The Cauchy-Net Mixture Model for Clustering with Anomalous Data

Matthew D. Slifko

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

Doctor of Philosophy
in
Statistics

Scotland C. Leman, Chair
David S. Bieri
Shyam Ranganathan
Eric P. Smith

July 24, 2019
Blacksburg, Virginia

Keywords: Bayesian Nonparametrics, Dirichlet Process Mixture Model, Clustering, Anomaly Detection
Copyright 2019, Matthew D. Slifko
The Cauchy-Net Mixture Model for Clustering with Anomalous Data

Matthew D. Slifko

(ABSTRACT)

We live in the data explosion era. The unprecedented amount of data offers a potential wealth of knowledge but also brings about concerns regarding ethical collection and usage. Mistakes stemming from anomalous data have the potential for severe, real-world consequences, such as when building prediction models for housing prices. To combat anomalies, we develop the Cauchy-Net Mixture Model (CNMM). The CNMM is a flexible Bayesian nonparametric tool that employs a mixture between a Dirichlet Process Mixture Model (DPMM) and a Cauchy distributed component, which we call the Cauchy-Net (CN). Each portion of the model offers benefits, as the DPMM eliminates the limitation of requiring a fixed number of components and the CN captures observations that do not belong to the well-defined components by leveraging its heavy tails. Through isolating the anomalous observations in a single component, we simultaneously identify the observations in the net as warranting further inspection and prevent them from interfering with the formation of the remaining components. The result is a framework that allows for simultaneously clustering observations and making predictions in the face of the anomalous data. We demonstrate the usefulness of the CNMM in a variety of experimental situations and apply the model for predicting housing prices in Fairfax County, Virginia.
We live in the data explosion era. The unprecedented amount of data offers a potential wealth of knowledge but also brings about concerns regarding ethical collection and usage. Mistakes stemming from anomalous data have the potential for severe, real-world consequences, such as when building prediction models for housing prices. To combat anomalies, we develop the Cauchy-Net Mixture Model (CNMM). The CNMM is a flexible tool for identifying and isolating the anomalies, while simultaneously discovering cluster structure and making predictions among the non-anomalous observations. The result is a framework that allows for simultaneously clustering and predicting in the face of the anomalous data. We demonstrate the usefulness of the CNMM in a variety of experimental situations and apply the model for predicting housing prices in Fairfax County, Virginia.
Dedication

For my amazing family and their unconditional love and support.
Acknowledgments

My time in Blacksburg was incredibly rewarding and among the happiest years of my life. Through the ups and downs of graduate school and life, I was surrounding by an amazing group of people to whom I am forever grateful.

First, I would like thank my advisor, Scotland Leman. I appreciate his time, guidance, and seemingly infinite amount of patience more than he will ever realize. He taught me countless lessons about statistics, research, and teaching. Although those lessons may one day fade, the lessons he taught me about life and treating others with patience, compassion, and understanding will never be forgotten.

I would also like to thank my dissertation committee, David Bieri, Shyam Ranganathan, and Eric Smith. Their insights and thoughtful conversations drastically improved the quality of my research and my abilities as a statistician.

I often brag that the Department of Statistics at Virginia Tech is filled with the best faculty, students, and staff. I will forever cherish the friendships and memories. I especially want to thank Nathan Wycoff, Sierra Merkes, and J.T. Fry. I spent more time with these three than everyone else combined. Not only did they help me grow professionally, but they kept me balanced and happy. I may have mentioned the phrase “terrible influence” to describe a subset of these three, but the reality is that I treasure our time together.

To my family and friends outside of Blacksburg, the never-ending support means the world to me. I realize that I have been difficult at times, canceling plans in order to work, ignoring messages, and being absent. Despite this, my family, and friends who became family, has stuck by my side no matter the circumstances.

Finally, I would like to God for helping me to find the right program, surrounding me with incredible people, and providing the strength, energy, and faith to keep going when it would have been easier to quit.

This work is supported by National Science Foundation Training Fellowship (NSF: NRT-DESE #1545362).
# Contents

<table>
<thead>
<tr>
<th>List of Figures</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
<td>xiv</td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 Motivation .............................................. 1  
1.2 Overview of Technique .................................. 3  
1.3 Thesis Preview ........................................... 4  

## 2 Clustering Background

2.1 Clustering .................................................. 6  
2.1.1 $K$-means ................................................. 8  
2.1.2 Hierarchical Clustering ................................ 11  
2.1.3 Model-based Clustering ................................. 12  
2.2 Clustering in the Presence of Anomalies .................. 15  

## 3 Dirichlet Process Mixture Model

3.1 Bayesian Modeling ......................................... 18  
3.2 Bayesian Nonparametric Modeling .......................... 20  
3.3 Overview of Dirichlet Processes ........................... 21  
3.3.1 Dirichlet Distribution .................................. 21  
3.3.2 Dirichlet Process ........................................ 23  
3.3.3 Stick-breaking Representation .......................... 27
# List of Figures

1.1 The 5 Vs of big data. Each V represents one of the challenges associated with big data. The challenges are intertwined and the overall challenge of adding value requires finding solutions to the other challenges. ........................................... 2

2.1 Possible two-dimensional cluster structures. The color and plotting symbol indicate cluster membership. .......................................................... 8

2.2 An example of the contours and a sample from a Gaussian mixture model. ........ 13

3.1 The 2-probability simplex (gray triangle) consists of 3-dimensional points constrained to be nonnegative and sum to 1. ................................. 22

3.2 Realizations of random probability distributions generated according to a Dirichlet process with a standard normal base measure, \( G_0 \). The rows correspond to different values of the concentration parameter, \( \alpha_0 \), and the columns represent two independent realizations. .................................................. 26

3.3 A representation of the the stick-breaking weights. ................................. 28

3.4 Chinese Restaurant process example showing the assignments of the first 10 customers (white squares) to tables (large, shaded circles). Tables with a solid outline represent active tables or clusters. Empty tables have a dashed outline. ............ 32

3.5 Modified Chinese restaurant process example showing the assignments of the first 10 customers (white squares) to tables (large, shaded circles). Tables with a solid outline represent active tables or clusters. Empty tables have a dashed outline. The table parameter, drawn from the base measure, applies to all customers at the table. 33

3.6 Schematic of DPMM data generating model based on the stick-breaking representation of a DP. .......................................................... 36
3.7 Modified Chinese restaurant process mixture model representation of a Dirichlet process mixture model showing the assignments of the first 10 customers (white squares) to tables (large, shaded circles). Observations for the individuals (small, shaded circles) are i.i.d. from parametric distribution with the table parameter.

4.1 Three scenarios illustrating the varying impacts of anomalies on clustering. The color and plotting symbol indicate cluster membership, with solid shapes representing the cluster centroid. In the first scenario, the anomalies prevent the discovery of cluster structure. Some structure (blue cluster) is recovered in the second, while most structure is recovered in the third.

4.2 Motivation for clustering anomalous data using a Cauchy distribution. When deciding whether the isolated observation (red circle, far right) belongs in one of the well-defined normally distributed components or the Cauchy distributed component, consider the height of each density curve at the isolated point weighted by the number of observations already in each component. In this case, the isolated observation most likely belongs to the Cauchy component.

4.3 Process and results of a CNMM. Applying the CNMM to unlabeled data partitions the observations into types: anomalies (solid red circles) and regular observations (solid gray circles). By isolating the anomalies in the CN and mitigating their impact, we discover regular clusters (dashed ellipses) containing the regular observations.

4.4 Modified Chinese restaurant process mixture model representation of a Cauchy-Net Mixture Model showing the assignments of the first 10 customers (white squares) to tables (large, shaded circles). Table 0 represents the Cauchy-Net, whose parameters are fixed, rather than randomly generated from $G_0$. Observations for the individuals (small, shaded circles) are i.i.d. from the parametric family with the density defined by the table parameter. Empty regular tables have a dashed outline.

5.1 A realization of data from a 2D DPMM illustrating the simulation study’s assumptions of a mixture of spherical, well-separated normal distributions. The symbols and colors indicate component membership. The five large superimposed circles represent the two standard deviation theoretical contour for each normal component.

5.2 Four randomly generated datasets using $n = 150$, $p = 2$, and $\pi_A = 0.05$ but different levels of $\alpha_0$. The solid pale purple circles represent the true anomalies. The observations in each regular component share a plotting symbol (e.g., $\Box$, $\star$, $+$, etc.). Components containing a small number of observations relative to the dimensionality are indistinguishable from the anomalies, thereby raising the Effective Proportion of Anomalies (EPA) beyond $\pi_A$. 

x
5.3 An illustration of the effective proportion of anomalies (EPA) under a variety of experimental conditions. The EPA is considerably larger than $\pi_A$ in cases where the number of components is large (i.e., $\alpha_0$ is large), the dimensionality is high, and the number of observations is small. In some instances, the EPA reaches 1, which suggests that all observations are indistinguishable from the anomalies.

5.4 A randomly generated dataset corresponding to the experimental condition $n = 300$, $p = 2$, $\alpha_0 = 2$, and $\pi_A = 0.25$. Observations belonging to the same component share both the true component label as the plotting symbol and a plotting color. The superimposed circles depict the theoretical three standard deviation contour for the ten regular components.

5.5 A comparison of the CNMM and DPMM methods by exploring the discovered component labels at the end of a given Gibbs iteration. The methods produce similar results but handle the anomalies differently.

5.6 The adjusted Rand index (ARI) versus the Gibbs sampling iteration for the CNMM and DPMM.

5.7 Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size ($n = 300$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using $\text{ARI(CNMM)} - \text{ARI(DPMM)}$.

5.8 Comparison of the ARI using a fixed sample size ($n = 300$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$).

5.9 Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size ($n = 300$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using $\text{ARI(CNMM Probabilistic)} - \text{ARI(CNMM Empirical)}$.

5.10 Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size ($n = 900$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using $\text{ARI(CNMM)} - \text{ARI(DPMM)}$.

5.11 Comparison of the ARI using a fixed sample size ($n = 900$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$).
5.12 Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size \((n = 900)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The difference is computed using \(\text{ARI(CNMM Probabilistic)} - \text{ARI(CNMM Empirical)}\). .......................................................... 107

6.1 Regression coefficient estimates for the three stable regular components discovered by the CNMM and DPMM. ................................................................. 113

A.1 Comparison of the difference in the mean ARI for the CNMM and DPMM using a fixed sample size \((n = 100)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The difference is computed using \(\text{ARI(CNMM)} - \text{ARI(DPMM)}\). ... 127

A.2 Comparison of the ARI using a fixed sample size \((n = 100)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). ................................................................. 128

A.3 Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size \((n = 100)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The difference is computed using \(\text{ARI(CNMM Probabilistic)} - \text{ARI(CNMM Empirical)}\). ................................................................. 129

A.4 Comparison of the difference in the mean ARI for the CNMM and DPMM using a fixed sample size \((n = 150)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The difference is computed using \(\text{ARI(CNMM)} - \text{ARI(DPMM)}\). ... 130

A.5 Comparison of the ARI using a fixed sample size \((n = 150)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). ................................................................. 131

A.6 Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size \((n = 150)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The difference is computed using \(\text{ARI(CNMM Probabilistic)} - \text{ARI(CNMM Empirical)}\). ................................................................. 132

B.1 Comparison of the difference in the mean ARI using a fixed sample size \((n = 100)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The labels represent the order of subtraction. The difference is computing using \(\text{ARI(CNMM)} - \text{ARI(DPMM)}\). ................................................................. 134
B.2 Comparison of the ARI using a fixed sample size \((n = 100)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). .................................................. 135

B.3 Comparison of the difference in the mean ARI using a fixed sample size \((n = 150)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The labels represent the order of subtraction. The difference is computing using ARI(CNMM) - ARI(DPMM). .................................................. 136

B.4 Comparison of the ARI using a fixed sample size \((n = 150)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). .................................................. 137

B.5 Comparison of the difference in the mean ARI using a fixed sample size \((n = 300)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The labels represent the order of subtraction. The difference is computing using ARI(CNMM) - ARI(DPMM). .................................................. 138

B.6 Comparison of the ARI using a fixed sample size \((n = 300)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). .................................................. 139

B.7 Comparison of the difference in the mean ARI using a fixed sample size \((n = 900)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The labels represent the order of subtraction. The difference is computing using ARI(CNMM) - ARI(DPMM). .................................................. 140

B.8 Comparison of the ARI using a fixed sample size \((n = 900)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). .................................................. 141
## List of Tables

3.1 Random discrete probability distribution generated by a CRP. ............................................ 33  
3.2 Notation for DPMM inference. ........................................................................................................... 42  
3.3 DPMM conditional prior probabilities for assigning $\theta_n$ given $\theta_1, \ldots, \theta_{n-1}$. ........ 44  
3.4 DPMM conditional prior probabilities for assigning $c_n$ given $c_1, \ldots, c_{n-1}$. ........ 44  
4.1 Conditional prior probabilities for assigning $\theta_n$ given $\theta_1:(n-1)$ in the DPMM and CNMM. .................................................................................................................................................. 63  
5.1 Factor levels for generating the datasets in the simulation studies. Each unique combination is referred to as an experimental condition. ................................................................. 75  
5.2 Example showing connection between insufficient components and effective proportion of anomalies (EPA). The component labels, mixture weights, and expected number of observations are shown in the first three rows. Using the expected counts and the threshold $1.5(p + 1)$, components are classified as sufficient (✓) or insufficient (✗). The EPA shown in the last column is found by summing the weights corresponding to the insubstantial components and the Cauchy. ................................................................. 82  
5.3 Tabular summary of the component labels discovered by the CNMM and shown in Figure 5.5a. The column totals represent the number of observations assigned to each true component. Each row depicts the composition of a discovered component where the cell entries show the frequencies of the true labels for the observations comprising the discovered component. The column label for the highlighted cell indicates the most prevalent true component in the discovered component. Each column describes the allocation of the true component’s observations to the discovered components. .................................................................................................................................................. 93
5.4 Tabular summary of the component labels discovered by the DPMM and shown in Figure 5.5b. The column totals represent the number of observations assigned to each true component. Each row depicts the composition of a discovered component where the cell entries show the frequencies of the true labels for the observations comprising the discovered component. The column label for the highlighted cell indicates the most prevalent true component in the discovered component. Each column describes the allocation of the true component’s observations to the discovered components.

6.1 The number of components, $K$, discovered by the CNMM and DPMM using the Fairfax County housing data.

6.2 The number of observations linked to each component in the CNMM and DPMM. The “Agreement” column indicates the number of observations for which both methods produce the same the linked assignment. As an example, of the 137 observations linked to the third component in the CNMM, 132 of the observations are also linked to the third component by the DPMM.

6.3 The percentile rank for the hedonic features and closing price for each anomalous observation. Percentile ranks of 0 and 100 indicate the minimum and maximum values in the dataset, respectively.

6.4 Empirical assignment probabilities for the anomalous observations under the CNMM and DPMM.
Chapter 1

Introduction

1.1 Motivation

A desire for data-driven solutions, increasingly complex scientific questions, and advancements in computational resources (storage, memory, and algorithms) are driving the explosion of data. Bolón-Canedo et al. (2015) report that humans collected about 5 exabytes ($10^{18}$ bytes) of data from the beginning of human existence until 2003. Now, we generate roughly 5 exabytes of data every 2 days. The unprecedented amount of data offers a potential wealth of knowledge provided that researchers find ways of successfully extracting the relevant information.

The nature of “big data” presents researchers with many challenges, which we summarize through the five “Vs” of big data: volume, variety, velocity, veracity, and value (Zikopoulos et al., 2013; Zhai et al., 2014). Volume refers to the challenges caused by the sheer size of big data and serves as an umbrella for issues stemming from a large number of observations (big $n$), a high number of dimensions (big $p$), or any other issue making storing, processing, and computing difficult. Variety alludes to the challenges associated with collecting and processing different types of data. Beyond traditional data sets with easily discernible observations and a fixed number of dimensions, which are usually quantitative, many data sources now include unstructured text, image, or video data, each of which requires a different approach for analyzing. Velocity of big data describes the
challenges associated with the rate at which we collect and use data. In many instances, collection occurs more quickly than analysis and traditional procedures are too slow to allow for real-time processing and decision making. Veracity refers to challenges associated with the quality of the data. We want accurate and trustworthy data, but when data are pieced together from multiple sources, incompleteness and inconsistencies are common. The final V, value, reinforces that the ultimate challenge is to use data to produce a benefit. We collect and analyze big data to extract information and gain knowledge. Figure 1.1 depicts the 5 Vs of big data and emphasizes that the Vs are not disjoint. For instance, the velocity at which we collect data contributes to the challenge of verifying the quality. Additionally, value surrounds the other four Vs because leveraging the data and adding value requires finding solutions to the other challenges.

Figure 1.1: The 5 Vs of big data. Each V represents one of the challenges associated with big data. The challenges are intertwined and the overall challenge of adding value requires finding solutions to the other challenges.

The prediction of housing prices is a challenging big data problem that exemplifies several of the
Vs. Beginning in 2006, websites, such as Zillow, Trulia, and Realtor, began popularizing the use of Automated Valuation Models (AVMs) for providing quick and relatively inexpensive estimates of property values. Due to the sheer volume of properties in major metropolitan areas, building city-wide AVMs is computationally challenging. Partitioning the properties into submarkets, and modeling each submarket, is one approach of reducing the computational burden. Unfortunately, there is still a veracity issue. Real estate websites, in part, base predictions of housing prices on data scraped from public records. These data are often contaminated in the sense that the data are outdated, incomplete, filled with errors, and contain inconsistencies in the way different locales report information.

The phrase “contaminated data” suggests an impurity. In many instances, such as with measurement, recording, and acquisition errors, there is an impurity. In other instances, such as with outliers, the data could be correct, just atypical. Along these lines, we also define an unsubstantiated observation as data that is atypical of the information in the dataset, but, unlike outliers, we expect to see similar observations provided we collect more data. For the remainder of this manuscript we use the phrase “anomalous data” and its variants to include outliers, contaminated data, unsubstantiated data, and any other type of potentially problematic observation.

The consequences of incorrectly predicting housing prices are potentially severe. When a real estate website estimates a property’s value at 75% of the asking price, the website’s prediction is likely to cost the seller money because potential buyers will undercut the asking price. Situations like this, particularly if systematically discriminating against a group of people or neighborhoods, are a serious concern, reminiscent of the discriminatory practice of redlining. Since predictions made by real estate websites have a significant impact on actual closing prices, there is a need for limiting the impact of anomalous data on predictions.

1.2 Overview of Technique

Motivated by the application of predicting housing prices, we seek a technique that allows for simultaneously clustering and making predictions on data that are potentially riddled with anom-
lies. Clustering is intuitive, not only on housing data but in many situations, because of our natural inclination to sort objects into groups. Clustering also offers the opportunity to reduce computation costs by allowing us to build localized models on subsets of the data. Unfortunately, many of the existing clustering techniques suffer from two major limitations: (1) the need to specify the number of clusters a priori and (2) the detrimental impact of anomalies on the ability of the clustering technique to identify underlying structures.

We present a general technique, the Cauchy-Net Mixture Model (CNMM), that overcomes both of these limitations. The CNMM extends a Dirichlet Process Mixture Model (DPMM) (Antoniak, 1974) by creating a mixture of a DPMM with an additional Cauchy distributed component, which we refer to as the Cauchy-Net (CN). Each portion of the model contributes desirable characteristics. The DPMM offers a probabilistic approach to inference and eliminates the limitation of requiring a fixed number of a components, while the CN leverages its heavy tails to capture observations that do not fit into the well-defined components. By isolating the anomalous observations in the CN, we simultaneously flag the observations in the net as warranting further inspection and prevent them from interfering with the formation of the remaining clusters and/or the predictions.

1.3 Thesis Preview

In this work, we develop the Cauchy-Net Mixture Model (CNMM). Our goal is to establish the CNMM as a viable option for clustering in the presence of anomalous observations. In particular, we seek to understand how the CNMM performs relative to a DPMM and determine under which conditions the use of a CNMM is beneficial. Understanding the behavior of the CNMM when clustering will serve as a strong foundation for future work in clustering and prediction.

In Chapter 2, we review some of the most popular clustering methods, discuss their strengths and weaknesses, and describe several methods for clustering data that contain anomalies. In Chapter 3, we provide an overview of Bayesian nonparametrics, review Dirichlet processes (DP) and DP-MMs, and explore approaches for inference. Chapter 4 develops the CNMM as an extension of the DPMM. Chapter 5 provides detailed simulation studies and the results of investigating the ef-
fectiveness of the CNMM for clustering. Chapter 6 extends the CNMM from a tool for clustering to a tool for clustering and prediction through a case study for housing prices in Fairfax County. Chapter 7 includes conclusions and future work.
Chapter 2

Clustering Background

2.1 Clustering

Clustering, or cluster analysis, is an unsupervised learning task commonly used for identifying patterns in multivariate data. For a set of observations with unknown group memberships, the goal of clustering is to identify clusters of observations (also called subsets or groups) such that observations within the same cluster are more similar to one another than to observations in other clusters. Different techniques for quantifying the similarity, or dissimilarity, between observations produce different clustering methods. Some of the most commonly employed techniques are based on distance between points, connectivity of clusters, density of observations, and probability distributions.

Jain (2010) identifies three common uses for the patterns discovered through clustering: (1) detection of underlying structure, (2) natural classification, and (3) compression. The first of these describes the role clustering plays as an exploratory tool for gaining insights into underlying structures such as the presence of clusters, anomalous observations, and differentiating features of the clusters. The second use, natural classification, refers to the way in which clustering is used to learn about the similarity between observations, as observations sharing the same cluster assignment have a higher level of similarity than those in different clusters. Finally, clustering aids in
the compression of data through the use of summary statistics for the elements of a cluster. For instance, in \( K \)-means (see Section 2.1.1), the centroid of a cluster represents the average, or prototypical, value for the observations within the cluster. Regardless of the intended use of clustering, decisions about how to perform the task of clustering are usually problem specific.

Clustering remains a challenging problem, despite its widespread use for applications across many disciplines. Challenges include determining the number of clusters, limiting the impact of anomalous data, and handling differences in cluster structures, such as variety in cluster shapes or densities. Figure 2.1 illustrates some of these scenarios on two-dimensional (2D) data. In 2D, the human eye is adept at visually identifying clusters, particularly in the well-separated cases shown in the first two plots (Figure 2.1a and Figure 2.1b); however, when algorithmically detecting clusters, differences in the shapes and/or densities, such as the case in Figure 2.1b, require more flexible tools than were needed for the similarly dense and similarly shaped clusters in Figure 2.1a. Without the aid of colors and/or plotting symbols, the visual identification of clusters proves difficult when the clusters are not well-separated or overlap, as shown in Figure 2.1c. The presence of anomalies further hinders the task of clustering. The final two plots (Figure 2.1d and Figure 2.1e) contain dissimilarly shaped clusters, namely a washer-like cluster (open red circles) and a circular, cloud-like cluster (open blue triangles). The addition of anomalies (black addition symbol) in Figure 2.1e makes the identification of clusters less obvious. As the dimensionality of data grows, the curse of dimensionality presents additional challenges and algorithmic approaches replace the option of visual identification.

In what follows, we explore some common clustering approaches, while also discussing their corresponding strengths and weaknesses. For the sake of consistent notation, we assume \( n \) independent observations, each of which contains information on \( p \) attributes. We denote the \( i^{th} \) observation as the \( p \)-vector \( \mathbf{x}_i \) and the collection of all \( n \) observations as the \( n \times p \) matrix \( \mathbf{X} \). The goal is to discover patterns in \( \mathbf{X} \) by clustering the observations. Letting \( K \) represent the total number of clusters, we index the clusters \( \{1, \ldots, k, \ldots, K\} \), where \( k \) is the index of an arbitrary cluster. Assume \( c_i \), where \( c_i \in \{1, \ldots, K\} \), denotes the cluster assignment of the \( i^{th} \) observation and the set of all \( n \) assignments is the \( n \)-vector \( \mathbf{c} = \{c_1, \ldots, c_n\}' \). Finally, let \( n_k \) represent the
MacQueen's procedure differs from the procedure now commonly referred to as K-means by sequentially group observations by assigning each observation to the group whose mean is closest. The term “K-means” is attributed to MacQueen (1967), who proposed using a set of K means to sequentially group observations by assigning each observation to the group whose mean is closest to the observation. After an observation is added to a group, the group’s mean is updated. Although MacQueen’s procedure differs from the procedure now commonly referred to as K-means, his

Figure 2.1: Possible two-dimensional cluster structures. The color and plotting symbol indicate cluster membership.

number of observations in the $k^{th}$ cluster.

2.1.1 K-means

K-means is one of the most simplistic, yet most famous, distance-based clustering techniques. The term “K-means” is attributed to MacQueen (1967), who proposed using a set of K means to sequentially group observations by assigning each observation to the group whose mean is closest to the observation. After an observation is added to a group, the group’s mean is updated. Although MacQueen’s procedure differs from the procedure now commonly referred to as K-means, his
work, along with Steinhaus (1956), Lloyd (1982), and Ball and Hall (1965), is important to the development of modern \( K \)-means.

The goal of \( K \)-means is to partition \( n \) observations into \( K \) clusters, by minimizing, across all clusters, the sum of squared deviations between a cluster’s observations and its empirical mean. Denoting the empirical mean of the \( k^{th} \) cluster as \( \mu_k \) and referring to it as the centroid, the procedure begins with the user specifying the number of clusters, \( K \) (the \( K \) in \( K \)-means), and a set of \( K \) centroids. The objective function, \( Z(\cdot, \cdot) \), represents the total, across all clusters, within cluster sum of squared deviations and is given by:

\[
Z(c, \{\mu_1, \ldots, \mu_K\}) = \sum_{k=1}^{K} \sum_{\{i: c_i = k\}} \| x_i - \mu_k \|^2, \tag{2.1}
\]

where \( \| \cdot \| \) indicates a distance metric and is typically the Euclidean distance in \( K \)-means. The left side of Equation 2.1 emphasizes that the objective is a function of the set of cluster assignments, \( c \), and the set of \( K \) centroids.

Finding the cluster assignments is an optimization problem, with the goal of finding the cluster assignments and centroids that minimize \( Z(\cdot, \cdot) \):

\[
\text{ArgMin}_{c, \{\mu_1, \ldots, \mu_K\}} Z(c, \{\mu_1, \ldots, \mu_K\}) = \text{ArgMin}_{c, \{\mu_1, \ldots, \mu_K\}} \sum_{k=1}^{K} \sum_{\{i: c_i = k\}} \| x_i - \mu_k \|^2.
\]

An iterative approach for minimizing \( Z(\cdot, \cdot) \) is to alternate between updating the cluster assignments and updating the centroids. The set of all cluster assignments, \( c \), is updated by assigning each observation to the cluster with the nearest centroid. After updating all cluster assignments, the set of centroids \( \{\mu_1, \ldots, \mu_K\} \) is updated using the empirical mean of the observations within each cluster. The \( K \)-means procedure alternates between these two steps until the cluster assignments stop changing. Algorithm 1 outlines the complete procedure.
Algorithm 2.1: $K$-means

Initialize: Specify number of clusters $K$ and initial set of centroids $\{\mu_1, \ldots, \mu_K\}$

while Not Converged do
  1. Update Cluster Assignments
     for $i \leftarrow 1$ to $n$ do
        Assign $x_i$ to the cluster whose centroid is closest to $x_i$
     end
  2. Update Centroids
     for $k \leftarrow 1$ to $K$ do
        Update centroid $\mu_k$ by finding the empirical mean of the observations in cluster $k$
     end
end

$K$-means clustering is popular due to its simplicity, but some drawbacks exist. First, it is important to note that this optimization procedure depends on the initialization and the algorithm and is not guaranteed to find a globally optimal solution. For this reason, common practice includes repeating the procedure using multiple initializations. Second, $K$-means requires the user to specify the total number of clusters, $K$. Tibshirani et al. (2001), and the references therein, provide a variety of options for choosing $K$. Third, the result of $K$-means clustering is a hard clustering assignment in which each observation is assigned to exactly one cluster, as opposed to a soft clustering procedure that assigns an observation a degree of membership. Unfortunately, procedures which produce hard clustering assignments fail to reflect the uncertainty of the assignments, such as for observations existing near the boundary between two clusters. Fourth, the choice of Euclidean distance as the dissimilarity metric produces spherical clusters, which limits the applicability of $K$-means to cases where the cluster structures are similarly-sized spheres. Finally, $K$-means clustering is susceptible to the presence of anomalous data. Many variations of $K$-means exist, each targeting different aspects of the drawbacks.
2.1.2 Hierarchical Clustering

Hierarchical clustering procedures serve as another method for partitioning \( n \) observations into clusters. Unlike \( K \)-means in which the user specifies the number of clusters in advance, hierarchical clustering procedures produce a sequence of nested clusters formed by repeatedly merging or splitting clusters. At the lowest level of the hierarchy, there are \( n \) clusters, each with a single observation. We refer to a cluster with a single observation as a singleton. At the highest level of the hierarchy, there is a single cluster containing all \( n \) observations. The \( k^{th} \) intermediate level represents a partition formed by merging clusters in the \( (k-1)^{th} \) level. A dendrogram graphically summarizes the complete hierarchical structure and serves as an aid for determining the number of clusters.

There are two common strategies for hierarchical clustering: (1) divisive and (2) agglomerative. Divisive strategies are top-down approaches in which all observations start in the same cluster and a series of splits divides the observations until there are \( n \) singleton clusters. Divisive clustering is uncommon because it is computationally expensive. For instance, an exhaustive search for splitting \( n \) observations into two non-empty subsets once requires consideration of \( 2^{n-1} - 1 \) possible partitions. Since this is infeasible for medium to large datasets, we will focus on the second strategy. Agglomerative clustering strategies are bottom-up approaches, which begin with \( n \) singleton clusters and successively merges the two “closest” clusters until there is a single cluster containing all of the observations.

Agglomerative clustering techniques require the user to specify a method for quantifying the dissimilarity between two clusters and a rule for merging clusters. A variety of merging rules exist, but three of the most common include single linkage, complete linkage, and Ward’s method. In single linkage agglomerative clustering, we merge the two clusters with the smallest dissimilarity between clusters, where the dissimilarity is the distance between the closest pair of points (one point from each cluster). Complete linkage is similar, but the dissimilarity between clusters is given by the maximum distance between a pair of points (one from each cluster). Thus, complete linkage merges the pair of clusters with the minimum maximum distance. In Ward’s method (Ward Jr, 1963), the dissimilarity between two clusters is the increase in the total sum of squares.
resulting from combining the two clusters. Ward’s method merges the clusters that produce the smallest increase in the total sum of squares.

As with any method, hierarchical clustering procedures have strengths and weaknesses. Hierarchical clustering offers greater flexibility than \( K \)-means for discovering anisotropic clusters (Nagy, 1968); however, the computational burden is typically higher and the procedures are only applicable for small to medium sized datasets. Hierarchical clustering also suffers from many of the same drawbacks as \( K \)-means. The hierarchical structures are the result of greedy searches, which are not guaranteed to find globally optimal solutions. Finally, like many of the classical methods, hierarchical clustering procedures are susceptible to the presence of anomalous data (Xu and Wunsch, 2005).

### 2.1.3 Model-based Clustering

Model-based clustering employs a more principled approach than the largely heuristic \( K \)-means and hierarchical clustering techniques. Model-based clustering assumes the observations are independent and identically distributed (i.i.d) realizations from a probability model. Specifically, the model-based clustering approach uses a finite mixture model (FMM), which treats the population as consisting of \( K \) subpopulations, or mixture components, each defined by a probability density function. A \( K \)-component mixture model describing the density of \( x \) is given by:

\[
p(x \mid \phi) = \sum_{k=1}^{K} \pi_k f_k(x \mid \phi_k).
\]

In (2.2), \( \pi_k \), often called the mixture weight or mixing proportion, represents the proportion of the population associated with the \( k^{th} \) mixture component. The \( K \)-vector \( \pi = \{\pi_1, \ldots, \pi_K\}^T \) lies on the \((K - 1)\)-probability simplex, meaning the mixture weights are nonnegative and sum to 1. The probability density function for the \( k^{th} \) component is denoted by \( f_k(\cdot \mid \phi_k) \), where \( \phi_k \) represents the parameters defining the specific distribution. If all of mixture components are from the same parametric family, the parameter values differentiate the components and we shorten the notation \( f_k \) to \( f \). Finally, \( \phi \) is an abbreviated way of denoting the entire set of unknown parameters,
The Gaussian Mixture Model (GMM), a mixture of normal distributions, is the most common example of a mixture model. Figure 2.2a shows the contours for the probability density function of a two-component GMM. The modal heights are different, indicating a difference in the mixture weights. The model in Equation (2.2) is often called a generative model, as we can generate data from the mixture by (1) picking the mixture component with probability equal to the corresponding mixture weight and (2) generating the observation using the density function and its associated parameter vector. Figure 2.2b shows a sample generated in this manner, with the true mean for each component represented by a solid circle. The sample contains more red points than blue, reinforcing the difference in mixture weights.

Figure 2.2: An example of the contours and a sample from a Gaussian mixture model.
idea by treating the mixture components as clusters. Inferences about the model are then used for learning about the assignment of observations to clusters. McNicholas (2016) traces the idea of representing clusters as mixture components to (Tiedeman, 1955) and attributes the first published use of a GMM for model-based clustering to (Wolfe, 1965). McNicholas also credits the review paper by Fraley and Raftery (2002) and their open-source MCLUST software (first appearing in (Fraley and Raftery, 1999)) as one of the reasons for the popularity of model-based clustering. Recent review papers by McNicholas (2016) and Bouveyron and Brunet-Saumard (2014) provide a wealth of information about model-based clustering, with the latter focusing on the issues of model-based clustering in high dimensions.

The use of a probability model as a basis for model-based clustering provides an opportunity for formal probabilistic inferential procedures. Assume we have a set of \( n \) i.i.d. observations generated from the model in Equation (2.2), where the number of clusters, \( K \), is known. Two common approaches for inference are based on maximizing the marginal likelihood of the observed data, for which the Expectation Maximization (EM) algorithm (Dempster et al., 1977) is useful, and a Bayesian approach that explores the posterior distribution of the parameters using Markov chain Monte Carlo (MCMC) methods. Since the CNMM is directly related to the Bayesian approach, we focus on the latter. We introduce a latent variable representing the cluster assignment, \( c_i \), for the \( i^{th} \) observation, where \( c_i \in \{1, \ldots, K\} \). The set of all cluster assignments \( c = \{c_1, \ldots, c_n\} \) is treated as an unknown parameter, along with the set of mixture weights \( \pi = \{\pi_1, \ldots, \pi_K\} \) and the component-specific parameters \( \{\phi_1, \ldots, \phi_K\} \). Specifying a prior distribution for the component-specific parameters, latent assignments, and mixture weights, the model is given by:

\[
\begin{align*}
    x_i \mid c_i, \pi, \{\phi_1, \ldots, \phi_K\} &\sim f_{c_i}(x_i \mid \phi_{c_i}) \\
    \phi_k \mid G_0 &\sim G_0 \\
    c_i \mid \pi &\sim \text{Discrete}(\pi_1, \ldots, \pi_K) \\
    \pi &\sim \text{Dir}(\alpha/K, \ldots, \alpha/K),
\end{align*}
\]

where \( \text{Dir} \) represents the Dirichlet distribution (Section 3.3.1) and \( G_0 \) is a prior distribution for component-specific parameters. In (2.3), a subscript of \( i \) indicates a observation-level value and a
subscript of $k$ indicates a component-level value. Inference on the joint posterior distribution of the unknown parameter follows using MCMC procedures.

The use of formal probabilistic inference procedures in model-based clustering provides several advantages over $K$-means and hierarchical clustering procedures. First, model-based clustering generates soft clustering assignments. Soft assignments provide an observation with a degree of membership in each component, rather than a binary in-out classification. In many cases, the soft assignments represent the probability of the observation belonging to a given component. A hard assignment is easily found from the soft by assigning the observation to the cluster with the highest probability. Second, model-based clustering can leverage the probability model for selecting the number of mixture components, $K$. One common way of approaching this is to run the mixture model for a range of $K$ and compare the results using a model selection technique, such as BIC (Schwarz, 1978), for selecting the number of components. A Bayesian approach places a prior distribution the number of clusters. Despite the advantages of model-based clustering over $K$-means and hierarchical approaches, model-based clustering approaches are still susceptible to anomalous data. In the next section (Section 2.2), we explore some modifications to clustering techniques previously used for combating anomalous data.

2.2 Clustering in the Presence of Anomalies

Most clustering algorithms, unless specifically designed to be robust, are sensitive to the presence of anomalous data. Consequentially, a wide variety of approaches attempt to solve the issue. Much like clustering algorithms themselves, the potential solutions range from heuristic to grounded in probability models.

There are many variations on $K$-means to allow for clustering in the presence of anomalies. Ahmed and Mahmood (2013) propose Outlier Detection and Clustering (ODC) which alternates between performing $K$-means and removing anomalies. After determining the assignments from $K$-means, any observation that is more than $p$ times the average distance from its centroid is flagged as anomalous and removed. ($p > 1$ and “chosen experimentally.”) Once an observation is removed, it never
returns to the analysis. After removing all anomalies, the centroids are updated and observations reassigned to the nearest centroid. The procedure repeats until no more observations are flagged as anomalous. Chawla and Gionis (2013) present another take on $K$-means, called $K$-means--, in which the user inputs the number of components, $K$, and the number of anomalies to be removed, $l$. After performing $K$-means, the distance between each observation and its centroid is computed. The $l$ observations with the largest distances are labeled anomalous and removed. Using the remaining observations, $K$-means is applied until convergence. Gan and Ng (2017) propose $K$-means with Outlier Removal (KMOR). Unlike ODC and $K$-means--, which alternate between clustering and removing anomalies, KMOR designates a cluster for storing the anomalies and assigns them during the clustering stage. The KMOR approaches uses a modified $K$-mean objective function to allow for the possibility of assigning observations to the outlier cluster.

Peel and McLachlan (2000) and Coretto and Hennig (2016) offer principled approaches based on probability models for mitigating the impact of anomalous observations. Mixtures of normal distributions are useful for describing many phenomena and offer computational benefits; however, when observations not well defined by a normal component are assigned to the component, the estimation of the mean and covariance may be adversely affected. Peel and McLachlan (2000) present a mixture of t distributions for handling anomalies and/or observations from long-tailed distributions. The long tails of the t distribution have the ability to dampen the impact of the anomalies when estimating component parameters. Observations in the low density region of a t component are flagged as anomalous. Coretto and Hennig (2016) proposes the “optimally tuned improper maximum likelihood estimator” (OTRIMLE). The OTRIMLE augments a mixture of normals with a “pseudo mixture component” with uniform density for capturing outliers. The approach, inspired by Banfield and Raftery (1993), uses an improper uniform and the authors offer details on optimally tuning its height.

Our methodology, the Cauchy-Net Mixture Model (CNMM) employs some of the same elements as these methods. Specifically, the CNMM uses a probabilistic approach and a special mixture component (the Cauchy-Net) for simultaneously clustering regular observations and capturing anomalies. The CNMM differs by offering a greater level of flexibility though its nonparamet-
ric nature. The opportunity for the number of components to grow with the size of the data is advantageous for clustering applications.
Dirichlet Process Mixture Model

A Dirichlet Process Mixture Model (DPMM) is a powerful Bayesian nonparametric tool for modeling. Beyond possessing the same advantages as the model-based clustering techniques discussed in Section 2.1.3, DPMMs provide additional flexibility by removing the restriction that the user fixes the number of mixture components, $K$, in advance of an analysis. Allowing $K$ to vary is especially useful for clustering applications since we expect the number of clusters to grow as we collect more data. To provide the reader with the necessary foundation, this chapter briefly reviews the premise of Bayesian and Bayesian nonparametric modeling before presenting a thorough review of DPMMs. The backbone of a DPMM is a Dirichlet Process (DP). We give a formal definition of a DP and explore several metaphors for illustrating the nature of realizations from a DP. We conclude with a discussion of a DPMM and summarize two approaches for performing inference.

3.1 Bayesian Modeling

Consider a collection of data $X = \{x_1, \ldots, x_n\}$ generated from an unknown probability distribution. Assume the distribution is characterized by probability density function $f(x_i | \theta)$, which depends on model parameter $\theta$, and that the $x_i$’s are conditionally independent. We frequently
presume the parametric family of the distribution is known. For example, we might conjecture that the data are normally distributed with a spherical covariance matrix, which makes \( f(x_i \mid \theta) \) the density of a multivariate normal distribution with parameter \( \theta = \{\mu, \Sigma = \sigma^2 I\} \). A typical inferential task seeks to use the observed data to learn about the unknown parameter \( \theta \).

A Bayesian approach to inference views \( \theta \) as a random variable. The analysis incorporates a prior distribution, \( f(\theta) \), that reflects the researcher’s prior belief regarding \( \theta \). Using the observed data, \( X \), and the joint density, \( f(X \mid \theta) \), a direct application of Bayes’ rule updates the prior belief. The resulting distribution is called the posterior distribution of \( \theta \) and is given by:

\[
f(\theta \mid X) = \frac{f(X \mid \theta)f(\theta)}{f(X)}, \tag{3.1}
\]

where the joint density function is:

\[
f(X \mid \theta) = \prod_{i=1}^{n} f(x_i \mid \theta),
\]

and \( f(X) \) is the marginal distribution of the data computed using:

\[
f(X) = \int f(X \mid \theta)f(\theta)d\theta.
\]

The inferential goal of learning about \( \theta \) follows from an analysis of the posterior distribution. Unfortunately, due to the difficulty of computing the marginal distribution of the data, the analytic form of the posterior is often unavailable. MCMC approaches offer a solution by using samples generated from the posterior, or a distribution with similar behavior, for approximating characteristics of \( f(\theta \mid X) \).

Finally, we wish to comment on the difference between the joint density and the likelihood function. The joint density, \( f(X \mid \theta) \), lends itself nicely to illustrating the direct application of Bayes’ rule in (3.1). Conditional on \( \theta \), the joint density allows us to make probability statements about \( X \); however, changing the perspective and conditioning on \( X \), the likelihood function, denoted \( L(\theta \mid X) \), allows us to quantify the plausibility of parameter values given the observed data. Since
the prior and posterior distributions are statements about $\theta$, the posterior calculation in (3.1) often replaces the joint density with the likelihood. Being that the expressions for the joint density and the likelihood are the same, replacing the joint density with the likelihood represents a change in the perspective but has no impact on the calculations.

### 3.2 Bayesian Nonparametric Modeling

The difference between parametric and nonparametric modeling techniques lies in an assumption about the size of the parameter space. The phrase “nonparametric modeling” is a misnomer and actually refers to a class of very flexible models that allow the number of parameters to grow with the number of observations, potentially infinitely. In contrast, parametric models employ a more restrictive assumption of a fixed size for the parameter space, regardless of the sample size.

To illustrate the benefit of nonparametric techniques, consider the data described in Section 3.1. Assuming $f(x_i \mid \theta)$ is a multivariate normal density with $\theta = \{\mu, \Sigma = \sigma^2 I\}$, the normality and covariance assumptions determine the size of the parameter space, making this a parametric approach. For data with cluster structure, such as is depicted in Figure 2.2, the assumption of a single normal model is too restrictive. Finite mixture models offer greater flexibility by employing a mixture of $K$ probability distributions, each with its own parameters. The increase in the size of the parameter space is less restrictive; however, the technique remains parametric because the choice of $K$ determines the size of the parameter space. Nonparametric techniques build on the flexibility afforded by increasing the size of the parameter space but place no restriction on the number of parameters. As we collect and incorporate more data into the model, the number of parameters has room to grow. This feature of nonparametric approaches is especially appealing for clustering, where it is reasonable to expect the number of clusters (or alternatively, the number of parameters defining the clusters) to grow as we collect more data.

Bayesian approaches, whether parametric or nonparametric, require a prior distribution for the unknown parameters. Since nonparametric approaches place no limit on the number of parameters, a Bayesian nonparametric approach requires a prior for an infinite dimensional parameter space.
Two of the most popular priors in Bayesian nonparametrics are the Gaussian process (GP), whose realizations are random functions, and the Dirichlet process (DP), whose realizations are random discrete probability distributions. We focus on DPs and explore their usefulness for clustering applications.

### 3.3 Overview of Dirichlet Processes

A DP is a distribution whose realizations are random probability distributions with the characteristic that the marginals of the random probability distribution are Dirichlet distributed. Ferguson (1973) formalized the idea, but advances in computing and MCMC methodologies took DPs from a theoretical construct to a practical tool for Bayesian nonparametric modeling.

Two characteristics of DPs are especially valuable. First, DP realizations are discrete probability distributions with probability one (Blackwell and MacQueen, 1973). This feature makes DPs useful for clustering, where Bayesians leverage them as priors for inducing clustering in cluster and mixture modeling applications. Second, DPs are nonparametric tools that allow the number of parameters to grow with the size of the data. The ability to grow the parameter space is especially important for clustering applications since, reasonably, we could expect the number of clusters to increase as we collect more data. This feature is a major advantage over many of the clustering procedures described in Section 2.1 in which the user must fix the number of clusters prior to running an analysis.

In this section, we review the Dirichlet distribution, provide the formal definition of a DP, and explore a DP’s properties through the stick-breaking and Chinese restaurant process representations.

#### 3.3.1 Dirichlet Distribution

The Dirichlet distribution is a multivariate probability distribution for random $K$-vectors $\mathbf{x} = \{x_1, \ldots, x_K\}$ residing on the $(K - 1)$-probability simplex. The restriction of being on the probability simplex means that the $K$ elements of $\mathbf{x}$ satisfy the following two conditions:
\[ \sum_{i=1}^{K} x_i = 1, \]
\[ x_i \in [0, 1]. \]

As an example, if \( K = 3 \), \( x \in \mathbb{R}^3 \), but the elements are constrained to lie in the 2-dimensional probability simplex (gray triangle) shown in Figure 3.1. The Dirichlet distribution is parameterized by the \( K \)-vector \( \alpha = \{ \alpha_1, \alpha_2, \ldots, \alpha_K \} \), where \( \alpha_i > 0 \), and we define the concentration (also called the scale) as \( \alpha_0 \equiv \sum_{i=1}^{K} \alpha_i \). If \( x \) is distributed according to a Dirichlet distribution with parameter \( \alpha \), then we write \( x \sim \text{Dir}(\alpha) \) and the probability density function of \( x \) is given by:

\[
p(x \mid \alpha) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \prod_{i=1}^{K} x_i^{\alpha_i-1},
\]

where \( \Gamma \) is the gamma function.

\[ (0, 1, 0) \]
\[ (0, 0, 1) \]
\[ (1, 0, 0) \]

\textbf{Figure 3.1:} The 2-probability simplex (gray triangle) consists of 3-dimensional points constrained to be nonnegative and sum to 1.

The Dirichlet distribution is related to several other important probability distributions. For instance, a beta distribution is a special case of the Dirichlet distribution. To see the connection,
consider a Dirichlet distribution with $K = 2$. Using the fact that $x_1 + x_2 = 1$, Equation (3.2) becomes:

$$p(x \mid \alpha) = \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} x_1^{\alpha_1-1}(1-x_1)^{\alpha_2-1}.$$

This is equivalent to saying that $x_1 \sim \text{Beta}(\alpha_1, \alpha_2)$. Thus, the Dirichlet distribution serves as the multivariate extension of a beta distribution. We will see in the next section that the Dirichlet distribution also plays a role in defining a Dirichlet process (DP) and we can think of a DP as an infinite dimensional Dirichlet distribution.

Both the beta and Dirichlet distributions serve as useful prior distributions in Bayesian analyses. The beta distribution is commonly used for modeling probabilities and is the conjugate prior for the binomial success probability. When we generalize the binomial situation to allow for $K \geq 2$ outcomes, the binomial success probability becomes a vector of probabilities associated with each outcome. This probability vector lies on the $(K-1)$-probability simplex and represents the parameter for a multinomial distribution. As the beta distribution is conjugate for the binomial, the Dirichlet distribution is conjugate for the multinomial distribution.

### 3.3.2 Dirichlet Process

A DP (Ferguson, 1973) is a distribution over random probability distributions. A DP is parameterized by a positive scalar known as the concentration parameter (also called the scale, strength, precision, or inverse variance) and a probability distribution, called the base measure. We denote the concentration and base measure of a DP as $\alpha_0$ and $G_0$, respectively.

The defining feature of a random probability distribution distributed according to a DP lies in its marginal distributions. Consider a measurable space $(\Omega, B)$, where $\Omega$ represents the sample space and $B$ is a $\sigma$-algebra on $\Omega$. (A $\sigma$-algebra on set $\Omega$ is a nonempty collection of subsets of $\Omega$ such that the collection contains $\Omega$, is closed under complement, and is closed under countable unions.) Suppose that $B = \{B_1, \ldots, B_m\}$ is a finite $m$-partition of $\Omega$, $G$ is a random probability distribution over $(\Omega, B)$, $G_0$ is the base measure over $(\Omega, B)$, and $\alpha_0$ is the concentration. If, for
every value of \( m \) in the positive integers and every finite \( m \)-partition of \( \Omega \), the random probability vector \((G(B_1), \ldots, G(B_m))\) is distributed according to the following Dirichlet distribution:

\[
(G(B_1), \ldots, G(B_m)) \sim \text{Dir}(\alpha_0 G_0(B_1), \ldots, \alpha_0 G_0(B_m)),
\]

we say \( G \) is distributed according to the DP with base measure \( G_0 \) and concentration \( \alpha_0 \) and write \( G \sim \text{DP}(\alpha_0, G_0) \). In such cases, we say \( G \) is a DP.

The key to understanding the usefulness of DPs lies in the nature of the samples from a DP. The formal definition of a DP provides limited insight into the nature of \( G \), which is a realization or sample from the DP, including characteristics of \( G \), methods for generating \( G \), and the roles of the base measure and the concentration parameter.

To better understand the nature of \( G \), consider an example in which the base measure, \( G_0 \), is the standard normal distribution. Using \( G_0 = N(0, 1) \), Figure 3.2 represents two randomly generated realizations for each of four settings of the concentration \( \alpha_0 \). Each subplot is based on 1000 draws and the point masses correspond to values randomly drawn from \( G_0 \). The height of each point mass indicates the relative frequency of times that mass was selected during the 1000 draws. The unlabeled vertical axis showing relative frequencies are comparable within a row but not between rows.

Figure 3.2 emphasizes four important features of a DP. First, the realizations of a DP are random probability distributions, as illustrated through the columns. For instance in the \( \alpha_0 = 1 \) row, there are five obvious point masses in the first realization but only three in the second. Further, the point masses are at different locations. The second important feature is that the realizations are discrete probability distributions with probability one (Ferguson, 1973; Blackwell and MacQueen, 1973). The discrete nature is evident from the fact that only a finite number of values from \( G_0 \) are selected in each realization and, thus, associated with positive probability. This discrete nature leads to the next important feature: the DP induces a clustering effect. All draws associated with the same point mass may be viewed as a cluster. Finally, examining the rows reveals that as \( \alpha_0 \) increases, so do the number of clusters. Thus, the DP parameters play a critical role in shaping the output of a DP.
In summary, realizations of a DP are random discrete probability distributions. The discrete nature of realizations from a DP make the DP useful for inducing a clustering effect. Additionally, the probabilistic nature of DPs make them a flexible tool for Bayesian modeling, especially in the context of mixture modeling. As we will see in Section 3.4, DPs are easily extended to infinite mixture models, which offer the flexibility of mixture modeling, without the restriction of using a fixed number of clusters, $K$. 
Figure 3.2: Realizations of random probability distributions generated according to a Dirichlet process with a standard normal base measure, $G_0$. The rows correspond to different values of the concentration parameter, $\alpha_0$, and the columns represent two independent realizations.
3.3.3 Stick-breaking Representation

The formal definition of a DP fails to offer insight as to how to generate a random probability distribution, \( G \), satisfying (3.3). The stick-breaking representation of a DP (Sethuraman, 1994) serves as an alternative definition that simultaneously provides a procedure for generating realizations of a DP and illustrates key features of those realizations.

The stick-breaking representation uses two countably infinite sets for generating a random probability distribution from a DP with concentration parameter \( \alpha_0 \) and base measure \( G_0 \). The set \( \varphi = \{ \phi_1, \phi_2, \ldots \} \) consists of independent draws from the base measure, \( G_0 \), with each draw representing the location of a point mass. The other set, \( \nu = \{ v_1, v_2, \ldots \} \), consists of independent draws from a Beta\((1, \alpha_0)\) distribution and is used for constructing the weight associated with each point mass in \( \varphi \). We constrain the point mass weights to sum to 1 and denote the set of weights as \( \pi = \{ \pi_1, \pi_2, \ldots \} \). The point mass weights are given by:

\[
\begin{align*}
\pi_1 &= v_1, \\
\pi_j &= v_j \prod_{j=1}^{j-1} (1 - v_j) \quad \text{for } j \in \{2, 3, \ldots \}. 
\end{align*}
\]

(3.4)

To better understand the formula in (3.4), consider a stick of unit length, as shown in the top level of Figure 3.3. In general, the random variable \( v_i \) represents the proportion of the stick’s remaining length to be broken off at time \( i \). The actual length of the piece broken off \( (\pi_i) \) is the product of the proportion \( (v_i) \) and the length of the stick prior to the break. In Figure 3.3, the piece broken off at each time is shaded and the calculation of its length shown above the stick. Finally, the length of the piece remaining after the break is the product of the proportion we did not use \( (1 - v_i) \) and the length of the stick prior to the break. The calculation of the length remaining is above the unshaded portion. At time 1, the stick begins with length 1 (labeled in right margin). The proportion of the remaining stick we wish to break off is \( v_1 \). We break off the piece (shaded) of length \( \pi_1 = v_1 \cdot 1 \). The length of the remaining piece is \( (1 - v_1) \cdot 1 \), which carries over as the length at the start of the next level. At time 2, the remaining length of the stick is \( (1 - v_1) \) and we break off \( v_2 \) of that. Thus, we break off a piece of length \( \pi_2 = v_2 \cdot (1 - v_1) \). The length of the remaining piece after the break is \( (1 - v_2) \cdot (1 - v_1) \), which carries over to the start of the next level. The stick-breaking continues...
and the end result is a countably infinite set of weights, $\pi$, whose sum is 1. As we can see from the bottom level of Figure 3.3, it only takes a small number of weights to account for nearly the entire length of the stick.

![Figure 3.3: A representation of the stick-breaking weights.](image)

Using the point masses, $\phi$, and the corresponding weights, $\pi$, produced by this stick-breaking procedure, Sethuraman (1994) constructed an infinite mixture of point masses given by:

$$G = \sum_{k=1}^{\infty} \pi_k \delta(\phi_k),$$

where $\delta(\phi_k)$ is a degenerate distribution with all of its mass at $\phi_k$. He showed that this random probability distribution is distributed according to a $DP(\alpha_0, G_0)$.

The nature of $G$ provides several important insights. First, random probability distributions from a DP are discrete with probability one (Ferguson, 1973; Blackwell and MacQueen, 1973). To
intuitively see this, consider repeatedly sampling from $G$:

$$\theta_i \sim G \quad \text{for} \quad i \in \{1, 2, \ldots \}.$$ 

Since $G$ is a mixture model, this requires sampling a component $k$ from the mixture weights, $\pi$, and sampling from the corresponding distribution, $\delta(\phi_k)$. Most of the weights will be practically 0 because only a few of the countably infinite number weights account for most of the probability mass. Consequentially, only a few mixture components have a non-negligible chance of being selected. This, combined with the fact that sampling from a point mass is deterministic given the component, implies that for a sufficiently large number of samples from $G$, the sampled values will repeat. For a continuous distribution, samples with the same value occur with probability zero. Thus, $G$ must be discrete. The second important insight relates to an implication of the discrete nature of $G$. Observations, in this case $\{\theta_i\}$, drawn from the same mixture component share the same value. This induces a clustering effect. If $\theta_i = \theta_j$ for $i \neq j$, then observations $i$ and $j$ are in the same cluster. This feature makes DPs a popular tool for clustering and mixture model applications.

The stick-breaking representation also helps to illustrate the effect of the concentration parameter, $\alpha_0$. The value of $\alpha_0$ impacts the number of clusters, with larger values producing more clusters. Consider the expectation of $v_i \sim Beta(1, \alpha_0)$:

$$E[v_i] = \frac{1}{1 + \alpha_0}.$$ 

If $\alpha_0 = 1$, we would break off 50% of the length remaining of the stick at each step, on average. This produces the expected sequence of weights:

$$E[\pi_1] = 0.5,$$
$$E[\pi_2] = 0.5(1 - 0.5) = 0.25,$$
$$E[\pi_3] = 0.5(1 - 0.5 - 0.25) = 0.125,$$
$$\vdots$$

Applying similar reasoning, for $\alpha_0 = 4$, we break off, on average, 20% of the length remaining at
each step, resulting in the expected sequence of weights:

\[
\begin{align*}
\mathbb{E}[\pi_1] &= 0.2 \\
\mathbb{E}[\pi_2] &= 0.2(1 - 0.2) = 0.16 \\
\mathbb{E}[\pi_3] &= 0.2(1 - 0.2 - 0.16) = 0.128 \\
&\quad \vdots
\end{align*}
\]

From this, we can see that when the concentration parameter is larger, the weights accumulate more slowly, requiring more breaks to account for nearly all of the weight. The implication is that more point masses have non-negligible weights, leading to a greater number of clusters.

Finally, we generalize the expected lengths in the stick-breaking process with \( \alpha_0 \). The expected length of the \( i^{th} \) piece is:

\[
\mathbb{E}[\pi_i] = \alpha_0^{i-1} \cdot (\mathbb{E}[v])^i,
\]

where \( v \sim Beta(1, \alpha_0) \). We apply (3.6) for generating data in the simulation studies in Chapter 5.

### 3.3.4 Chinese Restaurant Process Representation

The Chinese Restaurant Process (CRP) (Aldous, 1985) provides another metaphor for illustrating the nature of DPs. A CRP, parameterized by the concentration parameter \( \alpha_0 \) and denoted \( CRP(\alpha_0) \), is a distribution over partition structures of the positive integers. Partitioning integers induces a clustering in that integers in the same subset may be viewed as elements with a shared cluster assignment.

The CRP metaphor considers seating a sequence of \( n \) customers in a Chinese restaurant consisting of an infinite number of tables, each with infinite capacity. The integers \( \{1, \ldots, n\} \) index the order in which customers are seated. The process creates a partition by assigning customers from the same table to the same subset of the partition, or alternatively, to the same cluster. The assignment of customers to tables is stochastic and based on the following procedure. Since the restaurant is initially empty, the first customer goes to a previously unoccupied table, now called Table 1, and receives the cluster label \( c_1 = 1 \). The second customer joins the first customer at Table 1
with probability $\frac{1}{\alpha_0+1}$ or starts a new table with probability $\frac{\alpha_0}{\alpha_0+1}$. For the sake of illustration, suppose the second customer begins Table 2 (i.e., $c_2 = 2$). The third customer joins Table 1 with probability $\frac{1}{\alpha_0+2}$, joins Table 2 with probability $\frac{1}{\alpha_0+2}$, or starts a new table, with probability $\frac{\alpha_0}{\alpha_0+2}$.

After seating $n-1$ customers, the restaurant consists of $K$ active tables labeled $\{1, \ldots, K\}$. We denote the number of customers at the $k$th table as $n_k$. The $n$th customer either joins one of the active tables or starts Table $K + 1$ according to the following probabilities:

$$Pr(\text{Customer } n \text{ joins existing Table } k \mid c_{1: (n-1)}) = \frac{n_k}{\alpha_0 + n - 1},$$

$$Pr(\text{Customer } n \text{ starts new Table } K + 1 \mid c_{1: (n-1)}) = \frac{\alpha_0}{\alpha_0 + n - 1},$$

(3.7)

where $c_{1: (n-1)}$ represents the set of cluster labels for the $n-1$ customers previously seated. After seating the $n$th customer, the state of the restaurant represents a partition of the integers $\{1, \ldots, n\}$ or a clustering of the $n$ customers.

The CRP exhibits several important features. First, (3.7) clearly illustrates the influence of the concentration parameter, $\alpha_0$, on the number of clusters in a CRP. As in a DP, the larger the value of $\alpha_0$, the greater the probability of opening a new cluster. The second feature, also related to (3.7), is the “rich-get-richer” property. The probability of joining cluster $k$ is proportional to the number of observations in the cluster, $n_k$. As $n_k$ increases, the next customer is more likely to join, creating an attraction effect. Finally, realizations from a CRP are exchangeable. To see this informally, consider the example realization in Figure 3.4 containing three clusters (tables) with sizes 5, 2, and 3, respectively. The corresponding partition is $\{\{1, 4, 6, 7, 10\}, \{2, 3\}, \{5, 8, 9\}\}$ with the set of cluster labels $c = \{c_1 = 1, c_2 = 2, c_3 = 2, c_4 = 1, c_5 = 3, c_6 = 1, c_7 = 1, c_8 = 3, c_9 = 3, c_{10} = 1\}$.

The probability of the partition structure $c = \{1, 2, 2, 1, 3, 1, 1, 3, 3, 1\}$ is:

$$p(c_{1:10}) = p(c_1) \cdot p(c_2 \mid c_1) \cdot p(c_3 \mid c_{1:2}) \cdots p(c_{10} \mid c_{1:9})$$

$$= \frac{\alpha_0}{\alpha_0 + 1} \cdot \frac{\alpha_0}{\alpha_0 + 2} \cdot \frac{1}{\alpha_0 + 3} \cdot \frac{\alpha_0}{\alpha_0 + 4} \cdot \frac{2}{\alpha_0 + 5} \cdot \frac{3}{\alpha_0 + 6} \cdot \frac{1}{\alpha_0 + 7} \cdot \frac{2}{\alpha_0 + 8} \cdot \frac{4}{\alpha_0 + 9}.$$  

(3.8)

Now, suppose $\sigma(c_{1:10})$ represents a permutation of the labels, maintaining five elements assigned
to Table 1, two assigned to Table 2, and three assigned to Table 3. When computing $p(\sigma(c_{1:10}))$, the denominators do not change. The order of the numerators changes, but the product remains the same. For three tables of size 5, 2, and 3, we still need $\alpha_0^3$ (to open the three tables) and $4!, 1!, \text{and } 2!$ to represent adding a $5^{th}$ observation to a cluster of size 4, a $2^{nd}$ observation to a cluster of size 1, and a $3^{rd}$ observation to a cluster of size 2. Since permuting the cluster labels does not change the joint probability, data generated from a CRP are exchangeable. As we will see in Section 3.5, this fact plays an important role for MCMC procedures for DPs.

![Chinese Restaurant process example](image)

**Figure 3.4:** Chinese Restaurant process example showing the assignments of the first 10 customers (white squares) to tables (large, shaded circles). Tables with a solid outline represent active tables or clusters. Empty tables have a dashed outline.

There are obvious connections between a $\text{CRP}(\alpha_0)$ and $\text{DP}(\alpha_0, G_0)$, especially due to the discrete nature of realizations from both processes inducing clustering. The connection between a CRP and a DP becomes even clearer by including an additional step in the CRP. Each time a new table opens, a parameter is randomly drawn from the base measure, $G_0$, and assigned to the table, as represented in Figure 3.5. The table parameter applies to all customers at the table and, metaphorically, represents the first person at the table ordering a shared plate for everyone. The result of the modified CRP in Figure 3.5 is the discrete probability distribution in Table 3.1. Beyond the commonalities of producing random discrete probability distributions, inducing clustering, and serving as a distribution over partition structures, Blackwell and MacQueen (1973) formalized the connection between the CRP and the DP. The modified version of the CRP is equivalent to a Pólya
Figure 3.5: Modified Chinese restaurant process example showing the assignments of the first 10 customers (white squares) to tables (large, shaded circles). Tables with a solid outline represent active tables or clusters. Empty tables have a dashed outline. The table parameter, drawn from the base measure, applies to all customers at the table.

Table 3.1: Random discrete probability distribution generated by a CRP.

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$\theta_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability</td>
<td>0.5</td>
<td>0.2</td>
<td>0.3</td>
</tr>
</tbody>
</table>

urn scheme in which the base measure in the Pólya urn scheme represents a continuum of colors. They proved that the distribution over colors in the a Pólya urn scheme converges to a DP as $n \to \infty$, provided the draws from the base measure are independent. Due to the equivalency of the CRP and Pólya urn scheme, the same result holds for the CRP.

3.4 Dirichlet Process Mixture Model

The discrete nature of realizations from a DP is useful for inducing clustering; however, the discreteness prevents DPs from directly serving as a model for continuous data. To address this issue, Antoniak (1974) extended DPs to the Dirichlet Process Mixture Model (DPMM). DPMMs leverage the nonparametric nature of DPs to allow the model complexity, as represented by the number
of mixture components, to grow with the data. The result is a countably infinite mixture model that
overcomes the limitation of needing to fix the number of components.

A finite mixture model (Section 2.1.3) models independent and identically distributed observations
\{x_1, \ldots, x_n\} by treating the distribution of \(x_i\) as a mixture of \(K\) components, each component
described by a parametric distribution. We represent the \(K\)-component mixture distribution for \(x_i\) as:

\[
p(x_i \mid \vartheta) = \sum_{k=1}^{K} \pi_k f_k(x_i \mid \theta_k),
\]

where the \(k^{th}\) component’s mixture weight (or mixing proportion), distribution over observations,
and component parameters are \(\pi_k\), \(f_k\), and \(\theta_k\), respectively. The mixture components may have dif-
ferent distributions, but for the sake of notational convenience, we assume that all components are
of the same parametric family and drop the subscript on \(f_k\). The entire set of unknown parameters,
consisting of the mixture weights and distribution specific parameters for all mixture components,
is denoted by \(\vartheta\). One weakness of finite mixture modeling is the need to fix the number of compo-
nents in advance of the analysis. Practitioners often run the model for a range of \(K\) and use model
selection techniques, such as BIC, for selecting the number of components.

A DPMM removes the finite mixture model’s restriction of specifying \(K\) by allowing the number
of mixture components to grow with the number of observations. The nonparametric approach
produces a countably infinite mixture model given by:

\[
p(x_i \mid \vartheta) = \sum_{k=1}^{\infty} \pi_k f(x_i \mid \theta_k).
\]

As a Bayesian approach, a DPMM uses a \(DP(\alpha_0, G_0)\) as the joint prior distribution for the un-
known mixture weights and component parameters. The stick-breaking representation of a DP
clearly illustrates the rationale for this choice of prior. Recall that the random distribution gener-
ated according to the stick-breaking process:
is distributed according to a $DP(\alpha_0, G_0)$. Using $G$ as the mixing distribution over the component parameters, sampling from the countably infinite set of point masses, $\{\phi_1, \phi_2, \ldots\}$, most of which have 0 weight, induces repeated values in $\{\theta_1, \ldots, \theta_n\}$. Thus, the DP prior naturally induces clustering, without fixing the number of mixture components, making it an ideal tool for mixture modeling. We represent the DPMM hierarchically as:

$$
\begin{align*}
G | \alpha_0, G_0 &\sim DP(\alpha_0, G_0). \\
\theta_i &| G \sim G \\
x_i | \theta_i &\sim f(x_i | \theta_i)
\end{align*}
$$

This representation emphasizes that the distribution of $x_i$ is a mixture of distributions of the form $f$. The parameter $\theta_i$ determines the component of $x_i$ and observations for which $\theta_i = \theta_j$ are in the same cluster. Although not explicitly shown in the model, the sampling of the component parameters from $G$ reflects the mixture weights inherent in $G$.

The model depicted in (3.10) is a Bayesian hierarchical model (top-down view) or a data generating model (bottom-up view). The top-down perspective is useful for performing inference and is discussed in Section 3.5. A schematic representation of the bottom-up perspective, as shown in Figure 3.6, emphasizes the flow of logic for generating data according to the DPMM in (3.10). Using the stick-breaking representation, the data generating model first constructs a countably infinite set of point masses $\{\phi_1, \phi_2, \ldots\}$ from the continuous distribution $G_0$ and a countably infinite set of mixture weights $\{\pi_1, \pi_2, \ldots\}$ from $\alpha_0$. In the schematic, the value (in red) in the bottom right corner of a box represents the number of values generated. Due to the fact that only a small number of weights are non-negligible, $G$ is a discrete probability distribution, despite $G_0$’s continuous nature. Using $G$, we sample a parameter value, $\theta_i$. Since $G$ is discrete, this is equivalent to choosing one of the point masses with probability from the associated mixture weight. Conditional on $\theta_i$, the distribution is then used to generate the continuous value $x_i$. For
this reason, we often refer to \( f(x_i \mid \theta_i) \) as the data model. In total, the steps in the bottom box are repeated \( n \) times to produce the \( n \) observations.

Figure 3.6: Schematic of DPMM data generating model based on the stick-breaking representation of a DP.

The stick-breaking construction provides an intuitive perspective but is not the only appropriate metaphor for working with DPMMs. For instance, integrating over \( G \) in (3.10) has the effect of drawing the \( \theta_i \)’s directly from the prior \( DP(\alpha_0, G_0) \). This representation is shown in Figure 3.7. After the customer’s table is assigned a parameter value from \( G_0 \), the parameter value is fed into the distribution to generate the customer’s continuous observation (depicted as small, shaded circle).

DPMMs, like FMMs, offer a more principled approach than the heuristic methods of \( K \)-means and hierarchical clustering. The assumption that a probability distribution governs the behavior of observations allows for the use of formal inferential procedures, including the opportunity to express the level of certainty for the clustering assignments. Due to the removal of the restriction
that \( K \) is fixed in advance of the procedure, DPMMs offer expanded flexibility beyond FMMs. For instance, when using DPMMs to predict the cluster membership of a new observation, the ability to open a new cluster means that the new observation is not necessarily forced into one of the existing clusters. Unfortunately, the effectiveness of DPMMs suffer in the presence of anomalous data.

\section*{3.5 Dirichlet Process Mixture Model Inference}

Recent advances in computing and MCMC methodologies generated traction for DPMMs as practical modeling tools (Neal, 2000). Inference on a DPMM uses the posterior distribution of the parameters defining the mixture components and/or the posterior distribution of the component labels. MCMC techniques, such as Gibbs sampling (Gelfand and Smith, 1990), provide a mechanism for exploring such posterior distributions through the use of sampling. In this section, we explain the relevant notation and summarize several MCMC algorithms for performing inference.
on DPMMs. As the algorithms are notation intensive, Table 3.2 contains a brief description the notation for easy reference.

### 3.5.1 Review of Notation for Dirichlet Process Mixture Model Inference

Suppose we wish to model a collection of independent observations \( X = \{x_1, \ldots, x_n\} \) using a mixture model with an unknown number of mixture components. Assuming all of the components belong to the same parametric family characterized by probability density \( f(x_i \mid \theta_i) \), the parameter values defining each density distinguish the components. Let \( \theta_i \) and \( \theta = \{\theta_1, \ldots, \theta_n\} \) represent the parameter associated with observation \( i \) and the collection of parameter values for all observations, respectively. Using mixing distribution \( G \) from a DP as the prior on the the parameter values, a DPMM, shown in (3.10) and repeated here for convenience, is given by:

\[
\begin{align*}
    x_i \mid \theta_i &\sim f(x_i \mid \theta_i) \\
    \theta_i \mid G &\sim G \\
    G \mid \alpha_0, G_0 &\sim DP(\alpha_0, G_0).
\end{align*}
\]

Sampling parameter values from the discrete mixing distribution \( G \) generates repeated values in \( \theta \) and induces a clustering effect.

The information contained in \( \theta \) is often represented through a collection of latent component labels, \( c \), and a collection of unique parameter values, \( \phi \). Assuming \( \theta \) contains \( K \) unique values, we denote the collection of unique parameter values as \( \phi = \{\phi_1, \ldots, \phi_K\} \), each of which defines one of the \( K \) mixture components. The subscript on each element of \( \phi \) serves as an indicator of the component, e.g., \( \phi_k \) the parameter defining the \( k^{th} \) component. Similarly, we use a latent component label for observation \( i \), denoted \( c_i \), for indicating component membership. If \( c_i = k \), then observation \( i \) belongs to the \( k^{th} \) component and \( \theta_i = \phi_k \). We summarize the relationships between \( c_i, \theta_i, \) and \( \phi_{c_i} \), using the rule:
Based on the rule in (3.11), the parameter values induce a clustering effect because observations sharing a parameter value also share a component label. Supposing observations $i$ and $j$ share a parameter value, say $\phi_k$, then $\theta_i = \theta_j = \phi_k$ and the observations belong to the same component because $c_i = c_j = k$. Expressing the collection of all component labels \{c_1, \ldots, c_n\} as $c$, useful summaries related to $c$ include the number of observations assigned to the $k^{th}$ component (denoted $n_k$), the set of indices for the observations assigned to the $k^{th}$ component ($I_k$), and the collection of data assigned to the $k^{th}$ component ($X_k$). To illustrate the notation, consider a scenario in which the current state is given by:

\[
\theta = \{\theta_1 = 2.1, \theta_2 = -0.7, \theta_3 = 2.1, \theta_4 = 0.3, \theta_5 = 2.1, \theta_6 = 0.3\},
\]

where the subscript is the observation number. The information contained in $\theta$ is equivalent to:

\[
K = 3,
\]

\[
c = \{c_1 = 3, c_2 = 2, c_3 = 3, c_4 = 1, c_5 = 3, c_6 = 1\},
\]

\[
\phi = \{\phi_1 = 0.3, \phi_2 = -0.7, \phi_3 = 2.1\},
\]

\[
\{n_1, n_2, n_3\} = \{2, 1, 3\}.
\]

It is important to recognize that the values of the component labels are significant for telling us which observations belong together, but the numerical values themselves are arbitrary. For instance, in the above example, the component labels indicate that observations 1, 3, and 5 share a component, 4 and 6 share a component, and 2 is a singleton. Alternatively, we could represent the same clustering using the labels $c = \{c_1 = 1, c_2 = 2, c_3 = 1, c_4 = 3, c_5 = 1, c_6 = 3\}$. The only consequence of doing so is that the elements indexed by $c_i$ require reordering, namely $\phi = \{\phi_1 = 2.1, \phi_2 = -0.7, \phi_3 = 0.3\}$ and $\{n_1, n_2, n_3\} = \{3, 1, 2\}$.

Finally, the Gibbs sampling procedures in Section 3.5.2 require sequentially removing an observation from its current component and using the resulting full conditional posterior distribution for
making an update. We denote the removal, which we also refer to as the deletion, of observation
\(i\) using a superscript of \(-i\). For instance, the summary \(c^{-i}\) represents the collection of component
labels remaining when excluding observation \(i\). Removing an observation complicates the nota-
tion and also potentially impacts the summaries in two ways. First, removing \(i\) from its current
component potentially changes the value of the summaries. An an example, \(K^{-i}\) represents the
number of components associated with at least one observation, or the number of unique param-
ter values, when excluding observation \(i\). If observation \(i\) represents a singleton component, then
\(K^{-i} = K - 1\); otherwise \(K^{-i} = K\). Second, when observation \(i\) corresponds to a singleton,
its removal creates an empty component and generates some bookkeeping issues. For the sake of
the algorithms presented in the next section, we want the unique labels contained in \(c^{-i}\) to range
from 1 to \(K^{-i}\), without any gaps. When \(i\) represents a singleton, its component label is absent
from among the unique labels in \(c^{-i}\). We denote the numerical value of the missing label as \(L\)
(i.e., \(c_i = L\) when \(i\) is a singleton). When gaps exist, we recommend the following procedure for
“reindexing” to remove the gap:

1. Store the missing label as \(L\) (\(c_i = L\)).
2. For \(j \in \{1 : n\} \setminus \{i\}\), if \(c_j^{-i} > L\), then \(c_j^{-i} = c_j^{-i} - 1\).
3. Using the adjusted labels in \(c^{-i}\), adjust the indices in \(\phi^{-i}\) and \(\{n_k^{-i}\}\) so there are no gaps.

It is worth emphasizing that the reindexing procedure is only necessary when removing an ob-
servation from a singleton component. We illustrate the reindexing procedure using the previous
scenario based on the original component labels. Removing observation \(i = 2\) produces the fol-
lowing summaries:

\[
\theta^{-2} = \{\theta_1^{-2} = 2.1, \theta_2^{-2} = ?, \theta_3^{-2} = 2.1, \theta_4^{-2} = 0.3, \theta_5^{-2} = 2.1, \theta_6^{-2} = 0.3\}
\]
\(K^{-2} = 2\)
\(c^{-2} = \{c_1^{-2} = 3, c_2^{-2} = ?, c_3^{-2} = 3, c_4^{-2} = 1, c_5^{-2} = 3, c_6^{-2} = 1\}\)
\(\phi^{-2} = \{\phi_1^{-2} = 0.3, \phi_2^{-2} = ?, \phi_3^{-2} = 2.1\}\)
\(\{n_1^{-2}, n_2^{-2}, n_3^{-2}\} = \{2, 0, 3\}\).
The current unique labels in $c^{-2}$ are $\{1, 3\}$, which do not align with the values in $1$ to $K^{-2}$, specifically $\{1, 2\}$. Due to the gap in $\{1, 3\}$ at $2$, we realign the labels using the reindexing procedure with $L = 2$. Applying the procedure decreases the labels larger than $L$ by $1$ resulting in:

$$\theta^{-2} = \{\theta_1^{-2} = 2.1, \theta_2^{-2} = ?, \theta_3^{-2} = 2.1, \theta_4^{-2} = 0.3, \theta_5^{-2} = 2.1, \theta_6^{-2} = 0.3\}$$

$$K^{-2} = 2$$

$$c^{-2} = \{c_1^{-2} = 2, c_2^{-2} = ?, c_3^{-2} = 2, c_4^{-2} = 1, c_5^{-2} = 2, c_6^{-2} = 1\}.$$  

Observations $1, 3,$ and $5$, which were originally associated with the only label larger than $L$, are the only labels affected. Finally, using the adjusted values in $c^{-2}$, we also adjust the indices in $\phi^{-2}$ and the component counts to produce:

$$\phi^{-2} = \{\phi_1^{-2} = 0.3, \phi_2^{-2} = 2.1\}$$

$$\{n_1^{-2}, n_2^{-2}\} = \{2, 3\}.$$  

We now have two components (i.e., $K^{-2} = 2$), two unique labels in $c^{-2}$ corresponding to the values $1$ to $K^{-2}$, two unique parameters defining the components in $\phi^{-2}$ indexed $1$ to $K^{-2}$, and two component counts indexed $1$ to $K^{-2}$. If we had chosen to remove any other value of $i$, reindexing would be unnecessary.
Table 3.2: Notation for DPMM inference.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>number of observations</td>
<td>$n \in \mathbb{Z}^+$</td>
</tr>
<tr>
<td>$n$</td>
<td>collection of component counts</td>
<td>${n_1, \ldots, n_K}$</td>
</tr>
<tr>
<td>$i$ and $j$</td>
<td>general indices for observations</td>
<td>$i, j \in {1, \ldots, n}$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>$i^{th}$ observation</td>
<td>$x_i \in \mathbb{R}^p$</td>
</tr>
<tr>
<td>$c_i$</td>
<td>component label for $i^{th}$ observation</td>
<td>$c_i \in {1, \ldots, K}$</td>
</tr>
<tr>
<td>$c$</td>
<td>collection of component labels</td>
<td>$c = {c_1, \ldots, c_n}$</td>
</tr>
<tr>
<td>$c^{-i}$</td>
<td>collection of component labels excluding $i^{th}$ observation</td>
<td>$c^{-i} = {c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n}$</td>
</tr>
<tr>
<td>$K$</td>
<td>number of components with at least one observation</td>
<td>$K \in \mathbb{Z}^+$</td>
</tr>
<tr>
<td>$k$</td>
<td>general index for mixture component</td>
<td>$k \in {1, \ldots, K}$</td>
</tr>
<tr>
<td>$K^{-i}$</td>
<td>number of components with at least one observation excluding the $i^{th}$</td>
<td>$K^{-i} \in {K - 1, K}$</td>
</tr>
<tr>
<td>$\theta_i$</td>
<td>parameter associated with $i^{th}$ observation</td>
<td></td>
</tr>
<tr>
<td>$\theta$</td>
<td>collection of parameter values</td>
<td>$\theta = {\theta_1, \ldots, \theta_n}$</td>
</tr>
<tr>
<td>$\theta^{-i}$</td>
<td>collection of parameter values excluding the $i^{th}$</td>
<td>$\theta^{-i} = {\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n}$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>collection of unique parameter values</td>
<td>$\phi = {\phi_1, \ldots, \phi_K}$</td>
</tr>
<tr>
<td>$\phi^{-i}$</td>
<td>collection of unique parameter values excluding the $i^{th}$</td>
<td>$\phi^{-i} = {\phi_1, \ldots, \phi_{K^{-i}}}$</td>
</tr>
<tr>
<td>$n_k$</td>
<td>number of observations in component $k$</td>
<td></td>
</tr>
<tr>
<td>$n_k^{-i}$</td>
<td>number of observations in component $k$ when excluding the $i^{th}$ observation</td>
<td></td>
</tr>
<tr>
<td>$n^{-i}$</td>
<td>collection of component counts excluding $i^{th}$ observation</td>
<td>${n_1^{-i}, \ldots, n_K^{-i}}$</td>
</tr>
</tbody>
</table>
3.5.2 Algorithms for Dirichlet Process Mixture Model Inference

A DPMM is Bayesian hierarchical model. The joint posterior distribution of the model’s parameters and/or the latent component labels serves as the foundation for inference. We discuss the prior and posterior distributions for a DPMM and conclude with two Gibbs sampling approaches for performing posterior inference. The first approach (West et al., 1994) requires that the base measure, \( G_0 \), is conjugate to the data model, \( f(x_i | \theta_i) \), whereas the second approach (Neal, 2000) applies to both the conjugate and non-conjugate cases.

The DPMM shown in (3.10) places priors on the mixing distribution \( G \) and the parameter values \( \theta \). Blackwell and MacQueen (1973) integrate over \( G \) leading to the Pólya urn predictive rule for the next parameter value, conditional on the previous parameter values:

\[
\theta_n | \theta_1, \ldots, \theta_{n-1} \sim \frac{\alpha_0}{\alpha_0 + n - 1} G_0 + \frac{1}{\alpha_0 + n - 1} \sum_{j=1}^{n-1} \delta(\theta_j),
\]

where \( \delta(\theta_j) \) indicates a degenerate distribution with all of its mass at \( \theta_j \). As \( n \) becomes sufficiently large, there will be repeated values in \( \{\theta_1, \ldots, \theta_{n-1}\} \) causing different amounts of mass to accumulate at the \( K \) locations associated with the collection of unique parameter values, \( \phi \). For example, the amount of mass at \( \phi_k \) is proportional to the number of instances of \( \phi_k \) in \( \{\theta_1, \ldots, \theta_{n-1}\} \). Denoting this count as \( n_k \) and rewriting the Pólya urn predictive rule in terms of the unique parameter values, we have the following conditional prior distribution:

\[
\theta_n | \theta_1, \ldots, \theta_{n-1} \sim \frac{\alpha_0}{\alpha_0 + n - 1} G_0 + \frac{1}{\alpha_0 + n - 1} \sum_{k=1}^{K} n_k \delta(\phi_k). \tag{3.12}
\]

The conditional prior distribution in (3.12) represents the data generating process of a DP as viewed through the Pólya urn scheme or modified CRP. The conditional distribution of the next parameter value, given the previously observed parameter values, is a mixture distribution between the base measure, \( G_0 \), and the collection of degenerate distributions corresponding to the previously observed parameter values. Consequentially, the next parameter value, \( \theta_n \), is either: (1) a new value, denoted \( \phi_{K+1} \), drawn from \( G_0 \), with probability \( \frac{\alpha_0}{\alpha_0 + n - 1} \), or (2) one of the previously observed
parameter values, $\phi_k$, with probability $\frac{n_k}{\alpha_0 + n - 1}$. We refer to the probabilities from the DPMM’s conditional prior distribution as the DPMM conditional prior probabilities. These probabilities play a key role in the development of the Cauchy-Net Mixture Model methodology and are summarized in Table 3.3.

**Table 3.3:** DPMM conditional prior probabilities for assigning $\theta_n$ given $\theta_1, \ldots, \theta_{n-1}$.

<table>
<thead>
<tr>
<th>Action</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign $\theta_n$ existing value $\phi_k$</td>
<td>$Pr(\theta_n = \phi_k \mid \theta_1, \ldots, \theta_{n-1}) = \frac{n_k}{\alpha_0 + n - 1}$</td>
</tr>
<tr>
<td>Assign $\theta_n$ new value $\phi_{K+1}$ from $G_0$</td>
<td>$Pr(\theta_n = \phi_{K+1} \mid \theta_1, \ldots, \theta_{n-1}) = \frac{\alpha_0}{\alpha_0 + n - 1}$</td>
</tr>
</tbody>
</table>

Inference for DPMMs often incorporates the collection of latent component labels, $c$. Since $\theta$ conveys the same information as $c$, $\phi$, and $K$, sampling $\theta_n$ directly from $\theta_n \mid \theta_1, \ldots, \theta_{n-1}$ is equivalent to sampling a value for component label $c_n$. If $c_n$ is assigned a previously unseen label, then we add a new parameter value $\phi_{K+1}$, drawn from $G_0$, to $\phi$ and increase $K$ by one. Due to the equivalency of sampling $\theta_n$ and $c_n$, the probabilities for sampling a value of $c_n$ in $\{1, \ldots, K+1\}$ follow from (3.12) and are identical to those reported in Table 3.3. Table 3.4 depicts the probability statements in terms of component labels and helps connect the Pólya urn predictive scheme and the CRP in Section 3.3.4. Specifically, the component label version of the DPMM conditional prior probabilities matches the CRP probabilities from (3.7) and allocating parameter values according to (3.12) equates to the modified CRP.

**Table 3.4:** DPMM conditional prior probabilities for assigning $c_n$ given $c_1, \ldots, c_{n-1}$.

<table>
<thead>
<tr>
<th>Action</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign $c_n$ existing value $k$</td>
<td>$Pr(c_n = k \mid c_1, \ldots, c_{n-1}) = \frac{n_k}{\alpha_0 + n - 1}$</td>
</tr>
<tr>
<td>Assign $c_n$ new value $K + 1$</td>
<td>$Pr(c_n = K + 1 \mid c_1, \ldots, c_{n-1}) = \frac{\alpha_0}{\alpha_0 + n - 1}$</td>
</tr>
</tbody>
</table>
The methods we present for performing inference utilize Gibbs sampling. Exploration of the joint posterior of $\theta$ follows from sampling from the full conditional posterior distribution of $\theta_i$ for $i = 1, \ldots, n$. Since allocating parameter values according to (3.12) generates a sequence of exchangeable values, we can treat each observation as the last. Leveraging the exchangeability and using the $-i$ notation introduced in Section 3.5.1, the conditional prior from (3.12) becomes the conditional prior for $\theta_i$, given the other $n - 1$ parameter values in $\theta^{-i}$, and is represented by:

$$
\theta_i \mid \theta^{-i} \sim \frac{\alpha_0}{\alpha_0 + n - 1}G_0 + \frac{1}{\alpha_0 + n - 1} \sum_{k=1}^{K^{-i}} n_k^{-i} \delta(\phi_k^{-i}),
$$

(3.13)

where we assume the reindexing procedure of Section 3.5.1 has taken place, if $i$ corresponds to a singleton component. Finally, updating the conditional prior in (3.13) using the data model, $f(x_i \mid \theta_i)$, and the observed data, $x_i$, yields the conditional posterior distribution for $\theta_i$:

$$
\theta_i \mid x_i, \theta^{-i} \sim \pi_{i,K^{-i}+1}G_{i,0} + \sum_{k=1}^{K^{-i}} \pi_{i,k} \delta(\phi_k^{-i}),
$$

(3.14)

where $G_{i,0}$ denotes the normalized posterior for $\theta_i$ resulting from updating the prior $G_0$ with the single observation $x_i$ and is given by:

$$
G_{i,0} = \frac{f(x_i \mid \theta_i)G_0(\theta_i)}{\int f(x_i \mid \theta_i)dG_0(\theta_i)}.
$$

Much like the conditional prior distribution for $\theta_i$, the conditional posterior for $\theta_i$ in (3.14) is a mixture between $G_{i,0}$ and the collection of degenerate distributions corresponding to the previously observed parameter values, with mixture weights:

$$
\pi_{i,k} = \begin{cases} 
  b \cdot \frac{\alpha_0}{\alpha_0 + n - 1} \cdot \int f(x_i \mid \theta_i)dG_0(\theta_i) & \text{if } k = K^{-i} + 1, \\
  b \cdot \frac{n_k^{-i}}{\alpha_0 + n - 1} \cdot f(x_i \mid \phi_k^{-i}) & \text{if } 1 \leq k \leq K^{-i},
\end{cases}
$$

(3.15)
where \( b \) is the appropriate normalizing constant so that \( \sum_{k=1}^{K-i+1} \pi_{i,k} = 1 \). Methods based on the conditional posterior for \( \theta_i \) in (3.14) or the corresponding posterior weights in (3.15) require that we can compute the integral for obtaining the marginal distribution of the data. This operation is viable when \( G_0 \) is conjugate for the data model, \( f(x_i | \theta_i) \).

Escobar (1994) and Escobar and West (1995) proposed early approaches for DPMM inference via Gibbs sampling. Inference proceeds by directly sampling \( \theta_i \) from the conditional posteriors in (3.14) for \( i = 1, \ldots, n \). Unfortunately, this approach suffers from slow mixing since the algorithm only updates the parameter for one observation at a time and ignores the fact that some observations have a high probability of sharing a parameter value (Neal, 2000).

West et al. (1994) present an alternative Gibbs sampler operating on \( c \) and \( \phi \), rather than \( \theta \). Originally proposed in the context of a Dirichlet process mixture of normals by MacEachern (1994), and later generalized by West et al. (1994) and Neal (2000), the method alternates between Gibbs sampling updates for \( c \) and \( \phi \). (In what follows, our notation most closely aligns with the presentation in (West et al., 1994).) Algorithm 3.1, deemed Gibbs Sampling with Conjugate Priors for DPMMs, outlines the approach. In the first step, we begin by removing observation \( i \) from its current component. In the case that observation \( i \) was a singleton component, the component closes and reindexing occurs as described in Section 3.5.1. We sample a new value of \( c_i \) using the logic from the conditional posterior distribution for \( \theta_i \) shown in (3.14). Sampling directly from \( \theta_i | x_i, \theta^{-i} \) either assigns \( \theta_i \) an existing value associated with one of the degenerate distributions in \( \theta^{-i} \) or a new value from \( G_{i,0} \). This is equivalent to assigning \( c_i \) one of the existing labels for the non-empty components, \( \{1, \ldots, K^{-i}\} \), or a new label, \( K^{-i} + 1 \). The probability of assigning \( c_i \) to each label in \( \{1, \ldots, K^{-i} + 1\} \) corresponds to the mixture weight in (3.15). Thus, the new value of \( c_i \) is drawn from a categorical distribution with the following probabilities:

\[
Pr(c_i = k | x_i, \phi^{-i}, c^{-i}) = \pi_{i,k} \quad \text{for } k = 1, \ldots, K^{-i} + 1.
\]  

(3.16)

In the case \( c_i = K^{-i} + 1 \), a new component opens and we must draw a new parameter value, \( \phi_{K^{-i}+1} \), from \( G_{i,0} \). The process for updating \( c_i \), which contains the possibility of both opening and closing components, highlights the dynamic nature of \( K \) in a DPMM. After updating all of the
component labels, the second step updates the parameter associated with each component. Draw a new value for \( \phi_k \) from the posterior based on the prior \( G_0 \) and all of the data assigned to the \( k^{th} \) component, \( X_k \). Up to proportionality, the posterior distribution for the parameter associated with the \( k^{th} \) component is given by:

\[
\phi_k \mid X_k, c \propto \left( \prod_{i: c_i = k} f(x_i \mid \phi_k) \right) \cdot G_0.
\]  

(3.17)

By updating the parameter for the component, the approach of West et al. (1994) updates \( n_k \) values in \( \theta \) at a time, allowing for more efficient mixing than the procedures directly sampling from the full conditional posterior for \( \theta \) in (3.14).

Algorithm 3.1: Gibbs Sampling with Conjugate Priors for DPMMs

**Current State:** Component labels (\( c \)), unique parameters (\( \phi \)), number of components (\( K \)), and component counts (\( n \))

**Step 1: Update Component Labels**

\[
\text{for } i \leftarrow 1 \text{ to } n \text{ do}
\]

- Remove observation \( i \) from current component
- Determine reduced state (\( c^{-i}, \phi^{-i}, K^{-i}, n^{-i} \)), reindexing if necessary
- Draw new component label, \( c_i \), from a categorical distribution with probabilities from (3.16)
  \[
  \text{if opening new component then}
  \]
  \[
  \mid \text{Draw a new unique parameter value, } \phi_{K^{-i}+1}, \text{ from } G_{i,0}
  \]
- Update current state to reflect changes in \( c, \phi, K, \) and \( n \)

\[
\text{end}
\]

**Step 2: Update Component-Specific Parameters**

\[
\text{for } k \leftarrow 1 \text{ to } K \text{ do}
\]

- Draw new parameter value, \( \phi_k \), from the posterior based on \( G_0 \) and the data assigned to component \( k \)

\[
\text{end}
\]

A number of variations on Algorithm 3.1 exist. For instance, MacEachern (1994) further exploits conjugacy to also integrate out \( \phi \). As a result, the mixture weight for existing components in (3.15)
depends on the posterior predictive distribution instead of the data model and the Gibbs sampler only operates on \( c \). Other variations of Algorithm 3.1 attempt to solve the limitation regarding the need to compute the integral in (3.15). The “No Gaps” algorithm of (MacEachern and Müller, 1998) and “Algorithm 8” of (Neal, 2000) are two such algorithms that explore approaches for replacing the integral with likelihood calculations for handling the non-conjugate cases. In what follows, we detail Neal’s approach.

Neal (2000) proposes a Gibbs sampling approach based on auxiliary parameters for use in the non-conjugate case. Algorithm 3.2, which Neal calls Gibbs Sampling with Auxiliary Parameters (or Algorithm 8), modifies the procedure for updating the component labels in Step 1 of Algorithm 3.1. The modified procedure generates auxiliary parameters to represent potential, or unused, components, not yet assigned an observation. The inclusion of the auxiliary parameters impacts the form of the prior and posterior distributions used in the update of \( c_i \) but eliminates the need for evaluating the troublesome integral.

The update for \( c_i \) begins with removing observation \( i \) from its current component, which produces a reduced state including \( K^{-i} \) and the reindexed versions of \( c^{-i}, \phi^{-i}, \) and \( \{n^{-i}_k\} \). Gibbs Sampling with Auxiliary Parameters then augments the reduced state with \( m \) temporary auxiliary components, defined by the parameters in \( \tilde{\phi} = \{\tilde{\phi}_{K^{-i}+1}, \ldots, \tilde{\phi}_{K^{-i}+m}\} \) and labeled \( \{K^{-i}+1, \ldots, K^{-i}+m\} \). The values in \( \tilde{\phi} \) come from \( m \) independent draws from \( G_0 \). The only exception to this occurs when observation \( i \) represents a singleton. In such cases, define one of the auxiliary components using \( \theta_i \) and only draw \( m-1 \) values from \( G_0 \). For notational convenience, we assign the singleton’s previous parameter to the first auxiliary component \( (\tilde{\phi}_{K^{-i}+1} = \theta_i) \).

The augmentation of the reduced state with the auxiliary components changes the form of the prior and posterior distributions. Previously, the conditional prior for \( \theta_i \) in (3.13) offered two options for assigning the next value: draw an existing value \( \phi^{-i}_k \), with probability proportional to \( n^{-i}_k \), or draw a new value from \( G_0 \), with probability proportional to \( \alpha_0 \). In comparison, Gibbs Sampling with Auxiliary Parameters offers the opportunity to choose from among \( m \) values drawn from \( G_0 \). Neal splits \( \alpha_0 \) equally among the \( m \) potential components producing the conditional prior distribution for \( \theta_i \), given by:
\[ \theta_i \mid \theta^{-i}, \bar{\phi} \sim \frac{\alpha_0/m}{\alpha_0 + n - 1} \sum_{k=K^{-i}+1}^{K^{-i}+m} \delta(\bar{\phi}_k) + \frac{1}{\alpha_0 + n - 1} \sum_{k=1}^{K^{-i}} n^{-i}_k \delta(\phi^{-i}_k). \] (3.18)

Despite the seemingly different appearances, the conditional prior distribution for \( \theta_i \) incorporating the auxiliary points in (3.18) maintains the DPMM prior probabilities from (3.13). Specifically, the conditional prior probability of assigning a previously unseen parameter value remains at \( \frac{\alpha_0}{\alpha_0 + n - 1} \) and the conditional prior probability of assigning the previously observed parameter value \( \phi^{-i}_k \) remains at \( \frac{n^{-i}_k}{\alpha_0 + n - 1} \). Further, since the latter probability remains unchanged, the conditional prior probability of assigning an existing parameter value is not affected by the inclusion of the auxiliary components. Updating (3.18) using the data model, \( f(x_i \mid \theta_i) \), and the observed data, \( x_i \), the conditional posterior for \( \theta_i \) is given by:

\[ \theta_i \mid x_i, \theta^{-i}, \bar{\phi} \sim \sum_{k=K^{-i}+1}^{K^{-i}+m} \pi_{i,k} \delta(\bar{\phi}_k) + \sum_{k=1}^{K^{-i}} \pi_{i,k} \delta(\phi^{-i}_k), \]

where

\[ \pi_{i,k} = \begin{cases} 
    b \cdot \frac{\alpha_0/m}{\alpha_0 + n - 1} \cdot f(x_i \mid \bar{\phi}_k) & \text{if } K^{-i} + 1 \leq k \leq K^{-i} + m \\
    b \cdot \frac{n^{-i}_k}{\alpha_0 + n - 1} \cdot f(x_i \mid \phi^{-i}_k) & \text{if } 1 \leq k \leq K^{-i},
\end{cases} \] (3.19)

where \( b \) is the appropriate normalizing constant so that \( \sum_{k=1}^{K^{-i}+m} \pi_{i,k} = 1 \). Thus, the new value of \( c_i \) is drawn from a categorical distribution with the following probabilities:

\[ P_r(c_i = k \mid x_i, \phi^{-i}, c^{-i}, \bar{\phi}) = \pi_{i,k} \quad \text{for } k = 1, \ldots, K^{-i} + m, \] (3.20)

where \( \pi_{i,k} \) follows from (3.19). In the case where the new label for \( c_i \) represents any of the auxiliary components, we adjust the labeling so that observation \( i \) gets assigned to the first non-empty component, i.e., \( c_i = K^{-i} + 1 \). This approach is in accordance with wanting to make sure no gaps exist in the labels. Finally, discard the auxiliary parameters for any the unselected components and
generate new auxiliary parameters during the update for the next component label. After updating all of the component labels, we update each component’s parameter using the posterior distribution based on the component’s data. This step is identical to Step 2 in Algorithm 3.1. Algorithm 3.2 summarizes the complete Gibbs Sampling with Auxiliary Parameters approach.
Algorithm 3.2: Gibbs Sampling with Auxiliary Parameters

Select number of auxiliary parameters: \( m \)

Current State: Component labels \( (c) \), unique parameters \( (\phi) \), number of components \( (K) \), and component counts \( (n) \)

Step 1: Update Component Labels

for \( i \leftarrow 1 \) to \( n \) do
  Remove observation \( i \) from its current component
  Determine the reduced state \( (c^{-i}, \phi^{-i}, K^{-i}, n^{-i}) \), reindexing if necessary
  Generate Auxiliary Points
  if \( i \) is a singleton then
    Assign the current parameter to one of the auxiliary components \( (\phi_{K^{-i}+1} = \theta_i) \)
    Draw remaining auxiliary parameters independently from \( G_0 \)
    \( (\phi_{K^{-i}+2}, \ldots, \phi_{K^{-i}+m} \sim G_0) \)
  else
    Draw auxiliary parameters independently from \( G_0 \)
    \( (\phi_{K^{-i}+1}, \ldots, \phi_{K^{-i}+m} \sim G_0) \)
  end
  Draw a new component label, \( c_i \), from a categorical distribution with probabilities from (3.20)
  if \( c_i \) corresponds to auxiliary component then
    Relabel \( c_i = K^{-i} + 1 \)
  end
  Update current state to reflect changes in \( c, \phi, K, \) and \( n \)
  Delete all empty auxiliary components
end

Step 2: Update Component-Specific Parameters

for \( k \leftarrow 1 \) to \( K \) do
  Draw new parameter value, \( \phi_k \), from the posterior based on \( G_0 \) and the data assigned to component \( k \)
end

The MCMC algorithms for DPMM inference presented in this section are appropriate for small-to-medium sized datasets. For larger datasets, the computational burden of MCMC is too expensive and faster alternatives are necessary for performing inference. Blei et al. (2006) develop a vari-
ational inference approach based on approximating the posterior distribution with the variational distribution. This approach replaces computationally expensive MCMC methodologies with a deterministic optimization problem for selecting the variational parameters through a minimization of the KL divergence between the variational and posterior distributions. Wang and Dunson (2011) develop a sequential updating and greedy search algorithm for finding a “good partition.” The procedure assigns one observation at a time until all observations are associated with a component. Each assignment selects the component with the highest conditional posterior probability, given the observed data and component labels for the previously assigned observations. Based solely on deterministic calculations, the approach makes a single pass through assigning the observations to components making the approach computationally feasible even for large datasets.

### 3.5.3 Estimating Hyperparameters

Algorithms 3.1 and 3.2 assume the hyperparameters are known but are easily extendable to include steps for hyperparameter updates. We are interested in estimating the DP’s concentration parameter, $\alpha_0$, which plays a key role in determining the number of mixture components, $K$. For DPMMs, $K$ is a random variable whose distribution depends on the number of observations, $n$, and $\alpha_0$ (Antoniak, 1974). Liu (1996) explores the estimation of $\alpha_0$ and proves that the MLE of $\alpha_0$ must satisfy:

$$ \mathbb{E}[K \mid \alpha_0, n] = \sum_{i=1}^{n} \frac{\alpha_0}{\alpha_0 + i - 1}, \tag{3.21} $$

where $\mathbb{E}$ represents the expectation. McAuliffe et al. (2006) utilize Liu’s result and estimate the expectation by averaging over the number of components discovered in each of the last $B$ Gibbs samples. Replacing the expectation with the estimate in (3.21), they numerically solve for $\alpha_0$ to provide an updated value for the next Gibbs step. Since this approach violates the Markovian property of a Gibbs sampler, we follow the recommendation of Song (2017) and use $B = 1$.

DPMMs are flexible tools for modeling and discovering cluster structure. In Chapter 4, we explore an extension of a DPMM, known as the CNMM that is useful when the data contain anomalies.
Chapter 4

Cauchy-Net Mixture Model

We present a Bayesian nonparametric tool, called a Cauchy-Net Mixture Model (CNMM), for clustering data containing anomalous observations. The primary goal of the CNMM methodology is to offer a flexible procedure for identifying and isolating anomalies, while simultaneously discovering cluster structure among the non-anomalous, or regular, observations. To achieve this goal, a CNMM employs a mixture between a Dirichlet Process Mixture Model (DPMM) and a Cauchy distributed component, which we call the Cauchy-Net (CN). Each portion of the model offers benefits, as the DPMM eliminates the limitation of requiring a fixed number of components and the CN captures observations that do not fit into the well-defined components by leveraging its heavy tails. By isolating the anomalous observations in a single cluster, we simultaneously identify the observations in the net as warranting further inspection and prevent them from interfering with the formation of the remaining clusters. In this chapter, we motivate the need for the CNMM, formulate the model, and present methods for performing inference on a CNMM.

4.1 Motivation for the Cauchy-Net Mixture Model

Clustering is a popular and intuitive task, but suffers from many challenges. As a result of the difficulties, an abundance of clustering approaches exist, each bringing about benefits and limita-
tions. $K$-means, benefiting from its simplicity, is one of the most popular clustering approaches, but its heuristic nature fails to provide opportunities for performing inference and for quantifying the uncertainty in cluster assignments. Finite mixture modeling improves upon the heuristic approaches of $K$-means and hierarchical clustering by employing a probability model as the basis for describing the data. The use of a probability model enables formal inference procedures and soft clustering assignments for expressing the level of uncertainty regarding an observation’s cluster membership. Despite such advantages, most finite mixture modeling approaches suffer from the restriction that the number of components, $K$, is fixed. (Bayesian approaches placing a prior on $K$ are an exception.) DPMMs exhibit the same benefits as finite mixture models but allow for a dynamic $K$. Unfortunately, DPMMs, like most clustering approaches, struggle in the presence of anomalous data.

Anomalies have the potential to interfere with our ability to discover cluster structure among the non-anomalous, or regular, observations. To connect the type of observations in a cluster to the type of cluster, we refer to clusters containing the regular observations as “regular clusters” or “regular components” and clusters containing the anomalies as “anomalous components” or the Cauchy-Net. Consider a hypothetical situation in which researchers wish to perform a clustering analysis, and, based on a multitude of historical studies on similar data, two clusters are expected. Figure 4.1 depicts three datasets containing anomalous data. The first column provides the true labels and the second column shows the labels discovered during the clustering analysis. Common to all scenarios, the dataset consists of 20 observations, 2 anomalies (red addition symbols), and 2 regular clusters (black open triangles and blue open circles) defined by 18 regular points. Each cluster’s centroid is labeled with a solid symbol matching the shape and color of the observations in the cluster. The anomalies have varying impacts on our ability to discover structure among the regular observations in the three cases. Clustering in the first scenario (Figure 4.1, row 1) assigns one of the anomalies to a singleton cluster, while forcing the other anomaly into a cluster with the remainder of the data. Since all of the regular observations are grouped together, this attempt failed to recover any of the structure in the two regular components, despite only 10% of the data being anomalous. If, in addition to clustering, the researchers wish to treat the discovered centroids as being representative of cluster members for making predictions, the predictions will
suffer since the centroid does not closely resemble any of the observed data. In the second scenario, the anomalies have less of an impact. The larger regular cluster (blue open circles in the first column) is recovered perfectly. The smaller cluster absorbs both anomalies, which clearly have an attractive force on the discovered centroid. As the discovered centroid moves away from the true centroid of the smaller regular cluster, predictions suffer. Finally, in the third scenario, each regular cluster absorbs one anomaly; however, the anomalies only have a minimal impact on the centroids. Despite the anomalies, the clustering approach recovers most of the structure in the regular observation and the discovered centroids closely approximate the true centroids. These three scenarios illustrate two major concerns. First, the presence of anomalies has the potential for interfering with our ability to find structure in data. Although the anomalies may not always be a hindrance, there is certainly potential for anomalies to have a strong influence. Second, fixing $K$, without allowing any flexibility, and/or forcing anomalies into clusters where they do not belong are risky practices.
Figure 4.1: Three scenarios illustrating the varying impacts of anomalies on clustering. The color and plotting symbol indicate cluster membership, with solid shapes representing the cluster centroid. In the first scenario, the anomalies prevent the discovery of cluster structure. Some structure (blue cluster) is recovered in the second, while most structure is recovered in the third.
With these concerns in mind, we develop a method for clustering and prediction with four desirable characteristics. First, we want the method to cluster observations, without the restriction of specifying the number of clusters in advance of the analysis. The ability to grow the number of clusters, which requires an expansion of the parameter space describing the clusters, offers flexibility lacking in many of the previously discussed methods. Second, the method should reflect uncertainties in cluster assignments by using soft, rather than hard, clustering assignments. Soft clustering allows us to distinguish between the observations whose cluster memberships are nearly certain and those observations existing near the intersection of clusters whose memberships are less clear. Third, we seek a procedure that offers the flexibility of allowing for simultaneously clustering and making predictions, as opposed to a procedure that only clusters, only predicts, or only predicts after the conclusion of the clustering process. Particularly when the goal is to make predictions, incorporating predictive ability into the formation of the clusters is desirable. Although our current focus is on clustering, prediction is a long-term goal that we want to be able to achieve with our method. Finally, the method should be robust to anomalous data. Not only do we want to prevent anomalous data from impacting clustering and prediction, but we also seek an approach that identifies observations warranting further investigation, rather than just dampening the effect of such observations.

We propose the CNMM for clustering and prediction in the presence of anomalous data. The CNMM extends a DPMM by creating a mixture of a DPMM with an additional Cauchy distributed component, which we refer to as the Cauchy-Net (CN). Each portion of the model contributes desirable characteristics. DPMMs, only lacking a mechanism for handling anomalous data, eliminate the limitation of requiring a fixed $K$, provide soft clustering assignments, and allow for simultaneously clustering and predicting. The CN offers a solution for handling anomalous data by leveraging the heavy tails of a Cauchy distribution for capturing observations that are poorly described by the well-defined components.

Figure 4.2 illustrates the rationale for the use of a Cauchy component when clustering data containing anomalies. Suppose we employ a mixture of two normal distributions (black, solid lines) and a Cauchy distribution (red, dashed) for clustering univariate data. There is a well-defined clus-
The application of a CNMM to a dataset partitions observations into two categories, regular and anomalous, while simultaneously extracting cluster structure among the regular observations. Figure 4.3 depicts the process and the nature of the end result. Beginning with unlabeled data and applying the CNMM, we identify regular (small gray circles) and anomalous (small red circles) observations. Isolating the anomalies in their own component (i.e., the CN) provides two benefits.
First, we gain the interpretation that observations in the CN do not match the patterns displayed in the regular data, thereby warranting further investigation. Second, due to their isolation, the anomalous observations are not being forced into, nor interfering with the formation of, the regular components (black dashed ellipses) containing the regular observations. The end result of using the CNMM is the discovery of cluster structure among the regular observations, despite the presence of anomalous data. By integrating a mechanism for handling anomalies into a DPMM, the CNMM boasts all four of the desired characteristics.

**Figure 4.3:** Process and results of a CNMM. Applying the CNMM to unlabeled data partitions the observations into types: anomalies (solid red circles) and regular observations (solid gray circles). By isolating the anomalies in the CN and mitigating their impact, we discover regular clusters (dashed ellipses) containing the regular observations.
4.2 Cauchy-Net Mixture Model

Suppose we wish to explore a collection of independent observations $X = \{x_1, \ldots, x_n\}$ using a mixture model with an unknown number of components. To identify and prevent anomalies in $X$ from interfering with our ability to learn about the cluster structure of the regular observations, we apply a CNMM. A CNMM employs a mixture of the CN, which is a Cauchy distributed component for capturing anomalies, and a DPMM, which is used for describing structure among the regular observations. Let $\theta_i$ and $\theta = \{\theta_1, \ldots, \theta_n\}$ represent the parameter associated with observation $i$ and the collection of parameter values for all observations, respectively. The collection of unique values in $\theta$, denoted by $\phi$, distinguishes the components. Knowledge of an observation’s parameter value, $\theta_i$ or the corresponding $\phi_k$, establishes component membership, depicted by component label $c_i$. Each component is characterized by a density function, $f$, conditioned on the component’s parameter value, but the form of the density varies depending on the parametric family governing the component. We assume that the parameter value $\theta_i$ not only identifies the component but also indicates the corresponding parametric family. In this manner, if $\theta_i$ maps observation $i$ to the CN, then $\theta_i$ also establishes $f$ as the Cauchy density; if $\theta_i$ maps to one of the DPMM components, then $f$ corresponds to the density of the parametric family chosen for the DPMM. To emphasize density $f$’s dependence on the parameter value, we subscript $f$ with the same index as the parameter, e.g., $f_i(x_i | \theta_i)$ or $f_k(x_i | \phi_k)$. Building on the formulation of a DPMM, we represent the CNMM as:

$$
\begin{align*}
  x_i \mid \theta_i & \sim f_i(x_i \mid \theta_i) \\
  \theta_i \mid G, \phi_0 & \sim \pi_A \delta(\phi_0) + (1 - \pi_A)G \\
  G \mid \alpha_0, G_0 & \sim DP(\alpha_0, G_0) \\
  \phi_0 & = \{\mu_0, \gamma_0\},
\end{align*}
$$

(4.1)

where $\pi_A$ is the proportion of anomalous observations and $\phi_0 = \{\mu_0, \gamma_0\}$ represents the location and scale of the CN, whose density we denote with $C(\cdot | \mu_0, \gamma_0 I_p)$. We assume $\phi_0$ is fixed and recommend $\mu_0 = 0_p$ and $\gamma_0 = 1$ for use with standardized data in $p$-dimensional space. Finally, we point out that (4.1) is based on a single CN but is easily extendable by specifying additional
degenerate distributions for the location and scale of the other nets.

The mixing distribution $\theta_i \mid G, \phi_0$ in (4.1) plays a crucial role in the CNMM. The distribution is a mixture between the degenerate distribution at $\phi_0$, representing the known parameters for the CN, and the discrete probability distribution $G$, drawn from the DP and representing the distribution of parameters in the DPMM. With probability $\pi_A$, $\theta_i$ is assigned the value $\phi_0$, which implies observation $i$ belongs to the CN and $f_i(x_i \mid \theta_i)$ is the Cauchy density. Otherwise, $\theta_i$ is drawn from $G$, which assigns $i$ to one of the components in the DPMM and $f_i(x_i \mid \theta_i)$ is the density of the parametric family chosen for the DPMM. Thus, upon conditioning on $\theta_i$ induced by the mixing distribution, the data model for $x_i$ is the mixture of a Cauchy distribution and a DPMM.

We compare the DPMM and the CNMM by exploring the data generating (bottom-up) perspective of each through the modified CRP metaphor described in Section 3.3.4. The key difference lies in the mixing distributions from (3.10) and (4.1) and the resulting induced data models. Starting with the DPMM case, Blackwell and MacQueen (1973) marginalize $G$ in the joint prior distribution of $\{\theta, G\}$ given by:

$$
\theta_i \mid G \sim G \\
G \mid \alpha_0, G_0 \sim DP(\alpha_0, G_0),
$$

which leads to the Pólya urn predictive rule for $\theta_n \mid \theta_{1:(n-1)}$ in (3.12). (Here, $\theta_{1:(n-1)}$ denotes $\{\theta_1, \ldots, \theta_{n-1}\}$). Using (3.12), we can also illustrate the allocation of parameter value $\theta_n$ through the modified CRP metaphor in which customer $n$ either: (1) joins an existing table with probability $n_{k} \alpha_{0} + n - 1$ and takes on the table’s parameter value $\phi_{k}$, or (2) starts a new table with probability $\frac{\alpha_{0}}{\alpha_{0} + n - 1}$ and draws a new parameter value from $G_0$. As discussed in Section 3.5.2, these probabilities represent the DPMM’s conditional prior probabilities and are repeated in the DPMM column of Table 4.1. In the CNMM, if we marginalize over $G$ in:
\[ \theta_i \mid G, \phi_0 \sim \pi_A \delta(\phi_0) + (1 - \pi_A)G \]

\[ G \mid \alpha_0, G_0 \sim DP(\alpha_0, G_0) \]

\[ \phi_0 = \{\mu_0, \gamma_0\}. \]

then we arrive at the same general rule for assigning parameter values for the regular observations (i.e., those observations associated with parameters that link to the DPMM portion of the CNMM); however, a complication arises because the DP is not the only source for generating parameter values. With probability \( \pi_A \), \( \theta_i \) is assigned parameter value \( \phi_0 \), linking observation \( i \) to the CN and placing it in the anomalous category. Denoting the number of observations in \( \theta_{1:(n-1)} \) assigned \( \phi_0 \) as \( n_0 \), the \( n - 1 \) previous observations are partitioned into \( n_0 \) anomalies with parameter \( \phi_0 \) and \( n - n_0 - 1 \) regular observations with parameter values assigned from the DP. Incorporating the mixture weights and counts, the updated predictive rule of (3.12) becomes:

\[ \theta_n \mid \theta_{1:(n-1)}, \phi_0 \sim \pi_A \delta(\phi_0) + \frac{1 - \pi_A}{\alpha_0 + n - n_0 - 1} \left( \alpha_0 G_0 + \sum_{k=1}^{K} n_k \delta(\phi_k) \right). \quad (4.2) \]

We represent the allocation of parameter value \( \theta_n \) for the updated rule through an extension of the modified CRP metaphor in which customer \( n \) either: (1) joins an existing table of regular observations with probability \( \frac{(1-\pi_A)n_k}{\alpha_0 + n - n_0 - 1} \) and takes on the table’s parameter value \( \phi_k \), (2) starts a new table for regular observations with probability \( \frac{(1-\pi_A)\alpha_0}{\alpha_0 + n - n_0 - 1} \) and draws a new parameter value from \( G_0 \), or (3) joins the Cauchy table with probability \( \pi_A \) and is assigned parameter value \( \phi_0 \). The CNMM column of Table 4.1 summarizes these probabilities, which we call the CNMM conditional prior probabilities. In summary, a comparison the DPMM and CNMM columns reveals that supplementing the DPMM with the Cauchy component provides a third option for the source of the next parameter value (\( \delta(\phi_0) \)) and alters the conditional prior probabilities. As we will see in Section 4.3, these changes require a modification to the MCMC algorithms for performing inference.
Table 4.1: Conditional prior probabilities for assigning $\theta_n$ given $\theta_{1:(n-1)}$ in the DPMM and CNMM.

<table>
<thead>
<tr>
<th>Action and Probability</th>
<th>DPMM</th>
<th>CNMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign $\theta_n$ existing regular parameter $\phi_k$</td>
<td>$P_r(\theta_n = \phi_k \mid \theta_{1:(n-1)}) = \frac{n_k}{\alpha_0 + n - 1}$</td>
<td>$\frac{(1 - \pi_A) \cdot n_k}{\alpha_0 + n - n_0 - 1}$</td>
</tr>
<tr>
<td>Assign $\theta_n$ new regular parameter $\phi_{K+1}$ from $G_0$</td>
<td>$P_r(\theta_n = \phi_{K+1} \mid \theta_{1:(n-1)}) = \frac{\alpha_0}{\alpha_0 + n - 1}$</td>
<td>$\frac{(1 - \pi_A) \cdot \alpha_0}{\alpha_0 + n - n_0 - 1}$</td>
</tr>
<tr>
<td>Assign $\theta_n$ anomalous parameter $\phi_0$</td>
<td>$P_r(\theta_n = \phi_0 \mid \theta_{1:(n-1)}) = NA$</td>
<td>$\pi_A$</td>
</tr>
</tbody>
</table>

Finally, we extend the modified CRP metaphor to the full-blown mixture model representation of the CNMM shown in Figure 4.4. For ease of notation, we assume the Cauchy table is Table 0 and the existing regular tables, which correspond to the regular observations, are labeled $\{1, \ldots, K\}$. This produces a total of $K + 1$ components, $K$ regular and 1 anomalous. We emphasize that according to (4.1), once the Cauchy table opens, all future new tables come from the DP and represent the regular observations, reflecting the CNMM’s single CN.
4.3 Cauchy-Net Mixture Model Inference

Inference for a CNMM follows the same principles as inference for DPMMs but must reflect changes brought about by the CN. The two most significant consequences of augmenting a DPMM with the CN relate to having different types of components and the impact on the model. In this section, we present modifications to the two Gibbs sampling procedures described in Section 3.5.2 in order to accommodate the changes.

The CN, a Cauchy distributed component for capturing anomalies, serves a different purpose than the DPMM components, which are intended to discover structure among the regular observations. To distinguish between the types of components, we index the DPMM components using $1, \ldots, K$. 

---

**Figure 4.4:** Modified Chinese restaurant process mixture model representation of a Cauchy-Net Mixture Model showing the assignments of the first 10 customers (white squares) to tables (large, shaded circles). Table 0 represents the Cauchy-Net, whose parameters are fixed, rather than randomly generated from $G_0$. Observations for the individuals (small, shaded circles) are i.i.d. from the parametric family with the density defined by the table parameter. Empty regular tables have a dashed outline.
with no gaps, while the CN is indexed with 0. This implies $K + 1$ total components, $K$ DPMM components, and 1 CN. Both Gibbs sampling procedures rely on sequentially removing an observation from its current component and using the full conditional posterior distribution for making an update. If the observation represents a singleton component, the component closes and reindexing occurs as described in Section 3.5.1; however, we assume both portions of the CNMM (i.e., the CN and the DPMM) remain open throughout the process to reflect the distinct purposes of the components. This requires at least one observation, which could be an actual observation from the dataset or an auxiliary observation, to remain in each portion at all times. Consequentially, the CNMM always has at least two components open in support of our belief that the dataset contains a combination of anomalous and regular observations.

The more significant impact on inference relates to the changes in the prior and posterior distributions induced by the inclusion of the CN. As we saw in Section 4.2, the option of assigning an observation the parameter value associated with the CN, $\phi_0$, altered the form of the conditional prior distribution for $\theta_i$ and the corresponding probabilities shown in Table 4.1. Leveraging exchangeability and using the $-i$ notation, the conditional prior from (4.2) becomes:

$$
\theta_i \mid \theta^{-i}, \phi_0 \sim \pi_A \delta(\phi_0) + \frac{1 - \pi_A}{\alpha_0 + n - n_{0^{-i}} - 1} \left( \alpha_0 G_0 + \sum_{k=1}^{K^{-i}} n_{k^{-i}} \delta(\phi_{k^{-i}}) \right). \quad (4.3)
$$

where $n_{0^{-i}}$ represents the number of times a value in $\theta^{-i}$ matches the CN parameter $\phi_0$, or, alternatively, the number of observations in the CN when excluding observation $i$. Updating the conditional prior in (4.3) using the data model, $f(x_i \mid \theta_i)$, and the observed data $x_i$, yields the conditional posterior distribution for $\theta_i$ given by:

$$
\theta_i \mid x_i, \theta^{-i}, \phi_0 \sim \pi_{i,0} \delta(\phi_0) + \pi_{i,K^{-i}+1} G_{i,0} + \sum_{k=1}^{K^{-i}} \pi_{i,k} \delta(\phi_{k^{-i}}). \quad (4.4)
$$

As before, $G_{i,0}$ denotes the normalized posterior for $\theta_i$ resulting from updating the prior $G_0$ with the single observation $x_i$ and is shown by:
\[ G_{i,0} = \frac{f_i(x_i \mid \theta_i)G_0(\theta_i)}{\int f_i(x_i \mid \theta_i)dG_0(\theta_i)}. \]

The conditional posterior for \( \theta_i \) in (4.4) is a mixture between the degenerate distribution for the CN parameter (\( \delta(\phi_0) \)), the collection of degenerate distributions corresponding to the previously observed parameter values for DPMM components (\( \delta(\phi_{-i}^k) \)), and \( G_{i,0} \). The corresponding mixture weights are defined by:

\[
\pi_{i,k} = \begin{cases} 
  b \cdot \pi_A \cdot C(\mathbf{x}_i \mid \phi_0) & \text{if } k = 0, \\
  b \cdot \frac{(1 - \pi_A)n_k^{-i}}{\alpha_0 + n - n_0^{-i} - 1} \cdot f_k(x_i \mid \phi_k^{-i}) & \text{if } 1 \leq k \leq K^{-i}, \\
  b \cdot \frac{(1 - \pi_A)\alpha_0}{\alpha_0 + n - n_0^{-i} - 1} \cdot \int f_i(x_i \mid \theta_i)dG_0(\theta_i) & \text{if } k = K^{-i} + 1,
\end{cases}
\]  

(4.5)

where \( C \) represents the Cauchy density and \( b \) is the appropriate normalizing constant so that \( \sum_{k=0}^{K^{-i}+1} \pi_{i,k} = 1 \). The limitation of methods based on this form of the posterior distribution is the calculation of the integral in (4.5). The integral is generally feasible when \( G_0 \) is conjugate to the data model, \( f_i(x_i \mid \theta_i) \), for the DPMM components.

**Gibbs Sampling with Conjugate Priors for CNMMs**

Algorithm 3.1, replaces direct sampling from the conditional posterior for \( \theta_i \) with equivalent procedures operating on \( c \) and \( \phi \). The rationale comes from realizing that drawing \( \theta_i \) from its full conditional posterior with probabilities given by the mixture weights equates to assigning a component label by sampling from a categorical distribution with probabilities given by the same weights. Thus, using the CNMM’s conditional posterior for \( \theta_i \) given in (4.4) and the corresponding mixture weights, \( \pi_{i,k} \), in (4.5), we adapt Algorithm 3.1 by updating the component label \( c_i \) with a draw from a categorical distribution using the following probabilities:
\[ Pr(c_i = k \mid x_i, \phi^{-i}, c^{-i}) = \pi_{i,k}, \quad \text{for} \quad k = 0, \ldots, K^{-i} + 1, \]

where \( \pi_{i,k} \) is defined in (4.5). Aside from changes in the number of options (i.e., we now have the additional option of assigning the observation to the CN) and in computing the probabilities, the remainder of Algorithm 3.1 stays the same. Finally, we point out that since the CN parameters are assumed known and fixed, we only update the parameters corresponding to the DPMM components in the second step of the algorithm.

**Gibbs Sampling with Auxiliary Parameters for CNMMs**

Algorithm 3.2 provides a method for inference that does not require conjugacy of the base measure and data model. The approach augments the parameter space with \( m \) auxiliary parameters representing potential components, not yet been assigned an observation. When extending Algorithm 3.2 to the CNMM, we emphasize that the potential components correspond to the DPMM portion of the CNMM, thereby representing components for the regular observations. The procedure for generating the auxiliary parameters remains unchanged from Algorithm 3.2; however, including the auxiliary parameters alter the form of the prior and posterior distributions for \( \theta_i \). Splitting \( \alpha_0 \) equally among the \( m \) potential components results in the conditional prior distribution for \( \theta_i \), given by:

\[ \theta_i \mid \theta^{-i}, \phi_0, \bar{\phi} \sim \pi_A \delta(\phi_0) + \frac{1 - \pi_A}{\alpha_0 + n - n_0^{-i} - 1} \left( \frac{\alpha_0}{m} \sum_{k=K^{-i}+1}^{K^{-i}+m} \delta(\bar{\phi}_k) + \sum_{k=1}^{K^{-i}} n_k^{-i} \delta(\phi_k^{-i}) \right). \]

\( \bar{\phi} = \{ \bar{\phi}_{K^{-i}+1}, \ldots, \bar{\phi}_{K^{-i}+m} \} \) depicts the auxiliary parameters, and \( n_0^{-i} \) represents the number of observations in the CN, excluding observation \( i \). As in the case of the DPMM, the inclusion of the auxiliary parameters does not change the CNMM conditional prior probabilities. Updating with the data, we get the full conditional posterior for \( \theta_i \) represented by:
\[
\theta_i \mid x_i, \theta^{-i}, \phi_0, \bar{\phi} \sim \pi_{i,0}(\phi_0) + \sum_{k=K^{-i}+1}^{K^{-i}+m} \pi_{i,k}(\bar{\phi}_k) + \sum_{k=1}^{K^{-i}} \pi_{i,k}(\phi_k^{-i}),
\]

where

\[
\pi_{i,k} = \begin{cases} 
    b \cdot \pi_A \cdot C(x_i \mid \phi_0) & \text{if } k = 0, \\
    b \cdot \frac{(1 - \pi_A)}{\alpha_0 + n - n_0^{-i} - 1} \cdot n_k^{-i} \cdot f_k(x_i \mid \phi_k^{-i}) & \text{if } 1 \leq k \leq K^{-i}, \\
    b \cdot \frac{(1 - \pi_A)}{\alpha_0 + n - n_0^{-i} - 1} \cdot \frac{\alpha_0}{m} \cdot f_k(x_i \mid \bar{\phi}_k) & \text{if } K^{-i} + 1 \leq k \leq K^{-i} + m.
\end{cases}
\]

Thus, based on the updated version of the CNMM’s conditional posterior for \(\theta_i\) given in (4.6) and the corresponding mixture weights, \(\pi_{i,k}\), in (4.7), we adapt Algorithm 3.2 by updating the component label \(c_i\) with a draw from a categorical distribution using the following probabilities:

\[
Pr(c_i = k \mid x_i, \phi^{-i}, c^{-i}) = \pi_{i,k}, \text{ for } k = 0, \ldots, K^{-i} + m,
\]

where \(\pi_{i,k}\) is defined in (4.7). Aside from changes in probability formulas, the remainder of Algorithm 3.2 stays the same. As before, we only update the parameters corresponding to the DPMM components in the second step of the algorithm since the CN parameters are assumed known and fixed.

### 4.4 Hyperparameter Updates for the Cauchy-Net Mixture Model

We can extend the Gibbs Sampling with Conjugate Priors for CNMMs and Gibbs Sampling with Auxiliary Parameters for CNMMs procedures to incorporate hyperparameter estimations. In particular, we are interested in updating the DP concentration parameter, \(\alpha_0\), and the mixture weight
for the Cauchy-Net, $\pi_A$.

### 4.4.1 Estimation of Dirichlet Process Concentration Parameter

When performing inference on the CNMM, we update $\alpha_0$ after each Gibbs sampling step using the method described in Section 3.5.3. Since $\alpha_0$ corresponds to the DP governing the DPMM portion of the CNMM, we make a simple modification to the procedure. We want the estimation of $\alpha_0$ to depend on the number of regular observations, $n - n_0$, and the number of regular components, $K$, discovered after updating the component labels in the current Gibbs sampling iteration. Approximating $\mathbb{E}(K \mid \alpha_0, n)$ with $K$ from the current iteration, rather than average over the last $B$ iterations, we use:

$$K \approx \sum_{i=1}^{n-n_0} \frac{\alpha_0}{\alpha_0 + i - 1}.$$  \hspace{1cm} (4.8)

### 4.4.2 Estimation of Cauchy-Net Mixture Weight

We consider two methods for updating $\pi_A$:

1. (Empirical Approach) At the end of each Gibbs sampling step, estimate $\pi_A$ using the proportion of the observations currently assigned to the CN:

$$\hat{\pi}_A = \frac{n_0}{n},$$  \hspace{1cm} (4.9)

where $n_0$ is the number of observations in the CN after updating all component labels and $n$ is the total number of observations. In the case of a DPMM, $n_0 = 0$ and $\pi_A = 0$, reducing the CNMM in (4.1) to the general DPMM in (3.10).

2. (Probabilistic Approach) At the end of each Gibbs sampling step, estimate $\pi_A$ using the posterior probability of the CN at the current state. Denoting the current state’s component-specific parameter estimates as $\hat{\phi} = \{\phi_0, \hat{\phi}_1, \ldots, \hat{\phi}_K\}$, the posterior probability of assigning observation $i$ to component $k$ conditional on the data and current state is:
\[ \hat{\pi}_{i,k} \equiv \Pr(c_i = k \mid x_i, \hat{\phi}) = \frac{f_k(x_i \mid c_i = k, \hat{\phi}_k)\Pr(c_i = k)}{\sum_{k=0}^{K} f_k(x_i \mid c_i = k, \hat{\phi}_k)\Pr(c_i = k)}, \quad (4.10) \]

where \( f_k \) is the density associated with component \( k \) and \( \Pr(c_i = k) \) is the prior probability that an observation is assigned to component \( k \). Recall that we are indexing the CN with a 0, so \( \phi_0 \) represents the known parameter for the CN, which is the reason it does not have a hat in \( \hat{\phi} \). For the formula in (4.10), \( \hat{\phi}_0 \) is replaced with \( \phi_0 \) and \( f_0 \) represents the Cauchy density.

Thus, the probabilistic approach estimates \( \pi_A \) using:

\[ \hat{\pi}_i \equiv \frac{\sum_{i=1}^{n} \hat{\pi}_{i,0}}{n}. \quad (4.11) \]

The estimate provided by (4.9) serves as an upper bound for the estimate from (4.11). As we become certain that all \( n_0 \) observations currently in the CN belong to the CN and none of those currently outside the net belong, \( \hat{\pi}_{i,0} \to 1 \) and \( \sum_{i=1}^{n} \hat{\pi}_{i,0} \to n_0 \), causing the estimates converge.

The CNMM is a flexible Bayesian nonparametric tool designed for clustering in the presence of anomalous data. In Chapter 5, we explore the performance of the CNMM under a variety of experimental conditionss.
Chapter 5

Simulation Studies

A Dirichlet Process Mixture Model (DPMM), like most clustering approaches, struggles in the presence of anomalous data. The Cauchy-Net Mixture Model (CNMM) offers a solution by augmenting a DPMM with a Cauchy distributed component, which we call the Cauchy-Net (CN). The intuition behind the CNMM is to maintain the flexibility afforded by a DPMM, while leveraging the heavy tails of the CN to capture observations that are poorly described by the well-defined components. The goal is to isolate such observations in a single component, thereby preventing them from interfering with the formation of the remaining components. Beyond the modeling benefit, the CN methodology also provides the straightforward interpretation that observations assigned to the net are anomalous and warrant further inspection. In this chapter, we compare the performance of the CNMM and DPMM on a variety of simulated datasets in order to investigate the effectiveness of including the CN. We detail the experimental conditions, the procedure for generating datasets, and examine the results of the simulation studies.

5.1 Model Assumptions

A CNMM is a mixture between a DPMM for the regular observations, and the CN for the anomalous observations. A CNMM, originally described in (4.1) of Section 4.2 follows as:
\[ x_i \mid \theta_i \sim f_i(x_i \mid \theta_i) \]
\[ \theta_i \mid G, \phi_0 \sim \pi_A \delta(\phi_0) + (1 - \pi_A)G \]
\[ G \mid \alpha_0, G_0 \sim DP(\alpha_0, G_0) \]
\[ \phi_0 = \{\mu_0, \gamma_0\}. \]

The DPMM is a special case of the CNMM without the net, or, equivalently, with \( \pi_A = 0 \).

The simulation studies explore whether augmenting a DPMM with the CN improves the model’s ability to discover structure when clustering in the presence of anomalies. For a given dataset, we fit both a DPMM and the corresponding CNMM, where the only difference is the inclusion of the net. In order to perform inference using the CNMM, we must select a parametric family for the DPMM portion governing the regular observations, specify the fixed values for the CN parameters (\( \phi_0 \)), and determine the hyperparameters (\( \pi_A, \alpha_0, \) and \( G_0 \)).

For our studies, let the DPMM be a mixture of \( p \)-dimensional normal distributions, and component \( k \)’s density, \( f_k \), depends on mean, \( \mu_k \), and covariance, \( \Sigma_k \). Within each DPMM component, the \( p \) dimensions are independent and have constant variance; however, the magnitude of the constant variance differs between components (i.e., \( \Sigma_k = \sigma_k^2 I_p \), where \( I_p \) is the \( p \times p \) identity matrix). The assumptions on the covariance structure produce spherical clusters of varying size, as shown in Figure 5.1 of Section 5.2. We denote the collection of parameters defining the \( k^{th} \) normal component as \( \phi_k = \{\mu_k, \sigma_k^2\} \). The CNMM maintains the same normality and covariance assumptions for its DPMM portion but also includes the CN. Indexing the CN as the \( 0^{th} \) component, component \( 0 \)’s density, \( f_0 \), is the Cauchy density with location and scale known and fixed at \( \mu_0 = 0_p \) and \( \Gamma_0 = \gamma_0 I_p \), where \( 0_p \) is the \( p \)-dimensional vector of zeros and \( \gamma_0 = 1 \). We denote the collection of parameters defining the CN as \( \phi_0 = \{0_p, 1\} \). Regarding the hyperparameters, we estimate \( \alpha_0 \) and \( \pi_A \) using the methods described in Section 4.4. Finally, we choose the base measure, \( G_0 \), to be the following normal-inverse gamma distribution:

\[ G_0 \equiv N(0_p, \gamma_0^2 I_p)IG(a_0, b_0), \quad (5.1) \]

where \( N \) and \( IG \) represent the normal and inverse gamma distributions, respectively. We assume
the hyperparameters $\tau_0^2$, $a_0$, and $b_0$ are known and leave approaches for estimating them as future work.

### 5.2 Experimental Conditions and the Data Generation Process

We compare the performance of the DPMM and the CNMM for the situation in which the true data generating process is a CNMM that satisfies the assumptions in Section 5.1. The comparison uses a randomly generated dataset from (4.1), fits both models, and assesses each model’s ability to discover cluster structure using the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985). The process of comparing the models is repeated using 100 randomly generated datasets. Since the goal is to determine the effectiveness of including the CN for discovering structure in the presence of anomalous data, we eliminate the potentially confounding effect of overlapping regular components. We make the additional assumption that when generating data, the spherical clusters from the normally distributed components exhibit “some” level of separation, as shown in Figure 5.1. (Later in this section, Algorithm 5.1 details a procedure for producing components separated by a minimum number of user-specified standard deviations.)
Figure 5.1: A realization of data from a 2D DPMM illustrating the simulation study’s assumptions of a mixture of spherical, well-separated normal distributions. The symbols and colors indicate component membership. The five large superimposed circles represent the two standard deviation theoretical contour for each normal component.

As a data generating tool, the CNMM requires us to specify the number of observations \((n)\), dimensionality \((p)\), CN parameters \((\phi_0)\), proportion of anomalous observations \((\pi_A)\), and hyperparameters for the DP \((\alpha_0 \text{ and } G_0)\). To learn about whether, and under which conditions, the CN is useful, we are interested in the impact \(n\), \(p\), \(\pi_A\), and \(\alpha_0\) have on each model’s ability to discover cluster structure. We allow these factors to vary, taking on the levels shown in Table 5.1. We refer to each of the 384 unique combinations of these factor levels as an experimental condition. The simulation studies use 100 randomly generated datasets for each method under each experimental condition. It is important to emphasize that the values of \(\pi_A\) and \(\alpha_0\) listed in Table 5.1 are for generating the datasets. As mentioned in Section 5.1, we treat \(\pi_A\) and \(\alpha_0\) as unknown and estimate them during the inference phase. For all simulations, we fix the remaining factors (i.e., the CN’s parameters and the base measure). We let \(\phi_0 = \{0_p, 1\}\) and choose \(G_0\) to be the normal-inverse gamma distribution shown in (5.1). The normal distribution in \(G_0\) controls the distribution of the centroids defining the components and the inverse gamma distribution controls the within-component vari-
ation. As a general rule of thumb, we want the ratio of between-component variation to be large relative to the within-component variation. We fix \( \tau_0^2 = 1 \), \( a_0 = 2.04 \) and \( b_0 = 0.0208 \). The values for \( a_0 \) and \( b_0 \) were chosen so that 95% of the generated within-component standard deviations are between 0.06 and 0.28. The fixed values for \( \phi_0 \) and \( G_0 \) are treated as known and used in both the data generating process and the modeling/inference stage.

**Table 5.1:** Factor levels for generating the datasets in the simulation studies. Each unique combination is referred to as an experimental condition.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Notation</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observations</td>
<td>( n )</td>
<td>{100, 150, 300, 900}</td>
</tr>
<tr>
<td>Dimensionality</td>
<td>( p )</td>
<td>{2, 3, 5, 10, 20, 30}</td>
</tr>
<tr>
<td>Proportion of anomalies</td>
<td>( \pi_A )</td>
<td>{0.02, 0.05, 0.15, 0.25}</td>
</tr>
<tr>
<td>Willingness to open</td>
<td>( \alpha_0 )</td>
<td>{0.5, 1, 2, 4}</td>
</tr>
</tbody>
</table>

The procedure for generating datasets from the CNMM is central to the simulation studies for comparing the methods. The process for generating a dataset, given an experimental condition, consists of five steps: (1) determining the number of components (\( K \)), (2) determining the mixture weights (\( \pi \)), (3) generating the component labels (\( c \)), (4) generating the parameters defining each mixture component’s probability density, and (5) generating the observed data. In what follows, we detail our approach to each step.

**Step 1: Determining the number of components (\( K \)).**

In order to reduce variability in our comparisons between CNMMs and DPMMs, it is convenient to compare the methods using a fixed number of components, \( K \); that is, by fixing \( K \), we expect to reduce the variability in the ARI values. Eliminating this source of variability gives us a clearer understanding of how much the CN improves results over the DPMM.

The number of components generated from a DPMM is a random variable whose distribution depends on the number of observations, \( n \), and the DP’s concentration parameter, \( \alpha_0 \). Specifically, Liu (1996) establishes that:
\[
\mathbb{E}[K \mid \alpha_0, n] = \sum_{i=1}^{n} \frac{\alpha_0}{\alpha_0 + i - 1},
\]
where \( \mathbb{E} \) represents the expectation. As \( n \to \infty \), the expected number of components can be approximated by:

\[
\mathbb{E}[K \mid \alpha_0, n] \approx \alpha_0 \log(n). \tag{5.2}
\]

In our study, we deterministically fix \( K \) by rounding the approximation in (5.2) to the nearest integer. We take this approach to remove any potential confounding in the ability of the methods to cluster, caused by differences in the number of components. For instance, if \( K \) is random, different values of \( K \) change the number of parameters that we estimate. Assuming \( n \) is fixed, estimating more parameters could negatively affect our ability to discover structure. By fixing \( K \), we are trying to eliminate this source of uncertainty.

**Step 2: Determining the mixture weights (\( \pi \)).**

The \( K \)-vector of mixture weights, \( \pi = \{\pi_0, \ldots, \pi_{K-1}\}' \), gives the proportion of the population associated with each of the \( K \) mixture components. Since every experimental condition stemming from Table 5.1 calls for some anomalies (\( \pi_A > 0 \)), one of the \( K \) components in the data generation phase must represent the CN. For notational convenience, we assume that the CN is the component indexed with a 0. Thus, \( \pi_0 = \pi_A \) and the proportion of non-anomalous, henceforth referred to as regular, data is \( 1 - \pi_A \). Due to the rich-get-richer nature of the DP, \( 1 - \pi_A \) is unevenly split among the remaining \( K - 1 \) clusters. Inspired by the stick-breaking representation of a DP (Sethuraman, 1994), which uses random draws from the \( \text{Beta}(1, \alpha_0) \) distribution to compute the mixture weights and the truncated DP, we find the remaining weights using the following steps:

1. For \( i \in \{1, \ldots, K-2\} \), find the mixture weights for all but the last regular component using:

\[
\pi_i = (1 - \pi_A) \cdot \alpha_0^{i-1} \cdot \mathbb{E}[\text{Beta}(1, \alpha_0)]^i. \tag{5.3}
\]
Equation (5.3) modifies (3.6) by scaling the weights by \((1 - \pi_A)\) to account for the weight already assigned to the CN.

2. Since we know the total number of components, we truncate the stick-breaking process and assign the proportion of \(1 - \pi_A\) remaining to the final regular component using:

\[
\pi_{K-1} = 1 - \pi_A - \sum_{i=1}^{K-2} \pi_i.
\]

**Step 3: Generating the component labels \((c)\).**

The collection of component labels, \(c = \{c_1, \ldots, c_n\}\), contains the label for each observation. Because of the calculation in Step 1., we want \(c\) to contain exactly \(K\) non-empty groups. Hence, under this study, we are assured that at least one observation is assigned to each of the \(K\) components. The remaining \(n - K\) observations are randomly assigned to the \(K\) components using a random draw from a multinomial distribution with parameter \(\pi\) found in Step 2.

**Step 4: Generating the parameters defining each mixture component’s probability density.**

There are two types of parameters defining the \(k^{th}\) component: (1) the mixture weight \((\pi_i\) established in Step 2 and (2) the parameters defining the probability density associated with the \(k^{th}\) mixture component \((\phi_k)\). We assume parameters defining the CN are known and given by \(\phi_0 = \{0_p, I_p\}\). The regular components are normally distributed, with the \(k^{th}\) component’s parameters, \(\phi_k = \{\mu_k, \sigma_k^2\}\), sampled from the base measure \(G_0\). We choose the base measure to be the normal-inverse gamma distribution given in (5.1).

We wish to explore the effectiveness of the clustering methods for well-separated components. When generating the regular components, we limit the overlap between components heuristically by checking the distance between centroids. If the proposed centroid is within a user-specified number of standard deviations of any existing centroid, we reject the proposal. Algorithm 5.1 outlines the procedure. For our data generation, we used Euclidean distance as the measure of distance between centroids and \(\eta = 4\). This allows a small amount of tail overlap but keeps the components reasonably separated.
Algorithm 5.1: Generating Well-separated Components

Initialize: Specify number of components $K$, base measure $G_0$, and minimum allowable number of standard deviations between clusters, $\eta$

Generate $1^{st}$ Component: $\phi_1 \sim G_0$

for $k \leftarrow 2$ to $K - 1$ do

Proposal Step: Propose parameters for $k^{th}$ component: $\phi_k^* \sim G_0$

Check Distances Step:

for $j \leftarrow 1$ to $(k - 1)$ do

Compute distance between proposed and $j^{th}$ centroids: $d \equiv dist(\mu_k^*, \mu_j)$

Compute maximum standard deviation between proposed and existing standard deviation:

$\sigma \equiv \max(\sigma_k^*, \sigma_j)$

Determine if new centroid is too close: If $d < \eta \sigma$, go back to Proposal Step

end

Assign Parameters: $\phi_k = \phi_k^*$

end

Step 5: Generating the observed data.

The final step is to randomly generate the data using the information from the previous steps. Each observation with component label $c_i = k$ gets assigned to the $k^{th}$ component and receives the parameter $\phi_k$. We randomly generate the $i^{th}$ observation, $x_i$, using:

$$x_i | c_i = k \sim f_k(x_i | \phi_k).$$

If $c_i = 0$, where 0 corresponds to the Cauchy component, then $f_0$ is a Cauchy density. For all other component labels, $f_k$ is a normal density.

5.3 User-specified and Effective Proportions of Anomalies

The data generating process enables the creation of random datasets for use when comparing the methods. Our goal is to understand whether, and under which experimental conditions, the addition
of the CN improves our ability to discover structure when clustering in the presence of anomalies. Although we fix a level for $\pi_A$, decisions about the remaining factors ($n$, $p$, and $\alpha_0$) also contribute to the proportion of anomalies. In this section, we explore the difference between the user-specified proportion of anomalies, $\pi_A$, and the Effective Proportion of Anomalies (EPA).

We motivate the reasoning for the difference between $\pi_A$ and the EPA using Figure 5.2. Figure 5.2 shows four datasets randomly generated using $n = 150$, $p = 2$, and $\pi_A = 0.05$, where the only difference is the value of $\alpha_0$. The solid blue circles represent the true anomalies, and anomalies on the frame of the plotting window exist beyond the truncated range of the plot. Observations belonging to the same component share a common plotting symbol, and all regular components share a common color. For a fixed $n$, increasing $\alpha_0$ produces more components but decreases the number of observations assigned to each component. When the number of observations in a component becomes small relative to the dimensionality, these regular observations (i.e., ones generated from one of the normal components and not technically anomalous) are indistinguishable from the true anomalies. For example, the plot for $\alpha_0 = 4$ depicts multiple components containing only a pair of observations (e.g., the 2 ! points, the 2 # symbols, the 2 ⊕ signs, etc.) that, if shown without plotting labels, appear anomalous. Thus, observations in sparsely populated regular components contribute to the anomalies, thereby raising the EPA beyond $\pi_A$. 
Figure 5.2: Four randomly generated datasets using $n = 150$, $p = 2$, and $\pi_A = 0.05$ but different levels of $\alpha_0$. The solid pale purple circles represent the true anomalies. The observations in each regular component share a plotting symbol (e.g., $\boxplus$, $\ast$, $+$, etc.). Components containing a small number of observations relative to the dimensionality are indistinguishable from the anomalies, thereby raising the Effective Proportion of Anomalies (EPA) beyond $\pi_A$. 

$\pi_A = 0.05$

$\alpha_0 = 0.5$  $\alpha_0 = 1$

$\alpha_0 = 2$  $\alpha_0 = 4$
The dimensionality, \( p \), also contributes to the EPA. In the first two steps of the data generating process outlined in Section 5.2, the number of components and the mixture weights are deterministic functions of \( n \), \( \alpha_0 \), and \( \pi_A \). In conjunction with the distributional assumptions, \( p \) establishes the size of the component-defining parameter. The simulation assumption of normality with a spherical covariance of \( \sigma_k^2 I_p \) implies that \( p + 1 \) parameters define each regular component. Based on our empirical work, regular components with fewer than \( 1.5(p + 1) \) observations prove difficult to consistently discover since there is not enough signal to accurately estimate the parameters. Using this threshold, we refer to regular components containing fewer observations as “insubstantial” components, while those with more are “substantial.” We expect the observations comprising insubstantial components to be indistinguishable from the anomalies, which aligns with the idea that observations are anomalous when there is not enough data to substantiate a strong pattern.

We connect the notion of insubstantial components to the EPA using the example provided in Table 5.2. For the fixed setting of \( \alpha_0 = 2 \), \( \pi_A = 0.25 \), and \( n = 300 \), the data generating process yields 10 regular components and the Cauchy component (indexed by 0). The Cauchy component receives weight \( \pi_A \) and the remaining weights follow from Step 2 in Section 5.2. Each component’s expected number of observations, shown in the Count row of the table, is the product of \( n \) and the mixture weight. The values in the first three rows of the table are independent of \( p \); however, classification as substantial or insubstantial depends on the size of the component-defining parameter, which is a function of \( p \), and the expected count. Using a threshold of \( 1.5(p + 1) \), the remaining rows of Table 5.2 identify the substantial and insubstantial components using \( \checkmark \) and \( \times \), respectively. The Cauchy component contains no markings because the classification only applies to regular components. The EPA for each value of \( p \), shown in the final column of Table 5.2, is found by adding the weights corresponding to the insubstantial components and the weight for the Cauchy component. The bottom row illustrates the possibility that all observations are indistinguishable from the anomalies when \( n \) is small relative to the number of components and dimensionality.
Table 5.2: Example showing connection between insufficient components and effective proportion of anomalies (EPA). The component labels, mixture weights, and expected number of observations are shown in the first three rows. Using the expected counts and the threshold $1.5(p + 1)$, components are classified as sufficient (✓) or insufficient (✗). The EPA shown in the last column is found by summing the weights corresponding to the insubstantial components and the Cauchy.

<table>
<thead>
<tr>
<th>Component</th>
<th>0*</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>EPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>0.25</td>
<td>0.25</td>
<td>0.17</td>
<td>0.11</td>
<td>0.07</td>
<td>0.05</td>
<td>0.03</td>
<td>0.02</td>
<td>0.02</td>
<td>0.01</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>Count</td>
<td>75</td>
<td>75</td>
<td>51</td>
<td>33</td>
<td>21</td>
<td>15</td>
<td>9</td>
<td>6</td>
<td>6</td>
<td>3</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

- $p = 3$: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ 0.26
- $p = 5$: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ 0.32
- $p = 10$: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ 0.40
- $p = 30$: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ 0.58
- $p = 50$: ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ 1.00

*Cauchy component

Figure 5.3 expands the EPA calculations to a larger subset of the factor levels used in our simulation study. In the subplot for $\alpha_0 = 1$ and $p = 5$, the EPA equals $\pi_A$, regardless of the level for $n$. This pattern indicates an absence of sparsely populated components from the data generating process. Although not shown, the same pattern is also typical in the $\alpha_0 = 0.5$ case, regardless of $p$. For a given value of $\alpha_0$ (i.e., within a column), increasing $p$ raises the EPA, but the impact is less severe for larger values of $n$. For a given $p$ (i.e., within a row), increasing $\alpha_0$ raises the EPA, but, again, the impact is lessened by having more observations. The EPA reaches 1 in a number of instances, most notably when $n$ is small and both $p$ and $\alpha_0$ are large. This suggests that small datasets with a large number of components in high dimensional space lack substantial components. The components contain too few observations to make them indistinguishable from being anomalous and the methods will struggle to discover structure in such scenarios.
Figure 5.3: An illustration of the effective proportion of anomalies (EPA) under a variety of experimental conditions. The EPA is considerably larger than \( \pi_A \) in cases where the number of components is large (i.e., \( \alpha_0 \) is large), the dimensionality is high, and the number of observations is small. In some instances, the EPA reaches 1, which suggests that all observations are indistinguishable from the anomalies.
5.4 Implementation for Simulation Studies

In this section, we discuss the implementation of the CNMM and DPMM for performing inference on the simulated datasets. In particular, we describe the initialization procedures and tailor the Gibbs Sampling with Auxiliary Parameters algorithm to the assumptions of the simulation studies.

The process begins with a standardized \( n \times p \) dataset \( X \), where each column has a mean of 0 and a variance of 1. We expect \( X \) to contain some proportion of anomalies and compare the performance of the CNMM and DPMM methods. As in the data generation process, an unknown number of \( p \)-dimensional normal distributions comprise the DPMM and the regular portion of the CNMM. Indexing the regular components using \( 1, \ldots, K \), the \( k^{th} \) component’s normal density, \( f_k \), depends on mean, \( \mu_k \), and covariance of the form \( \Sigma_k = \sigma^2_k I_p \). Denote the collection of parameters defining the \( k^{th} \) normal component as \( \phi_k = \{ \mu_k, \sigma^2_k \} \). We select the base measure, \( G_0 \), to be the normal-inverse gamma distribution given by:

\[
G_0 \equiv N(0_p, \tau^2_0 I_p)IG(a_0, b_0), \tag{5.4}
\]

where the hyperparameters \( \tau^2_0, a_0, \) and \( b_0 \) are known and match those used in the data generating process. In the case of the CNMM, we index the Cauchy component using \( 0 \) and use the known collection of Cauchy parameters given by \( \phi_0 = \{ 0_p, 1 \} \).

The CNMM and DPMM inference procedures require an initialization of the collection of component labels, \( c \), and the component-defining parameter values, \( \phi \). For the initial component labels, we consider using: (1) the true labels, (2) singleton components, and (3) the \( K \)-means assignments. When the components have well-separated centroids, the three methods produce similar results; however, initializing to the truth is infeasible when the true labels are unknown and \( K \)-means is susceptible to anomalies. As a result, we initialize each observation as a singleton component with the mean and covariance set to the observation’s value and identify matrix, respectively. The CNMM procedure also requires an initialization of the CN, which we initialize as a singleton containing the observation with the most severe value in any dimension.
Following the initialization step, inference for both models employs a Gibbs sampler that alternates between updating $c$ and $\phi$. The simulation studies assume that the densities of the regular components, $f_k$, are normal and the base measure, $G_0$, is the normal-inverse gamma distribution in (5.4). Due to the choice for $G_0$, we implement the Gibbs Sampling with Auxiliary Parameters algorithm discussed in Section 3.5.2 (for the DPMM) and Section 4.3 (for the CNMM). We tailor the algorithm to accommodate the simulation assumptions using three modifications. The first modification occurs in the step for updating $c$. Updating an individual component label requires a draw from a categorical distribution with probabilities given by:

$$\Pr(c_i = k \mid x_i, \phi^{-i}, \tilde{\phi}, c^{-i}) = \pi_{i,k}, \quad \text{for } k = 0, \ldots, K^{-i} + m.$$  

Previously, the values of $\pi_{i,k}$ depended on the general density $f_k$. Replacing $f_k$ with the normal density assumed in the simulation studies produces the following probabilities:

$$\pi_{i,k} \propto \begin{cases} 
\pi_A \cdot C(x_i \mid \phi_0) & \text{if } k = 0, \\
(1 - \pi_A) \cdot \frac{n_k^{-i}}{\alpha_0 + n - n_0^{-i} - 1} \cdot N(x_i \mid \phi_{k}^{-i}) & \text{if } 1 \leq k \leq K^{-i}, \\
(1 - \pi_A) \cdot \frac{\alpha_0}{\alpha_0 + n - n_0^{-i} - 1} \cdot \frac{m}{n_0^{-i}} \cdot N(x_i \mid \tilde{\phi}_k) & \text{if } K^{-i} + 1 \leq k \leq K^{-i} + m, 
\end{cases} \quad (5.5)$$

where $C$ and $N$ represent the Cauchy and normal densities, respectively. The formula in (5.5) is specific to the CNMM; however, the same result holds for a DPMM if we set $\pi_A = 0$, thereby eliminating the possibility of assigning the observation to the CN. Consequentially, none of the observations are considered anomalous, making $n_0^{-i} = 0$. The second modification, which is the same for the CNMM and DPMM, applies to the step for updating the $k^{th}$ component’s parameter, $\phi_k = \{\mu_k, \sigma_k^2\}$. The joint posterior distribution for $\phi_k$ follows from the normality of $f_k$, the base measure, and all of the data assigned to the $k^{th}$ component, $X_k$. We update $\phi_k$ in two steps using the full conditional posterior distributions for $\mu_k$ and $\sigma_k^2$. First, assuming $c$ and $\sigma_k^2$ are known, we update the $k^{th}$ component’s centroid, $\mu_k$, using:
The posterior distribution in (5.6) follows directly from the conjugacy of the normal portion of $G_0$ when the likelihood is also normal. Second, assuming $c$ and $\mu_k$ are known, update the $k^{th}$ component’s variance using:

$$\sigma_k^2 \mid X_k, c, \mu_k \sim IG(\tilde{a}_0, \tilde{b}_0), \quad (5.7)$$

where

$$\tilde{a}_0 = a_0 + \frac{n_k}{2},$$

$$\tilde{b}_0 = b_0 + \frac{1}{2} \sum_{\{i \in c_i = k\}} (x_i - \mu_k)^T(x_i - \mu_k).$$

The posterior distribution in (5.7) follows directly from the conjugacy of the inverse gamma portion of $G_0$ when the likelihood is normal. The third modification regards updating the hyperparameters. We update the concentration parameter, $\alpha_0$, using (4.8) from Section 4.4.1. Although this formula is framed in the context of the CNMM, the approach also works for the DPMM by setting $n_0 = 0$. Finally, as discussed in Section 4.4.2, we employ both empirical and probabilistic approaches for updating $\pi_A$. The simulation assumptions do not affect the empirical estimate for $\pi_A$ given in (4.9). For the probabilistic approach, we assume that all components are equally likely, a priori. This allows us to rewrite (4.10) as:
\[ \hat{\pi}_{i,k} \equiv \Pr(c_i = k \mid x_i, \phi) = \begin{cases} 
\frac{C(x_i \mid c_i = 0, \phi_0)}{C(x_i \mid c_i = 0, \phi_0) + \sum_{k=1}^{K} N(x_i \mid c_i = k, \phi_k)} & \text{if } k = 0, \\
\frac{N(x_i \mid c_i = k, \phi_k)}{C(x_i \mid c_i = 0, \phi_0) + \sum_{k=1}^{K} N(x_i \mid c_i = k, \phi_k)} & \text{if } 1 \leq k \leq K. 
\end{cases} \]

The corresponding estimate of \( \pi_A \) is given by:

\[ \hat{\pi}_A = \frac{\sum_{i=1}^{n} \hat{\pi}_{i,0}}{n}. \quad (5.8) \]

Putting all three modifications together, Algorithm 5.2 summarizes the Gibbs Sampling with Auxiliary Parameters algorithm for our simulation study.
Algorithm 5.2: Gibbs Sampling with Auxiliary Parameters - Simulation Study

Select number of auxiliary parameters: $m$

Current State: Component labels ($c$), unique parameters ($\phi$), number of components ($K$), and component counts ($n$)

Step 1: Update Component Labels

for $i \leftarrow 1$ to $n$

  Remove observation $i$ from its current component

  Determine the reduced state ($c^{-i}$, $\phi^{-i}$, $K^{-i}$, $n^{-i}$), reindexing if necessary

  Generate Auxiliary Points

  if $i$ is a singleton then
   
   Assign the current parameter to one of the auxiliary components ($\phi_{K-i+1} = \theta_i$)

   Draw remaining auxiliary parameters independently from $G_0$

   ($\phi_{K-i+2}, \ldots, \phi_{K-i+m} \sim G_0$)

  else
   
   Draw auxiliary parameters independently from $G_0$ ($\phi_{K-i+1}, \ldots, \phi_{K-i+m} \sim G_0$)

  end

  Draw a new component label, $c_i$, from a categorical distribution with probabilities from (5.5)

  if $c_i$ corresponds to auxiliary component then
   
   Relabel $c_i = K^{-i} + 1$

  end

  Update current state to reflect changes in $c$, $\phi$, $K$, and $n$

  Delete all empty auxiliary components

end

Step 2: Update Component-Specific Parameters

for $k \leftarrow 1$ to $K$

  Update the centroid: Sample $\mu_k$ from $\mu_k | X_k, c, \sigma_k^2$ using (5.6)

  Update the variance: Sample $\sigma_k^2$ from $\sigma_k^2 | X_k, c, \mu_k$ using (5.7)

end

Step 3: Update Hyperparameters

Update $\alpha_0$ using (4.8). (For a DPMM, $n_0 = 0$ in (4.8).)

Update $\pi_A$ using Empirical approach (4.9) or Probabilistic approach (5.8). (For a DPMM, skip.)
5.5 Simulation Studies

In this section, we start with an in-depth analysis of the CNMM and DPMM methods using a 2D dataset, which allows us to visualize the results. Through visualization, we ensure that the discovered clusters are coherent and gain insight into the ways the models handle anomalous data. The section concludes with an analysis of the full-scale simulation studies.

5.5.1 Illustrative Results

Figure 5.4 depicts a randomly generated dataset produced under the experimental condition corresponding to $n = 300, p = 2, \alpha_0 = 2,$ and $\pi_A = 0.25$. We truncate the plot’s axes to avoid extreme anomalous observations from interfering with the scale of the plot. Observations belonging to the same component share both the true component label (e.g., 0, 1, ..., $K$) as the plotting symbol and a plotting color. The superimposed circles represent the theoretical three standard deviation contour for each normally distributed component. Each circle’s color matches the plotting color for observations assigned to the component. For consistency with past notation, the anomalous observations from the CN are assigned a label of “0.”

Figure 5.4 illustrates variation in the number of observations assigned to each component. The largest component (true component 1) consists of 75 observations, while the smallest (true component 8) contains only 2 observations. The true proportion of anomalies matches the highest level for any experimental condition in the study ($\pi_A = 0.25$), and, in this case, the dataset contains 72 anomalies. The bottom row of Table 5.3 and Table 5.4 show the true number of observations assigned to each component.

Visualizing the dataset reveals several situations for which component membership is uncertain. First, components with very low density relative to the dimensionality (i.e., the components containing as few as two or three observations in 2D space) are nearly indistinguishable from the anomalies. Second, several observations lie within the mass of another component. For example, at least three anomalies reside inside of the three standard deviation contour for the component
Figure 5.4: A randomly generated dataset corresponding to the experimental condition $n = 300$, $p = 2$, $\alpha_0 = 2$, and $\pi_A = 0.25$. Observations belonging to the same component share both the true component label as the plotting symbol and a plotting color. The superimposed circles depict the theoretical three standard deviation contour for the ten regular components.

labeled 3. Also for component 3, a regular observation from component 8 lies in the overlap of the tails of the two distributions. Most clustering methods fail to consistently recover the true labels when faced with distinguishing low density regular observations from anomalies and when assigning component memberships for observations existing in overlapping regions.

Inference using CNMMs and DPMMs relies on MCMC approaches. We employ Gibbs sampling for making probabilistic updates of the component labels, $c$, and component-defining parameters, $\phi$. Due to the stochastic nature of Gibbs, an observation’s component label potentially changes during any given iteration. Additionally, due to the possibility of opening and closing components, the meaning of the labels may change. For these reasons, we start with a comparison of the methods by exploring the component labels at snapshot in time (i.e., at the end of a given Gibbs iteration). We distinguish between the true labels assigned by the data generating process and
those discovered through the inference process. We refer to “true component $x$” or “true label $x$” as “true $x$,” where $x \in \{0, \ldots, K\}$. Similarly, “discovered $x$” is a shortened form for “discovered component $x$” or “discovered label $x$.” Figure 5.5 shows the snapshot results of the CNMM and DPMM approaches. Both methods perform well on this iteration and discover the substantial components (i.e., true 1-7). The biggest difference between the methods lies in the handling of the anomalies. At this moment, the CN captures 67 of the 72 anomalies. Of the 5 remaining, two exist as singletons and one each joins true 2, 4, and 7. None of the latter three raise concerns because the anomalies lie near the three standard deviation contour for each of their respective assignments. On the other hand, the DPMM divided the anomalous observations among 12 clusters, three of which are substantial components (discovered 8, 9, and 10 in the DPMM solution).

**Figure 5.5:** A comparison of the CNMM and DPMM methods by exploring the discovered component labels at the end of a given Gibbs iteration. The methods produce similar results but handle the anomalies differently.

The DPMM’s handling of the anomalies raises three potential concerns. First, the DPMM exhibits a greater number of discovered clusters than the CNMM, largely due to the DPMM’s splitting of the anomalies into 12 components. The inflated number of components has a self-reinforcing
effect. As the number of discovered components increases, the estimate for \(\alpha_0\) increases. This leads to a serious concern that the inflated value of \(\alpha_0\) could result in a splitting of one of the substantial components. Second, three of the discovered DPMM components cover a wide range of the space. For instance, discovered 8, 9, and 16 cover nearly the entire span of the plot window. As the DPMM assigns distant observations to the same component, the component’s variance increases. Observations residing in the space between the distant observations now exist in the mass of the component, which can lead to a black hole effect. In the CNMM, the CN covered a wide range but the remaining components are compact. Further, distant observations in the CN do not affect the CN’s scale, as it is fixed by the researcher. Third, if we plan on using the components for predictive modeling, it is dangerous to base a model on a collection of largely anomalous observations. As an example in the DPMM result, discovered 8 contains 33 anomalies and 4 regular observations. Although we could blindly arrive at a prediction, the variance, and corresponding level of uncertainty, would be extremely high. The CNMM solves all three of these concerns by isolating the anomalies in the CN, flagging them for further inspection, and not forcing the anomalies into components defined by the regular observations.

Representing component labels graphically proves challenging, if not impossible, as \(n\) and/or \(p\) increases; however, tabular summaries offer an alternative. Table 5.3 and Table 5.4 represent the same CNMM and DPMM component assignments shown in Figure 5.5a and Figure 5.5b, respectively. Such tables provide several pieces of information. First, the column totals represent the number of observations assigned to each true component during the data generation process. Second, each row depicts the composition of a discovered component where the cell entries show the frequencies of the true labels for the observations comprising the discovered component. The column label for the highlighted cell in each row indicates the most prevalent true component in the discovered component. For example, discovered 2 in Table 5.3 consists of one true anomaly and 9 regular observations from true 7. Since true 7 makes up the majority of the observations in the discovered 2, the cell is highlighted. Third, each column describes the allocation of a true component’s observations to the discovered components. For instance, true 2 in Table 5.3 consists of 47 observations, all of which are allocated to discovered 4. Similarly, true 10 consists of 3 observations, where 2 belong to discovered 10 and 1 belongs to discovered 5. Finally, a perfect
recovery of the component labels produces a square table in which each row and column has exactly one non-zero element.

**Table 5.3**: Tabular summary of the component labels discovered by the CNMM and shown in Figure 5.5a. The column totals represent the number of observations assigned to each true component. Each row depicts the composition of a discovered component where the cell entries show the frequencies of the true labels for the observations comprising the discovered component. The column label for the highlighted cell indicates the most prevalent true component in the discovered component. Each column describes the allocation of the true component’s observations to the discovered components.

<table>
<thead>
<tr>
<th>True Component Labels</th>
<th>0*</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>47</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>48</td>
</tr>
<tr>
<td>5*</td>
<td>67</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

*Cauchy-Net
Table 5.4: Tabular summary of the component labels discovered by the DPMM and shown in Figure 5.5b. The column totals represent the number of observations assigned to each true component. Each row depicts the composition of a discovered component where the cell entries show the frequencies of the true labels for the observations comprising the discovered component. The column label for the highlighted cell indicates the most prevalent true component in the discovered component. Each column describes the allocation of the true component’s observations to the discovered components.

<table>
<thead>
<tr>
<th>True Component Labels</th>
<th>0*</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>72</td>
<td>75</td>
<td>47</td>
<td>32</td>
<td>22</td>
<td>18</td>
<td>17</td>
<td>9</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>300</td>
</tr>
</tbody>
</table>

*Cauchy-Net
Instead of qualitatively analyzing the results using plots of the component labels and/or tabular summaries, we can quantify the results of each Gibbs iteration using a single number: the adjusted Rand index (ARI) (Hubert and Arabie, 1985). The ARI compares the true and discovered labels to assess how well the method recovered the true structure. The maximum value for the ARI is 1, which indicates complete agreement between the discovered and true labels. As the ARI values move further away from 1, the method is recovering less structure. The discovered components shown in Figure 5.5 produce ARIs of 0.944 and 0.807 for the CNMM and DPMM, respectively.

Exploring the results of a single Gibbs iteration is impractical for assessing the overall performance of either method. By visual investigation of the trace plots depicted in Figure 5.6, we are able to see that the CNMM consistently outperforms the DPMM over the entire Gibbs procedure. Both plots indicate convergence, as the ARI values are bouncing around a typical value. Since the plots use the same scale for the ARI axis, we see that the CNMM is consistently recovering more structure than the DPMM.

![ARI for the CNMM](image1.png)

![ARI for the DPMM](image2.png)

**Figure 5.6**: The adjusted Rand index (ARI) versus the Gibbs sampling iteration for the CNMM and DPMM.
We have several options for quantitatively summarizing the overall performance of the methods. First, if we want to know how the methods performed individually, we can compute the mean ARI post burn-in for each method. In this example, the mean ARIs for the CNMM and DPMM are 0.852 and 0.757, respectively. These values suggest that both methods are performing reasonably well, but the CNMM is recovering more structure than the DPMM. Second, if our goal is to determine whether, and by how much, adding the CN helped to recover structure, we can compute the difference of the mean ARIs post burn-in. Subtracting the DPMM value from the CNMM value in this example gives 0.095, which quantifies the gain achieved by including the CN. The limitation of using the difference in the means is that we are unable to immediately discern if either method recovers structure; however, we clearly see that the CNMM dominates the DPMM.

For the analysis of the large-scale simulation studies in the next section, we compare the performance of the methods on 38,400 randomly generated datasets. In order to reduce the results to a manageable amount of information, we summarize the overall performance of a method on a given dataset using the mean ARI post burn-in or the difference of the mean ARIs post burn-in.

### 5.5.2 Analysis of Simulation Studies

In this section, we present the results of the simulation studies for investigating the effectiveness of the CNMM for clustering in the presence of anomalous data. The results are based on randomly generating 100 datasets for each of the 384 experimental conditions, which follow from using all combinations of the factor levels in Table 5.1. Each of the 38,400 datasets are analyzed using three methods: (1) a DPMM, (2) a CNMM where \( \pi_A \) is estimated through the empirical approach, and (3) a CNMM where \( \pi_A \) is estimated through the probabilistic approach. Our primary goal for comparing the methods is to address whether, and under which conditions, augmenting the DPMM with the CN is useful for clustering in the presence of anomalies. We investigate the effectiveness of the CNMM under a variety of experimental conditions by exploring whether:

1. there is a difference between the performance the CNMM and the DPMM, as measured through the difference of the mean ARI values.
2. The CNMM and the DPMM are viable options for recovering clustering structure, as measured through the mean ARI values.

Our secondary goal is to determine whether the method for estimating $\pi_A$ plays a significant role in the performance of the CNMM. We investigate this goal by examining the difference of the mean ARI values for the two CNMM approaches. In what follows, we provide a detailed analysis for the $n = 300$ and $n = 900$ cases. The comparable plots for the $n = 100$ and $n = 150$ cases are found in Appendix A and demonstrate similar patterns.

Figure 5.7 contains boxplots of the differences in the mean ARI for the CNMM and DPMM for datasets based on 300 observations. We compute the differences by subtracting the ARI of the DPMM from the ARI of the CNMM for analyses based on the same dataset. Each boxplot represents the differences for 100 randomly generated datasets, but we have removed outliers so as not to distract from the overall patterns. For the interested reader, the boxplots containing the outliers are shown in Appendix B. Figure 5.7 consists of a grid with six rows and four columns, resulting in a total of 24 subplots. Each subplot displays four boxplots, corresponding to the true levels of $\pi_A$ labeled on the bottom horizontal axis. Each row of the grid represents a level for the dimensionality and is labeled on the right vertical axis. Similarly, the columns correspond to the levels for the true value of the DP’s concentration parameter, $\alpha_0$, and are labeled on top horizontal axis. As $\alpha_0$ increases, the number of components, $K$, also increases, and we can think of the columns as representing an increase in $K$ as we move to the right.

Boxplots separated from 0, in either direction, represent situations in which the CNMM and DPMM perform differently. In all cases, the boxplots are entirely above, or hover about, 0. Based on the order of subtraction, this suggests that the CNMM is either outperforming the DPMM or performing comparably. The majority of the cases where the CNMM is outperforming the DPMM occur in the higher dimensional situations and when the proportion of anomalies is high. The $p = 30$ row, and to some degree the $p = 20$ row, exhibit a “peaked” behavior in the sense that the differences are small at the beginning and end of the row but larger in the middle. Initially, this result seems counterintuitive, but it is explained by the EPA. According to Figure 5.3, the EPA is 1 for $p = 30$, $\pi_A = 0.25$, $\alpha_0 = 4$, and $n = 300$. Effectively, all of the observations are indistinguish-
able from the anomalies and neither method works well. Finally, the situations where the methods perform most comparably occur in the low $p$, low $\pi_A$, and low $\alpha_0$ cases.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.7}
\caption{Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size ($n = 300$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using $\text{ARI}(\text{CNMM}) - \text{ARI}(\text{DPMM})$.}
\end{figure}
In certain situations, the boxplot of the difference in the mean ARI is away from 0, which suggests a difference in the methods; however, when the upper boundary (either the top of the box or the end of the whisker) remains small, we lack insight into whether the methods successfully recovered structure. For example, the boxplot for $p = 30$, $\pi_A = 0.05$, and $\alpha_0 = 0.5$ in Figure 5.7 ranges from 0.01 to 0.17 with a median line at 0.06. We are unable to distinguish between the following situations: (1) the typical mean ARI values for the CNMM and DPMM are 0.09 and 0.03, respectively or (2) the typical mean ARI values for the CNMM and DPMM are 0.92 and 0.86, respectively. In the first situation, both methods fail to recover much structure, whereas in the latter situation, both perform well. For this reason, we also explore boxplots of the ARI by method to investigate whether the methods are viable options for recovering structure.

Figure 5.8 compares the ARI values for the CNMM and DPMM methods. The format of the plot is similar to Figure 5.7 with two exceptions. The left vertical axis represents the mean ARI values, rather than the difference in mean ARI. Second, each subplot contains a pair of boxplots for every level of $\pi_A$, one for each method. Figure 5.8 reinforces many of the results from the plot of the differences in mean ARI. Namely, the CNMM is either outperforming the DPMM or performing comparably, with the biggest advantage in the higher dimensional situations and when the proportion of anomalies is high. In terms of ability to recover structure, the CNMM achieves ARI values near 1 for all circumstances with $\alpha_0 = 0.5$. The DPMM performs similarly in the case of $\alpha_0 = 0.5$ but experiences a decrease in performance and an increase in variability as $p$ and $\pi_A$ increase. When $\alpha_0 \geq 1$, we see a degradation of the ARI values within a subplot (i.e., as $\pi_A$ increases), within a row (i.e., as $\alpha_0$ increases), and within a column (i.e., as $p$ increases). This result is expected as all three of those situations complicate the clustering problem and increase the EPA. The important realization is that the DPMM’s performance degrades faster and exhibits more variability than that of the CNMM.
Matthew D. Slifko

Chapter 5. Simulation Studies

Figure 5.8: Comparison of the ARI using a fixed sample size ($n = 300$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$).

Figure 5.7 and Figure 5.8 establish the CNMM as a useful tool for clustering in the presence of anomalous data; however, we still need to explore whether the approach for estimating $\pi_A$ has a
large impact on the performance of the CNMM. We consider the empirical and probabilistic approaches, both of which are described in Section 5.4. Figure 5.9 displays the boxplots corresponding to the difference in mean ARI for the two methods. We compute the difference by subtracting the ARI of the empirical approach from the ARI of the probabilistic approach. In higher dimensions, the boxplots hover around 0 and exhibit very little variability, regardless of the levels for $\alpha_0$ and $\pi_A$. This implies that the different estimation techniques have little impact on the CNMM in the higher dimensional spaces. Based on the order of subtraction, the empirical approach enjoys a small advantage in the low dimensional cases of $p = 2$ and $p = 3$, provided $\pi_A$ takes on its highest value. Overall, the estimating procedure for $\pi_A$ appears to have little impact on the performance of the CNMM.
Figure 5.9: Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size ($n = 300$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using ARI(CNMM Probabilistic) - ARI(CNMM Empirical).
In order to improve our understanding of the conditions for which the CNMM is a useful tool, we repeat the analyses using the same type of plots but for \( n = 100, 150, \) and 900. We show the summary for \( n = 900 \) and the plots for the other values of \( n \) are included in Appendix A.

Figure 5.10 shows the boxplots for the difference in the mean ARI values based on the randomly generated datasets containing 900 observations. We compute the differences by subtracting the ARI for the DPMM from the ARI for the CNMM. As with \( n = 300 \), all of the boxplots are entirely above, or hovering about, 0. This supports the previous results and provides strong evidence that the CNMM is either outperforming the DPMM or performing comparably. The experimental conditions where the CNMM excels remain the same. Specifically, the CNMM is recovering more structure, as measured by the ARI, than the DPMM for the scenarios with a higher dimensionality and a higher proportion of anomalous observations.

Figure 5.10 does exhibit a few differences from its counterpart for \( n = 300 \). The differences are most noticeable in the \( p = 30 \) row. For instance, in the case of \( \alpha_0 = 4 \) and \( \pi_A \in \{0.15, 0.25\} \), the differences were small and the boxplot was only slightly above 0 in the \( n = 300 \) case. The reasoning for this follows from Figure 5.3, which reveals the EPA is 1 for \( p = 30, \pi_A = 0.25, \alpha_0 = 4, \) and \( n = 300 \). At the \( n = 300 \) level, nearly all observations are indistinguishable from the anomalies; therefore, both methods find little structure and produce ARI values near 0. As a consequence, the difference in the mean ARI is also near 0, producing boxplots near 0. We previously mentioned this result when discussing the “peaked” behavior of the \( p = 30 \) row. As we now increase the number of observations to \( n = 900 \), the EPA drops to 0.5. This implies more signal, which the CNMM successfully discovers. The DPMM discovers some structure but not as much as the CNMM, as seen in Figure 5.11; thereby, the difference in the mean ARI increases and is now above 0.25.
Figure 5.10: Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size ($n = 900$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using $\text{ARI(CNMM)} - \text{ARI(DPMM)}$. 
Figure 5.10 illustrates that the CNMM performs better than the DPMM, but fails to provide insight into how well either method worked. Figure 5.11 shows the boxplots of the ARI by method to investigate the amount of structure recovered by each model. Comparing Figure 5.11 to the corresponding plot for \( n = 300 \), we observe similar patterns. As expected, we see a degradation of the ARI values within a subplot (i.e., as \( \pi_A \) increases), within a row (i.e., as \( \alpha_0 \) increases), and within a column (i.e., as \( p \) increases). All three of these change increase the EPA, making cluster discovery more difficult. As we saw in the \( n = 300 \) case, the CNMM’s performance degrades at a slower rate and exhibits less variability than that of the DPMM.

The biggest insight from comparing Figure 5.11 to its counterpart for \( n = 300 \) is that both methods benefit from the influx of the extra data. The result is evidenced by the boxplots for both methods moving closer to 1 in the \( n = 900 \) case, for all experimental conditions and most easily seen in the \( p = 20 \) and \( p = 30 \) rows. The finding that the methods perform better with more data is a common sense result; however, the result still helps to improve our understanding that both methods are capable of recovering structure when clustering in the presence of anomalies, provided there is enough signal. The difference is that by augmenting the DPMM with the CN, we can recover more structure.

Figure 5.10 and Figure 5.11 provide evidence that the CNMM is outperforming the DPMM. As before, we investigate whether the procedure for estimating \( \pi_A \), either through the empirical or probabilistic approaches, has a significant impact on the performance of the CNMM. Figure 5.12 shows the difference in mean ARI values for the methods in the case of \( n = 900 \). These results are nearly identical to the \( n = 300 \) case, and we conclude that the estimating procedure for \( \pi_A \) appears to have little impact on the CNMM’s ability to discover cluster structure.
Figure 5.11: Comparison of the ARI using a fixed sample size \( (n = 900) \) and different levels for the proportion of anomalous observations \( (\pi_A) \), dimensionality \( (p) \), and true value of Dirichlet process concentration \( (\alpha_0) \).
Figure 5.12: Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size \((n = 900)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The difference is computed using \(\text{ARI(CNMM Probabilistic)} - \text{ARI(CNMM Empirical)}\).
The results of the comparisons of the CNMM and DPMM for clustering in the presence of anomalous data reveal that augmenting the DPMM with the CN is useful for mitigating the impact of anomalies and discovering structure. In addition to the results illustrated for $n = 300$ and $n = 900$, the experimental conditions corresponding to $n = 100$ and $n = 150$ produce similar results. The biggest difference is that the cases with fewer observations are overwhelmed by the amount of the EPA in higher dimensions. The lack of signal in relation to the dimensionality limits the amount of discoverable structure.
Chapter 6

Fairfax County Analysis

The simulation studies establish the CNMM as a viable tool for clustering under well-controlled experimental conditions. Our long-term goal is to extend the use of the CNMM for clustering and predicting using large-scale datasets. Specifically, we aim to employ Bayesian model averaging for predicting property values in the presence of anomalous housing data. As a first step, we apply the CNMM and DPMM for clustering and predicting property values using a small-scale dataset and demonstrate that the CNMM improves predictive performance.

The dataset consists of the properties associated with 434 single family homes in Fairfax County (Virginia) that sold during the year 2010. Each property represents a regular resale, as opposed to a sale for a newly constructed home or a foreclosure. The property-specific information includes the closing price for the property, the home’s location (latitude and longitude), and five hedonic features (living area in square feet, lot area in square feet, number of bedrooms, number of bathrooms, and year built). To remove any complications arising from missing data, all 434 properties in the dataset have complete information.

Using closing price as a proxy for property value, we model the natural logarithm of the property value using a hedonic first-order multiple linear regression model based on the home’s location and hedonic features. To allow for heterogeneity in the way in which the characteristics contribute to the value of the response, we employ the CNMM and DPMM. Making the typical regression
assumptions that the errors within each component are i.i.d. normal with mean 0 and constant variance, the corresponding CNMM is given by:

\[
y_i \mid \boldsymbol{x}_i, \beta_i, \sigma_i^2 \sim f_i(y_i \mid \boldsymbol{x}_i^T \beta_i, \sigma_i^2)
\]

\[
\{\beta_i, \sigma_i^2\} \mid G, \phi_0 \sim \pi_A \delta(\phi_0) + (1 - \pi_A)G
\]

\[
G \mid \alpha_0, G_0 \sim DP(\alpha_0, G_0)
\]

\[
\phi_0 = \{0_8, 1\}
\]

\[
G_0 = N(0_8, I_p)IG(a_0, b_0).
\]

In (6.1), the response, \(y_i\), represents the standardized value of the natural logarithm of the \(i^{th}\) property value and \(\boldsymbol{x}_i = \{1, x_{i1}, \ldots, x_{iT}\}^T\), where \(x_{ij}\) is the standardized value of the \(j^{th}\) characteristic for the \(i^{th}\) property. The component-specific regression parameters, namely the regression coefficients, \(\beta\), and error variance, \(\sigma^2\), distinguish the mixture components. We fix \(\phi_0\), which corresponds to the regression parameters for the CN, at \(\{0_8, 1\}\), where \(0_8\) is the 8-dimensional vector of zeros. The regression parameters for the regular components are governed by the DP with the normal-inverse gamma base measure, \(G_0\), where \(a_0\) and \(b_0\) are fixed at 2.04 and 0.0208, respectively. Depending on observation \(i\)’s regression parameters, denoted by \(\{\beta_i, \sigma_i^2\}\), \(f_i\) is either a Cauchy or normal density. Finally, the DPMM is a special case of (6.1) where the proportion of anomalous observations, \(\pi_A\), is 0.

We implement the CNMM and DPMM for the Fairfax County dataset. The standardization procedure uses the entire dataset to center and scale the response and each regressor to have a mean and variance of 0 and 1, respectively. After standardizing, we initialize each observation as a singleton component. Inference for both models employs the Gibbs Sampling with Auxiliary Parameters algorithm and alternates between updating the component labels and each component’s regression parameters. The updates for the regression parameters exploit the conjugacy of the base measure, \(G_0\), and the data model, \(f_i\). We monitor convergence of the Gibbs sampler by examining trace plots of the log likelihood and regression parameters, all of which indicated convergence.

Table 6.1 compares the number of components, \(K\), discovered by each method during the infer-
ence procedure. Using the post burn-in iterations, the CNMM finds fewer components (4.35) than the DPMM (6.24), on average, and exhibits less variability in the 95% credible intervals for $K$. To determine whether the extra components discovered by the DPMM are meaningful, we monitor the iteration on which each component initially develops. We define a “stable” component as one that materializes during the burn-in phase and remains open throughout the remaining Gibbs iterations. Stable components contain a large number of observations, relative to the dimensionality, and are largely unaffected when a portion of the observations are reassigned to other components during a given Gibbs iteration. Consequentially, the stable components allow us to generate many samples from the posterior distribution of each component’s regression parameters to learn about the underlying structure. On the contrary, components containing a small number of observations regularly develop but are greatly impacted when a portion of the observations are reassigned. Components that fail to remain open long enough to provide meaningful information are referred to as “unstable.” Table 6.1 shows that the CNMM discovers four stable components, one of which is the CN. Nearly all (roughly, 433 out of 434) observations belong to the CNMM’s stable components. Despite discovering a greater number of components than the CNMM, the DPMM only produces three stable components that account for, on average, 418 observations. The remaining observations are distributed among the small, unstable components. This result supports the simulation discovery that the DPMM has a tendency to produce small unstable components, often as a result of dividing the anomalies into many small groups.

Table 6.1: The number of components, $K$, discovered by the CNMM and DPMM using the Fairfax County housing data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean $K$</th>
<th>95% CI</th>
<th>Stable $K$</th>
<th>Mean number of observations assigned to stable components</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMM</td>
<td>4.35</td>
<td>(4, 6)</td>
<td>4</td>
<td>433.3</td>
</tr>
<tr>
<td>DPMM</td>
<td>6.24</td>
<td>(5, 9)</td>
<td>3</td>
<td>418.2</td>
</tr>
</tbody>
</table>

The estimated regression coefficients in Figure 6.1 provide insight into how the characteristics influence the natural logarithm of the property value within each stable component. Violin plots hovering about the red line at 0 indicate that the characteristic does not influence the response,
while those strictly above 0 indicate a positive effect. The further a violin plot is from 0, the stronger the effect. Regardless of the component, Figure 6.1 reveals that none of the characteristics exhibit a negative impact on the response; however, the characteristics having the strongest influence on the response depend on the component. For instance, all characteristics in the first component, except for the number of bedrooms, positively impact the response, with the location variables and the number of bathrooms having the strongest influence. This differs from the third component where the response is primarily driven by the lot and living areas.

The side-by-side violin plots for each component’s characteristics illustrate both similarities and differences between the CNMM and DPMM estimates. Both methods recover the same stable components and produce the same overall patterns regarding how characteristics influence the response; however, some differences in the estimated regression coefficients exist. As an example, the coefficient estimates for lot area and year built in the third component of the DPMM suggest a stronger influence on the response than is indicated by the CNMM.

An exploration of the component labels reveals the source of the similarities and differences in the methods. The Gibbs sampler produces a vector of stochastically generated labels for each observation, whose length is equal to the number of post burn-in iterations. For comparing the methods, we use the vector to link each observation to the single component for which it is most likely assigned during the post burn-in phase. As an example, analyzing the vector of labels for observation 178 in the CNMM reveals much uncertainty regarding the observation’s component membership. Specifically, 178 has empirical probabilities of 0.43, 0.14, 0.11, and 0.32 for being assigned to components 0 (the CN), 1, 2, and 3, respectively. Since the 0 component is the most probable, we link 178 to the CN. Based on this approach, Table 6.2 summarizes the number of observations linked to each component of the CNMM and DPMM. The “Agreement” column indicates the number of observations for which both methods produce the same the linked assignment. As an example, of the 137 observations linked to the third component in the CNMM, 132 of the observations are also assigned to the same component of the DPMM. In total, the methods agree on the linked assignments for 89% (386 out of 434) of the observations. The high level of agreement explains the reason for the similarities in the regression coefficient estimates; however, the
differences stem from the disagreements on the remaining 11%.

**Figure 6.1:** Regression coefficient estimates for the three stable regular components discovered by the CNMM and DPMM.
Table 6.2: The number of observations linked to each component in the CNMM and DPMM. The “Agreement” column indicates the number of observations for which both methods produce the same the linked assignment. As an example, of the 137 observations linked to the third component in the CNMM, 132 of the observations are also linked to the third component by the DPMM.

<table>
<thead>
<tr>
<th>Component</th>
<th>CNMM Observations</th>
<th>DPMM Observations</th>
<th>Agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>14</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>1</td>
<td>156</td>
<td>149</td>
<td>144</td>
</tr>
<tr>
<td>2</td>
<td>127</td>
<td>120</td>
<td>110</td>
</tr>
<tr>
<td>3</td>
<td>137</td>
<td>159</td>
<td>132</td>
</tr>
<tr>
<td>Unstable</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

*Cauchy-Net

The severity of the impact of the observations on the model for which the methods disagree drive the differences between the the CNMM and DPMM. In particular, when anomalous observations are forced into regular components, the anomalies have the potential to highly influence the parameter estimates and overall model performance. On the other hand, if the observations have little impact on the model, then the methods perform comparably.

To demonstrate the usefulness of the CNMM, we are interested in the identifying and handling of the anomalies. The CNMM identifies 14 observations that link to the CN. Table 6.3, which shows the percentile rank for each anomaly’s hedonic features and closing price, provides insight as to the reasons that the anomalous observations do not match the overall patterns found in the data. (The observations are listed in the order provided in Table 6.4.) Based on the violin plots in Figure 6.1, we expect larger values of the hedonics to produce larger values for the natural logarithm of the property values. The majority of the first six anomalies exhibit large values for the hedonic features but low prices. For instance, observation 427 is above the 60th percentile for all five hedonics, but the corresponding price is greater than only 1% of the observations. Similarly, observation 347 is near the max in terms of living area, bedrooms, bathrooms, and year built and above the third quartile for lot area; however, its price is only at the 71st percentile. The exception to this pattern
among the first six observations is observation 32 whose hedonics are below average, aside from lot size, but whose price is at the 90th percentile. Observation 62 is large in nearly all hedonics and has the highest price in the dataset; however, it is not surprising that 62 is viewed as anomalous since its price ($3.9M) is nearly 35% larger than the second most valuable property in the dataset. There is not enough information to substantiate this observation.

Table 6.3: The percentile rank for the hedonic features and closing price for each anomalous observation. Percentile ranks of 0 and 100 indicate the minimum and maximum values in the dataset, respectively.

<table>
<thead>
<tr>
<th>Observation</th>
<th>Living Area</th>
<th>Bedrooms</th>
<th>Bathrooms</th>
<th>Year Built</th>
<th>Lot Area</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>347</td>
<td>99</td>
<td>98</td>
<td>99</td>
<td>99</td>
<td>76</td>
<td>71</td>
</tr>
<tr>
<td>225</td>
<td>93</td>
<td>74</td>
<td>81</td>
<td>99</td>
<td>34</td>
<td>33</td>
</tr>
<tr>
<td>371</td>
<td>92</td>
<td>98</td>
<td>96</td>
<td>87</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>81</td>
<td>74</td>
<td>81</td>
<td>99</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>31</td>
<td>21</td>
<td>35</td>
<td>14</td>
<td>83</td>
<td>90</td>
</tr>
<tr>
<td>427</td>
<td>61</td>
<td>74</td>
<td>81</td>
<td>99</td>
<td>62</td>
<td>1</td>
</tr>
<tr>
<td>62</td>
<td>94</td>
<td>74</td>
<td>81</td>
<td>99</td>
<td>95</td>
<td>100</td>
</tr>
<tr>
<td>120</td>
<td>19</td>
<td>98</td>
<td>54</td>
<td>14</td>
<td>69</td>
<td>1</td>
</tr>
<tr>
<td>207</td>
<td>68</td>
<td>74</td>
<td>98</td>
<td>22</td>
<td>51</td>
<td>14</td>
</tr>
<tr>
<td>317</td>
<td>47</td>
<td>74</td>
<td>84</td>
<td>31</td>
<td>74</td>
<td>85</td>
</tr>
<tr>
<td>305</td>
<td>41</td>
<td>74</td>
<td>35</td>
<td>44</td>
<td>72</td>
<td>81</td>
</tr>
<tr>
<td>190</td>
<td>58</td>
<td>21</td>
<td>96</td>
<td>17</td>
<td>83</td>
<td>22</td>
</tr>
<tr>
<td>393</td>
<td>52</td>
<td>21</td>
<td>54</td>
<td>97</td>
<td>20</td>
<td>14</td>
</tr>
<tr>
<td>178</td>
<td>15</td>
<td>74</td>
<td>17</td>
<td>33</td>
<td>66</td>
<td>41</td>
</tr>
</tbody>
</table>

For a comparison of how the CNMM and DPMM handle the anomalies, we consider Table 6.4. Table 6.4 lists the anomalies, which are sorted by the empirical probability of being assigned to the CN, and the probabilities of assignment to the different types of components under both
methods. According to the CNMM, the first six observations have at least a 97% chance of being assigned to the CN, which suggests a high degree of belief that the observations are anomalous. Although the DPMM does not have a CN, the DPMM does not force these observations into the stable regular components. Under the DPMM, all six have at least a 90% chance of being assigned to one of the unstable components. The agreement to keep these six anomalies out of the stable regular components suggest that neither method will be adversely influenced by the first six anomalies. The differences arise based on how the methods allocate the remaining eight anomalies. For instance, observation 62 has nearly an 85% chance being classified as anomalous by the CNMM but a 68% chance of being assigned to one of the stable regular components in the DPMM. Continuing through the list of anomalies, we see that each is linked to a stable regular component in the DPMM. In fact, the DPMM forces three anomalies into the first component and five into the third component. This explains the behavior of the violin plots in Figure 6.1 that exhibit small differences in the first and third components but have nearly identical estimates for the second component.
Table 6.4: Empirical assignment probabilities for the anomalous observations under the CNMM and DPMM.

<table>
<thead>
<tr>
<th>Observation</th>
<th>CNMM</th>
<th>DPMM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stable</td>
<td>Unstable</td>
</tr>
<tr>
<td>347</td>
<td>0.000</td>
<td>0.008</td>
</tr>
<tr>
<td>225</td>
<td>0.007</td>
<td>0.004</td>
</tr>
<tr>
<td>371</td>
<td>0.000</td>
<td>0.012</td>
</tr>
<tr>
<td>22</td>
<td>0.000</td>
<td>0.028</td>
</tr>
<tr>
<td>32</td>
<td>0.001</td>
<td>0.027</td>
</tr>
<tr>
<td>427</td>
<td>0.000</td>
<td>0.028</td>
</tr>
<tr>
<td>62</td>
<td>0.118</td>
<td>0.034</td>
</tr>
<tr>
<td>120</td>
<td>0.143</td>
<td>0.018</td>
</tr>
<tr>
<td>207</td>
<td>0.297</td>
<td>0.010</td>
</tr>
<tr>
<td>317</td>
<td>0.378</td>
<td>0.011</td>
</tr>
<tr>
<td>305</td>
<td>0.474</td>
<td>0.015</td>
</tr>
<tr>
<td>190</td>
<td>0.486</td>
<td>0.004</td>
</tr>
<tr>
<td>393</td>
<td>0.480</td>
<td>0.012</td>
</tr>
<tr>
<td>178</td>
<td>0.564</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Forcing anomalous observations into the stable regular components affects the regression parameter estimates and, ultimately, our ability to make predictions. The CNMM identifies 14 observations as anomalous. We compare the predictive ability of the CNMM and DPMM by examining the root mean square error (RMSE) for the remaining 420 non-anomalous observations. The eight anomalies forced into the DPMM’s regular components influence the DPMM’s parameter estimates and predictions; however, we do not include the residuals for these eight observations when computing the RMSE for each method. Using the parameter estimates generated during each Gibbs iteration, we sample from the posterior predictive distribution of the response and average over all iterations to produce a prediction. In terms of the standardized data, the RMSE for the
CNMM and DPMM is 0.207 and 0.217, respectively. Although this indicates greater predictive ability when using the CNMM, the number is difficult to interpret since the responses are logged and standardized. In terms of dollars, the addition of the CN reduces the RMSE from $82,840 for the DPMM to $74,130 for the CNMM. Thus, by isolating the anomalous observation and preventing them from interfering with the formation of the remaining components, the CNMM reduces the average residual by more than $8,700. This represents a significant improvement over the DPMM.
Chapter 7

Conclusions and Future Work

We live in the data explosion era. The unprecedented amount of data offers a potential wealth of knowledge but also brings about concerns regarding ethical collection and usage. As collection continues at ever-increasing rates, ensuring the correctness of our data becomes increasingly difficult. Mistakes stemming from low-quality data have the potential for severe real-world consequences. For instance, real estate websites build automated valuation models (AVMs) using data scraped from the web. If incorrect data fed into the models produces poor predictions, buyers and sellers suffer real-life consequences.

Motivated by the application of predicting housing prices, we developed a framework for clustering and prediction in the presence of anomalous data. We presented the Cauchy-Net Mixture Model (CNMM) that augments a Dirichlet Process Mixture Model (DPMM) with a Cauchy distributed component, called the Cauchy Net (CN). The rationale is to build upon the flexibility afforded by a DPMM, while leveraging the heavy tails of the CN for capturing the anomalous observations. Isolating anomalies in a single component simultaneously identifies the observations as warranting further investigation and prevents them from interfering with the formation of the remaining components. The result is a framework that allows for simultaneously clustering observations and making predictions in the face of the anomalous data.

We demonstrated the effectiveness of combining the DPMM with the CN for clustering data con-
taminated with anomalies using the simulation studies and the Fairfax County case study. Our goal was to develop an understanding of whether, and under which conditions, the CNMM is a useful tool and to illustrate its applicability for real data. By considering a variety of well-controlled settings, we demonstrated that the CNMM is either outperforming the DPMM or performing comparably, under all of our experimental conditions. The CNMM exhibited the greatest advantages over the DPMM in scenarios with a high dimensionality and a high proportion of anomalous data. The advantages come with virtually no additional computation cost beyond that of the DPMM, aside from the evaluation of Cauchy densities. The extension of the CNMM to clustering and prediction using the Fairfax dataset provide a strong foundation for the usefulness of the CNMM as a general framework for clustering and predicting on large-scale, real-world datasets.

In order to achieve the long-term goal of effectively using the CNMM for clustering and prediction through Bayesian model averaging using large-scale datasets, we consider a number of extensions. The simulation studies explore the performance of the CNMM using several simplifying assumptions. A number of steps may be taken to improve the model’s flexibility. For example, the normal distributions use a spherical covariance structure. To allow the components to take other shapes, we plan on incorporating dense covariance matrices. The simulation studies and case study also assume known hyperparameters for the base measure. In limited testing, the performance of the CNMM was not sensitive to mild misspecifications of these hyperparameters under the experimental conditions; however, when applying the CNMM to real datasets, which are less well-behaved than our simulation data, more work is necessary to understand the impact of hyperparameter specification and/or selection. We feel that it is beneficial to explore methods, such as empirical Bayes, for choosing the hyperparameter values. Finally, the simulation studies assume a single CN with known parameters. We believe there are opportunities to relax this assumption and allow for multiple CNs, estimating the Cauchy parameters, and the possibility of replacing the Cauchy with a t and estimating the degrees of freedom. By relaxing the simulation assumptions, we can further improve our understanding of the ability of the CNMM to discover structure and make predictions in the presence of anomalies.

Although the CNMM only adds a negligible constant factor to the computational complexity of
a DPMM, performing MCMC on either method is computationally expensive. The inference approaches we presented are practical for small to moderately-sized datasets. In order to extend the CNMM to large-scale datasets, we need to consider alternative inference approaches. One possibility is to explore parallelizing the CNMM. Another option is to replace the MCMC with an optimization approach, such as the variational inference approach of (Blei et al., 2006) or the greedy search of (Wang and Dunson, 2011).

Finally, clustering and predicting on real social datasets are challenging due to the high level of noise. Our current work makes no distinction between noise and anomalies, but the source and behavior of the noise and anomalies are plausibly different. For instance, background noise may be random, but the anomalies may be described by an underlying structure. In our future work, we will consider approaches that allow for modeling the noise and anomalies separately.

The CNMM is a general framework for clustering and prediction in the face an anomalous data. Due to the reality that most datasets contain anomalies, there is widespread need for the CNMM methodology. Our work suggests that supplementing a DPMM with the CN provides an approach for discovering structure that outperforms the basic DPMM, without incurring any significant additional computational cost. Through the current and future work, the CNMM offers great promise for clustering and prediction in the presence of anomalies.
Bibliography


Appendix A

Additional Plots for Simulation Studies (Section 5.5.2)
Figure A.1: Comparison of the difference in the mean ARI for the CNMM and DPMM using a fixed sample size ($n = 100$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using $ARI(CNMM) - ARI(DPMM)$. 

<table>
<thead>
<tr>
<th>Number of Points = 100</th>
<th>$\alpha_0 = 0.5$</th>
<th>$\alpha_0 = 1$</th>
<th>$\alpha_0 = 2$</th>
<th>$\alpha_0 = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p = 2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 10$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 20$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 30$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $\pi_A$ represents the proportion of anomalous points.
- $\alpha_0$ is the true value of Dirichlet process concentration.
- $p$ indicates the dimensionality of the data.
Figure A.2: Comparison of the ARI using a fixed sample size ($n = 100$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$).
Figure A.3: Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size ($n = 100$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using $\text{ARI(CNMM Probabilistic)} - \text{ARI(CNMM Empirical)}$. 

The plots show the difference in the Adjusted Rand Index (ARI) between the CNMM and DPMM methods for different values of $\alpha_0$, $p$, and $\pi_A$. Each subplot represents a combination of these parameters, with the x-axis showing the proportion of anomalous points and the y-axis showing the difference in ARI values. The legend indicates two methods: Prob–Emp and Prob–Emp.
Figure A.4: Comparison of the difference in the mean ARI for the CNMM and DPMM using a fixed sample size ($n = 150$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The difference is computed using $\text{ARI(CNMM)} - \text{ARI(DPMM)}$. 
Figure A.5: Comparison of the ARI using a fixed sample size \( n = 150 \) and different levels for the proportion of anomalous observations \( (\pi_A) \), dimensionality \( (p) \), and true value of Dirichlet process concentration \( (\alpha_0) \).
Figure A.6: Comparison of the difference in the mean ARI for the CNMM and DPMM, using a fixed sample size \((n = 150)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\). The difference is computed using \(\text{ARI(CNMM Probabilistic)} - \text{ARI(CNMM Empirical)}\).
Appendix B

Rand Comparisons with Outliers
Figure B.1: Comparison of the difference in the mean ARI using a fixed sample size \( (n = 100) \) and different levels for the proportion of anomalous observations \( (\pi_A) \), dimensionality \( (p) \), and true value of Dirichlet process concentration \( (\alpha_0) \). The labels represent the order of subtraction. The difference is computing using ARI(CNMM) - ARI(DPMM).
Figure B.2: Comparison of the ARI using a fixed sample size \( n = 100 \) and different levels for the proportion of anomalous observations \( (\pi_A) \), dimensionality \( (p) \), and true value of Dirichlet process concentration \( (\alpha_0) \).
Figure B.3: Comparison of the difference in the mean ARI using a fixed sample size \( (n = 150) \) and different levels for the proportion of anomalous observations \( (\pi_A) \), dimensionality \( (p) \), and true value of Dirichlet process concentration \( (\alpha_0) \). The labels represent the order of subtraction. The difference is computing using ARI(CNMM) - ARI(DPMM).
Figure B.4: Comparison of the ARI using a fixed sample size \( n = 150 \) and different levels for the proportion of anomalous observations \( \pi_A \), dimensionality \( p \), and true value of Dirichlet process concentration \( \alpha_0 \).
Figure B.5: Comparison of the difference in the mean ARI using a fixed sample size ($n = 300$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The labels represent the order of subtraction. The difference is computing using $\text{ARI(CNMM)} - \text{ARI(DPMM)}$. 
Figure B.6: Comparison of the ARI using a fixed sample size ($n = 300$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$).
Figure B.7: Comparison of the difference in the mean ARI using a fixed sample size ($n = 900$) and different levels for the proportion of anomalous observations ($\pi_A$), dimensionality ($p$), and true value of Dirichlet process concentration ($\alpha_0$). The labels represent the order of subtraction. The difference is computing using ARI(CNMM) - ARI(DPMM).
Figure B.8: Comparison of the ARI using a fixed sample size \((n = 900)\) and different levels for the proportion of anomalous observations \((\pi_A)\), dimensionality \((p)\), and true value of Dirichlet process concentration \((\alpha_0)\).