8-17-2023 2:00 PM

Data Heterogeneity and Its Implications for Fairness

Ghazaleh Noroozi, *Western University*

Supervisor: Milani, Mostafa, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Recommended Citation

Noroozi, Ghazaleh, "Data Heterogeneity and Its Implications for Fairness" (2023). *Electronic Thesis and Dissertation Repository*. 9623.
https://ir.lib.uwo.ca/etd/9623

Abstract

Data heterogeneity, referring to the differences in underlying generative processes that produce the data, presents challenges in analyzing and utilizing datasets for decision-making tasks. This thesis examines the impact of data heterogeneity on biases and fairness in predictive models. The research investigates the correlation between heterogeneity and protected attributes, such as race and gender, and explores the implications of such heterogeneity on biases that may arise in downstream applications.

The contributions of this thesis are fourfold. Firstly, a comprehensive definition of data heterogeneity based on differences in underlying generative processes is provided, establishing a conceptual framework for understanding and quantifying heterogeneity. Secondly, two distribution-based clustering techniques, namely sum-product networks and mixture models, are employed to detect and identify data heterogeneity in real-world datasets. These techniques offer insights into the underlying structures and patterns of heterogeneity within the data. Furthermore, the research explores the relationship between data heterogeneity and biases, specifically investigating the impact on fairness in decision-making processes. By studying the correlation between heterogeneity and protected attributes, the thesis sheds light on how biases may arise due to the presence of heterogeneity in the data. Finally, the thesis suggests ideas and directions for addressing biases caused by data heterogeneity, paving the way for future research in debiasing techniques that consider the unique challenges posed by heterogeneous datasets.

Experimental results are presented using various datasets, including the UCI Adult Dataset, ACS Income Dataset, COMPAS Dataset, and German Credit Dataset, showcasing the practical implications of data heterogeneity on bias and fairness. The findings highlight the importance of understanding and addressing heterogeneity-related biases in predictive models, particularly when protected attributes are involved. By addressing these challenges, the thesis aims to contribute to the development of fairer and more robust decision-making systems in the face of heterogeneous data.

Keywords: Fairness, Heterogeneity, Probabilistic Model, Sum-Product Network, Latent Variable

Lay Summary

In today's data-driven world, understanding the complexities of data heterogeneity is crucial for making fair and unbiased decisions. This thesis delves into the concept of data heterogeneity, which refers to differences in how data is generated, and explores its impact on biases and fairness in machine learning models.

The research begins by defining data heterogeneity based on the underlying processes that create the data. By understanding these differences, we can gain insights into the unique challenges posed by heterogeneous datasets. To detect and identify data heterogeneity in real-world datasets, two clustering techniques, called sum-product networks and mixture models, are utilized. These techniques help us uncover hidden patterns and structures within the data that contribute to its heterogeneity.

The thesis also examines the relationship between data heterogeneity and biases, particularly focusing on how heterogeneity can lead to unfairness in decision-making. By studying the correlation between heterogeneity and protected attributes like race and gender, we uncover how biases can emerge due to variations in the data generation process. To address these biases, the thesis proposes ideas and directions for future research in debiasing techniques tailored to heterogeneous datasets.

Through extensive experiments using different datasets, the thesis demonstrates the practical implications of data heterogeneity on biases and fairness in predictive models. By identifying and addressing these challenges, we aim to develop more equitable and robust decision-making systems in the presence of diverse data.

In summary, this thesis offers a comprehensive understanding of data heterogeneity and its impact on biases and fairness. By uncovering the hidden complexities of heterogeneous datasets and proposing solutions for addressing biases, we strive to create a more inclusive and trustworthy data-driven society.

Acknowledgements

I would like to express my gratitude towards my supervisor, Dr. Mostafa Milani for his unwavering support throughout the two years of my master's program. The successful completion of this project owes a great deal to their consistent encouragement and expert guidance.

I want to acknowledge the collaborative nature of this thesis, which results from a fruitful collaboration with Yasaman Jafari and Dr. Babak Salimi from Halıcıoğlu Data Science Institute, Department of Computer Science and Engineering, University of California, San Diego. Their contributions and expertise have greatly enriched this work. Their valuable insights, discussions, and collaboration have played an integral role in shaping this thesis's research direction and outcomes. I am thankful for the opportunity to work alongside such dedicated and talented researchers.

And finally, I would like to thank my family for their unconditional support. Even though they may not have been physically close, their constant presence and encouragement have been my pillars of strength.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Data heterogeneity plays a crucial role in the realm of data analysis and information processing. Data heterogeneity refers to the inherent diversity and variations within a dataset. These variations encompass multiple dimensions, including the differences in data formats, structures, and the underlying processes by which the data is generated. Heterogeneity poses significant challenges in extracting meaningful insights and drawing accurate conclusions from the data. Understanding and addressing data heterogeneity is paramount in various domains, ranging from scientific research and business analytics to healthcare and social sciences.

As stated, the diversity in heterogeneous data can be regarding data formats and structures. Data can be represented in various formats, such as numeric, textual, categorical, or spatial, each requiring specific techniques for analysis. The data structure can vary widely, with some datasets exhibiting a tabular structure, others organized as networks or graphs, and others following hierarchical or semi-structured formats. This heterogeneity in formats and structures necessitates the development of flexible and adaptable methodologies that can handle diverse data representations, enabling practical analysis and interpretation across different domains.

The diversity in data can be with regard to data generation processes. This thesis focuses on this aspect rather than structural or formatting heterogeneity. These disparities in data generation can occur in two main areas. Firstly, heterogeneity can be due to different data sampling and collection techniques. This involves various sources, from surveys and experiments to observations and simulations, each with unique characteristics, assumptions, and measurement methods. Secondly, it can also arise due to the inherent differences within the application domain's subpopulations, each exhibiting unique attributes and characteristics.

The data generation process goes beyond the mere mechanics of data collection and involves deep-rooted differences within the data source itself. Comprehending these multi-dimensional disparities is critical, as they can introduce biases, latent factors, or variations in the data. These factors can ultimately influence the reliability and generalizability of any findings or insights derived from the dataset. By acknowledging and exploring the multifaceted nature of data heterogeneity, researchers and practitioners can develop robust data preprocessing, integration, and analysis strategies, paving the way for more accurate and comprehensive knowledge extraction from complex and diverse datasets.

1.1 Motivation and Research Questions

Data heterogeneity, characterized by the variations in the underlying generative processes of the data, produces biases with consequential implications in decision-making models. Consider a dataset comprising patient records from two hospitals with unique characteristics, such as differing specialties and demographic profiles. Each hospital provides critical information, but their distinct generative processes can create inherent biases. For example, one hospital might predominantly serve elderly patients with chronic diseases, while the other primarily treat a younger demographic with acute conditions. The prediction task for mortality may differ substantially between the two for various reasons, leading to heterogeneity that can profoundly influence the accuracy and fairness of models trained on such data, particularly when these variations correlate with protected characteristics like race or gender.

If the data in one of the hospitals is very imbalanced with regard to the class label, a binary classification model would have a substantially harder time classifying individuals accurately. Also, a particular hospital can have fewer samples in the dataset, making it difficult for the model to predict outcomes accurately for its patient population. The complexity of the relationship between input attributes and the work (e.g., mortality) might differ across the two hospitals. For instance, one hospital's mortality rate might be influenced by a broader range of factors, making the prediction task more complex. In this scenario, the model tends to learn the prediction task mainly on the hospital with more data points, as it would contribute to the aggregate accuracy more.

Similar to the examples, biases can emerge if the two hospitals have distinct generative processes that result in data heterogeneity. Suppose a model is trained on a combined dataset from both hospitals. In that case, it might unintentionally favor the hospital with a more straightforward prediction task or a larger sample size due to these inherent biases. This could lead to an unintentional skew, resulting in better prediction accuracy for patients from one hospital. In contrast, predictions for patients from the hospital with a more complex or less represented generative process could suffer. If these differences in generative processes are associated with protected attributes, such as race or gender, these biases could be further intensified. For example, if the hospital with fewer samples predominantly serves a specific racial group, this could lead to biases in the accuracy and quality of predictions made by a model trained on this data.

Addressing the biases introduced by data heterogeneity and understanding their association with protected attributes can empower researchers and practitioners to proactively mitigate them, improve model fairness, and secure equitable outcomes. This thesis explores a critical research question: Can data heterogeneity, defined by variations in the dataset's underlying generative processes, induce biases? More specifically, by characterizing a latent variable representing data heterogeneity, we break our primary question into two specific inquiries: 1) Is there a correlation between the latent variable and protected attributes central to unfairness and biases in data? 2) Can disparities in the prediction task for subpopulations, as defined by the latent variable, contribute to biases in downstream prediction models? We seek answers to these questions in real datasets prevalent in fairness research, drawing conclusions based on quantitative findings.

1.2 Existing Research and Research Gap

The emerging fields of machine learning and data science have shed new light on the complexity of data. A crucial aspect of this complexity is data heterogeneity. Recognizing and understanding such heterogeneity has profound implications, influencing everything from the fairness of machine learning models to the robustness and generalizability of these models across diverse data contexts. This section reviews existing research related to data heterogeneity and explores the gap in current knowledge, setting the stage for the investigation presented in this thesis.

1.2.1 Uncovering Data Heterogeneity via Latent Variable Discovery

In the landscape of machine learning, the impact of data heterogeneity — defined as differences in the underlying generative processes for a dataset — on the fairness of models has emerged as a significant area of concern. A heterogeneous dataset can be partitioned into homogeneous clusters, meaning the underlying distribution in each cluster is simple and can be learned with standard approaches. Such heterogeneity can be characterized by latent or hidden attributes that label those clusters in the datasets.

Various techniques can be exploited to extract the latent variables in data. Probabilistic graphical models such as Latent Dirichlet Allocation (LDA) [3, 25] or Hidden Markov Models (HMM) [46] have been widely used to discover hidden structures in textual and temporal data, respectively. Other unsupervised methods, like clustering and dimensionality reduction techniques, provide alternative ways of unveiling concealed variables. Advancements in deep learning have given rise to powerful models like Variational Autoencoders (VAE) [31] and Generative Adversarial Networks (GAN) [20], which are adept at extracting latent features in complex, high-dimensional data. These techniques have diverse applications, from natural language processing and computer vision to recommendation systems and healthcare. Unearthing these hidden variables allows us to navigate the complex landscape of data heterogeneity and significantly influence the fairness, robustness, and generalizability of models trained on such data.

This thesis employs two conventional techniques commonly used to extract hidden or latent variables: Mixture Models (MMs) and Sum-Product Networks (SPNs). The former is chosen for its simplicity and pervasive use, while the latter is selected for its efficiency in probabilistic inference. Exploring additional techniques for identifying latent variables and their subsequent influence on fairness presents an intriguing direction for future research.

1.2.2 Data Heterogeneity in Prediction Models

Previous sections have well explained the implications heterogeneity can have on prediction tasks. One of the earliest studies incorporating data heterogeneity in constructing predictive models is detailed in Karpatne et al. [29]. In this research, the authors partition a given dataset based on the relationship between input variables and a target variable, creating segments that reflect the inherent heterogeneity within the data. These partitions define latent variables. They assign a prediction model to each partition and exploit the structure between partitions by creating a graph with nodes as partitions and edges as the similarity between them. This is done by regularizing the objective function to ensure similar partitions have similar values for their corresponding model parameters.

Similarly, Karpatne et al. propose an algorithm for constructing an ensemble prediction model considering data heterogeneity [30]. They assume the classes encompass multiple modes, and some of these modes are highly overlapping with each other, making the prediction task challenging to distinguish between them. They learn a classification model per each pair of these modes, and their Adaptive Heterogeneous Ensemble Learning (AHEL) algorithm generates an ensemble of them. This ensemble is designed to adapt its weighting scheme based on the local context of test scenarios, thus further enhancing model performance in diverse data contexts.

A more recent area of research within data heterogeneity is the concept of hidden stratification [44, 40, 54]. Hidden stratification represents the existence of essential but unidentified subgroups within a dataset, a characteristic that underscores data heterogeneity. These subgroups can exhibit different characteristics or distributions and often go unrecognized or unaddressed during modeling. This can inadvertently affect the model's predictions in ways that aren't immediately obvious or fully understood.

Multigroup learning is a subfield of machine learning that explicitly acknowledges and addresses distinct groups or subgroups within a dataset [48, 56]. These groups could be defined based on various factors such as demographic attributes, different experimental conditions, different geographical regions, and more. In multigroup learning, separate models or model components are often trained for each group, allowing the model to learn and adapt to the unique characteristics of each group. Hidden stratification and data heterogeneity are closely related to and categorized under a more general term of multi-group learning, where the goal is to learn multiple models to address heterogeneity.

Multicalibration is a recent data heterogeneity and fairness research and is highly related to our work [21, 24]. Calibration is a property of machine learning models that requires the model's predictions to align closely with the true label. Multicalibration extends this notion to multi-group settings where the predictions should be calibrated across the subgroups. This concept is inherently linked to fairness, as a model that lacks multicalibration could produce systematically biased predictions for certain subgroups, potentially leading to unfair results. The challenge of achieving multicalibration arises from data heterogeneity, which means having diverse subgroups within a dataset, each exhibiting unique characteristics. A prediction model must account for them to ensure well-calibrated predictions across all subsets.

1.3 Thesis Contributions and Structure

This work distinguishes itself from the studies on hidden stratification and multigroup learning in two main ways: firstly, our work focuses on the influence of heterogeneity on fairness. The aforementioned works primarily concentrate on prediction tasks and accuracy. While our work shares connections with multicalibration, it diverges by prioritizing the identification of data heterogeneity and its implications for fairness rather than developing fair and well-calibrated models, a characteristic feature of multicalibration. In a multicalibration context, subgroups are provided a priori. However, we extract these subgroups directly from the data in our approach. Secondly, most of the works mentioned above assume that homogeneous subgroups or clusters in the data are available to them in the input or a secondary source. They assume heterogeneity exists in the datasets while we conduct and perform experiments to detect and measure whether a dataset is heterogeneous. The novelty of this work is explained thoroughly in the next chapter in Table 2.1.

Finally, the full thesis contributions are:

1. This thesis introduces a comprehensive definition of data heterogeneity based on the differences in the underlying generative processes. A nuanced understanding of data heterogeneity is established by explicitly considering the generative aspects of data. This definition provides a solid foundation for further exploration and analysis of heterogeneity in real-world datasets.

2. To effectively identify and detect data heterogeneity in real data, this thesis utilizes two distribution-based clustering techniques: SPNs and MMs. By leveraging the capabilities of these clustering techniques, the thesis develops a robust framework for detecting and identifying instances of data heterogeneity. This approach enables researchers and practitioners to gain deeper insights into the presence and characteristics of heterogeneity within datasets.

3. This thesis investigates the correlation between identified heterogeneity and protected attributes such as race and gender. The impact of data heterogeneity on biases and fairness is evaluated through rigorous analysis. By uncovering potential relationships between heterogeneity and protected attributes, this research sheds light on the implications of heterogeneity in decision-making processes and the potential for biased outcomes.

4. This experimental result emphasizes the complications of heterogeneity on the accuracy of prediction models and how much they can be improved by learning different models for homogeneous sections of the data.

5. This thesis puts forward novel ideas and recommendations to address the biases stemming from data heterogeneity. Drawing upon the findings from the previous contributions, potential strategies and interventions are suggested to mitigate biases and promote fairness in predictive models. Additionally, this research presents future research directions for debiasing approaches that take into account the unique challenges posed by data heterogeneity.

Chapter 2

Related Work

This chapter explores important research areas related to this work: data heterogeneity, fairness, and bias in predictive models. We review related studies and techniques for handling diverse datasets, including multi-group learning, hidden stratification, multicalibration, and multimodality. Additionally, we delve into the realm of fairness and bias, examining efforts to identify and address discriminatory outcomes in predictive models. By delving into these topics, this chapter offers valuable insights into the advancements made in data analysis, fostering a more robust and morally sound approach to predictive modeling.

2.1 Studies on Data Heterogeneity

In this section, we explore existing work related to data heterogeneity, encompassing various approaches to handling complex scenarios where the underlying data exhibits diverse characteristics. This includes multi-group learning, hidden stratification, and multimodality. These concepts help address the challenges posed by data heterogeneity by accounting for differences across subgroups, identifying hidden patterns and biases, and calibrating models to mitigate distributional shifts.

The relationship between heterogeneity and prediction models has been explored by work [29]. They explore a prediction task on heterogeneous data with insufficient data samples in some subgroups. They exploit a structural variable Z from a secondary source, cluster the data based on it, and learn one model per cluster. They then proceed to build a graph with nodes as partitions and edges as the similarity between partitions. Finally, they ensure that similar partitions have similar model parameters by penalizing their difference in the objective function.

Multi-Group Learning: Multi-group learning is another technique to guarantee prediction model accuracy across subpopulations [48, 56]. It considers an arbitrary collection of potentially overlapping subpopulations and learns a single predictor from a class of possible predictors. In multi-group learning, the loss experienced by every subpopulation cannot be much larger than the loss of the best predictor for that subpopulation. Rothblum et al. extend the idea of agnostic PAC learning to take advantage of multi-group learning, ensuring the performance of each underlying sub-group is close to the performance of the optimal predictor [48]. Agnostic PAC learning focuses on finding prediction models with low generalization error on unseen data, regardless of the data's underlying distribution or the presence of noise in the training data.

Hidden Stratification: Learning predictive models over subpopulations has been recently studied more extensively in the healthcare domain, where prediction models often perform poorly for some unidentified subpopulations. This challenges the reliability and fairness of the machine learning models in this field as the underperformed subpopulations are usually clinically significant. This problem is addressed as hidden stratification [44, 40, 54]. Hidden stratification is defined as the presence of unobserved factors or characteristics that systematically influence data, creating patterns or variations not captured by observed variables [44], which can introduce bias and impact the validity of statistical analyses. Hidden stratification is encountered in various fields, such as genetics, epidemiology, and social sciences. It requires careful consideration and appropriate techniques like stratification, matching, or regression modeling to account for unobserved factors. By addressing hidden stratification, researchers can improve the accuracy and reliability of their data analysis. In healthcare, hidden stratification refers to unknown patient subpopulations for which prediction models often perform poorly. An example is a cancer prediction model that fails to identify cancer patients of a particular type.

Several techniques are defined and introduced to measure hidden stratification. Oakden-rayner et al. defines and compares three practical methods in medical computer vision: schema completion, error auditing, and algorithmic measurement [40]. Sohoni et al. proposes an algorithm that learns an ERM model to estimate the subclass labels. Then, it leverages them as proxy cluster labels to learn a prediction model that minimizes the highest loss amongst the subpopulations [54].

Multimodality: Multimodality is a type of population heterogeneity characterized by multiple modes for each class in the feature spaces. [30] incorporates ensemble learning to address heterogeneity. In a binary classification setting, they assume that the data has unknown modes in the positive and negative classes, and they are highly overlapping, making it difficult for the model to differentiate between them. To address this issue, they learn a binary classifier to distinguish between each pair of positive and negative class modes and combine these classifiers using ensemble learning techniques.

Multicalibration: Multicalibration is a new measure of algorithmic fairness that extends the calibration notion into a multi-group setting. Multicalibration guarantees an accurate and well-calibrated prediction model across arbitrary and possibly overlapping subpopulations [21]. The subpopulations are defined by a given set of membership functions that make it possible to provide strong fairness guarantees; multi-calibration with respect to a general set of functions protects all the subpopulations identified by the functions [24].

Multi-group learning, hidden stratification, and multimodality are related to this work as they train highly accurate prediction models across all subpopulations. However, they do not assume predefined demographic groups defined by sensitive attributes.

Table 2.1: Position of this research in the literature

Related Work	Heterogeneity Definition	Detecting Subpopulations	Considering Fairness	Measuring Heterogeneity
This Research	Difference in underlying data distribution	✓	✓	✓
[29]	Difference in underlying data distribution	✗	✗	✓
[30]	Multimodality	✗	✗	✓
[54]	Hidden stratification	✓	✓	✗
[44]	Hidden stratification	✓	✗	✗
[40]	Hidden stratification	✗	✗	✗
[48, 56]	Multi-group setting	✓(individual fairness)	✗	✗
[21, 24]	Multi-group setting	✓	✗	✗

2.2 Fairness in Machine Learning

Fairness in machine learning has received significant attention in recent years, and numerous studies have explored different aspects of this topic. A critical line of research has focused on developing fairness metrics and measures to assess the degree of bias in machine learning models. For example, Kamiran et al. introduced a set of metrics to measure disparate impact and disparate mistreatment [28], which have become widely used in the fairness literature [15]. Other popular fairness metrics include statistical parity, equal opportunity, and demographic parity.

Various techniques have been proposed to mitigate bias and improve fairness in machine learning, including preprocessing, in-processing, and post-processing methods. Preprocessing methods aim to address discrimination in the training data before training the model, and they include techniques such as data augmentation, and data resampling. In-processing methods modify the learning algorithm or objective function to incorporate fairness constraints, and examples include adversarial training and regularization. Post-processing methods adjust the model output to enforce fairness constraints, such as thresholding or reweighting.

Several studies have also explored the interpretability and explainability of machine learning models in the context of fairness. For example, Doshi et al. proposed a framework for "actionable" and "interpretable" models that can provide insight into how decisions are made and can be used to identify and address bias [14].

2.2.1 Bias in Machine Learning

Bias in machine learning can significantly impact the fairness and performance of predictive models. These biases typically fall into three main categories: inherent data biases, algorithm-induced biases, and user-contributed biases.

- *Inherent data biases* These biases, arising from data collection and preprocessing stages, can distort model predictions. For example, measurement bias [49, 55] refers to inaccuracies in data collection or measurement, leading to systematic errors that misrepresent underlying trends. Omitted variable bias [9] occurs when an essential variable is excluded, skewing the predictions. Representation bias [55] and sampling bias [33] occur when the collected data is unrepresentative. Representation bias happens when data does not reflect the population accurately, while sampling bias arises from non-random data selection for model training. Both biases can cause models to perform well on sample data but poorly on the overall population. Other forms of inherent data biases, aggregation bias [55] (or Simpson's paradox [4]) and linking bias [41], are related to how data is grouped and linked. Aggregation bias can obscure or reverse individual data trends, while linking bias occurs when linking datasets creates a biased information view.

- *Algorithm-induced Biases* These biases stem from the design and operation of predictive models and their learning algorithms. Algorithmic bias results from the assumptions and oversimplifications an algorithm makes during the learning process, and it may arise when an algorithm fails to capture complex data patterns. Evaluation bias [55], on the other hand, occurs due to biased evaluation metrics, leading to inaccurate assessments of model capabilities. Presentation, ranking, and popularity bias are common in information retrieval systems. Presentation and ranking bias [37] arises from the influence of item order on user choices, while popularity bias [52] refers to a feedback loop where frequently recommended items gain further popularity. A unique type of algorithm-induced bias is emergent bias, which surfaces when a model trained on unbiased data starts adopting biases from user interactions or feedback.

- *User-contributed Biases* These biases stem from the behaviors and characteristics of the users who interact with predictive models or contribute data. Historical and social biases [55] reflect societal prejudices and stereotypes, both past and present. They result from discriminatory or prejudiced patterns in data, perpetuating these biases in machine learning models. Population bias and self-selection bias originate from skewed user representation. Population bias [57] arises from unequal user group representation, leading to a model bias towards majority groups. Conversely, self-selection bias [37] stems from voluntary data contributions, leading to potential data skewness. Behavioral, temporal, and content production biases [37] are additional forms of user-contributed biases. Behavioral bias arises from user interactions with the system, while temporal bias originates from outdated data that may not accurately represent current or future states. Content production bias, on the other hand, emerges when the content producers influence the data available for model training. Addressing these biases is vital for developing fair and robust machine learning models, with the first step being awareness of their existence and potential impacts.

Bias and Data Heterogeneity Data heterogeneity, the presence of subpopulations in data where each subpopulation's data is generated from different distributions, is intricately related to several types of biases inherent in datasets.

Measurement bias, for instance, is a significant issue when dealing with data heterogeneity [49, 55]. This form of bias arises when there are inconsistencies or errors in data collection or measurement across different subpopulations. These inconsistencies can originate from various factors, such as different data collection methods or inconsistencies in how the data is recorded or processed among different groups. As a result, the distribution of values in each subpopulation may be misrepresented, leading to systematic errors in the dataset that can skew predictions or analysis. When these measurement errors differ across subpopulations, they can introduce disparities and lead to biased model predictions or analyses.

Sampling bias is another type of bias that is particularly relevant when dealing with heterogeneous data [33]. This bias occurs when specific subpopulations are over- or under-represented in the dataset due to how data is collected. Specific subpopulations might be under-sampled in data heterogeneity, while others might be over-sampled. This non-uniformity in sampling can skew the overall dataset, which can, in turn, lead to biased results since the model may overfit to overrepresented groups and underfit to underrepresented ones.

Omitted variable bias can also arise in the context of data heterogeneity [9]. This bias occurs when a significant variable that influences the model's output is not included in the model. In the case of heterogeneous data, this omitted variable could be the latent variable that separates the subpopulations. If this latent variable is not accounted for in the model, it may cause the model to overlook the inherent heterogeneity in the data and lead to biased predictions. In other words, failing to consider the subpopulation structure in the data can lead to a misleading understanding of the relationships within the dataset.

Understanding and addressing these biases associated with data heterogeneity is crucial for achieving accurate and fair predictions or analyses. It is essential to consider these biases during the data collection and preprocessing stages and the model design and evaluation phases.

Chapter 3

Preliminaries

This chapter provides an overview of the preliminary concepts relevant to the thesis. It reviews distribution-based clustering techniques, focusing on prominent methods such as MMs and SPNs that are used in this work. Additionally, it delves into the related concepts of fairness and different types of biases in predictive models.

3.1 Distribution-Based Clustering

Clustering can be based on various metrics. Distance-based clustering methods, like K-means [23], partition data points into clusters based on their proximity or similarity in the feature space without explicitly modeling the data's underlying distribution. These methods are sensitive to the choice of distance metric and can create clusters of varying shapes and sizes based on distance thresholds. Unlike distance-based clustering algorithms, which rely on proximity measures, distribution-based clustering focuses on modeling the data distribution and identifying regions of high density as clusters. This approach offers a flexible framework for handling datasets with complex structures and is particularly useful when dealing with data that does not exhibit well-defined spherical or convex clusters. In this thesis, I use distribution-based clustering techniques since the goal is to capture heterogeneity in the data, characterized by the difference in the underlying data generation and distribution.

Distribution-based clustering is a widely used approach that partitions data into clusters based on the underlying probability distribution of the data points [58]. There are several prominent techniques for distribution-based clustering, including MMs and its variations [34, 35, 47, 36], non-parametric MMs such as Dirichlet Process Mixture (DPM) [2], Kernel Density Estimation (KDE) [42], Mean Shift [7], and Self-Organizing Maps (SOM) [32].

Many clustering techniques, including MMs, apply Expectation-Maximization (EM) [38]. This is a general-purpose algorithm that iteratively estimates the parameters of the data distribution. It maximizes the likelihood of the data by alternating between the expectation step, where cluster membership probabilities are computed, and the maximization step, where the parameters are updated. Clustering based on EM can handle various distributions and is not restricted to Gaussian assumptions.

Some of the other clustering techniques mentioned above do not rely on EM. KDE estimates the underlying probability density function of the data by placing a kernel function on each data point and summing them to form a density estimate. Clusters are identified as regions of high density. KDE effectively captures clusters with different shapes and densities and does not require specifying the number of clusters in advance [42].

Mean Shift starts by defining a kernel function and iteratively shifts data points towards the mode of the kernel density estimate. The algorithm seeks to find the modes or peaks in the density function corresponding to the cluster centers. Mean Shift is particularly effective for clustering datasets with unevenly distributed clusters or clusters with different sizes. It does not require specifying the number of clusters and can handle complex and non-linear data distributions [7].

SOMs use a neural network-based approach to clustering. SOMs represent clusters as prototypes or codebook vectors in a low-dimensional grid. The algorithm iteratively adjusts the prototypes to capture the underlying data distribution, effectively grouping similar data points. SOMs are particularly useful for visualizing high-dimensional data and discovering topological structures [32].

Non-parametric MMs are flexible clustering techniques that do not receive the number of clustering as a prior and do not use EM. They employ techniques such as Markov chain Monte Carlo (MCMC) methods [19], such as Gibbs sampling [16] or Metropolis-Hastings [8], which allow for sampling from the posterior distribution, or variational inference, which approximates the posterior distribution using predefined families of distributions. These alternative approaches provide efficient inference and estimation tailored to the characteristics of non-parametric models, allowing for flexible estimation of the number of components and a more data-driven approach to modeling and clustering.

In this thesis, we employ finite MMs and SPNs with the Expectation-Maximization (EM) algorithm. This choice is driven by their advantages over non-EM-based techniques, providing efficient parameter estimation and enhanced interpretability in clustering results. Using EM-based approaches, we aim to achieve robust and principled clustering outcomes in our study.

3.1.1 Clustering Metrics

Measures for evaluating the quality of clustering play a crucial role in assessing the effectiveness and performance of clustering algorithms. These measures provide quantitative assessments of how well the data points are grouped into clusters and aid in comparing different clustering solutions. There are two types of measures. Extrinsic measures evaluate clustering results by comparing them to external criteria, such as known class labels, to assess how well the clusters align with the ground truth or external information. Intrinsic measures, which are relevant to this thesis, assess clustering quality based on internal characteristics of the data, such as the compactness and separation of the clusters. Examples are the Davies-Bouldin index [10], the Calinski-Harabasz Index [6], and the Silhouette measure [50].

This thesis uses log-likelihood as one widely used measure for distribution-based clustering, particularly relevant to EM-based techniques based on maximum likelihood estimation [39]. Maximum likelihood estimation aims to find the parameters that maximize the likelihood of observing the given data. In the context of distribution-based clustering, the EM algorithm is often employed to estimate the parameters of the underlying probability distribution. The EM algorithm iteratively maximizes the likelihood function, adjusting the model parameters until convergence. Maximizing the likelihood ensures that the model captures the data distribution accurately, resulting in high-quality clustering.

In maximum likelihood, the log-likelihood is used to measure clustering quality. Given a set of observed data points $D = \{x_1, x_2, ..., x_n\}$ and a fitted distribution-based clustering model with parameters θ, the log-likelihood function $L(\theta \mid D)$ evaluates the likelihood of observing the data points in D based on the model with parameters θ. The log-likelihood is defined as the logarithm of the likelihood function: $L(\theta \mid D) = log(P(D \mid \theta))$. The higher the log-likelihood value, the better the clustering quality, as it indicates a higher likelihood of the observed data under the given model. Maximizing the log-likelihood corresponds to finding the parameter values that best fit the data.

Maximizing the log-likelihood is often achieved using iterative algorithms like the EM algorithm. The EM algorithm maximizes the expected log-likelihood by iteratively updating the distribution-based model's parameters, improving the data's fit [38]. Convergence of the algorithm occurs when the log-likelihood reaches a plateau or when a stopping criterion is met. Comparing the log-likelihood values across different clustering solutions or models can help assess their relative quality and identify the most suitable model for the given dataset. However, it's important to note that the absolute value of the log-likelihood is not interpretable by itself, and comparisons should be made within the same model or across models with the same underlying assumptions.

We evaluate the quality of distribution-based clustering primarily based on log-likelihood. This is motivated by using EM-based techniques for parameter estimation in distribution-based models. Maximum likelihood estimation, inherent in the EM algorithm, aims to find the parameter values that maximize the likelihood of observing the given data. By prioritizing maximum likelihood, we aim to assess the clustering solutions based on their ability to accurately capture the data distribution, aligning with the fundamental principle of EM-based techniques. While other measures such as BIC, AIC, and external validation indices provide valuable insights, emphasizing maximum likelihood enables a focused evaluation aligned with the core methodology employed in this thesis.

3.2 Mixture Models

Mixture models (MMs) [36] are statistical models widely used in machine learning and statistics. They are beneficial for modeling complex, multi-modal data distributions. A MM represents a complex distribution as a mixture of simpler distributions, e.g., Gaussian distributions [47], where each simpler distribution is called a *component*. Each component is defined by a set of parameters that determine its shape, location, and scale. A MM is a weighted sum of K component distributions, where each component distribution represents a distinct cluster. Let $D = \{x_1, x_2, \ldots, x_n\}$ denote the observed data set with n data points. Each data point x_i is assumed to be generated from one of the K components. The probability density function (pdf) of the finite MM is given by the following

$$f(x_i \mid \theta) = \sum_{k=1}^{K} \pi_k \cdot p(x_i \mid \phi_k), \tag{3.1}$$

where π_k represents the *mixing weight* for the k-th component, satisfying $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$, and $p(x_i \mid \phi_k)$ represents the pdf of the k-th component with parameters ϕ_k.

The goal of finite mixture modeling is to estimate the set of component parameter pairs $\theta = \{(\pi_i, \phi_i)\}_{i=1}^{K}$ that maximize the likelihood of the observed data set D. This is typically achieved using the Expectation-Maximization (EM) algorithm, where the E-step computes the posterior probabilities of data points belonging to each cluster, and the M-step updates the parameters by maximizing the expected log-likelihood.

The estimated parameters $\hat{\theta}$ can then be used to assign new data points to clusters and perform various probabilistic inferences within the finite MM framework.

In a Gaussian Mixture Model, each data point x_i is assumed to be generated from one of K Gaussian components [47]. The GMM probability density function (pdf) is given by $f(x_i \mid \theta) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$, where μ_k represents the mean vector, and Σ_k represents the covariance matrix of the k-th Gaussian component. GMMs provide a flexible framework for capturing complex data distributions and can be used for various tasks such as clustering, density estimation, and sample generation. For MMs, the log-likelihood is computed as the sum of the logarithms of the individual probability density functions (pdfs) for each data point: $L(\theta \mid D) = \sum_i log(f(x_i \mid \theta))$. In this work, we use MM to refer to Gaussian MM in the rest of the thesis.

3.2.1 Learning and Expectation Maximization

Learning MMs involves estimating the parameters that maximize the likelihood of the observed data. Maximum likelihood estimation (MLE) is commonly used for this purpose. Given a set of observed data points $D = \{x_1, x_2, \ldots, x_n\}$, we aim to find the parameters θ that maximize the likelihood function. The likelihood function of the MM is defined as $L(\theta \mid D) = \prod_{i=1}^{n} f(x_i \mid \theta)$, where $f(x_i \mid \theta)$ represents the probability density function of the MM. To simplify the optimization, it is common to work with the log-likelihood function $\ell(\theta \mid D) = \log L(\theta \mid D)$, the sum of the logarithms of the individual probabilities.

The EM algorithm is commonly employed to estimate the parameters in MMs [13]. The algorithm involves two main steps: the E-step (Expectation step) and the M-step (Maximization step). In the E-step, the algorithm computes the expected value of the log-likelihood function with respect to the posterior probabilities, which represent the responsibility of each component for generating each data point. The posterior probability $p(z_i = k \mid x_i, \theta^{(t)})$ indicates the likelihood of data point x_i belonging to component k at iteration t. In the M-step, the algorithm maximizes the expected log-likelihood function with respect to the parameters, updating the parameter estimates.

The EM algorithm iteratively alternates between the E-step and M-step until convergence. At each iteration t, the log-likelihood function $\ell(\theta \mid D)$ increases, improving the parameter estimates $\theta^{(t)}$. Once the algorithm converges, the estimated parameters $\hat{\theta}$ represent the maximum likelihood estimates for the MM.

The EM algorithm provides a principled approach for learning MMs by iteratively maximizing the likelihood of the observed data. It is widely used in practice and is particularly well-suited for models with latent variables, such as MMs.

3.3 Sum-Product Networks

Sum-Product Networks (SPNs) are similar to probabilistic graphical models (PGMs) and compactly represent probability distributions [45]. The main difference between SPNs and PGMs, such as Bayesian networks (BNs) and Markov networks (also called Markov random fields) is that in a PGM, every node represents a variable; roughly speaking, links represent probabilistic dependencies between random variables, sometimes due to causal influences. In contrast, in an SPN, every node represents a probability function.

SPNs are powerful probabilistic models that provide a tractable representation of joint probability distributions. Given a joint probability distribution P over a set of random variables $X = \{X_1, X_2, \ldots, X_n\}$, an SPN is a directed acyclic graph (DAG) that captures the dependencies and interactions between these variables. An SPN consists of two types of nodes.

Sum Nodes: Sum nodes represent a weighted sum of their children. Let S be a sum node with children $C_1, C_2, \ldots, C_m$. The value of the sum node S is computed as follows:

$$S = \sum_{i=1}^{m} w_i \cdot C_i, \tag{3.2}$$

where w_i represents the weight associated with the child node C_i, sum nodes capture the additive decomposition of the joint probability distribution by considering weighted contributions from their children.

Product Nodes: Product nodes in an SPN represent their children's products. Let P be a product node with children $C_1, C_2, \ldots, C_n$. The value of the product node R is calculated as $R = C_1 \cdot C_2 \cdot \ldots \cdot C_n$, where C_i represents the child node of the product node. Product nodes capture the multiplicative interactions and dependencies between random variables.

The structure and connections of sum and product nodes in an SPN collectively represent the joint probability distribution P. The leaf nodes in the network correspond to univariate distributions over individual random variables or indicator functions [53]. An SPN provides a hierarchical and decomposable representation of the joint probability distribution, allowing for efficient inference and learning. The weighted sums computed at sum nodes decompose the probability distribution, while the product nodes capture the dependencies and interactions between the random variables through multiplication.

By traversing the SPN graph, starting from the root node and propagating values through sum and product nodes, the network ultimately computes the probability of any configuration of the random variables in X. This ability to efficiently represent and calculate the joint probability distribution makes SPNs valuable for various tasks, including probabilistic inference and learning.

Example Figure 3.1 is an SPN that represents the joint probability of three Boolean variables $P(X1, X2, X3)$in the Bayesian network $X_2 \leftarrow X_1 \rightarrow X_3$. Each leaf node is the probability function, e.g., the nodes with labels x_1 and $\bar{x}_1$ respectively represent the probability functions $P(X_1 = 1)$ and $P(X_1 = 0)$. The edge labels represent the weights in the sum nodes. Starting from the leaf nodes and using the weights, one can compute the probability of any variable assignments, e.g., $P(X_1 = 1, X_2 = 1) = 0.8 \times 0.3$ and $P(X_1 = 1, X_2 = 0) = 0.8 \times 0.7$. This example is from [17], and we only use it as a running example to explain SPNs in the rest of Section 3.3. ■

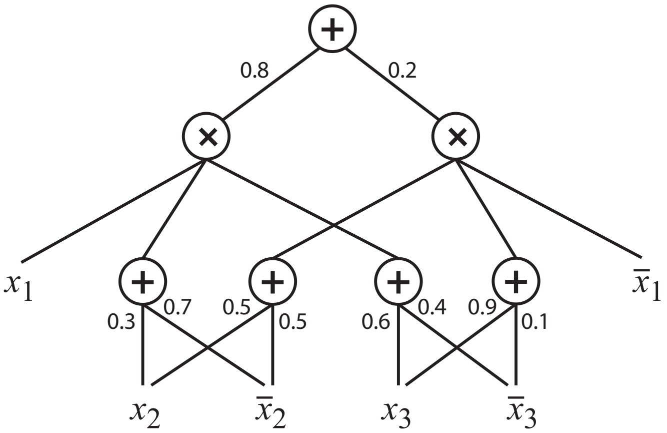


Figure 3.1: The SPN in Example 3.3

3.3.1 Properties of SPNs

SPNs exhibit several important properties, including completeness, decomposability, validity, and selectivity. Understanding these properties is essential for grasping the theoretical foundations and practical applications of SPNs in probabilistic modeling and inference tasks. For more detail see [45, 53].

- *Completeness:* A sum node is complete if its children have the same scope. The scope of a node is the set of random variables that appear in the probability functions in the leaf nodes of the subtree rooted at the node. An SPN is complete if all its sum nodes are complete. The SPN in Exapmle 3.3 is complete. As an example node, the root is complete as the scope of both child product nodes is $\{X_1, X_2, X_3\}$. Similarly, the first sum node from left in the third level is complete as the scope of both children is $\{X_2\}$.

- *Decomposability (consistency):* A product node is decomposable if its children have disjoint scopes, meaning they have no variables in common. SPN is decomposable if all its product nodes are decomposable. The SPN in Example 3.3 is also decomposable. Both product nodes are decomposable because the domains of their children are $\{X_1\}, \{X_2\}, \{X_3\}$ that are disjoint.

- *Validity:* An SPN is valid if it is complete and consistent. Validity is proven to ensure an SPN respects the laws of probability and provides a valid probabilistic model. Therefore, the SPNs in this work are expected to be valid, i.e., consistent and complete.

- *Selectivity:* A sum node in an SPN is selective if it has at most one child with a non-zero value for any sum weights. Intuitively, the probability of a variable assignment (a.k.a. a configuration) is computed starting from all leaf nodes and aggregating the sum and product values to the root. When a node is selective, the probability can be only non-zero in one child. The SPN in Example 3.3 is selective. Every sum node is selective as only one child product node has a positive probability for a configuration, e.g., for $X_1 = 0, X_2 = 0, X_3 = 0$, only the leftmost children have positive probabilities, and the probability of the others is zero.

Selectivity is integral to this work, as it empowers the network to cluster data effectively. While occasionally referred to as *determinism*, the term selectivity is deemed more appropriate since it avoids the implication that the SPN models deterministic relationships among observable variables. It's possible to augment non-selective sum nodes to achieve selectivity in an SPN, all while preserving its validity and faithful representation of the original probability distribution [53, 43]. This augmentation procedure entails adding new hidden nodes, which represent deterministic variables, to each child of the sum node. In our work, these nodes that guarantee selectivity signify the latent variable responsible for data heterogeneity. How we leverage an SPN for clustering is elaborated in Section 4.2.2.

3.3.2 Learning in SPNs

Similar to GPMs, learning in SPNs encompasses two main components: structure learning and parameter learning. Structure learning pertains to determining the network topology, which includes the configuration of sum and product nodes in the network and the connections or edges among them. On the other hand, parameter learning involves identifying the appropriate weights. These aspects are briefly reviewed here, with more comprehensive details available in [18, 17].

Structure Learning: Structure learning in SPNs refers to determining the optimal or near-optimal graph representing the SPN. We review two main algorithms for structure learning:

- *BuildSPN* is a pioneering algorithm in this realm and discovers subsets of highly correlated variables and introduces latent variables to account for those dependencies [11]. These latent variables create sum nodes, which are repeated recursively to find additional latent variables. BuildSPN progressively merges smaller SPNs into larger ones by applying a statistical dependence test to surmount this limitation. BuildSPN was critiqued due to several reasons: (1) the clustering process might isolate highly dependent variables, (2) the size of the SPN and the processing time can grow exponentially with the number of variables, and (3) it necessitates an additional step to learn the weights.

- *LearnSPN* recursively partitions a dataset into independent variable subsets (chopping) and similar instance groups (slicing) to form the SPN graph [18]. This process, underpinned by two base cases, creates product and sum nodes. The first base case generates a terminal node with a univariate distribution when only a single variable remains after chopping. The second applies a naive Bayes factorization when slicing yields multiple columns with few rows. The algorithm is flexible, allowing for different methods of splitting and clustering. Notably, it initially employs the G-Test for splitting and hard incremental EM for clustering.

Our experimental setup employed an implementation grounded in the LearnSPN algorithm due to its versatility, resilience, and potential for customized fine-tuning. LearnSPN incorporates the following two notable hyperparameters that are crucial in modulating its functionality:

- `min-slice` dictates the smallest permissible number of rows in a leaf node, establishing the model's granularity and providing a halt condition for the split-slice routine. Selecting a smaller value may enable the capture of intricate data patterns, albeit with an increased risk of overfitting.

- `threshold` adjusts the strictness of the independence test during the variable splitting phase. A higher threshold infers a more liberal acceptance of significant associations between variables, potentially leading to a larger and more complex SPN. These hyperparameters present a valuable means to balance model complexity and its fit to the data.

In Section 5.2.1, we offer a series of experiments to fine-tune SPNs using these parameters.

Parameter Learning Once the structure of an SPN is learned, the subsequent step is to learn the network parameters. These include the weights associated with the sum nodes and the parameters of the distribution functions located in the leaf nodes—for instance, the means and covariances if the distributions are Gaussian. Parameter learning is performed using Maximum Likelihood Estimation (MLE) techniques. In essence, MLE seeks to select the parameters that maximize the likelihood of observing the given data under the model. This effectively means that the chosen parameters make the provided data as probable as possible given the SPN model. Two primary methods for implementing MLE in the context of parameter learning for SPNs are the EM algorithm and Gradient Descent, as we briefly explain next (see [53] for more detail):

- *The EM algorithm:* EM for parameter learning in SPN consists of the E-step, which calculates the expected log-likelihood of the data given the current parameter estimates, and the M-step, which maximizes this expectation to update the parameters. EM guarantees a non-decreasing likelihood with each iteration, leading to a local optimum.

- *Gradient descent:* Gradient descent and its variations are widely used for MLE in parameter learning for SPNs where the function to minimize is the negative log-likelihood [51]. The parameters are iteratively updated in the direction that reduces this function the most, guided by a learning rate. SPNs' differentiability allows for efficient computation of gradients, which are used to update the weights of sum nodes and distribution parameters in the leaf nodes. Although gradient descent is computationally efficient, it might only converge to a local minimum, depending on the initial parameter values.

3.3.3 Inference in SPNs

Inference in SPNs is pivotal in this thesis, enhancing data clustering effectiveness. Through marginal inference, SPNs assign clusters to data points based on the computed marginal probabilities of cluster variables given the observed data. Concurrently, conditional inference estimates the cluster membership probabilities, facilitating the identification of cluster assignments. This approach, harnessing the probabilistic modeling capabilities of SPNs, improves the clustering process and informs subsequent analyses in my work. The versatility of inference types supported by SPNs enables robust probabilistic reasoning and computation. For more information see [53].

- *Marginal inference* involves calculating the marginal probability of a specific variable or set of variables. It computes the probability distribution over a subset of variables by summing out the remaining variables.

- *Conditional inference* estimates the conditional probability of one or more variables given evidence or observed values for other variables. It calculates the probability distribution over the target variables conditioned on the observed values.

- *Maximum a posteriori (MAP)* aims to find the most probable assignment of values to a set of variables given evidence or observed values. It seeks to maximize the joint probability distribution over the variables, considering both the prior probabilities and the observed evidence.

- *Most Probable Explanation (MPE)* in SPNs refers to determining the assignment of the variable value that maximizes the joint probability distribution given observed evidence. It entails traversing the SPN, considering its structure, weights, and parameters, to compute probabilities for varying assignments. The assignment with the utmost probability is then chosen as the most probable explanation.

These different types of inference in SPNs provide a range of probabilistic reasoning capabilities, enabling tasks such as marginal and conditional probability estimation, MAP inference, sampling-based approximation, and updating probabilities based on evidence.

3.4 Accuracy Measures

Machine learning models are typically evaluated using a variety of measures that capture different aspects of the model's performance. This thesis uses the following measures: accuracy, precision, recall, F1 score, and AUC.

- *Accuracy* is the most straightforward metric, defined as the ratio of correctly predicted observations to the total observations. It generally measures how well the model performs across all classes.

- *Precision* is the ratio of correctly predicted positive observations to the total predicted positives. High precision indicates a low false positive rate, meaning the model is reliable when it predicts a positive class.

- *Recall (Sensitivity)* is the ratio of correctly predicted positive observations to all observations in the actual class. It captures the ability of the model to find all the positive instances.

- *F1 Score* or the F-measure is the harmonic mean of precision and recall, balancing the two metrics. It is beneficial in situations where the data has imbalanced classes [26].

- *AUC* stands for the area under the receiver operating characteristics curve or ROC. It illustrates the performance of a binary classifier as its discrimination threshold is varied. The AUC measures the entire two-dimensional area underneath the ROC curve, providing an aggregate performance measure across all possible classification thresholds [5, 27]. A model whose predictions are 100% wrong has an AUC of 0.0, while a model whose predictions are 100% correct has an AUC of 1.0.

Each of these metrics provides different insights, and their use depends on your machine learning task's specific objectives and constraints. The best measure will depend on these factors, as well as the particular characteristics of your data.

3.5 Fairness and Measures of Bias

The widespread use of machine learning models in critical decisions that significantly impact individuals' lives, such as in credit scoring, hiring, or criminal justice, has raised concerns about fairness. Fairness in machine learning refers to the absence of systematic bias or discrimination in the decisions made by machine learning models. A fair model does not disproportionately harm or benefit any particular group of individuals based on their protected attributes, such as race, gender, age, or religion. However, achieving fairness is challenging due to the presence of biases in training data that reflect historical or societal inequalities. Furthermore, fairness is subjective and context-dependent, with different stakeholders holding varied perspectives on what fairness entails in a given context. Multiple fairness measures have been proposed, though no single measure can capture all fairness aspects, and appropriateness depends on the context. A comprehensive evaluation may involve multiple metrics, considering their strengths and limitations.

Fairness assessment in machine learning can be approached from individual or group perspectives, resulting in two primary fairness categories: Individual and group fairness. Individual fairness pertains to treating similar individuals, while group fairness ensures fair outcomes for groups concerning a protected attribute. This work primarily focuses on improving group fairness, using various metrics, including:

- *Demographic parity (DP)* measures whether the positive outcome probability is the same across different groups concerning a protected attribute [59, 15].

- *Equal Opportunity (EO)* measures if the true positive rate is the same across different groups concerning a protected attribute [22].

Beyond DP and EO, several other standard fairness measures, such as Equalized Odds (EOD) [22], Treatment Equality (TE) [1], False Positive Rate Equality (FPR Equality) [12], False Negative Rate Equality (FNR Equality) [12], Overall Accuracy Equality (OAE) [1], and Disparate Impact (DI) [59], contribute to different facets of fairness. A single measure cannot encapsulate all fairness aspects; hence, the context determines the suitability of each measure. A thorough fairness assessment may necessitate multiple metrics, carefully considering their relative strengths and application-specific limitations.

This thesis employs EO and DP. These widespread measures are conceptually simple, straightforward to implement, and lay a robust foundation for fairness assessment in machine learning models. Primarily, they guarantee outcome equality for protected groups, serving as this work's main criteria for fairness evaluation.

Chapter 4

Methodology

This chapter provides a definition of data heterogeneity and explains how we can measure its impact on fairness and bias systematically and principledly.

4.1 Problem Description

The problem we aim to address in this study is the detection of data heterogeneity through distribution-based clustering using Maximum Likelihood Estimation (MLE) and investigating its subsequent effect on fairness. Given a dataset D characterized by a set of attributes X, the goal is to identify k distributions, denoted as $P_1, P_2, \ldots, P_k$, that are most likely to have generated the observed data D. Each distribution, P_i, is parameterized (for instance, a Gaussian distribution defined by mean and variance parameters).

In data heterogeneity, the parameter k holds significant relevance. A value of $k > 1$ suggests data heterogeneity as it implies the data originates from more than one distribution. However, this formulation's primary challenge lies in that an MLE-based solution tends to favor multiple distributions, leading to overfitting. This issue is mitigated by cross-validation to ensure that the likelihood calculation for determining k is performed on unseen data.

Whether the data is heterogeneous can be determined through statistical tests, contingent on the specific application domain. Therefore, this thesis focuses on how MLE and distribution-based clustering can identify the most likely set of clusters and the most probable k for the underlying data generative processes.

Let's denote the dataset comprising n data points as $D = x_1, x_2, \ldots, x_n$. The aim is to estimate the parameters of the k distributions $P_1, P_2, \ldots, P_k$, denoted as $\theta_1, \theta_2, \ldots, \theta_k$, using MLE. The parameters θ_i define the respective distributions P_i. The likelihood function $L(D \mid \theta)$ represents the probability of observing the dataset D given the parameters θ: $L(D \mid \theta) = \prod_{i=1}^{n} p(x_i \mid \theta)$, where $p(x \mid \theta)$ represents the probability density function of the data point x based on the parameters θ. The goal is to discover the set of parameters $\theta_1^*, \theta_2^*, ..., \theta_k^*$ that maximize the likelihood of the observed data:

$$\theta_1^*, \theta_2^*, \ldots, \theta_k^* = \arg \max_{\theta_1, \theta_2, \ldots, \theta_k} L(D \mid \theta) \tag{4.1}$$

This optimization problem can be addressed using algorithms such as the EM algorithm, which iteratively estimates the parameters and assigns data points to the most probable distribu-

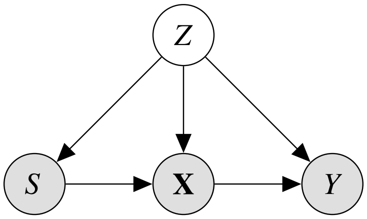


Figure 4.1: A graphical model representing the heterogeneity bias problem. In the model, **X** is the set of input features, S is the protected attribute, Z is a latent variable that manifests data heterogeneity, and Y is the output or label attribute.

tions. We aim to identify the parameterized distributions that most accurately capture the data's heterogeneity, shedding light on the underlying patterns and structures within the dataset.

The resulting clustering introduces a latent variable Z that represents data heterogeneity. Given the protected attribute S, we can analyze heterogeneity's impact on fairness by assessing the correlation between Z and S. A high correlation could indicate biases in the data, as illustrated in Figure 4.1. We utilize the conditional entropy of S given Z to measure this correlation.

The conditional entropy of S given Z measures the average amount of information needed to specify the protected attribute S given that the latent value of Z is known. Formally, the conditional entropy of S given Z, denoted as $H(S \mid Z)$, is defined as follows:

$$H(S \mid Z) = \sum_{z \in Z} p(z) \times H(S \mid Z = z) \tag{4.2}$$

for all z in the domain of Z, where $p(z)$ is the probability mass function of Z, and $H(S \mid Z = z)$ is the entropy of S given $Z = z$. $H(S \mid Z = z)$ is computed as:

$$H(S \mid Z = z) = -\sum_{s \in S} p(s \mid z) \times \log(p(s \mid z)) \tag{4.3}$$

for all s in the domain of S, where $p(s \mid z)$ is the conditional probability mass function of S given $Z = z$. In other words, the conditional entropy of S given Z is the expected entropy of S over all possible values of Z. The conditional entropy $H(S|Z)$ is always less than or equal to the entropy $H(S)$, and it is equal to $H(S)$ if and only if S and Z are independent. The difference between $H(S)$ and $H(S|Z)$, denoted as $I(S;Z)$, is the mutual information between S and Z, which measures the amount of information shared by S and Z.

Here is your revised chunk with the title of the section explaining the use of MMs updated:

4.2 Detecting Data Heterogeneity

We frame the detection of heterogeneity as a problem of distribution-based clustering, employing MMs and SPNs to implement the clustering. This section provides an overview of how these two models detect heterogeneity in a given dataset.

4.2.1 Leveraging MM Components for Detecting Heterogeneity

Detecting data heterogeneity using MMs hinges on accurately determining the number of components in the model. Each component encapsulates a portion of the data's heterogeneity, signifying a unique subgroup within the broader data distribution. Specifying the number of components in the MM is typically accomplished using validation data through methods such as cross-validation. This procedure serves to prevent both oversimplification and overfitting. By evaluating the model's likelihood for varying numbers of components, the model that renders the highest likelihood on the validation data can be chosen, thereby providing the optimal number of components.

Upon establishing the MM, it acts as a quantitative measure of data heterogeneity, where the number of components directly signifies the level of heterogeneity. Furthermore, statistical tests can be performed on the model's likelihood values for a formalized measure of heterogeneity. For instance, likelihood ratio tests can be utilized to compare the fitting of models with differing numbers of components, yielding a statistically validated estimate of data heterogeneity.

4.2.2 Clustering with SPN

Distribution-based clustering can be reframed as an MPE problem in SPNs by conceptualizing it as an assignment problem where each data point is assigned to a specific cluster. This transformation entails constructing an SPN that models the joint probability distribution over the data points and cluster assignments, followed by identifying the assignment of cluster labels that maximize the posterior probability given the observed data.

For data points, $D = \{x_1, x_2, \ldots, x_n\}$ and clusters $C = \{c_1, c_2, \ldots, c_k\}$, the objective is to find the assignment of cluster labels C to the data points D that maximizes $P(C, D)$. This can be articulated as:

$$\arg\max_{C} P(C, X) = \arg\max_{C} P(C \mid X) \cdot P(X), \tag{4.4}$$

where $P(C \mid X)$ represents the posterior probability of the cluster assignments given the data points, and $P(X)$ is the marginal probability of the data points.

To transform clustering into an MPE problem in SPNs, an SPN is constructed that models the joint probability distribution over the data points and cluster assignments denoted as $P(C, X)$. The SPN encapsulates the dependencies between the data points and clusters, enabling probabilistic reasoning about cluster assignments.

The MPE problem in SPNs involves discovering the assignment of cluster labels C that maximizes $P(C \mid X)$. This is accomplished by executing inference in the SPN, considering the structure, weights, and parameters associated with the variables. The most probable clustering configuration can be determined by computing the probabilities for different cluster assignments and selecting the one with the maximum probability. Subsequently, the values of these features are estimated using SPN MPE inference.

The values of augmented variables in the selective SPN are estimated and employed to segregate the data into homogeneous subpopulations. To avoid very small subpopulations, feature selection is performed to select a subset of augmented variables that are more important in a prediction task. The values of the selected augmented variables are consolidated into a single latent variable. We use recursive feature elimination (RFE) as the feature selection method. RFE initially determines the importance of each feature using an estimator that determines the importance of each feature. This algorithm then removes the least important features from the current feature set. This process is iteratively applied to the pruned set until the desired number of selected features is achieved.

To further refine the division of the data into homogenous subpopulations, the likelihood of the chosen augmented features conditioned on the values of the original features is calculated. These probabilities are used to probabilistically divide the data into homogenous groups, offering a more comprehensive understanding of the relationship between the input features and the target class attribute.

Chapter 5

Experiments

In this experimental chapter of the thesis, we focus on three primary objectives, utilizing several real-world datasets:

- The first objective revolves around fine-tuning Mixture Models (MMs) and Sum-Product Networks (SPNs) to detect data heterogeneity effectively. Our aim here is to identify the optimal clustering of the datasets and ascertain the latent attribute associated with this clustering.
- The second objective involves calculating the correlation between the latent attribute and the protected attribute in the datasets. This calculation will help determine whether data heterogeneity and inherent biases are intertwined.
- Finally, our third objective examines the impact of incorporating the latent variable and the detected data heterogeneity on various prevalent downstream prediction models. This analysis will provide insights into our approach's potential benefits and challenges.

5.1 Experimental Setup

In this section, we outline our experimental setup. We detail datasets, models, and configurations used in our experiments, highlighting the rationale behind these decisions to ensure the clarity and reproducibility of our work.

5.1.1 Datasets

We initiate our experimental analysis with a comprehensive discussion of the datasets utilized. Four distinct datasets – UCI Adult, ASC Income, German Credit, and COMPAS – each, with its unique contributions, form the cornerstone of our exploration into data heterogeneity. Table 5.1 highlights the key statistics for these datasets, providing a clear understanding of their characteristics and respective roles in our experiments.

The datasets used in the experiments were retrieved from various sources. The German Credit and UCI Adult datasets were obtained from the UCI archive website[1], while the COM-

[1] https://archive.ics.uci.edu/

PAS dataset was retrieved from the ProPublica website[2]. As for the ACS Income dataset, we accessed it from the Folktables [3] library, a Python package that facilitates benchmarking of machine learning algorithms by providing access to datasets derived from the US Census.

UCI Adult Dataset: The UCI Adult dataset is frequently used as a binary classification task benchmark. Comprising 32,000 records and 14 features, including age, work class, education level, marital status, occupation, race, gender, and weekly work hours, it provides a rich source of information. The class label, "income," is represented as "$\leq 50K$" or "$> 50K$," denoting if an individual's annual income falls below or above $50,000. Race and gender are sensitive demographic attributes. Notably, the UCI Adult dataset demonstrates class imbalance, with a smaller number of individuals earning above $50,000, which could affect the performance of classification models built on this dataset.

ACS Income Dataset: The ACS Income dataset, sourced from the annual American Community Survey (ACS) conducted by the U.S. Census Bureau, represents a large-scale income data collection. It comprises millions of records covering income, demographics, education, occupation, and employment status across the United States. Unlike the UCI Adult dataset, the ACS Income class label is a continuous variable, enabling flexibility in establishing income thresholds for classification. We extract a subset of 31,000 samples from this dataset for our experiments.

German Credit Dataset: The German Credit dataset, used extensively in credit risk analysis and classification tasks, consists of 1,000 records and 20 features. These features illuminate various facets of credit applications, providing valuable information on the applicants' demographics, financial status, and credit-related details. The credit score class label is bifurcated into "Good" and "Bad," indicating the credit application's approval status. Sensitive attributes include age, categorized as young (below 25 years) and old (above 25 years), and sex. It's worth noting that the dataset presents class imbalance, with approximately 70% of applications denied, requiring careful handling to ensure accurate analysis.

Propablica COMPAS Dataset: The COMPAS Recidivism dataset, designed for predicting future crime propensity, includes records of approximately 6,000 defendants from Broward County, Florida. With 137 features covering demographics, criminal history, and social history, it provides a rich dataset for predictive modeling. However, when employing this dataset, the potential bias and fairness issues concerning race need careful consideration. Previous studies have noted potential racial bias in the COMPAS algorithm, with a higher misclassification rate for African-American defendants, underlining the importance of thorough evaluation and bias mitigation when using such data.

[2] https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis

[3] https://github.com/socialfoundations/folktables

Table 5.1: Datasets statistics

Dataset	#Records	#Features	Class balance (%)	Protected	Label
German	1,000	20	69.97	Gender, Age	Credit [Good/Bad]
COMPAS	6,000	137	54.49	Gender, Race	Recidivism [Yes/No]
ACS Income	31,000	14	56.74	Gender, Race	Salary [High/Low]
Adult	32,000	14	75.92	Gender, Race	Salary [High/Low]

5.1.2 Implementation and Evaluation of Predictive Models

To investigate the influence of data heterogeneity on downstream predictive tasks, we examine three widely used predictive models: Logistic Regression (LR), Random Forest (RF), and Support Vector Machines (SVM). Our experiments are executed using Python version 3.10.9, leveraging several external libraries that offer robust implementations of these models. We primarily utilize Scikit-learn[4] and SPFlow[5], known for their efficient, reliable functionalities and thorough documentation. Specifically, we employ Scikit-learn's GaussianMixture, LR, RF, and SVM implementations, while SPFlow provides SPN structure learning capabilities. We use the RFE feature selection implementation of Scikit-learn and a Scikit-learn prediction model, such as a random forest, as the estimator. Evaluation of these models entails the computation of various accuracy metrics, for which we also employ Scikit-learn's metrics functionalities. For conditional entropy calculations, we turn to the SciPy library.[6] Lastly, we utilize the Fairlearn library[7] for calculating fairness metrics, ensuring a thorough evaluation of our predictive models.

5.2 Experimental Results

In this section, we present the outcomes of our experiments, examining the effects of data heterogeneity on the selected predictive models. We discuss the results in correlation with our experimental design, offering insights into the impact of data heterogeneity on downstream prediction tasks.

5.2.1 Tuning SPN's Hyperparemeters

To learn an SPN utilizing the SPFlow library, two hyperparameters must be tuned: `threshold` and `min-slice`. A two-level grid search is conducted to approximate the values that optimize the probability of the validation data given the training data parameters. The grid search experiment, depicted in Figures 5.1 and 5.2, was executed on each dataset, with each sensitive attribute examined independently. Since the chosen sensitive attribute is omitted from the SPN, it could modify the model's structure and parameters depending on how informative that attribute is. The sensitive attribute remains unused throughout the clustering process in order to prevent any compromise on fairness.

[4] https://scikit-learn.org

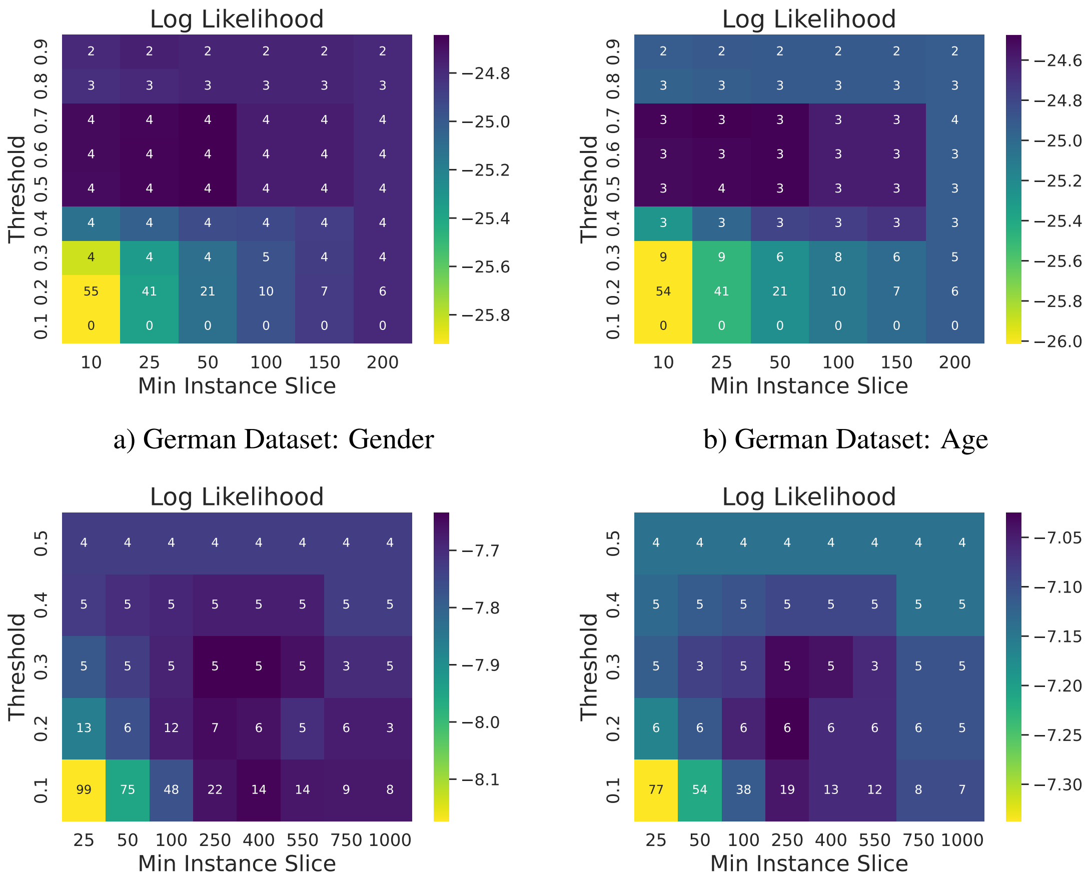
[5] https://spflow.github.io

[6] https://scipy.org/

[7] https://fairlearn.org/v0.5.0

The optimal values for the SPN's hyperparameters for each dataset and sensitive attribute are displayed in Table 5.2. With these hyperparameters set, the final number of clusters is also determined. The parameter noc refers to the number of clusters the tuned SPN returns. It is observed that the German Credit dataset has fewer clusters compared to other datasets.

Takeaway: Experimental results confirm that higher values of threshold and min-slice result in fewer clusters. This is because the SPN structure learning process would be more stringent in deciding whether some variables are independent of each other, and the tree would not be as deep. In other words, higher values SPN hyperparameters would quickly assume the clusters are homogeneous. Another point worth mentioning is that datasets with small sample sizes, e.g., German Credit, are susceptible to overfitting and will have fewer clusters.



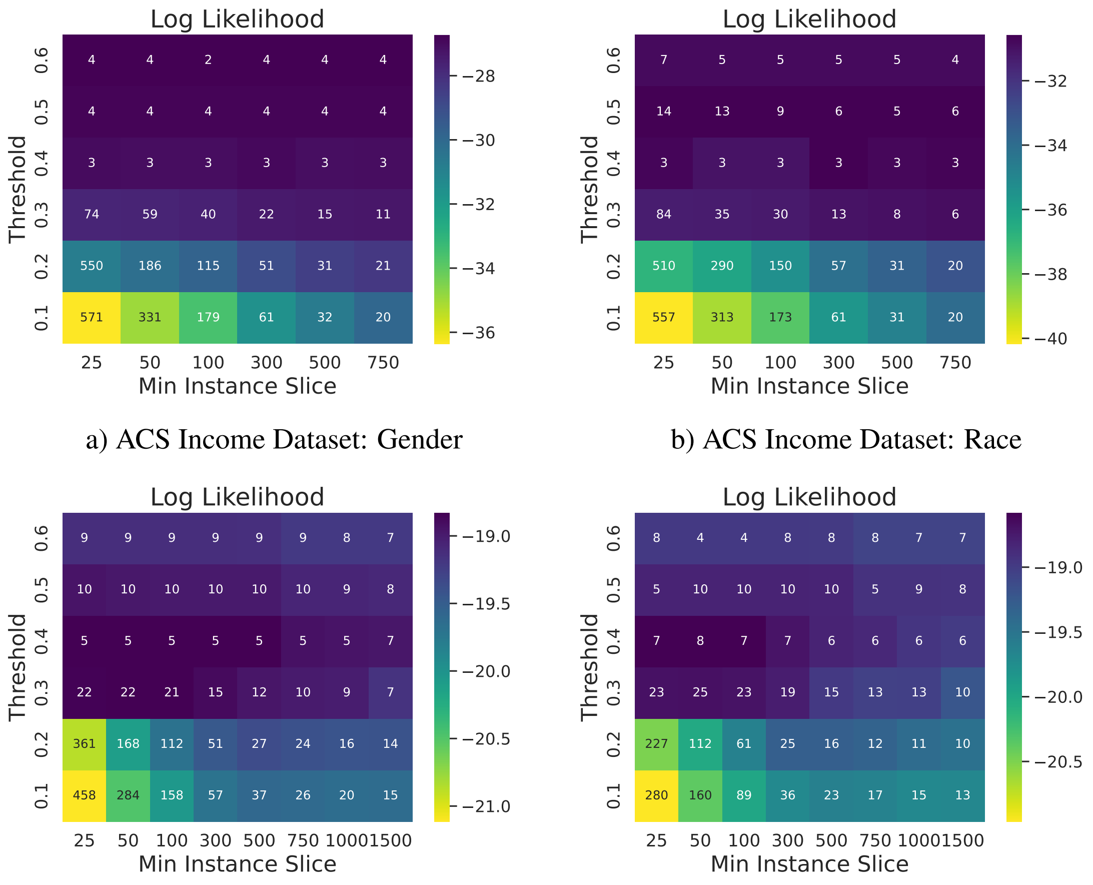
a) German Dataset: Gender

b) German Dataset: Age

c) COMPAS Dataset: Gender

d) COMPAS Dataset: Race

Figure 5.1: A heatmap of grid search for finding optimal values of the tuning parameters min-slice and threshold in German Credit and COMPAS. Each subfigure shows the grid search for a dataset and a sensitive attribute. The number of clusters based on the retrieved latent variable can be seen as an integer on each corresponding cell.



a) ACS Income Dataset: Gender

b) ACS Income Dataset: Race

c) UCI Adult Dataset: Gender

d) UCI Adult Dataset: Race

Figure 5.2: A heatmap of grid search for finding optimal values of the tuning parameters `min-slice` and `threshold` in ACS Income and UCI Adult.

Table 5.2: Best parameters values for each dataset and sensitive attribute

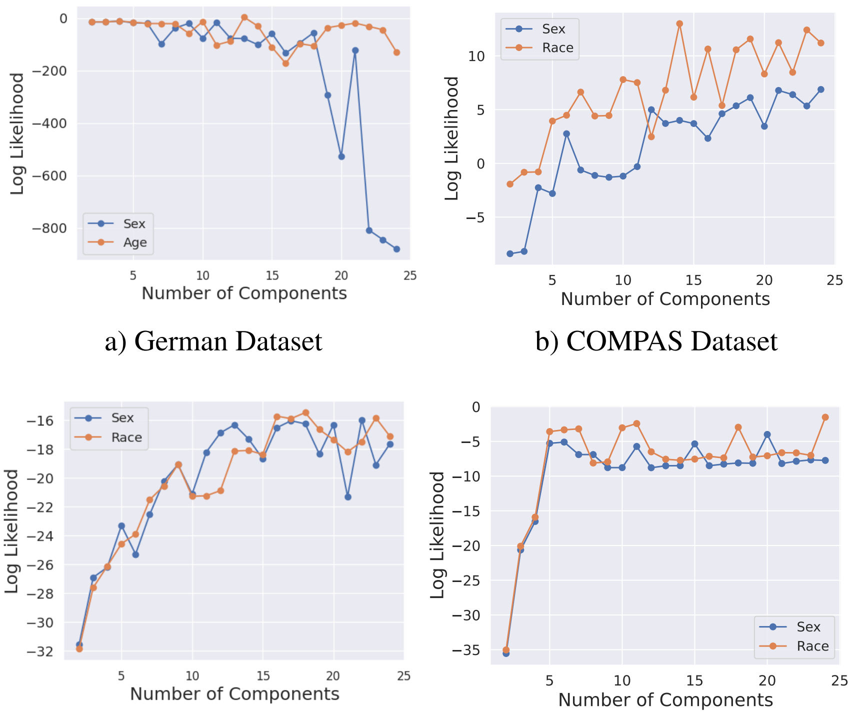
Dataset	**Sensitive Attribute**	`threshold`	`min-slice`	`noc`
German	Gender	0.5	50	3
German	Age	0.7	25	4
COMPAS	Gender	0.3	400	5
COMPAS	Race	0.2	250	6
ACS Income	Gender	0.6	25	4
ACS Income	Race	0.5	25	14
Adult	Gender	0.4	300	5
Adult	Race	0.4	50	8

5.2.2 Tuning MM's Hyperparemeters

Finite MMs need the number of clusters, shown by `noc`, as an input. This hyperparameter is tuned using a simple search with varying number of components to determine its optimal value. The goal is to identify the value that enhances the probability of the validation data, given the model parameters learned from the training data.

Figure 5.3 demonstrates how the log-likelihood fluctuates with different values of `noc`. The optimal value selected for each dataset and sensitive attribute is displayed in Table 5.3. Besides, Figure 5.4 depicts the distribution of data samples across each cluster as classified by the latent variable. Table 5.4 compares the log-likelihood of the optimal clusters in SPN vs. MMs.

Takeaway: The diagrams in Figure 5.4 reveal that the MM cluster size distribution is imbalanced, with a few clusters containing numerous data points and many clusters having a small size. Where MMs outperformed SPNs based on this clustering criteria. Also, similar to SPN hyperparameter tuning, the distribution-based clustering models on datasets with small sample sizes such as German Credit, will get overfitted fast. After tuning both models, by comparing tuned SPN and MM's log-likelihood on the data in Table 5.4, we decide that distributions learned by MMs are more likely to have generated the data. Hence, MMs are better equipped to handle clustering in this work. Another insight from both models' final number of cluster values is that the datasets are detected to be heterogeneous as the models need to learn multiple distributions rather than one to be able to represent the data.



a) German Dataset

b) COMPAS Dataset

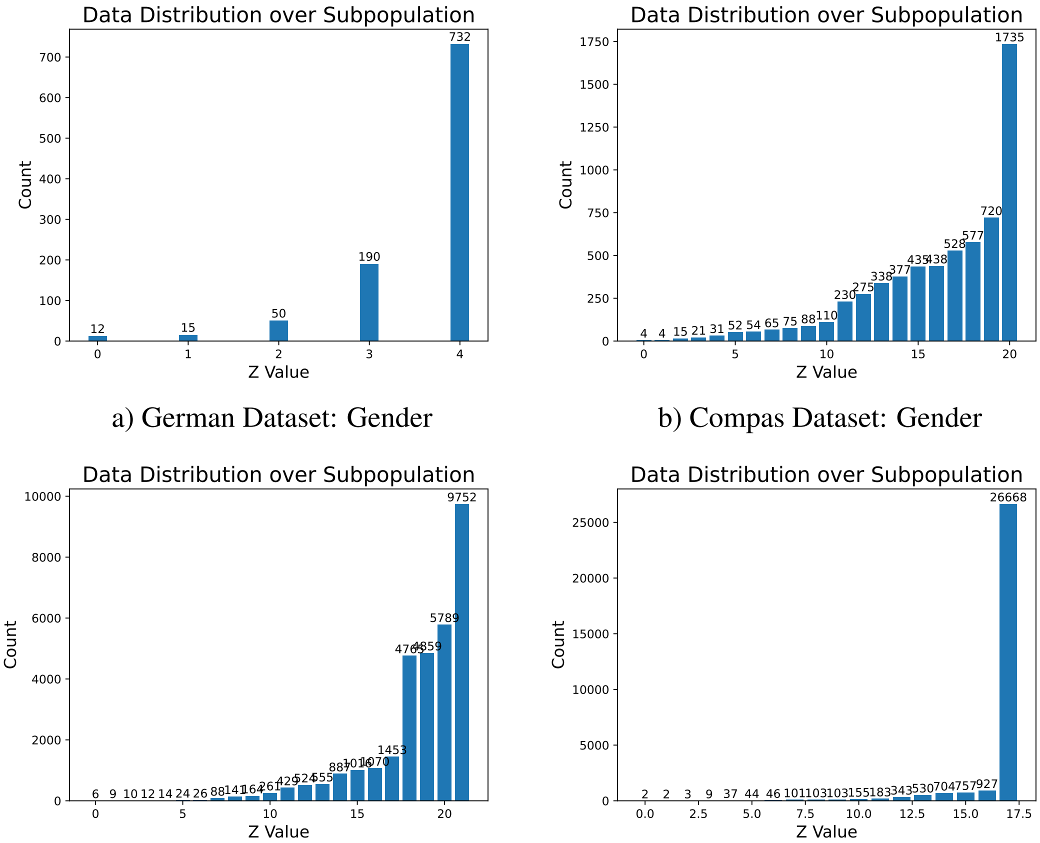
c) ACS Income Dataset

d) UCI Adult Dataset

Figure 5.3: Log likelihood of Gaussian mixture models with varying `noc`.

Table 5.3: Best parameter values for each dataset and sensitive attribute

Dataset	Sensitive Attribute	noc
German	Gender	5
German	Age	4
COMPAS	Gender	21
COMPAS	Race	14
ACS Income	Gender	22
ACS Income	Race	18
Adult	Gender	18
Adult	Race	24



a) German Dataset: Gender

b) Compas Dataset: Gender

c) ACS Income Dataset: Gender

d) UCI Adult Dataset: Gender

Figure 5.4: Number of data samples for each cluster for the best noc value

Table 5.4: Log likelihood of tuned SPN and MM

Dataset	Sensitive Attribute	SPN	MM
German	Gender	-24.64	-18.11
German	Age	-24.47	-11.78
COMPAS	Gender	-7.63	6.78
COMPAS	Race	-7.64	6.49
ACS Income	Gender	-26.73	-17.65
ACS Income	Race	-26.83	-16.55
Adult	Gender	-18.82	-8.10
Adult	Race	-18.58	-1.51

5.2.3 Exploring Sensitive and Latent Attribute Relations

This section will uncover the potential correlation between sensitive attributes and the identified latent variables. To achieve this, we calculate the mutual information between these variables and the conditional entropy of the sensitive attribute given the latent variable. We then compare these values for the MM and SPN with the equivalent values obtained from random clustering.

The experiment result, including the correlation between the two variables, is shown in Table 5.5 for a given dataset and a sensitive attribute. The result indicates a lack of correlation between the latent variable and the sensitive attribute. The similarity of these values to those obtained from random clustering, which is equivalent to $H(S)$ in the case of entropy, serves as the supporting proof for the observation.

Takeaway: The lack of correlation between the two variables suggests that incorporating the learned clusterings will not create bias in the prediction tasks. This means that the fairness of prediction models should not deteriorate when using the latent variable. This finding is essential as it supports the fairness of our modeling approach, reaffirming the efficacy of the MM and SPN in managing sensitive attributes during model formulation. Further exploration and validation of this relationship would be necessary to generalize these results, potentially using a broader range of datasets and varying the selection of sensitive and latent attributes.

Table 5.5: Conditional entropy (CE) and mutual information of sensitive and latent attributes

Dataset	**Sensitive**	**SPN CE**	**MM CE**	**Random CE**	**SPN MI**	**MM MI**	**Random MI**
German	Gender	0.89	0.88	0.89	0.0077	0.0173	0.0032
German	Age	0.67	0.70	0.70	0.0279	0.0002	0.0001
COMPAS	Gender	0.70	0.60	0.70	0.0062	0.0084	0.0007
COMPAS	Race	1.55	1.57	1.60	0.0590	0.0346	0.0014
ACS Income	Gender	0.99	0.97	1.00	0.0054	0.0273	0.0002
ACS Income	Race	0.75	0.73	0.75	0.0013	0.0218	0.0002
Adult	Gender	0.91	0.9	0.91	0.0000	0.0080	0.0005
Adult	Race	0.79	0.79	0.79	0.0005	0.0052	0.0004

5.2.4 Evaluating Models

Our methodology outlines a process where an SPN is trained on the data, augmented for selectivity, and subjected to a feature selection algorithm. The resulting subset of augmented variables is estimated for each data record via SPN's inference and forms a latent variable within the dataset. This latent variable is used to partition the data into subgroups so that we can train different prediction models for each subgroup. After clustering data using MMs and SPNs and choosing the best clustering, we assess two strategies: a "single model" approach, which uses one model without the latent variable, and a "multiple models" approach, where a suite of models is trained on subgroups defined by the latent variable. The single model serves as a baseline for comparison. Finally, accuracy and fairness metrics, including equalized odds and demographic parity, are computed to compare the two approaches.

Fairness considerations led us to examine any correlation between the latent variable and sensitive attributes. Tables 5.6- 5.13 present the performance of base models—LR, RF, and SVM—across various datasets. Notably, accuracy improved significantly. The increase in accuracy and F1 score is highlighted in the tables. For instance, the models saw increases of roughly 1% (Table 5.6, and 5.7), 30% (Table 5.8, and 5.9), 20% (Table 5.10, and 5.11), and 2% (Table 5.12, and 5.13) for the German Credit, COMPAS, ACS Income, and UCI Adult datasets, respectively. Fairness measures, such as DP and EO, also demonstrated slight enhancements for the COMPAS, Income, and Adult datasets and remained unchanged for the German dataset. This observation supports the results in Table 5.5, showing no correlation between latent variables and sensitive attributes, leading us to anticipate minimal impact on fairness metrics.

Takeaway: Our preliminary data analysis suggested latent variables could improve model performance, with ensemble models surpassing baseline models. The increase in accuracy by using multiple models on the partitioned data confirms this theory and shows that distribution-based clustering did capture the inherent heterogeneity within the data. On the other hand, the results show that this approach did not deteriorate the fairness, confirming that cluster labels and sensitive attributes are not correlated. This point was demonstrated in the previous experiment and is why fairness does not alter significantly compared to learning a model on the whole dataset. Another point worth mentioning is that an RF prediction model is more accurate than a single LR or SVM model on most datasets. This performance advantage can be attributed to the ensemble nature of RF, which excels at grasping the complexity of data, resulting in enhanced predictive capabilities.

Table 5.6: German & Gender evaluation

Model	Accuracy	F1	Precision	Recall	AUC	EO	DP
Single LR	0.71 ± 0.02	0.34 ± 0.07	0.53 ± 0.07	0.25 ± 0.06	0.69 ± 0.03	0.16 ± 0.07	0.06 ± 0.07
Multiple LR	0.72 ± 0.04	0.39 ± 0.08	0.57 ± 0.14	0.29 ± 0.07	0.72 ± 0.02	0.17 ± 0.07	0.09 ± 0.09
Single RF	0.76 ± 0.01	0.51 ± 0.05	0.66 ± 0.05	0.42 ± 0.06	0.77 ± 0.04	0.08 ± 0.04	0.06 ± 0.04
Multiple RF	0.76 ± 0.02	0.50 ± 0.06	0.67 ± 0.09	0.40 ± 0.06	0.77 ± 0.03	0.09 ± 0.06	0.07 ± 0.05
Single SVM	0.67 ± 0.02	0.46 ± 0.06	0.45 ± 0.05	0.47 ± 0.09	0.65 ± 0.04	0.05 ± 0.01	0.02 ± 0.02
Multiple SVM	0.67 ± 0.04	0.46 ± 0.09	0.45 ± 0.08	0.48 ± 0.10	0.66 ± 0.05	0.09 ± 0.06	0.06 ± 0.04

Table 5.7: German & Age evaluation

Model	Accuracy	F1	Precision	Recall	AUC	EO	DP
Single LR	0.71 ± 0.02	0.34 ± 0.07	0.53 ± 0.07	0.25 ± 0.06	0.69 ± 0.03	0.12 ± 0.08	0.08 ± 0.04
Multiple LR	0.71 ± 0.02	0.35 ± 0.07	0.54 ± 0.09	0.27 ± 0.07	0.68 ± 0.02	0.22 ± 0.15	0.10 ± 0.04
Single RF	0.76 ± 0.01	0.51 ± 0.05	0.66 ± 0.05	0.42 ± 0.06	0.77 ± 0.04	0.14 ± 0.07	0.07 ± 0.07
Multiple RF	0.74 ± 0.01	0.46 ± 0.05	0.61 ± 0.08	0.37 ± 0.05	0.75 ± 0.05	0.14 ± 0.07	0.10 ± 0.11
Single SVM	0.67 ± 0.02	0.46 ± 0.06	0.45 ± 0.05	0.47 ± 0.09	0.65 ± 0.04	0.10 ± 0.07	0.06 ± 0.06
Multiple SVM	0.67 ± 0.02	0.47 ± 0.05	0.46 ± 0.06	0.48 ± 0.06	0.67 ± 0.03	0.16 ± 0.07	0.09 ± 0.11

Table 5.8: COMPAS & Gender evaluation

Model	Accuracy	F1	Precision	Recall	AUC	EO	DP
Single LR	0.68 ± 0.01	0.62 ± 0.01	0.68 ± 0.01	0.57 ± 0.02	0.74 ± 0.01	0.11 ± 0.06	0.12 ± 0.03
Multiple LR	0.96 ± 0.01	0.96 ± 0.01	0.98 ± 0.01	0.94 ± 0.02	0.99 ± 0.00	0.02 ± 0.02	0.13 ± 0.04
Single RF	0.67 ± 0.02	0.61 ± 0.03	0.66 ± 0.01	0.56 ± 0.04	0.72 ± 0.02	0.10 ± 0.04	0.11 ± 0.01
Multiple RF	0.96 ± 0.01	0.96 ± 0.01	0.97 ± 0.01	0.94 ± 0.02	0.99 ± 0.00	0.03 ± 0.02	0.13 ± 0.04
Single SVM	0.66 ± 0.02	0.60 ± 0.02	0.66 ± 0.01	0.55 ± 0.03	0.66 ± 0.02	0.09 ± 0.05	0.10 ± 0.03
Multiple SVM	0.96 ± 0.01	0.96 ± 0.01	0.97 ± 0.01	0.94 ± 0.03	0.99 ± 0.00	0.03 ± 0.02	0.13 ± 0.04

Table 5.9: COMPAS & Race evaluation

Model	Accuracy	F1	Precision	Recall	AUC	EO	DP
Single LR	0.68 ± 0.01	0.62 ± 0.01	0.68 ± 0.01	0.57 ± 0.02	0.74 ± 0.01	0.87 ± 0.14	0.61 ± 0.12
Multiple LR	0.97 ± 0.01	0.96 ± 0.01	0.98 ± 0.02	0.95 ± 0.02	1.00 ± 0.00	0.82 ± 0.39	0.51 ± 0.11
Single RF	0.67 ± 0.02	0.61 ± 0.03	0.66 ± 0.01	0.56 ± 0.04	0.72 ± 0.02	0.81 ± 0.15	0.57 ± 0.14
Multiple RF	0.97 ± 0.01	0.97 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	1.00 ± 0.00	0.80 ± 0.43	0.47 ± 0.09
Single SVM	0.66 ± 0.02	0.60 ± 0.02	0.66 ± 0.01	0.55 ± 0.03	0.66 ± 0.02	0.81 ± 0.15	0.56 ± 0.14
Multiple SVM	0.97 ± 0.02	0.97 ± 0.02	0.98 ± 0.01	0.96 ± 0.02	1.00 ± 0.00	0.80 ± 0.42	0.47 ± 0.09

Table 5.10: ACS Income & Gender evaluation

Model	Accuracy	F1	Precision	Recall	AUC	EO	DP
Single LR	0.76 ± 0.00	0.72 ± 0.01	0.72 ± 0.01	0.71 ± 0.01	0.84 ± 0.00	0.05 ± 0.01	0.06 ± 0.02
Multiple LR	0.98 ± 0.03	0.98 ± 0.03	0.98 ± 0.03	0.98 ± 0.03	1.00 ± 0.00	0.01 ± 0.01	0.17 ± 0.02
Single RF	0.80 ± 0.01	0.76 ± 0.01	0.77 ± 0.01	0.76 ± 0.01	0.88 ± 0.00	0.04 ± 0.01	0.14 ± 0.02
Multiple RF	0.99 ± 0.02	0.99 ± 0.02	0.99 ± 0.02	0.98 ± 0.02	1.00 ± 0.00	0.00 ± 0.01	0.17 ± 0.02
Single SVM	0.73 ± 0.01	0.69 ± 0.01	0.67 ± 0.02	0.72 ± 0.01	0.78 ± 0.01	0.03 ± 0.02	0.10 ± 0.03
Multiple SVM	0.98 ± 0.03	0.98 ± 0.03	0.98 ± 0.03	0.98 ± 0.03	1.00 ± 0.01	0.00 ± 0.00	0.17 ± 0.02

Table 5.11: ACS Income & Race evaluation

Model	Accuracy	F1	Precision	Recall	AUC	EO	DP
Single LR	0.76 ± 0.00	0.72 ± 0.01	0.72 ± 0.01	0.71 ± 0.01	0.84 ± 0.00	0.07 ± 0.01	0.10 ± 0.02
Multiple LR	0.96 ± 0.03	0.95 ± 0.03	0.96 ± 0.02	0.93 ± 0.04	0.99 ± 0.01	0.04 ± 0.03	0.10 ± 0.01
Single RF	0.80 ± 0.01	0.76 ± 0.01	0.77 ± 0.01	0.76 ± 0.01	0.88 ± 0.00	0.06 ± 0.01	0.10 ± 0.01
Multiple RF	0.97 ± 0.02	0.96 ± 0.02	0.97 ± 0.02	0.96 ± 0.03	1.00 ± 0.00	0.03 ± 0.02	0.10 ± 0.02
Single SVM	0.73 ± 0.01	0.69 ± 0.01	0.67 ± 0.02	0.72 ± 0.01	0.78 ± 0.01	0.06 ± 0.02	0.09 ± 0.03
Multiple SVM	0.95 ± 0.03	0.94 ± 0.04	0.94 ± 0.04	0.94 ± 0.04	0.99 ± 0.01	0.03 ± 0.02	0.09 ± 0.01

Table 5.12: Adult & Gender evaluation

Model	Accuracy	F1	Precision	Recall	AUC	EO	DP
Single LR	0.84 ± 0.00	0.63 ± 0.01	0.73 ± 0.01	0.55 ± 0.02	0.89 ± 0.00	0.24 ± 0.05	0.20 ± 0.01
Multiple LR	0.85 ± 0.01	0.66 ± 0.01	0.77 ± 0.02	0.58 ± 0.01	0.90 ± 0.01	0.21 ± 0.06	0.19 ± 0.01
Single RF	0.85 ± 0.00	0.68 ± 0.01	0.74 ± 0.01	0.63 ± 0.01	0.90 ± 0.00	0.09 ± 0.02	0.18 ± 0.01
Multiple RF	0.86 ± 0.00	0.70 ± 0.01	0.75 ± 0.01	0.65 ± 0.01	0.91 ± 0.01	0.08 ± 0.01	0.18 ± 0.01
Single SVM	0.81 ± 0.01	0.60 ± 0.01	0.63 ± 0.01	0.57 ± 0.02	0.83 ± 0.01	0.11 ± 0.03	0.16 ± 0.02
Multiple SVM	0.82 ± 0.01	0.63 ± 0.02	0.66 ± 0.03	0.61 ± 0.02	0.86 ± 0.01	0.07 ± 0.03	0.16 ± 0.02

Table 5.13: Adult & Race evaluation

Model	Accuracy	F1	Precision	Recall	AUC	EO	DP
Single LR	0.84 ± 0.00	0.63 ± 0.01	0.73 ± 0.01	0.55 ± 0.02	0.89 ± 0.00	0.39 ± 0.31	0.21 ± 0.06
Multiple LR	0.88 ± 0.06	0.74 ± 0.14	0.81 ± 0.10	0.68 ± 0.18	0.93 ± 0.04	0.43 ± 0.28	0.23 ± 0.04
Single RF	0.85 ± 0.00	0.68 ± 0.01	0.74 ± 0.01	0.63 ± 0.01	0.90 ± 0.00	0.28 ± 0.14	0.19 ± 0.04
Multiple RF	0.89 ± 0.06	0.77 ± 0.13	0.80 ± 0.11	0.73 ± 0.15	0.94 ± 0.04	0.28 ± 0.21	0.19 ± 0.04
Single SVM	0.81 ± 0.01	0.60 ± 0.01	0.63 ± 0.01	0.57 ± 0.02	0.83 ± 0.01	0.38 ± 0.27	0.18 ± 0.03
Multiple SVM	0.86 ± 0.07	0.71 ± 0.15	0.73 ± 0.14	0.70 ± 0.17	0.90 ± 0.05	0.38 ± 0.25	0.19 ± 0.04

Chapter 6

Conclusion and Future Work

This chapter presents final conclusions that were acquired by conducting and observing the experimental results. Furthermore, it provides the possible future research directions.

6.1 Conclusion

Data heterogeneity, referring to variations in the underlying data distribution, can cause many complications in prediction models. These complications range from accuracy drop on all or part of the data to possible introduction of bias. Many works in the state-of-the-art addressed the heterogeneity and its complications in predictive learning. These works can have different names and definitions for data heterogeneity. Many of them do not delve into the detection and measurement of heterogeneity in data and assume the subgroups that were structured by the underlying heterogeneity are given. They mostly focus on increasing the aggregate accuracy of prediction models and achieving robustness in prediction models' accuracy. These works do not investigate the relationship between heterogeneity and fairness.

To address the knowledge gap in the literature, we used and compared distribution-based clustering models to study the heterogeneity in four real-world datasets. These clusters were used to study the relationship between heterogeneity and fairness. Furthermore, these clusters were utilized to reduce the impact of heterogeneity on the accuracy of prediction models by learning a different one for each cluster.

In this research, we learned the distribution of heterogeneous data using SPNs and MMs, which were exploited to detect heterogeneity and estimate the latent variables in data. After fine-tuning the hyperparameters of these models on validation data, the results revealed that the log-likelihood of data conditioning on the parameters of the MM is much higher than the SPN.

Initially, we hypothesized that the latent variable could be correlated with the sensitive attribute, which means incorporating the latent variables could've potentially resulted in introducing more bias. This hypothesis was proven wrong after evaluating the accuracy and fairness measures of the multiple models approach. The results showed that using ensemble methods with the help of latent variables improves accuracy without decreasing the fairness value.

Several important points can be concluded from the experimental results' observations. Firstly, we are truly dealing with heterogeneous datasets that have been generated from more than one generative process. Secondly, by incorporating log-likelihood to investigate the quality of clustering models, data is more likely to have been generated from the distributions that were learned by MMs than by SPNs.

By knowing that MM is better suited for this task, the latent variables corresponding to the MM clusterings have been extracted. Referring to the results, these latent variables are not correlated with the sensitive attributes, indicating the heterogeneity defined by the underlying distributions in the data does not affect the fairness at all. Lastly, We discovered that by clustering the dataset into homogeneous partitions, we can achieve higher accuracy without deteriorating fairness. This means that distribution-based clustering captures the data heterogeneity well, and it is a good idea to incorporate them to find the homogeneous subpopulations and train separate prediction models on them.

6.2 Future Work

We explored the impact of data heterogeneity on model accuracy and fairness, revealing that incorporating latent variables can improve accuracy and mitigate the adverse effects of performance caused by heterogeneity. However, several aspects warrant further investigation.

Firstly, defining and measuring heterogeneity bias across diverse datasets and sensitive attributes would help quantify data heterogeneity more effectively. Heterogeneity measures would be vital to increase the interpretability and produce applicable explanations for the data.

Secondly, other ways of estimating the latent variable exist, such as Generative Adversarial Networks (GANs). GANs are a machine learning model consisting of two neural networks, the generator and the discriminator, which are pitted against each other in a game-like setup. The generator creates fake data samples, while the discriminator differentiates between real and synthetic data. In addition to GANs, various unsupervised clustering techniques should also be used and compared. Other models beyond LR, RF, and SVM can be used to evaluate the approach.

Thirdly, limited data points for underlying subpopulations can cause an increased risk of overfitting. This issue is prominent, especially in the subpopulations. This is addressed by employing different techniques in data augmentation methods. We can leverage the similarities between subpopulations, so transfer learning must also be explored.

Lastly, heterogeneity can cause severe problems for accuracy, fairness, and other unexplored aspects of the data. Heterogeneity can negatively affect the data's privacy. Data privacy takes different forms, from data anonymization to differential privacy. Not all the previously explained techniques can be easily used in the data privacy realm. When working with data privacy, all analysis done on the data will need to maintain the privacy of the data.

Bibliography

[1] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021.

[2] David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. 2006.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] Colin R Blyth. On simpson's paradox and the sure-thing principle. *Journal of the American Statistical Association*, 67(338):364–366, 1972.

[5] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[6] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[7] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

[8] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[9] Kevin A Clarke. The phantom menace: Omitted variable bias in econometric research. *Conflict management and peace science*, 22(4):341–352, 2005.

[10] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[11] Aaron Dennis and Dan Ventura. Learning the architecture of sum-product networks using clustering on variables. *Advances in Neural Information Processing Systems*, 25, 2012.

[12] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73, 2018.

[13] Chuong B Do and Serafim Batzoglou. What is the expectation maximization algorithm? *Nature biotechnology*, 26(8):897–899, 2008.

[14] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[16] Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

[17] Robert Gens and Pedro Domingos. Discriminative learning of sum-product networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[18] Robert Gens and Domingos Pedro. Learning the structure of sum-product networks. In *International conference on machine learning*, pages 873–880. PMLR, 2013.

[19] Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pages 473–483, 1992.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[21] Parikshit Gopalan, Michael P Kim, Mihir A Singhal, and Shengjia Zhao. Low-degree multicalibration. In *Conference on Learning Theory*, pages 3193–3234. PMLR, 2022.

[22] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[23] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[24] Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

[25] Matthew Hoffman, Francis Bach, and David Blei. Online learning for latent dirichlet allocation. *advances in neural information processing systems*, 23, 2010.

[26] George Hripcsak and Adam S Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American medical informatics association*, 12(3):296–298, 2005.

[27] Jin Huang and Charles X Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310, 2005.

[28] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012.

[29] Anuj Karpatne, Ankush Khandelwal, Shyam Boriah, and Vipin Kumar. Predictive learning in the presence of heterogeneity and limited training data. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 253–261. SIAM, 2014.

[30] Anuj Karpatne and Vipin Kumar. Adaptive heterogeneous ensemble learning using the context of test instances. In *2015 IEEE International Conference on Data Mining*, pages 787–792. IEEE, 2015.

[31] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[32] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

[33] David Lane. *Online statistics education: A multimedia course of study*. Association for the Advancement of Computing in Education (AACE), 2003.

[34] Bruce G Lindsay. Mixture models: theory, geometry, and applications. Ims, 1995.

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

[36] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

[37] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

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

[39] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

[40] Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Re. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, page 151–159, 2020.

[41] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in big data*, 2:13, 2019.

[42] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[43] Robert Peharz, Robert Gens, and Pedro Domingos. Learning selective sum-product networks. In *LTPM workshop*, volume 32. Citeseer, 2014.

[44] Stephen R Pfohl, Haoran Zhang, Yizhe Xu, Agata Foryciarz, Marzyeh Ghassemi, and Nigam H Shah. A comparison of approaches to improve worst-case predictive model performance over patient subpopulations. *Scientific reports*, 12(1):1–13, 2022.

[45] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690. IEEE, 2011.

[46] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[47] Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[48] Guy N Rothblum and Gal Yona. Multi-group agnostic pac learnability. In *International Conference on Machine Learning*, pages 9107–9115. PMLR, 2021.

[49] Kenneth J Rothman, Sander Greenland, Timothy L Lash, et al. *Modern epidemiology*, volume 3. Wolters Kluwer Health/Lippincott Williams & Wilkins Philadelphia, 2008.

[50] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[51] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[52] David L Sackett. Bias in analytic research. In *The case-control study consensus and controversy*, pages 51–63. Elsevier, 1979.

[53] Raquel Sanchez-Cauce, Iago París, and Francisco Javier Díez. Sum-product networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3821–3839, 2021.

[54] Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

[55] Harini Suresh and John V Guttag. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*, 2(8), 2019.

[56] Christopher J Tosh and Daniel Hsu. Simple and near-optimal algorithms for hidden stratification and multi-group learning. In *International Conference on Machine Learning*, pages 21633–21657. PMLR, 2022.

[57] Rebecca M Turner, David J Spiegelhalter, Gordon CS Smith, and Simon G Thompson. Bias modelling in evidence synthesis. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 172(1):21–47, 2009.

[58] Xiaowei Xu, Martin Ester, H-P Kriegel, and Jörg Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *Proceedings 14th International Conference on Data Engineering*, pages 324–331. IEEE, 1998.

[59] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Learning fair classifiers. *arXiv preprint arXiv:1507.05259*, 1(2), 2015.

Curriculum Vitae

Name:	Ghazaleh Noroozi
Post-Secondary Education and Degrees:	Amirkabir University of Technology (Tehran Polytechnic) Tehran, Iran 2017 - 2021 Bs.c
	University of Western Ontario London, ON 2021 - 2023 Ms.c.
Related Work Experience:	Teaching Assistant Amirkabir University of Technology (Tehran Polytechnic) 2019 - 2021 The University of Western Ontario 2021 - 2023