UNIVERSITY OF CALIFORNIA SAN DIEGO

Active Learning and Epistemic Defenses of Fairness

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

in

Computer Science

by

Praveen Nair

Committee in charge:

Professor David Danks, Chair
Professor Julian McAuley, Co-Chair
Professor Yoav Freund

2024

The thesis of Praveen Nair is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to state my gratitude to my advisor, Professor David Danks, for conceptualizing this project, technical assistance throughout, and the time he's taken to work with me over the last year, even during periods of weeks at a time where I had very little idea of what was going on. His guidance through the field of ML fairness, beginning when I was an undergraduate, has been invaluable, and I am aware of how little one can take that sort of support from a professor and advisor for granted.

I'd also like to thank my friends both at and outside of UC San Diego, who have heard many haphazard explanations of this project of wildly varying quality. Finally, I am appreciative of my parents, Ravi and Chitra, and my sister Pooja, who have supported me throughout my academic journey, and whose confidence in me is the basis of any confidence I have in myself.

VITA

2018–2022	Bachelor of Science, Halıcıoğlu Data Science Institute University of California San Diego
2022–2024	Teaching Assistant, Halıcıoğlu Data Science Institute University of California San Diego
2024	Master of Science, Computer Science University of California San Diego

ABSTRACT OF THE THESIS

Active Learning and Epistemic Defenses of Fairness

by

Praveen Nair

Master of Science in Computer Science

University of California San Diego, 2024

Professor David Danks, Chair
Professor Julian McAuley, Co-Chair

In many high-stakes machine learning problems, outcomes for a given input are only observed if a certain decision is made. For example, in loan prediction, an applicant's loan repayment is only observed if the loan is provided. In these cases, the resulting missing data can lead to uncertainty in models trained on that data, and even with decisions that are globally optimal across both groups, the model can have differences in uncertainty between groups. In this paper, we use active learning with mutual information, or infomax learning, to establish that it could be more informative for a model to select from groups with more missing values. This establishes an epistemic argument, rather than a moral one, for intervening by sampling

more from one group than another, indicating new opportunities and questions for fair, accurate prediction over time in these settings.

Chapter 1

Introduction

In many high-stakes contexts for algorithmic decisionmaking, our ability to collect a high-quality training dataset is negatively impacted by a lack of counterfactual data points, based on decisions made in the past. For example, in the setting of loan repayment prediction, if an applicant is denied a loan, we do not get any information about whether or not they would have repaid if they had received the loan; in this way, we never learn whether this person was a true or false negative prediction. The same is true with recidivism prediction, as we only learn whether a person recidivates if they are released from prison. In healthcare settings, where algorithms might be used to decide whether to provide a certain treatment for a patient, we might see a different informational profile. In these situations, if a patient receives a treatment, we do not know whether or not they would have recovered without it; if they do not receive a treatment, we do not know whether their condition would have improved with the treatment. Previous work aims to rectify this imbalance in training data in multiple ways, including methods for handling missing data, modifying datasets and outputs with respect to fairness constraints, and using Bayesian methods to quantify uncertainty.

In this paper, we draw from two machine learning paradigms that aim to improve our performance and understanding of machine learning models, active learning and causal modeling. Active learning approaches improve model performance over time in situations where the next point to be observed can be chosen. In the long term, incorporating more points into the dataset in

areas of the input space that the model is uncertain about will lead to greater model performance than if new data points are observed randomly. In this work, we use infomax active learning, which selects points from the dataset which maximize mutual information between a point's label y and the model parameters θ given inputs X and the dataset. In order to capture our uncertainty about predictions, and to provide a tractable framework for inference in the presence of latent variables, we use Bayesian structural equation modeling (BSEM) as our model framework, which describes linear-Gaussian relationships between features, decisions, outcomes, and latent variables. Most importantly, using a Bayesian model means that we can easily estimate our level of uncertainty about the parameters of a model in order to conduct infomax active learning.

Crucially, these issues around uncertainty in models with missing outcomes have effects when it comes to group fairness in algorithmic decisionmaking. A rapidly growing literature in machine learning fairness aims to align algorithms, and the decisionmaking structures surrounding them, with philosophical notions of fairness, justice, and equality, particularly in high-stakes areas where people are directly impacted by algorithmic decisions. This field has raised many valid arguments about ethical and consequentialist implications of unfairness, and made a case for intervention to rectify group imbalances through various methods. However, in this work, we aim to establish that such interventions could be justified not solely under moral grounds, but also as a matter of epistemic best practice. If we have a situation in which multiple subgroups of the population have different underlying relationships between observed features and outcomes, and in which decisions yield asymmetric amounts of information, we can make an epistemic case for accepting applicants from groups with fewer historical acceptances purely from an active learning standpoint. In doing so, we provide a reason for even decisionmakers whose only incentive is creating a better model, and who have no vested interest in fairness or equality, to have to consider how their models, both historically and in the present, affect groups differently.

In Chapter 2, we describe previous work in causal modeling, active learning, and fairness with relevance to this paper. In Chapter 3, we introduce our methodology, including our framework for specifying and estimating a causal model with missing data, and calculating

mutual information for that model. Chapter 4 describes our results, based on simulations with synthetic data, and Chapter 5 provides a discussion of these results, and of the implications of some of the questions this work raises.

Chapter 2

Previous Work

2.1 Active Learning & Causal Modeling

2.1.1 Active Learning and Mutual Information

Active learning is a subfield of machine learning that deals with settings in which a decisionmaker, or model, has the ability to choose inputs to get labels for, with the intent of choosing inputs whose labels, when incorporated into the training data, will improve the model as much as possible [1]. In theory and in practice, allowing the model the freedom to select points it is uncertain about leads to more sample-efficient improvements in task performance when compared to feeding the model training samples in a random fashion [2, 3, 4]. However, active learning often requires much higher computational costs and can sometimes underperform random sampling, depending on the heuristic used to select the most informative point [5].

In general, heuristics used in active learning methods seek to find areas of the input space for which the model is uncertain. The method that we use for this paper, which tackles this in an information-theoretic way, is infomax active learning, which uses the mutual information between labels and model parameters as a measure of how informative a point will be. Mutual information was defined as an entropy measure in Claude Shannon's seminal "A Mathematical Theory of Communication" [6], but was not named as such or used as a metric of the relationship between two variables until later [7, 8]. Formally, the mutual information between two variables X and Y is defined (from [9]) as the difference between the entropy of the distribution of X, and

the entropy of the distribution of X conditional on Y:

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{2.1}$$

Note that this measure is symmetric between X and Y. Intuitively, this value captures the degree to which knowledge of the variable Y reduces the entropy of our beliefs over X. Mutual information has its lowest value of 0 when knowledge of Y does not change the entropy of the distribution over X.

In infomax learning, X and Y are generally replaced with the outcomes of a model Y and the parameters of the model θ, with both distributions conditional on an input X and the rest of the dataset that has been observed, $\mathscr{D}$. So, higher values of mutual information mean that knowledge of a particular label, or outcome, Y for input X provides more information about (i.e. reduces the entropy over) the distribution of possible model parameters θ. One optimization of the mutual information calculation is given in [10], which rearranges the previously used equation for mutual information such that entropy is calculated in outcome space, which is generally much lower in dimensionality; the objective for active learning with this formulation is called Bayesian Active Learning by Disagreement (BALD). In addition to allowing sampling from the posterior without estimating the entire posterior distribution, BALD provides the simple framing that active learning with this objective selects the input X for which there is the most disagreement about the outcomes in the posterior distribution over θ. This framework is extended in [11], from which we ultimately derive our calculation for mutual information in our model, starting at Equation 3.1. Mutual information and infomax learning have been widely used in active learning research, in feature preprocessing [12], as a sample-efficient method for continual learning in a neural network [13], as a method for learning representations over time in deep reinforcement learning algorithms [14], and as a data-efficient method for estimating neuron responses from stimuli [15].

While active learning is possible with a number of machine learning models, we choose

to use a causal modeling framework in this paper, because it allows us to both parameterize assumptions and ask questions about our problem setting that traditional machine learning methods couldn't. At their most basic, machine learning methods only learn associations between features and outcomes, and in general, do not particularly care if those represent real-world causal relationships. But in many settings, we want to know those real-world causal relationships, especially if we have the ability to intervene on them. Causal learning methods model features and outcomes as part of a causally-linked graph that, ideally, represents patterns of causation between variables. In addition to graphs, causal models can also be represented by a set of equations that formulate effects as the function of their causes.

As described in Pearl (2010) [16], causal modeling, and more specifically structural equation modeling, provides a formal mechanism for asking questions of a counterfactual form, for example, "Y would be y had X been x in situation $U = u$" [16, 17]. This is useful for the interpretation of decisions; for example, causal modeling allows us not only to say how a feature could vary to change the model's prediction, but also to outline the causal pathway between a feature and a prediction. Causal modeling also neatly describes relationships of conditional independence, dependence, and confounding that occur in real data. Most importantly, causal modeling allows for estimation of unobserved, or latent, factors that have impacts on observed values. If one can specify a hypothesized causal model, as a graph or path diagram, of how latent variables relate to observed ones, this makes it possible to capture constructs that are either unobserved in a particular dataset or impossible to observe as part of our analysis of a dataset. These features of causal modeling will be useful in our setting by allowing us to define latent features like "creditworthiness" that we can relate to features and outcomes, and allow us to define our uncertainty about specific parts of the causal model.

In particular, in this paper, we use structural equation modeling, or SEM, as our causal modeling framework. Structural equation modeling is a general causal framework for understanding relationships between variables, and consists of a "measurement model" which relates latent variables to observed variables, and a "structural model," which relates latent variables to one

another [18]. In the SEM we will set up, effects are described as a linear equation of their inputs, plus a Gaussian error term. A more in-depth analysis of SEM parameters and their estimation in such a model is provided as part of Modeling Framework A in [19], and the specifics of how our model is learned from our data, including the handling of missing values using maximum likelihood methods based on the covariance matrix of the data, are provided in Section 3.1.

Active learning on causal models has been conducted with multiple strategies. These methods can operate either at the level of causal graphs themselves, or with respect to a given sampled dataset from the true causal graph. Eberhardt et al. (2005) [20] formalize types of experiments, varying multiple nodes at a time, that can be conducted for learning more about a causal graph, and derive an upper bound of $\log_2(N) + 1$ experiments required to completely determine a causal graph with N variables. However, due to the nature of these experiments, these results do not apply when latent variables or selection bias are present. He and Geng (2008) [21] chose intervention experiments in chain graphs that required the fewest variables to manipulate such that directions of causal links could be determined for all graphs within a Markov equivalence class. Hauser and Bühlmann (2014) [22] extended this framework with optimization frameworks for single- or multiple-target interventions that either maximized the number of edges that could be oriented (rather than being undirected), or minimized the largest clique of undirected edges in the hypothesized graph, and evaluated this in both the ”oracle” case assuming infinite samples, and with finite samples from observational data.

Other works have conducted active learning on causal models with Bayesian parameter estimation, mostly in the last few years. Zhang et al. (2023) [23] learn a Bayesian causal model, then design an acquisition function over possible interventions such that minimizing this function will yield the intervention that will maximally reduce the variance of the belief distribution over the parameters. Toth et al. (2022) [24], similarly to our paper, use mutual information to perform active learning on a Bayesian structural causal model, but unlike this paper, rather than directly optimizing the likelihood, estimate causal additive noise models using Gaussian processes in order to compute likelihood. This provides both computational optimization and allows more

general causal models to be used. Our paper most directly draws on Jha et al. (2024) [11], which performs infomax active learning on discrete latent variable models; while we focus on a simpler causal model, [11] provides a starting point for calculating the mutual information using MCMC parameters, which we adapt to our model beginning with Equation 3.1.

2.2 Inference with Missing Outcomes

Many machine learning problems that deal with prediction suffer from not being able to observe counterfactual outcomes – what would have occurred had another choice been made, or another possibility have transpired. In this paper, we deal with outcomes that are missing due to past decisions. For example, in a loan repayment prediction setting, if a loan was not granted previously to an applicant, we are never able to observe if they were actually creditworthy, so the amount of information we're able to learn from them is limited. This sort of missingness structure takes place in a number of other high-stakes contexts for machine learning, such as the prediction of recidivism, the prediction of healthcare outcomes, or child welfare screening [25]. Missingness can also occur for reasons that are non-random, but not explicitly intentional, such as participants dropping out of clinical or research trials [26, 27].

A great deal of previous work exists on handling missing data. Under the taxonomy created by Rubin (1976) [28], these settings we discuss have missingness that can be classified both as missing at random (MAR), because missingness of the outcome is related to observed features, and not missing at random (NMAR), because if the decision that leads to missing values is even slightly informative, the missingness of the outcome is also correlated with the value of the outcome itself. For example, in the loan prediction case, we might be less likely to observe outcomes for applicants who would not have actually repaid a loan, because these are also the applicants who should be less likely to receive a loan in the first place.

In practice, missing data can be handled in a number of ways in a causal model. The most basic is listwise deletion, where any observations with any missing values are deleted in

their entirety. However, estimating a model after listwise deletion assumes that, per Rubin's [28] taxonomy, the data are missing completely at random (MCAR), which is very rarely the case, and certainly not the case for our setting [29]. In a loan prediction setting where only accepted applicants have outcomes observed, for example, this would mean only training a model of loan repayment on the applicants who received loans, which would do little to tell us about creditworthiness for the people who were unlikely to receive loans under the historical decisionmaking procedure. Pairwise deletion, meanwhile, estimates covariance parameters with only available data, ignoring missing values of the relevant pair of variables, therefore removing less information than listwise deletion. But this means that covariance parameters are estimated on different samples of the data (meaning it also assumes MCAR missingness), and this method will not adjust variance due to missing data points [29]. Another common approach, also used in non-causal modeling methods, is imputation, either simply with the mean of the available values of a column, or multiple imputation regressing the missing value on available data. Finally, the method that we use is full information maximum likelihood (FIML) [30], which adapts standard maximum-likelihood methods for estimating SEMs to use all available, non-missing data, and requires fewer assumptions about the missing data patterns than aforementioned methods [29]. In multiple analyses, FIML outperformed or performed similarly to other methods for handling missingness, such as listwise deletion, pairwise deletion, simple mean imputation, and multiple imputation [31, 32, 33], although the differences between FIML and multiple imputation have been noted to be modest, and could depend on the metrics used to compare the complete and estimated data [32].

It is also possible to include missingness patterns as variables themselves in a structural equation model. Muthen et al. (1987) [34] does so with "latent selector variables" s_{ij} that are learned based on whether given values j for observations i are missing. In this paper, we will not need to do so, since missingness in the cases that we're studying does not have a direct causal impact on outcomes (the assumption being, for example, that being rejected for a loan does not inherently make one less likely to repay a loan, all else equal), and these indicator variables are

only necessary if we want to estimate a impact that missingness of a given variable has on a different variable's numerical value.

However, unlike many of the causes of missingness for which the above methods are designed, as mentioned above, the missingness we concern ourselves with in this paper occurs deterministically based on previous predictions made by the decisionmaking agent. Relatively little work exists for performing inference in these settings. In credit scoring models, [35] found that the stricter a decision threshold was, the more a model trained only on accepted applicants overperformed a model trained on all applicants. This indicates that not only does such a model reflect the entire dataset poorly, but it also reports misleadingly high performance compared to its actual performance on the intended task of classifying all applicants. Within credit scoring, methods collectively known as "reject inference" are used to learn from both accepted and rejected applicants. These include re-weighting scores trained for accepted applicants to consider similar rejected applicants, and extrapolating possible values for the missing probability of default for rejected applicants with a regression model [36]. However, in practice, the use cases for these methods are limited, and their effects on performance range in effectiveness, performing particularly poorly in situations with a low rejection rate [36].

2.3 Machine Learning Fairness

In the last decade, a rapidly growing field of study has examined the ability of algorithmic decisionmaking processes to be fair, unbiased, and equitable [37, 38], with an increasing number of definitions of fairness [39], and in a growing number of domains of machine learning [40]. Fairness definitions and analyses are most commonly covered for binary classification tasks, considering fairness with respect to a defined set of protected, often demographic groups, such as race or gender. Definitions of fairness most commonly use statistical measures such as balancing false positive rate, predictive parity, or equalized odds [39].

However, other work has extended the field to include other conceptions of fairness

than between-group equity, and to include other, more complex definitions of fairness. These include individual fairness [41], which operates on the principle that similar people (according to a context-dependent measure of similarity) should receive similar classifications. This relates closely with the framing used in latent variable modeling in multiple ways. Latent variable modeling posits that there are unobserved, latent constructs that are nevertheless important in understanding and predicting outcomes. For example, for a loan prediction problem, one latent construct of value might be ”creditworthiness” – we believe that two people who are equally creditworthy should be as likely as each other to receive loans. Of course, this shifts the problem from a fairness issue to one of defining who is and is not creditworthy, but this is exactly the sort of analysis that causal modeling with latent variables is designed to handle, as long as we provide the model with simple assumptions of causality. (For example, we can specify the link between latent values and the outcome variable in our model, which we will do in this paper, to provide meaning to the latent as a construct.)

Multiple analyses have combined the individual fairness approach with causal modeling, especially with respect to counterfactuals – in this case, counterfactuals refer to what would have occurred if the protected group was different, rather than if the prediction was changed [42, 43, 44]. In particular, Madras et al. (2019) [45] conceived of protected group membership as a confounding variable between treatments and outcomes, unlike previous causal fairness approaches which generally held the group membership as just one of the features; the authors also note that this causal framework has implications for dealing with missing data through methods like imputation. A follow-up work [46] considers fairness through causal modeling in dynamical systems, where models are updated over time and the effects of predictions on new input data need to be considered. That paper considers the fact that non-random missingness is present in the outcome, and handles this by using a doubly robust estimator derived in [47] to improve performance. There are also multiple fairness metrics that use causality as a framework for measuring fairness and discrimination, as once a causal model is learned, it is simple to query it about how different adjustments to protected characteristics could affect outcomes [48].

Machine learning fairness research has also overlapped with active learning, as some approaches seek to use active learning as a method to remedy imbalances in information between groups present in datasets. These methods include selecting features that maximally reduce classification disparities within a budget for exploration [49], using Bayesian Active Learning by Disagreement (BALD) to sample in order to balance datasets between groups [50], and other methods [51, 52]. These methods primarily use machine learning to collect more information from groups with fewer data points available in the current training set; however, this is still only considering cases with fully available data, where disparities in information are only due to the relative size of the subgroups, and our paper will have to conduct this active learning without access to outcome labels in many cases. For work on fairness with missing values, [53] considered the impact of missingness on algorithms for fair ML, and developed a algorithm for multi-stage decisionmaking where values at given stages could be missing.

Several methods exist to adjust the decisionmaking pipeline in order to consider fairness prominently. These methods operate throughout the lifecycle of an ML model, from preprocessing methods like ”massaging,”, reweighing, or other adjustments in a dataset to produce classifiers independent of sensitive attributes [54, 55], in-processing techniques such as formulating fairness as a constraint or penalty term [56, 57], to post-processing model outputs to meet different definitions of group or individual fairness [58, 59, 60]. Primarily, these fairness methods adjust the predictions from a model with a purely accuracy-based loss function, and in many cases, the addition of constraints or adjustment of procedure to consider fairness incurs a nontrivial cost in performance. These costs are often justified with reference to avoiding disparate impact among protected groups [56], or more generally, with regards to what applicants deserve from a decisionmaking procedure. While, as discussed above, fairness research has considered other framings for thinking about group imbalances in prediction, we will aim to make a direct case for such interventions without requiring an appeal to either disparate impact or to what applicants deserve, not because these aren’t important considerations, but because they aren’t incentives for many stakeholders with interest in and power over algorithmic decisionmaking.

While the majority of work on fairness seeks to implement some statistical measure of fairness under fixed datasets and a particular learning problem, some higher-level work has considered that the fairness-relevant choices made in the process of developing a machine learning system are not isolated to the performance of the algorithm at that system's core. Many papers have considered that in the process of creating an ML system, outside of just the model used, practitioners make a wide variety of morally relevant choices in their design of decisionmaking procedures [61, 62, 63]. These include how training data are chosen, how models trained on one context can be transferred to use in another context, how we choose to encode concepts like outcomes, accuracy, fairness, and protected group membership, and ultimately, how the a model's predictions are translated into concrete actions. These questions are not directly addressed in this paper, but it is nonetheless important to note, when examining the sorts of high-stakes machine learning contexts that we care about, that fairness in machine learning involves more than just training the right prediction algorithm.

Chapter 3

Methods

In order to investigate the performance of active learning methods for situations with missing outcomes, we set up a two-step procedure for learning causal models from data, then apply active learning to yield the point that is most informative about model parameters. The motivation behind this procedure is two-fold. The first, in the generation and estimation of the SEM, is to establish that learning a single decisionmaking model, even if it is globally optimal for the union of two subgroups, can lead to imbalances in the model's ability to predict for both groups if the groups' true parameters differ. Of course, these imbalances are widely studied in the field of machine learning fairness. But in our setting, we don't just care about imbalances in prediction, but due to the pattern of missingness we've described, we also care about imbalances in information we can learn about each group. In other words, we care not only that the best global model can make more mistakes for one group than another, but also that these mistakes could lead to missing data in future training sets, such that we are much more uncertain about true parameters for one group than another.

The second motivation is to establish that there could be epistemic cause for intervening in favor of points from an underrepresented group. While literature on machine learning fairness and fairness in decisionmaking differs wildly on what the goals of the field and what it means to be fair [64, 39], the case for why fairness matters, and why it might be worthwhile to intervene to enforce it, is normally justified on ethical grounds. However, in this paper, we aim to show

that even if a decisionmaker were not to care about the moral issues with treating multiple groups within datasets differently, they might still have cause to care about this differential treatment, because in the long run it could lead to a worse model. This is important, because in many of the high-stakes decision problems that we attempt to model, the primary objective of the decisionmaker is often agnostic to fairness or impacts for individuals. For example, in loan repayment prediction, a typical bank's primary goal in lending, at least in the absence of regulation, is to profit from loans, not to serve the interests of applicants. But in situations where a negative prediction also means that a point will not be observed, we hypothesize, the points that would most benefit the model come from the group most disproportionately harmed by the decisionmaking procedure.

In order to achieve these goals, we design a structural equation model (SEM) that represents the prediction settings we are interested in, and formulate a definition for mutual information (MI) for that model. We then use an optimization procedure to find the points for which mutual information is highest, perform this procedure for a selected group of datasets, and analyze the results for implications for prediction and fairness.

3.1 Structural Equation Modeling

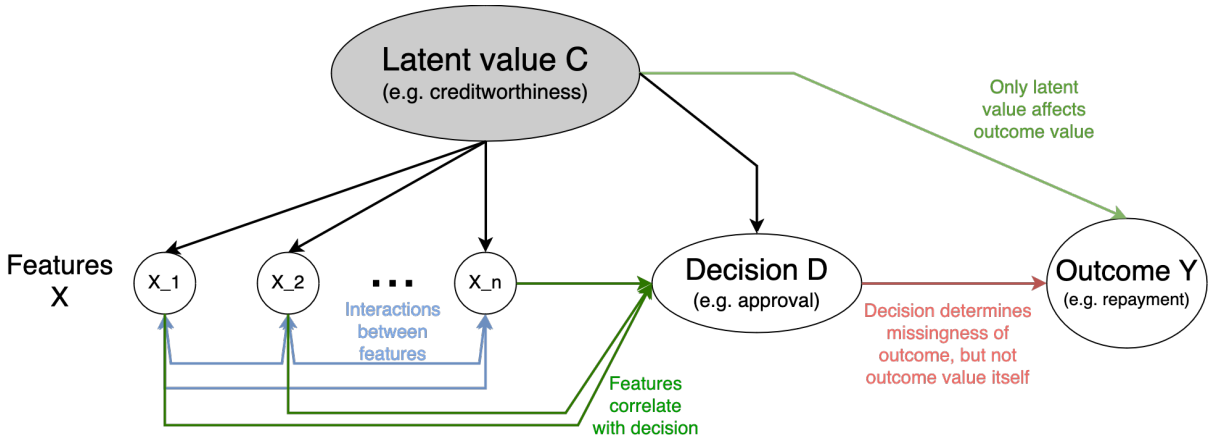

Figure 3.1. The SEM path diagram we use for estimation throughout this paper.

In this paper we use the SEM path diagram pictured in Figure 3.1, in which a single latent C leads to a vector of observed features X, and also is the sole variable that affects the value of the outcome Y. Because the features X might be correlated with each other in ways that are not captured by the latent C, the features X can also interact through residual covariances. We also model the relationship between the features X and a decision variable D. To simulate the presence of missing outcomes, whenever the decision D is negative, we delete the value of Y for that observation. The decision D does not affect the actual value of the outcome Y, although they are likely correlated through the latent and the features, and the only effect of the decision is in controlling the missingness of Y. However, we do estimate the SEM with a link between the latent C and the decision D, to capture the idea that past decisions might have been made with additional knowledge about the latent than is available in the features X, allowing us to better understand the information that a positive or negative decision tells us about the latent. In this way, the latent represents the "true," unobserved construct that controls an observation's likelihood of a positive or negative outcome, which is what the decision aims to capture. It should be noted, however, that while the link between the latent and the decision is used in estimation, it need not be included in the dataset generation, where we learn the decision just from the features, or the calculation of mutual information, where the model parameters for the decision are not required.

Our SEM is specified as follows:

- $C \sim \mathcal{N}(0, \varepsilon_C)$
  - where C is a scalar latent variable that controls the outcome Y, and is distributed according to a normal distribution centered at 0 with variance ε_C
- $X = \alpha + \Gamma C + E_X$
  - where X is a d-dimensional vector of observed features, α is a d-dimensional vector of intercepts for X, Γ is a d-dimensional vector of regression coefficients, C is the

scalar value of the latent, and E_X is the d-dimensional vector of residuals of each dimension in X, which can have different variances and be correlated with each other

- $D = \tau + KX + \phi C + \varepsilon_D$
  - where $D \in \{-1, 1\}$ is the historical decision made for each point, τ is the scalar intercept for D, K is the d-dimensional regression coefficients of D from X, ϕ is the scalar regression coefficient of C on D, and ε_D is the uncorrelated variance of D
- $Y = \nu + \Lambda C + \varepsilon_Y$
  - where $Y \in \{-1, 1\}$ is the true outcome for each point or null if not observed, where ν is the scalar intercept for Y, Λ is the scalar coefficient of C on Y, and ε_Y is the uncorrelated variance of Y

To provide an example of how these variables might work in practice, we'll turn to the setting of credit, where a decisionmaker (e.g. a bank) is deciding whether to provide loans to a particular applicant. In this situation, the latent C represents the true, underlying creditworthiness of the applicant. The relationship between C and the outcome Y can be noisy, but C is the only value which has a causal link to Y. The bank can observe a set of features X about the applicant, which are causally downstream from C, and based on these features, the bank makes a decision D. If this decision D is positive, then the bank is able to add a full feature vector containing the features X, the decision D, and the outcome Y to their training dataset for future predictions. However, if the decision D is negative, then the bank can only add X and D to their data. Of course, this is not useless information – this could help better understand the relationships between features in X, and better calibrate decisionmaking, but it does not provide information about whether this point was a false or true negative.

In our experiments, we treat both the decision variable D and the outcome variable Y as continuous when learning the model, although in practice these are discrete values of either -1 or 1. While there are methods for specifying SEMs with discrete and categorical

values, this increases the number of parameters required to estimate; in our setting, although our SEM will return predictions as continuous values, we can use a threshold value that will allow us to discretize those variables in both our generation of data and our calculation of mutual information.

SEMs can be estimated in a number of ways, but in order to leverage active learning tools, we use Bayesian Structural Equation Modeling, which in addition to learning a single best model, returns a distribution over our beliefs that quantifies our uncertainty about each parameter. We are able to do this with the use of the R [65] package `blavaan` [66, 67], which builds Bayesian estimation onto the earlier package `lavaan` [68], provides a simple syntax for describing causal models, and allows for Markov Chain Monte Carlo (MCMC) draws from the posterior distribution of the model parameters. `blavaan` performs Bayesian inference with the use of the software Stan [69, 70].

More specifically, after the model is specified as described above, the parameters of the model are estimated using maximum likelihood (ML) methods. In the presence of missing data, `blavaan` makes use of full-information maximum likelihood (FIML), a method for learning SEM's in the presence of missing observations, first developed as part of the SPSS Amos software [30]. In the Bayesian structural equation modeling procedure implemented in `blavaan`, the FIML estimation method is combined with sampling over the latent variables in order to produce a set of parameter draws from the model, with the mean values of each parameter providing the maximum-likelihood model, and the distribution of draws around that mean quantifying uncertainty in the model [67].

In our implementation of the SEM model, we specify the model with the syntax provided by `lavaan`, with the additional constraint that the latent variable C has a standard normal distribution $\mathcal{N}(0,1)$. Then, we run 4 MCMC chains with different starting points, running in parallel, which each draw 2000 burn-in samples, then draw and save 1000 more MCMC samples which will form our actual model later. However, we first check whether the chains have converged by assessing the $\hat{R}$ measure of the model, which checks whether chains have

mixed by comparing the variance of the parameter values between chains to the variance within each chain [71]. As recommended in the Stan documentation [70], we only continue to use the MCMC samples when our 4 chains yield an $\hat{R}$ below 1.05. If this is not the case, we re-run our model estimation with a larger burn-in period, increasing by 1000 samples each time our model has not converged enough. We then export our parameter draws to a CSV file, where they are then read in by a Python script that performs mutual information calculations and optimization.

In our active learning procedure, we aim to find the point X for which getting a label y would be most informative about our model parameters θ. In this paper, as described in Section 2.1.1, we will use mutual information (MI) between the label y and the model parameters θ.

To calculate mutual information between y and θ, or $I(\theta; y|x, \mathscr{D})$, at a given point using MCMC samples, we begin with a formulation given in Jha *et al.* (2024) [11]:

$$I(\theta; y|x, \mathscr{D}) \approx \frac{1}{M} \sum_{j=1}^{M} D_{KL}(P(y|\theta^j, x, \mathscr{D}) \;||\; P(y|x, \mathscr{D})) \tag{3.1}$$

In equation 3.1, D_{KL} refers to the Kullback-Leibler (K-L) divergence, which measures the distance between two probability distributions, θ^j refers to the j-th of M MCMC parameter draws, and $\mathscr{D}$ refers to the information present in the dataset. More specifically, Jha *et al.* define the K-L divergence with respect to model predictions as follows:

$$D_{KL}(P(y|\theta^j, x, \mathscr{D}) \;||\; P(y|x, \mathscr{D})) = \int_{\mathscr{Y}} P(y|\theta^j, x, \mathscr{D}) \log \frac{P(y|\theta^j, x, \mathscr{D})}{P(y|x, \mathscr{D})} \tag{3.2}$$

where $\mathscr{Y}$ refers to the output space of Y. With this equation, as pointed out in [11], the point with the highest mutual information will be the one where the predictions of y given by individual MCMC parameter samples θ^j tend to deviate more from the average prediction over all MCMC samples, capturing the entropy of the Bayesian confidence distribution over the parameters.

Now, all we have to do to adapt this formulation of mutual information to our setting is

to specify an equation for $P(y|\theta,x)$ in our SEM. (In this and all future equations, we will not include the $\mathscr{D}$ used in [11] representing the dataset explicitly, since every quantity we look at is calculated with respect to the dataset.) $P(y|\theta,x)$ is not difficult to specify, but it is difficult to compute, as due to the structure of our SEM it involves marginalizing over C, and therefore involves an integral. In the following equations, we will use the variables for SEM parameters established in Section 3.1.

$$P(y|x,\theta) = \int_C P(y,C|x,\theta) = \int_C P(C|x,\theta)P(y|C,x,\theta) = \int_C P(C|x,\theta)P(y|C,\theta) \tag{3.3}$$

$$P(C|x,\theta) = \frac{P(x|C,\theta)P(C|\theta)}{P(x|\theta)}, \text{ and } P(x|\theta) = \int_C P(x,C|\theta) = \int_C P(C|\theta)P(x|C,\theta) \tag{3.4}$$

$$P(y|x,\theta) = \int_C \frac{P(x|C,\theta)P(C|\theta)}{\int_C P(C|\theta)P(x|C,\theta)} P(y|C,\theta) \tag{3.5}$$

The integral in the denominator will have the same value for every value of C in the outer integral, so we can pull the denominator out.

$$P(y|x,\theta) = \frac{\int_C P(x|C,\theta)P(C|\theta)P(y|C,\theta)}{\int_C P(C|\theta)P(x|C,\theta)} \tag{3.6}$$

While we were not able to simplify either integral solve directly for $P(y|x,\theta)$, we note that we can solve for every probability in the above equation with parameters θ:

- $P(x|C,\theta) = P(\mathscr{N}(\alpha+\Gamma C, E_X) = X)$
- $P(C|\theta)$ is a normal distribution described by its coefficients
- $P(y|C,\theta) = P(\mathscr{N}(\nu+\Lambda C, \varepsilon_Y) = y)$

So, we can numerically estimate both integrals through numerical methods, by sampling from the known distribution of C, calculating the quantities above, and aggregating them to estimate $P(y|x,\theta)$. In our simulations, we do this with 500 samples of C, which in model

estimation is fixed to have a standard normal distribution. Once we do this, we can plug $P(y|x,\theta)$ back into the equation for mutual information from [11], which we simplify as follows:

$$I(\theta;y|x) = \frac{1}{M}\sum_{j=1}^{M}\int_{Y} P(y|\theta^j,x)\log\frac{P(y|\theta^j,x)}{P(y|x)} \tag{3.7}$$

Since Y is a binary, discrete variable for our purposes, despite learning Y as a continuous variable, we can replace the integral over all possible outcomes $\mathscr{Y}$ with a sum over the two possible outcomes, positive and negative. In our data, we denote positive decisions as $y = +1$, and negative decisions as $y = -1$, and so we can simplify Equation 3.7 to:

$$= \frac{1}{M}\sum_{j=1}^{M}\left[\left(P(y\leq 0|\theta^j,x)\log\frac{P(y\leq 0|\theta^j,x)}{P(y\leq 0|x)}\right)+\left(P(y>0|\theta^j,x)\log\frac{P(y>0|\theta^j,x)}{P(y>0|x)}\right)\right] \tag{3.8}$$

By substituting in $y \leq 0$ or $y > 0$ for y into Equation 3.6, we can then compute mutual information using Gaussian PDF's and CDF's with known parameters. To calculate $P(y > 0|x)$, we simply take the mean of $P(y > 0|\theta^j,x)$ over all MCMC samples θ_j for j from 1 to M, the total number of MCMC samples.

Once we are able to calculate mutual information, to perform active learning, we need to find the value of X, X^*, that maximizes $I(\theta;y|x)$. Because we are using numerical methods to estimate mutual information, we cannot directly compute the gradient, making optimization difficult. However, there are a many methods for gradient-free optimization, and we experimented with several methods implemented through `scipy`'s `scipy.optimize` interface [72]. However, due to there being some amount of noise in the mutual information landscape due to the gradient being sampled, we received poor performance from these methods designed for local optimization, since our use case was a global optimization problem. These issues included stopping before reaching maxima, and the returned solution being sensitive to start location. Another issue we encountered was that, since mutual information captured uncertainty in the model, the highest values were often found at values unrealistically far from the rest of the data.

In order to better capture the entire input space, while still making sure we returned a realistic result, we implemented a two-phase optimization procedure. In the first step, we restricted our search space to the convex hull of the feature vectors present in the dataset, using the Qhull software, implemented through `scipy` [72, 73]. After doing this, we sample 100 points from the dataset and evaluate mutual information at these points, in a similar approach to a grid search. We do this in the first step instead of an actual grid search because the data are often distributed in uneven ways, and the convex hull is often quite narrow, depending on the parameters generating the dataset. Sampling points allows us to get more samples in areas with more density of data points. Then, in the second step, we perform a grid search in the neighborhood within 2 units in Euclidean space around the point with highest mutual information. (The neighborhood size 2 was chosen as a lenient value, based on experimentation, for how far the nearest of the 100 sampled points could be from the optimal point.) After performing this grid search, we perform the Nelder-Mead simplex algorithm from the grid search point with highest mutual information, again implemented through `scipy` [72, 74]. The optimal input X returned by this algorithm is then returned.

3.2 Synthetic Data Procedure

In order to evaluate our procedure, and understand its effects with respect to situations with multiple groups with different true parameters, we ran the above procedure on a series of synthetic datasets, each consisting of data generated from two different SEM's, each representing one group.

In each iteration, we generate a dataset for each group from a SEM with parameters we control. In these generated datasets, we fix the weight Λ and the variance ε_Y of the link between the latent C and the outcome Y, so that the value of C has equivalent meaning between datasets. To simplify the procedure, we also fix the covariance matrix of the residuals of our features, E_X, such that the residuals are uncorrelated with each other. However, when this model is learned in

`blavaan`, the only one of these simplifications that is known during estimation is the true value of Λ, which is 1, and the model's parameters for E_X and ε_Y can still vary.

One key assumption we make in our simulations is that the distribution of the latent variable C is the same between both groups. By doing this, we can establish that differences in points being observed are not due to true differences between the groups in their relationship to the outcome variable, but rather due to choices made in modeling and decisionmaking. In each simulation we run, the only two values that we vary are the weights, or loadings, from the latent to the features, Γ, and the relative size of the two datasets. We set Γ to be different for each group, to model a situation where despite two groups having the same underlying distribution of some latent like creditworthiness that is directly related to the outcome, the relationship from that latent to the features could differ between groups. The two datasets are set to have a combined size of 2000, with the lengths of the two groups being one of (1000,1000), (1200,800), (1400,600), and (1600,400). The values of Γ and the relative dataset sizes used are reported in Section 4.

In the following section, the amount of results we are able to report is limited by long runtimes, which are due mostly to the computationally expensive process of calculating mutual information over MCMC samples. When we use Equation 3.7, this requires computing and storing Gaussian PDF and CDF values for each of 500 samples from the latent distribution C, which we do once for each parameterization given by one of 4000 MCMC samples θ^j, 1000 for each of the four MCMC chains we ran. On the computational environment we ran this on, as a containerized application on a Kubernetes cluster with access to 16 processors and 128 GB of RAM, even while parallelizing calculations for the samples from C, calculating mutual information at a given input point X took around 15 to 20 seconds, and therefore, since conducting our optimization procedure required something on the order of 150 to 200 points, this procedure could take up to an hour. Since for each iteration of this algorithm, we learn the model and optimize for mutual information for two different groups, this meant that a single result reported below took multiple hours of calculation.

We generate values from the SEM for each group by sampling from the latent C, since

C is causally prior to any other variables in the SEM, and propagating using the parameter values to generate features X and outcome Y. However, to generate the decision D, instead of setting known parameters, in order to make sure that our results are not an artifact of poor decisionmaking, we design the decision D to be learned from the best possible *global* linear classifier from features X to outcomes Y across both datasets, even including information that will not be available to the SEM at estimation time. This means that the decision boundary for both groups is the same, even though they might have different true parameters between the latent and the features, Γ. So as the disparity between group dataset size increases, the best global classifier will more closely approximate the best within-group classifier for the majority group, and depending on how the two groups are distributed, could begin to perform worse for the minority group. This models situations where, for example, despite applicants from different demographic (or other) groups having different relationships between their observed features and their ability to repay loans, the financial institution making decisions trains only one global model for both groups, which could better serve one group than the other. After generating the two datasets, we then separately learn a SEM using `blavaan` for each dataset, then perform our optimization procedure for mutual information on each one. As a result, we report the optimal points and values for mutual information on both datasets, as well as the variance on the Bayesian belief distribution for each parameter in the SEM.

Chapter 4

Results

In our experiments described in Section 3.2, we generated data from a SEM, described in Section 3.1, whose parameters were fixed except for Γ, the loadings from the latent C to two-dimensional features X, and the sizes of the two groups. When generating data, the latent was fixed to a standard normal distribution, the link between the latent C and outcome Y was fixed to a loading of one with a standard deviation of 0.5, and the residual covariance of the features to $\frac{1}{4}$ times the identity matrix, such that for small values of Γ noise did not dominate the signal from C. However, as noted previously, the only parameters known to the SEM estimation program were the latent was a standard normal, and that the link between C and Y had a loading of one; the remaining parameters were allowed to be learned.

The values of Γ were first chosen to be random integer coefficients between 1 and 10; in other words, each of the four values (two values in each vector Γ_1 and Γ_2) was randomly selected between 1 and 10. We will note that negative coefficients could also be possible, indicating features in X that had negative associations with the latent, and that it would be possible for the vectors Γ_1 and Γ_2 to point in opposite directions, indicating that a feature could be positively related to the latent for one group and negatively related for another group. Still, we restricted ourselves to situations where features have positive, but different, relationships to the features in both groups, as might be realistic for features in the settings we want to model. We chose 40 parameterizations of the SEM, equally balanced between relative dataset sizes, in this

fashion. However, as will be discussed later, the value of mutual information is only sensitive to missingness of the outcome when very few observed outcomes remain; therefore, for random values of Γ, it is very difficult to see the relationship that we discuss. So, in order to better visualize the patterns we are trying to describe, we then ran the same analysis, but restricting coefficients Γ such that (1) the proportion of missingness of outcome Y in one of the datasets, no matter the dataset size, was at least 0.9, and (2) there were still at least 5 observed negative outcomes in the dataset. If missingness was lower than this threshold, then the resulting patterns were difficult to interpret, and if they were higher than this threshold, then the MCMC sampling for estimation of the SEM was likely to fail. Since generating datasets is computationally much cheaper than doing estimation, we generated datasets for all possible combinations of parameters $\{0, 1, \cdots, 10\}^4$, and chose 40 parameterizations at random, with the same number for different dataset sizes. Finally, we ran 32 parameterizations with the same set of conditions, but restricted to only source datasets that were balanced in size, with 1000 points in each group. This was in order to show that the effects described could occur not only because of imbalanced dataset sizes before any missingness was introduced, but purely because two groups had different parameters between features and the latent. Therefore, it should be noted that the following results are not calculated from a random sample of values of Γ in two dimensions, but rather a mixture between random values, and values chosen due to high missingness in one group that helps to demonstrate the effect we're describing.

The following chart shows the relationship between the number of observed values of the outcome Y for a given group in a given dataset, and the value of mutual information at the optimal point reached in the optimization procedure. The value on the x-axis is not the total number of points in the dataset – this is always one of 400, 600, 800, 1000, 1200, 1400, or 1600 – but rather, the number of points for which the true label Y is observed in the dataset. In other words, the number of observed points is the number of points which received a positive decision using the optimal global linear model trained across both groups.

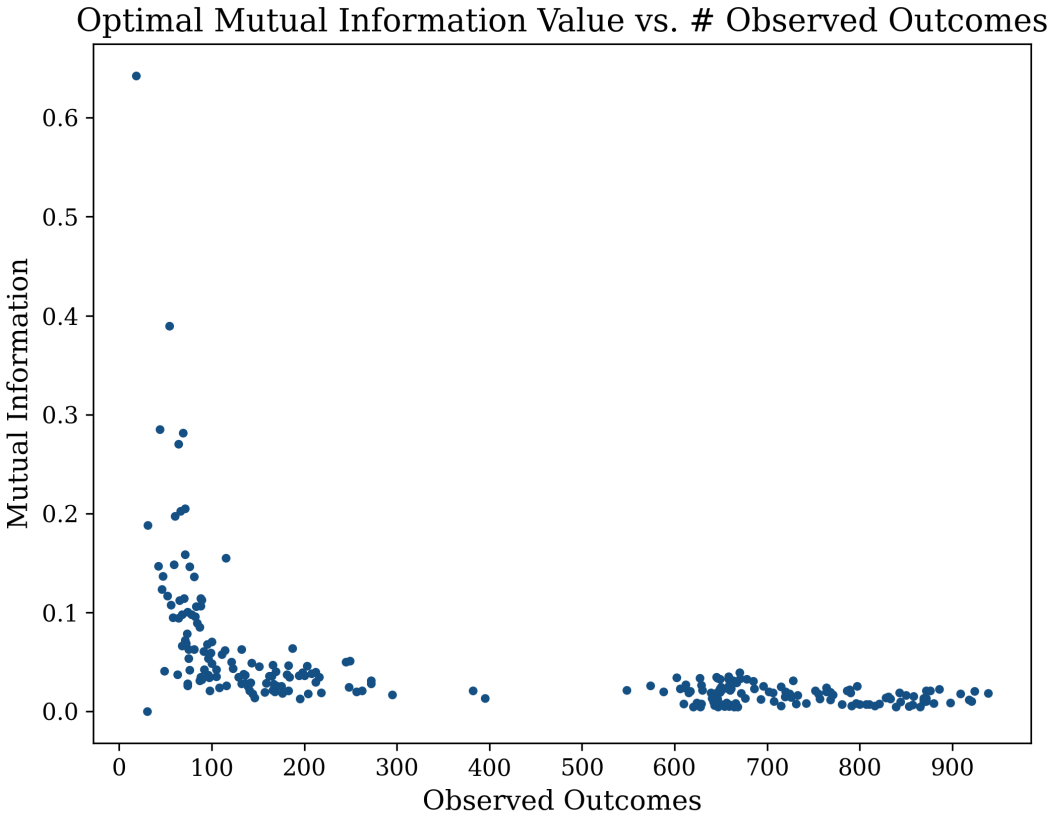

Figure 4.1. The relationship between observed outcomes and optimal mutual information.

In Figure 4.1, the optimal value of mutual information has a clear elbow at around 100 observed points; when the number of observed outcomes gets larger, the value of mutual information at the optimal point remains low, but when fewer than 100 outcomes are observed, we see a strong increase in optimal values of mutual information. This squares with our reasoning about active learning, namely that there is a diminishing return for new information after some amount of samples are observed. Since mutual information captures the amount of disagreement between the MCMC samples of parameters θ, the above result indicates that after around 100 values of the outcome are observed, the Bayesian belief distribution over parameters at even the most unconfident point stabilizes, at least in its degree of entropy. We note that the large gap in the graph for datasets with observed values between 300 and 600 is due to the choices

of parameterization we made above, as the left cloud of points represents the groups with high missingness, and the right cloud represents the counterpart groups with low missingness. (Throughout this section, when we refer to a group's *counterpart*, we mean the group that it was paired with in the simulation, and which shares the same decision boundary that determined which outcomes would be observed.)

In order to show that the relationship in the above graph does not just come about because of the different total dataset sizes, as opposed to the number of observed outcomes, that are in most of our simulations, we report the following Figure 4.2, which is a subset of Figure 4.1 for just parameterizations with both groups having 1000 total points each. So, in these simulations, the only factor that leads to one group having more observed outcomes than another is that the global decision boundary learned across both groups ends up accepting more from one group than another, purely because of the different parameters Γ between the latent and the features. Once again, each point on the left of the graph is the direct counterpart to a point on the right, since very high missingness in one group implied low missingness in the other. As in Figure 4.1, in Figure 4.2, we find that the optimal mutual information value only reaches an inflection point when fewer than 150 outcomes are observed, and greater than this threshold, there are diminishing returns for mutual information as the model gains more outcomes.

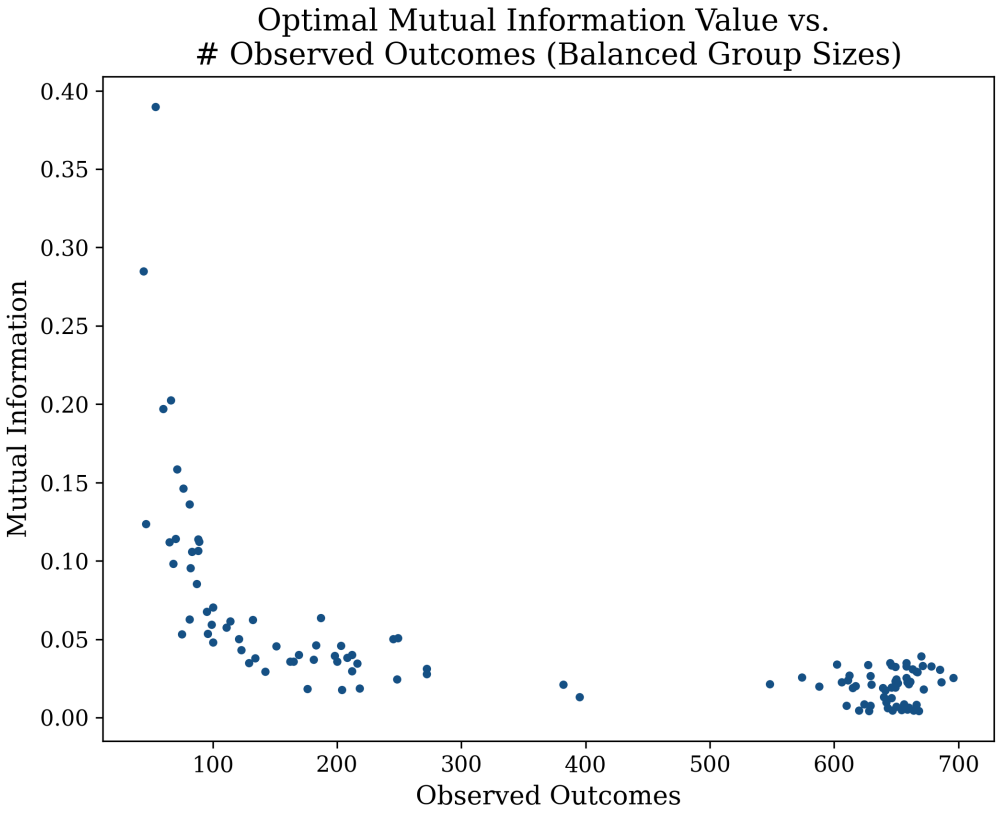

Figure 4.2. The relationship between observed outcomes and optimal mutual information, for simulations with 1000 total observations (including where Y is missing) in each group.

To demonstrate how active learning would work in this situation, Figure 4.3 shows the amount of observed outcomes in each group on the x-axis, and the y-axis represents whether or not the group was selected over its counterpart due to having higher mutual information, with ”True” at the top meaning the group had higher mutual information than its counterpart, and ”False” at the bottom meaning it had lower mutual information. Again, for the same reasons as the results in the prior graphs, we can see that the number of observed outcomes provides a very strong threshold for whether or not a group has higher mutual information; this graph simply adds the context that these are the groups that are being selected in active learning experiments that operate on two groups at a time with the same decision boundary. Figure 4.3 does not display anything about the pairwise relationship between which group was selected, and which

group had fewer observations. In total, however, the group with fewer observed outcomes of the two had a higher optimal value of mutual information 95.76% of the time, across all dataset sizes that we looked at (which as mentioned before, is a nonrandom sample that exaggerates the differences between dataset missingness), and this figure was 94.64% for graphs of balanced total size (1000 total points in each group, though the number of observed outcomes, clearly, differed.)

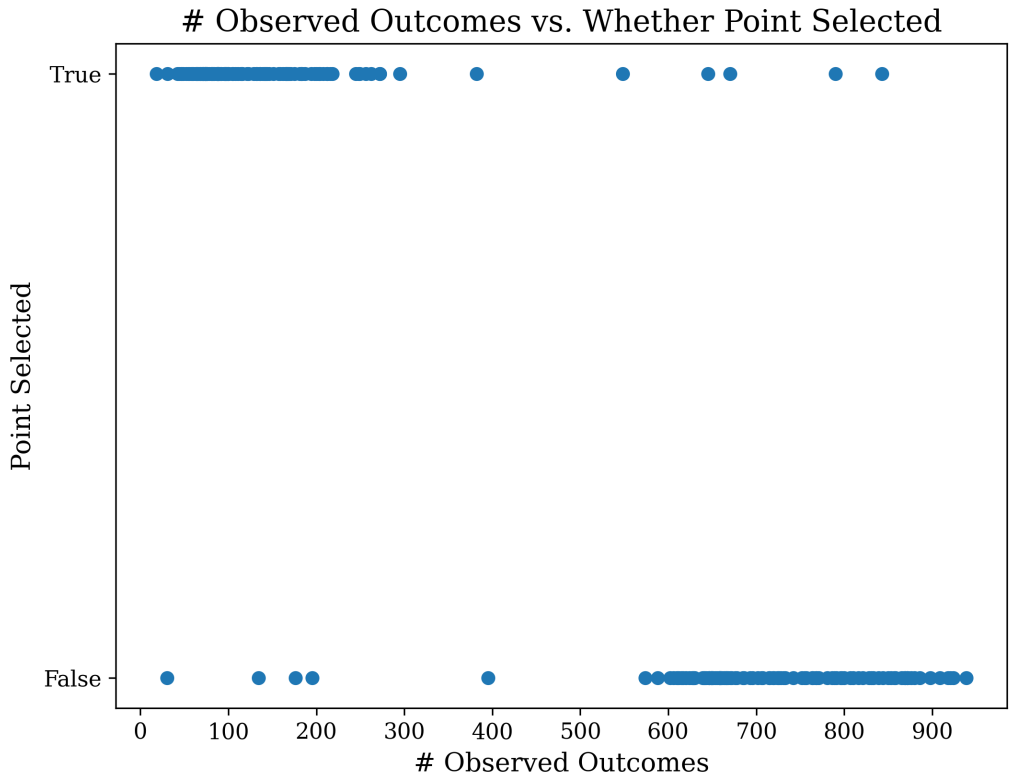

Figure 4.3. The relationship between observed outcomes and whether or not a point was selected over its counterpart.

Another way to frame these results is in terms of the relative performance of classifiers over the two groups. As described in Section 3.2, the decision boundary used to determine which outcomes are present is learned across both groups. Therefore, very high missingness in a group is due to almost all of the points in one group being on one side of the global decision boundary.

This is more likely to happen when one group is much larger than the other, but this also happens when the parameters connecting latent C to features X, Γ, are very different. Since both groups have the same underlying distribution of the latent, this means that, up to noise, both groups should have the same likelihood for positive outcomes; so, a radical difference between the two groups' classifications under the best global classifier represents a failure of using a single global classifier.

With this framing in mind, we can consider one red flag for imbalance in prediction to be a change in false negative rate from the best *local* linear classifier, trained only on the data points within a group, and the best *global* linear classifier, trained on both groups and what we use to generate datasets in our simulations. False negatives here refer to when data points which would yield positive outcomes are predicted as negatives and do not have their outcomes observed; this analogizes to a loan applicant who would repay a loan if given one, but is denied a loan by a financial institution. (The false negative rate is calculated from the generated data before unobserved outcomes are deleted, so this calculation would not be available in the real-world analogue of this procedure.) If the false negative rate increases massively between the best local and best global classifiers, it means that there are many more potential applicants whose loans are unjustifiably being denied; the increase in proportion of false negatives, then, represents those disadvantaged by the choice of using a global model over a local one.

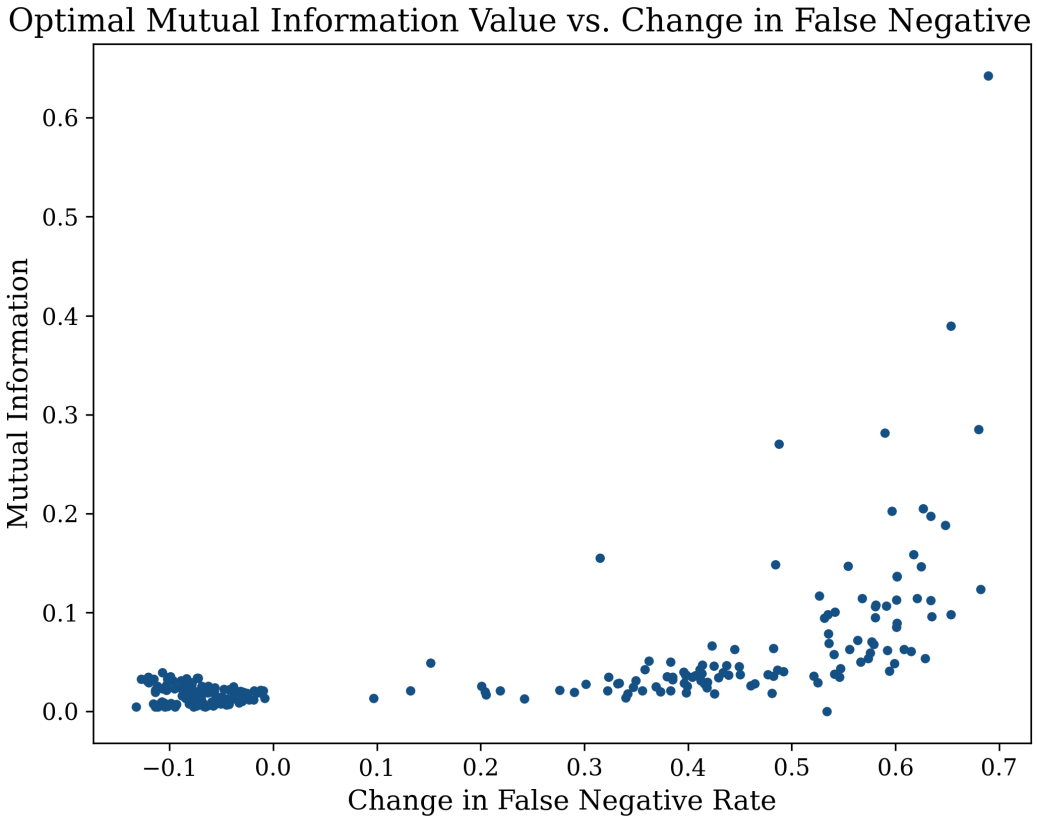

Figure 4.4. The relationship between change in false negative rate for a group and that group's optimal value of mutual information.

Like in previous graphs, due to the choices of parameterizations we used, the values cluster in two directions. The values on the left represent groups where false negative rate went down using a global rather than local classifier. This is because the introduction of the other group dataset skewed the decision boundary such that these points were no longer likely to be classified as negatives. Meanwhile, on the right, we can see groups that had much larger changes in false negative rate due to using a global classifier. Just like with mutual information, we can see a gradual increase until a much steeper curve once the change in false negative rate reaches 0.5. Note that the data points plotted here are the same as the ones plotted in Figure 4.1. This indicates that the groups for which sampling a point from a group are most informative are the groups where the choice to use a global decision boundary leads to much higher false negative

rate, and that there might be value in sampling new data from these points due to the uncertainty that these missing values have produced.

Similarly, we can see in Figure 4.5 the relationship between the accuracy of the learned global classifier and the ultimate optimal value for mutual information that we yield at the end of our optimization procedure. Mutual information increases as accuracy of the model decreases, which comports with the basic underpinnings behind active learning – that we place more value on points (and more broadly, on datasets as a whole) when we are uncertain about their true values, which is more likely with an inaccurate global model responsible for determining whether or not the outcome value is missing.

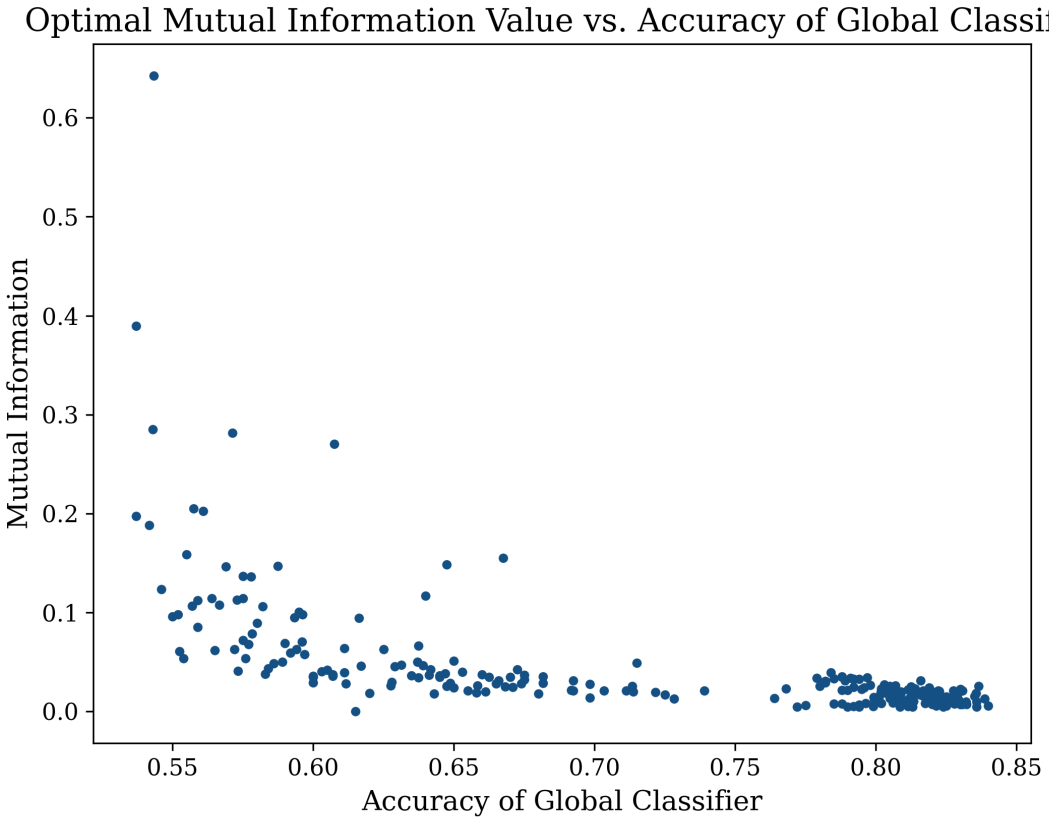

Figure 4.5. The relationship between the accuracy of the best learned global model, learned across data from both groups, and the ultimate optimal value of mutual information.

In addition to learning about what causes values of mutual information to increase, we

might also be interested in where the optimal input point that maximizes mutual information between Y and θ might be. While the actual value of the points varies between different simulations because the distributions of both groups can vary widely, what we can look at is the distance of the optimal point for mutual information from the global decision boundary used to determine missingness in the data. In this case, we're using the optimal point with highest mutual information between the two groups, which would be the point that active learning would suggest we get a label for next. Figure 4.6, below, shows the distribution of points with respect to the global decision boundary, where a positive value $+n$ indicates that the point was n units from the decision boundary and on the predicted-positive side, and a negative value $-n$ means that a point was n units from the decision boundary and on the predicted-negative side.

We find, as expected, that the points that maximize mutual information are distributed near the decision boundary, as these are the points whose outcomes that the model is most uncertain of. However, we find that the distribution of distances to the decision boundary is slightly skewed towards points on the positive side of the decision boundary, indicating that more of the points that mutual information was optimal at would have been predicted as positives by the global model; we might have expected that the opposite would be the case, since the model should have an incentive to select points from regions of the input space with fewer existing data points, and outcomes for observations below the decision boundary are missing, so active learning might be expected to select from this region. However, one effect that we have not accounted for is that the decision boundary that our causal model implicitly estimates is only learned for the subset of outcomes that are observed. Therefore, when we perform active learning, the point for which Y provides the most information about model parameters might be closer to the best decision boundary for the *observed* subset of points, which we have not recorded, but which will be different from the best global decision boundary on all outcomes. Indeed, we might expect the best decision boundary for observed points to be on the predicted-positive side, since that is where all of the points the causal model can see lie. Still, this is only a hypothesis about the behavior of such classification models with missing data. When conducting a one-sample

t-test for whether this distribution has a mean at 0, we get a p-value of 0.047588, indicating that a difference from a mean of 0 could be present. But Figure 4.6 represents only a small, skewed sample of the locations of these optimal points, and therefore we would require further investigation before drawing conclusions from the results, such as examining a larger, random sample of possible values of Γ used to create our datasets.

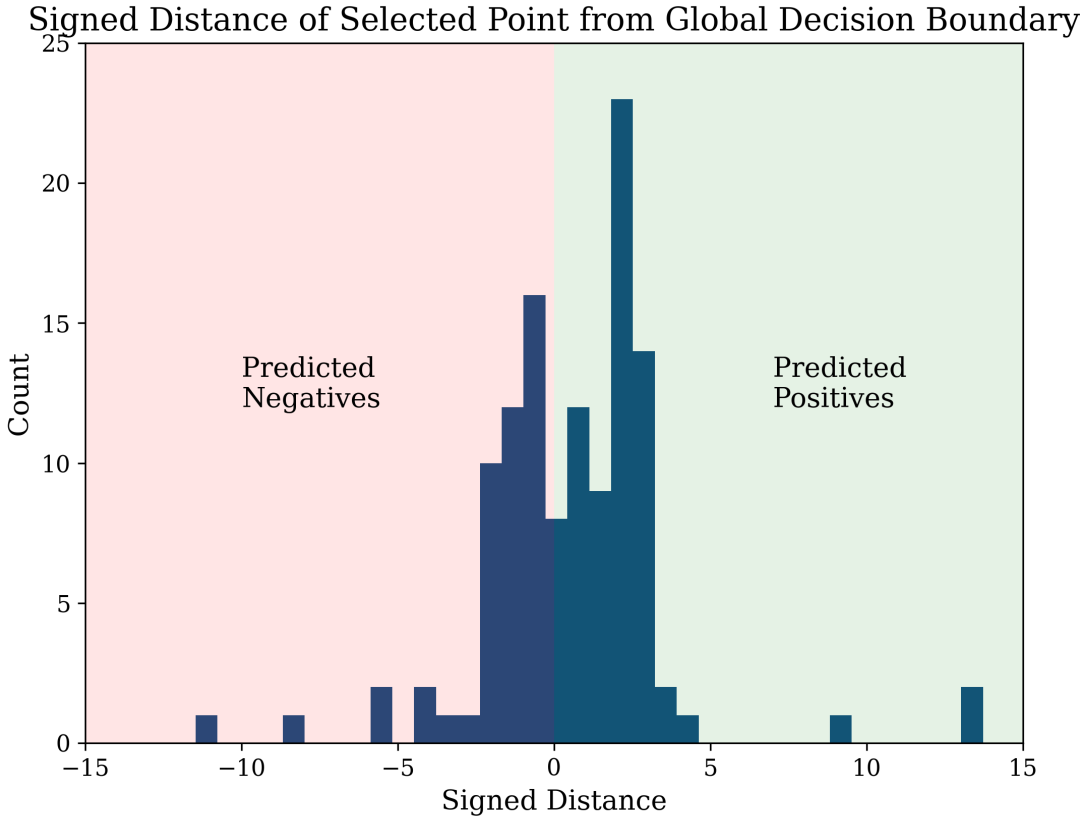

Figure 4.6. The signed distance between the global decision boundary learned over both groups, and the optimal input point with respect to mutual information for each group. Points in the red area would have been predicted negative by the best global linear classifier on the training data without missingness, while points on the right would be predicted positive.

Chapter 5

Discussion

5.1 Epistemic Value

In Chapter 4, we showed that infomax active learning could indicate that, in problems where decisions determine whether or not an outcome is observed, sampling from an underrepresented group would provide more statistical information about model parameters, indicating an epistemic case for sampling from these groups. This case is meant to complement the already well-covered ethical arguments for selecting from underrepresented group. But since the epistemic case is that sampling from underrepresented groups would help the model's performance in the long-term as demonstrated in previous work on active learning, the epistemic case will likely have value to stakeholders that might not be concerned with moral arguments. However, it should be noted that as Figure 4.1 indicates, this epistemic case hits a point of diminishing returns relatively quickly. In the particular way we set up our procedure, the mutual information between Y and θ at the optimal point $X*$, which in active learning quantifies the value of observing the true label at the best possible point, only has a slight decrease after a certain number of observations (around 150 to 200) are observed. This rapid diminishing of returns, of course, is not present with the types of moral cases for sampling points from underrepresented groups discussed in Section 2.3, where the fact that a given model predicts differently for one group than another could be enough cause to adjust our decisionmaking, even if there's no reason to believe that sampling from an underrepresented group will help the model in the long run.

It might not be immediately clear why this sort of epistemic value from a prediction might be important. More precisely, when we refer to epistemic benefits of getting a label from a particular observation, we refer to tightening the distribution of our belief about the parameters of our model, as infomax learning and many other active learning methods seek to accomplish. We believe that doing so will get us closer to the best possible model, which in a causal model is ideally close to the ”true,” unobservable data-generating process that underlies the observed data. Of course, having a more confident version of our model is only useful if we intend our model to be used repeatedly over time, being updated as new points are observed. This is not always the case, but in practice ML algorithms are likely to be updated as new information is available, rather than being rebuilt from scratch for every new data point. The existence of epistemic value for acquiring a certain label means that it might not always be the right thing to do to trust the model entirely, but rather to treat the model as a fallible, uncertain approximation of the real constructs we want to study, which might involve deviating from the model in order to improve the quality of that approximation. Whether that deviation is justified is context-dependent, and presents an explore-exploit tradeoff that mirrors the tradeoffs found in many sequential decision theory problems.

To further ground what we mean by epistemic benefit, we can appeal to the concepts of *aleatoric* versus *epistemic* uncertainty, which have been discussed widely in engineering and machine learning settings [75]. In this framework, aleatoric uncertainty refers to uncertainty inherent in the data-generating process, and therefore in the data itself, and epistemic uncertainty refers to the uncertainty we have due to our lack of knowledge about that data-generating process, due to our finite amount of data and limited mechanisms for modeling it. As noted by [75], aleatoric uncertainty is irreducible, since it exists in the data-generating process, or ”nature” itself, while epistemic uncertainty is reducible, if not completely eliminable, through better data and better models. For example, causal models seek to directly estimate the aleatoric uncertainty in data-generating processes with error terms, even if this is an unrealistic picture of how things really work. Previous work has even applied these concepts to fairness and discrimination [76],

with [77] defining aleatoric and epistemic *discrimination* as bias inherent in the distribution of the data, and discrimination due to choices in modeling and predicting from data, respectively. [77] continues to state that existing fairness interventions perform well at mitigating epistemic discrimination, but not in the presence of additional aleatoric discrimination, in particular the presence of missing values.

In this paper, we aim to demonstrate how having few samples from a given group produces a large amount of epistemic uncertainty, which clouds our ability to make good inferences about data points. Following these definitions further, our setting does involve aleatoric discrimination, not in the distribution of outcomes between groups, which are identical, but in how missingness comes about. Indeed, our setting provides an example of how epistemic uncertainty can lead to further epistemic uncertainty and discrimination in future iterations of a model, in that a combination of uncertainty about a model and having fewer points for it can lead to more false negative decisions for a group; with the missingness pattern we used, this means that a group will have more missing values due to the epistemic uncertainty that we had previously. Finally, we show how active learning, which aims to reduce epistemic uncertainty, can do so by querying labels for groups which have so far been underrepresented in the training data, which does not reduce the underlying aleatoric uncertainty, but does mean that a future model will have marginally less epistemic uncertainty by having access to that label.

5.2 Implications for Fairness

Throughout this paper, we have referred to the behavior of our model and active learning procedure with respect to two generic groups, where the only differences between the two are their relative size and distribution of features given the latent. Indeed, these results might be most interesting for protected groups for which we have strong moral reasons for models not to be biased towards, and most fairness methods aim to balance model performance with respect to these kinds of groups. But the two groups discussed in this paper could be any pair of groups

within a dataset for which the true parameters between the latent and features, or the relative group sizes, vary. For example, the patterns in features associated with creditworthiness for one group might be different than those associated with creditworthiness for another. Depending on the definitions of those groups, we might not necessarily have strong beliefs about enforcing equity between them. But, as we've shown, if those differences between groups exist, then any single decision boundary we learn for both groups to predict Y from X can fail to capture these differences, and either due to differences in group size or just the configuration of parameters, it is possible that the global decision boundary will fail to classify accurately for one or both of the groups.

This is an example of what is referred to in Suresh and Guttag (2021) [78] as *aggregation bias*, where a one-size-fits-all model is used for a dataset with multiple groups that should be considered differently. But while this is a general problem for any machine learning task, in the regime of missingness we study, this is especially troubling, as predicted negatives in our context will have missing outcomes in the training data. False negatives, for example, represent people who should have received a loan, but did not *solely because of* the choice to use a global rather than local model for this group. Therefore, in our setting, the pattern of missingness dependent on decisions compounds aggregation bias, such that not only do our decisions perform differently per group, but our degree of uncertainty about our model also differs per group.

It is important to note that we chose to model the decisions made in the training data as the best possible linear classifiers, even when considering outcome data not available to the model. This is to ensure that the results reported above could not be construed to be due to suboptimal decisionmaking. (While non-linear model classes could possibly provide some performance improvements, since the data-generating process is a linear-Gaussian SEM, these performance gains would likely be modest.) However, in the real world, the decisions that inform missingness in the training data would be markedly suboptimal, as they would not have access to outcomes for any previous negative decisions. Just as importantly, we also do not consider the impact of *unfair* decisions in the training data. Unfairness in historical data is a well-established

problem in fair machine learning, with work both examining its impacts [79] and establishing methods to work around it [54, 55]. Once again, the type of missingness we describe here compounds existing bias in the modeling procedure, as having outcomes missing for negative predictions means that if a group is more likely to, for unfair reasons, have predicted negative decisions in the training data, then the model will also be more uncertain about members of that group. This is important in areas like credit or bail prediction, when there can be a high threshold for positive decisions, and greater uncertainty alone about an individual might be cause for a negative decision, leading to a feedback loop that exists for people in a certain protected group or with a similar set of features. Even if this is not the case, active learning measures could be useful to rectify historical unfairness by observing more points from groups who had unfairly been rejected.

5.3 Limitations

This analysis was subject to some limitations that should be considered when interpreting results. Most prominently, the number of simulations we were able to run was limited by computational resources and by time, since the choices we made in modeling, and especially how we chose to do the optimization procedure, had to be iterated on multiple times before arriving at the procedure reported in Chapter 3. Therefore, the results reported above only represent a modest, non-random sample of all the possible datasets we could have analyzed with this procedure. We only looked at the case where the features X were two-dimensional, and where the features did not have any residual covariance with each other, although our code is designed to handle any number of features and an arbitrary covariance structure. We also made the powerful assumption that both groups would have an identical distribution of the latent, and therefore an equivalent probability of having a positive outcome. While this allows us to narrow down the possible causes of the results we obtained, it is ultimately unrealistic, and to model many of the challenges with doing machine learning fairness in practice, we might need to understand

how active learning and causal modeling might perform if the groups were to have different propensities for positive decisions.

In addition, it should be noted that the results above were found for only one type of causal model (a Bayesian SEM), and for only one active learning measure (mutual information). It is possible that had we made different choices of either of these, the results above would look different; at the very least, some of the exact results we observed might differ slightly, like where the inflection point in Figure 4.1 was, or the central tendency in Figure 4.6. We also have some level of noise in our results due to the fact that we had to use numerical methods to estimate the integral in Equation 3.6, and because we needed to use derivative-free optimization in order to optimize mutual information with respect to the input point. While we took measures to ensure that the number of samples of the integral was high enough to converge in test cases, and that the optimization procedure reached a globally near-minimal point, we cannot independently mathematically verify the results of either of those procedures, which also makes it more difficult to examine the properties of the mutual information function.

5.4 Future Work

This results described in this paper serve mostly to establish an epistemic case for sampling from underrepresented groups in settings where outcomes are only observed for positive decisions, using active learning methods as a well-validated measure for how informative a given point is for the model. However, this is just the surface of the sorts of analysis that could be done for these sorts of problems. As described above, we could extend this analysis to more parameterizations within the same model, which we have already written a framework for doing, or examine other causal modeling and active learning methods. This could also involve freeing some of the assumptions we have made, such as the assumption that both groups have equivalent distributions of the latent, that the features are independent given the latent, and that the decision is an unrealistically performant linear classifier.

In some contexts, our procedure might not be tractable for active learning. For example, one worrying possibility is that the two groups have different amounts of uncertainty due to aggregation bias, but that we do not even know the groups in question. This might occur either due to the groups not being measured in the training data, or because group membership is impossible to measure at all. While in this case, we could not train two within-group models because we lack group information, it might be possible to still perform active learning in such a way that we are directed to sample points from groups with many missing outcomes. This could be done by having both groups be learned through a single causal model, and if the groups are relatively distinct in input space due to different Γ parameters, then optimizing for active learning might still find the same points; the idea being that there will be regions of the input space that have fewer observed values in general between both groups.

While in practice, we wouldn't have a way to know which group we were drawing from, if we were using synthetic data, we could still record which underlying group the active learning process selected from. In addition, we could also use clustering methods to try and estimate which group each point was in, which would allow us to estimate different models by group, but would add an additional layer of uncertainty about cluster accuracy, especially if as is very often the case, the distributions of both groups overlap significantly. We might also consider larger numbers of groups, or groups that each represent the intersection of multiple attributes, as when group membership becomes smaller, our amount of uncertainty about each group, as evidenced by our results, increases dramatically. These types of groups are also important because bias in real-world machine learning has intersectional characteristics, and existing methods for machine learning fairness do not always account for more complex groups [80, 81, 82].

Another possible avenue for future work, of both a computational and social-scientific nature, is how stakeholders could incorporate active learning into real-world machine learning practice while maintaining high performance and without violating legal or moral constraints around predictions. For example, if we are predicting loan repayment, a bank might not care whether an applicant will provide information to our model in the long term if they believe that

providing a loan to this applicant would result in a default, and therefore a large cost to the bank. This is both due to discounting of future impacts, but also because the incentives for the bank are more complex than just the accuracy of the model.

This gets even more complex when we consider other constraints that high-stakes predictions may be held to. For example, might it be justified to provide applicant A, who we currently believe to be less likely to repay a loan but more informative to the model than applicant B, a loan over applicant B for purely epistemic reasons? Are there groupings of applicants for which we might want more information, so much so that we'd bias towards accepting applicants from that group? If so, which ones? Depending on one's beliefs about who deserves allocative benefits from decisions, and what fairness in context looks like, the answer to these might vary. Extending these questions into the sorts of high-importance machine learning problems that we've described is a difficult task, but ultimately necessary to gain a broader understanding of how to build a fair decisionmaking system with an algorithm at its core. This paper aims to establish a framework with which to understand the epistemic benefits of different options in such situations, providing a new lens for thinking about fair algorithmic decisionmaking, but also creating new questions which require more work to settle.

Bibliography

[1] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[2] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[3] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15:201–221, 1994.

[4] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[5] Andrew I Schein and Lyle H Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68:235–265, 2007.

[6] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[7] Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

[8] J Kreer. A question of terminology. *IRE Transactions on Information Theory*, 3(3):208–208, 1957.

[9] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[10] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[11] Aditi Jha, Zoe C Ashwood, and Jonathan W Pillow. Active learning for discrete latent variable models. *Neural Computation*, 36(3):437–474, 2024.

[12] Kari Torkkola and William M Campbell. Mutual information in learning feature transformations. In *ICML*, pages 1015–1022. Citeseer, 2000.

[13] Ziqi Gu, Chunyan Xu, Jian Yang, and Zhen Cui. Few-shot continual infomax learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19224–19233, 2023.

[14] Bogdan Mazoure, Remi Tachet des Combes, Thang Long Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. *Advances in Neural Information Processing Systems*, 33:3686–3698, 2020.

[15] Christopher DiMattina and Kechen Zhang. Active data collection for efficient estimation and comparison of nonlinear neural models. *Neural computation*, 23(9):2242–2288, 2011.

[16] Judea Pearl. Causal inference. *Causality: objectives and assessment*, pages 39–58, 2010.

[17] Eric Hiddleston. A causal theory of counterfactuals. *Noûs*, 39(4):632–657, 2005.

[18] Jodie B Ullman and Peter M Bentler. Structural equation modeling. *Handbook of Psychology, Second Edition*, 2, 2012.

[19] Bengt O Muthén. Beyond sem: General latent variable modeling. *Behaviormetrika*, 29(1):81–117, 2002.

[20] Frederick Eberhardt, Clark Glymour, and Richard Scheines. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 178–184, 2005.

[21] Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547, 2008.

[22] Alain Hauser and Peter Bühlmann. Two optimal strategies for active learning of causal models from interventional data. *International Journal of Approximate Reasoning*, 55(4):926–939, 2014.

[23] Jiaqi Zhang, Louis Cammarata, Chandler Squires, Themistoklis P Sapsis, and Caroline Uhler. Active learning for optimal intervention design in causal models. *Nature Machine Intelligence*, 5(10):1066–1075, 2023.

[24] Christian Toth, Lars Lorch, Christian Knoll, Andreas Krause, Franz Pernkopf, Robert Peharz, and Julius Von Kügelgen. Active bayesian causal inference. *Advances in Neural Information Processing Systems*, 35:16261–16275, 2022.

[25] Amanda Coston, Alan Mishler, Edward H Kennedy, and Alexandra Chouldechova. Counterfactual risk assessments, evaluation, and fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 582–593, 2020.

[26] Robert Brame and Raymond Paternoster. Missing data problems in criminological research: Two case studies. *Journal of Quantitative Criminology*, 19:55–78, 2003.

[27] Angela M Wood, Ian R White, and Simon G Thompson. Are missing outcome data adequately handled? a review of published randomized controlled trials in major medical journals. *Clinical trials*, 1(4):368–376, 2004.

[28] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[29] Rufus Lynn Carter. Solutions for missing data in structural equation modeling. *Research & Practice in Assessment*, 1:4–7, 2006.

[30] James L Arbuckle. Amos™ 7.0 user's guide. *Amos Development Corporation*, 1995.

[31] Craig K Enders and Deborah L Bandalos. The relative performance of full information maximum likelihood estimation for missing data in structural equation models. *Structural equation modeling*, 8(3):430–457, 2001.

[32] Taehun Lee and Dexin Shi. A comparison of full information maximum likelihood and multiple imputation in structural equation modeling with missing data. *Psychological Methods*, 26(4):466, 2021.

[33] Paul T Von Hippel. New confidence intervals and bias comparisons show that maximum likelihood can beat multiple imputation in small samples. *Structural Equation Modeling: A Multidisciplinary Journal*, 23(3):422–437, 2016.

[34] Bengt Muthén, David Kaplan, and Michael Hollis. On structural equation modeling with data that are not missing completely at random. *Psychometrika*, 52(3):431–462, 1987.

[35] John Banasik, Jonathan Crook, and Lyn Thomas. Sample selection bias in credit scoring models. *Journal of the Operational Research Society*, 54(8):822–832, 2003.

[36] Jonathan Crook and John Banasik. Does reject inference really improve the performance of application scoring models? *Journal of Banking & Finance*, 28(4):857–874, 2004.

[37] Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

[38] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

[39] Sahil Verma and Julia Rubin. Fairness definitions explained. In *Proceedings of the international workshop on software fairness*, pages 1–7, 2018.

[40] 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.

[41] 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.

[42] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

[43] Yongkai Wu, Lu Zhang, Xintao Wu, and Hanghang Tong. Pc-fairness: A unified framework for measuring causality-based fairness. *Advances in neural information processing systems*, 32, 2019.

[44] Junzhe Zhang and Elias Bareinboim. Fairness in decision-making—the causal explanation formula. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[45] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Fairness through causal awareness: Learning causal latent-variable models for biased data. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 349–358, 2019.

[46] Elliot Creager, David Madras, Toniann Pitassi, and Richard Zemel. Causal modeling for fairness in dynamical systems. In *International conference on machine learning*, pages 2185–2195. PMLR, 2020.

[47] Baqun Zhang, Anastasios A Tsiatis, Eric B Laber, and Marie Davidian. A robust method for estimating optimal treatment regimes. *Biometrics*, 68(4):1010–1018, 2012.

[48] Karima Makhlouf, Sami Zhioua, and Catuscia Palamidessi. Survey on causal-based machine learning fairness notions. *arXiv preprint arXiv:2010.09553*, 2020.

[49] Alejandro Noriega-Campero, Michiel A Bakker, Bernardo Garcia-Bulle, and Alex'Sandy' Pentland. Active fairness in algorithmic decision making. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 77–83, 2019.

[50] Frédéric Branchaud-Charron, Parmida Atighehchian, Pau Rodríguez, Grace Abuhamad, and Alexandre Lacoste. Can active learning preemptively mitigate fairness issues? *arXiv preprint arXiv:2104.06879*, 2021.

[51] Jacob Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, Chris Russell, and Jie Zhang. Active sampling for min-max fairness. *arXiv preprint arXiv:2006.06879*, 2020.

[52] Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in reinforcement learning. In *International conference on machine learning*, pages 1617–1626. PMLR, 2017.

[53] Naman Goel, Alfonso Amayuelas, Amit Deshpande, and Amit Sharma. The importance of modeling data missingness in algorithmic fairness: A causal perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7564–7573, 2021.

[54] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE international conference on data mining workshops*, pages 13–18. IEEE, 2009.

[55] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[56] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, pages 797–806, 2017.

[57] Yahav Bechavod and Katrina Ligett. Penalizing unfairness in binary classification. *arXiv preprint arXiv:1707.00044*, 2017.

[58] Felix Petersen, Debarghya Mukherjee, Yuekai Sun, and Mikhail Yurochkin. Post-processing for individual fairness. *Advances in Neural Information Processing Systems*, 34:25944–25955, 2021.

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

[60] Alan Mishler, Edward H Kennedy, and Alexandra Chouldechova. Fairness in risk assessment instruments: Post-processing to achieve counterfactual equalized odds. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 386–400, 2021.

[61] Justin B Biddle. On predicting recidivism: epistemic risk, tradeoffs, and values in machine learning. *Canadian Journal of Philosophy*, 52(3):321–341, 2022.

[62] David Danks and Alex John London. Algorithmic bias in autonomous systems. In *Ijcai*, volume 17, pages 4691–4697, 2017.

[63] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Prediction-based decisions and fairness: A catalogue of choices, assumptions, and definitions. *arXiv preprint arXiv:1811.07867*, 2018.

[64] Reuben Binns. Fairness in machine learning: Lessons from political philosophy. In *Conference on fairness, accountability and transparency*, pages 149–159. PMLR, 2018.

[65] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[66] Edgar C. Merkle and Yves Rosseel. blavaan: Bayesian structural equation models via parameter expansion. *Journal of Statistical Software*, 85(4):1–30, 2018.

[67] Edgar C. Merkle, Ellen Fitzsimmons, James Uanhoro, and Ben Goodrich. Efficient Bayesian structural equation modeling in Stan. *Journal of Statistical Software*, 100(6):1–22, 2021.

[68] Yves Rosseel. lavaan: An r package for structural equation modeling. *Journal of statistical software*, 48:1–36, 2012.

[69] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

[70] Stan Development Team. The Stan Core Library, 2024. Version 2.34.

[71] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved r-hat for assessing convergence of mcmc (with discussion). *Bayesian analysis*, 16(2):667–718, 2021.

[72] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[73] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[74] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[75] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

[76] Junaid Ali, Preethi Lahoti, and Krishna P Gummadi. Accounting for model uncertainty in algorithmic discrimination. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 336–345, 2021.

[77] Hao Wang, Luxi He, Rui Gao, and Flavio Calmon. Aleatoric and epistemic discrimination: Fundamental limits of fairness interventions. *Advances in Neural Information Processing Systems*, 36, 2024.

[78] Harini Suresh and John Guttag. A framework for understanding sources of harm throughout the machine learning life cycle. In *Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–9, 2021.

[79] Nathan Kallus and Angela Zhou. Residual unfairness in fair machine learning from prejudiced data. In *International Conference on Machine Learning*, pages 2439–2448. PMLR, 2018.

[80] Angelina Wang, Vikram V Ramaswamy, and Olga Russakovsky. Towards intersectionality in machine learning: Including more identities, handling underrepresentation, and performing evaluation. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 336–349, 2022.

[81] Reuben Binns. On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 514–524, 2020.

[82] Avijit Ghosh, Lea Genuit, and Mary Reagan. Characterizing intersectional group fairness with worst-case comparisons. In *Artificial Intelligence Diversity, Belonging, Equity, and Inclusion*, pages 22–34. PMLR, 2021.