent activities.

In addition to Petri nets, researchers use stochastic activity networks (SANs) (Sanders and Meyer 2000) to model processes. SANs change the formalism of stochastic Petri nets to allow more readily for branching or parallel paths through a process, but still usually assume exponentially distributed durations since this allows for representing the SAN as a CTMC. SANs have been used to model workflows (Javadi Mottaghi and Abdollahi Azgomi 2009) and complex fault-tolerant industrial systems (Maza 2014).

3.2.3 Hidden Markov Models

While the Petri net and SAN models are useful for representing the dynamics and structure of a process, they do not provide a framework for associating an observed data stream with hidden process activities, which is a key challenge for monitoring arms control processes. A popular modeling framework that couples a simple process model with an observable data stream is the hidden Markov model (HMM) (Rabiner and Juang 1986). An example two-state HMM is illustrated in Figure 3.3.

An HMM is characterized by three sets of parameters which are highlighted by the

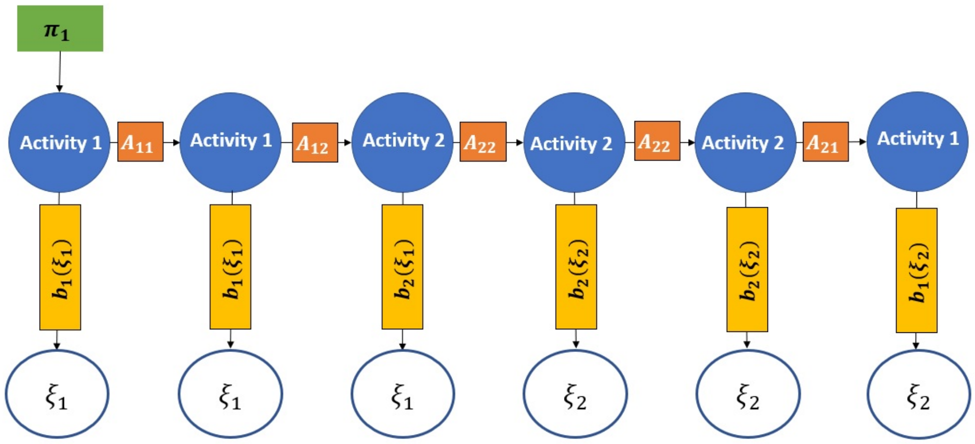


Figure 3.3: Example HMM

colored boxes in Figure 3.3. The first two sets of parameters are used to define a discrete-time Markov chain (DTMC) to model the underlying hidden process. The first set of parameters is an M-length vector $\boldsymbol{\pi} = (\pi_1, ..., \pi_M)$, called the *initial probabilities vector*. Here M is the number of activities in the process and where π_i is the probability that we begin observing data from the process when it is in state/activity i; i.e., $\pi_i = P(X_{t_1} = i)$, where X_{t_s} is the hidden process activity at time t_s and $s = 1, ..., n$ is the observation index. The second set of parameters, $\mathbf{A}$ is an $M \times M$ transition probability matrix, where the ij-the element, A_{ij} is the probability that the process is in activity i at the current observation given that it was in state j at the previous observation; i.e., $A_{ij} = P(X_{t_s} = i | X_{t_{s-1}} = j)$. The final set of parameters, $(\mathbf{b}_1, ..., \mathbf{b}_M)$, specifies the link between the hidden process and the observations. When the observable data is discrete or categorical, as in our application with different pieces of equipment, each $\mathbf{b}_i$ is a set of K categorical probabilities, denoted as $b_i(\xi_1), ..., b_i(\xi_K)$, that give the probability of seeing an observation of type or category ξ_k when the process is in state i. That is, $b_i(\xi_k) = P(O_{t_s} = \xi_k | X_{t_s} = i)$.

HMMs have been used extensively in many applications where the objective is to monitor a process with hidden states, such as the progression of a disease (Martino et al. 2020; Sukkar et al. 2012) or the degradation process of machines (Cholette and Djurdjanovic 2014). Because Bayesian analysis naturally facilitates uncertainty quantification, Zhang et al. (2016) use a Bayesian HMM to monitor a machine degradation process in order to detect system failures. Researchers also use HMMs to make process-based inferences. De Angelis and Paas (2013) use HMMs to detect crises in financial markets and Rahmawati et al. (2017) use HMMs to detect fraud in banking processes using banking log data. Pandey et al. (2011)

employ HMMs to predict the completion time of business processes from event logs, and Du et al. (2017) use an HMM to model the degradation of lubricating oil and to predict the remaining useful life of the oil.

There has been work on the use of HMMs for inference in noisy environments (Jagadeesh and Srikanthan 2015; Kelly et al. 2012; Punzo and Maruotti 2016) especially for speech recognition applications where there is noise in the speech signal (Cui et al. 2013; Arrowood and Clements 2002). For most of these applications, the observed data is a continuous signal, like a speech signal, so the observations can be modeled with a Gaussian that can straightforwardly incorporate noise. In many cases, these models are also fit on noisy data before attempting to perform inference. There has not been work that focuses on modeling observation noise in categorical data when our model is informed mostly by subject matter expert opinion.

Unlike in Petri net models, activity duration does not typically play a role in conventional HMMs. In our application, however, as in Yue et al. (2008) and Rathore and Chirayath (2019), time is a key feature of the process we are modeling. Hidden semi-Markov models (Yu 2010) permit the explicit modeling of state duration, but the algorithms used for fitting and performing inference for these models can be very computationally expensive compared to standard HMMs (Johnson 2005). Other approaches model time as being discrete (Langrock and Zucchini 2011). This does not allow for observations that come at arbitrary, irregular times. For this, continuous-time Hidden Markov models are typically employed (Liu et al. 2015). These approaches usually require the computationally demanding evaluation of matrix exponentials of the form $\exp(A) = \sum_{i=0}^{\infty} \frac{A^i}{i!}$, where A is a square matrix. We show that, assuming exponentially distributed duration times, certain properties of the processes we are monitoring allow us to use a reparameterization of the transition probabilities as analytical functions of the mean durations.

3.3 Methodology

The Bayesian HMM provides a basic framework for analyzing data emitted by a hidden process and provides rigorous uncertainty quantification. We introduce the standard Bayesian HMM as discussed in Cappé et al. (2005). We then discuss how to modify the standard HMM to incorporate activity duration and discuss how to incorporate subject matter expertise and account for noise when specifying the prior distributions. We describe a Gibbs algorithm that can be used for obtaining posterior draws from the model we propose and show how the model can be used to answer the process monitoring questions introduced in Section 3.1.

3.3.1 Conventional Bayesian HMM for Process Monitoring

Consider the conventional Bayesian HMM (Cappé et al. 2005), which serves as the basis for our proposed model framework. In our application, we focus on observations that represent a piece of equipment or a collection of equipment in use during the process. We can enumerate the set of all possible observation types, which we will call *observation symbols*, and use a categorical distribution for the state-specific observation model. Suppose there are K possible symbols. The observation model for state i is categorical: $O_{t_s}|X_{t_s} = i \sim Cat(b_i(\xi_1), ..., b_i(\xi_K))$ where ξ_j is the j-th observation symbol. We assume the observations are independent conditional on state. A requirement of the categorical distribution is that $\sum_{j=1}^{K} b_i(\xi_j) = 1$. Thus, a natural choice for the prior distribution for $\mathbf{b}_i$ is the Dirichlet distribution: $\mathbf{b}_i \sim Dir(\phi_1, ..., \phi_K)$.

The rows of $\mathbf{A}$ must also represent a valid discrete probability distribution. As a result, the standard Bayesian HMM also uses a Dirichlet prior distribution for each row of $\mathbf{A}$: $\mathbf{A}_i \sim Dir(\psi_1, ..., \psi_M)$.The last set of model parameters, π, are usually treated as nuisance variables that are fixed and known (Cappé et al. 2005). A common choice for π_i and the choice we use in this work is $\pi_i = 1/M$.

Given the above prior distribution specifications for the HMM model parameters, Scott (2002) proposes a Gibbs algorithm for simulating draws from the posterior distribution of $\mathbf{A}$ and each of $\mathbf{b}_1, ..., \mathbf{b}_M$ given the observed data sequence, $\mathbf{O}$. The algorithm treats the hidden state sequence $\mathbf{X}$ as a set of auxiliary parameters and then iterates between drawing $\mathbf{X}$ given $\mathbf{O}$ and $\mathbf{\Omega} = \{\mathbf{A}, \mathbf{b}_1, ..., \mathbf{b}_M, \pi\}$ and drawing from the posterior of $\mathbf{\Omega}$ given $\mathbf{X}$ and $\mathbf{O}$. Thus, the sampler also produces a posterior distribution over $\mathbf{X}$. In Section 3.3.5, we discuss the details of a modified version of this Gibbs algorithm which we propose for drawing from the posterior distribution of the model parameters of our model.

While the standard Bayesian HMM is a useful model framework, it is not suited to some of the specific characteristics of our problem. In the next sections we will propose extensions to the standard HMM.

3.3.2 Modeling Activity Duration with Exponential Random Variables

A critical part of most process models is a statistical model for the duration of each process activity. Discrete-time Markov chains do not explicitly model state/activity duration, require us to assume that observations are only observed at evenly spaced time-intervals, and assume the state transition probabilities in **A** are independent of the time between observations. These assumptions are too restrictive in the process monitoring context, where the underlying process is at least assumed to be a continuous-time Markov chain where activity duration impacts activity transitions. Intuitively, we want the probability of the process transitioning to a new state or staying in the same state between two observations to depend on the time between observations. That is, given that we are in state i at time t_s, the probability of transitioning to a new state or staying in state i by time t_{s+1} should depend on $t_{s+1} - t_s$ and the assumed distribution over state durations. More formally, we want to consider $P(X_{t_s} = j | X_{t_{s+1}} = i, t_{s+1} - t_s)$, the time-dependent transition probabilities, in our HMM instead of the time-independent transition probabilities.

As in SANs and stochastic Petri nets, we assume that the duration for each activity is exponentially distributed: $T_i \sim Exp(\gamma_i)$, where T_i is the duration of activity i and $E(T_i) = \gamma_i$. This assumption allows us to derive analytic expressions for the time-dependent transition probabilities, maintains the Markov property that is essential to an HMM, and allows us to obtain the distributions for quantities critical to answering the process monitoring questions of interest.

In general, it is not possible to derive an analytic expression for $P(X_{t_s} = j | X_{t_{s+1}} = i, t_{s+1} - t_s)$. However, because we assume exponentially distributed activity durations and because completed activities are not repeated, it is possible to derive analytic expressions for the time-dependent transition probabilities (Ross 2014). Let $t_{s+1} - t_s = \Delta_t$ and let $A_{ij}(\Delta_t) = P(X_{t_s} = j | X_{t_{s+1}} = i, \Delta_t)$. Then

$$A_{ij}(\Delta_t) = \begin{cases} 0 & i > j \\ e^{-\Delta_t/\gamma_i} & i = j \\ \sum_{s=i}^{j} e^{-\Delta_t/\gamma_s} \prod_{r \neq s, r=i}^{j} \frac{\gamma_s}{\gamma_s - \gamma_r} - \sum_{s=i}^{j-1} e^{-\Delta_t/\gamma_s} \prod_{r \neq s, r=i}^{j-1} \frac{\gamma_s}{\gamma_s - \gamma_r} & i < j. \end{cases} \tag{3.1}$$

The first expression of (3.1) is true because we are assuming we cannot return to previous activities in the process. The second expression follows from properties of exponential distributions:

$$A_{ii}(\Delta_t) = P(X_t = i | X_{t-\Delta_t} = i) = P(T_i > \Delta_t) = e^{-\Delta_t/\gamma_i}.$$

We obtain the third expression by noting that

$$A_{ij}(\Delta_t) = P(X_t = j | X_{t-\Delta_t} = i) = P\left(\sum_{m=i}^{j} T_m > \Delta_t\right) - P\left(\sum_{m=i}^{j-1} T_m > \Delta_t\right).$$

The sum of independent exponentials with different mean durations is hypoexponentially distributed (Kadri et al. 2015). Expressions for $P(\sum_{m=i}^{j} T_m > \Delta_t)$ can then be derived from the distribution function of the hypoexponential distribution.

Thus, we can formulate the time-dependent transition probabilities as functions of the duration means, γ_i and the time gap between observations. We have also reduced our set of parameters from an $M \times M$ matrix to a vector of M duration means.

We still wish to use a Bayesian framework for inference, so we must specify a prior distribution for the mean durations. Because the exponential distribution (as parameterized with the mean) and the inverse gamma distribution are a conjugate pair, it is common to use the inverse gamma as a prior distribution for γ_i. Thus, we have $\gamma_i \sim IG(\alpha_i, \beta_i)$.

3.3.3 Expert Elicitation of Prior Distributions

The key idea behind process monitoring is that we have some knowledge of the process before we begin monitoring. This knowledge may come in the form of previously acquired process data or subject matter expertise and usually includes knowledge of what activities make up the process, what order they follow, how long the activities of the process might last, what equipment is employed by the process, and the relative likelihood of observing different pieces of equipment being in use conditional on the process being in a particular state. This knowledge is used to determine what the observed data indicates about the process.

Unfortunately, in most arms control scenarios, the only process data we have access to is the single stream of data we are using for monitoring. For sequential processes, it is known that a single data stream, where each process activity is observed at most one time, is not enough to learn the HMM process model (Cappé et al. 2005; Rabiner and Juang 1986).

Fortunately, we can use existing expert elicitation techniques to leverage subject matter expertise to construct informative prior distributions (O'Hagan et al. 2006). We cannot usually elicit prior distribution for parameters directly. Experts do, however, have some knowledge of the shape and center of the distribution of each T_i. When eliciting information about a scalar quantity like T_i, O'Hagan et al. (2006) recommend eliciting the median and quartiles of the quantity of interest and describe procedures for doing this.

Suppose we learn from our experts that the p^{th} quantile of the duration for activity i has a value q_p. That is:

$$P(T_i \leq q_p) = p.$$

By the law of iterated expectations, we obtain

$$E(E(I(T_i \leq q_p)|\gamma_i)) = p \text{ implies } E(P(T_i \leq q_p)|\gamma_i) = p,$$

where $I(\cdot)$ is an indicator function. By properties of exponential distributions and since expectation is taken with respect to γ_i, which has an inverse-gamma distribution

$$E(P(T_i \leq\ q_p)|\gamma_i) = p \text{ implies } \int_0^\infty (1 - e^{q_p/\gamma_i}) \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} \gamma_i^{\alpha_i - 1} e^{-\beta_i/\gamma_i} d\gamma_i = p.$$

It is straightforward to show

$$\int_0^\infty (1 - e^{q_p/\gamma_i}) \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} \gamma_i^{\alpha_i - 1} e^{-\beta_i/\gamma_i} d\gamma_i = \left(1 - \frac{\beta_i^{\alpha_i}}{(\beta_i + q_p)^{\alpha_i}}\right)$$

Rearranging $p = \left(1 - \frac{\beta_i^{\alpha_i}}{(\beta_i + q_p)^{\alpha_i}}\right)$ we have

$$\beta_i = \frac{q_p (1-p)^{1/\alpha_i}}{(1 - (1-p)^{1/\alpha_i})}.$$

For a given α_i, we can set β_i to insure the p quantile of T_i is q_p. For instance, if we learn from our experts that the median duration for T_i is $q_{0.5}$ then for given α_i, we set

$$\beta_i = \frac{q_{0.5} 0.5^{1/\alpha_i}}{(1 - 0.5^{1/\alpha_i})}. \tag{3.2}$$

To set α_i, we can also elicit the upper quartile ($q_{0.75}$) and use an optimization routine to obtain an α_i so that $P(T_i \leq q_{0.75}) \approx 0.75$. The prior distribution specification can be further refined by iterating between presenting the expert with the prior predictive median and quartiles of T_i from the selected prior distribution and making adjustments to the prior specification based on expert feedback.

In addition to having knowledge about the distribution for T_i, experts typically know what pieces of equipment are likely to be employed during each of the process activities and the proportion of time certain pieces of equipment are used in a particular activity. We can interpret these proportions as an estimate of b_i in a noiseless environment. We use this knowledge to inform our prior distribution for each $\mathbf{b}_i$. Standard Bayesian HMMs employ a Dirichlet prior distribution for each $\mathbf{b}_i$. Dickey et al. (1983) propose an approach for using expert knowledge of discrete probabilities to inform a Dirichlet distribution. They reparameterize the Dirichlet as $\mathbf{b}_i \sim Dir(\kappa_i b_1^*(\xi_1), ..., \kappa_i b_i^*(\xi_K))$. Here $b_i^*(\xi_j)$ represents the expert's best estimate of $P(O_t = j|X_t = i)$. After obtaining $\mathbf{b}_i^*$, we present the expert with a hypothetical sample of observations for state i. Based on this sample, the expert is asked to provide a set of adjusted $\mathbf{b}_i^*$ based on the sample. The sample and the $\mathbf{b}_i^*$ can be used to obtain a reasonable setting for κ_i. See Dickey et al. (1983) for details.

3.3.4 Accounting for Noisy Data

As we noted in Section 3.1, the observed data will likely be noisy. Noise in arms control applications can be divided into two types: *concealment noise* and *incidental noise*. Concealment noise is when we do not see any equipment in use because the proliferator is hiding the equipment or there is something obstructing our observation device. Incidental noise is when we see equipment being used to accomplish some incidental operation (loading or unloading materials, repairing facilities, etc.) that is unrelated to a process activity. To reflect these two types of noise, we propose replacing the single Dirichlet with a hierarchical mixture of Dirichlets similar to the Latent Dirichlet Allocation (LDA) model used in topic modeling (Blei et al. 2003). Specifically, we use the following hierarchical prior distribution for $\mathbf{b}_i$:

$$\mathbf{b}_i|\mathbf{b}_i^{act}, \mathbf{b}_i^{noise}, \rho_i = (1-\rho_i)\mathbf{b}_i^{act} + \rho_i \mathbf{b}_i^{noise} \tag{3.3}$$

$$\mathbf{b}_i^{act} \sim Dir(\kappa_i b_i^*(\xi_1), ..., \kappa_i b_i^*(\xi_K)) \tag{3.4}$$

$$\mathbf{b}_i^{noise} \sim Dir(\phi_{i1}^{noise}, ..., \phi_{iK}^{noise}) \tag{3.5}$$

$$\rho_i \sim Unif(0,1) \tag{3.6}$$

where $b_i^{act}(\xi_k)$ is the probability of seeing ξ_k in activity i due to the execution of activity i, and $b_i^{act}(\xi_k)$ is the probability of seeing ξ_k in activity i due to concealment or incidental noise. The ρ_i is interpreted as the probability of an observation during activity i being due to noise. Thus, using this hierarchical model, the probability of seeing a particular observation symbol in activity i is a convex combination of the probability of seeing that symbol because of the activity currently underway or seeing that observation because of noise.

Experts usually can identify certain pieces of equipment as being unlikely to be observed as noise during a particular activity. If our experts indicate that it is unlikely for observations of ξ_k during activity i to be due to noise, we set $\phi_{ik}^{noise} = 0$. The remaining ϕ_{ik}^{noise} we set to 1.

3.3.5 Modified Metropolis within Gibbs Algorithm for Posterior Inference

In this section, we discuss the details of a Metropolis within Gibbs algorithm that is a modification of the Gibbs sampler used for HMM posterior inference proposed in Scott (2002) and Chib (1996). The sampler works by alternating between drawing posterior samples of $\mathbf{\Omega}$ given $\mathbf{X}$ and $\mathbf{O}$ and drawing samples of $\mathbf{X}$ given $\mathbf{\Omega}$ and $\mathbf{O}$.

First, we specify the full conditional distributions for the model parameters given $\mathbf{O}$ and $\mathbf{X}$. The conditional likelihood of the data given the hidden state sequence is

$$p(\mathbf{O}|\mathbf{X},\mathbf{\Omega}) = \pi_{X_{t_1}} \prod_{s=1}^{n-1} A_{X_{t_s},X_{t_{s+1}}}(t_{s+1} - t_s) b_{X_{t_{s+1}}}(O_{t_{s+1}}). \tag{3.7}$$

From the conditional likelihood, we obtain the full conditional density of γ_i:

$$p(\gamma_i|\mathbf{O},\mathbf{X}) \propto \prod_{s=1}^{n-1} A_{X_{t_s},X_{t_{s+1}}}(t_{s+1} - t_s) p(\gamma_i) \tag{3.8}$$

$$\propto e^{(t_s^{start} - t_s^{end})/\gamma_i} A_{i-1,i}(t_i^{start} - t_{i-1}^{end}) A_{i,i+1}(t_{i+1}^{start} - t_i^{end}) \gamma_i^{-\alpha_i - 1} e^{-\beta_i/\gamma_i} = g(\gamma_i), \tag{3.9}$$

where t_i^{start} is the time of the first observation in state i, and t_i^{end} is the time of the last observation in state i. This full conditional density distribution does not have a known distributional form. Thus, we perform a Metropolis step to draw from the posterior of the γ_i.

To derive the full conditionals of $\mathbf{b}_i^{act}$,$\mathbf{b}_i^{noise}$, and ρ_i, we recall that the model for $\mathbf{b}_i$ is the same as the LDA model. We have two "topics" (activity and noise) and K "words" represented by the K different observation symbols. A detailed treatment of Gibbs sampling for the LDA model is given in Darling (2011). Let $\mathbf{O}^{i(h)}$ be the set of n_i observations corresponding to activity i as determined from $X^{(h-1)}$, where h is index of the current draw from the sampler. To obtain posterior samples of $\mathbf{b}_i^{act}$, $\mathbf{b}_i^{noise}$, and ρ_i given $\mathbf{O}^{i(h)}$, we condition on n_i unobserved binary variables, $\mathbf{Z}^i$, that indicate whether each observation in $\mathbf{O}^{i(h)}$ was due to the process activity or due to noise. We draw $Z_j^{i(h)}$, the noise indicator for the j-th observation in $\mathbf{O}^{i(h)}$, as

$$Z_j^{i(h)}|\mathbf{b}_i^{act(h-1)},\mathbf{b}_i^{noise(h-1)},\rho_i^{(h-1)},\mathbf{O}^{i(h)} \sim Bern\left(\frac{\rho_i^{(h-1)} b_i^{noise(h-1)}(o_j^{i(h)})}{(1-\rho_i^{(h-1)})b_i^{act(h-1)}(o_j^{i(h)})+\rho_i^{(h-1)} b_i^{noise(h-1)}(o_j^{i(h)})}\right)$$

where $Z_j^{i(h)}=1$ if $O_j^{i(h)}$ is due to incidental or concealment noise and 0 otherwise. We draw $\mathbf{b}_i^{act(h)}$, $\mathbf{b}_i^{noise(h)}$, and $\rho_i^{(h)}$ conditional $\mathbf{Z}^{i(h)}$ and $\mathbf{O}^{i(h)}$ as

$$\mathbf{b}_i^{act(h)}|\mathbf{Z}^{i(h)},\mathbf{O}^{i(h)} \sim Dir(\kappa_i b_i^*(\xi_1)+N_{i1}^{act},...,\kappa_i b_i^*(\xi_K)+N_{iK}^{act}) \tag{3.10}$$

$$\mathbf{b}_i^{noise(h)}|\mathbf{Z}^{i(h)},\mathbf{O}^{i(h)} \sim Dir(\phi_{i1}^{noise}+N_{i1}^{noise},...,\phi_{iK}^{noise}+N_{iK}^{noise}) \tag{3.11}$$

$$\rho_i^{(h)}|\mathbf{Z}^{i(h)},\mathbf{O}^{i(h)} \sim Beta(1+N^{noise},1+n_i-N^{noise}), \tag{3.12}$$

where N_{ij}^{act} is the number of observations of ξ_j in activity i due to the process activity, N_{ij}^{noise} is the number of observations of ξ_j due to noise, and N^{noise} is the total number of observations due to noise. Note that these quantities are functions of $\mathbf{Z}^{i(h)}$. For more details on how these full conditional distributions are derived, see Darling (2011).

Now we discuss how we draw $\mathbf{X}$ given $\mathbf{O}$ and $\mathbf{\Omega}$. The joint full-conditional distribution of $\mathbf{X}$ given $\mathbf{O}$ and $\mathbf{\Omega}$ can be expressed as

$$p(\mathbf{X}|\mathbf{O},\mathbf{\Omega}) = p(X_{t_n}|\mathbf{O},\mathbf{\Omega})\prod_{s=1}^{n-1} p(X_{t_s}|X_{t_{s+1}},...,X_{t_n},\mathbf{O},\mathbf{\Omega}) \tag{3.13}$$

$$= p(X_{t_n}|\mathbf{O},\mathbf{\Omega})\prod_{s=1}^{n-1} p(X_{t_s}|X_{t_{s+1}},\mathbf{O},\mathbf{\Omega}). \tag{3.14}$$

Here the last equality follows because of the Markov assumption as pointed out in Chib (1996). When sampling from this joint distribution, we start by sampling X_{t_n} from $p(X_{t_n}|\mathbf{O},\mathbf{\Omega})$.

We then sample each of the other hidden state values, X_{t_s} from $p(X_{t_s}|X_{t_{s+1}}, \mathbf{O}, \mathbf{\Omega})$. We are able to sample X_{t_s} conditional on the next state in the sequence because we draw the sequence in reverse order. Notice that conditional on $X_{t_{s+1}}$, X_{t_s} does not depend on the other previously drawn state values, $X_{t_{s+2}},..., X_n$. In particular, this implies that the posterior probability that X_{t_s} is in state j conditional on $X_{t_{s+1}}$ does not depend on the total number of observations already assigned to state j.

Scott (2002) showed that the full conditional distribution for X_{t_s} is

$$X_{t_s}|\mathbf{O}, \mathbf{\Omega}, \mathbf{X}_{t_{s+1}} \sim Cat\left(\frac{w_1^{t_s}(x_{t_{s+1}})}{\sum_{i=1}^M w_i^{t_s}(x_{t_{s+1}})}, ..., \frac{w_M^{t_s}(x_{t_{s+1}})}{\sum_{i=1}^M w_i^{t_s}(x_{t_{s+1}})}\right), \tag{3.15}$$

where

$$w_i^{t_s}(x_{t_{s+1}}) = \begin{cases} q_{t_s}(i) & s = n \\ q_{t_s}(i) A_{i, x_{t_{s+1}}}(t_{s+1} - t_s) b_{x_{t_{s+1}}}(o_{t_{s+1}}) & 1 \leq s < n \end{cases},$$

and $q_{t_s}(i)$ is defined as

$$q_{t_s}(i) = \begin{cases} \sum_{j=1}^M \frac{w_j^{t_s - 1}(x_{t_s})}{C} & 1 < s \leq n \\ \pi_i b_i(o_{t_s}) & s = 1. \end{cases}$$

Here C is chosen so that $\sum_{i=1}^M q_{t_s}(i) = 1$. Thus, we recursively build up the $w_i^{t_s}(x_{t_{s+1}})$ and $q_{t_s}(i)$ starting with $q_{t_1}(i)$ to obtain the categorical distributions from which we draw each state. We then draw the state sequence in reverse order starting with the last state X_{t_n} using (3.15).

3.3.6 Answering Process Monitoring Questions

With the parameterization of the transition probabilities described in Section 3.3.2, we are able to answer the three process monitoring questions enumerated in Section 3.1. The first question asks what hidden process activities were underway at what times during the monitoring period. Because of the sequential nature of the process, we can equivalently find the times at which the process transitions out of a particular state. Let Tr_i be the time at which the process transitions from activity i to the next activity in $\mathbf{X}$. If we knew $\mathbf{X}$ we would have

$$Tr_i(\mathbf{X}) = \max\{t_s | X_{t_s} = i\}. \tag{3.16}$$

We can obtain a posterior predictive distribution for the transition time from state i by computing $Tr_i(\mathbf{X}^{(h)})$ for each posterior sample $\mathbf{X}^{(h)}$, of the hidden state sequence.

We also wish to perform inference on when the process likely started and ended. We will formulate the distribution function of T_{start}, the time difference between the first observation time (t_1) and the start time of the process, and T_{end}, the time difference between the last observation time (t_n) and the end time of the process. We will formulate these distribution functions conditional on the observation sequence $\mathbf{O}$ and the model parameters $\mathbf{\Omega}$.

By the law of total probability,

$$P(T_{start} < t|\mathbf{O},\mathbf{\Omega}) = \sum_{i=1}^{M} P(T_{start} < t, X_{t_1} = i|\mathbf{\Omega})P(X_{t_1} = i|\mathbf{O},\mathbf{\Omega})$$

and

$$P(T_{end} < t|\mathbf{O},\mathbf{\Omega}) = \sum_{i=1}^{M} P(T_{end} < t, X_{t_n} = i|\mathbf{\Omega})P(X_{t_n} = i|\mathbf{O},\mathbf{\Omega}).$$

$P(X_{t_1} = i|\mathbf{O},\mathbf{\Omega})$ and $P(X_{t_n} = i|\mathbf{O},\mathbf{\Omega})$ are often called the smoothing probabilities. The formulas for computing the smoothing probabilities can be found in most references on HMMs (Rabiner and Juang 1986).

The derivations for $P(T_{start} < t, X_{t_1} = i|\mathbf{\Omega})$ and $P(T_{end} < t, X_{t_n} = i|\mathbf{\Omega})$ are essentially the same, so we will only discuss the derivation for $P(T_{start} < t, X_{t_1} = i|\mathbf{\Omega})$. Note that because the process is sequential, it follows that

$$T_{start} = T_1 + T_2 + ... + T'_{X_{t_1}},$$

where $T'_{X_{t_1}}$ is the time spent in state X_{t_1} (the process state at the first observation) before we make our first observation. By the memoryless property of exponential random variables, $T'_{X_{t_1}} \sim Exp(\gamma_{X_{t_1}})$. Thus, given X_{t_1}, T_{start} has the same distribution as $T_1 + T_2 + ... + T_{X_{t_1}}$, a sum of independent exponential random variables. Thus,

$$P(T_{start} < t, X_{t_1} = i|\mathbf{\Omega}) = P\left(\sum_{j=1}^{i} T_j < t|\mathbf{\Omega}\right).$$

The sum of independent exponentials has a hypoexponential distribution as noted in Section 3.3.2. By properties of hypoexponential distributions it is known that:

$$P\left(\sum_{j=1}^{i} T_j < t|\mathbf{\Omega}\right) = 1 - \sum_{j=1}^{i} e^{-t/\gamma_j} \prod_{r \neq j, r=1}^{i} \frac{\gamma_j}{\gamma_j - \gamma_r}.$$

Thus,

$$P(T_{start} < t|\mathbf{O},\mathbf{\Omega}) = \sum_{i=1}^{M}\left(1-\sum_{j=1}^{i} e^{-t/\gamma_j} \prod_{r\neq j, r=1}^{i} \frac{\gamma_j}{\gamma_j-\gamma_r}\right) P(X_{t_1} = i|\mathbf{O},\mathbf{\Omega}). \tag{3.17}$$

By similar reasoning, we obtain

$$P(T_{end} < t|\mathbf{O},\mathbf{\Omega}) = \sum_{i=1}^{M}\left(1-\sum_{j=i}^{n} e^{-t/\gamma_j} \prod_{r\neq j, r=1}^{i} \frac{\gamma_j}{\gamma_j-\gamma_r}\right) P(X_{t_n} = i|\mathbf{O},\mathbf{\Omega}). \tag{3.18}$$

By computing $P(T_{start} < t|\mathbf{O},\mathbf{\Omega}^{(h)})$ and $P(T_{end} < t|\mathbf{O},\mathbf{\Omega}^{(h)})$ for each posterior sample $\mathbf{\Omega}^{(h)}$, we can obtain posterior predictive distributions for any value of the distribution function for T_{start} and T_{end}. We can also use the distribution function to find quantiles of the distributions for T_{start} and T_{end}, so we can also obtain posterior predictive distributions for any quantile of T_{start} and T_{end}.

An alternative approach for computing summaries of the distribution for T_{start} and T_{end} is to simulate from the conditional distributions $T_{start}|\mathbf{\Omega},\mathbf{O}$ and $T_{end}|\mathbf{\Omega},\mathbf{O}$. We will describe this simulation approach for $T_{start}|\mathbf{\Omega},\mathbf{O}$; the approach for $T_{end}|\mathbf{\Omega},\mathbf{O}$ is analogous.

To simulate from $T_{start}|\mathbf{\Omega},\mathbf{O}$, we use the following procedure:

1. Draw $X_{t_1} \sim Cat(P(X_{t_1} = 1|\mathbf{O},\mathbf{\Omega}),...,P(X_{t_1} = M|\mathbf{O},\mathbf{\Omega}))$
2. Draw $T_1 \sim Exp(\gamma_1),...,T_{X_{t_1}} \sim Exp(\gamma_{X_{t_1}})$
3. Obtain $T_{start} = T_1 + ... + T_{X_{t_1}}$
4. Repeat steps 1-3 for desired number of samples.

Using this simulation approach, we can obtain approximate moments, quantiles, and other functions of T_{start} and T_{end} given model parameters and observed data. As with the distribution functions, we can obtain posterior predictive draws of distributional summaries computed from simulated values by repeating the simulation and summary calculation for multiple posterior samples of the model parameters.

3.4 Simulation Study: Effect of Observation Order

For most Bayesian models, the inference on the model parameters given the data does not depend on the order of the observations. In the case of the HMMs we consider in this chapter, however, the parameter inference does depend on the observation order.

To demonstrate the dependence on observation order, we will analyze the unconditional likelihood function and note how it can change with different observation orders. For simplicity, we will assume the conventional parameterization and prior distribution for b_i introduced in Section 3.3.1 throughout this discussion. The likelihood for HMMs is typically computed using the forward-backward algorithm, see Rabiner and Juang (1986):

$$\mathscr{L}(\mathbf{O}|\mathbf{\Omega}) = \sum_{i=1}^{M} a_n(i)$$

$$a_s(i) = \begin{cases} \pi_i b_i(O_{t_1}) & s = 1 \\ \sum_{j=1}^{M} a_{s-1}(j) A_{ji}(t_s - t_{s-1}) b_i(O_{t_s}) & 1 < s \leq n. \end{cases}$$

In general, if $A_{ji}(t_s - t_{s-1}) \neq A_{ij}(t_s - t_{s-1})$ and for some ξ_j, $b_i(\xi_j)$ changes across $i = 1, ..., M$, reordering the observation sequence will yield a different likelihood value. Such lack of symmetry in transition probabilities and observation probabilities between states is what causes the ordering of the data points to impact inference.

To illustrate the effect of different observation order concretely, consider a model with $M = 2$ and $K = 2$ where the set of possible observation symbols is $\{A, B\}$. Consider two two-observation datasets: $\mathbf{O}_1 = [O_{t_1} = A, O_{t_2} = B]$ and $\mathbf{O}_2 = [O_{t_1} = B, O_{t_2} = A]$. Suppose also that $\Delta_t = t_2 - t_1$. The two likelihoods of these observation sequences are

$$\begin{aligned} \mathscr{L}(\mathbf{O}_1|\mathbf{\Omega}) = & \pi_1 b_1(A) A_{11}(\Delta_t) b_1(B) + \pi_1 b_1(A) A_{12}(\Delta_t) b_2(B) + \\ & \pi_2 b_2(A) A_{21}(\Delta_t) b_1(B) + \pi_2 b_2(A) A_{22}(\Delta_t) b_2(B) \end{aligned}$$

and

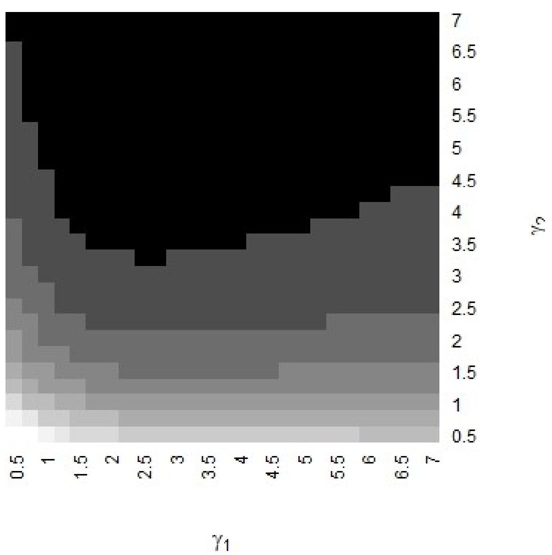
$$\begin{aligned} \mathscr{L}(\mathbf{O}_2|\mathbf{\Omega}) = & \pi_1 b_1(B) A_{11}(\Delta_t) b_1(A) + \pi_1 b_1(B) A_{12}(\Delta_t) b_2(A) + \\ & \pi_2 b_2(B) A_{21}(\Delta_t) b_1(A) + \pi_2 b_2(B) A_{22}(\Delta_t) b_2(A). \end{aligned}$$

Note $\mathscr{L}(\mathbf{O}_1|\mathbf{\Omega}) = \mathscr{L}(\mathbf{O}_2|\mathbf{\Omega})$ if $A_{12}(\Delta_t) = A_{21}(\Delta_t)$ or if $b_1(A) = b_2(A)$ and $b_1(B) = b_2(B)$. In our application, we are assuming that $A_{21}(\Delta_t) = 0$ and $0 < A_{12}(\Delta_t) < 1$, and we are assuming the observation probabilities for a particular symbol are different across states (this is what allows us to distinguish between states).

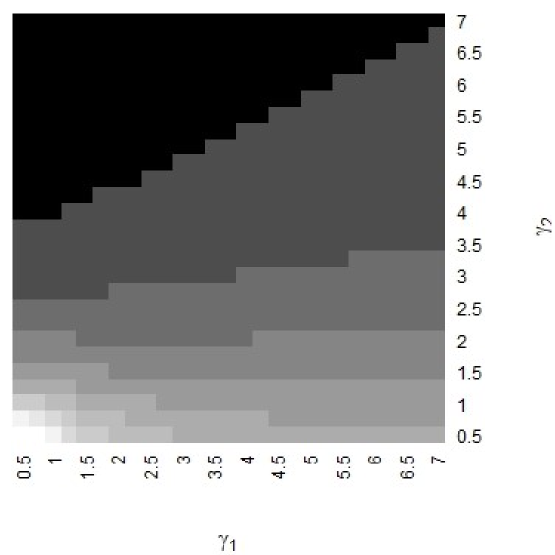
To further illustrate the impact of observation order on the likelihood, we will consider two four-observation datasets $\mathbf{O}_1 = [A, A, B, B]$ and $\mathbf{O}_2 = [A, B, A, B]$ and examine plots of the likelihood.

Figure 3.4 displays the joint profile log-likelihood of γ_1 and γ_2 while $b_1(A)$, $b_1(B)$, $b_2(A)$, and $b_2(B)$ are fixed at their posterior medians for $\mathbf{O}_1$ and $\mathbf{O}_2$. Darker colors correspond to higher values of the likelihood. When analyzing dataset 1, very small values of γ_1 yield lower likelihood values, but the influence of γ_1 is relatively weak. Higher values of γ_2 lead to higher values of the likelihood for both datasets 1 and 2. For dataset 2, the impact of γ_1 does change and tends to be stronger when $\gamma_2 > 4$, in which case, smaller values of γ_1 yield higher likelihood values.

Figure 3.5 displays the joint profile log-likelihood of $b_1(A)$ and $b_2(A)$ where γ_1 and γ_2 are fixed at their posterior medians for $\mathbf{O}_1$ and $\mathbf{O}_2$. We do not examine $b_1(B)$ and $b_2(B)$ since $b_1(B) = 1 - b_1(A)$ and $b_2(B) = 1 - b_2(A)$. When analyzing dataset 1, high likelihood values are seen for high values of $b_1(A)$ and low values of $b_2(A)$. This pattern changes when considering dataset 2, where higher likelihood values are achieved for medium to high values of $b_1(A)$ and low to medium values of $b_2(A)$.



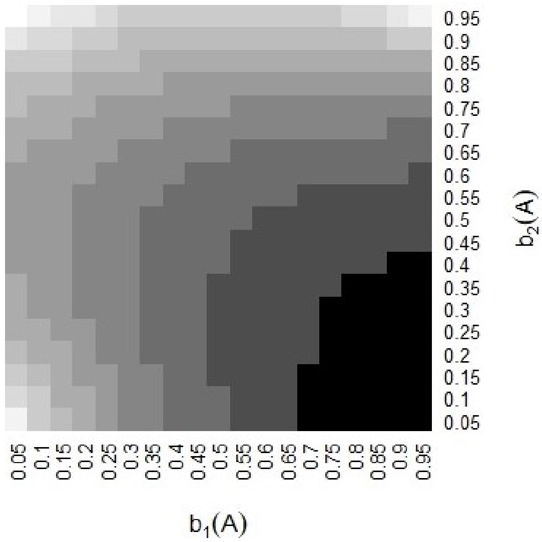
(a) Log likelihood for $\mathbf{O}_1$



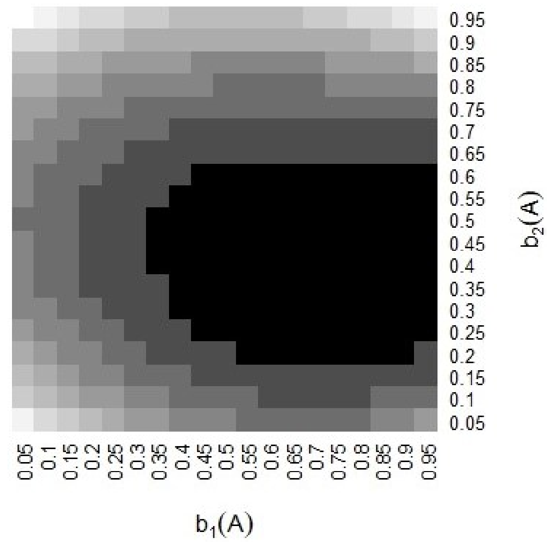
(b) Log likelihood for $\mathbf{O}_2$

Figure 3.4: Log likelihood of γ_1 and γ_2 for $\mathbf{O}_1$ and $\mathbf{O}_2$

We can also examine the effect of observation order on the posteriors of the model parameters. Figure 3.6 displays boxplots of samples from the posterior for γ_1 calculated



(a) Log likelihood of $b_1(A)$ and $b_2(A)$ for $\mathbf{O}_1$



(b) Log likelihood of $b_1(A)$ and $b_2(A)$ for $\mathbf{O}_2$

Figure 3.5: Log likelihoods of $b_1(A)$ and $b_2(A)$, γ_is fixed

using datasets 1 and 2. The two posteriors are similar with the median being slightly lower when computed using dataset 2. Figure 3.6 is an analogous plot for γ_2. The posterior median for γ_2 computed using dataset 2 is slightly higher and the tail is longer. This agrees with the results of analyzing the likelihood, which showed the likelihood was higher for higher values of γ_2 when analyzing dataset 2.

Figures 3.8 and 3.9 are analogous plots for $b_1(A)$ and $b_2(A)$. These results also agree with the likelihood analysis. Both parameters have more uncertain posteriors when analyzing dataset 2, and the posterior median for $b_1(A)$ decreases, while the posterior median for $b_2(A)$ increases when analyzing dataset 2.

We can also examine the effect of the observation order on the posterior of the transition time between activity 1 and 2, T_{start}, and T_{end}. Figure 3.10 displays the posterior predictive distributions for the transition time under datasets 1 and 2.

The prior distributions used for the activity-specific observation probabilities are $\mathbf{b}_1 \sim Dir(4.5, 0.5)$ and $\mathbf{b}_2 \sim Dir(1.5, 3.5)$. For these prior distributions, we are much more likely to see A under activity 1 and B under activity 2. For dataset 1, the posterior for the transition time has most of its mass concentrated between O_{t_2} and O_{t_3} because before this divide, we see only As and after this divide, we see only Bs. Thus, given the prior distribution, it seems clear the dividing line between states is between the 2nd and 3rd observation. For dataset 2, it is less clear where to draw the line between activities because the As and Bs

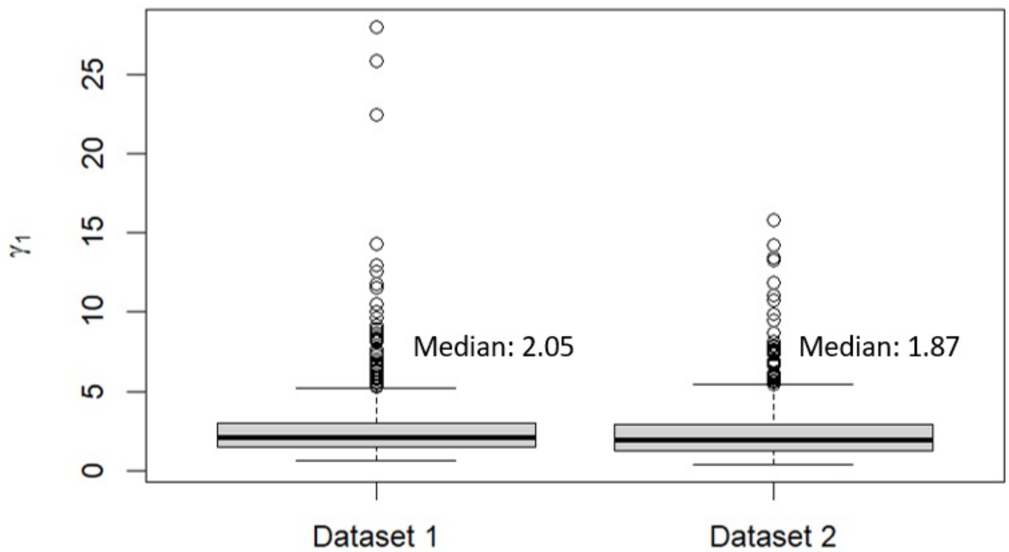


Figure 3.6: Posteriors for γ_1

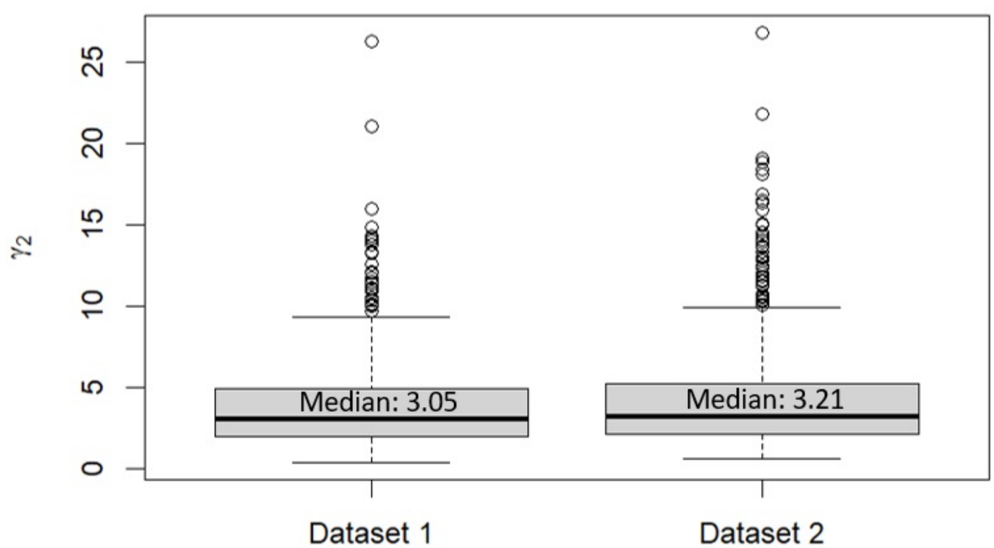


Figure 3.7: Posteriors for γ_2

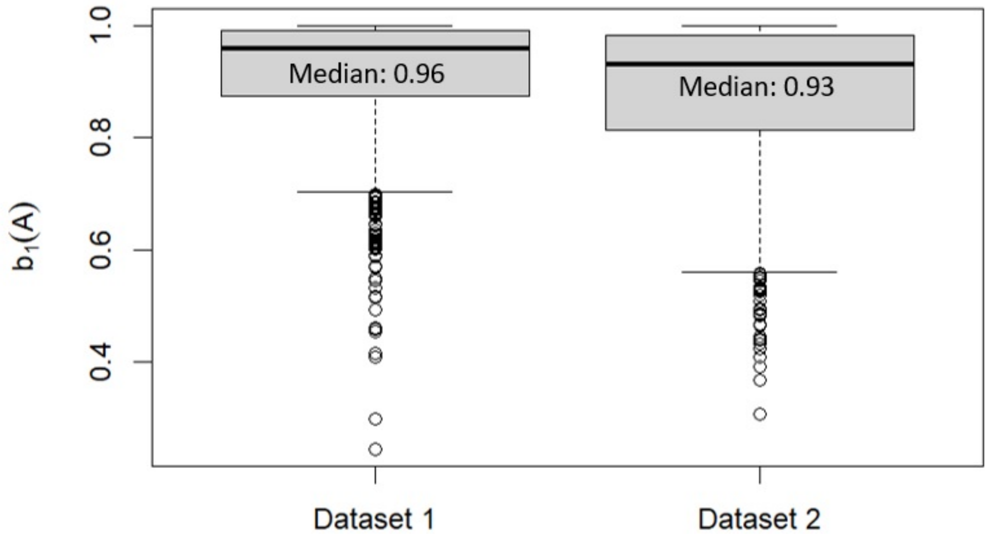


Figure 3.8: Posteriors for $b_1(A)$

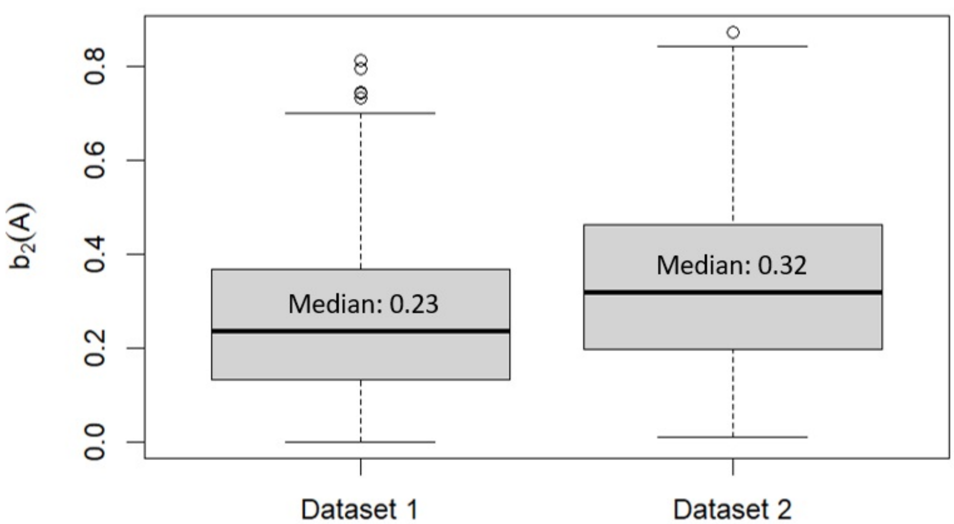
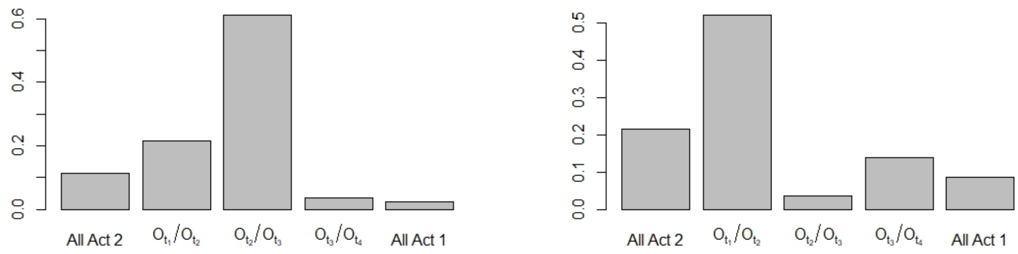


Figure 3.9: Posteriors for $b_2(A)$



(a) Posterior Predictive Under Dataset 1 (b) Posterior Predictive Under Dataset 2

Figure 3.10: Posterior Predictive Distribution for Transition Time between Activity 1 and 2

are intermixed. Because the prior distribution for $\mathbf{b}_2$ is more concentrated in regions of the simplex where $\mathbf{b}_2$ is more uniform, the posterior concentrates most of its mass between observation 1 and 2. Thus, the second A is labeled as being part of activity 2 because a priori it is more likely that the second A is seen under activity 2 than that the first B is seen under activity 1. This illustrates how the data order interacts with the prior distributions to effect inference on the process monitoring questions.

Figures 3.11 and 3.12 give the posterior predictive distributions for T_{start} and T_{end} computed using datasets 1 and 2. The posterior median for T_{start} is slightly higher for dataset 2 since it is more likely when considering dataset 2 that the first observation is in activity 2 (thus, some of activity 2 and all of activity 1 must have proceeded our observations).

3.5 Case Study: DAG Process

3.5.1 Process Description

From 2010 to 2019, the US Department of Energy (DOE) conducted a series of underground explosives tests, called the dry alluvium geology (DAG) tests, at the Nevada National Security Site (NNSS). The purpose of these tests was to improve detection of underground nuclear explosive tests in noisy geological environments. These tests, however, used only conventional high explosives in a canister which was placed in a borehole and detonated underground.

In this case study we focus specifically on the process of preparing an explosives canister

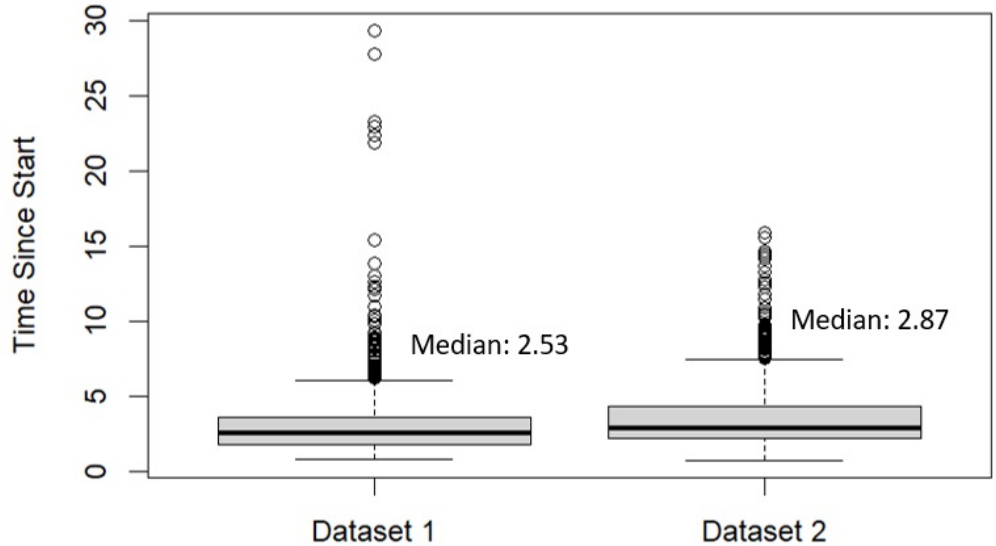


Figure 3.11: Posterior Predictive Distribution over Time Since Start

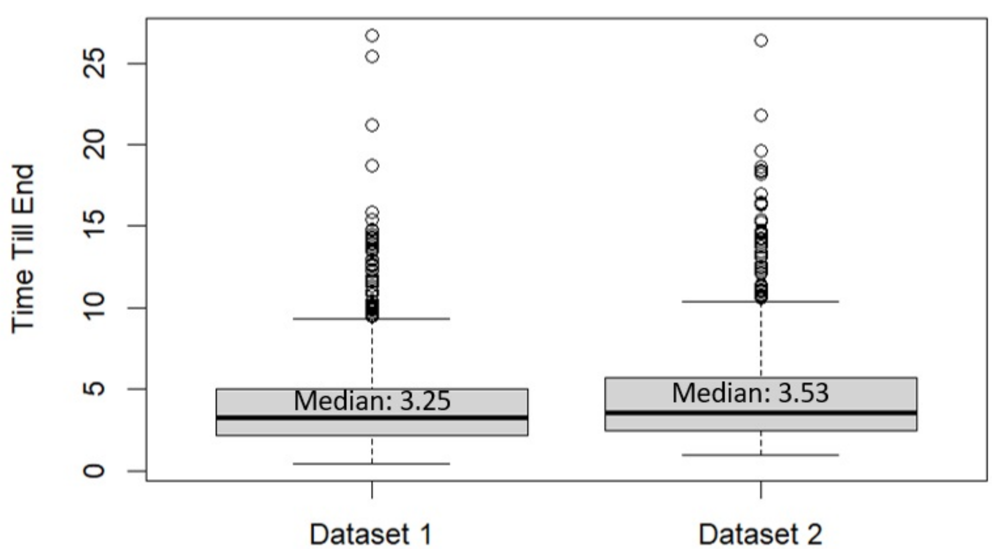


Figure 3.12: Posterior Predictive Distribution over Time Until Process End

for detonation. In discussions with scientists at Los Alamos National Laboratory (LANL), we learned that the preparation process can be broken down into a series of seven sequential activities:

1. Receive the explosives canister
2. Remove the cover from the canister
3. Position rings for controlling the canister during preparation
4. Position canister for filling
5. Conduct test operational readiness review
6. Install initiation charge
7. Fill canister with explosive

The full process is illustrated as a process diagram in Figure 3.13.

There are ten observation symbols we can see during the canister preparation procedure (see Table 3.1). These symbols represent certain combinations of equipment including an 82 ton crane, 14 ton crane, 7 ton forklift, and a high explosives (HE) forklift. Each of the process activities has an anchoring observation (AO), which is observed with high probability during the activity in noiseless environments. Because the 14 ton crane and 7 ton forklift are more general-purpose pieces of equipment, it is possible to see these pieces of equipment as incidental noise. We also expect to see no equipment in cases of concealment noise.

In this case study, we analyze a simulated set of process data from the canister preparation phase of the DAG test process. We demonstrate how to make process inferences

Table 3.1: Observation Symbols

ξ_1: 82 and 14 ton crane (AO for activities 1 and 4)	ξ_6: HE forklift (AO for activity 7)
ξ_2: 82 and 14 ton crane and 7 ton forklift	ξ_7: HE forklift and 7 ton forklift
ξ_3: 7 ton forklift (AO for activities 2 and 3)	ξ_8: HE forklift and 14 ton crane
ξ_4: 7 ton forklift and 14 ton crane in use	ξ_9: HE forklift and 7 ton forklift and 14 ton crane
ξ_5: 14 ton crane (AO for activity 6)	ξ_{10}: No equipment (AO for activity 5)

Figure 3.13: Diagram of Dry Alluvium Geology (DAG) Process

Table 3.2: State Duration Generation Parameters

	a_i	b_i	m_i	r_i
T_1	2.3	3.7	5	15
T_2	2.3	3.7	1	3
T_3	2.3	3.7	1	3
T_4	2.5	3.5	2	8
T_5	2.3	3.7	5	3
T_6	5	1	5	5
T_7	2.3	3.7	45	15

using the proposed model framework in high and low noise environments. In low noise environments, we expect to determine the transition times ($Tr_i(\mathbf{X})$) with high accuracy. In high noise environments, we expect this accuracy to degrade, and the uncertainty of our inferences should increase indicating our greater uncertainty when dealing with high noise.

3.5.2 Data Generation

In consultation with scientists from LANL, we created a simulator to generate simulated observation data for the canister preparation portion of the DAG test process.

Although we model an exponential distribution for the duration times in the HMM, collaborators at LANL suggest that a shifted beta distribution is more realistic for data generation: $T_i \sim r_i Beta(a_i, b_i) + m_i$, where m_i is the lowest possible duration for activity i, $m_i + r_i$ is the maximum possible duration for activity i and a_i and b_i are shape parameters controlling the shape of the distribution for T_i. We obtained values of r_i, m_i, a_i and b_i directly from our collaborators at LANL; these values are presented in Table 3.2.

We assume that data is collected at regular intervals, once every half hour. To generate observations at each observation time, we use a set of observation probabilities developed based on discussions with LANL collaborators. We use two sets of observation probabilities,

Table 3.3: State Observation Probabilities - Low Noise

	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6	ξ_7	ξ_8	ξ_9	ξ_{10}
Receive Canister	0.7	0.075	0.045	0.045	0.015	0	0	0	0	0.12
Remove Cover	0	0	0.7	0.075	0.075	0	0	0	0	0.15
Position Rings	0	0	0.7	0.12	0.03	0	0	0	0	0.15
Position Canister	0.7	0.03	0.06	0.06	0.015	0	0	0	0	0.135
Operational Readiness Review	0	0	0.18	0.03	0.09	0	0	0	0	0.7
Install Initiator	0	0	0.12	0.03	0.7	0	0	0	0	0.15
Fill Canister	0	0	0.018	0.018	0.018	0.7	0.06	0.06	0.006	0.12

one representing a low noise scenario (Table 3.3), the other representing a high noise scenario (Table 3.4). Notice that in Table 3.4, the observations usually associated with noise (ξ_3, ξ_4, ξ_5, and ξ_{10}) have higher associated probabilities relative to the same probabilities in Table 3.3, and the probabilities associated with the anchor observations for each state are lower in Table 3.4 relative to the corresponding probabilities in Table 3.3.

Table 3.4: State Observation Probabilities - High Noise

	ξ_1	ξ_2	ξ_3	ξ_4	ξ_5	ξ_6	ξ_7	ξ_8	ξ_9	ξ_{10}
Receive Canister	0.35	0.1625	0.0975	0.0975	0.0325	0	0	0	0	0.26
Remove Cover	0	0	0.35	0.1625	0.1625	0	0	0	0	0.325
Position Rings	0	0	0.35	0.26	0.065	0	0	0	0	0.325
Position Canister	0.35	0.065	0.13	0.13	0.0325	0	0	0	0	0.2925
Operational Readiness Review	0	0	0.39	0.065	0.195	0	0	0	0	0.35
Install Initiator	0	0	0.26	0.065	0.35	0	0	0	0	0.325
Fill Canister	0	0	0.039	0.039	0.039	0.35	0.13	0.13	0.013	0.26

For the final step of the data generation process, we make two draws from a uniform distribution over the whole duration of the process to give us an observation start and end time. For the dataset generated in this study, we drew an observation start time of 7.5, which is in the middle of activity 1, and we drew an observation end time of 24.5 which is in the middle of activity 5.

Table 3.5: Prior Parameters for γ_i

	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7
α_i	20.9	23.9	23.9	28.1	30.9	19.1	38.51
β_i	312.6	71.8	71.8	210.1	269.4	253	2783.01

3.5.3 Model Priors

We construct prior distributions for the state durations using the techniques described in Sections 3.3.3 and 3.3.4. To construct the prior distributions of the γ_is, we obtain a 50% and 75% quartile from the shifted Beta distributions described in the 3.3.4. The prior distribution parameters used in this study are listed in Table 3.5.

For the prior distribution for $\mathbf{b}_i$, we construct a prior using the LDA method described in Section 3.3.4. For comparison, we also consider a Dirichlet prior distribution (the *strong prior*) that puts almost all of the prior mass on the anchoring observations. This prior distribution represents an analysis without elicited information about potential concealment and incidental noise. For both priors, we use $\kappa_i = \kappa = 15$. The other parameters for the prior distributions are listed in Tables 3.6-3.8.

Table 3.6: State specific $\mathbf{b}_i^*$ for LDA Prior

	$b_i^*(\xi_1)$	$b_i^*(\xi_2)$	$b_i^*(\xi_3)$	$b_i^*(\xi_4)$	$b_i^*(\xi_5)$	$b_i^*(\xi_6)$	$b_i^*(\xi_7)$	$b_i^*(\xi_8)$	$b_i^*(\xi_9)$	$b_i^*(\xi_{10})$
Receive Canister	0.85	0.05	0.05	0	0.05	0	0	0	0	0
Remove Cover	0	0	0.8	0.05	0.05	0	0.1	0	0	0
Position Rings	0	0	0.8	0.05	0.05	0	0.1	0	0	0
Position Canister	0.85	0.05	0.05	0	0.05	0	0	0	0	0
Operational Readiness Review	0	0	0.05	0	0	0	0	0	0	0.95
Install Initiator	0.15	0	0	0	0.85	0	0	0	0	0
Fill Canister	0	0	0	0	0	0.95	0	0.05	0	0

To illustrate the differences between the prior distributions, Figures 3.14 and 3.15 display draws from the LDA and strong prior distributions for the elements of $\mathbf{b}_1$. Notice that for the strong prior distribution, the marginal distributions for each element of $\mathbf{b}_1$ have very small variance and the probability for the anchor observation symbol for state 1 (ξ_1) is significantly higher than the probabilities of any other symbol. For the LDA prior, there is clear separation between the median of the marginal prior distribution for the AO and

Table 3.7: State specific ϕ_i^{noise} for LDA Prior

	ϕ_{i1}^{noise}	ϕ_{i2}^{noise}	ϕ_{i3}^{noise}	ϕ_{i4}^{noise}	ϕ_{i5}^{noise}	ϕ_{i6}^{noise}	ϕ_{i7}^{noise}	ϕ_{i8}^{noise}	ϕ_{i9}^{noise}	ϕ_{i10}^{noise}
Receive Canister	0	1	1	1	1	0	0	0	0	1
Remove Cover	0	0	0	1	1	0	0	0	0	1
Position Rings	0	0	0	1	1	0	0	0	0	1
Position Canister	0	1	1	1	1	0	0	0	0	1
Operational Readiness Review	0	0	1	1	1	0	0	0	0	0
Install Initiator	0	1	1	1	0	0	0	0	0	1
Fill Canister	0	0	1	1	1	0	1	1	1	1

Table 3.8: State specific $\mathbf{b}_i^*$ for Strong Prior

	$b_i^*(\xi_1)$	$b_i^*(\xi_2)$	$b_i^*(\xi_3)$	$b_i^*(\xi_4)$	$b_i^*(\xi_5)$	$b_i^*(\xi_6)$	$b_i^*(\xi_7)$	$b_i^*(\xi_8)$	$b_i^*(\xi_9)$	$b_i^*(\xi_{10})$
Receive Canister	0.8	0.04	0.04	0.04	0.04	0	0	0	0	0.04
Remove Cover	0	0	0.8	0.05	0.05	0	0.05	0	0	0.05
Position Rings	0	0	0.8	0.05	0.05	0	0.05	0	0	0.05
Position Canister	0.8	0.04	0.04	0.04	0.04	0	0	0	0	0.04
Operational Readiness Review	0	0	0.06	0.06	0.08	0	0	0	0	0.8
Install Initiator	0.04	0.04	0.04	0.04	0.8	0	0	0	0	0.04
Fill Canister	0	0.025	0.025	0.025	0.025	0.8	0.025	0.025	0.025	0.025

the medians for the other symbols, but each marginal distribution has significantly more variability than the marginal distributions of the strong prior distribution.

3.5.4 Results and Discussion

When analyzing the simulated data, we examine plots of the posterior predictive distributions (PPD) for $Tr_i(\mathbf{X})$, T_{start}, and T_{end}, as defined in 3.3.6. When computing $Tr_i(\mathbf{X}^{(h)})$ from the posterior draws of the hidden state sequence, sometimes a particular state, i, will not be represented in the sequence either because the the first state of $X^{(h)}$ comes after i, the last state of $X^{(h)}$ comes before i, or i is short enough that the activity corresponding to state i is completed in the time gap between observations. In these cases, observations from activity i are censored and we assign $Tr_i(X^{(h)}) = 0$.

Consider inference in the low noise setting. Since our observation window begins when the process is in activity 1 and ends when the process is in activity 5, we only look at the posterior predictive distributions for $Tr_1(\mathbf{X})$-$Tr_6(\mathbf{X})$. Figures 3.16 and 3.17 display the posterior predictive distributions for $Tr_1(\mathbf{X})$-$Tr_6(\mathbf{X})$ when employing the LDA prior and strong prior respectively with the true transition time indicated by the black line. For both priors, we have the most uncertainty about the transition after activity 2. A priori, activities

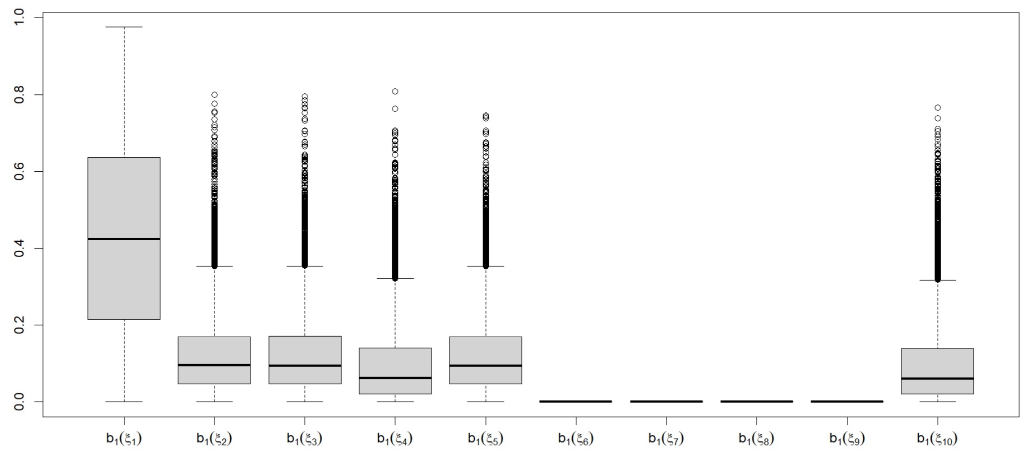


Figure 3.14: Prior draws from LDA Prior for $\mathbf{b}_1$

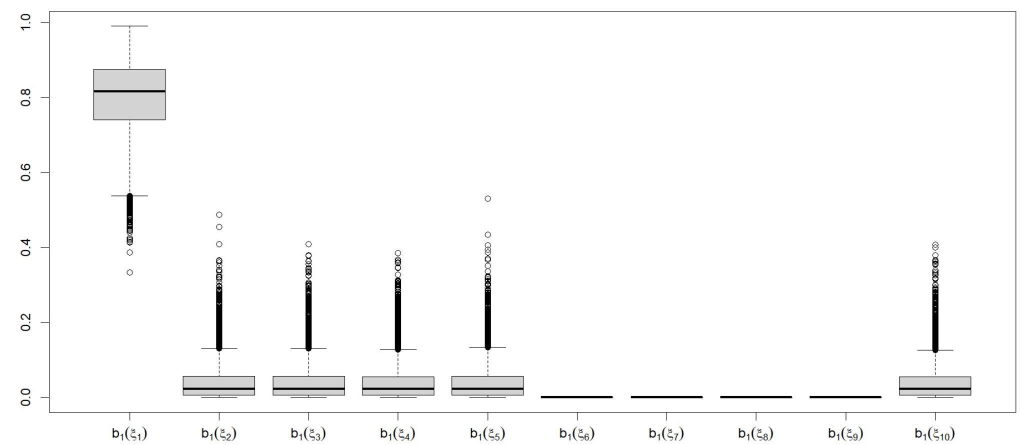


Figure 3.15: Prior draws from Strong Prior for $\mathbf{b}_1$

2 and 3 are expected to look very similar (same AO). Thus, given the data and the prior, it can be difficult to tell when activity 2 ends and activity 3 begins. This explains the wider spread for the posterior distribution of $Tr_2(\mathbf{X})$. The inference also indicates we may not have observations from the process during activities 2 and 3 (spikes at zero for $Tr_2(\mathbf{X})$ and $Tr_3(\mathbf{X})$) because these activities are expected to be very short relative to the other activities. For both priors it is clear that the observation of the process began during activity 1 and ended during activity 5 (spike at 25 for $Tr_5(\mathbf{X})$ and spike at zero for $Tr_6(\mathbf{X})$).

The black diamonds on the bottom of the plots represent the 95% highest posterior density (HPD) credible set for the transition times. These credible sets, denoted $Cr(Tr_i(\mathbf{X}))$, are the smallest finite collection of transition times such that $\sum_{j\in Cr(Tr_i(\mathbf{X}))}\sum_{h=1}^{N}\frac{I(Tr_i(\mathbf{X}^{(h)})=j)}{N} \geq 0.95$, where N is the number of posterior samples. That is, the credible set is the smallest

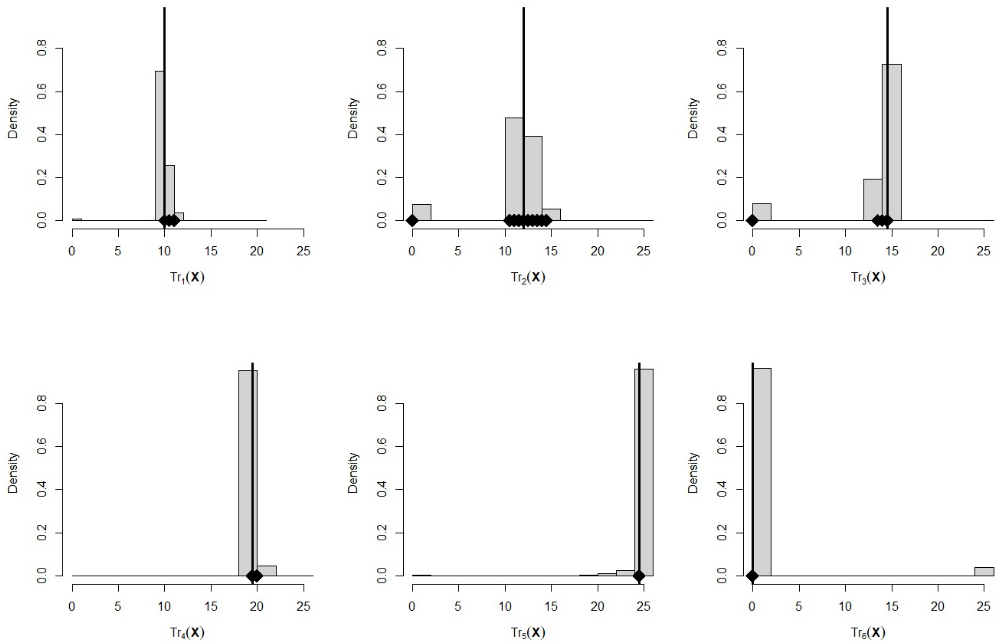


Figure 3.16: PPD of $Tr_i(\mathbf{X})$, LDA Prior-Low Noise ($Tr_i(X^{(h)}) = 0$ if i is censored)

set of potential transition times whose combined posterior probability is at least 0.95. We use discrete credible sets rather than continuous, connected credible intervals because the $Tr_i(\mathbf{X}^{(h)})$ must equal one of the n discrete observation times or 0. From the plots we can see that there are sometimes large gaps between the points in the credible set because of the sometimes multi-modal nature of the posterior predictive. A likely candidate for a transition time (corresponding to a mode) is often sandwiched between sets of AOs from different states. Time points nearby these likely candidates, however, tend to be surrounded on both sides by AOs from only one of the two states, making these times much less likely to be transition times. This causes the multi-modal structure of the posterior. The HPD credible sets allow us to capture the multiple modes of the transition time posterior predictive while excluding the low probability regions between the modes.

Figure 3.18 displays the posteriors for T_{start} and T_{end} for the LDA and strong prior distributions respectively. Note that the time until the end of the process is projected to be much longer than the time since the start of the process. This is reasonable since our observation window began in the middle of activity 1, but ends before activity 5 is finished. Activities 6 and 7 must finish before the process is over. For both T_{start} and T_{end}, both prior distributions yield similar results. This is expected as the distributions for T_{start} and T_{end}

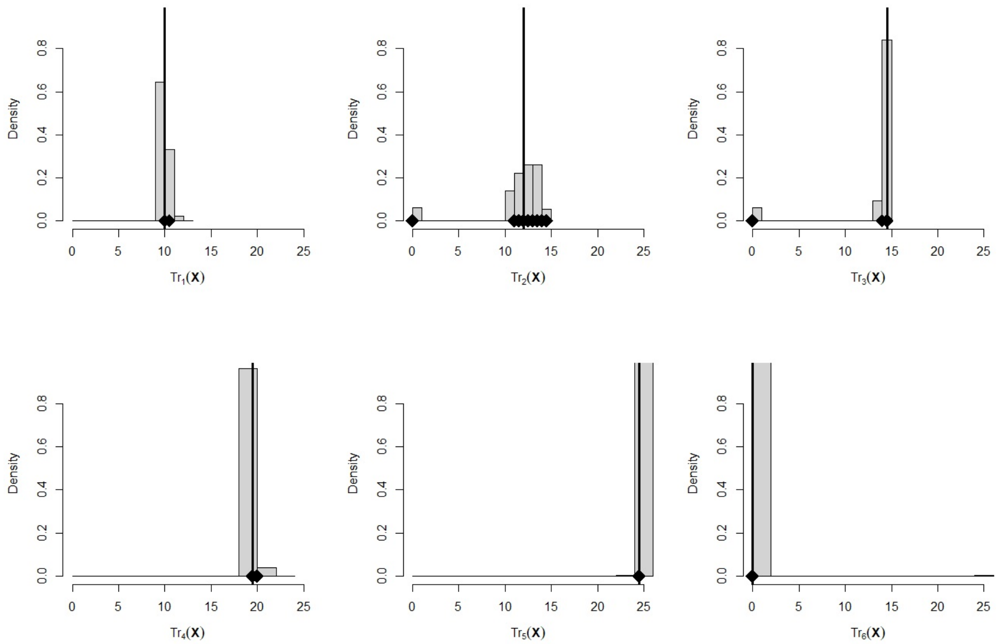


Figure 3.17: PPD of $Tr_i(\mathbf{X})$, Strong Prior-Low Noise ($Tr_i(X^{(h)}) = 0$ if i is censored)

only depend on what state we believe the process was in when we started and ended our observations. For both prior distributions, our inference strongly suggests that the process was in activity 1 when our observation started and was in activity 5 when our observation ended, which corresponds to the simulation ground truth.

Consider the inference results for the high noise case, which are shown in Figures 3.19 and 3.20. For both prior distributions, there is less posterior mass concentrated near the true transition time for many of the $Tr_i(\mathbf{X})$. This is because the observations due to incidental and concealment noise become more frequent in all states, making the states more difficult to distinguish from each other and the true transition times more difficult to identify. For both priors, there are spikes at 0 for $Tr_1(\mathbf{X})$, $Tr_2(\mathbf{X})$, and $Tr_3(\mathbf{X})$ indicating that for many samples of the hidden states, activities 1, 2, and 3 were not represented. Thus, the inference indicates our observation of the process began in activity 3 or 4, instead of activity 1. Despite this, when using the LDA prior, the credible sets for all 6 transitions contain the true transition time. When using the strong prior only the credible sets for $Tr_4(\mathbf{X}) - Tr_6(\mathbf{X})$ contain the true transition time. We hypothesize that the LDA prior more flexibly represents concealment and incidental noise yet is informative enough to distinguish between states even in the presence of noise.

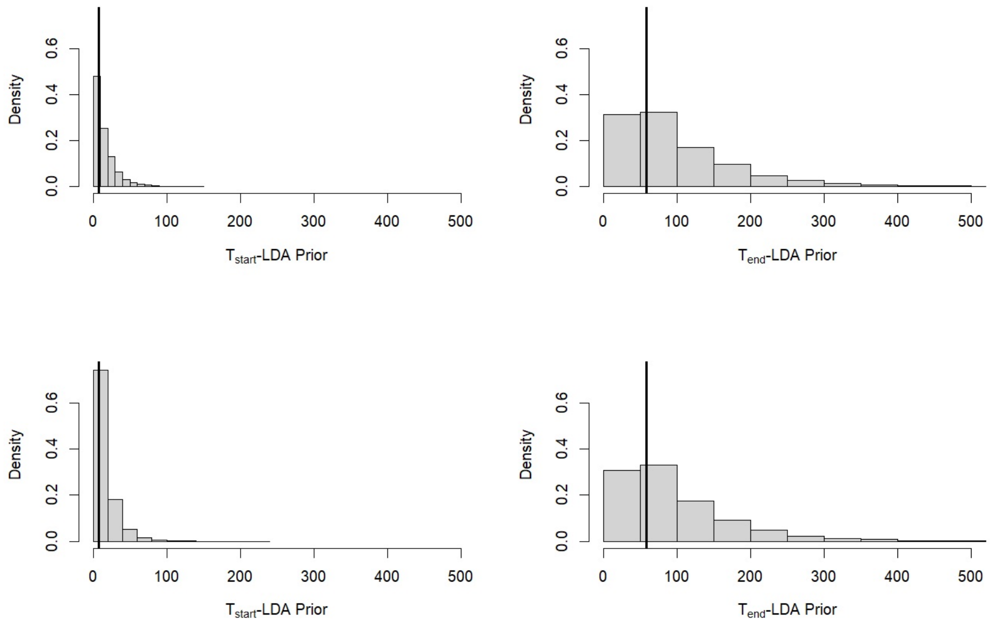


Figure 3.18: PPD for T_{start} (Left) and T_{end} (Right) for LDA (Top Row) and Strong Prior (Bottom Row) in Low Noise Scenario

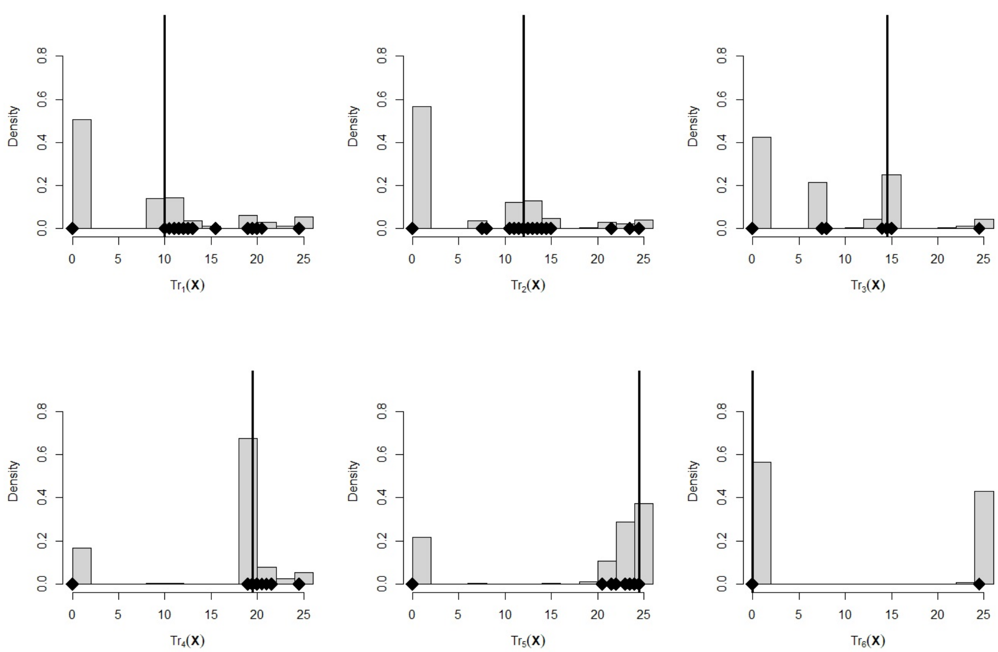


Figure 3.19: PPD of $Tr_i(\mathbf{X})$, LDA Prior-High Noise ($Tr_i(X^{(h)}) = 0$ if i is censored)

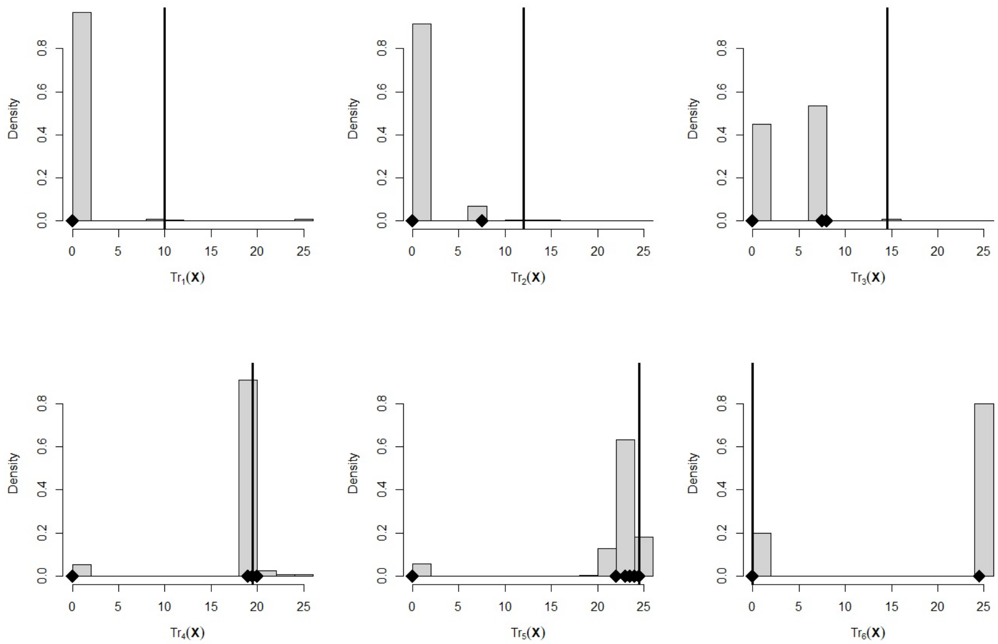


Figure 3.20: PPD of $Tr_i(\mathbf{X})$, Strong Prior-High Noise ($Tr_i(X^{(h)}) = 0$ if i is censored)

Figure 3.21 displays the posterior for T_{start} and T_{end}. Once again, the posterior distributions are similar for both the prior distributions. The posterior distributions are more variable than they were in the low noise case and the mean value for T_{start} has increased for both prior distributions (15.3 and 15.1 vs. 21.1 and 25.5) since the transition time inference indicates that the observation period began after activity 1 and 2 were finished.

3.6 Conclusion & Future Work

This chapter presents a Bayesian modeling framework for monitoring hidden arms development processes by combining subject matter expertise with noisy equipment observations. By reparameterizing the transition probabilities as a function of time between observations, we incorporate activity duration into the model allowing us to answer key process monitoring questions with uncertainty quantification. When employing the proposed framework, we demonstrated that the order of the observations can impact inference. This would imply that the ordering of the observations, not just the observations themselves, contains information about the process that is useful for inference. From the case study, we can

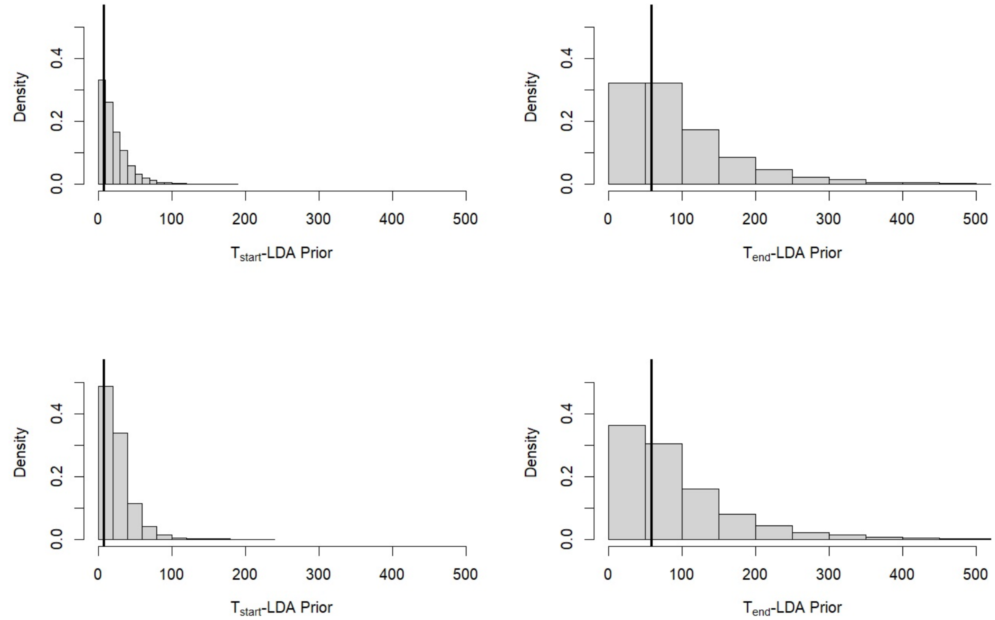


Figure 3.21: PPD for T_{start} (Left) and T_{end} (Right) for LDA (Top Row) and Strong Prior (Bottom Row) in High Noise Scenario

see the importance of using a prior specification that is well-informed by subject matter expertise including prior information about potential noise.

While our proposed approach represents an initial framework for analyzing data emitted by hidden arms control processes, there are simplifications in our proposed approach that must be relaxed to make the framework more generally applicable. While the assumption of exponentially distributed durations is mathematically convenient, the form of the exponential distribution is very restrictive and may not agree with our experts' intuition about the actual distribution of activity duration. Future work should accommodate non-exponentially distributed durations. We assume that the processes we monitor are sequential processes in which only one activity is underway at any given time. For many real-world processes, several activities may be running in parallel. We also assume only one stream of observed data. In many arms control monitoring scenarios we have access to multiple data streams including satellite imagery, seismic readings, environmental readings, etc. Future work should incorporate these other streams of data. Performing inference with multiple data streams can be particularly helpful in noisy environments as the combined data signal is stronger than the signal in any one data stream.

CHAPTER

4

UNCERTAIN TEXT CLASSIFICATION FOR PROLIFERATION DETECTION

4.1 Introduction

With the advent of nuclear weapons in the middle of the 20th century, a key global security policy, supported by the United States and other foreign powers, is nuclear nonproliferation. One goal of nuclear nonproliferation is keeping nuclear weapons technology from spreading, especially to state and non-state actors who do not yet possess nuclear weapons capabilities. The development of nuclear weapons is a highly technical undertaking, requiring research into nuclear materials, explosives engineering, physics, and other technical disciplines that have applications for civilian and military use (dual-use). Because of this, the research activities of scientists can be an important indicator of potential proliferation activity. To aid in detecting potential proliferation-related research in dual-use contexts, some authors have suggested constructing a machine learning frameworks that can automatically classify scientific research documents to different scientific disciplines, some of which may be related to proliferation activities (Burke et al. 2021; Lee et al. 2021).

In the last several years, natural language processing (NLP) has increasingly relied on deep learning models for text classification. One such model that has achieved state-of-the-art results is the Bidirectional Encoder Representations from Transformers (BERT) family of models (Devlin et al. 2018). BERT models are a type of pre-trained deep learning model, which means the classification model goes through two phases of training (pre-training and fine-tuning). During pre-training the model is supplied unlabeled documents and learns bidirectional semantic relationships between words by performing fill-in-the-blank tasks, also called masked language modeling. This "teaches" the model the language and the subject domain in which one wishes to perform document classification. Pre-trained models output high dimensional vector representations of the documents that are useful when performing NLP tasks like document classification, sentiment analysis, entity recognition, question answering, etc. To perform document classification, a final classification neural network layer is attached to the BERT model. We then use labeled documents to fit the final classification layer and adjust the pre-trained model to the classification task; this process is called fine-tuning.

Whereas BERT models can achieve a high-degree of classification accuracy, there has been recent interest in better quantifying the uncertainty associated with these classifications (Sankararaman et al. 2022; Xue et al. 2021). Good uncertainty quantification (UQ) is particularly important for higher risk applications like proliferation detection, where it is important for the model to accurately reflect its uncertainty when it is unsure about a classification.

Good classification UQ should meet three objectives: it should be well-calibrated, have good coverage, and should be understandable to human decision-makers. By well-calibrated, we mean the class probabilities predicted by the model for a given object should reflect our uncertainty about the classification of that object. The maximum class probability should be high, i.e. confident, when we classify correctly, and should be relatively low, i.e. uncertain, when we classify incorrectly. When we report our classification to a decision maker, we might report a set of several plausible classification labels, as indicated by the uncertainty quantification, in addition to the class with the highest classification probability. If our uncertainty quantification has good coverage, even when the predicted class is the wrong class, this set of plausible class predictions should contain the right class with high probability. Finally, the set of predicted class probabilities should make sense to a human decision-maker. Classes with a similar predicted probability should be semantically related. For example, if we have a document about nuclear physics, we would expect our classifier to indicate that there is a high probability the document is about labels related to nuclear physics, nuclear energy, nuclear engineering, etc. and a low probability that the document is about other scientific disciplines like biology or health sciences. The classification results would not agree with our intuition if the document was likely to be about nuclear power plant design or botany, but unlikely to be about the nuclear fuel cycle or nuclear waste disposal.

Unfortunately, the current approaches for deep classification uncertainty quantification do not scale well to NLP models and tend to focus only on achieving well-calibrated uncertainty quantification, ignoring the need for good coverage and human-interpretable class probabilities. In this chapter, we introduce an extension of the Bayesian belief matching framework proposed by Joo et al. (2020) that easily scales to large NLP models and better achieves the desired uncertainty quantification properties.

4.2 Related Work

Since BERT models are just very large, deep neural network models, many of the uncertainty quantification methods developed for deep classification models could in theory be applied to BERT language models. For a thorough review of uncertainty quantification for deep learning see Gawlikowski et al. (2021).

A deep learning classifier is multi-layered neural network model characterized by a set of weights, $\mathbf{W}$. The network weights are learned from a set of training data, $\mathscr{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ where $\mathbf{x}_i$ is a vector representation of an object, e.g., a text or an image, and y_i is the corresponding class label. Let K be the number of classes used for classification. For each object, the network returns a set of K estimated class probabilities: $\hat{\pi}(\mathbf{x}) = (\hat{\pi}_1(\mathbf{x}), \ldots, \hat{\pi}_K(\mathbf{x}))$. The class probabilities are used to predict the label of a given object where $\hat{y} = \text{argmax}_{k\in\{1,\ldots,K\}}\hat{\pi}(\mathbf{x})$. The $\hat{\pi}(\mathbf{x})$ are functions of the neural network and its weights:

$$\hat{\pi}(\mathbf{x}) = \text{softmax}(\mathbf{f}(\mathbf{x}, \mathbf{W})) = \left(\frac{e^{f_1(\mathbf{x},\mathbf{W})}}{\sum_{j=1}^{K} e^{f_j(\mathbf{x},\mathbf{W})}}, \ldots, \frac{e^{f_K(\mathbf{x},\mathbf{W})}}{\sum_{j=1}^{K} e^{f_j(\mathbf{x},\mathbf{W})}} \right),$$

where $\mathbf{f}(\cdot, \mathbf{W}) : \mathbb{R}^h \to \mathbb{R}^K$ is a multi-layered neural network model that maps the object representations, $\mathbf{x}$, with dimension h into vectors of length K. We denote the output of $\mathbf{f}(\cdot, \mathbf{W})$ as $\mathbf{z}(\mathbf{x})$ and refer to these outputs as logits.

When characterizing uncertainty for a deep learning model, it is common to distinguish between two types of uncertainty: aleatoric and epistemic uncertainty. Aleatoric uncertainty refers to variability inherent in the data, often due to imperfect measuring devices or imperfect data representation. Epistemic uncertainty refers to uncertainty in the model structure and model parameters. The class probabilities represent aleatoric uncertainty for classification (Gawlikowski et al. 2021). Quantifying aleatoric uncertainty is usually not enough. One must also have a way to account for epistemic uncertainty. Two ways in which epistemic uncertainty may be modeled are through weight-space uncertainty quantification and output-space uncertainty quantification. In most applications of uncertainty quantification in the deep learning literature, the aim is to use uncertainty quantification to obtain better calibrated models (Gustafsson et al. 2020; Gawlikowski et al. 2021; Rahaman et al. 2021; Wilson 2020) or to perform out-of-distribution (OOD) detection (Gawlikowski et al. 2021; Malinin and Gales 2018; Liu et al. 2020). The idea behind OOD detection is that objects that are out of the domain of the training data should have greater predictive uncertainty. Thus, we can use the level of uncertainty associated with the classification of a specific object to determine whether or not the object is within the data domain of our trained deep learning model.

4.2.1 Weight-Space Uncertainty Quantification

In weight-space approaches, we quantify uncertainty associated with the model weights, $\mathbf{W}$ and propagate this uncertainty to the class probabilities. It is typical to use a Bayesian framework that specifies a prior distribution for the weights and then computes a posterior distribution. These approaches are often collectively referred to as Bayesian neural networks (BNN). Because of the complexities and non-linearities of neural network models, it is not possible to obtain an exact, closed form representation of the posterior distribution. As with other Bayesian analysis problems, it is possible in theory to use Markov Chain Monte Carlo techniques to draw samples from the weight posterior (Neal 2012; Zhang et al. 2019).

For larger deep learning models with hundreds of millions or billions of parameters, however, it is impractical to generate and store multiple posterior samples. As a result, it has become more common to use closed-form approximations to the posterior. A classical Bayesian approach to closed-form posterior approximation is the Laplace approximation. This approach has been successfully adapted to weight-space uncertainty quantification for neural networks (Daxberger et al. 2021; Ritter et al. 2018).

Another more recently developed approach for obtaining approximate closed form approximations is variational inference. In variational inference (VI), we assume that the distributional family of the posterior belongs to some distributional family $q(\cdot)$ parameterized by $\boldsymbol{\theta}$. We find $\boldsymbol{\theta}$ that minimizes the KL-divergence between the approximate posterior and the true posterior. VI has the computational advantages of Laplacian approximations as since we are fitting a closed-form posterior that is easy to sample from, but it can be more flexible than Laplacian approximations. It is very common to assume independent normal distributions for the weights (Blundell et al. 2015; Steinbrener et al. 2020). Some authors have also suggested VI techniques that assume multivariate normal posteriors with some correlation structure (Louizos and Welling 2016; Posch and Pilz 2020). Whereas it is common to use a normal distribution for the weight prior distribution, some authors have suggested using a horseshoe prior to induce sparsity in the weights (Ghosh et al. 2018).

There are also some approaches that employ tools commonly used for improving neural network fit to implicitly perform epistimic uncertainty quantification for the model. Gal and Ghahramani (2016) show that dropout can be used to obtain approximate posterior draws for the class probabilities, and they show that training with dropout is approximately equivalent to Bayesian variational inference. Lakshminarayanan et al. (2017) propose using ensembles of models to quantify uncertainty, and Hoffmann and Elster (2021) show this can also be viewed as a form of approximate Bayesian inference.

Unfortunately, many of the Bayesian approaches reviewed previously require modifications to the model to obtain uncertainty estimates. This can make these approaches difficult to use with existing implementations of transformer-based NLP models like those available in PyTorch or Tensorflow. It is also challenging to specify a prior on the weights that will yield an induced posterior for the class labels with the desired characteristics described in Section 4.1. In addition, because modern NLP approaches typically have tens of millions of weights, it is computationally and conceptually challenging to specify priors and posteriors for each weight. Despite this, authors have used independent Gaussian variational inference (Xue et al. 2021) and dropout (Sankararaman et al. 2022) for performing uncertainty quantification with BERT NLP models. Xue et al. (2021), however, do not apply their approach to document classification, and Sankararaman et al. (2022) demonstrate the effectiveness of their approach in performing active learning, but do not demonstrate that their approach yields good results for the uncertainty quantification properties we discuss in Section 4.1.

4.2.2 Output-Space Uncertainty Quantification

Because of the challenges associated with performing Bayesian inference in the weight space, many authors have suggested modeling uncertainty in the class probabilities. Instead of treating the model weights as uncertain, they treat the class probabilities as uncertain.

In all the approaches we discuss, the authors fit a Dirichlet distribution to the class probabilities where the concentration parameters of the Dirichlet distributions are functions of the neural network model.

Some of these approaches are not Bayesian. Sensoy et al. (2018) use Dempster-Shafer evidential reasoning to propose a loss function that can be used to learn a Dirichlet distribution for $\pi(\mathbf{x})$. Their aim is to produce uncertainty quantification that improves OOD detection. The authors show that when applied to image classification, their approach produces greater uncertainty for images outside the training data distribution than standard BNN approaches. Similarly, Malinin and Gales (2018) learn a Dirichlet distribution for the class probabilities to model what they call distributional uncertainty, which is uncertainty in classification due to objects outside of the training data distribution. They propose a loss function based on KL-divergence that encourages flat, high-entropy Dirichlet distributions for OOD objects and sharp, low-entropy Dirichlet distributions for in-distribution objects.

Other authors use Bayesian modeling of the class probability uncertainty. Chen et al. (2018) assume a Dirichlet prior and use variational inference methods to fit a Dirichlet posterior for $\pi(\mathbf{x})$. As in Malinin and Gales (2018), the main goal of performing the uncertainty quantification is also to perform OOD detection, and they alter the standard evidence lower-bound (ELBO) loss function used in VI to make the posteriors for OOD objects more uncertain. Joo et al. (2020) propose an approach similar to variational inference called belief-matching. They use a flat Dirichlet prior with all concentration parameters equal to 1 and assume a Dirichlet posterior for $\pi(\mathbf{x})$. They use the ELBO to fit the Dirichlet posterior. Their aim with this approach is to demonstrate how the prior regularizes the model thus preventing over-fitting yielding better calibrated class probabilities. Hobbhahn et al. (2022) assume a Gaussian prior on the weights of the final layer and use the Laplace approximation to obtain a Gaussian posterior on the weights. This induces a multivariate Gaussian posterior on the logits, $\mathbf{z}(\mathbf{x})$. The authors use a Dirichlet approximation of the softmax-transformed Gaussian logits called the Laplacian bridge as the posterior distribution for $\pi(\mathbf{x})$. Both Hobbhahn et al. (2022) and Joo et al. (2020) emphasize the computational advantages of their approach over weight-space approaches.

The output-modeling approaches are easier to scale to large models because they only model uncertainty for the K output probabilities directly rather than through the hundreds of millions of model weights. It is also easier to specify meaningful priors in the output space than it is in the weight space. All these approaches model the posterior distribution for $\pi(\mathbf{x})$ as a Dirichlet distribution. While the Dirichlet distribution is a natural choice for modeling vectors on the simplex, its expressiveness is somewhat limited. The Dirichlet has a restrictive correlation structure. Thus, it can be challenging to specify a Dirichlet prior that reflects our prior understanding of similarity between classes. It can also be challenging to control the variability of the Dirichlet distribution as the mean and variance of the Dirichlet distribution are both controlled by the same set of concentration parameters.

4.2.3 Modeling Class Similarity

One of the characteristics of good uncertainty quantification for our application is that the UQ reflect our prior beliefs about the similarity structure between classes. Other authors have considered the role of class similarity in deep learning image classification (Negi et al. 2018; Olmo et al. 2020; Tong et al. 2019). Standard cross-entropy loss, the loss function typically employed for learning deep classification models, penalizes all misclassifications equally, which can often lead to class probability distributions that are difficult to interpret and do not agree with our human expectations. This can, in turn, lead human decision makers to have a lack of trust in the output of the classifier. Methods that make use of class similarity alter the standard loss function to assign smaller penalties for misclassifying to "similar" classes and higher penalties for missclassifying to classes that are not "similar". This produces similar class probabilities for semantically similar classes and in instances where there is misclassification, the selected class tends to be semantically similar to the ground truth class (Tong et al. 2019).

For our application, we also wish to incorporate class similarity into the model while capturing uncertainty in the output class probabilities. When learning our Bayesian model, we wish to encourage classes that are considered semantically similar to have similar class probabilities. We do this by incorporating class similarity structure into our model prior distribution.

4.3 Methodology

Because of its scalability to large models and its Bayesian interpretation, our proposed framework is an extension of the belief-matching framework proposed in Joo et al. (2020). In that work, the class probabilities, $\pi(\mathbf{x})$, are assumed a priori to be Dirichlet distributed random variables with concentration parameters all equal to 1, $\pi(\mathbf{x}) \sim Dir_K(1,...,1)$. To obtain a posterior distribution for $\pi(\mathbf{x})$, the authors adopt an approach similar to variational inference in which they assume a Dirichlet posterior distribution with the concentration parameters being equal to the exponentiated logits produced by the neural network classifier model, $q_{\pi(\mathbf{x})}(\mathbf{W}) = Dir_K(e^{z_1(\mathbf{x})}, \ldots, e^{z_K(\mathbf{x})})$. They fit the weights and obtain the corresponding Dirichlet posterior by maximizing the ELBO used in variational inference:

$$\mathscr{L}_{BM}(\mathbf{W}; \mathbf{x}, y) = -E_{q_{\pi(\mathbf{x})}(\mathbf{W})} \log \pi_y(\mathbf{x}) + \lambda KL(q_{\pi(\mathbf{x})}(\mathbf{W}) || p(\pi(\mathbf{x}))),$$

where $p(\pi(\mathbf{x}))$ is the Dirichlet prior distribution assumed for the class probabilities, λ is a tuning parameter that controls the influence of the prior on the posterior approximation, and $KL(\cdot||\cdot)$ is the KL divergence between two distributions. The authors show that the ELBO can be expressed analytically and is thus very simple to implement as a loss function for learning the model weights.

Instead of specifying a prior and approximate posterior for $\pi(\mathbf{x})$, we develop a framework analogous to the belief-matching framework that fits an approximate posterior for the logits. We do this because unlike $\pi(\mathbf{x})$, which has entries constrained to sum to one, the logits are unconstrained, so their prior and posterior distributions can be multivariate Gaussians. The softmax-transformed multivariate Gaussian is more expressive than the Dirichlet distribution, and it is much easier to control the variance-covariance structure of the class probabilities. This will be an important feature in allowing us to achieve the UQ characteristics we want.

The softmax transformation of a multivariate Gaussian does not have a known closed form, and its moments and covariances cannot be expressed analytically (Daunizeau 2017). This makes it challenging to state precisely how the variance-covariance structure for $\mathbf{z}(\mathbf{x})$ impacts the distribution for $\pi(\mathbf{x})$. To gain some intuition, we examine the results of a simple Monte Carlo study. Suppose we have a three-class problem, and we have three alternative multivariate Gaussian distributions for our logits:

$$\mathbf{z} \sim N\left(\begin{pmatrix} 0 \\ 0, \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right), \ \mathbf{z} \sim N\left(\begin{pmatrix} 0 \\ 0, \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0.85 & 0 \\ 0.85 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right), \ \mathbf{z} \sim N\left(\begin{pmatrix} 0 \\ 0, \\ 0 \end{pmatrix} \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix}\right).$$

The respective Monte Carlo estimates for the covariance and correlation matrices of the class probabilities (softmax transforms of $\mathbf{z}$) are

$$\text{cov}(\pi) = \begin{bmatrix} 0.05 & -0.02 & -0.03 \\ -0.02 & 0.05 & -0.03 \\ -0.03 & -0.03 & 0.05 \end{bmatrix}, \ \text{corr}(\pi) = \begin{bmatrix} 1.00 & -0.49 & -0.48 \\ -0.49 & 1.00 & -0.52 \\ -0.48 & -0.52 & 1.00 \end{bmatrix}$$

$$\text{cov}(\pi) = \begin{bmatrix} 0.02 & 0.01 & -0.03 \\ 0.01 & 0.02 & -0.03 \\ -0.03 & -0.03 & 0.05 \end{bmatrix}, \ \text{corr}(\pi) = \begin{bmatrix} 1.00 & 0.32 & -0.81 \\ 0.32 & 1.00 & -0.82 \\ -0.81 & -0.82 & 1.00 \end{bmatrix}$$

$$\text{cov}(\pi) = \begin{bmatrix} 0.11 & -0.05 & -0.05 \\ -0.05 & 0.10 & -0.05 \\ -0.05 & -0.05 & 0.11 \end{bmatrix}, \ \text{corr}(\pi) = \begin{bmatrix} 1.00 & -0.50 & -0.49 \\ -0.50 & 1.00 & -0.51 \\ -0.49 & -0.51 & 1.00 \end{bmatrix}.$$

Note that due to the high positive correlation between z_1 and z_2 for the second distribution for the logits, there is a substantial positive increase in the correlation between π_1 and π_2. As a consequence, the proportion of samples where the ranked class probabilities for z_1 and z_2 are adjacent is 0.87 for distribution 2, but only 0.66 for distributions 1 and 3. Thus, the higher logit correlation causes classes 1 and 2 to have more similiar class probabilities. Because of the sum-to-one constraint, this increase in positive correlation between π_1 and π_2 causes a lower variance for π_1 and π_2 relative to π_3. We also see from the third distribution for the logits that by increasing the variance of the logits, the marginal variances for the class probabilities also increases. This small numerical experiment inspires us to consider multivariate Gaussian prior distributions and variational posterior distributions for the logits that allows us to capture class similarity structure through the prior distribution's correlation structure.

4.3.1 Logit Uncertainty Quantification, Correlated Belief-Matching

We begin with the simplifying assumption that we have a way of knowing or estimating the similarity structure between classes. As we are classifying texts, it may be possible to obtain measures of class similarity by computing some aggregation of pairwise semantic similarities between texts in two classes (Sunilkumar and Shaji 2019). We may also be able to obtain some measure of class similarity from experts in the domain for which we are building our text classifier. Because we use the class similarities to specify the multivariate Gaussian correlations between logits, the similarity metric must be rescaled to lie between -1 and 1 and the resulting correlation matrix must be positive semi-definite (psd). If the initial correlation matrix is not psd, we can find the nearest valid correlation matrix using a method similar to the one presented in Qi and Sun (2006).

Once we have a fully-specified prior correlation matrix, which we denote as $\mathbf{P}_0$, we can specify the full prior for the logits as

$$\mathbf{z}(\mathbf{x}) \sim N_K\left(\mathbf{0}, \Sigma_0\right),$$

where $\Sigma_0 = \sigma_0^2 \mathbf{P}_0$, with σ_0^2 being chosen by the user. As with the original belief matching framework, we minimize the negative ELBO to fit the approximate posterior, $q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})$ for the logits. We use a different representation of the ELBO similar to the one used in Ghosh et al. (2018):

$$\begin{aligned}
\mathscr{L}_{CorrBM}(\boldsymbol{\theta}; y, \mathbf{x}) &= -E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})} \log p(y, \mathbf{z}(\mathbf{x})) - \psi H(q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})) \\
&= -E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})} \log p(y|\mathbf{z}(\mathbf{x})) - E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})} \log p(\mathbf{z}(\mathbf{x})) - \psi H(q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})) \\
&= -E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})} \log \pi_y(\mathbf{x}) - E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})} \log p(\mathbf{z}(\mathbf{x})) - \psi H(q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})),
\end{aligned}$$

where $\pi_y(\mathbf{x})$ is the class probability corresponding to class label y, $p(\mathbf{z}(\mathbf{x}))$ is the prior density for the logits, and $H(q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta}))$ is the entropy for the variational posterior. In the standard ELBO, $\psi = 1$. We show in our numerical experiments that by choosing different values for ψ, we can control the posterior variance for the logits which allows us to obtain uncertainty quantification with good coverage. As in Joo et al. (2020), we minimize $\mathscr{L}(\boldsymbol{\theta}; y, \mathbf{x})$ using stochastic gradient descent, and we evaluate the batch-wise loss as $\frac{1}{B}\sum_{b=1}^{B} \mathscr{L}(\boldsymbol{\theta}; y_b, \mathbf{x}_b)$, where B is the batch size.

A natural choice for $q_{\mathbf{z}(\mathbf{x})}$ is a multivariate Gaussian. Similar to the original belief-matching approach, we assume that the posterior mean and variance of the logits are parameterized by the network. To do this, we double the dimension of the network output, which we had denoted as $\mathbf{f}(\mathbf{x}, \mathbf{W})$. The first K entries of the network output, which we denote as $\mathbf{f}(\mathbf{x}, \mathbf{W})$, correspond to the logit means. The last K entries of the network output, which we denote as $\mathbf{h}(\mathbf{x}, \mathbf{W})$, are transformed as $\mathbf{h}^{\dagger}(\mathbf{x}, \mathbf{W}) = \log(1 + e^{\mathbf{h}(\mathbf{x}, \mathbf{W})})$, so that the entries of $\mathbf{h}^{\dagger}(\mathbf{x}, \mathbf{W})$ are positive and represent valid standard deviations as suggested in Blundell et al. (2015). This is similar to the approach proposed in Kendall and Gal (2017).

Thus we have

$$q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta}) = N_K\left(\mathbf{f}(\mathbf{x}, \mathbf{W}), \Sigma(\mathbf{x}, \mathbf{W})\right),$$

where $\Sigma(\mathbf{x}, \mathbf{W}) = D(\mathbf{h}^{\dagger}(\mathbf{x}, \mathbf{W}))^T \mathbf{P} D(\mathbf{h}^{\dagger}(\mathbf{x}, \mathbf{W}))$. Here, $D(\mathbf{h}^{\dagger}(\mathbf{x}, \mathbf{W}))$ is a diagonal matrix with diagonal entries equal to $\mathbf{h}^{\dagger}(\mathbf{x}, \mathbf{W})$ and $\mathbf{P}$ is a correlation matrix characterized by $\frac{K(K-1)}{2}$ unique correlation parameters which we denote as $\boldsymbol{\rho}$. The $\boldsymbol{\rho}$ pose a challenge to standard deep learning procedures because they are restricted to be between -1 and 1 and $\mathbf{P}$ is constrained to be positive semi-definite. To avoid having to perform a much more complicated constrained optimization problem, we reparameterize $\boldsymbol{\rho}$ as $\boldsymbol{\rho}^{\dagger}$, an unconstrained vector of size

$\frac{K(K-1)}{2}$. We use the spherical parameterization described in Pinheiro and Bates (1996) to construct a unique, analytical mapping from $\boldsymbol{\rho}^{\dagger}$ to $\boldsymbol{\rho}$. Thus, our set of variational parameters is $\boldsymbol{\theta} = (\mathbf{W}, \boldsymbol{\rho}^{\dagger})$.

To make the ELBO function useable, we must describe how each of its terms can be evaluated. The first expectation, $E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})}(\log \pi_y(\mathbf{x}))$ has no analytical expression because, as we have already noted, the moments of $\pi(\mathbf{x})$ with respect to a multivariate Gaussian like $q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})$ cannot be expressed analytically. Thus, we must approximate the expectation. For us to perform valid stochastic optimization, however, we must select an approximation of the expectation that is unbiased and has a gradient that is an unbiased estimate of the gradient of the expectation. A popular approach for ensuring this property when using Gaussian variational distributions is the "reparameterization trick" introduced in Kingma and Welling (2013). We obtain a Monte Carlo approximation for the expectation by drawing $\epsilon_j \sim N(\mathbf{0}_K, \mathbf{I}_K)$, then obtain $\mathbf{z}(\mathbf{x})_{ij} = \mathbf{L}\epsilon_j + \mathbf{f}(\mathbf{x}_i, \mathbf{W})$, where $\mathbf{L}$ is the Cholesky factor of $\boldsymbol{\Sigma}(\mathbf{x}, \mathbf{W})$. We then have

$$E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})} \log \pi_{y_i}(\mathbf{x}_i) \approx \frac{1}{M} \sum_{j=1}^{M} \log \pi_{y_i}(\mathbf{z}(\mathbf{x})_{ij}),$$

where M is often taken to be 1. We also use $M = 1$.

By properties of multivariate Gaussian distributions, it is straightforward to show that

$$E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})} \log p(\mathbf{z}(\mathbf{x}) = -\frac{1}{2}(K \log(2\pi) + \log(\det(\Sigma_0)) + tr\left(\Sigma_0^{-1}\Sigma(\mathbf{x}, \mathbf{W})\right) + \mathbf{f}(x, \mathbf{W})^T \Sigma_0^{-1} \mathbf{f}(\mathbf{x}, \mathbf{W})).$$

Also by properties of multivariate Gaussian distributions, we have

$$H(q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})) = \frac{1}{2} \log \det(2\pi e \Sigma(\mathbf{x}, \mathbf{W})).$$

Thus, when we know the similarity measure between classes in advance, and these similarities can be made to form a valid prior correlation matrix for the logits corresponding to the classes, we have a variational inference framework for learning the corresponding approximate posterior for the logits that induces a posterior on the class probabilities $\pi(\mathbf{x})$ that we can use for uncertainty quantification.

4.3.2 Logit Uncertainty Quantification, Hierarchical Correlated Belief-Matching

In many applications, it may be difficult to obtain values for the class similarities. We may not be able to decide on a suitable similarity metric, and it might not be possible to elicit similarity values from experts. In this case, it may still be possible to obtain grouping information from experts. That is, we may have our experts partition the classes into G non-overlapping groups where classes in the same group are considered similar. Classes in different groups are considered unrelated and are assumed to have zero similarity a priori. In this case our prior covariance matrix would have the form

$$\Sigma_0 = \sigma_0^2 \begin{pmatrix} \mathbf{P}_0^1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_0^2 & \dots. & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}_0^G \end{pmatrix},$$

where $\mathbf{P}_0^g$ specifies the correlation structure for the labels in group g and $\mathbf{0}$ is a matrix of zeros.

Let K_g be the number of classes in group g. Then $\mathbf{P}_0^g$ has $D_g = \frac{K_g(K_g-1)}{2}$ unique parameters, which we denote $\boldsymbol{\rho}_0^g$. If we do not know what the prior correlation parameters are, we can specify the following hierarchical prior for the logits:

$$\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0 \sim N_K(\mathbf{0}, \Sigma_0)$$

$$p(\boldsymbol{\rho}_0) = \prod_{g=1}^{G} p(\boldsymbol{\rho}_0^g) = \prod_{g=1}^{G}\prod_{d=1}^{D_g} p(\rho_{0d}^g).$$

Note we are assuming all the prior correlation parameters are independent a priori.

For this formulation, we need to obtain an approximate posterior, $q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta}, \boldsymbol{\vartheta})$, for the posterior of $\mathbf{z}(\mathbf{x})$ and $\boldsymbol{\rho}_0$, $p(\mathbf{z}(\mathbf{x}), \boldsymbol{\rho}_0|\mathscr{D})$. For simplicity, we assume we can partition the variational posterior into independent parts. Specifically, we assume $q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta}, \boldsymbol{\vartheta}) = q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta}) q_{\boldsymbol{\rho}_0}(\boldsymbol{\vartheta})$ and $q_{\boldsymbol{\rho}_0}(\boldsymbol{\vartheta}) = \prod_{g=1}^{G}\prod_{d=1}^{D_g} q_{\boldsymbol{\rho}_{0d}^g}(\boldsymbol{\vartheta}_{gd})$. This is a common simplifying assumption called the mean-field approximation in the variational inference literature (Blei et al. 2017). We assume $q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})$ has the same form and parameterization as in Section 4.3.1, and $\boldsymbol{\theta} = (\mathbf{W}, \boldsymbol{\rho}^{\dagger})$.

The ELBO loss has the form

$$
\begin{aligned}
\mathscr{L}_{HierBM}(\boldsymbol{\theta},\boldsymbol{\vartheta};\mathbf{x},y) &= -E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta})}\log p(y,\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0) - H(q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta}))\\
&= -E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta})}\log p(y|\mathbf{z}(\mathbf{x})) - E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta})}\log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0)\\
&- E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta})}\log p(\boldsymbol{\rho}_0) - H(q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta}))\\
&= -E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})}\log \pi_y(\mathbf{x}) - E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta})}\log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0)\\
&- \sum_{g=1}^{G} E_{q_{\boldsymbol{\rho}_0^g}(\boldsymbol{\vartheta}_g)}\log p(\boldsymbol{\rho}_0^g) - H(q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta})).
\end{aligned}
$$

Using the mean-field approximation, we can further simplify:

$$
\begin{aligned}
\mathscr{L}_{HierBM}(\boldsymbol{\theta},\boldsymbol{\vartheta};\mathbf{x},y) = &-E_{q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})}\log \pi_y(\mathbf{x}) - E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0}(\boldsymbol{\theta},\boldsymbol{\vartheta})}\log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0)\\
&- \sum_{g=1}^{G} E_{q_{\boldsymbol{\rho}_0^g}(\boldsymbol{\vartheta}_g)}\log p(\boldsymbol{\rho}_0^g) - \psi H(q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta})) - \sum_{g=1}^{G} H(q_{\boldsymbol{\rho}_0^g}(\boldsymbol{\vartheta}_g))
\end{aligned}
\tag{4.1}
$$

We already discussed how to estimate the first term (see Section 4.3.1) using the reparameterization trick, and we also saw how to compute $H(q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta}))$. To address the other terms in the model, we must specify the exact form of $q_{\boldsymbol{\rho}_{0d}^g}(\boldsymbol{\vartheta}_{gd})$ and $p(\rho_{0d}^g)$.

To simplify the variational inference, we reparameterize the $\boldsymbol{\rho}_0^g$ as $\boldsymbol{\rho}_0^{g\dagger}$, where $\boldsymbol{\rho}_0^{g\dagger}$ are unconstrained. As with the $\boldsymbol{\rho}^\dagger$, we can employ the spherical reparameterization of correlation matrices to construct an analytical mapping from $\boldsymbol{\rho}_0^{g\dagger}$ to $\boldsymbol{\rho}_0^g$. Because $\boldsymbol{\rho}_0^{g\dagger}$ is unconstrained, we assume a priori that

$$\rho_{0d}^{g\dagger} \sim N(0,\nu^2) \text{ for all } g \text{ and } d,$$

and

$$q_{\boldsymbol{\rho}_{0d}^{g\dagger}}(\boldsymbol{\vartheta}_{gd}) = N\left(\mu_{gd},\tau_{gd}^2\right).$$

While μ_{gd} is unconstrained, τ_{gd} is constrained to be positive. Again, because it is easier to perform unconstrained optimization, we reparameterize each τ_{gd} as $\tau_{gd} = \log(e^{\tau_{gd}^\dagger}+1)$, where $\tau_{gd}^\dagger$ is unconstrained. Thus, our set of variational parameters for $\boldsymbol{\rho}_0^{g\dagger}$ is $\boldsymbol{\vartheta}_g = (\boldsymbol{\mu}_g,\boldsymbol{\tau}_g^\dagger)$. We are now able to address the other terms of the loss function.

We know immediately from properties of Gaussian distributions that

$$H(q_{\boldsymbol{\rho}_0^{g\dagger}}(\boldsymbol{\vartheta}_g)) = \frac{1}{2}\sum_{d=1}^{D_g}(\log(2\pi\tau_{gd}^2)+1)$$

and

$$E_{q_{\rho_0^{g\dagger}}(\vartheta_g)} \log p(\boldsymbol{\rho}_0^{g\dagger}) = -\frac{1}{2}\left(D_g \log(2\pi) + 2D_g \log \nu + \prod_{d=1}^{D_g} \frac{\tau_{gd}^2}{\nu^2} + \nu^{-2} \boldsymbol{\mu}_g^T \boldsymbol{\mu}_g \right).$$

The only unresolved term of the ELBO loss is $E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\theta},\boldsymbol{\vartheta})} \log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0^\dagger)$ which is an expectation with respect to the joint variational posterior distribution of $\mathbf{z}(\mathbf{x})$ and $\boldsymbol{\rho}_0^\dagger$. Because $q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\theta},\boldsymbol{\vartheta}) = q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta}) q_{\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\vartheta})$, we obtain

$$\begin{aligned} E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\theta},\boldsymbol{\vartheta})} \log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}^\dagger) &= \int\int \log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0^\dagger) q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\theta},\boldsymbol{\vartheta}) d\mathbf{z}(\mathbf{x}) d\boldsymbol{\rho}_0^\dagger \\ &= \int\int \log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0^\dagger) q_{\mathbf{z}(\mathbf{x})}(\boldsymbol{\theta}) q_{\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\vartheta}) d\mathbf{z}(\mathbf{x}) d\boldsymbol{\rho}_0^\dagger \\ &= E_{q_{\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\vartheta})}\left(E_{q_{\mathbf{z}(\mathbf{x})}} \log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0^\dagger)\right). \end{aligned}$$

Using properties of multivariate Gaussians, we have

$$E_{q_{\mathbf{z}(\mathbf{x})}} \log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}_0^\dagger) = -\frac{1}{2}\left(K \log(2\pi) + \log(|\Sigma_0|) + \mathbf{f}(\mathbf{x},\mathbf{W})^T \Sigma_0^{-1} \mathbf{f}(\mathbf{x},\mathbf{W}) + tr(\Sigma_0^{-1}\Sigma(\mathbf{x},\mathbf{W})\right)$$

which implies that

$$\begin{aligned} E_{q_{\mathbf{z}(\mathbf{x}),\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\theta},\boldsymbol{\vartheta})} \log p(\mathbf{z}(\mathbf{x})|\boldsymbol{\rho}^\dagger) = -\frac{1}{2}(K \log(2\pi) + E_{q_{\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\vartheta})}(\log(|\Sigma_0|) \\ + \mathbf{f}(\mathbf{x},\mathbf{W})^T \Sigma_0^{-1} \mathbf{f}(\mathbf{x},\mathbf{W}) + tr(\Sigma_0^{-1}\Sigma(\mathbf{x},\mathbf{W})))) \end{aligned} \tag{4.2}$$

We can approximate $E_{q_{\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\vartheta})}(\log(|\Sigma_0|) + \mathbf{f}(\mathbf{x},\mathbf{W})^T \Sigma_0^{-1} \mathbf{f}(\mathbf{x},\mathbf{W}) + tr(\Sigma_0^{-1}\Sigma(\mathbf{x},\mathbf{W})))$ using the reparameterization trick since $q_{\boldsymbol{\rho}_0^\dagger}(\boldsymbol{\vartheta})$ is Gaussian. Thus, we have shown that (4.1) can be evaluated using standard techniques and can thus be implemented as a standard ELBO loss function for learning the variational parameters.

4.3.3 Uncertainty Evaluation Metrics

In this section we review the metrics that we use to evaluate how effectively our proposed uncertainty quantification approach achieves the desired characteristics. Our uncertainty quantification should be well calibrated, have good coverage, and should reflect human expectations about class similarity.

A widely used measure of model calibration for deep learning classification models is the expected calibration error (ECE) (Gawlikowski et al. 2021). The ECE uses a function of the estimated class probabilities called the confidence which is defined as $\text{conf}(\mathbf{x}) = \max_{k\in\{1,\dots,K\}} \hat{\pi}(\mathbf{x})$. ECE is defined as

$$ECE(\mathscr{D},\mathbf{f}) = \frac{1}{N}\sum_{v=1}^{V}|b_v|\left|\sum_{i\in b_v}\frac{I(\hat{y}_i = y_i)}{|b_v|} - \sum_{i\in b_v}\frac{\text{conf}(\mathbf{x}_i)}{|b_v|}\right|.$$

The b_v's bin the data by confidence score. Each b_v is the set of objects for which the confidence score is within a partitioning sub-interval of [0,1]. A common choice is $b_1 = \{i|\text{conf}(\mathbf{x}_i)\in[0.0,0.1)\}$, $b_2 = \{i|\text{conf}(\mathbf{x}_i)\in[0.1,0.2)\}, \dots, b_{10} = \{i|\text{conf}(\mathbf{x}_i)\in[0.9,1.0]\}$. The ECE is interpreted as the weighted average across bins of the absolute difference between the accuracy and average confidence where the weights are the bin sizes, $|b_v|$.

To quantify model coverage, we employ a metric suggested in Kompa et al. (2021). This metric requires us to compute the highest-density credible set for each object in the data. The credible set for object i, $\text{cred}_\alpha(\mathbf{x}_i)$ is the smallest subset of the class labels, $\{1,\dots,K\}$, for which $\sum_{k\in\text{cred}_\alpha(\mathbf{x}_i)} p(\hat{y}_i = k|\mathscr{D}) \geq \alpha$, α is some threshold such as 0.95 and $p(\hat{\mathbf{y}}_i|\mathscr{D})$ is the posterior predictive distribution for class labels for object i. We do not have this predictive distribution in closed form, but we can approximate it by simulation by repeatedly drawing a sample of class probabilities from the variational posterior for $\pi(\mathbf{x}_i)$ and selecting the class with the maximum class probability. We approximate $p(\hat{y}_i = k|\mathscr{D})$ as the proportion of samples where label k was selected. The coverage is defined as

$$CVG(\mathscr{D},\mathbf{f}) = \sum_{i=1}^{N}\frac{I(y_i\in\text{cred}_\alpha(\mathbf{x}_i))}{N}.$$

Models with good coverage should have CVG values close to α.

To quantify the precision of the uncertainty quantification, we use average size of the credible sets as an analogue to credible interval width (Kompa et al. 2021). We denote this "width" metric as

$$WDTH(\mathscr{D},\mathbf{f}) = \sum_{i=1}^{N}\frac{|\text{cred}_\alpha(\mathbf{x}_i)|}{N}.$$

To evaluate how well the class probabilities produced by our model reflect our expectations about class similarities, we use the ranking accuracy metric proposed in Tong et al. (2019):

$$RA(\mathscr{D}, \mathbf{f}) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\sum_{j=1}^{K} \sum_{j' \neq j}^{K} |r_{ij} - r_{ij'}||s_{ij} - s_{ij'}|}{\sum_{j=1}^{K} \sum_{j' \neq j}^{K} (r_{ij} - r_{ij'})^2} \right),$$

where r_{ij} is the ranking of class j with respect to the estimated class probabilities, $\hat{\pi}(\mathbf{x}_i)$, for object i, and s_{ij} is the raking of class j with respect to its prespecified similarity to the true class for object i. Thus, the ranking accuracy measures how well the ordering of classes by the predicted class probabilities matches the ordering we would impose on the classes based on how similar we as humans perceive the classes to be.

4.4 Experiments

4.4.1 Small OSTI Data Experiments

First, we use a small text data set to demonstrate some of the differences between the correlated belief-matching framework and the original belief-matching framework and illustrate the impact of different settings of the model tuning parameters on the inference. The data is a subset of the US Department of Energy Office of Scientific and Technical Information (OSTI) text data set used in Burke et al. (2021). This data set contains abstracts of scientific articles classified by subject category. The data set consists of 12,495 texts (11,865 for training, 630 for testing) each assigned to one of six classes: Nuclear Fuels (1866 training texts, 99 test texts), Nuclear Fuel Cycle and Fuel Materials (2314 training texts, 123 test texts), Material Science (2400 training texts, 127 test texts), Chemistry (1032 training texts, 55 test texts), Military Technology, Weaponry, and National Defense (1125 training texts, 60 test texts), Physics (3128 training texts, 166 test texts). We assume a priori that classes 1 and 2 have strong semantic similarity because they are both about nuclear energy.

Our classifier is the NukeLM model developed in Burke et al. (2021) with a two layer classifier head on top. The NukeLM model is a RoBERTa model pre-trained on a large number of OSTI texts. The classifier architecture is illustrated in Figure 4.1. We train our classifier by fine-tuning the NukeLM model on our data set. We use the data set to explore the impact of using $\mathscr{L}_{BM}(\mathbf{W}; \mathbf{x}, y)$ or $\mathscr{L}_{CorrBM}(\boldsymbol{\theta}; y, \mathbf{x})$ as loss functions with different tuning parameter settings. For training our models, we use the Adam optimization algorithm with weight decay of 0.01. We fine-tuned our models for 10 epochs with a batch size of 16 and a learning rate of 10^{-5} and 445 warm-up steps.

First, we examine how changing the λ tuning parameter when using the original belief-

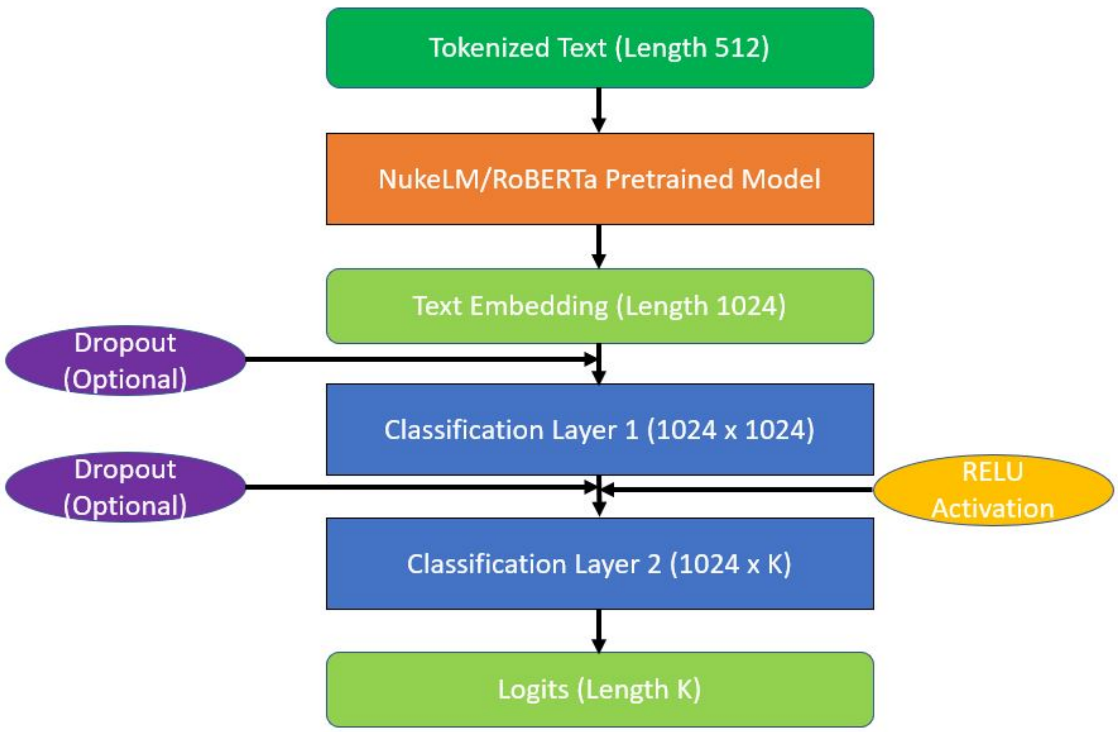


Figure 4.1: Text Classifier Architecture

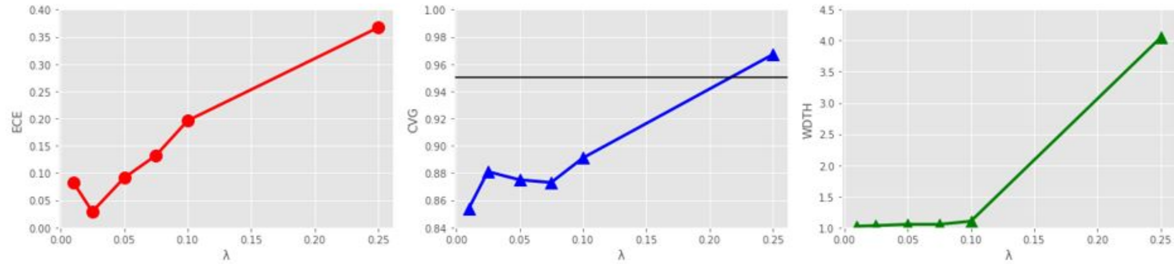


Figure 4.2: Test Set ECE (left), CVG (middle), and WDTH (right) for Different Levels of λ for Original Belief-Matching Framework

matching loss function impacts inference and uncertainty quantification. Figure 4.2 displays the test set ECE (expected calibration error), CVG (credible set coverage), and WDTH (credible set "width") after fitting the belief-matching model with different choices of λ.

As λ increases, the influence of the $Dir(1,1,...,1)$ prior distribution increases and causes the predicted class probabilities $\hat{\pi}(\mathbf{x})$ to approach a discrete uniform distribution. This regularizes the logits and causes them to get smaller and closer to each other as λ increases. This yields predictive class probabilities that are more uniform. In addition, as the magnitude of the logits decreases, this causes the magnitude of the concentration parameters for the Dirichlet variational posterior to decrease. This yields a more variable posterior. The more variable and more uniform posterior for class probabilities together explain why the increase in λ leads to an increase in coverage (CVG).

Recall that ECE measures the degree to which our classifier's confidence (magnitude of the maximum class probability) matches its accuracy. Increasing λ regularizes (shrinks) the logits and first leads to a decrease in ECE then an increase. This is because for lower values of λ, the $\hat{\pi}(\mathbf{x})$ are too confident; i.e., conf($\mathbf{x}$) is too high. For higher values of, λ, however, the predicted class are not confident enough, leading to increasing ECE. Unfortunately, in order to achieve an acceptable CVG, we often must choose a λ that leads to a very high ECE, as in this example. Thus, we cannot achieve good calibration and coverage at the same time with the original belief-matching framework.

For the correlated belief-matching loss function that we propose in Section 4.3.1, we have two tuning parameters: σ_0 and ψ. The impact of these two parameters is illustrated in Figure 4.3. In order to illustrate the individual effects of these tuning parameters we show how the uncertainty quantification metrics change as one tuning parameter is changed and the other tuning parameter is held fixed.

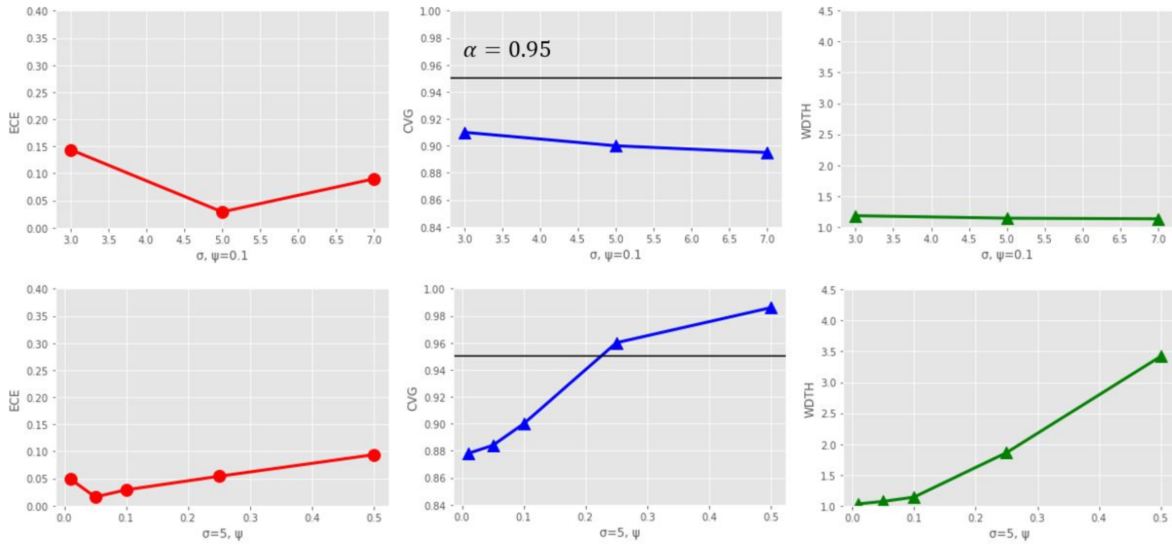


Figure 4.3: Top Row: Test Set ECE (left), CVG (middle), and WDTH (right) for Different Levels of σ_0 with $\psi = 0.1$. Bottom Row: Test Set ECE (left), CVG (middle), and WDTH (right) for Different Levels of ψ with $\sigma_0 = 5$

As σ_0 decreases, the prior becomes stronger, has more impact on the inference, and has a more powerful regularizing effect on the model fitting. This has a pronounced effect on the ECE. The regularizing effect of the prior shrinks the magnitude of the logits, and, as with the original belief-matching framework, causes the ECE to first decrease and then increase as the estimated class probabilities go from overconfident to underconfident. There is very little effect on the CVG and WDTH metrics. However, when we vary ψ, the CVG and WDTH metrics change substantially while ECE remains fairly stable. This is because ψ controls the impact of the variational posterior entropy on the loss function $\mathscr{L}_{CorrBM}$. Higher entropy corresponds to greater variability, thus, as we increase ψ, we encourage more variable variational posteriors on the logits. This leads to larger credible sets and greater values of CVG and WDTH. Increasing ψ also increases the relative impact of the entropy compared to the expected prior density term of the loss function. As ψ is increased, the regularizing effect of the expected prior density term is reduced, leading to a slightly overconfident model and an increased ECE. However, the increase in ECE that comes from choosing a ψ that yields good CVG is much lower than the ECE associated with a good CVG for original belief-matching.

This is further illustrated in Figure 4.4 which plots pairs of ECE and CVG values for different hyperparameter settings for the belief-matching and correlated belief-matching

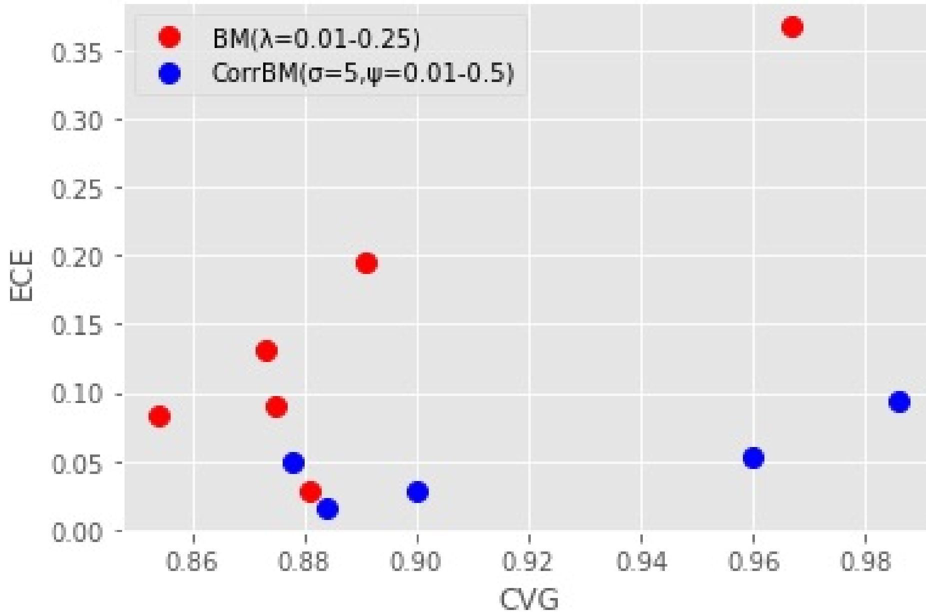


Figure 4.4: CVG vs. ECE for Small OSTI Dataset

framework. The figure shows that while for the original belief-matching model, ECE must grow substantially to reach high CVG values, ECE grows very slowly to attain high CVG values with correlated belief-matching. Thus, we can use the hyperparameters of the correlated belief-matching model to achieve uncertainty quantification that has good calibration and good coverage, which was not possible with the original belief-matching approach. Notice also the sharp dip in the plot for the original belief-matching approach. This dip corresponds to the dip in ECE illustrated in Figure 4.2 and is because if λ is too low, we do not have enough regularization from the prior and if λ is too high, we have too much regularization from the prior.

Aside from coverage and calibration, we are also interested in ensuring that the $\hat{\pi}(\mathbf{x})$ reflect what human analysts expect about class similarity, as this makes the model outputs more easily understandable and trustworthy to human users. We believe classes 1 and 2 are similar a priori. To capture this prior assumption, we use the prior correlation matrix

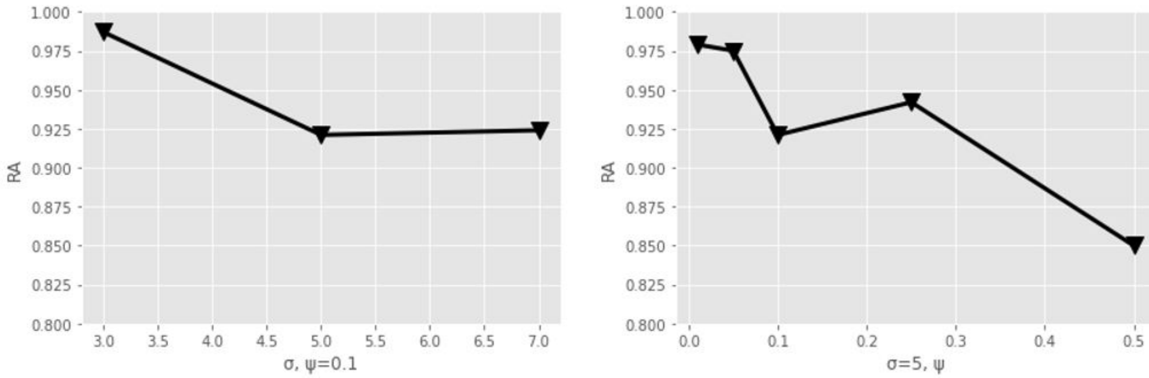


Figure 4.5: Left: RA for Different Levels of σ with $\psi = 0.1$. Right: RA for Different Levels of ψ with $\sigma_0 = 5$

$$\mathbf{P}_0 = \begin{pmatrix} 1 & 0.75 & 0 & 0 & 0 & 0 \\ 0.75 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

By fitting the model with $\mathscr{L}_{CorrBM}$, we regularize the predicted class probabilities to reflect this prior correlation structure. We wish to quantify how well the resulting class probabilities reflect our prior assumptions about class similarity. We use the ranking accuracy (RA) metric introduced in Section 4.3.3 to do this. Figure 4.5 displays how RA changes for different choices of σ_0 and ψ. Unsurprisingly, as σ_0 is decreased and the prior gets stronger, RA improves. The reason RA decreases as ψ increases is that as the variational posterior entropy is given more weight, the expected prior density term, which encourages the posterior to not depart too much from the prior distribution, has relatively less weight and magnitude. As this term's influence on the loss weakens, RA decreases. Because the original belief-matching framework does nothing to encourage particular relationships between class probabilities, the RA is very low for the original belief-matching framework. For the values of λ considered in Figure 4.2, the highest value for the RA was 0.672.

To compare our approach and the original belief-matching approach with a more commonly used deep learning uncertainty quantification approach, we consider the Monte Carlo Dropout approach suggested in Gal and Ghahramani (2016). El Anigri et al. (2021)

Table 4.1: Small OSTI Data-Method Comparison

	Accuracy	ECE	CVG	WDTH	RA	Sampling
BM($\lambda = 0.025$)	**86.7%**	0.029	88.1%	1.03	0.645	0.66 min
CorrBM($\sigma_0 = 5, \psi = 0.1$)	85.1%	0.029	90.0%	1.14	**0.921**	**0.38 min**
Drop($c_p = 0.5, h_p = 0.4, a_p = 0.55$)	82.4%	0.030	**93.7%**	1.58	0.844	51.9 min

use dropout with BERT-based text classification models and recommend using separate dropout probabilities for the hidden fully-connected layers (h_p) and the attention layers (a_p) with dropout probabilities ranging between 0.1 and 0.5. We also apply a separate dropout probability for the final classification head layer (c_p). We use a grid search to obtain the best combination of dropout probabilities. Table 4.1 compares the metrics for the original belief-matching, correlated belief-matching, and dropout methods at tuning-parameter settings chosen so that the ECE metric is comparable. Additional results with different hyperparameter settings are available in Table D.1 in Appendix D.

From the table, we see that the the belief-matching and correlated belief-matching approaches have similar accuracy and coverage values, with the correlated belief-matching approach having slightly higher coverage and slightly lower accuracy. We note, however, that while the coverage values are comparable at this calibration level, we demonstrated in Figures 4.4 and 4.3 that correlated belief-matching can attain higher coverage values with little change in ECE compared to the original belief-matching approach. Using dropout, we can obtain a much higher coverage. In order to obtain this coverage value and a comparable ECE value, however, the dropout probabilities have to be so high that the accuracy starts to degrade substantially. We also see from the table that because correlated belief-matching enforces our prior beliefs about class similarity, it produces a higher ranking accuracy than both standard belief-matching and dropout. Thus, correlated belief-matching is the best approach for producing models that yield human-understandable outputs.

The most serious disadvantage of dropout is the computational cost. Generating a posterior sample using dropout requires us to complete one forward pass through the model for every posterior sample. Both original belief-matching and correlated belief matching only require us to perform a single forward pass to obtain the variational posterior parameters for a given object. Then we can sample from the closed-form logit posterior or the class probability posterior. The last column of the table reports the computation time required to draw 200 posterior samples. These times were obtained by running the model in parallel on a machine with 16 CPU cores and 2 GPUs. The savings in computational costs when using either of the belief-matching approaches is substantial, especially for large deep learning architectures like NukeLM.

4.4.2 MNIST-Fashion Image Classification

We now compare the performance of the original belief-matching, correlated belief-matching, and dropout on the MNNIST-fashion image classification data set. The data set consists of 70,000 images (60,000 training images and 10,000 testing images) of clothing items classified into 10 classes, each with 6,000 training images and 1,000 test images. The 10 classes are: Top/T-Shirt, Trousers, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Boot. We will also fit the hierarchical correlated belief-matching introduced in Section 4.3.2 to compare this model to the correlated belief-matching model. The hiearchical model is also a natural choice for analyzing the MNIST data because while we are able to measure the similartity between classes, it is also possible to break up the classes into groups based on the type of clothing item, e.g. shoes, shirts, etc. The classifier architecture is made of 5 fully-connected layers with RELU activation functions in between. Batch normalization is also performed after the inputs and after the 4th layer. Figure 4.6 summarizes the architecture used for the MNIST image classification.

When training the model, we use the Adam optimizer with a batch size of 150 and a learning rate of 10^{-3}. We trained all models for 20 epochs. For the dropout model, only one dropout probability, c_p, was used for all layers.

For the correlated belief-matching framework, we pre-specify the prior class correlation matrix. We did this by computing the Wu-Palmer (WUP) word similarity (Wu and Palmer 1994) between the class labels using WordNet (Miller 1998) synsets. This metric measures the semantic similarity between two English words. Because all the class labels are clothing items, the WUP similarity is high between all the MNIST-Fashion class labels. We rescale the WUP similiarity by subtracting the minimum class label simliarity (0.56) and dividing by $1 - 0.56 = 0.44$. The resulting prior correlation matrix is displayed in Figure 4.7.

When using the hierarchical correlated belief-matching loss function introduced in Section 4.3.2, we need to specify a parition of the class labels as well as a hyperparameter, ν. We use $\nu = 100$ and we group the class labels based on the type of clothing. We have a top group (Top/T-Shirt, Pullover, Coat, Shirt) and a shoe group (sandal, sneaker, boot). All the other labels (trousers, dress, bag) are singleton groups.

We examine the results for settings of the hyperparameters that yielded comparable

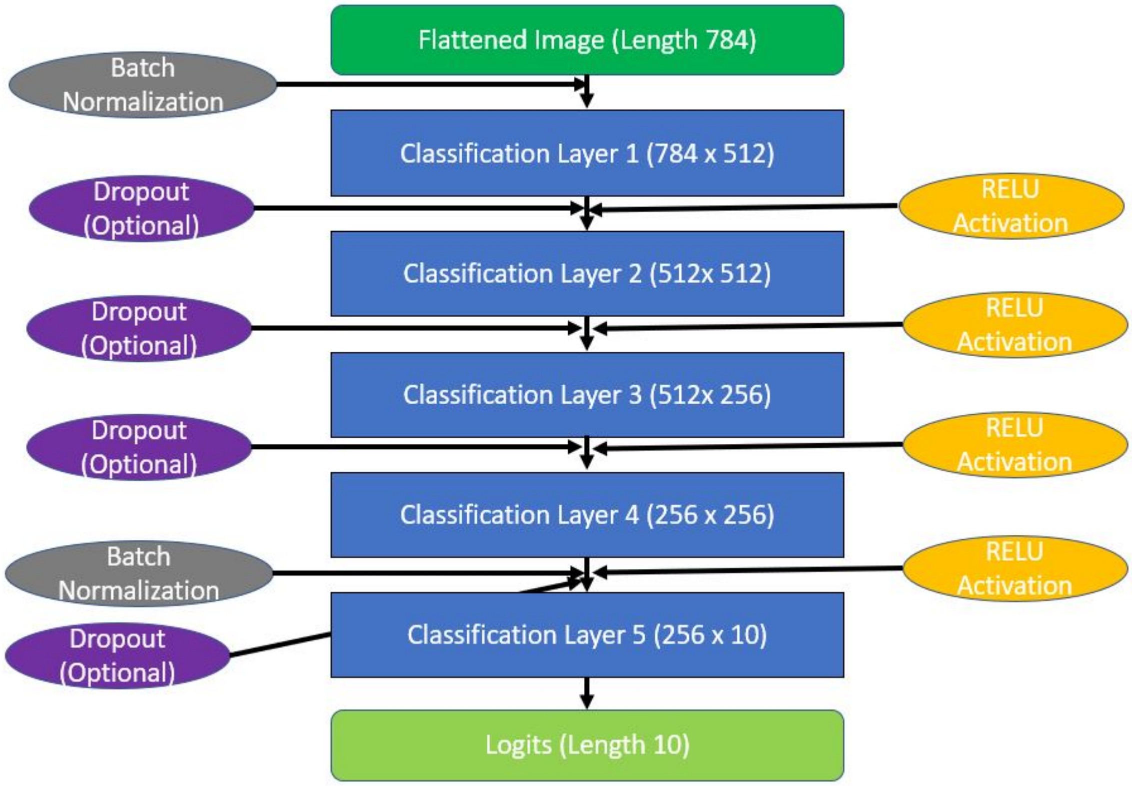


Figure 4.6: Image Classifier Architecture

	Top	Trousers	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Boot
Top	1	0.55	0.45	0.77	0.45	0.25	0.55	0.25	0.07	0.34
Trousers	0.55	1	0.68	0.55	0.68	0.25	0.77	0.25	0.07	0.34
Pullover	0.45	0.68	1	0.45	0.59	0.16	0.68	0.16	0.00	0.25
Dress	0.77	0.55	0.45	1	0.45	0.25	0.55	0.25	0.07	0.34
Coat	0.45	0.68	0.59	0.45	1	0.16	0.68	0.16	0.00	0.25
Sandal	0.25	0.25	0.16	0.25	0.16	1	0.25	0.75	0.07	0.59
Shirt	0.55	0.77	0.68	0.55	0.68	0.25	1	0.25	0.07	0.34
Sneaker	0.25	0.25	0.63	0.25	0.63	0.75	0.25	1	0.07	0.59
Bag	0.07	0.07	0.00	0.07	0.00	0.07	0.07	0.07	1	0.16
Boot	0.34	0.34	0.25	0.34	0.25	0.59	0.34	0.59	0.16	1

Figure 4.7: Prior Correlation Matrix for MNIST-Fashion Labels

Table 4.2: MNIST-Fashion Method Comparison

	Accuracy	ECE	CVG	WDTH	RA	Sampling
BM($\lambda = 0.0025$)	**90.4%**	0.031	90.9%	1.01	0.77	7.2 sec
CorrBM($\sigma_0 = 10, \psi = 0.1$)	90.1%	0.031	**95.3%**	1.20	**0.90**	8.1 sec
Drop($c_p = 0.3$)	90.1%	0.029	95.9%	1.22	0.82	48.8 sec
HierCorrBM($\sigma_0 = 1, \psi = 5.5$)	90.2%	0.03	93.2%	1.09	NA	**6.0 sec**

calibration results for all methods. The results are presented in Table 4.2. Additional results for other hyperparameter settings are available in Table D.2. We can see from the table that the original belief-matching framework is not able to produce good calibration and good coverage at the same time. The other three approaches do not have this problem although the hierarchical correlated belief-matching coverage is a bit lower than the nominal level. As with the small OSTI data example, the correlated belief-matching framework produces the highest ranking accuracy compared to dropout and original belief-matching. We do not compute the ranking accuracy for the hierarchical correlated belief-matching because in this scenario we assume we do not have a fully specified set of class similarities that we can use to compute the similarity ranking used in the RA metric. As in the small OSTI data set, generating posterior samples using dropout requires significantly more time than any of the belief matching frameworks, as can be seen from the last column in the table which reports the compute time for drawing 200 posterior samples. These times were obtained by running the model on a machine with a single 4-core CPU.

4.4.3 Full OSTI Text Classification

In this section, we use the hierarchical correlated belief-matching framework to analyze a larger subset of the OSTI dataset from Burke et al. (2021) that represents all classes from the original dataset. The original dataset consists of 198,654 scientific abstracts divided into 60 classes. We consider a subset of 52,756 documents where the number of documents per class is more balanced. The classes along with the number of train and test documents for each class are listed in Appendix C. The OSTI dataset is typical of the kind of data we might encounter when doing text classification for proliferation detection in a dual-use context. The texts cover a broad range of scientific topics including nuclear fuel cycle and fuel materials, nuclear/radiation physics, and nuclear/radiation chemistry that involve dual-use technology.

Table 4.3: OSTI Method Comparison

	Accuracy	ECE	CVG	WDTH	GH	Sampling
BM($\lambda = 0.01$)	68.3%	0.129	70.4%	1.09	0.120	2.47 min
HierCorrBM($\sigma_0 = 10, \psi = 0.5$)	67.3%	0.132	79.5%	4.61	**0.479**	**1.47 min**
Drop($c_p = 0.3$, $h_p = 0.3$, $a_p = 0.1$)	**68.7%**	0.131	**83.3%**	1.76	0.266	216.75 min

We do not have a predefined way to measure the similarity between the 60 classes making up the OSTI dataset and it is difficult to elicit similarities for all 1770 possible class pairs. Instead, we can use the hierarchical correlated belief-matching framework to estimate the posterior for the logits and for each of the $\mathbf{P}_0^g$. Using this approach only requires that we partition the classes into groups. Fortunately, a partitioning of the OSTI categories has already been done and is given in the OSTI subject categories booklet published on the OSTI website [1]. Appendix C gives the grouping for the classes used in our dataset based on the official OSTI class grouping.

For the OSTI dataset, we compare the hierarchical correlated belief-matching framework with regular belief-matching and dropout. Because we do not have a known $\mathbf{P}_0$ a priori, we cannot use the ranking accuracy metric to assess how well our model results agree with prior beliefs about class similarity. Instead, we compute the proportion of examples from the test set for which the top 3 most probable classes are all in the same group. We call this proportion group homogeneity (GH). We chose to look at the top 3 most probable classes because for larger numbers of top classes, the GH metric can get quite small, and the relative ordering of the methods by this metric does not depend on the choice of the number of top classes considered (see Appendix D.1). Our hypothesis is that our hierarchical correlated belief-matching approach should have a higher GH than the dropout and original belief-matching framework.

We use the same deep learning architecture given in Figure 4.1. We use a batch size of 16 and a learning rate of 10^{-5}. We train the model for 10 epochs using the Adam optimizer with 0.01 weight decay and 1879 warm-up steps.

Table 4.3 displays the results for different hyperparameter settings of the original belief-matching, hierarchical correlated belief-matching, and dropout frameworks so that the ECE values are comparable. Additional results are available in Appendix D.3.

As with previous experiments, the hierarchical correlated belief-matching model has a

[1] https://www.osti.gov/stip.old/system/files/Subject_Categories_Booklet.pdf

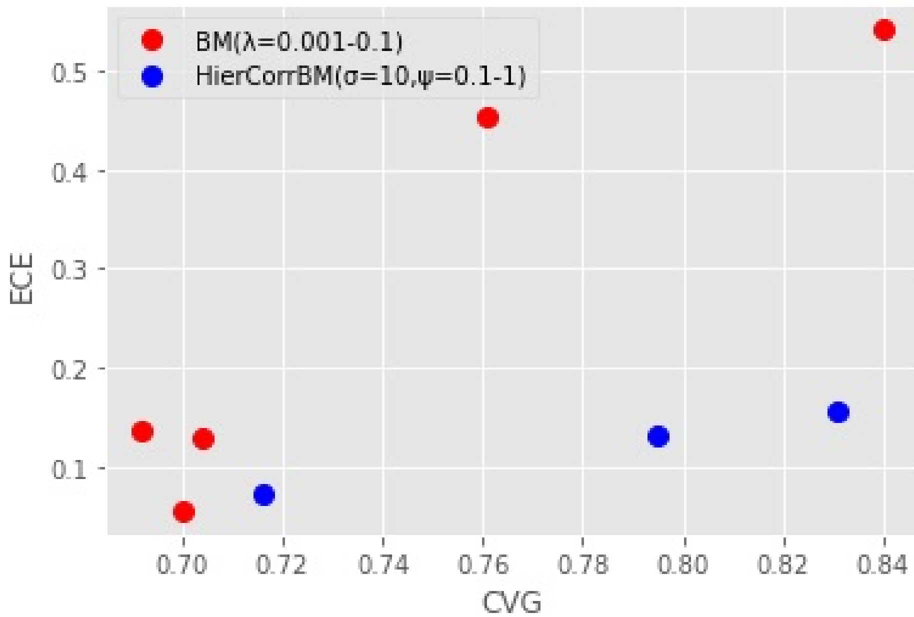


Figure 4.8: CVG vs. ECE for OSTI Dataset

much higher CVG value, despite having a slightly lower accuracy compared to the original belief-matching model. Figure 4.8 displays pairs of ECE and CVG for different hyperparameter settings of the original belief-matching and the hierarchical correlated belief-matching framework. As with the small OSTI data set, when using the original belief-matching model, ECE must increase substantially compared to the hierarchical correlated belief-matching model to attain high values of CVG.

As we expected, the GH is much higher for hierarchical correlated-belief matching, indicating that this method allows us to produce class probability results where the most probable classes are more likely to be in the same class grouping. As in previous experiments, the belief-matching approaches have much lower computational costs associated with drawing from the posterior, as can be seen from the last column of the table. These times were obtained by running the model in parallel on a machine with 16 CPU cores and 2 GPUs.

4.5 Conclusion and Future Work

In this chapter, we propose a novel approach for performing uncertainty quantification with large deep learning classification models, particularly natural language processing models like the BERT model. Our proposed approach provides efficient uncertainty quantification that is well-calibrated, has good coverage, and is understandable to human analysts. Having uncertainty quantification with these properties is particularly important for text classification that impacts downstream analysis about potential proliferation threats in dual-use contexts.

We propose two novel deep learning classification loss functions that extend the belief-matching framework proposed in Joo et al. (2020). These loss functions use the prior to better control the class probability similarity for classes that are known a priori to be similar and control the coverage of our credible set of classification labels without sacrificing good model calibration. We demonstrate the advantages of the proposed approaches on several classification data sets including the OSTI data set that contains several classes that encompass research activities with dual-use applications.

Our numerical experiments showed that while both original belief-matching and our proposed approaches are computationally efficient for posterior sampling compared to dropout, our approaches often allow for a better trade-off between ECE and CVG because of the two hyperparameters that can be tuned during training. Dropout often produces good uncertainty quantification, but posterior sampling can be prohibitively expensive for large architectures like those employed for text classification. In producing outputs that are easily understandable and agree with prior notions of class similarity, our proposed approach is substantially better than any of the competing approaches considered.

Uncertainty quantification for deep learning models is dependent on other areas of active statistical research, and future extensions of our work may be possible as improvements in related fields occur. For instance, functional variational inference (Sun et al. 2019) is a relatively new approach to Bayesian inference for neural networks that is specifically designed for modeling outputs of deep neural networks. Many current approaches that use functional variational inference focus on constructing functional priors for the outputs because these priors are easier to interpret. Using a similar approach for obtaining posterior and prior distributions for the logits may be an interesting extension of the present work.

In addition, our approach to modeling the prior and posterior logit correlation relied on unconstrained parameterizations of correlation matrices that do not allow for modeling structure in the correlation matrix. As new, structured, unconstrained parameterizations for correlation matrices are developed, future work should examine using these approaches for more flexible correlation structures that better capture what we know about the classes.

CHAPTER

5

SUMMARY OF CONTRIBUTIONS AND FUTURE WORK

In this dissertation, we propose statistical solutions to challenges associated with three problems related to counterproliferation and nonproliferation. These three problems address different nonproliferation concerns including nuclear counter-terrorism, arms control, and dual-use technology. These problems also explore a variety of important statistical themes including experimental design, surrogate modeling, expert elicitation, and uncertainty quantification.

Chapter 2 focuses on constructing an optimal configuration of detectors for localizing a radiation emitting-source in a 2D urban environment. We develop a framework for optimizing the mutual information design criterion for situations where it is computationally expensive to compute for arbitrary detector configurations. In particular, we construct a Gaussian process surrogate model of mutual information which can be efficiently optimized over the space of all possible detector configurations. We also develop an approach for constructing designs for this surrogate model that adapts to updates in the posterior for the source location. We use the mobile detector problem with simulated data for an urban environment in Washington, DC to demonstrate our approach. We show that our proposed approach yields localization results that are comparable to or better than competing methods for detector configuration construction.

We develop our methodology for 2D representations of urban environments, but our approach is usable with modifications in a 3D urban environment. Miles et al. (2021) have already worked on source localization in 3D urban environments, but their work focused on fitting surrogate models for the detector response model, not design criteria, and they do not address the optimal detector configuration problem in their work. To perform source localization in 3D, we must have an efficient 3D equivalent of the detector response model given in Equation (2.3). The primary cost associated with this model comes from computing the intersections between the buildings in the urban environment and the ray connecting the source to the detector. These intersections can be computed in parallel on a GPU using software specially designed for ray-tracing computations. We implement an efficient 3D detector response model (Hollis and Gavane 2021) and describe the implementation and its advantages in Appendix B.

Another challenge associated with 3D source localization is scalability. Because of the additional dimension, we need more detectors for each configuration to make localization effective. This, along with the additional dimension, increases the input dimension for our surrogate model. We also need larger designs for building accurate surrogate models of mutual information in this higher dimensional space. It is well known that while Gaussian processes are generally very accurate and have advantageous inference properties because of the Gaussian assumption, they do not scale well to high dimensional problems and large datasets because fitting and prediction with Gaussian processes requires inverting a matrix that scales with the data size and the input dimension. Thus, while the Gaussian process model is a good surrogate model in the 2D case, we may need to consider other, more efficient surrogate models such as multivariate adaptive regression splines (Friedman 1991) or computationally efficient adaptations of Gaussian processes (Quinonero-Candela and Rasmussen 2005; Banerjee et al. 2013b; Gu and Berger 2016) in the 3D scenario.

Finally, it is worth noting that our approach to the source localization problem has many simplifications, which should be relaxed in the future to make the approach robust to challenges in real world environments. We only consider a single, fixed-location source in our experiments. Other authors consider moving sources or multiple sources (Morelande et al. 2007; Vilim and Klann 2009; Chandy et al. 2008). We also assume that background radiation can be described as a normally distributed random variable with constant mean and variance over the whole urban environment. This is a simplistic view of background radiation and future work should incorporate more sophisticated models of background as in Michaud et al. (2021). The radiation transport model we use to describe the expected detector readings in Equation (2.3) also relies on several physical simplifications such as only modeling the radiation counts due to photons that have not collided with other objects in the environment. This assumption makes the model more tractable, but it may not always be appropriate.

In Chapter 3, we create models for monitoring activities for arms control applications. We adapt the standard Bayesian hidden Markov model (HMM) to model activity duration and utilize subject matter expertise. Specifically, we propose a reparameterization of the time-dependent (continuous time) transition probabilities as simple, closed-form functions of the mean activity duration. Our reparameterization is applicable to any HMM that does not have states that repeat. Because activity duration is a parameter of the model, we are able to straightforwardly answer time-based process monitoring questions that are of interest to decision-makers.

Because data is scarce in many arms-control applications, we also propose a framework for incorporating subject matter expertise about activity duration, observables, and noise to ensure we are making the most of all available information before performing inference. We derive a new Metropolis-within-Gibbs algorithm to sample from the posterior distribution of our proposed model. Through numerical experiments, we demonstrate that the inference depends not only on the observations symbols we see, but also the order in which we see them. This suggests that there is important information in the observation order. From the Dry Aluvium Geology (DAG) case study, we learn that it is important to incorporate subject matter expertise about noise through the LDA prior in order to obtain appropriate uncertainty quantification in high-noise scenarios.

Whereas our proposed approach is an initial framework for analyzing data emitted by hidden arms development processes, there are several simplifications in our proposed approach that must be relaxed to make the framework useful for more general problems. While the assumption of exponentially distributed durations does allow for the simple derivation of analytic forms for the time-dependent transition probabilities and distribution functions for T_{start} and T_{end}, the form of the exponential distribution is very restrictive and may not agree with our experts' intuition about the actual distribution of activity duration. A potential approach for extending to more general duration distributions is using phase type distributions, which are approximations of general distributions, using a series of exponential distributions. Isensee et al. (2006) employed this approach for hidden non-Markov models. An advantage of this approach is that while the activities can have non-exponentially distributed durations, we can interpret the activities as being composed of several sub-activities with exponentially distributed durations. If we treat each of these sub-activities as a process state, our framework would not need to change substantially to accommodate more general activity durations, but the process state space would be significantly larger, which would make inference more computationally costly. It could also be challenging to interpret the inference if several hidden states correspond to a single hidden process activity.

We also assume that the processes we monitor are simple, sequential processes in which only one activity is underway at any given time. For many real-world processes, several activities may be running in parallel. These kinds of processes were part of the inspiration for stochastic activity networks as extensions of Petri nets. The minimum of an exponentially distributed random variable is also exponentially distributed. Thus, if we treat all possible combinations of parallel activities as process model states, we would still have an HMM with exponentially distributed duration times. Without the exponentially distributed duration assumption, however, inference for parallel activities can be very complicated. Buchholz (2012) discuss some of the intricacies and challenges of working with hidden state models with parallel activities that may not have exponentially distributed duration times.

In our proposed approach, we assume only one stream of observed data. In many arms-control monitoring scenarios, we have access to multiple data streams including imagery data, seismic data, power usage data, effluent readings, etc. Future approaches should incorporate these other streams of data. Previous work in the HMM literature has discussed the incorporation of multiple data streams (Bulla et al. 2012; Martino et al. 2020; DeRuiter et al. 2017), and several authors have found performing inference with multiple data streams can be particularly helpful in noisy environments as the combined signal from various noisy data streams is stronger than the signal in any one data stream (Pan et al. 2004; Lin et al. 2013). Thus, the incorporation of other data streams would be particularly beneficial in our case where noisy data is a challenge.

Finally, our approach assumes that we know what process we are monitoring with certainty. In many real-world arms-control monitoring scenarios, we are uncertain about the process that may have generated the data. Future work might investigate using Bayesian model selection techniques that can be used to determine which process model from a set of candidate process models best fits the data. Another approach might be using a mixture of HMMs where each component represents a candidate process. In this setting, posterior inference on the mixture weights could be used as an indication of what process likely produced the data.

In Chapter 4, we propose a novel framework for performing uncertainty quantification with deep learning classification models. The approach is especially useful for extremely large architectures like those used for natural language processing and is based on the belief-matching approach introduced in Joo et al. (2020). This approach allows for efficient uncertainty quantification because an arbitrary number of posterior samples can be generated using only one forward pass through the model. This is less computationally demanding than weight-space methods, dropout methods, or ensembling methods that all require multiple forward passes and/or multiple model fits to generate a posterior sample.

We expand on the belief-matching framework by introducing two new loss functions that fit a variational posterior distribution for the logits before they are transformed into class probabilities via the softmax function. We introduce one loss function that assumes complete knowledge of the similarity structure across classes including a measure of similarity for each pair of classes. We also introduce a loss function that only requires that we be able to partition the set of classes into groups based on similarity. Because the logits are unconstrained, we can use a Gaussian variational posterior, which we show allows us to exercise greater control over the posterior distribution and achieve better uncertainty quantification results than the original belief-matching approach.

In particular, our approach permits a better trade-off between calibration and coverage than original belief-matching and is more efficient than dropout. In producing outputs that are easily understandable and agree with prior notions of class similarity, our proposed approach is substantially better than any of the competing approaches considered. Efficient, well-calibrated, understandable uncertainty quantification is especially important in the non-proliferation context where classification results will feed into down stream analyses.

A natural extension of our approach would be to consider alternative variational inference frameworks. In particular the functional variational inference framework (Sun et al. 2019) is specifically designed for constructing priors over model outputs, such as logits or class probabilities using a functional prior. This approach, however, is challenging to implement and is not designed for constructing functional posteriors on the outputs, requiring us to use the more expensive and more difficult to control weight-space posteriors. As functional variational inference for deep neural networks evolves, future extensions of our approach should consider using the functional variational inference to model the logits or class probabilities.

One of the limitations of our approach is our inability to impose structure on the correlation matrices of the variational posterior distribution for the logits or the prior group correlation matrices ($\mathbf{P}_0^g$). To simplify the learning of these parameters, we use unconstrained parameterizations of these matrices, like the hyper-spherical parameterization. Unfortunately, these approaches do not permit us to impose structure on the correlation matrix. Developing unconstrained parameterizations of structured correlation matrices is an area of active research in statistics (Archakov and Hansen 2021; Tsay and Pourahmadi 2017), and as approaches are developed, they should be incorporated with the correlated belief-matching and hierarchical correlated belief-matching approaches that we propose.

In future work, it would be useful to also consider the multi-label text classification problem. In multi-label text classification, we predict a collection of multiple labels for each text rather than only a single label. This kind of approach is necessary when the texts we are dealing with address multiple label categories simultaneously, as can often be the case when working with scientific articles in dual-use domains. Multi-label text classification methods often make use of label similarity information (Cai et al. 2020; Huang et al. 2022), so our proposed method, which exploits prior label similarity, would likely be most useful in this context.

Another consideration for future work, specific to natural language processing models, is the incorporation of information from different, but similar subject-matter domains into the model prior distribution. For example, the SciBERT model (Beltagy et al. 2019) or the NukeBERT (Jain et al. 2020) are BERT models pre-trained on scientific articles similar to the NukeLM model. It would be informative to investigate the possibility of combining the SciBERT, NukeBERT, and NukeLM pre-trained models to take advantage of the information in all these model frameworks when performing classification and uncertainty quantification. To do this, we might use a framework similar to the one proposed in Rothfuss et al. (2021) or Chandra and Kapoor (2020) to transfer the knowledge from the SciBERT and NukeBERT model to the NukeLM model through a Bayesian framework.

REFERENCES

Anderson, J., Devine, T., and Gibbons, R. D. (2014). Nonproliferation and counterproliferation. *Oxford Bibliographies Online Datasets*.

Archakov, I. and Hansen, P. R. (2021). A New Parametrization of Correlation Matrices. *Econometrica*, 89(4):1699–1715.

Arrowood, J. A. and Clements, M. A. (2002). Using Observation Uncertainty in HMM Decoding. In *Seventh International Conference on Spoken Language Processing*.

Baidoo-Williams, H. E., Dasgupta, S., Mudumbai, R., and Bai, E. (2013). On the Gradient Descent Localization of Radioactive Sources. *IEEE Signal Processing Letters*, 20(11):1046 – 1049.

Banerjee, A., Dunson, D. B., and Tokdar, S. T. (2013a). Efficient Gaussian Process Regression for Large Datasets. *Biometrika*, 100(1):75–89.

Banerjee, A., Dunson, D. B., and Tokdar, S. T. (2013b). Efficient Gaussian Process Regression for Large Datasets. *Biometrika*, 100(1):75–89.

Beaumont, P., Evans, N., Huth, M., and Plant, T. (2015). Confidence Analysis for Nuclear Arms Control: SMT Abstractions of Bayesian Belief Networks. In *European Symposium on Research in Computer Security*, pages 521–540. Springer.

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. (2018). Mutual Information Neural Estimation. In *35th International Conference on Machine Learning, ICML 2018*, pages 531–540, Stockholm, Sweden. Proceedings of Machine Learning Research.

Beltagy, I., Lo, K., and Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. *arXiv preprint arXiv:1903.10676*.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight Uncertainty in Neural Network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR.

Bonner, E., Burr, T., Guzzardo, T., Krieger, T., Norman, C., Zhao, K., Beddingfield, D., Geist, W., Laughter, M., and Lee, T. (2016). Ensuring the Effectiveness of Safeguards through Comprehensive Uncertainty Quantification. *J. Nucl. Mater. Manage.*, 44(2):53–61.

Briesmeister, J. F. (2000). MCNPTM – A General Monte Carlo N-Particle Transport Code.

Buchholz, R. (2012). *Conversive Hidden Non-Markovian Models.* PhD thesis, Magdeburg, Universität, Diss., 2012.

Bulla, J., Lagona, F., Maruotti, A., and Picone, M. (2012). A multivariate hidden markov model for the identification of sea regimes from incomplete skewed and circular time series. *Journal of Agricultural, Biological, and Environmental Statistics*, 17(4):544–567.

Burke, L., Pazdernik, K., Fortin, D., Wilson, B., Goychayev, R., and Mattingly, J. (2021). NukeLM: Pre-Trained and Fine-Tuned Language Models for the Nuclear and Energy Domains. *arXiv preprint arXiv:2105.12192*.

Cai, L., Song, Y., Liu, T., and Zhang, K. (2020). A Hybrid BERT Model that Incorporates Label Semantics via Adjustive Attention for Multi-Label Text Classification. *Ieee Access*, 8:152183–152192.

Cappé, O., Moulines, E., and Ryden, T. (2005). *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Chandra, R. and Kapoor, A. (2020). Bayesian Neural Multi-Source Transfer Learning. *Neurocomputing*, 378:54–64.

Chandy, M., Pilotto, C., and McLean, R. (2008). Networked Sensing Systems for Detecting People Carrying Radioactive Material. In *Proceedings of INSS 2008 - 5th International Conference on Networked Sensing Systems*, pages 148–155, Kanazawa, Japan. IEEE.

Chen, W., Shen, Y., Jin, H., and Wang, W. (2018). A Variational Dirichlet Framework for Out-of-Distribution Detection. *arXiv preprint arXiv:1811.07308*.

Chib, S. (1996). Calculating Posterior Distributions and Modal Estimates in Markov Mixture Models. *Journal of Econometrics*, 75(1):79–97.

Chin, J. C., Yau, D. K., Rao, N. S., Yang, Y., Ma, C. Y., and Shankar, M. (2008). Accurate Localization of Low-Level Radioactive Source Under Noise and Measurement Errors. In *SenSys'08 - Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems*, pages 183–196, Raleigh, NC. ACM.

Cholette, M. E. and Djurdjanovic, D. (2014). Degradation Modeling and Monitoring of Machines Using Operation-Specific Hidden Markov Models. *IIE Transactions*, 46(10):1107–1123.

Ciardo, G., German, R., and Lindemann, C. (1994). A Characterization of the Stochastic Process Underlying a Stochastic Petri Net. *IEEE Transactions on Software Engineering*, 20(7):506–515.

Cook, J. A., Smith, R. C., Hite, J. M., Stefanescu, R., and Mattingly, J. (2019). Application and Evaluation of Surrogate Models for Radiation Source Search. *Algorithms*, 12(12):269.

Cox, J. and Partensky, M. B. (2007). Spatial Localization Problem and the Circle of Apollonius. *arXiv*.

Cui, X., Afify, M., Gao, Y., and Zhou, B. (2013). Stereo Hidden Markov Modeling for Noise Robust Speech Recognition. *Computer Speech & Language*, 27(2):407–419.

Darling, W. M. (2011). A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 642–647.

Daunizeau, J. (2017). Semi-Analytical Approximations to Statistical Moments of Sigmoid and Softmax Mappings of Normal Variables. *arXiv preprint arXiv:1703.00091*.

Daxberger, E., Nalisnick, E., Allingham, J. U., Antorán, J., and Hernández-Lobato, J. M. (2021). Bayesian Deep Learning via Subnetwork Inference. In *International Conference on Machine Learning*, pages 2510–2521. PMLR.

De Angelis, L. and Paas, L. J. (2013). A Dynamic Analysis of Stock Markets Using a Hidden Markov Model. *Journal of Applied Statistics*, 40(8):1682–1700.

Denison, D. G., Mallick, B. K., and Smith, A. F. (1998). Bayesian MARS. *Statistics and Computing*, 8:337–346.

DeRuiter, S. L., Langrock, R., Skirbutas, T., Goldbogen, J. A., Calambokidis, J., Friedlaender, A. S., and Southall, B. L. (2017). A multivariate mixed hidden markov model for blue whale behaviour and responses to sound exposure. *The Annals of Applied Statistics*, 11(1):362–392.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Dickey, J., Jiang, J., Kadane, J., of New York at Albany, S. U., and of America, U. S. (1983). Bayesian Methods for Multinomial Sampling with Missing Data using Multiple Hypergeometric Functions. *State University of New York at Albany Department of Mathematics and Statistics: Research Report*, 6.

Du, Y., Wu, T., and Makis, V. (2017). Parameter Estimation and Remaining Useful Life Prediction of Lubricating Oil with HMM. *Wear*, 376:1227–1233.

El Anigri, S., Himmi, M. M., and Mahmoudi, A. (2021). How BERT's Dropout Fine-Tuning Affects Text Classification? In *International Conference on Business Intelligence*, pages 130–139. Springer.

Francom, D., Sansó, B., Kupresanin, A., and Johannesson, G. (2018). Sensitivity Analysis and Emulation for Functional Data using Bayesian Adaptive Splines. *Statistica Sinica*, 28:791–816.

Francom, D. and Sansó, B. (2020). BASS: An R Package for Fitting and Performing Sensitivity Analysis of Bayesian Adaptive Spline Surfaces. *Journal of Statistical Software*, 94(8):1–36.

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–67.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Gao, S., Ver Steeg, G., and Galstyan, A. (2015a). Efficient Estimation of Mutual Information for Strongly Dependent Variables. In *Artificial Intelligence and Statistics*, pages 277–286, San Diego, CA.

Gao, S., Ver Steeg, G., and Galstyan, A. (2015b). Estimating Mutual Information by Local Gaussian Approximation. In *Uncertainty in Artificial Intelligence - Proceedings of the 31st Conference, UAI 2015*, pages 278–287, Amsterdam, Netherlands. AUAI Press.

Garcia, H. E., Burr, T. L., Coles, G. A., Edmunds, T. A., Garrett, A. J., Gorensek, M. B., Hamm, L. L., Krebs, J. F., Kress, R. L., Lamberti, V. E., et al. (2012). Integration of Facility Modeling Capabilities for Nuclear Nonproliferation Analysis. *Progress in Nuclear Energy*, 54(1):96–111.

Garcia, H. E., Simpson, M. F., Lin, W.-C., Carlson, R. B., and Yoo, T.-S. (2017). Application of Process Monitoring to Anomaly Detection in Nuclear Material Processing Systems via System-Centric Event Interpretation of Data from Multiple Sensors of Varying Reliability. *Annals of Nuclear Energy*, 103:60–73.

Gastelum, Z. N., Goldblum, B., Shead, T., and Luttman, A. (2019). Integrating Physical and Informational Sensing to Support Nonproliferation Assessments of Nuclear-Related Facilities. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Gauld, I., Croft, S., Pigni, M., Nicholson, A., Williams, M., Mozin, V., Mattoon, C., Vogt, R., Caggiano, J., and Swinhoe, M. (2016). Systematic Approach to Nuclear Data Uncertainty Quantification for Nuclear Security Applications. In *Advances in Nuclear Nonproliferation Technology & Policy Conference: Bridging the Gaps in Nuclear Nonproliferation*.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al. (2021). A Survey of Uncertainty in Deep Neural Networks. *arXiv preprint arXiv:2107.03342*.

Ghosh, S., Yao, J., and Doshi-Velez, F. (2018). Structured Variational Learning of Bayesian Neural Networks with Horseshoe Priors. In *International Conference on Machine Learning*, pages 1744–1753. PMLR.

Gillen, D. S. (2014). A Visual Analytics Approach to Structured Data Analysis to Enhance Nonproliferation and Arms Control Verification Activities. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States).

Gu, M. and Berger, J. O. (2016). Parallel Partial Gaussian Process Emulation for Computer Models with Massive Output. *The Annals of Applied Statistics*, 10(3):1317–1347.

Gunatilaka, A., Ristic, B., and Gailis, R. (2007). On Localisation of a Radiological Point Source. In *Conference Proceedings of 2007 Information, Decision and Control, IDC*, pages 236 – 241, New Orleans, LA. IEEE.

Gustafsson, F. K., Danelljan, M., and Schon, T. B. (2020). Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 318–319.

Haario, H., Laine, M., Mira, A., and Saksman, E. (2006). DRAM: Efficient Adaptive MCMC. *Statistics and Computing*, 16:339–354.

Han, Z. H., Zhang, K. S., Song, W. P., and Qiao, Z. D. (2010). Optimization of Active Flow Control Over an Airfoil Using a Surrogate-Management Framework. *Journal of Aircraft*, 47(2):603–612.

Hardy, S. and Robillard, P. N. (2004). Modeling and Simulation of Molecular Biology Systems using Petri Nets: Modeling Goals of Various Approaches. *Journal of bioinformatics and computational biology*, 2(04):619–637.

Hite, J., Mattingly, J., Archer, D., Willis, M., Rowe, A., Bray, K., Carter, J., and Ghawaly, J. (2019). Localization of a Radioactive Source in an Urban Environment using Bayesian Metropolis Methods. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 915:82–93.

Hobbhahn, M., Kristiadi, A., and Hennig, P. (2022). Fast predictive uncertainty for classification with bayesian deep networks. In *Uncertainty in Artificial Intelligence*, pages 822–832. PMLR.

Hoffmann, L. and Elster, C. (2021). Deep Ensembles from a Bayesian Perspective. *arXiv preprint arXiv:2105.13283*.

Hollis, A. and Gavane, A. (2021). "Efficient Ray-Tracing for Urban Radiation Source Localization". High Performance Graphics 2021.

Horne, S. M., Thoreson, G. G., Theisen, L. A., Mitchell, D. J., Harding, L., and Amai, W. A. (2014). GADRAS-DRF 18.5 User's Manual. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States).

Hruby, J. (2020). Challenges and Opportunities in Scientific and Technological Support for Monitoring in the Non-proliferation Regime. *International Cooperation for Enhancing Nuclear Safety, Security, Safeguards and Non-proliferation*, pages 83–89.

Huang, J., Vong, C.-M., Chen, C. P., and Zhou, Y. (2022). Accurate and Efficient Large-Scale Multi-Label Learning With Reduced Feature Broad Learning System Using Label Correlation. *IEEE Transactions on Neural Networks and Learning Systems*.

Isensee, C., Wickborn, F., and Horton, G. (2006). Training Hidden Non-Markov Models. In *Proceedings of 13th International Conference on Analytical and Stochastic Modelling Techniques and Applications, Bonn, Germany*.

Jagadeesh, G. R. and Srikanthan, T. (2015). Probabilistic Map Matching of Sparse and Noisy Smartphone Location Data. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 812–817. IEEE.

Jain, A., Meenachi, D. N., and Venkatraman, D. B. (2020). NukeBERT: A Pre-Trained Language Model for Low Resource Nuclear Domain. *arXiv preprint arXiv:2003.13821*.

Jarman, K. D., Miller, E. A., Wittman, R. S., and Gesh, C. J. (2011). Bayesian Radiation Source Localization. *Nuclear Technology*, 175(1):326–334.

Javadi Mottaghi, F. and Abdollahi Azgomi, M. (2009). Workflow Modeling Using Stochastic Activity Networks. In *International Conference on Information Systems, Technology and Management*, pages 220–231. Springer.

Johnson, M. T. (2005). Capacity and Complexity of HMM Duration Modeling Techniques. *IEEE Signal Processing Letters*, 12(5):407–410.

Joo, T., Chung, U., and Seo, M.-G. (2020). Being Bayesian about Categorical Probability. In *International Conference on Machine Learning*, pages 4950–4961. PMLR.

Kadri, T., Smaili, K., and Kadry, S. (2015). Markov Modeling for Reliability Analysis using Hypoexponential Distribution. In *Numerical Methods for Reliability and Safety Assessment*, pages 599–620. Springer.

Kay, T. L. and Kajiya, J. T. (1986). Ray Tracing Complex Scenes. *ACM SIGGRAPH computer graphics*, 20(4):269–278.

Kelly, D., Dillingham, M., Hudson, A., and Wiesner, K. (2012). A New Method for Inferring Hidden Markov Models from Noisy Time Sequences. *PloS One*, 7(1):e29703.

Kendall, A. and Gal, Y. (2017). What Uncertainties do We Need in Bayesian Deep Learning for Computer Vision? *Advances in Neural Information Processing Systems*, 30.

Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*.

Knoll, G. G. (2010). *Radiation Detection and Measurement (4th Edition)*. John Wiley and Sons, Inc., Hoboken, NJ, 4th edition.

Kompa, B., Snoek, J., and Beam, A. L. (2021). Empirical Frequentist Coverage of Deep Learning Uncertainty Quantification Procedures. *Entropy*, 23(12):1608.

Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating Mutual Information. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 69.

Kuzmin, Y. P. (1994). Ray Traversal of Spatial Structures. In *Computer Graphics Forum*, volume 13, pages 223–227. Wiley Online Library.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems*, 30.

Langrock, R. and Zucchini, W. (2011). Hidden Markov Models with Arbitrary State Dwell-Time Distributions. *Computational Statistics & Data Analysis*, 55(1):715–724.

Lee, S., Song, W., and Yang, J.-S. (2021). Nuclear Fuel Cycle–Related R&D Classification for Implementing the IAEA's Additional Protocol. *Progress in Nuclear Energy*, 139:103884.

Lin, J.-C., Wu, C.-H., and Wei, W.-L. (2013). A probabilistic fusion strategy for audiovisual emotion recognition of sparse and noisy data. In *2013 1st International Conference on Orange Technologies (ICOT)*, pages 278–281. IEEE.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. (2020). Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512.

Liu, Y.-Y., Li, S., Li, F., Song, L., and Rehg, J. M. (2015). Efficient Learning of Continuous-Time Hidden Markov Models for Disease Progression. *Advances in Neural Information Processing Systems*, 28:3599.

Louizos, C. and Welling, M. (2016). Structured and Efficient Variational Deep Learning with Matrix Gaussian Posteriors. In *International Conference on Machine Learning*, pages 1708–1716. PMLR.

Ludvigsen, H. and Elster, A. C. (2010). Real-Time Ray Tracing Using Nvidia OptiX. In *Eurographics (Short Papers)*, pages 65–68.

Malinin, A. and Gales, M. (2018). Predictive Uncertainty Estimation via Prior Networks. *Advances in Neural Information Processing Systems*, 31.

Marsan, M. A. (1988). Stochastic Petri Nets: An Elementary Introduction. In *European Workshop on Applications and Theory in Petri Nets*, pages 1–29. Springer.

Martino, A., Guatteri, G., and Paganoni, A. M. (2020). Multivariate Hidden Markov Models for Disease Progression. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 13(5):499–507.

Maza, S. (2014). Stochastic Activity Networks for Performance Evaluation of Fault-Tolerant Systems. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 228(3):243–253.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):239–245.

Michaud, I. J. (2019). *Simulation-Based Bayesian Experimental Design using Mutual Information*. PhD thesis, North Carolina State University.

Michaud, I. J., Schmidt, K., Smith, R. C., and Mattingly, J. (2021). A Hierarchical Bayesian Model for Background Variation in Radiation Source :ocalization. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1002:165288.

Miles, P. (2019). pymcmcstat: A Python Package for Bayesian Inference Using Delayed Rejection Adaptive Metropolis. *Journal of Open Source Software*, 4(38).

Miles, P. R., Cook, J. A., Angers, Z. V., Swenson, C. J., Kiedrowski, B. C., Mattingly, J., and Smith, R. C. (2021). Radiation Source Localization using Surrogate Models Constructed from 3-D Monte Carlo Transport Physics Simulations. *Nuclear Technology*, 207(1):37–53.

Miller, G. A. (1998). *WordNet: An Electronic Lexical Database*. MIT press.

Morelande, M., Ristic, B., and Gunatilaka, A. (2007). Detection and Parameter Estimation of Multiple Radioactive Sources. In *FUSION 2007 - 2007 10th International Conference on Information Fusion*, Quebec City, Canada. IEEE.

Neal, R. M. (2012). *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media.

Negi, P. S., Mahoor, M., et al. (2018). Leveraging Class Similarity to Improve Deep Neural Network Robustness. *arXiv preprint arXiv:1812.09744*.

Neto, J. P. (2013). Radial Basis Functions.

O'Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., Oakley, J. E., and Rakow, T. (2006). *Uncertain Judgements: Eliciting Experts' Probabilities*. John Wiley & Sons.

Olmo, A., Sengupta, S., and Kambhampati, S. (2020). Not all Failure Modes are Created Equal: Training Deep Neural Networks for Explicable (Mis)classification. *arXiv preprint arXiv:2006.14841*.

Østergård, T., Jensen, R. L., and Maagaard, S. E. (2018). A Comparison of Six Metamodeling Techniques Applied to Building Performance Simulations. *Applied Energy*, 211:89–103.

Pan, H., Levinson, S. E., Huang, T. S., and Liang, Z.-P. (2004). A fused hidden markov model with application to bimodal speech processing. *IEEE Transactions on Signal Processing*, 52(3):573–581.

Pandey, S., Nepal, S., and Chen, S. (2011). A Test-Bed for the Evaluation of Business Process Prediction Techniques. In *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 382–391. IEEE.

Penny, R. D., Crowley, T. M., Gardner, B. M., Mandell, M. J., Guo, Y., Haas, E. B., Knize, D. J., Kuharski, R. A., Ranta, D., Shyffer, R., Labov, S., Nelson, K., Seilhan, B., and Valentine, J. D. (2015). Improved Radiological/Nuclear Source Localization in Variable NORM Background: An MLEM Approach with Segmentation Data. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 784:319–325.

Peterson, J. L. (1977). Petri Nets. *ACM Computing Surveys (CSUR)*, 9(3):223–252.

Pinheiro, J. C. and Bates, D. M. (1996). Unconstrained Parametrizations for Variance-Covariance Matrices. *Statistics and Computing*, 6(3):289–296.

Posch, K. and Pilz, J. (2020). Correlated Parameters to Accurately Measure Uncertainty in Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(3):1037–1051.

Punzo, A. and Maruotti, A. (2016). Clustering Multivariate Longitudinal Observations: The Contaminated Gaussian Hidden Markov Model. *Journal of Computational and Graphical Statistics*, 25(4):1097–1098.

Qi, H. and Sun, D. (2006). A Quadratically Convergent Newton Method for Computing the Nearest Correlation Matrix. *SIAM journal on matrix analysis and applications*, 28(2):360–385.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Kevin Tucker, P. (2005). Surrogate-Based Analysis and Optimization. *Progress in Aerospace Sciences*, 41(1):1–28.

Quinonero-Candela, J. and Rasmussen, C. E. (2005). A Unifying View of Sparse Approximate Gaussian Process Regression. *The Journal of Machine Learning Research*, 6:1939–1959.

Rabiner, L. and Juang, B. (1986). An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16.

Rahaman, R. et al. (2021). Uncertainty Quantification and Deep Ensembles. *Advances in Neural Information Processing Systems*, 34:20063–20075.

Rahmawati, D., Sarno, R., Fatichah, C., and Sunaryono, D. (2017). Fraud Detection on Event Log of Bank Financial Credit Business Process using Hidden Markov Model Algorithm. In *2017 3rd International Conference on Science in Information Technology (ICSITech)*, pages 35–40. IEEE.

Rao, N. S., Shankar, M., Srivathsan, S., Iyengar, S. S., Chin, J. C., Yau, D. K., Yang, Y., and Hou, J. C. (2008). Identification of Low-Level Point Radiation Sources Using a Sensor Network. In *Proceedings - 2008 International Conference on Information Processing in Sensor Networks, IPSN 2008*, pages 493–504, St Louis, MO. IEEE.

Rathore, K. and Chirayath, S. S. (2019). Investigation on Iran's Nuclear Weapon Latency Time in Case of Non-Compliance with JCPOA. *Transactions*, 120(1):629–632.

Ritter, H., Botev, A., and Barber, D. (2018). A Scalable Laplace Approximation for Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning.

Ross, S. M. (2014). *Introduction to Probability Models*. Academic press.

Rothfuss, J., Fortuin, V., Josifoski, M., and Krause, A. (2021). PACOH: Bayes-Optimal Meta-Learning with PAC-Guarantees. In *International Conference on Machine Learning*, pages 9116–9126. PMLR.

Roustant, O., Ginsbourger, D., and Deville, Y. (2012). DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *Journal of Statistical Software*, 51.

Runkle, R. C. (2011). Neutron Sensors and Their Role in Nuclear Nonproliferation. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 652(1):37–40.

Sanders, W. H. and Meyer, J. F. (2000). Stochastic Activity Networks: Formal Definitions and Concepts. In *School Organized by the European Educational Forum*, pages 315–343. Springer.

Sankararaman, K. A., Wang, S., and Fang, H. (2022). BayesFormer: Transformer with Uncertainty Estimation. *arXiv preprint arXiv:2206.00826*.

Santner, T. J., Williams, B. J., and Notz, W. I. (2018). *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. Springer New York, New York, NY, 2nd edition.

Satkowiak, L. (2014). Technical Solutions to Nonproliferation Challenges. In *AIP Conference Proceedings*, volume 1596, pages 180–193. American Institute of Physics.

Schmidt, K. (2016). *Uncertainty Quantification for Mixed-Effects Models with Applications in Nuclear Engineering*. PhD thesis, North Carolina State University.

Scott, S. L. (2002). Bayesian Methods for Hidden Markov Models: Recursive Computing in the 21st Century. *Journal of the American Statistical Association*, 97(457):337–351.

Scrucca, L. (2013). GA: A Package for Genetic Algorithms in R. *Journal of Statistical Software*, 53.

Sellers, G. and Lukac, R. (2009). Computer Graphics Using Raytracing. In *Handbook of Multimedia for Digital Entertainment and Arts*, pages 529–549. Springer.

Sensoy, M., Kaplan, L., and Kandemir, M. (2018). Evidential Deep Learning to Quantify Classification Uncertainty. *Advances in Neural Information Processing Systems*, 31.

Ştefănescu, R., Hite, J., Cook, J., Smith, R. C., and Mattingly, J. (2019). Surrogate-Based Robust Design for a Non-Smooth Radiation Source Detection Problem. *Algorithms*, 12(6):113.

Stein, M. (1987). Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*, 29(2):143–151.

Steinbrener, J., Posch, K., and Pilz, J. (2020). Measuring the Uncertainty of Predictions in Deep Neural Networks with Variational Inference. *Sensors*, 20(21):6011.

Stracuzzi, D. J., Darling, M. C., Chen, M. G., and Peterson, M. G. (2018). Data-driven Uncertainty Quantification for Multisensor Analytics. In *Ground-Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*, volume 10635, pages 155–167. SPIE.

Sukkar, R., Katz, E., Zhang, Y., Raunig, D., and Wyman, B. T. (2012). Disease Progression Modeling using Hidden Markov Models. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2845–2848. IEEE.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). Functional Variational Bayesian Neural Networks. *arXiv preprint arXiv:1903.05779*.

Sunilkumar, P. and Shaji, A. P. (2019). A survey on semantic similarity. In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pages 1–8. IEEE.

Swinney, M. W., Peplow, D. E., Nicholson, A. D., and Patton, B. W. (2016). NORM Concentration Determination in Common Materials in an Urban Environment. *Transactions of the American Nuclear Society*, 114:635–638.

Tong, X., Wang, R., Cao, X., and Ren, W. (2019). Semantic Correlations Loss: Improving Model Interpretability for Multi-Class Classification. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5070–5079. IEEE.

Tsay, R. S. and Pourahmadi, M. (2017). Modelling Structured Correlation Matrices. *Biometrika*, 104(1):237–242.

Unwin, S. D. and Fecht, B. A. (2009). Consumer Choice and Dempster-Shafer Models of Threat Prioritization for Emerging Dual-Use Technologies: Their Application to Synthetic Biology. *Defense & Security Analysis*, 25(1):37–52.

Van Der Aalst, W. M. (2015). Business Process Management as the "Killer App" for Petri Nets. *Software & Systems Modeling*, 14(2):685–691.

Vilim, R. and Klann, R. T. (2009). RadTarc: A System for Detecting, Localizing, and Tracking Radioactive Sources in Real Time. *Nuclear Technology*, 168(1):61–73.

Volovoi, V. (2004). Modeling of System Reliability Petri Nets with Aging Tokens. *Reliability Engineering & System Safety*, 84(2):149–161.

White, A. M., Gastelum, Z. N., and Whitney, P. D. (2014). Developing a Validation Methodology for Expert-Informed Bayesian Network Models Supporting Nuclear Nonproliferation Analysis. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States).

Wilson, A. G. (2020). The Case for Bayesian Deep Learning. *arXiv preprint arXiv:2001.10995*.

Wilson, A. G., McNamara, L. A., and Wilson, G. D. (2007). Information Integration for Complex Systems. *Reliability Engineering & System Safety*, 92(1):121–130.

Wolters, M. A. (2015). A Genetic Algorithm for Selection of Fixed-Size Subsets with Application to Design Problems. *Journal of Statistical Software*, 68.

Wu, Z. and Palmer, M. (1994). Verb Semantics and Lexical Selection. *arXiv preprint cmp-lg/9406033*.

Xue, B., Yu, J., Xu, J., Liu, S., Hu, S., Ye, Z., Geng, M., Liu, X., and Meng, H. (2021). Bayesian Transformer Language Models for Speech Recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7378–7382. IEEE.

Yu, J., Cristea, A. I., Harit, A., Sun, Z., Aduragba, O. T., Shi, L., and Al Moubayed, N. (2022). Efficient Uncertainty Quantification for Multilabel Text Classification. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Yu, S.-Z. (2010). Hidden Semi-Markov Models. *Artificial Intelligence*, 174(2):215–243.

Yue, M., Cheng, L.-Y., and Bari, R. A. (2008). A Markov Model Approach to Proliferation-Resistance Assessment of Nuclear Energy Systems. *Nuclear Technology*, 162(1):26–44.

Zhang, D., Bailey, A. D., and Djurdjanovic, D. (2016). Bayesian Identification of Hidden Markov Models and their use for Condition-Based Monitoring. *IEEE Transactions on Reliability*, 65(3):1471–1482.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. (2019). Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. *arXiv preprint arXiv:1902.03932*.

Ştefănescu, R., Schmidt, K., Hite, J., Smith, R. C., and Mattingly, J. (2017). Hybrid Optimization and Bayesian Inference Techniques for a Non-Smooth Radiation Detection Problem. *International Journal for Numerical Methods in Engineering*, 111(10):955–982.

APPENDICES

APPENDIX

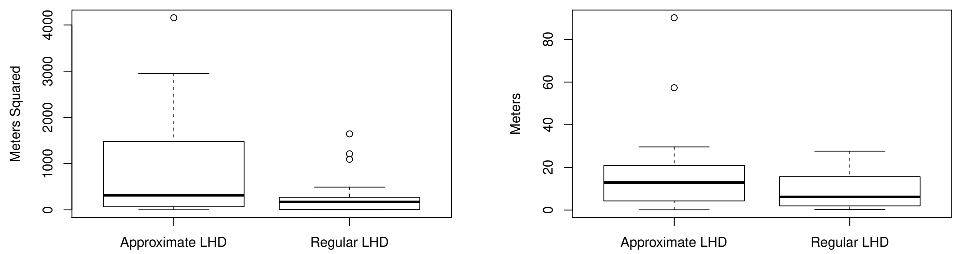
A

LOCALIZATION USING THE APPROXIMATE LATIN HYPERCUBE DESIGN

In order to compare the effect on source localization of using the approximate LHD, $\tilde{\mathbf{X}}$, instead of $\mathbf{X}$, the conventional LHD, we employ a similar procedure to the one employed for comparing the combinatorial and continuous detector configuration optimization methods. We use $\tilde{\mathbf{X}}$ and $\mathbf{X}$ to fit a surrogate between mutual information and detector configuration, and we optimize this surrogate to obtain an optimal detector configuration. We then use this configuration to obtain MCMC samples from $p(\boldsymbol{\theta}|\mathbf{Y})$. We repeat this 30 times. We compare the two procedures by using the HDR area and MAP error metrics.

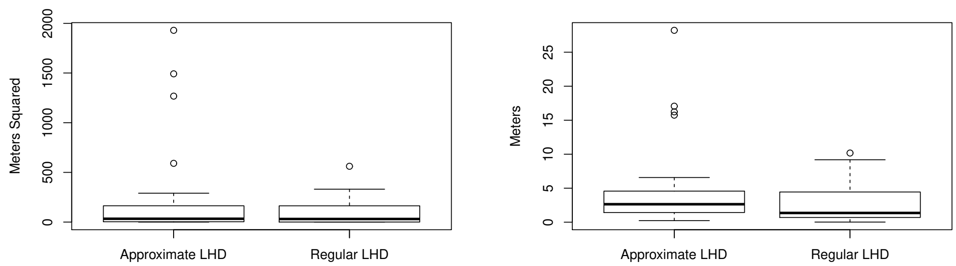
The procedure for generating the MCMC samples when using $\tilde{\mathbf{X}}$ is exactly the same as the procedure described in Section 2.4.1. The procedure when using $\mathbf{X}$ is also the same substituting $\mathbf{X}$ for $\tilde{\mathbf{X}}$.

Plots of the HDR area and MAP error for the 3-detector and 5-detector cases for initial



(a) HDR Area Comparison (b) MAP Error Comparison

Figure A.1: Comparison of Posterior Distributions in 3-Detector Case



(a) HDR Area Comparison (b) MAP Error Comparison

Figure A.2: Comparison of Posterior Distributions in 5-Detector Case

detector configuration selection are shown in Figures A.1 and A.2. For the 3-detector case, the approximate LHD produces less precise results, but with the exception of the outliers produced when using the approximate LHD, the accuracy of the two approaches seems comparable. For the 5-detector case, with the exception of the outliers, the HDR area and MAP error metrics appear comparable for the two approaches.

APPENDIX

B

3D RAY TRACING FOR URBAN SOURCE LOCALIZATION

For applying our methodology in Chapter 2 to more realistic 3D urban environments, we must adapt the computation of expected detector response as specified in Equation (2.3) to three dimensions. In theory, this should be simple. There is nothing about Equation (2.3) that is specific to two dimensions and most of the formula can be adapted very simply.

The detector-specific constants (Δt_i, A_i, and ϵ_i) remain unchanged and the distance between detector and source in the denominator simply becomes a three dimensional Euclidean distance calculation instead of a two dimensional distance calculation. The challenge is in computing the last term, specifically in computing the ℓ_p's. Recall that ℓ_p is the length of the ray from the source to the detector that intersects the p-th building on the path between the source and the detector. This intersection length is illustrated by the blue segment in Figure B.1. To compute this length, we must first determine the intersection points of the ray with the building (denoted by the purple stars in the figure). This can be a challenging task when dealing with general 3D objects. This problem is a subproblem of the larger ray-tracing problem that is studied extensively in the computer graphics literature (Kay and Kajiya 1986; Kuzmin 1994; Sellers and Lukac 2009).

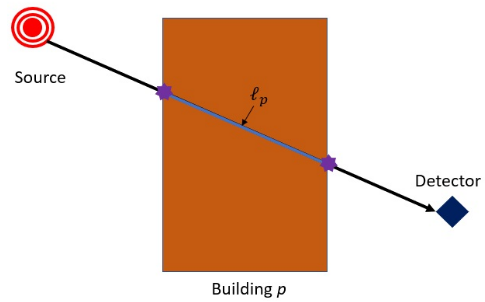


Figure B.1: Optimization of detector configurations over finite subset

The NVIDIA OptiX ray tracing engine is a computing framework that is optimized to perform ray-tracing calculations in parallel on a GPU (Ludvigsen and Elster 2010). In collaboration with colleagues in the North Carolina State University computer science department, we created a Python wrapper that computes the expected detector response and calls an OptiX subprocess to efficiently compute the ray intersections with buildings in the 3D urban environment Hollis and Gavane (2021).

To assess the improved efficiency of using OptiX for ray-tracing calculations, we also constructed a Python package that computes the expected detector response in 3D urban environments without calling an OptiX sub-process for ray-tracing. We tested these different implementations on a 3D version of the downtown Washington DC block used for the experiments in Chapter 2. This 3D urban environment is depicted in Figure B.2.

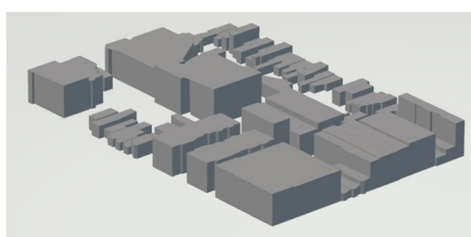


Figure B.2: 3D Urban Environment

For these experiments, we recorded the computation time for calculating the expected detector response for 1000 randomly placed radiation sources and a varying number of randomly placed detectors. The results of the experiment are shown in Figure B.3. As can be seen from the figure, the OptiX implementation is more efficient, and the efficiency gap increases as the number of detector-source pairs for which we are computing the expected detector response increases. This is to be expected as the OptiX engine is designed to be most efficient when dealing with a large number of rays.

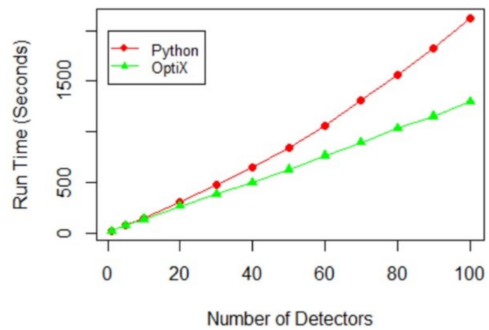


Figure B.3: OptiX vs. Python Ray-Tracing Computation Comparison

APPENDIX

C

GROUP BREAK DOWN OF OSTI CLASSES

Table C.1: OSTI Classes

Class Label	**Group**	**Train/Test**
1: Coal, Lignite, and Peat	Group 1: Fossil Fuels	1000 Train/53 Test
2: Petroleum	Group 1: Fossil Fuels	1000 Train/53 Test
3: Natural Gas	Group 1: Fossil Fuels	1000 Train/53 Test
4: Oil Shales and Tar Sands	Group 1: Fossil Fuels	900 Train/ 48 Test
5: Nuclear Fuels	Group 2: Fission and Nuclear Technologies	1000 Train/53 Test
7: Isotope and Radiation Sources	Group 2: Fission and Nuclear Technologies	388 Train/21 Test
11: Nuclear Fuel Cycle and Fuel Materials	Group 2: Fission and Nuclear Technologies	1000 Train/53 Test

Continued on next page

Table C.1 – *Continued from previous page*

Class Label	**Group**	**Train/Test**
12: Management of Radioactive Wastes, and Non-Radioactive Wastes from Nuclear Facilities	Group 2: Fission and Nuclear Technologies	1000 Train/53 Test
21: Specific Nuclear Reactors and Associated Plants	Group 2: Fission and Nuclear Technologies	1000 Train/53 Test
22: General Studies of Nuclear Reactors	Group 2: Fission and Nuclear Technologies	1000 Train/53 Test
8: Hydrogen	Group 3: Renewable Energy Sources	949 Train/50 Test
9: Biomass Fuels	Group 3: Renewable Energy Sources	1000 Train/53 Test
10: Synthetic Fuels	Group 3: Renewable Energy Sources	566 Train/30 Test
13: Hydro Energy	Group 3: Renewable Energy Sources	527 Train/28 Test
14: Solar Energy	Group 3: Renewable Energy Sources	1000 Train/53 Test
15: Geothermal Energy	Group 3: Renewable Energy Sources	1000 Train/53 Test
16: Tidal and Wave Power	Group 3: Renewable Energy Sources	72 Train/4 Test
17: Wind Energy	Group 3: Renewable Energy Sources	650 Train/35 Test
20: Fossil-Fueled Power Plants	Group 4: Power Generation and Distribution	1000 Train/53 Test
24: Power Transmission and Distribution	Group 4: Power Generation and Distribution	1000 Train/53 Test
25: Energy Storage	Group 5: Energy Storage, Conversion and Utilization	1000 Train/53 Test
29: Energy Planning, Policy, and Economy	Group 5: Energy Storage, Conversion and Utilization	1000 Train/53 Test

Continued on next page

Table C.1 – *Continued from previous page*

Class Label	**Group**	**Train/Test**
30: Direct Energy Conversion	Group 5: Energy Storage, Conversion and Utilization	1000 Train/53 Test
32: Energy Conservation, Consumption, and Utilization	Group 5: Energy Storage, Conversion and Utilization	1000 Train/53 Test
33: Advanced Propulsion Systems	Group 5: Energy Storage, Conversion and Utilization	1000 Train/53 Test
35: Arms Control	Group 6: National Defense	113 Train/6 Test
45: Military Technology, Weaponry, and National Defense	Group 6: National Defense	1000 Train/53 Test
98: Nuclear Disarmament, Safeguards, and Physical Protection	Group 6: National Defense	678 train/36 test
36: Material Science	Group 7: Materials	1000 Train/53 Test
37: Inorganic, Organic, Physical, and Analytical Chemistry	Group 8: Chemistry	1000 Train/53 Test
38: Radiation Chemistry, Radiochemistry, and Nuclear Chemistry	Group 8: Chemistry	623 Train/33 Test
40: Chemistry	Group 8: Chemistry	1000 Train/53 Test
39: Unlabeled	Group 9: Unlabeled Group	7 Train/0 Test
42: Engineering	Group 10: Engineering	1000 Train/53 Test
46: Instrumentation Related to Nuclear Science and Technology	Group 10: Engineering	1000 Train/53 Test
47: Other Instrumentation	Group 10: Engineering	1000 Train/53 Test
43: Particle Accelerators	Group 11: Physics	1000 Train/53 Test
66: Physics	Group 11: Physics	1000 Train/53 Test
70: Plasma Physics and Fusion Technology	Group 11: Physics	1000 Train/53 Test
71: Classical and Quantum Mechanics, General Physics	Group 11: Physics	1000 Train/53 Test

Continued on next page

Table C.1 – *Continued from previous page*

Class Label	**Group**	**Train/Test**
72: Physics Of Elementary Particles and Fields	Group 11: Physics	1000 Train/53 Test
73: Nuclear Physics and Radiation Physics	Group 11: Physics	1000 Train/53 Test
74: Atomic and Molecular Physics	Group 11: Physics	1000 Train/53 Test
75: Condensed Matter Physics, Superconductivity and Superfluidity	Group 11: Physics	1000 Train/53 Test
77: Nanoscience and Nanotechnology	Group 11: Physics	432 Train/23 Test
79: Astronomy and Astrophysics	Group 11: Physics	1000 Train/53 Test
44: Unlabeled	Group 12: Unlabeled Group	541 Train/29 Test
54: Environmental Sciences	Group 13: Environmental Sciences	1000 Train/53 Test
55: Unlabeled	Group 14: Unlabeled Group	951 Train/50 Test
56: Biology and Medicine	Group 15: Biology and Medicine	411 Train/22 Test
59: Basic Biological Sciences	Group 15: Biology and Medicine	1000 Train/53 Test
60: Applied Life Sciences	Group 15: Biology and Medicine	1000 Train/53 Test
61: Radiation Protection and Dosimetry	Group 15: Biology and Medicine	454 Train/24 Test
62: Radiology and Nuclear Medicine	Group 15: Biology and Medicine	1000 Train/53 Test
63: Effects of Radiation and Other Pollutants on Biological Materials and Organisms	Group 15: Biology and Medicine	1000 Train/53 Test
57: Unlabeled	Group 16: Unlabeled Group	10 Train/1 Test

Continued on next page

Table C.1 – *Continued from previous page*

Class Label	**Group**	**Train/Test**
58: Geosciences	Group 17: Geosciences	1000 Train/53 Test
96: Knowledge Management and Preservation	Group 18: Information Science	23 Train/1 Test
97: Mathematics and Computing	Group 19: Mathematics and Computing	804 Train/43 Test
99: General and Miscellaneous	Group 20: Law	1000 Train/53 Test

APPENDIX

D

BELIEF MATCHING ADDITIONAL EXPERIMENTAL RESULTS

D.1 Group Homogeneity for Different Choices of Top K

In the Section 4.4.3, the GH metric is computed using the top 3 selected topics. In Figure D.1, we compute the GH metric for top 2-6 selected topics for the BM($\lambda = 0.01$), HierCorrBM($\sigma_0 = 10, \psi = 0.5$), and Dropout($c_p = 0.3, h_p = 0.3, a_p = 0.1$) models. It is clear from the figure that no matter the choice of top-k, the ordering of the methods by the GH metric is the same with the Heirarchical Correlated Belief Matching approach yielding the best results.

D.2 Additional Experimental results for Toy, MNIST, and OSTI Datasets

Figure D.1: Group Homogeneity For Different Choices of Top K

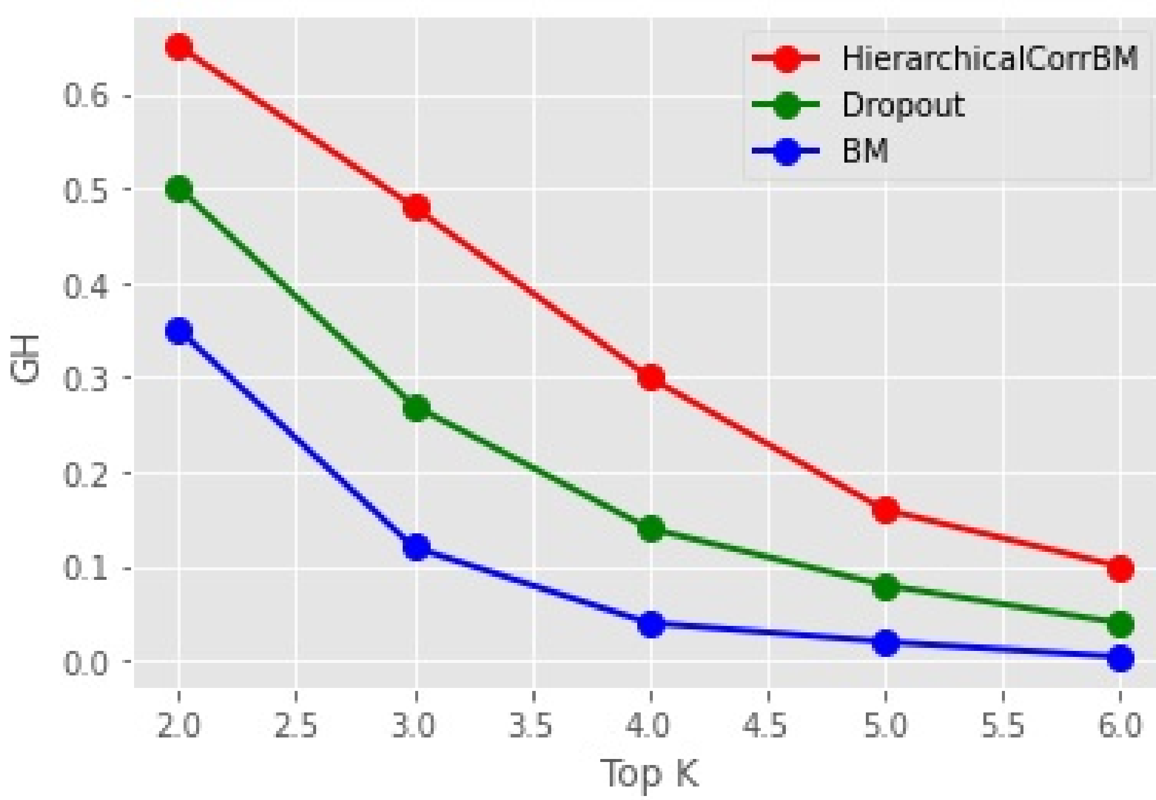


Table D.1: Toy Data Method Comparison

	Accuracy	ECE	CVG	WDTH	RA
BM($\lambda = 0.01$)	84.6%	0.083	85.4%	1.02	0.617
BM($\lambda = 0.025$)	86.7%	0.029	88.1%	1.03	0.645
BM($\lambda = 0.05$)	85.7%	0.091	87.5%	1.05	0.670
BM($\lambda = 0.075$)	85.4%	0.132	87.3%	1.05	0.644
BM($\lambda = 0.1$)	86.7%	0.196	89.1%	1.10	0.600
BM($\lambda = 0.25$)	85.7%	0.367	96.7%	4.05	0.672
CorrBM($\sigma_0 = 1, \psi = 0.1$)	77.0%	0.491	89.2%	1.75	0.991
CorrBM($\sigma_0 = 3, \psi = 0.1$)	85.7%	0.144	91.0%	1.18	0.987
CorrBM($\sigma_0 = 5, \psi = 0.01$)	86.7%	0.049	87.8%	1.03	0.979
CorrBM($\sigma_0 = 5, \psi = 0.05$)	86.2%	0.016	88.4%	1.07	0.975
CorrBM($\sigma_0 = 5, \psi = 0.1$)	85.1%	0.029	90.0%	1.14	0.921
CorrBM($\sigma_0 = 5, \psi = 0.25$)	86.2%	0.054	96.0%	1.86	0.942
CorrBM($\sigma_0 = 5, \psi = 0.5$)	84.6%	0.094	98.6%	3.42	0.849
CorrBM($\sigma_0 = 7, \psi = 0.1$)	84.4%	0.090	89.5%	1.13	0.924
CorrBM($\sigma_0 = 9, \psi = 0.1$)	84.8%	0.108	89.5%	1.13	0.870
Drop($c_p = 0.1, h_p = 0.4, a_p = 0.5$)	85.2%	0.033	93.2%	1.39	0.817
Drop($c_p = 0.3, h_p = 0.4, a_p = 0.5$)	82.2%	0.032	93.0%	1.41	0.825
Drop($c_p = 0.5, h_p = 0.4, a_p = 0.55$)	82.4%	0.030	93.7%	1.58	0.844

Table D.2: MNIST-Fashion Method Comparison

	Accuracy	ECE	CVG	WDTH	RA
BM($\lambda = 0.001$)	90.2%	0.046	90.6%	1.01	0.77
BM($\lambda = 0.0025$)	90.4%	0.031	90.9%	1.01	0.77
BM($\lambda = 0.005$)	90.5%	0.019	91.2%	1.02	0.77
BM($\lambda = 0.01$)	90.4%	0.06	91.2%	1.02	0.76
BM($\lambda = 0.1$)	89.8%	0.371	94.3%	1.25	0.78
CorrBM($\sigma_0 = 10, \psi = 0.05$)	90.2%	0.032	93.3%	1.10	0.89
CorrBM($\sigma_0 = 10, \psi = 0.1$)	90.1%	0.031	95.3%	1.20	0.90
CorrBM($\sigma_0 = 10, \psi = 0.15$)	90.3%	0.042	96.2%	1.37	0.91
CorrBM($\sigma_0 = 12, \psi = 0.05$)	90.2%	0.031	93.6%	1.11	0.89
CorrBM($\sigma_0 = 12, \psi = 0.1$)	90.2%	0.046	95.3%	1.20	0.91
CorrBM($\sigma_0 = 12, \psi = 0.15$)	90.2%	0.054	96.9%	1.47	0.90
Drop($c_p = 0.3$)	90.1%	0.029	95.9%	1.22	0.82
Drop($c_p = 0.5$)	90.1%	0.014	96.8%	1.27	0.82
HierCorrBM($\sigma_0 = 0.5, \psi = 5.5$)	89.0%	0.15	94.4%	1.17	NA
HierCorrBM($\sigma_0 = 1, \psi = 5$)	90.0%	0.025	93.3%	1.09	NA
HierCorrBM($\sigma_0 = 1, \psi = 5.5$)	90.2%	0.03	93.2%	1.09	NA
HierCorrBM($\sigma_0 = 1, \psi = 6$)	90.4%	0.046	93.3%	1.10	NA
HierCorrBM($\sigma_0 = 2, \psi = 5$)	90.3%	0.063	93.1%	1.09	NA

Table D.3: OSTI Method Comparison

	Accuracy	ECE	CVG	WDTH	GH
BM($\lambda = 0.001$)	68.5%	0.137	69.2%	1.03	0.151
BM($\lambda = 0.005$)	68.8%	0.056	70.0%	1.06	0.127
BM($\lambda = 0.01$)	68.3%	0.129	70.4%	1.09	0.120
BM($\lambda = 0.05$)	69.1%	0.454	76.1%	2.87	0.108
BM($\lambda = 0.1$)	68.3%	0.541	84.0%	10.38	0.083
HierCorrBM($\sigma_0 = 9, \psi = 0.1$)	66.1%	0.071	69.7%	1.18	0.457
HierCorrBM($\sigma_0 = 9, \psi = 0.5$)	67.5%	0.137	79.1%	4.31	0.504
HierCorrBM($\sigma_0 = 9, \psi = 1$)	67.1%	0.139	83.6%	9.45	0.470
HierCorrBM($\sigma_0 = 10, \psi = 0.1$)	66.8%	0.072	71.6%	1.20	0.357
HierCorrBM($\sigma_0 = 10, \psi = 0.5$)	67.3%	0.132	79.5%	4.61	0.479
HierCorrBM($\sigma_0 = 10, \psi = 1$)	64.4%	0.156	83.1%	12.28	0.491
HierCorrBM($\sigma_0 = 11, \psi = 0.5$)	67.3%	0.138	78.6%	4.27	0.461
HierCorrBM($\sigma_0 = 11, \psi = 1$)	66.0%	0.165	83.3%	11.43	0.498
Drop($c_p = 0.3$, $h_p = 0.3$, $a_p = 0.1$)	68.7%	0.131	83.3%	1.76	0.266
Drop($c_p = 0.5$, $h_p = 0.4$, $a_p = 0.5$)	63.0%	0.068	82.8%	2.92	0.322