The Dissertation Committee for Guy Wayne Cole
certifies that this is the approved version of the following dissertation:

Relational Learning and Fairness

Committee:

Sinead Williamson, Supervisor

Carlos Carvalho

Scott Moser

Peter Muller

Mingyuan Zhou

Relational Learning and Fairness

by

Guy Wayne Cole,

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2019

Dedicated to my wife Melissa.

Acknowledgments

I would like to take this opportunity to thank ...

Relational Learning and Fairness

Publication No. _____

Guy Wayne Cole, Ph.D.
The University of Texas at Austin, 2019

Supervisor: Sinead Williamson

This thesis will focus on relational learning in the modeling of text and user roles in networks, and the relative treatment of individuals as related to algorithmic fairness. With the exponential growth in social network data, the need for models of user interaction data is growing. This work presents a model which agglomerates users into archetypes based on topical modeling of the contents of their interactions. It further proposes models and a fairness metric for the creation of classifiers for individuals which control for the relative treatment of individuals.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

This thesis will focus on the relational learning in two areas: the modeling of text and user roles in networks, and the relative treatment of individuals as related to algorithmic fairness.

With the growth of social media, an increasing amount of new data generated occurs in a network context. In Chapter 2 I explore a model which combines the intuitive content description of topic modelling with the user archetyping of a stochastic blockmodel. The combination allows us to describe contextual relationships of users (nodes) in the network and the content typical of the messages they exchange.

In Chapter 3 I explore the concept of monotonic individual fairness. Individual fairness (Dwork et al., 2012) formalizes the intuitive idea similar individuals should be treated similarly. I extend this to the idea to formalize the idea that individuals' relative treatment should follow their relative qualification. I accomplish this by identifying non-protected attributes which should have a monotonic relationship with the outcome and enforcing that the learned prediction function maintains this monotonicity.

I extend this concept in Chapter 4 to the more realistic scenario in which monotonicity is derived from a sample of expert ratings. This allows for more complex relationships between individuals; fairness might dictate that an increase in one attribute has a larger effect than a similar increase in another. For example, a person a few violent felonies might considered more dangerous than a person with more non-violent misdemeanors.

1.2 Background

1.2.1 Bayesian Inference

Bayesian statistics view the parameters of a statistical model as themselves being random variables based on an interpretation of probability as the uncertainty of belief about a system or outcome. Mathematically, Bayesian statistics relies on Bayes' formula, derived from conditional probability as

$$\Pr(\theta|X) = \frac{\Pr(X|\theta)\Pr(\theta)}{\Pr(X)}.$$

While classical (or frequentist) statistics focus exclusively on the *likelihood* of the data ($\Pr(X|\theta)$ and what values of θ produce high likelihoods of the observed data, Bayesian statistics utlizes a *posterior* (*a posteriori*) distribution ($\Pr(\theta|X)$) of belief about θ. This posterior distribution is estimated via a *prior* (*a posteriori*) distribution ($\Pr(\theta)$, usually of belief about the distribution) of θ and, when a proportionate distribution is not sufficient, the *evidence* ($Pr(X)$) of the data under the model.

A variety of techniques have been developed for inference about parameters in Bayesian settings utilizing analytical, optimization, stochastic, and other approaches. Chapter 2 will utilize a stochastic method based on *Monte Carlo sampling*. Monte Carlo methods produce estimates about the posterior of a target parameter by producing samples from a distribution which approximates the posterior. For high dimensional θ, it is often intractible to produce samples of all parameters simultaneously, so sampling is accomplished by partitioning the parameter space and iteratively updating the partitions, e.g.

$$\Pr(\theta_1|X,\theta_2) = \frac{\Pr(X|\theta_1,\theta_2)\Pr(\theta_1,\theta_2)}{\Pr(X,\theta_2)}.$$

Since $\Pr(X,\theta_2)$ doesn't depend on θ_1, it can be viewed as a normalizing constant in the posterior and is usually omitted, leading to a proportional expression,

$$\Pr(\theta_1|X,\theta_2) \propto \Pr(X|\theta_1,\theta_2)\Pr(\theta_1,\theta_2).$$

Depending on the exact setting, a variety of tools can be used to generate samples of θ_1, including relatively old methods like the MetropolisMetropolis

et al. (1953) algorithm or Gibbs samplingGeman and Geman (1984) and relatively modern methods like Hamiltonian Monte CarloGirolami and Calderhead (2011) and the No-U-Turn Sampler (NUTS)Hoffman and Gelman (2014).

1.2.2 Topic Models

Topic models are a popular family of hierarchical Bayesian models for semantic analysis of corpora of documents. The canonical model of this type is Latent Dirichlet Allocation Blei et al. (2003), where each document is associated with a Dirichlet-distributed distribution over T "topics", which themselves are Dirichlet-distributed distributions over words that tend to concentrate on semantically coherent topics. Each word in the document is assumed to have been generated by sampling a topic from that document's distribution over topics, and then sampling a word from the topic's distribution over words.

This basic model has been extended in a number of directions. A hierarchy of Dirichlet processes can be used to construct a Bayesian nonparametric variant with an unbounded number of topics (Teh et al., 2007); a logistic normal distribution can be used to induce correlations between topics (Blei and Lafferty, 2007); time dependence has been incorporated to track topic evolution over time (Blei and Lafferty, 2006).

1.2.3 Stochastic Blockmodels

Stochastic blockmodels (Wang and Wong, 1987; Snijders and Nowicki, 1997) are a popular class of generative models that assume that each node within a network is associated with one of K latent clusters or communities. Each pair (k, ℓ) of communities is associated with a latent parameter $\lambda_{k,\ell}$, which parametrizes the interactions between members of those communities. Typically, the network is assumed to be binary, and the interactions are modeled as Bernoulli random variables.

A number of variants to the basic stochastic blockmodel have been proposed. (Karrer and Newman, 2011) uses a gamma/Poisson link in place of a beta/Bernoulli, to obtain distributions over integer-valued networks, and also incorporates a per-node parameter that allows nodes in the same community to

have different degree distribution. The Infinite Relational Model (Kemp et al., 2006) allows a potentially infinite number of communities, with membership probabilities distributed according to a Dirichlet process. Rather than restrict each node to a single cluster, the Mixed Membership Stochastic Blockmodel (Airoldi et al., 2008) associates each node with a distribution over clusters, allowing nodes to perform several social roles.

1.2.4 Neural Networks

Neural networks are a class of extremely functions defined by a series of alternating linear and non-linear transformations. The theory that neural networks can act as universal function approximators goes back several decadesCybenko (1989), and since then their use has grown steadily.

A single layer of a network can be expressed as

$$h_l = f_l(h_{l-1}; W_l, b_l, \sigma) = \sigma\left(W_l h_{l-1} + b_l\right)$$

with h_l being the output of the l'th layer, W_l being a matrix of weights describing a linear transformation from $\mathbb{R}^{|h_{l-1}|}$ to $\mathbb{R}^{|h_l|}$, a bias vector b_l, and a non-linear transformation function on $\mathbb{R}^{|h_{l-1}|}$. It commonly notated that h_0 is the input x and h_L, i.e. the output of the final L'th layer, is the output y.

There are several significant drawbacks of neural networks: the weight matrices W_l incur an extremely large number of parameters, the combination of exchangeable weight matrices and non-linear activation function creates a highly-multimodal and non-convex parameter space, and the resulting functions are prone to overfitting sample data.

1.2.5 Fairness in Machine Learning

Machine learning algorithms trained to infer relationships, classify individuals or predict individuals' future performance tend to replicate biases inherent in the data (Caliskan et al., 2017; Bornstein, 2018; Angwin et al., 2016). Worse, when these algorithms are used as tools in policy decision making, they can form parts of feedback loops that magnify discriminatory effects. For example, predictive policing algorithms aim to predict where crimes will

take place, but are trained on data from where crimes are reported or arrests are made – which can be skewed by biased policing and might not reflect the true crime map. If police officers are sent to areas with high predictive crime rate, they will tend to make more arrests there, increasing the algorithm's confidence and amplifying discrepancies between the crime rate and the arrest rate (Ensign et al., 2018; Lum and Isaac, 2016).

1.2.5.1 Fairness Metrics

Definitions of fairness in machine learning are generally (but not exclusively) divided into two camps based on their level of attention: group-level fairness and individual-level fairness.

Individual fairness aims to ensure that two individuals u and v with non-protected attributes X_u, X_v have similar outcomes if X_u and X_v are similar, even if their protected attributes differ. Concretely, (Dwork et al., 2012) describes a score function f as individually fair if it is Lipschitz-continuous w.r.t. some metric $\mathcal{D}$ on X, i.e.

$$d(f(X_u), f(X_v)) \leq \mathcal{D}(X_u, X_v) \; \forall \; u, v \in \mathcal{U} \tag{1.1}$$

where d is a metric on the space of outcomes. This encapsulates the notion that if two individuals are similar in terms of non-protected attributes, they should have similar outcomes.

Conversely, *group fairness* metrics aim to minimize population-level imbalances. For example, the notion of demographic parity (Dwork et al., 2012) requires that the predicted outcome $\hat{Y}$ is independent of the protected variable A. Equalized odds (Hardt et al., 2016) requires that the predicted outcome $\hat{Y}$ is independent of A conditioned on the true outcome Y, allowing a predictor to depend on A via Y. Equalized opportunity (Hardt et al., 2016) relaxes this condition in a classification task where the outcome $\hat{Y} = 1$ is seen as more desirable than $\hat{Y} = 0$, to require conditional independence between predictor $\hat{Y}$ and protected variable $\hat{A}$ only when $Y = 1$. Agarwal et al. (2018) show that demographic parity, equalized odds, and their variants can be expressed in terms of a set of linear constraints. In many cases, individual notions of fairness are at odds with group notions of fairness. For example, Dwork et al.

(2012) shows that individually fair functions achieve perfect demographic parity if and only if the distribution over individuals is similar across demographic groups.

Chapter 2

Stochastic Blockmodels with Edge Information

2.1 Overview

Stochastic blockmodels allow us to represent networks in terms of a latent community structure, often yielding intuitions about the underlying social structure. Typically, this structure is inferred based only on a binary network representing the presence or absence of interactions between nodes, which limits the amount of information that can be extracted from the data. In practice, many interaction networks contain much more information about the relationship between two nodes. For example, in an email network, the volume of communication between two users and the content of that communication can give us information about both the strength and the nature of their relationship.

In this work, we propose the Topic Blockmodel, [1] a stochastic blockmodel that uses a count-based topic model to capture the interaction modalities within and between latent communities. By explicitly incorporating information sent between nodes in the network representation, we are able to address questions of interest in real-world situations, such as predicting recipients for an email message or inferring the content of an unopened email. Further, by considering topics associated with a pair of communities, we are better able to interpret the nature of each community and the manner in which it interacts with other communities.

[1] Author's note: This work was developed concurrently and independently to Bouveyron et al. in Bouveyron et al. (2018), who develop a similar model, propose a different inference strategy, and apply it to the Enron data set as well as others.

2.2 Introduction

A key focus in statistical network analysis has been the search for low-dimensional representations of the observed structure. One of the most commonly used frameworks is the stochastic blockmodel (Wang and Wong, 1987; Snijders and Nowicki, 1997), where nodes are assumed to belong to one of K latent communities.

Typically, the networks modeled using stochastic blockmodels are binary, and interactions are modeled as Bernoulli random variables. However, binary interaction networks contain minimal information about the relationship between each pair of nodes, leading to a weakly informative likelihood. The presence or absence of an interaction between two nodes conveys only a single bit of information, meaning that for moderately-sized networks the posterior distribution can be very disperse. This in turn makes it difficult to infer fine-grained structure.

Fortunately, in real-life social networks, we typically have more information about the interaction between two entities. For example, in an email network, the number of emails sent between two users can be seen as a proxy for interaction strength. Further, the content of emails may be used to offer more information regarding the nature of the relationship between two individuals. Despite this rich trove of information associated with interactions, there has been little attempt in the blockmodel literature to exploit text sent across a network in learning community structure.

We propose the Topic Blockmodel, a network model that represents the interaction between two nodes not as a binary indicator variable, but as the totality of their communication. Concretely, we assume that an interaction comprises a sequence of words, such as an email chain or a conversation. Each pair of communities is associated with a count-based topic model which governs both the volume and the content of interactions between members of those communities.

The benefits of this richer formulation are two-fold. First, by associating a pair of communities with a distribution over topics rather than just a probability of interaction, we improve interpretability of the communities found. By considering the topics afforded high probability for a given

community-community pair, we can automatically generate an interpretable label characterizing the pair.

Secondly, we can use the resulting model to ask questions of interest about the network. For example, an email provider might wish to suggest recipients for an email being composed. By considering both the set of people with whom the author has previously corresponded and the text of the composed email, the Topic Blockmodel can make better predictions than a binary or integer-valued stochastic blockmodel. Another example might be flagging emails in a security application, where we want to identify emails on a given topic: by jointly modeling interactions and topics, we can use community information to make predictions about the topical content of an email based on its sender and recipient, even if the email is encrypted.

We begin in Section 2.3 by reviewing the stochastic blockmodel framework, and discussing existing methods that incorporate both network and topic information. We then present the Topic Blockmodel in Section 2.4. After briefly describing inference in Section 2.5, we showcase the performance of the Topic Blockmodel on real data in Section 2.6. By looking at a naturally generated network of emails, and a semi-realistic network based on characters in a play, we demonstrate that the Topic Blockmodel yields both interpretable clusters, and impressive predictive performance both in terms of recipient prediction given a communication's text and author, and topic prediction given a communication's sender and recipient. Finally, we discuss possible extensions in Section 2.7.

2.3 Background

The Topic Blockmodel presented in this work is a stochastic blockmodel that incorporates both a count model and a topic model in its likelihood. Sections 1.2.3 and 1.2.2 introduced stochastic blockmodels and topic models; here, I discuss existing models that combine these approaches.

2.3.1 Existing Network-Based Topic Models

A number of works have attempted to combine network and topic models. They loosely fall into two camps: models that treat the network as a fixed covariate used to guide the topic model; and models that jointly model a corpus of documents and an associated network. An example of the first type of model is the Author-Recipient Topic Model (McCallum et al., 2005), which uses the network to specify a separate topic distribution for each sender-recipient pair. This does not allow for the elucidation of community structure, or provide conditional distributions over recipients.

The second type of model treats the network and the text as two related datasets described using a single probabilistic model. The Relational Topic Model (Chang and Blei, 2009) and the Poisson mixed-topic link model (Zhu et al., 2013) use the topic assignments of two documents to determine the probability of an interaction between them, resulting in a binomially-distributed number of links associated with each document. The Citation Author Topic Model (Tu et al., 2010) associates each topic with a distribution over words and a distribution over cite-able authors, and uses this to generate a set of interactions.

Another model that falls under this framework is the Joint Gamma Process Poisson Factorization (J-GPPF) model (Acharya et al., 2015), which models interactions between nodes using an infinite blockmodel and associates each community with a distribution over topics; the topics associated with an author's community membership are used to generate documents written by that author. The J-GPPF model is the closest approach to this work, since it explicitly clusters users into communities and uses those communities to guide a topic model. However, like all the models described above, the J-GPPF model assumes the (binary) network is modeled as a distinct entity from the documents. This is appropriate where an individual is associated both with a collection of documents and a set of connections—for example, in a scientific setting, the documents might be an author's papers, and the connections might be the set of people they have cited. J-GPPF does not translate into this work's setting, where the network is implicitly defined by the text sent across it.

By contrast, rather than conditioning on the network, or modeling it jointly, the Topic Blockmodel explicitly uses a topic model as a link function in a stochastic block model. Treating the text and the relationship as equivalent captures the idea that the collection of documents sent from node s to node r encapsulates their relationship. In this setting, we extract information not just from the fact that Alice sent emails to Bob about football; we also make use of the fact that Alice sent *no* messages to Claire. This absence of a link between Alice and Claire is informative about the underlying community structure.

2.4 Stochastic Blockmodel with Topic Links

In summary, we adopt the Poisson links introduced by (Karrer and Newman, 2011) to capture the communication volume between nodes. We place conjugate Beta priors on the $\lambda_{k,\ell}$, and a Dirichlet-multinomial prior on the community memberships. This model could be extended to incorporate the nonparametric and mixed membership behavior described in 1.2.3; however as we discuss in Section 2.7, this would significantly increases the computational cost of the model and we leave this for future work.

Following the basic stochastic blockmodel framework, we assume a distribution $\phi \sim \text{Dirichlet}_K(\xi_0)$ over K communities, and associate each node s with a cluster $c_s \sim \text{Discrete}(\phi)$ sampled from this distribution. We then associate each pair (k, ℓ) of communities with a set of parameters $\lambda_{k,\ell}$.

Unlike the binary stochastic blockmodel, where $\lambda_{k,\ell} \in (0, 1)$ is a beta-distributed random variable used to parameterize Bernoulli links, we let $\lambda_{k,\ell} = (\lambda_{k,\ell}^{(1)}, \ldots \lambda_{k,\ell}^{(T)})$ be a vector of gamma-distributed random variables. The tth element of this vector, $\lambda_{k,\ell}^{(t)}$, controls the number of words in topic t that are sent from a member of community k, to a member of community ℓ. Concretely, we let $n_{s,r}^{(t)}$, the number of words in topic t sent from node s to node r, be distributed according to $\text{Poisson}(\lambda_{c_s,c_r}^{(t)})$. The total number of words, $n_{s,r}^{(\cdot)} = \sum_{t=1}^{T} n_{s,r}^{(t)}$, sent from node s to node r is therefore Poisson-distributed with parameter $\lambda_{c_s,c_r}^{(\cdot)} = \sum_{t=1}^{T} \lambda_{c_s,c_r}^{(t)}$. Marginally, $\lambda_{c_s,c_r}^{(\cdot)}$ is a $\text{Gamma}(T\alpha_\lambda, \beta_\lambda)$ random variable.

In order to complete the model specification, I specify a topic-specific distribution $\eta_t \sim \text{Dirichlet}_V(\kappa)$ over the size-V dictionary for each of the T

topics. For each of the $n_{s,r}^{(t)}$ words associated with topic t, we then sample a word token according to η_t. The full generative process can therefore be summarized as

$$\begin{aligned}
\eta_t &\sim \text{Dirichlet}_V(\kappa),\ t \in \{1,\ldots,T\} \\
\phi &\sim \text{Dirichlet}_K(\xi_0) \\
c_s &\sim \text{Discrete}(\phi),\ s \in \{1,\ldots,S\} \\
\lambda_{k,\ell}^{(t)} &\sim \text{Gamma}(\alpha_\lambda, \beta_\lambda),\ k,\ell \in \{1,\ldots,K\} \\
n_{s,r}^{(t)} &\sim \text{Poisson}(\lambda_{c_s,c_r}^{(t)})\ s,r \in \{1,\ldots,S\} \\
w_{s,r,i}^{(t)} &\sim \text{Discrete}(\eta_t),\ i \in \{1,\ldots,n_{s,r}^{(t)}\}.
\end{aligned} \tag{2.1}$$

Here, $w_{s,r,i}^{(t)}$ is the identity of the ith token sent from node s to node r under topic t. Rather than simply have two nodes' community memberships determine the probability of an interaction between them, in the Topic Blockmodel each pair of communities provides a distribution over the number of words sent in each of T topics, determining both the overall volume of communication and its semantic content.

An equivalent specification can be obtained by noting that, conditioned on the total number of words $n_{s,r}^{(\cdot)}$ sent from node s to node r, the assignment of words to topics is given by a multinomial distribution parameterized by $\theta_{c_s,c_r} = \left(\lambda_{c_s,c_r}^{(1)},\ldots,\lambda_{c_s,c_r}^{(T)}\right)/\lambda_{c_s,c_r}^{(\cdot)}$. Further, this vector of probabilities θ_{c_s,c_r} is independent of the normalizing constant $\lambda_{c_s,c_r}^{(\cdot)}$, and is distributed $\text{Dirichlet}_K(\alpha_\lambda)$. If we let $z_{s,r,i} = t$ if the ith word sent from node s to node r is in topic t, we can rewrite the model as

$$\begin{aligned}
\lambda_{k,\ell}^{(\cdot)} &\sim \text{Gamma}(T\alpha_\lambda),\ k,\ell \in \{1,\ldots,K\} \\
\theta_{k,\ell} &\sim \text{Dirichlet}_T(\alpha_\lambda) \\
n_{s,r}^{(\cdot)} &\sim \text{Poisson}(\lambda_{c_s,c_r}^{(\cdot)})\ s,r \in \{1,\ldots,S\} \\
z_{s,r,i} &\sim \text{Discrete}(\theta_{c_s,c_r}),\ i \in \{1,\ldots,n_{s,r}^{(\cdot)}\} \\
w_{s,r,i} &\sim \text{Discrete}(\eta_{z_{s,r,i}}),
\end{aligned} \tag{2.2}$$

where the distributions over η_t, ϕ and c_s are as given in Equation 2.1.

These two equivalent formulations prove useful for inference. As we will see in Section 2.5, the Dirichlet-multinomial formulation of Equation 2.2 allows

us to use standard LDA updates for the $z_{s,r,i}$. Conversely, the gamma-Poisson formulation of Equation 2.1 yields a straightforward-to-calculate likelihood for Gibbs sampling the cluster assignments.

The Topic Blockmodel described above offers clear advantages over the models described in Section 2.3.1, without adding unnecessary complexity. In the models discussed previously, either the network was treated simply as a covariate, or it was modeled separately in a manner that assumes a marginally Binomial distribution over the number of recipients. This model is appropriate in the setting where the documents *are* the network, and the strength of an interaction is directly implied by the length of a document. In addition, we obtain latent community structure, which was not available from most of the models discussed previously.

2.5 Inference

Since the hierarchical model is composed of conjugate pairs and we can separate the distribution over the total number of words from the conditional distribution over the nature of those words, construction of a Gibbs sampler is straightforward. This sampler iteratively updates the community assignments c_s for each node s, and the topic assignments $z_{s,r,i}$ for each word.

Conditioned on the community memberships c_s and the number $n_{s,r}^{(\cdot)}$ of words sent from node s to node r, the updates for the topic assignments $z_{s,r,i}$ are standard LDA updates (see for example (Griffiths and Steyvers, 2004)), except with a topic mixture for each cluster pair rather than each document.

Conditioned on the topic assignments, we can sample the cluster memberships according to

$$\begin{aligned} P(c_s = k|\text{rest}) \propto (m_k^{-s} + \xi_0) \\ \times \prod_{j=1}^{K} \prod_{t=1}^{T} P(\{n_{s,r}^{(t)} : c_r = j\}|c_s = k, \text{rest}) \\ \times \prod_{j=1}^{K} \prod_{t=1}^{T} P(\{n_{r,s}^{(t)} : c_r = j\}|c_s = k, \text{rest}), \end{aligned} \tag{2.3}$$

where m_k^{-s} is the number of nodes in community k (excluding the sth node). The likelihood terms in the second and third line are straightforward to calculate due to gamma-Poisson conjugacy.

2.6 Experimental Evaluation

In order to assess the interpretability and predictive power of the posterior obtained using the Topic Blockmodel, we ran experiments on two real-world datasets, comparing against a range of competing models.

2.6.1 Datasets

We considered two datasets: A real-world email network and a network of fictional characters.

ENRON emails: The ENRON email dataset (Leskovec et al., 2009) is a commonly used dataset for social network research, and is very well-suited to this setting: correspondents belong to a closed network of company employees resulting in a fairly dense network, and the text of emails is included in the dataset. We considered all emails found in the Sent folders of ENRON-based email addresses, that were sent only to other ENRON-based email addresses, and excluded individuals who sent and received fewer than 10 emails. We removed standard stopwords, plus any words that occur more than 500 or fewer than 10 times in the corpus. This resulted in a dataset with a total of 48,064 non-stopwords sent between 90 email addresses, with a dictionary of length of 944.

Interactions in "A Midsummer Night's Dream": Due to a lack of publicly-available email interaction networks, we supplement the ENRON dataset with an interaction network automatically generated from Shakespeare's "A Midsummer Night's Dream". We considered each speech a directed interaction from the speaker to the last person to speak; the first speech of each scene is not included in the dataset.

Admittedly, this dataset suffers limitations. The social network and interaction structure are not naturally occurring and are inherently stylized. Further, this data extraction method is imperfect: during multiple scenes between the Athenian characters, Puck and other fairy characters are onstage but assumed invisible to the humans. Puck's asides and soliloquies are recorded as messages to the last human to speak, although this is not the author's intended interpretation. Despite these limitations, we find this dataset a useful addition since the main characters will be familiar to many readers, and naturally fall into a range of communities, such as the young Athenian lovers (Hermia, Lysander, Demetrius, and Helena) and the characters in the play-within-a-play (Prologue, Lion, Pyramus, Thisbe, Wall, and Moonshine).

We removed standard stopwords, Elizabethan words that are equivalent to these stopwords, and the names of characters, plus words occurring more than 50 times in the play, resulting in a total of 5913 non-stopwords sent between 28 characters, with a dictionary of length 2,204.

2.6.2 Comparison Methods

We compare the Topic Blockmodel against a range of comparison models, including models for text that take a network as a covariate; network models that ignore text; and standard topic models.

- **Latent Dirichlet allocation** (LDA) (Blei et al., 2003), a topic model that ignores network structure.
- The **Author Recipient Topic Model** (ART) (McCallum et al., 2005), which uses the network as a covariate, and has a separate distribution over topics for each sender/recipient pair.
- A stochastic blockmodel with a gamma/Poisson link, which we will refer to as the **Poisson Stochastic Blockmodel** (Poisson-SBM). This can model the number of words exchanged, but not their content.
- The **Clustered Node Topic Model** (CNT), a reduced version of the Topic Blockmodel which does not use a distribution over counts, instead conditioning on the observed counts. This model begins with the same

distribution over community assignments and, similar to the specification in Equation 2.2, specifies a distribution for the vector of probabilities θ_{c_s,c_r} for each pair of communities, without any rate parameters. In full,

$$\begin{aligned}
\phi \sim & \text{Dirichlet}_K(\xi_0) \\
c_s \sim & \text{Dirichlet}(\phi),\ s = 1, \ldots, S \\
\theta_{k,\ell} \sim & \text{Dirichlet}_T(\alpha_\lambda) \\
z_{s,r,i} \sim & \text{Discrete}(\theta_{c_s,c_r}),\ i \in \{1, \ldots, n_{s,r}^{(\cdot)}\} \\
\eta_t \sim & \text{Dirichlet}_V(\kappa),\ t = 1, \ldots, T \\
w_{s,r,i} \sim & \text{Discrete}(\eta_{z_{s,r,i}}).
\end{aligned} \tag{2.4}$$

Due to the similarities between the models, all models were sampled using appropriately modified versions of the sampler described in Section 2.5. During the first 500 burn-in samples, we used simulated annealing to improve exploration, with the temperature set as $\tau = e^{1-m/500}$, where m is the iteration. Hyperparameters were sampled with low-information priors using Metropolis-Hasting sampling. The number of topics was selected by cross validation, and the number of communities was set to $S/3$ for Shakespeare and $S/4$ for ENRON, where S is the number of nodes.

2.6.3 Qualitative Evaluation

We begin with a qualitative analysis of the community structure found using the Topic Blockmodel on "A Midsummer Night's Dream", since reader familiarity with the characters allow for easy evaluation of the clusters found. Figure 2.1 shows the community structure obtained using a single sample from the Markov chain (to avoid alignment issues). Here, the shade of element (s, r) of the matrix represents the gamma random variable $\lambda_{c_s,c_r}^{(\cdot)}$ governing the total number of words sent from node s to node r. The community structure can be inferred by looking at the discontinuities: nodes in the same community have the same parameter.

The names of the characters are given on the left hand axis, and some interesting communities are manually annotated on the right. Note that the communities generated are fairly well aligned with the character groupings

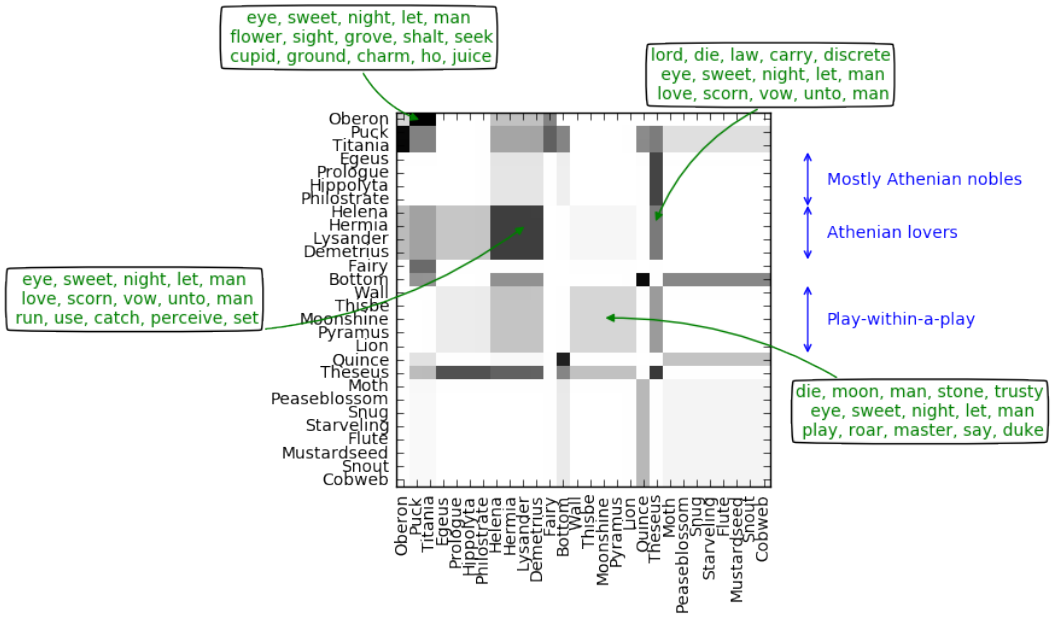

Figure 2.1: Communities found in "A Midsummer Night's Dream", with highest-probability topics associated with community pairs.

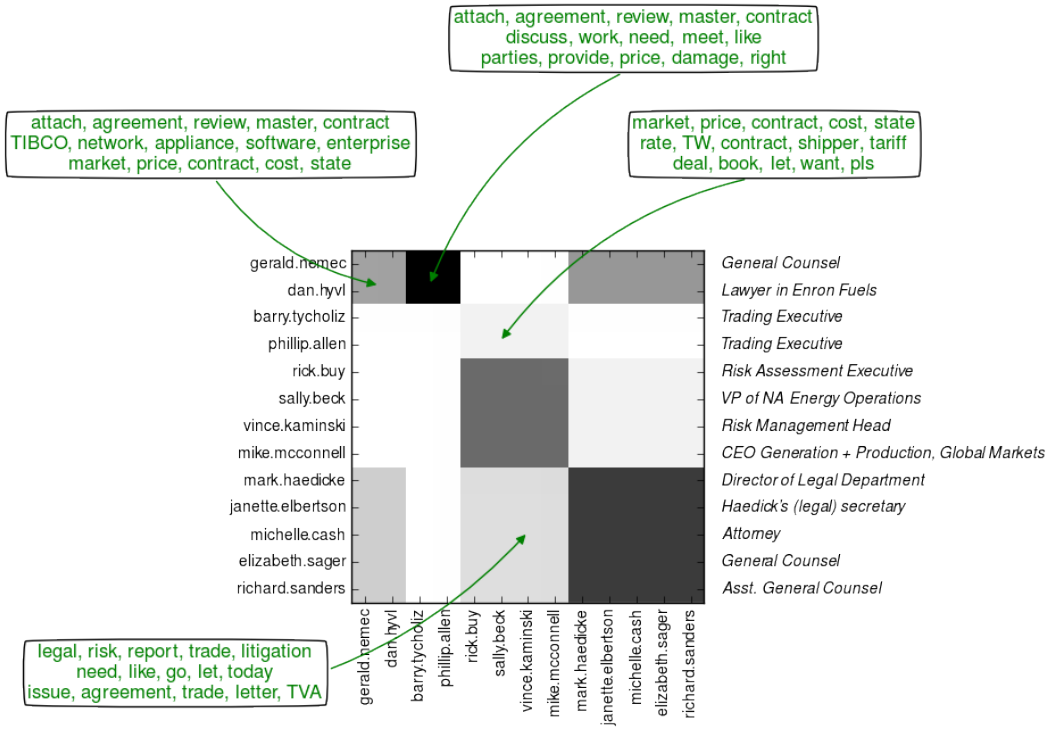

Figure 2.2: Communities found in the ENRON e-mail corpus for select e-mail participants, with highest-probability topics associated with community pairs.

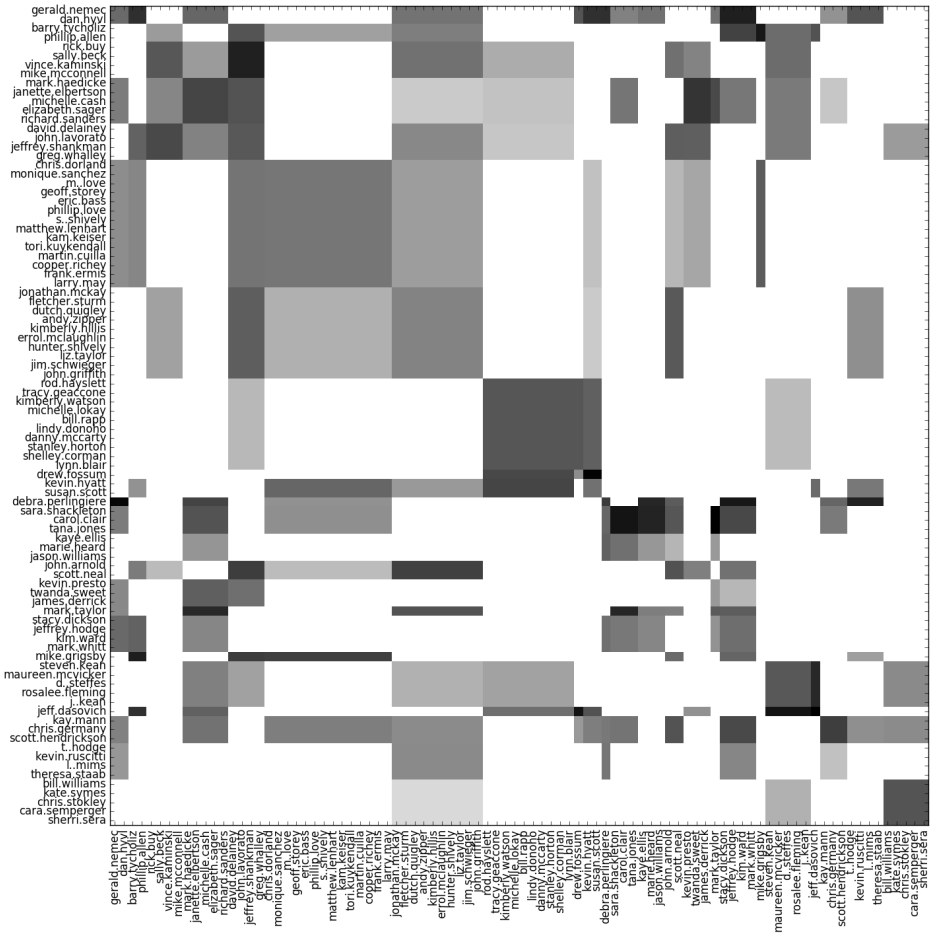

Figure 2.3: Communities found in the ENRON e-mail corpus for all ENRON internal e-mail participants.

present in the play. For example, Demetrius, Helena, Hermia, and Lysander represent a ring of romantically entangled Athenians; Egeus, Hippolyta and Philostrate are elder Athenian nobility; Wall, Prologue, Thisbe, Moonshine, Pyramus and Lion are all characters in the play-within-a-play; Titania and Puck are both fairies who interact with Oberon in a similar manner. The outliers are mostly characters with very few lines – for example the minor fairies and the minor mechanicals are intermingled, but all these characters have very few lines.

To demonstrate how the topics characterize the community's relationships, we consider four community-community pairs that discuss love - a major theme of "A Midsummer Night's Dream". While all the selected pairs contain a shared topic of romantic words, the additional topics shed nature on the communities' nature. The star-crossed Athenian lovers talk among themselves of love and hate, and talk to Duke Theseus about the consequences of their romantic choices; Oberon talks to Puck and Titania of magical slumber and fairy mischief; the play-within-a-play characters talk about aspects of the play and appeal to their audience.

Figures 2.2 and 2.3 show the discovered latent social network between a subset of the ENRON employees. For ease of interpretability, Figure 2.2 provides an annotated subset of the Enron employees, their exchanged topics, and the employees' roles in the company; Figure 2.3 shows the full network. From Figure 2.2 we see that attorneys Gerald Nemec and Dan Hyvl are in a community, as are the trading executives Barry Tycholiz and Phillip Allen, as are executives involved in energy development and risk Rick Buy, Sally Beck, Vince Kamisnki, and Mike McConnell. The fourth community shown is again legal professionals, but with different subject areas.

In the absence of this job title information, one could still use the topics associated with the community-community pairs to improve understanding of the latent network. We see that the attorneys' emails are focused on agreements and contracts, and supplying advice to the other employees. When the trading executives are talking with the development community, however, they are primarily discussing elements of economic forecasts (market, price, cost, rate, contract, tariff, etc.). When the second attorney group is writing to the risk group, their topics skew more toward legal risks (e.g. litigation) and

government affairs (e.g. dealing with the Tennessee Valley Authority (TVA)) than the contracts advice that the first group of attorneys gives to the trading executives.

2.6.4 Quantitative Evaluation

We evaluated the predictive performance of the topic blockmodel on four metrics:

1. Log predictive likelihood of the text of held-out documents (conditioned on number of words sent, since this is required for most of the comparison methods). This is designed to mimic the task of predicting the topical content of an email from its sender and recipient.

2. Log predictive likelihood of the recipient of a held-out email/speech, conditioned on the sender and the text of the communication. This is designed to mimic the task of suggesting recipients for an email.

3. Log predictive likelihood of the sender and recipient of a held-out email/speech. This is designed to showcase the fact that using the text information allows us to better model latent community structure.

4. Log predictive likelihood of the word counts of held-out sender-receiver pairs. This is designed to show that the inclusion of topic information improves count prediction.

Table 2.1: Log predictive likelihood (± one standard error) of document text, conditioned on sender and recipient where applicable.

Model	ENRON	Shakespeare
LDA	-410,110.2 ± 50.8	–48,716.2 ± 4.6
ART	-365,600.5 ± 47.7	-47,495.5 ± 4.8
CNT	-368,983.5 ± 89.2	**-46,076.6 ± 3.9**
Topic Blockmodel	**-345.632.5 ± 4.1**	-46,275.9 ± 4.0

Table 2.2: Log predictive likelihood (± one standard error) of document recipient, conditioned on document content and sender where applicable.

Model	ENRON	Shakespeare
ART	-204,585.3 ± 6.4	-19,809.7 ± 1.1
CNT	-216,278.9 ± <0.1	-19,703.3 ± <0.1
Poisson-SBM	-160,984.7 ± 148.6	-14,587.2 ± 35.9
Topic Blockmodel	**-137,199.8 ± 53.2**	**-12,997.8 ± 20.6**

Table 2.3: Log predictive likelihood (± one standard error) of document sender and recipient, conditioned on document content where applicable.

Model	ENRON	Shakespeare
ART	-416,588.6 ± 6.8	-39,580.0 ± 1.0
CNT	-432,557.7 ± <0.1	-39,406.7 ± <0.1
Poisson-SBM	-347,479.6 ± 148.6	-31,400.3 ± 35.9
Topic Blockmodel	**-321,127.8 ± 53.3**	**-29,614.0 ± 20.6**

2.6.4.1 Log-likelihood of words in held-out documents

For the first task, we randomly held out 10% of documents, and evaluated the predictive log likelihood of this test set using the comparison models with a topic model component (i.e. LDA, ART, and CNT). The log predictive likelihoods are shown in Table 2.1.

We see that the Topic Blockmodel performs significantly better than the competitors on the ENRON dataset. In this realistic setting, the number of emails sent between two individuals is highly indicative of their relationship, so we see a significant advantage from jointly modeling the number of words and their content. In particular, we see that the Topic Blockmodel outperforms our Clustered Node Topic Model variant, which does not model counts and treats zero edges as missing.

On the Shakespeare data, the Topic Blockmodel performs slightly worse than the Clustered-Node Topic model, though still better than LDA or ART. We believe that this is due to the artificial nature of the network. The community structure in "A Midsummer Night's Dream" is man-made, and designed

Table 2.4: Log predictive likelihood (± one standard error) of sender and recipient counts.

Model	ENRON	Shakespeare
Poisson-SBM	-92,851.2 ± 12.1	-103,411.4 ± 0.6
Topic Blockmodel	**-88,730.4 ± 3.1**	**-102,549.8 ± 0.2**

so that the many separate communities interact in complex, artful manners. Moreover, by assuming a speech is directed to (only) the previous speaker, we are working with a noisy approximation to Shakespeare's intended interaction network. Since the Clustered-Node Topic Model does not model the number of links, it will be less hampered by an unrealistic network structure.

2.6.4.2 Recipient Attribution

For the second task, designed to mimic automatic email recipient suggestion, we again held out 10% of documents and predicted the recipient of each document based on the document's length, text and sender. We compared against the three comparison methods with a network component, namely ART, CNT, and the Poisson Stochastic Blockmodel. Prediction in ART and CNT does not take into account the number of words sent; prediction in the Poisson Stochastic Blockmodel does not take into account the specific words sent. Table 2.2 shows the test set log predictive likelihood for the four methods on the recipient attribution task.

In the ENRON e-mail data, we again see that the Topic Blockmodel performs significantly better than any of the competitive models in identifying the correct sender-recipient pair, with the Poisson Stochastic Blockmodel coming second and the two models that do not consider word counts performing worst. The relative performance of the Poisson Stochastic Blockmodel (which does not consider topic distributions) versus CNT and ART (which do not consider word counts) suggests that count modeling, rather than topic modeling, is the more important component in this setting; however by combining these two components the Topic Blockmodel is able to make use of the topic distribution to improve prediction over the purely count-based model.

We see a similar pattern in the Shakespeare data: the Topic Blockmodel outperforms the Poisson Stochastic Blockmodel and all other models, and the models that just consider topical content of documents perform worse than the Poisson Stochastic Blockmodel that only considers counts. This is again likely for similar reasons to ENRON: the models on interaction intensity are able to down-weight pairs that very rarely interact, greatly boosting the likelihood of pairs that are expected to interact, and further identifying the correct topic mixture within high-intensity community pairings.

2.6.4.3 Sender/Recipient Attribution

For the third task, we again held out 10% of documents and predicted both sender and recipient based on a document's length and text, comparing against ART, CNT and the Poisson Stochastic Blockmodel. The resulting log predictive likelihoods, shown in Table 2.3, tell a similar story to the sender attribution task: the Poisson Stochastic Blockmodel, which only considers document length, outperforms CNT and ART which only consider document text, suggesting document length is more important than document semantic content in this task. However, the Topic Blockmodel, by making use of both length and semantic content, is able to outperform all three comparison methods on both tasks.

2.6.4.4 Edge Count Prediction

Finally, we withheld 10% of sender-receiver pairs in the network and predicted the word count of the withheld links based on the assigned communities of the sender and receiver. Table 2.4 shows that, in both the ENRON and Shakespeare data sets, the Topic Blockmodel significantly improves on the Poisson Stochastic Blockmodel, which is the only comparison model discussed which models the word counts of heldout links.

2.7 Discussion and Future Work

In this work we introduced a unified network and topic model, the Topic Blockmodel. Inspired by existing stand-alone network and topic models, the

Topic Blockmodel can be used to identify and label communities in a network and make predictions about interactions.

We have focused here on networks where the interactions are textual in nature. However, we may also have networks where interactions take the form of images, audio, or some combination of media. A future research direction might be to explore augmenting this model with other forms of media to better make use of information shared across the network, using likelihoods such as those described in (Cao and Fei-Fei, 2007), (Niu et al., 2012) or (Kim et al., 2009).

Other extensions could be obtained by using a richer distribution over the community structure. We chose a simple, parametric model with single-community membership to allow for straightforward computation; however the potential for mixed-membership or nonparametric versions is clear. Another interesting avenue for research is to make the distribution over communities explicitly dependent on some set of covariates such as time of email or geographical location of nodes, creating a dynamic model.

One limitation of the stochastic blockmodel framework is that it is only appropriate when our network is dense – that is, when the number of non-zero edges grows quadratically with the number of nodes. This is a reasonable assumption in relatively small networks where it is likely that all nodes have had a chance to interact with each other – for example, groups of individuals within a school, company or organization, as we have explored in this work.

An interesting parallel line of research, which we are currently exploring, is models for text-based interaction in *sparse* data. Such a model would require replacing the stochastic blockmodel component of the model with a distribution appropriate for sparse graphs, such as those described by (Caron and Fox, 2017), (Veitch and Roy, 2015), (Cai et al., 2016), (Crane and Dempsey, 2016) and (Williamson, 2016). Without such a significant change to the model, one possible direction would be to add node-specific degree-correcting parameters as proposed by (Karrer and Newman, 2011).

Chapter 3

Monotonic Fairness

3.1 Overview

Classifiers that achieve demographic balance by explicitly using protected attributes such as race or gender are often politically or culturally controversial due to their lack of individual fairness, i.e. individuals with similar qualifications will receive different outcomes. Individually and group fair decision criteria can produce counter-intuitive results, e.g. that the optimal constrained boundary may reject intuitively better candidates due to demographic imbalance in similar candidates. Both approaches can be seen as introducing individual resentment, where some individuals would have received a better outcome if they either belonged to a different demographic class and had the same qualifications, or if they remained in the same class but had objectively worse qualifications (e.g. lower test scores). We show that both forms of resentment can be avoided by using monotonically constrained machine learning models to create individually fair, demographically balanced classifiers.

3.2 Introduction

As discussed in Section 1.2.5, machine learning algorithms trained without care can reproduce latent biases in the training data used. This tendency can be counteracted by designing algorithms that aim to yield similar accuracy across different demographics. One approach is to design algorithms that explicitly use information about the protected variable in developing the algorithm, whether by transforming the attributes of each demographic group (Dwork et al., 2012), learning embeddings that transform each demographic group to comparable representations (Madras et al., 2018; Zemel et al., 2013), or training separate classifiers on each group (Dwork et al., 2018a).

While these approaches are powerful tools for combating systemic inequalities, algorithms that aim for demographic fairness can appear unfair or opaque on the individual level. For example, we can achieve demographic fairness in college admissions by applying different cutoffs for different groups, but individuals below the cutoff for their demographic group but above the cutoff for a different demographic group will feel unfairly treated. Even if the different cutoffs can be justified on a population level—for example, if certain demographic groups have statistically disparate access to educational resources, leading to lower average test scores—they are often unpopular among the class with the stricter cutoffs, and can result in complaints and legal action. For example, Universities' affirmative action policies have frequently been the target of legal action from students who feel that they have been unfairly denied entry when compared with similarly qualified members of other ethnic groups, both past (Court, 2013, 2016, 1978) and ongoing (Court, 2014). In practice, this often means that we must pick a single decision boundary for all groups, even if this limits the fairness of the resulting outcome.

Conversely, algorithms that exhibit individual fairness—where two similar individuals are treated similarly even if their demographic group differs—can easily propagate unfairness on a population level. Schools are often highly racially segregated due to location, and schools in wealthy, majority-white neighborhoods tend to have more resources and funding, which are in turn correlated with better academic performance in high school (on Civil Rights, 2018; for Education Statistics , Ed). This better performance in high school does not necessarily translate to better performance at the university level (Vidal Rodeiro and Zanini, 2015).

Further, even within an individually fair system, individuals might still feel resentment towards their peers. Individual fairness can be seen as minimizing resentment between two individuals with similar attributes but different demographic group memberships: neither individual feels they would have had a more favorable outcome if they could switch their membership. However, it can still lead to resentment between two individuals with different attributes, if those attributes admit a natural ordering: if student A has a higher SAT score than student B and is identical on all other axes, student A would feel resentment if student B had the higher acceptance probability. This can amplify

demographic discrepancies if the demographic-specific attribute distributions differ: if the SAT scores of a minority group trended notably higher than SAT scores of a majority group, an admissions system could still satisfy individual fairness while accepting primarily low-scoring individuals.

The goal of this work is to automatically design decision rules that avoid individual resentment—both resentment towards someone with similar attributes but a different demographic group membership, and resentment towards someone with "worse" attribute values—while minimizing population-level unfairness. We demonstrate that this approach allows us to design rules that trade off predictive accuracy with group notions of fairness, while avoiding perceived unfairness on an individual level.

3.3 Notions of fairness

We consider models for individuals characterized by some set of protected or sensitive attributes $A_i \in \mathcal{A}$ and non-protected attributes $X_i \in \mathcal{X}$. Our goal is to predict some outcome Y_i; in this work we focus on binary classification problems where $Y_i \in \{0, 1\}$, but our approach can easily be applied in a regression setting where $Y_i \in \mathbb{R}$.

Protected attributes might be race or gender; we assume that these attributes are categorical, but this assumption can be relaxed. Non-protected attributes include other information relevant to decision making, such as test scores or credit history. These attributes might be highly correlated with our protected variables (for example, attending a historically black university is highly correlated with race), meaning that we cannot avoid unfair outcomes simply by excluding the protected attributes from our analysis (sometimes referred to as fairness through unawareness (Dwork et al., 2012)).

As discussed in Section 1.2.5.1, fairness approaches are generally divided into treating at an *individual* or *group* level, however a number of approaches attempt to balance individual and group notions of fairness. Dwork et al. (2012) combine demographic parity with a relaxed notion of statistical parity, where members of group A' are first mapped to match the distribution of group A via a Lipschitz-continuous mapping. Later work expands this idea by mapping individuals' protected and non-protected attributes into some

latent embedding or representation that is uninformative of the protected attribute (Zemel et al., 2013; Madras et al., 2018). Using such a mapping can lead to individual resentment w.r.t. the protected attribute, however, since changing an individual's protected attribute value would change its embedding, and hence its outcome.

An alternative approach is to learn a single classifier on X to predict Y, and to encourage fairness by regularization using a fairness-promoting penalty (Kamishima et al., 2011, 2012; Berk et al., 2017) or constraints (Zafar et al., 2017a,b; Agarwal et al., 2018). If the classifiers used are Lipschitz-continuous, then they are all individually fair, since each individual is subject to the same classification function. The form of this function is governed by a trade-off between predictive accuracy, and some appropriate measure of (group-level) fairness. While this trade-off means regularization approaches may achieve lower accuracy and/or group-level fairness than representation-based approaches, their individual fairness yields transparency in implementation and avoids situations where individuals would have different outcomes under counterfactual protected attributes.

Our approach builds upon this family of regularization-based algorithms. We introduce a new measure of fairness that protects against counterfactual resentment w.r.t. shifts in both protected and non-protected variables, even outside the training set. Loosely, our idea of monotonic fairness protects against two sources of resentment: the perception that one would have been better off in a different demographic group, and the perception that one would have been better off had they under-performed along a given axis.

Our work also complements a body of work which explores definitions of fairness in which groups are collectively satisfied (Zafar et al., 2017a; Heidari et al., 2018), with variations on being *a priori* ambivalent or being *a posteriori* free of desire to switch labels as a group. These variations deal with the idea of resentment at a class level, while we examine it at an individual level.

Others have considered the idea of individual-level comparisons; Balcan et al. (2018) explore the concept of "envy freeness" in classification in the context of individual-specific utility functions, where a classifier can be optimal when no individual's utility function would be higher if they received the predicted outcome (or distribution of outcomes) given to an individual with

different attributes. This approach could not be applied to settings where the utility function is assumed to be identical among individuals, e.g. in most classification tasks where this is a preferred outcome that all individuals would prefer.

"Meritocratic fairness" Joseph et al. (2016) appears similar, but differs in that it ranks points based on the expected outcome for their attribute values rather than the actual attribute, i.e. it is monotonic w.r.t. the expected true outcome rather than the predictors so that (in one form) if $\mathbb{E}[Y|X_u] > \mathbb{E}[Y|X_v]$ then $\hat{f}(X_u) \geq \hat{f}(X_v)$). Our approach differs in that we require monotonicity w.r.t. those inputs believed to directly correlate with performance (detailed in section 3.4).

Lipton et al. Lipton et al. (2018) study concepts of impact disparity and treatment disparity which overlap our own. Their concept of impact disparity is similar to statistical parity, that protected classes should be treated similarly overall. They conceive of treatment disparity similarly to our own *class resentment*, that individuals' treatments differ based on their protected class. Our work expands on this to incorporate *score resentment*, and proposes and evaluates a concrete framework for structurally enforcing protection.

Others have considered the problem of monotonicity in fair methods. (Kearns et al., 2017) explores the notion of monotonicity in the context of combining rankings between groups which lack common attributes, e.g. when comparing the athleticism of athletes from different sports. Their method assumes that a perfect ranking is known within each sport, and compares athletes across sports using the sport-specific CDF of the outcome variables. Our method does not assume such a CDF estimate is obvious or accessible, and will not produce a separate classifier for each class of examples. Similarly, (Dwork et al., 2018b) consider decoupled classifiers for separate classes, and how they can be combined to produce fair classification. Our model does not learn separate classifiers, which can introduce resentment between classes, but instead seeks to learn a unified classifier which satisfies fairness and prediction goals.

3.4 Monotonic fairness

Consider a model that outputs a score $f(X, A)$ to an individual with non-protected attributes X and protected attribute A, where higher scores in some dimensions of X are seen as more desirable. An example of X_u being "better" than X_v might be if the non-protected attributes correspond to SAT score, with X_u being the higher score.

We assume in the remainder of this work that non-protected attributes X can be represented in $\mathbb{R}^d$. In general, we can subdivide X into X^+ and X°, where X^+ contains variables like SAT score, where certain values are deemed better than others, and X° variables like number of years in current position, where we do not wish to impose such value judgements.

This work considers the concept of *individual resentment*, which can take the form of either *class resentment* and/or *score resentment*, which we define below.

Definition 3.4.1. Protected Attribute Resentment (Class) Resentment: Individual u experiences *class resentment* under function f if $\exists\ A'$ s.t. $f(X_u, A_u) < f(X_u, A')$.

Class resentment occurs when an individual who differs from another only in protected attributes receives a less-preferred outcome than that other individual, despite having identical non-protected attributes. Even though there may be justifiable reasons for the discrepancy, the first individual is likely to perceive the system as penalizing them for their protected attribute.

Definition 3.4.2. Non-Protected Attribute (Score) Resentment: Individual u experiences *score resentment* under function f if there exists (X', A') such that X_u is objectively "better" than X' but $f(X_u, A_u) < f(X', A')$.

Score resentment captures the situation where an individual receives a less-preferred outcome than another individual who differs only in having "worse" scores in some dimensions – for example, a candidate being rejected for being over-qualified for a job. While score resentment is typically not encoded into hand-designed systems, it can easily appear in automatically learned systems, as we discuss later in this section.

Individually fair methods ensure that two individuals with similar non-protected attributes receive similar outcomes, avoiding the situation where an individual feels he or she would have been better treated had they belonged to a different demographic group—what we refer to above as protected attribute, or class, resentment. However, individual fairness does not necessarily avoid *non-protected* attribute, or score, resentment— the situation where an individual feels he or she would have been better treated had they performed worse on some axis.

We can ensure a score function has zero individual resentment by requiring that the function does not take the protected attribute as an input (guaranteeing zero protected attribute resentment) and is monotone non-decreasing w.r.t. all non-protected attributes in X^+ (guaranteeing zero non-protected attribute resentment). We refer to such a score function as being *monotonically fair*.

Definition 3.4.3. Monotonic Fairness: A function $f : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is *monotonically fair* if no possible individual $(X, A) \in \mathcal{X} \times \mathcal{A}$ experiences *class resentment* (Def 3.4.1) or *score resentment* (Def 3.4.2).

To understand the difference between individual fairness and monotonic fairness, consider a system that admits students to college on the basis of a single standardized test. If the predictor is not non-decreasing w.r.t. that test result, a student could be in the unfair situation where they would have been accepted if their test result were lower. Similarly, a loan applicant might find themselves rejected for borrowing *less* money. Such a predictor could arise, even if the true relationship between test score and probability of college success is monotonic, if our training data is sparse or demographically imbalanced in some area of the attribute space and especially in higher dimensional settings.

The synthetic example in Figure 3.1 demonstrates such a situation. We consider the setting where we wish to create a soft classifier, $\hat{p}_i = f(X_i)$ which maximizes the average score of positive predictions $\sum_i \hat{p}_i Y_i$ with a constraint on the expected number of positive classifications $\sum_i \hat{p}_i$—this might correspond to admitting a fixed number of students based on their predicted future performance. The true relationship is that $Y \sim N(X, \epsilon)$. Our classes

are imbalanced and have different distributions, as shown in Figure 3.1. An "unfair" classifier that does not aim to achieve demographic fairness, learns a hard threshold at $X = 1$ but leads to 2.58 times higher odds of acceptance for the majority class vs. the minority class.

We can achieve a more fair result by adding a penalty that encourages demographic parityHardt et al. (2016), which requires that the probability of a favorable outcome be independent of class, i.e. $\frac{\sum_{i:A_i=0} \hat{p}_i}{\sum_{i:A_i=0} 1} = \frac{\sum_{i:A_i=1} \hat{p}_i}{\sum_{i:A_i=1} 1}$. Adding such a penalty reduces the odds ratio from 2.58 to 1.13. However, in order to maximize demographic parity, the fair classifier ends up learning a non-monotone function. All those with $X \in (0.9, 4.0)$ receive predictions lower than those with $X = 0.9$ regardless of protected attribute. Clearly, this would lead individuals in the region to resent individuals with lower attribute values: individuals in this range would have a better chance of a positive outcome if they had a "worse" value of X.

By contrast, a monotonically fair classifier ("Mono. Fair") learns a function that avoids the score resentment present in the "Fair" classifier, while achieving similar demographic parity (odds ratio 1.11). No individual can claim that another individual with a lower non-protected attribute value received a higher probability of acceptance. This is achieved by reducing the certainty of acceptance from those with the highest attribute values, which are increasingly majority-dominated, and reducing the threshold attribute value required to have any chance of acceptance.

If we add in the requirement that our score function is Lipschitz-continuous, we can see monotonic fairness as an extension of individual fairness. Where $X^\circ \neq \emptyset$ and we have non-protected attributes that do not require monotonicity, incorporating a Lipschitz requirement avoids seemingly arbitrary discontinuities across X°. Where $X^\circ = \emptyset$ and where we require monotonicity along all dimensions of X, the Lipschitz requirement is likely to be less important, since any discontinuities will favor higher-valued attributes. Further, enforced monotonicity will likely lead to smoother functions with fewer discontinuities than non-monotone solutions.

3.5 Learning monotonic fair scores using neural networks

As described above, any score function whose value does not depend on the protected attribute, and that is monotonically non-decreasing with each dimension of X^+, will have zero individual resentment under the conditions discussed in Section 3.4.[1] A number of algorithms have been proposed to learn monotone functions; Cano et al. (2019) offers a detailed review. We choose to use feedforward neural networks, since they are flexible and easily adapted to a specific problem.

We restrict our analysis to situations where value comparisons are only made between individuals who differ in a single dimension of their non-protected attributes. In practice, this covers a large number of realistic use cases: it is easier for a practitioner to specify orderings in such settings. Ordinal categorical variables can be captured either by mapping the categories to integers, or by using dummy variables and setting the dummies for all categories worse than the actual category. We leave relaxation of these assumptions, and approaches for automatically learning orderings, to future work. We also assume that ordering of attributes $X^k \in X^+$ correspond to some notion of "value", where we wish to impose the requirement that increasing X^k does not decrease the chance of the more desirable outcome, provided the other attributes do not change, i.e. the relationship is monotonic. If necessary, the attributes may have been transformed by the practitioner to achieve this (e.g. mapping categories onto the reals).

If we desire our function to be monotone non-decreasing with respect to every dimension of X, we can enforce this by ensuring all weights in the network are strictly positive, for example by applying some transformation $\tau : \mathbb{R} \to \mathbb{R}_+$ Sill (1998). In the more general setting, where we wish to be monotone w.r.t. $X^k \in X^+$ but do not require this for $X^k \in X^\circ$, partition the weights in our neural network into those that will be multiplied by (functions of) X^+, and those which will not. In a simple feedforward neural network setting, that means that in the first layer, weights corresponding to $X^k \in X^+$

[1] In this section, we only consider the monotonically non-decreasing case; the monotonically non-increasing case can be considered analogously.

are forced to be positive, while weights corresponding to $X^k \in X^\circ$ are not. In subsequent layers, all weights are required to be positive. Concretely, we apply the following transformations to the unconstrained weights $w_{\ell,k,i}$ of the neural network:

$$\tilde{w}_{\ell,k,i} = \begin{cases} \tau(w_{\ell,k,i}) & \text{if } \ell > 1 \text{ or } X^k \in X^+ \\ w_{\ell,k,i} & \text{if } \ell = 1 \text{ and } X^k \in X^\circ \end{cases} \tag{3.1}$$

$$h_{\ell,k} = \sigma\left(\sum_i \tilde{w}_{\ell,k,i} h_{\ell-1,i} + b_{\ell,k}\right). \tag{3.2}$$

The output is clearly a monotone non-decreasing function[2] of each $X^k \in X^+$, since all weights in the path of such X^k are positive. Leaving $w_{1,k,i}$ unconstrained for $X^k \in X^\circ$ allows for the function to be non-monotonic w.r.t. those X^k.

In our experiments, we use an offset form of the exponential linear unit Clevert et al. (2015) transformation,

$$\tau(x) = \begin{cases} x & \text{if } x > 1 \\ e^{x-1} & \text{if } x \leq 1 \end{cases}, \tag{3.3}$$

in Equation 3.2 to transform the appropriate weights to be positive. Note that any continuously differentiable function with strictly positive range could be substituted; we selected the offset exponential linear unit based on experimental performance. We explore other choices in the supplement.

Figure 3.2 explores the effect of the transformations τ. We show the outputs of two neural networks: One where all weights are transformed according to Equation 3.3 (Mono. NN), and one where the first layer is untransformed but subsequent layers are (Non-Mono. NN). The first network demonstrates that this architecture is able to learn monotonic functions even when the true function is non-monotone. The second network demonstrates that, provided the first layer is not transformed, the transformation of weights in subsequent

[2]We assume the use of an activation function which is also monotone non-decreasing, which is common (e.g. ELU, ReLU, leaky ReLU, tanh, sigmoid) but not universal.

layers does not interfere with fitting arbitrary functions with the usual precision (and drawbacks) of feedforward neural networks. Since we can arbitrarily transform the edge weights between a subset of the inputs and the first layer, we can also fit higher-dimensional functions which are monotonic only on a subset of the inputs. See the supplement for two-dimensional examples.

Neural networks have been used to learn fair classifiers in a number of contexts (Louizos et al., 2016; Beutel et al., 2017; Madras et al., 2018; Xu et al., 2018). Dwork et al. Dwork et al. (2012) originally posited individual affirmative action within a framework of Lipschitz smoothness. In many commonly used architectures (including the ones used in this work), neural networks describe Lipschitz-continuous functions, although the Lipschitz constant may be large (Szegedy et al., 2014; Gouk et al., 2018; Balan et al., 2018). One could also enforce greater smoothness by Lipschitz continuity-aware regularization (Gouk et al., 2018). We choose not to do so in our experiments, relying on the monotonicity constraints to add additional regularization, to ensure that any jumps (w.r.t. $X_k \in X^+$) are individually fair, and to enforce that the effective decision rule does not create the potential for resentment.

In addition to monotonic fairness, we also want to ensure our algorithm has desirable group-level fairness properties. To do so, we train our monotonic neural network using backpropagation to minimize a compound loss

$$\mathcal{L}(\theta) = \lambda_P \mathcal{L}_P(\theta) + \lambda_F \mathcal{L}_F(\theta)$$

evaluated on a minibatch, where $\mathcal{L}_P$ is a prediction loss, $\mathcal{L}_F$ is a fairness loss, and $\lambda_P, \lambda_F \geq 0$ are weights governing the relative importance assigned to each loss.

The fairness loss, possibly derived from a constraint, encourages a desired form of fairness, and is calculated across the entire minibatch. A variety of differentiable losses have been developed that could be deployed here Kamishima et al. (2011, 2012); Berk et al. (2017); Zafar et al. (2017a,b); Agarwal et al. (2018). In our experiments, we use the demographic loss proposed by Zemel et al. (2013), $|\bar{y}_0 - \bar{y}_1|$, i.e. the absolute difference in mean prediction between majority and minority classes.

The prediction loss is some loss that penalizes predictions that are far from ground truth, for example cross-entropy or MSE. This loss is typically

evaluated individually for each data point, and then summed over the minibatch.

3.6 Experiments

We evaluated our method on three real-world examples of increasing complexity: law school admissions, COMPAS scoring of recidivism risk in bail decisions, and German credit assessment in granting loans. In each case, both our protected variable A and our target Y are binary. We specify our compound loss as a convex combination of cross-entropy and equality of outcome, following the example of Zemel et al. (2013), though other measures are interchangeable if they are differentiable. Concretely, for a minibatch $\mathcal{M} = (X_i, Y_i, A_i)_{i=1}^{M}$, we have:

$$\mathcal{L}(\theta; \alpha, \mathcal{M}) = \underbrace{(1-\alpha)\frac{1}{M}\sum_{i=1}^{M} -\left(Y_i \log\left(\hat{p}(X_i;\theta)\right) + (1-Y_i)\log\left(1-\hat{p}(X_i;\theta)\right)\right)}_{\mathcal{L}_P}$$
$$+\underbrace{\alpha\left|\frac{\sum_{i:A_i=1}\hat{p}(X_i;\theta)}{\sum_{i:A_i=1}1} - \frac{\sum_{i:A_i=0}\hat{p}(X_i;\theta)}{\sum_{i:A_i=0}1}\right|}_{\mathcal{L}_F}$$

where $\hat{p}(X_i;\theta)$ is the output of our neural network, and $\alpha \in (0,1)$ controls the balance between fairness and prediction.

We compare against both a neural network with the same compound loss but no monotonicity constraints—which is representative of the set of individual-classifier methods described in Section 3.3—and the Fair Representations method Zemel et al. (2013). The Fair Representations method establishes *prototypes* for the data, each equipped with a location in data space and a mean outcome value, with actual data given a mixed membership vector to these prototypes based on a spherical Gaussian kernel. A penalty for demographic balance within each prototype's membership rate forces predictions to have demographic balance. This method achieves individual fairness since any two individuals with similar (unprotected) attributes will be given a similar

outcome, and the mixed membership via kernels produces a Lipschitz-smooth outcome function.

3.6.1 Datasets

Law school admissions data (Wightman and Ramsey, 1998): This dataset contains data from 9800 male and 7600 female law school students[3] from 1991, with an outcome variable of normalized first year average (ZFYA) grades in law school and non-protected attributes of undergraduate grade point average (UGPA) and LSAT score (LSAT).[4] We use gender as our protected attribute, and binarize the outcome by setting $Y = 1$ whenever ZFYA ≥ 0.09, its median value. One result with an apparently erroneous UGPA of 0.0 was removed before analysis. Figure 3.3 shows contour plots of the per-gender non-protected attribute distributions, generated by adding uniform noise to counter the discretization of the data then using kernel density estimation. We see that female students tend to have higher GPA, but lower LSAT scores, than the male students (see Figure 3.3).

COMPAS data Larson et al. (2016): Released in 2016 following a public interest investigation into machine learning methods in criminal justice, the COMPAS dataset (named for the proprietary system which generated it) contains the risk factors, demographic information, and two-year recidivism information for over 7,000 individuals arrested in southern Florida in 2013 and 2014. We reduced this to a two-class problem by restricting our analysis to the 6,150 "African American" and "Caucasian" examples in the dataset,[5] and attempt to predict the two-year recidivism risk of the accused based on their

[3]The data has a pre-separated test set of 4,358 individuals; we additionally set aside 3,486 of the training examples as a validation set.

[4]The LSAT exam has undergone extensive change since this data was collected in 1991. Our analysis is motivated by the real-world dataset, but our conclusions are not necessarily applicable to the current exam. In addition, the dataset is limited to individuals admitted to law school and is not a representative sample of all test takers (many of whom would not have an observed outcome).

[5]We set aside 1,235 as a test set, and 658 as a validation set for the neural network models.

age (non-monotonic) and number of prior adult convictions, juvenile felony, misdemeanor, and other convictions (all monotonically non-decreasing).

German credit data Lichman (2013): Covers 1,000 credit applicants in Germany,[6] including their employment, financial, and residency information, as well as the type of loan they requested and whether they repaid it. We treat age (already binarized by the data source) as the protected attribute. There are 58 attributes in the dataset, of which we converted 7 into monotonic numeric variables: (monotonic non-decreasing) current checking account balance, credit history, employment tenure, and savings balance, and (monotonic non-increasing) investment as income percentage, length of loan in months, and credit amount. In the case of monotone non-increasing inputs, the corresponding weights in the first layer are transformed to be negative, rather than positive. These were done intuitively, based on the idea that no one should be penalized for having more money in reserve, more stable employment, or better credit history, and no one should be rewarded for increasing the borrowed amount or requesting more months to pay it back, holding all other things constant.

3.6.2 Models

For each dataset, we trained three models:

- FNN: A non-monotonic, feedforward Fair Neural Network with 4 hidden layers of 10 nodes and *tanh* activation functions[7] using an ADAM optimizer.
- FMNN: A Fair Monotonic Neural Network otherwise identical but with monotonically-transformed weights where appropriate.
- FR: Fair Representations Zemel et al. (2013) with 10 prototypes.

[6]We randomly select 20% (200) to use as a test set, and 20% of the training set (160) are set aside by the neural network models for validation data.

[7]For monotonic networks, an activation function with bounded range is useful in order to allow the function to be non-convex; see supplemental materials.

For each model, we trained 100 versions of the model with α randomly sampled according to a Beta$(0.5, 0.5)$ distribution. This distribution allowed us to heavily sample near the bounds to accommodate imbalanced losses. For the FR model, we also randomly sampled a value for their coverage penalty A_x from a log-uniform distribution between 10^{-2} and 10^2 (and setting $L_Y = 1-\alpha$ and $L_Z = \alpha$). All datasets were scaled to have marginal variance of 1 for all input dimensions, as unequal scales can affect coverage statistics. For the neural network models, minibatching (size 256 for COMPAS and Law School, 128 for German) and stepwise scoring on a 20% validation subset (taken from the training data) were used to prevent overfitting.

3.6.3 Results

In Figure 3.4 we see the usual accuracy-discrimination trade-off in the upper row of plots. Accuracy and discrimination are defined as in Zemel et al. (2013):

- Discrimination: $\left| \frac{\sum_{n:s_n=1} \hat{y}_n}{\sum_{n:s_n=1} 1} - \frac{\sum_{n:s_n=0} \hat{y}_n}{\sum_{n:s_n=0} 1} \right|$
- Accuracy: $1 - \frac{1}{N} \sum_{n=1}^{N} |y_n - \hat{y}_n|$

In most cases, we see that the monotonic neural network is of similar or slightly lower accuracy than the non-monotonic neural network or the Fair Representations approach for a given level of discrimination. This is unsurprising, since the non-monotonic methods are free to learn an unconstrained function. We would only expect the monotonic method to yield better predictions if the underlying data has a strictly monotonic generating function. However, we see that the loss in accuracy is generally small and likely tolerable across all three example datasets.

In the bottom row of plots in Figure 3.4, we see a different trade-off: the cost in individual resentment for improving group fairness. Here, resentment is measured as the proportion of individuals in the test set who experience individual resentment, as defined in Section 3.4. Specifically,

- Resentment: $\frac{1}{N} \sum_{i=1}^{N} \max_{j \in \mathcal{N}_i} \left(1_{\hat{y}_i < \hat{y}_j}\right)$

where $\mathcal{N}_i$ is the set of $j \neq i \in \{1 \dots N\}$ where X_i is "better" than X_j or $X_i = X_j$ and $A_i \neq A_j$. In practice, since none of the methods use the protected attribute as an input, this is equivalent to the number of individuals who experience non-protected attribute (score) resentment, i.e. they had a higher attribute in a monotonically non-decreasing dimension (or a lower one in a non-increasing dimension) than a hypothetical individual with a more favorable prediction (and identical non-protected attributes).

Due to the high dimensionality of some of the datasets, we restricted our consideration of resentment to individuals who feel resentment towards a peer in the test set, rather than resentment towards a hypothetical individual with worse scores. Note that, as the dimension of the attribute space increases, the sample estimate will underestimate resentment, due to a decreasing number of individuals with comparable attributes. For example, in the law school admissions setting, it is easy for an individual to find peers with lower UGPA but the same LSAT scores; conversely, for the German credit data, a comparable individual must match on 51 attributes. However, the resentment of the monotonic neural network will always be zero by design.

Let us explore the Law school dataset in more detail. Figure 3.3 shows the comparative distributions of males and females w.r.t. GPA and LSAT score. Note that the female distribution is shifted towards higher UGPA and lower LSAT score than the male distribution. In Figure 3.5, we see the admissions probabilities produced by the monotonic and non-monotonic neural networks. When α is high, we see that individuals would often do well to *lower* their reported LSAT score in order to increase their probability of admission. This is an artifact of the disproportionate number of women with high UGPA and low LSAT scores, resulting in a "fair" classifier which favors lower LSAT scores for individuals high UGPA, similar to the example in Figure 3.1. Even though there is no resentment across protected variable groups, there clearly would be resentment by those who are less likely to receive a favorable outcome due to a counter-intuitive admissions policy designed to produce demographic balance.

3.6.3.1 Lipschitz constant

Although our method is not primarily intended to produce a smoother function, i.e. one with a lower Lipschitz constant, it is a desirable property for individually-fair functions. Zhang et al. Zhang et al. (2019) provide a discussion of the advantages and disadvantages of several types of empirical estimators of the Lipschitz constant for a neural network.

We adopt a sample-based estimator similar to that of Wood and Zhang (1996), which uses a pairwise evaluation of the constant,[8] i.e.

$$\hat{L} = \max_{i,j} \left(\left| \frac{\hat{Y}_i - \hat{Y}_j}{d(X_i, X_j)} \right| \right)$$

We calculate our Lipschitz constant with respect to a standardized Euclidean distance,

$$d(X_i, X_j) = \sqrt{\sum_k \left(\frac{X_i^k - X_j^k}{\hat{s}_k} \right)^2}$$

where $\hat{s}_k$ is the sample standard deviation of X^k. We standardize in this manner so smoothness is comparable across dimensions. As discussed in Zhang et al. (2019), this sample estimate is a lower bound of the true constant, but we feel it is adequate for model comparison.

In Figure 3.6, we see that the monotonic neural network tends to produce smoother solutions for a given value of discrimination than other methods in more inherently-monotonic settings like the Law School dataset than in less inherently-monotonic settings like COMPAS or German Credit. This is unsurprising, since the monotonicity constraint acts as a regularizer, preventing overfitting to spurious non-monotonic trends in sampled data.

[8]The method proposed by Wood and Zhang (1996) further fits estimates a parametric distribution of the values to find an estimate of the maximum, but that method requires a random sample of points which is infeasible here. We instead use the maximum of empirical distribution, which is biased downwards but adequate for comparison purposes.

3.7 Discussion

Individually fair classifiers can exhibit unfair behavior on a population level, and can lead to the undesirable situation where an individual who performed worse on a given metric would have had a better outcome, leading to resentment. We show that a definition of individual fairness that incorporates monotonicity can avoid the latter situation, and can be combined with measures of demographic fairness to yield classifiers that trade off predictive power with demographic fairness.

Several recent works suggest important future directions.

Estimation of monotonic relationships: A critical requirement of individual fairness as originally proposed Dwork et al. (2012) is a distance metric over $\mathcal{X}$ to determine the degree of similarity between individuals. Our work sidesteps the problem by relaxing the requirement from a distance metric to a concept of ordering. Recent concurrent works Jung et al. (2019); Ilvento (2019) have explored the concept of estimating a distance metric by polling fair experts on what constitutes similarity. We can similarly imagine extending the current work by polling fair experts instead on which individuals should receive higher outcomes than others, and enforcing coherence between the trained prediction function and the poll results on orderings. This would allow one to relax the requirement of explicitly monotonic dimensions in the input data.

Post hoc adjustment for monotonicity: Recent works, e.g. Lohia et al. (2019), have attempted to use post hoc adjustments and model pooling prevent biases in machine learning. These methods approach machine learning methods as black box function estimators, and instead of modifying the input data or function space of the models, use post hoc adjustment of the trained models' predictions in order to create fairness. It is reasonable to consider whether we can extend this general applicability to the current approach; if we have a classifier which satisfies other concepts of fairness and accuracy, we may be able to manipulate its outputs to induce monotonicity on their outputs without interfering in the "black box."

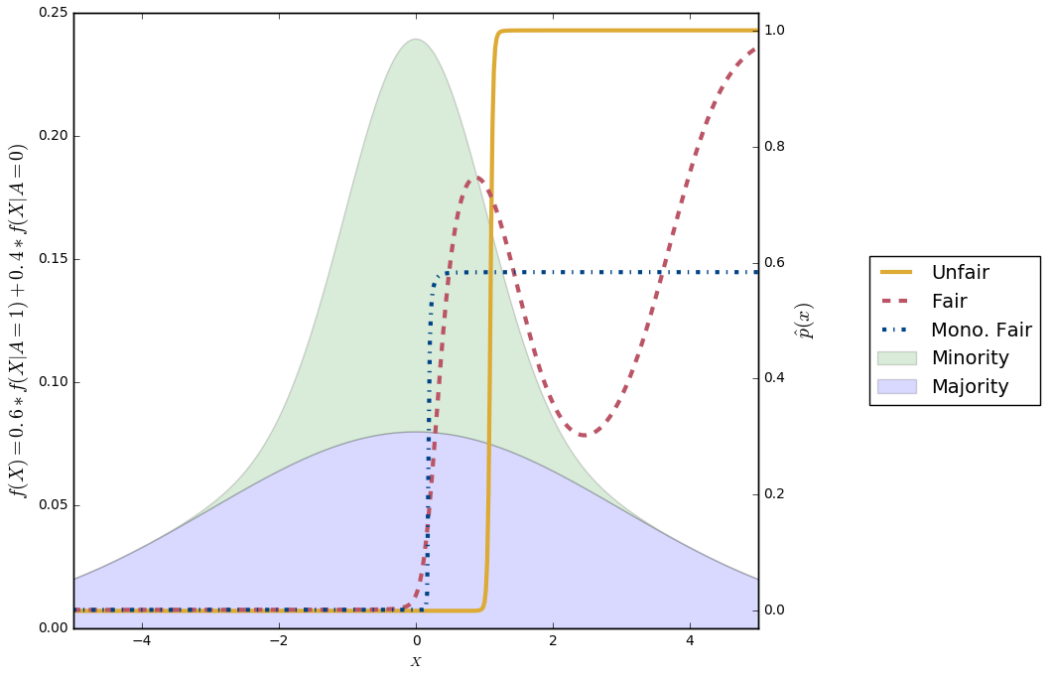

Figure 3.1: The distribution of X for the minority class (light green, $X|A = 0 \sim N(0,1)$) differs from that of the majority class (light blue, $X|A = 1 \sim N(0,3)$). We have $P(A = 1) = 0.6$. For both classes, $Y = X + \epsilon$, ϵ $N(\mu = 0, \sigma = 0.1)$ – i.e. the chance of success increases with X. "Unfair" (yellow solid line) is an unconstrained neural network soft classifier which maximizes expected outcome score of positive predictions subject to a constraint on expected number of positive predictions. "Fair" (red dashed line) adds the restriction that we must have equal expected probability of positive prediction for both classes. "Mono. Fair" (dark blue dash-dot line) adds the further constraint that the prediction function must be monotonic.)

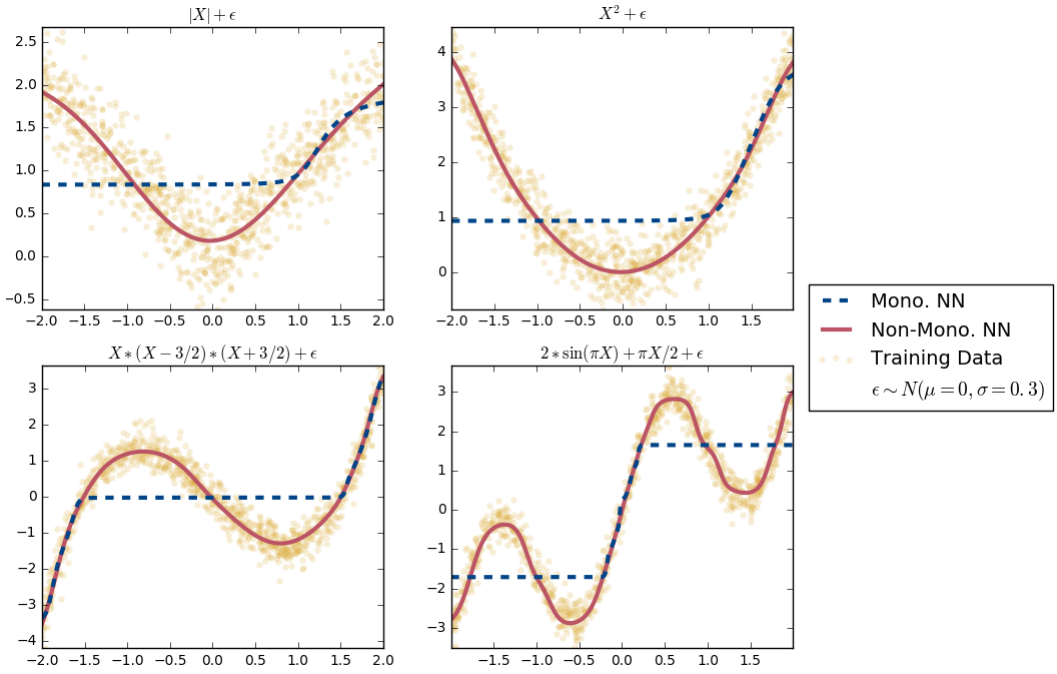

Figure 3.2: Training data (yellow circles, $n = 1000$ for each), monotonic neural network (dashed blue line), and non-monotonic neural network with transformed weights after the first layer (solid red line) approximations for training data sampled from four example functions.

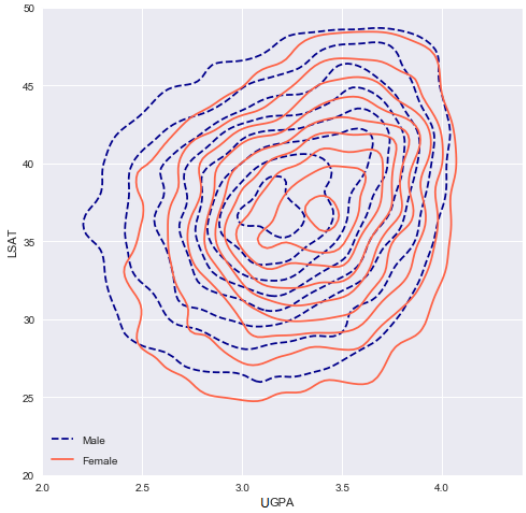

Figure 3.3: Distribution over UGPA and LSAT for male and female students. Female students tend to have higher GPA, but lower LSAT scores.

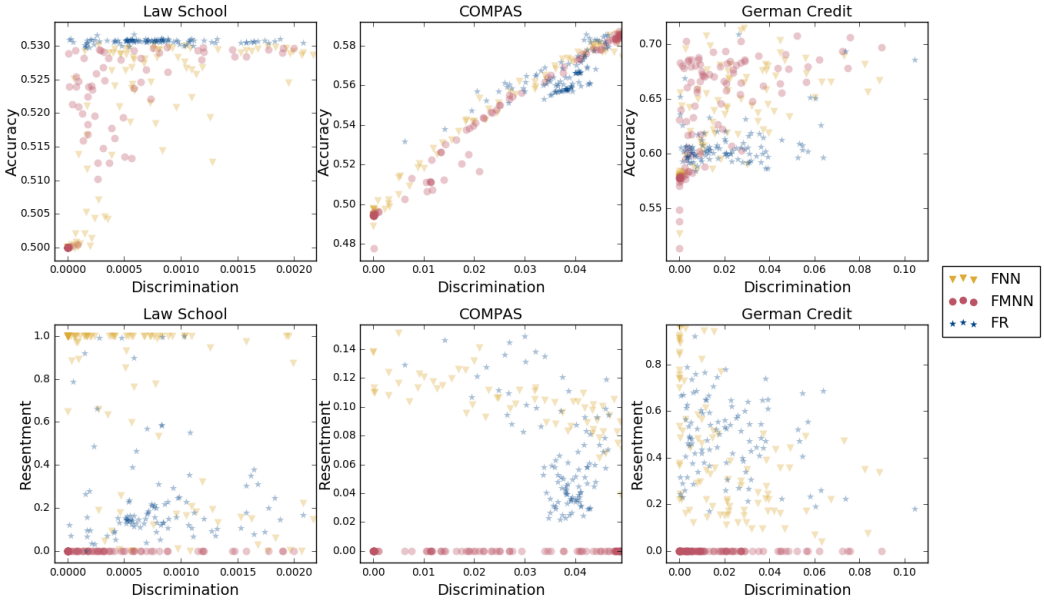

Figure 3.4: Accuracy vs Discrimination (top row) and Discrimination vs. Resentment (bottom row) across models and datasets. Yellow triangles are FNN, red circles are FMNN, blue stars are FR.

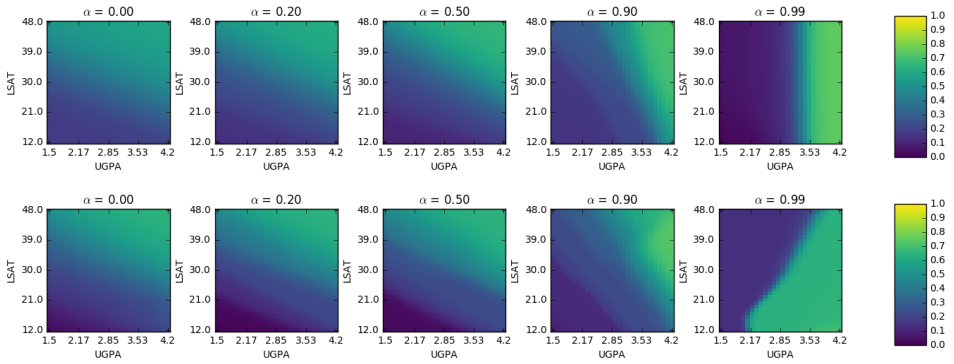

Figure 3.5: Plots of fitted solution for law school admissions data across range of α (fairness) levels, with unfairest left and fairest right. Top row: Monotonically fair classifier. Bottom row: Classifier with no monotonicity constraint. Lighter color indicates higher value.

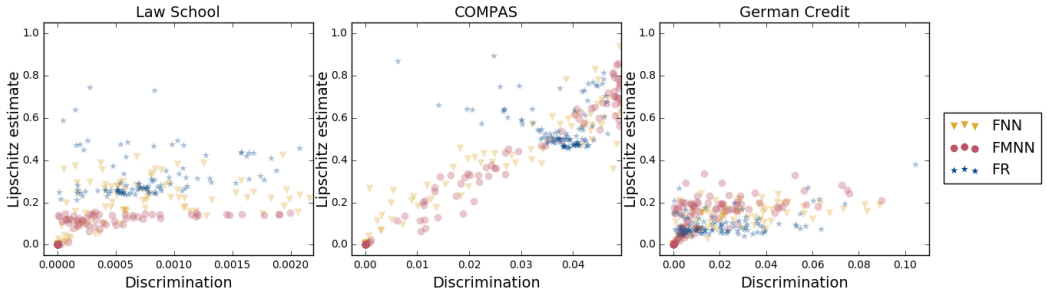

Figure 3.6: Lipschitz constant estimate vs. discrimination across models and datasets. Yellow triangles are FNN, red circles are FMNN, blue stars are FR.

Chapter 4

Elicited Monotonic Fairness

4.1 Overview

The monotonicity constraint discussed in Chapter 3 ensures that the outcome has a monotonic relationship with individual attributes conditioned on the other attributes. In practice, however, we often which to capture more complex definitions of "better" attribute sets that consider multiple attributes at once. For instance, consider the situation where two defendants are otherwise identical except that the first has committed ten more felonies and the second has committed one more misdemeanor; clearly, the first should be ranked as more likely to re-offend, but the two are incomparable according to these strict monotonicity rules. In addition, such strict interpretation of monotonicity prevents comparison between individuals with non-identical covariates on non-monotonic axes, i.e. when the condition of otherwise identical attributes doesn't hold. For example, if two defendants are 30 and 31 years old, they are incomparable.

More complex concepts of monotonicity are difficult to capture a priori. In this work, we explore the process of collecting and incorporating impartial arbiter information to ensure that individuals with commonly-accepted "better" attributes will receiver favorable outcomes while maintaining predictive. We do so by defining a loss function which depends only on the joint outcome of pairs so that arbiter comparisons can be combined with observed pairs of outcomes. We operate in a conditional setting so that fairness and accuracy can be balanced post hoc as desired. We also provide a means to incorporate group-level fairness to augment our individual-level protections.

This section explores a methodology for eliciting such non-axial monotonicity based on surveying arbiters, which may or may not be fair in their

judgments, and using those responses to regularize a classifier. The input of the arbiters is motivated as preventing intuitive resentment.

4.2 Introduction

We wish to extend the concept of *non-protected attribute resentment* (Def. 3.4.2) to widen the comparisons which can be made when defining what is a "better" X_u.

At its core, the problem to be addressed is that an individual will have resentment toward others if the individual feels the others received better treatment despite not being as deserving. Modeling each individual's views on relative treatment would be undesirable since we would like a rule which applies evenly to all individuals; we instead aim to learn a ranking function over individuals which can be used universally.

We propose to learn such a function by querying a set of arbiters by presenting pairs of non-protected attribute sets and asking for a judgment as to what the fair relative treatment of the pair would be. Once collected, these samples can be used to learn a *preference function* over pairs of non-protected attributes that captures the arbiters' notion of monotonic fairness. Similarly, we can use the rankings implied by the data to learn a similar function that captures ground truth. We learn these two functions jointly, in a conditional setting. The idea is that learning the two functions jointly will act as a regularizer on the model learned on ground truth data, pulling it towards the arbiter's orderings, and (by changing the conditional variable at prediction time) we can extrapolate between a more accurate prediction of ground truth, and a more faithful representation of the arbiters' orderings.

This is not the first work to incorporate the idea of utilizing arbiter queries in the area of fairness. The original definition of individual fairness given by Dwork Dwork et al. (2012) requires specifying a distance metric over attributes which can bound the difference in treatment, i.e. $D\left(f(X_i), f(X_j)\right) \leq \kappa d(X_i, X_j)$. IlventoIlvento (2019) and other recent works Jung et al. (2019); Lahoti et al. (2019); Wang et al. (2019) approach the problem of operationalizing an individual fairness distance metric by polling arbiters on which pairs

of individuals can be considered similar. Wang et al.Wang et al. (2019) similarly collect actual survey data and evaluate a variety of models to interpret such data. None of these approaches tackle the problem of dissimilar treatment, i.e. when an arbiter decides that two individuals should receive different predictions, especially the asymmetric case when arbiters indicate that one individual should receive a specifically more favorable or less favorable outcome than the other.

Models which are designed to identify the relative values of pairs can be classified as preference learning modelsPeters et al. (2018). The problem of preference learning centers around the paradox that individuals with strong preferences are often unable to systematize those preferences in useful way Lichtenstein and Slovic (2006). A variety of methods have been proposed, including several Bayesian approachesPeters et al. (2018); Guo et al. (2010) and neural network models Duman et al. (2019); Khannoussi et al. (2019) which propose methods of active preference elicitation. This work does not elaborate on the process of preference elicitation, i.e. the optimization of queries for information gain, but instead focuses the incorporation of a preference function in resentment prevention and fairness.

4.3 Model

First, we define out variables closely to those in Chapter 3. We assume $(X, A) \in \mathcal{D} \subset \Omega$ with corresponding binary output Y. Let X^{obs}, Y^{obs}, and A^{obs} be the non-protected attributes, observed binary outcomes $\in \{0, 1\}$, and protected attributes for some set of n individuals. We assume an arbiter that we can query with pairs X_i, X_j (or $(X_i, A_i), (X_j, A_j)$) and return an outcome $Z_{ij}^{arb} \in \{1, 2, 3, 4\}$, where the arbiters return 1 if it expects $f(X_i) > f(X_j)$, 2 if it expects $f(X_j) > f(X_i)$, 3 if it expects $f(X_i)$ and $f(X_j)$ to be similar, and 4 if it has no expectations as to the relative predictions. We will generally refer to those $Z_{ij}^{arb} \in \{1, 2\}$ as *dissimilar ratings* and those pairs $Z_{ij}^{arb} \in \{3\}$ as *similar ratings*.

We introduce an auxiliary variable Z_{ij}^{obs} which captures relationships of

Y_i, Y_j in the observed data with parallel meaning:

$$Z_{ij}^{obs} = \begin{cases} 1 & \text{if } Y_i^{obs} = 1 \text{ and } Y_j^{obs} = 0 \\ 2 & \text{if } Y_i^{obs} = 0 \text{ and } Y_j^{obs} = 1 \\ 3 & \text{if } Y_i^{obs} = Y_j^{obs} \end{cases} .$$

The crux of this model is moving from optimizing for the direct prediction of outcomes encompassed by Y to optimizing the relative outcomes encompassed by Z. When we survey our fairness arbiters, we ask them to evaluate whether one individual is more likely ($Z = 1$), less likely ($Z = 2$), or similarly likely ($Z = 3$) than another specific individual to have $Y = 1$.

We can then define a single loss function which incorporates both data sources:

$$\mathcal{L}_Z(Z, \hat{p}, \mathcal{Z}) = \sum_{(i,j)} \begin{pmatrix} \mathbf{1}_{Z_{ij}=1} & \log\left(\hat{p}_i(1-\hat{p}_j)\right) & + \\ \mathbf{1}_{Z_{ij}=2} & \log\left((1-\hat{p}_i)\hat{p}_j\right) & + \\ \mathbf{1}_{Z_{ij}=3} & \log\left(\hat{p}_i\hat{p}_j + (1-\hat{p}_i)(1-\hat{p}_j)\right) & \end{pmatrix},$$

letting $\mathcal{P}$ denote a set of pairs (i, j) of size $|Z|$.

In the above loss, the components for dissimilar outcomes can be reduced to the traditional log loss of the observed outcomes, i.e. if $Y_i^{obs} = 1$ and Y_j^{arb} then $\log\left(\hat{p}_i(1-\hat{p}_j)\right) = \log\left(\hat{p}_i\right) + \log\left(1-\hat{p}_j\right)$ is just the usual cross entropy loss. For the case that $Z_i j = 3$, the loss component can be viewed as pushing $\hat{p}_i$ and - $\hat{p}_j$ to take similar values; $\left(d\left(\hat{p}_i\hat{p}_j + (1-\hat{p}_i)(1-\hat{p}_j)\right)/d\hat{p}_i\right|_{\hat{p}_j} = 2\hat{p}_j - 1$, so that $\hat{p}_j$ is driven towards 0 if $\hat{p}_j < 1/2$ and towards 1 if $\hat{p}_j > 1/2$. This is symmetric, so that if both estimates are pushed towards the same pole, with an unstable equilibrium if both are $1/2$.

As mentioned above, we augment the neural network input with c, which acts as a conditional variable which is set to $c = 0$ when we wish to predict according to the observed data without concern for agreement with intuitive resentment, and which is set $c = 1$ when we wish to predict according to our arbiter data (and possibly a group fairness constraint) without concern for predictive accuracy according to the observed data. This design allows us to a tune a prediction continuously between being based entirely on real data without concern for agreeing with intuitive resentment and being based entirely on the arbiter data at the expense of predictive accuracy.

We can further add other losses to the prediction task when $c = 1$, e.g. a differentiable loss on a variant of group fairness like equality of outcome, odds, or opportunity. We assess this only on the predictions when $c = 1$ since that conditional setting corresponds to the "fair" setting; this effect of this loss can then also be balanced by setting $c \in (0, 1)$.

4.4 Experiments

We demonstrate the use of the above pairwise loss on two datasets. First, in Section 4.4.1 we consider a synthetic experiment without protected attributes where the true probability $\Pr(Y_i = 1|X_i)$ is known and we attempt to recover that probability using a simple feedforward neural network, as described in Sections 1.2.4 and 3.6, trained to minimize 4.3.

Second, in Section 4.4.2 we consider the COMPAS dataset, and utilize human survey responses and attempt to learn a network with a conditional prediction structure which allows for post-hoc compromise between fairness loss and prediction accuracy.

4.4.1 Synthetic - Proof of balancing objectives

We begin with a synthetic experiment where the ground truth is known. We have an individual set of attributes $X_i \sim N(0, 1)^2$, and two weight vectors, $\beta_{obs} = [0.9, 1.1]$ and $\beta_{arb} = [1.1, 0.9]$, which describe the relationship between X and, respectively, Z_{ij}^{obs} (via Y_i^{obs}) and Z_{ij}^{arb}. We set $P_i^{obs} = 1/(1 + \exp -X_i\beta^{obs} - 1)$, sample $Y_i \sim \text{Bernoulli}(P_i^{obs})$, and set Z_{ij}^{obs} as defined as above. We set Z_{ij}^{arb} according to:

$$Z_{ij}^{arb} = \begin{cases} 1 & \text{if} \quad X_i\beta^{arb} > X_j\beta^{arb} + 0.25 \\ 2 & \text{if} \quad X_j\beta^{arb} > X_i\beta^{arb} + 0.25 \\ 3 & \text{if} \quad |X_i\beta^{arb} - X_j\beta^{arb}| < 0.25 \end{cases} .$$

We sample 1,000 training examples of Z_{ij}^{obs} and 200 examples of Z_{ij}^{arb}, and evaluate losses on the same number of identically distributed held out samples. We trained using a small neural network of three hidden layers of width three

and a tanh activation function. We optimize for a compound loss,

$$\mathcal{L} = \underbrace{\mathcal{L}_Z\left(Z_{ij}^{obs}, \hat{p}_i = f(X_i, c=0), \mathcal{O}\right)}_{\mathcal{L}_Z^{obs}} + \underbrace{\mathcal{L}_Z\left(Z_{ij}^{arb}, \hat{p}_i = f(X_i, c=1), \mathcal{A}\right)}_{\mathcal{L}_Z^{arb}} + g(\theta)$$

where $\mathcal{O}$ is the set (or a subset) of pairs of observations, $\mathcal{A}$ is the set of arbiter-assessed pairs, and $g(\theta)$ is a regularization term on the parameters of the network. We arbitrarily set $g(\theta) = 0.01 * (||\theta||_1 + ||\theta||_2^2)$ to provide weak regularization of the network.

First, we wish to establish that the pairwise loss defined above can be used to estimate the probability function underlying probability function when $c = 0$ i.e. when attempting to predict based purely on the observed outcomes via the Z_{ij} pairs. In Figure 4.1, we show experimentally that $\hat{P}_i$ is accurate to within the limit of sampling error and (intentional) model misspecification.

Second, we wish to assess whether the model is able to interpolate via c between it's dual goals of predicting $\hat{Y}_i$ while adhering to the surveyed Z_{ij}^{arb} pairs. The trend of Z_{ij}^{arb} is exactly as expected; lowest when $c = 0$ and gradually increasing to a maximum when $c = 1$. The behavior of Z_{ij}^{arb} is less intuitive; it is highest when $c = 0$, but many random fits have an local minimum loss with $c < 1$. This is explained by the relatively small sample ($n^{arb} = 200$) leading to overfitting even in this modest network, and by $\mathcal{L}_Z^{obs}$ providing regularization which improves out-of-sample performance. We also, when examining the joint loss values available, that the models fits form appropriate trade off functions for performance, with reduction of one loss coming at the cost of increase of the other.

4.4.2 COMPAS

We augment the COMPAS dataset described in Section **??** in two ways: we add a feature for whether the current charge is violent, and we collect survey data to on random pairs and the possible ordering of their outcomes.

First, in adding the feature for violence of the current charge, we used the classification system described by ProPublicaLarson et al. (2016) and based on the US Department of Justice's definition of a violent crime: "murder and

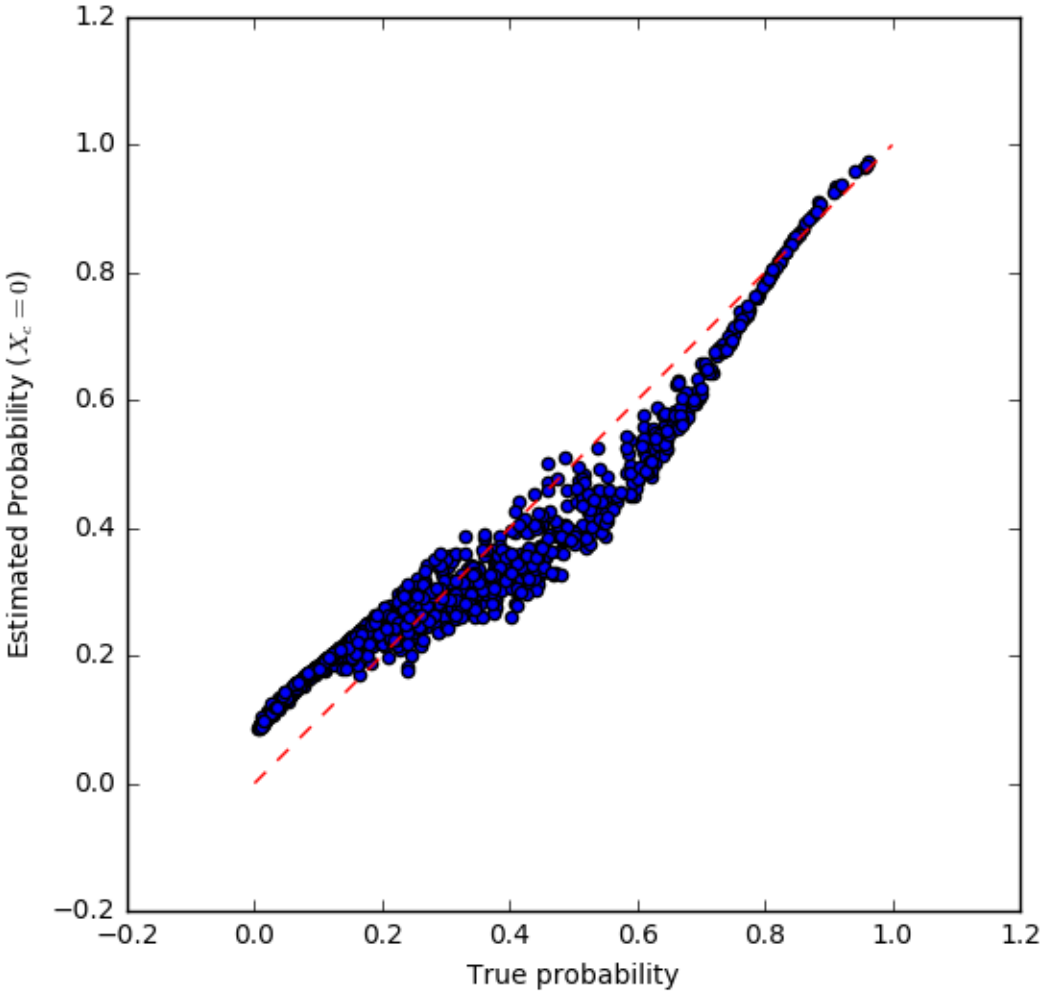

Figure 4.1: The model estimated probability of $\Pr(Y_i = 1|X_i, c = 0)$ versus the ground truth probabilities, with 1:1 line.

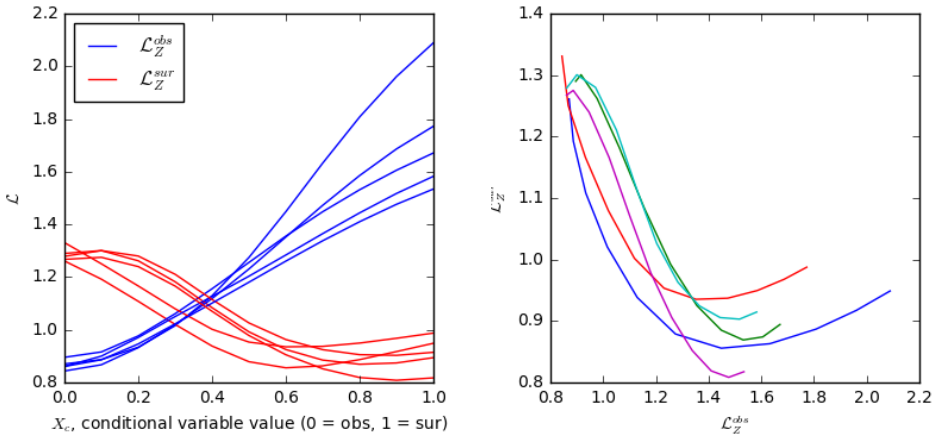

Figure 4.2: Model losses as a function of conditional variable (c) setting over 5 experiments with random initializations. Left: losses as a function of c, with $\mathcal{L}_Z^{obs}$ in blue and $\mathcal{L}_Z^{arb}$ in red. Right: parametric plot of $\mathcal{L}_Z^{arb}$ as a function of $\mathcal{L}_Z^{obs}$.

nonnegligent manslaughter, forcible rape, robbery, and aggravated assault." With this feature added, the non-protected attributes are age, (adult) priors count, juvenile prior felony count, juvenile prior misdemeanor count, juvenile prior other counts, arrest charge degree (felony or misdemeanor), and whether the arrest charge is violent. The protected attributes are race, classified as Caucasian, African American, or Other (comprising what the dataset labels as Other, Hispanic, Native American, and Asian), and Sex, classified as Male or Female.

The second augmentation was made by survey. Five volunteers were each presented 100 independent random pairings of non-protected attributes, i.e. excluding the protected attributes of race and sex. For each pairing, the two individuals were labeled "Individual A" and "Individual B", and the arbiter asked to provide one of four ratings:

- "A is at least as likely to (re)offend" ($Z = 1$)
- "B is at least as likely to (re)offend" ($Z = 2$)
- "A and B are similarly likely to (re)offend" ($Z = 3$)

- "No preference / any of the others are fair"

Seventeen responses indicating no preference were discarded, leaving 298 dissimilar responses ($Z \in \{1, 2\}$) and 185 similar responses ($Z = 3$).

We make no claim that the arbiters we surveyed have any qualification as unbiased judges; in the current setting where we do not provide protected attributes, the role of these arbiters is not to provide protected attribute-aware judgments. Instead, they are intended to provide feedback on what conditions they would find unfair via the individuals who should be more (or less) likely to get bail than others. This doesn't require expert knowledge because resentment occurs at an individual, non-expert level.

For group fairness loss, we chose to evaluate equality of odds, which requires that the prediction is independent of the protected attributes conditioned on the true outcome, i.e.

$$\Pr(\hat{Y} = 1|A = a, Y = y) = \Pr(\hat{Y} = 1|A = a', Y = y) \forall a, a', y.$$

We express this is a differentiable loss as

$$\mathcal{L}_F = \sum_y \sum_a \left(\bar{\hat{y}}_{ay} - \bar{\hat{y}}_{\cdot y}\right)^2$$

where $\bar{\hat{y}}_{ay} = \sum_{i:A_i=a,Y_i=y}(\hat{Y}_i)/n_{ay}$, i.e. the average prediction individuals of each protected attribute set and true outcome, and $\bar{\hat{y}}_{\cdot y} = \sum_{i:Y_i=y}(\hat{Y}_i)/n_{\cdot y}$, i.e. the average prediction for all individuals of that true outcome. Note that, as described above, we will always assess $\mathcal{L}_F$ using estimates $\hat{y}$ conditioned on $c = 1$, i.e. in the conditional setting where we care about fairness, both individual and group.

We then set our training loss, using the notation from the synthetic experiment above, as

$$\begin{aligned}\mathcal{L} = \; & \underbrace{\mathcal{L}_Z\left(Z_{ij}^{obs}, f(X_i, c = 0)\right)}_{\mathcal{L}_Z^{obs}} + \underbrace{\mathcal{L}_Z\left(Z_{ij}^{arb}, f(X_i, c = 1)\right)}_{\mathcal{L}_Z^{arb}} \\ & + \lambda_F \underbrace{\sum_{y \in \{0,1\}} \sum_{a \in \mathcal{A}} \left(\bar{\hat{y}}_{ay} - \bar{\hat{y}}_{\cdot y} | c = 1\right)^2}_{\mathcal{L}_F} + g(\theta)\end{aligned}$$

where λ_F is a parameter to weight $\mathcal{L}_F$ relative to $\mathcal{L}_Z^{arb}$.

Due to the increased data set size and complexity, we use a larger network in this problem with 4 hidden layers each of size 10. As long as we assume that axial monotonicity as discussed in Chapter 3 applies, we can train the present network using either a traditional feedforward neural network or the monotonic neural previously discussed; we present results from only the former to limit uncontrolled factors.

In Figure 4.3, we examine experimental results. Similar to our synthetic experiment, the pairwise loss on observed data $\mathcal{L}_Z^{obs}$ is lowest when $c = 0$ and increases steadily toward $c = 1$. The fairness loss $\mathcal{L}_F$ has a similar but opposite trend, and has an expected tradeoff curve with $\mathcal{L}_Z^{obs}$

An interesting anomaly, however, is the trend for the pairwise loss on arbiter data $\mathcal{L}_Z^{arb}$ (top right), which is highest at $c = 1$ when we would naively expect it to be lowest. Counterintuitively, this is explained by the fact that the arbiter ratings, which are assumed to be fair w.r.t. resentment but are in fact severely biased w.r.t group fairness. For instance, despite being unaware of race, arbiters' views on relative treatment led them to rate the non-protected attributes of African-Americans as more likely to re-offend than those of Caucasians on average; in the 105 ratings where they gave a dissimilar rating for an African-American and Caucasian pairing, they rated the African American defendant's non-protected attributes as more likely to re-offend 68 times (65%). The result of this contradiction in arbiter-based individual fairness and group fairness is predictable: one of them dominates when $c = 1$, and in this case it is group fairness, likely due to loss scaling.

Another factor causing $\mathcal{L}_Z^{arb}$ to be lower towards $c = 0$: the arbiters are actually fairly accurate. Of the 164 dissimilar ratings given, 128 ratings (78%) were directionally correct, which is comparable accuracy to a roughly three point difference on the COMPAS decile score system, despite COMPAS having access to additional attributes (both protected and other) that we didn't reveal to the arbiters. This relatively high accuracy of arbiter ratings allows them to act as additional information for training the structure of the classifer, improving the accuracy when $c = 0$ (and c *near* 0), even if no loss considers the accuracy of arbiter ratings when $c = 0$.

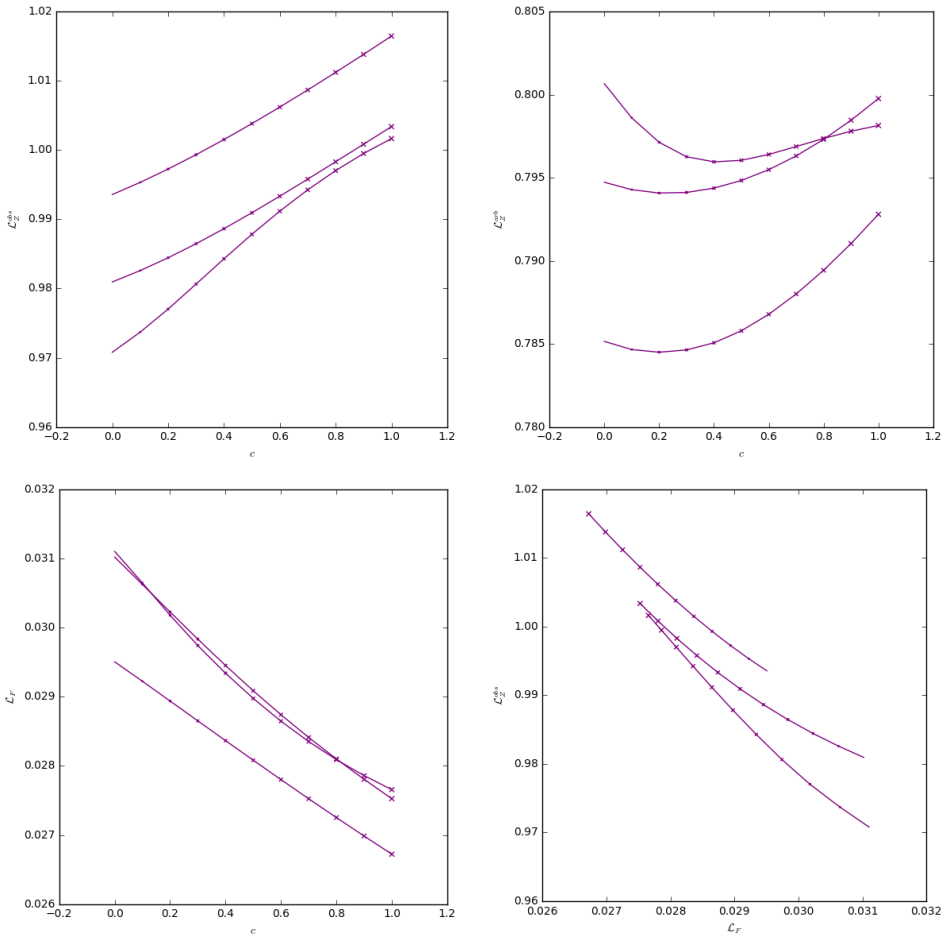

Figure 4.3: Clockwise from top left: Pairwise loss on observed data $\mathcal{L}_Z^{obs}$ as a function of c, pairwise loss on arbiter data $\mathcal{L}_Z^{arb}$ as a function of c, group fairness loss $\mathcal{L}_F$ as a function of c, and $\mathcal{L}_Z^{obs}$ as a function of group fairness loss $\mathcal{L}_F$. Lines represent three random training runs with $\lambda_f = 0.001$. Marker size is proportionate to c.

4.5 Discussion

The ability to incorporate preference learning into fairness models allows us to prevent individual resentment without a priori knowledge of what treatments would cause resentment. Such a priori knowledge is often hard to come by, requiring expert knowledge and a well structured problem. When such knowledge is available, it can still be difficult to systematize that a prior knowledge into a method for exact comparisons. We have shown a method for eliciting data informative of that a priori knowledge from arbiters via surveys, incorporating that data into a statistical model of preferences, and providing post hoc-tunable prediction which respects that arbiter input and, hopefully, prevents indvidiuals treated by the system from experiencing indvidual resentment.

The current work opens future research questions as well. While there are many methods available for preference function learning, they have not been widely integrated into the fairness literature. Integration of preference learning, and especially direct inference of the arbiter preference resulting in a well defined preference function, would open opportunities for an improved system with better guarantees. Although we have a loss function which can direct balance the observed and arbiter-provided data, we have not shown an easily-tuned system for incorporating more commonly used group fairness metrics. In addition, the current work is focused on neural networks and controlling violations of arbiter-provided ratings by loss penalization; it would be desirable to provide the hard guarantees non-resentment that is available for axial monotonicity.

Appendices

Appendix A

Appendix: Monotonic Fairness Supplement

In this supplement, we provide justification for our design choices for the neural network architecture, and demonstrate that such an architecture is able to capture monotonic functions, and impose monotonicity even when the true generating function is non-monotone.

A.1 Design choices

Below, we discuss several design choices, and their effect on the resulting functions.

Transformation Matters: The choice of transformation function in Equation 4 can have a significant effect on the probability of successful convergence of monotonic neural networks. We show in Figure A.1 that the choice of transformation can have different effects based on the nature of the underlying function, and affects both monotonic and non-monotonic fitting. We consider four non-linearities:

- Square: $\tau(x) = x^2$.
- Abs: $\tau(x) = |x|$.
- Offset exponential linear unit (elumod):
  $\tau(x) = \begin{cases} x & \text{if } x > 1 \\ e^{x-1} & \text{if } x \leq 1 \end{cases}$
- Softplus: $\tau(x) = \log(1 + e^x)$

We choose to use an offset exponential linear unit in our experiments, since it achieved optimal or near-optimal convergence in these comparisons.

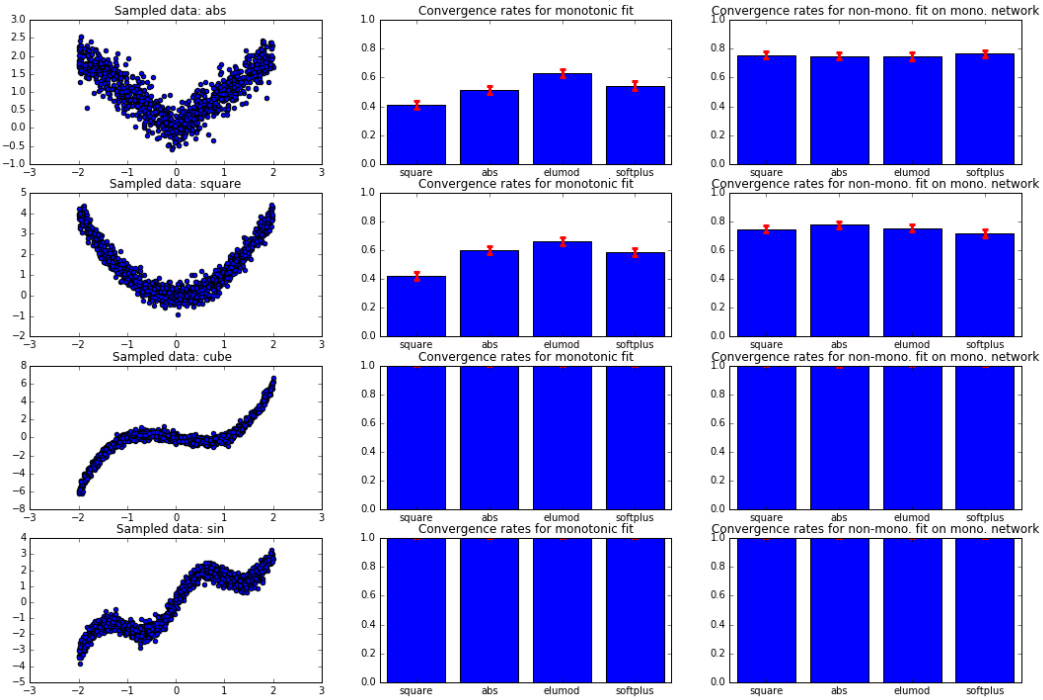

Figure A.1: Convergence rates for various functions used to enforce positive weights. The vertical exist for the middle and right columns is the proportion of random initialization which converge to a non-deviant ($\hat{y} = \bar{y}$) solution.

Activation Matters: Additional caution is needed in selecting an activation function for a monotonic neural network. If, for instance, a convex activation function is used (e.g. *elu* or *relu*), subsequent layers can only compound this convexity, and the resulting function can only be convex. It is easy to see this by considering the compounding of the first and second derivative across the layers. This may be a desirable feature in some settings, but generally prohibits it from approximating *any* monotonic function. As such, bounded (but monotonic) activation functions like *logistic* or *tanh* are advisable for general purposes.

A.2 Ability to Capture Mixed Monotonicity

We wish to emphasize that the network architecture described in this paper can simultaneously handle monotonic and non-monotonic relationships between the inputs and output. If we begin with the assumption that a network constrained to positive weights will produce a monotonically increasing function $f(x)$, we can briefly intuit the ability to fit a monotonically decreasing function by considering that $f(-x)$ would produce an identical function $f(x)$ but with reversed domain and therefore would be monotonically decreasing. Equivalently, we can enforce negativity on the weights in the network on edges leading out from any x with respect to which $f(x)$ is monotonically decreasing, i.e. set $\tilde{w} < 0$ in the connection between x and the first hidden layer (but keeping all weights in subsequent layers positive to maintain direction).

Further, if we accept that we can fit monotonically increasing and decreasing functions by constraining the weights, then consider what would happen if we fit $f(x, x)$, i.e. fed the same input twice, but constrained the first to be increasing and the second to be decreasing. By the argument of decomposing functions into positive and negative parts (or, here, decomposing the first derivative into positive and negative parts), we can construct a monotonic function from its increasing and decreasing parts. Further, each node in the first hidden layer would compute as $\sigma(\tilde{w}_+ x + \tilde{w}_- x + c)$, which could be simplified as $\sigma(wx + c)$ where w is unconstrained.

To demonstrate the result empirically, we show in Figure A.2 a two-dimensional experiment in which the true underlying function is non-monotonic

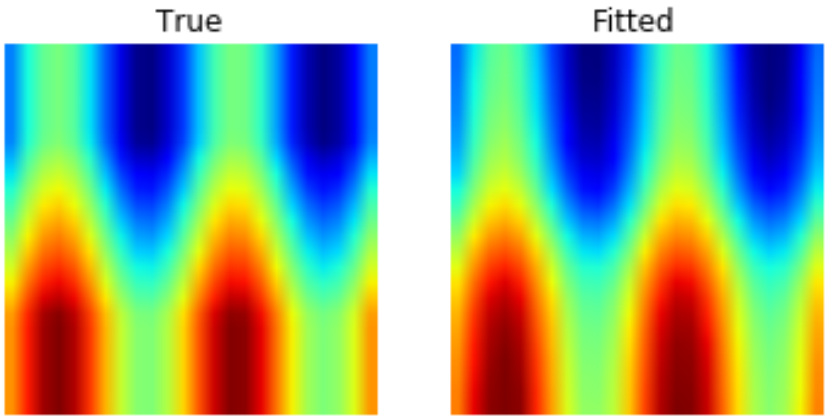

Figure A.2: Demonstration of our network architecture's ability to fit a function which is monotonic in one dimension and non-monotonic in another.

w.r.t to x_1 but strictly monotonically increasing w.r.t. x_2. Specifically,

$$f(x_1, x_2) = \sin(\pi x_1) + \max(-1, \min(1, x_2))$$

The estimated function shown is fit on a sample of 1,000 samples from the function and set to be non-monotonic w.r.t. x_1 and monotonic w.r.t x_2 and is able to recover the true function with reasonable precision.

Similarly, we show in Figure A.3 that a mixed-monotonicity function can be fit even if the underlying function is severely non-monotonic (with the expected error in fit). Here, $f(x_1, x_2) = x_0^2 + x_1^2$, and we again fit on a sample of 1,000 samples from the function and set to be non-monotonic w.r.t. x_1 and monotonic w.r.t x_2. As expected, it finds a function which is optimal subject to the (incorrect) constraints.

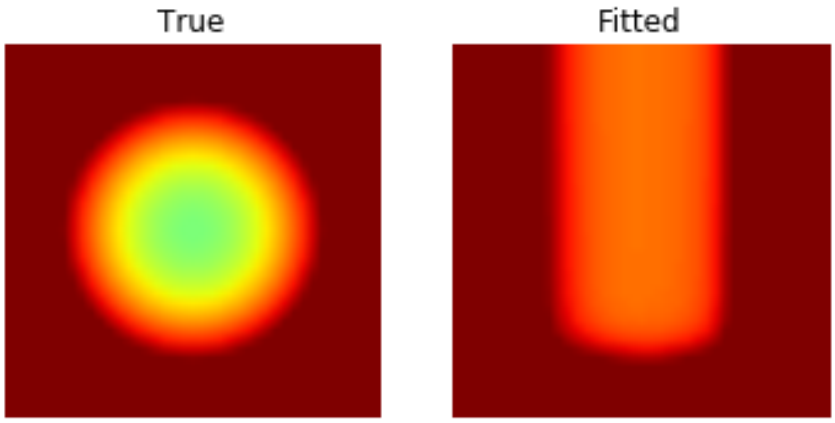

Figure A.3: Demonstration of our network architecture's ability to created a function which is monotonic in one dimension and non-monotonic in another, even when the data does not meet those qualifications.

Bibliography

Acharya, A., Teffer, D., Henderson, J., Tyler, M., Zhou, M., and Ghosh, J. (2015). Gamma process poisson factorization for joint modeling of network and documents. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. (2018). A reductions approach to fair classification. In *International Conference on Machine Learning*.

Airoldi, E., Blei, D., Fienberg, S., and Xing, E. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias: There's software used across the country to predict future criminals, and it's biased against blacks. *ProPublica*.

Balan, R., Singh, M., and Zou, D. (2018). Lipschitz properties for deep convolutional networks. *Contemporary Mathematics*, 706:129–151.

Balcan, M.-F., Dick, T., Noothigattu, R., and Procaccia, A. D. (2018). Envy-free classification. *arXiv:1809.08700*.

Berk, R., Heidari, H., Jabbari, S., Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. (2017). A convex framework for fair regression. *arXiv:1706.02409*.

Beutel, A., Chen, J., Zhao, Z., and Chi, E. H. (2017). Data decisions and theoretical implications when adversarially learning fair representations. *arXiv:1707.00075*.

Blei, D. and Lafferty, J. (2006). Dynamic topic models. In *International Conference on Machine Learning*.

Blei, D. and Lafferty, J. (2007). A correlated topic model of Science. *The Annals of Applied Statistics*, 1(1):17–35.

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

Bornstein, S. (2018). Antidiscriminatory algorithms. *Ala. L. Rev.*, 70:519.

Bouveyron, C., Latouche, P., and Zreik, R. (2018). The stochastic topic block model for the clustering of vertices in networks with textual edges. *Statistics and Computing*, 28(1):11–31.

Cai, D., Campbell, T., and Broderick, T. (2016). Edge-exchangeable graphs and sparsity. In *Advances in Neural Information Processing Systems*.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Cano, J.-R., Gutiérrez, P. A., Krawczyk, B., Woźniak, M., and García, S. (2019). Monotonic classification: an overview on algorithms, performance measures and data sets. *Neurocomputing*.

Cao, L. and Fei-Fei, L. (2007). Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *International Conference on Computer Vision*.

Caron, F. and Fox, E. (2017). Sparse graphs using exchangeable random measures. arXiv:1401.1137 [stat.ME].

Chang, J. and Blei, D. (2009). Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics*.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv:1511.07289*.

Court, M. D. (2014). Students for Fair Admissions, Inc. v. President and Fellows of Harvard College et al. 28(1:2014cv14176):1331.

Court, S. (1978). Regents of the University of California v. Bakke. 438(No. 76-811):265.

Court, S. (2013). Fisher v. University of Texas at Austin. 133(No. 11-345):2411.

Court, S. (2016). Fisher v. University of Texas at Austin. 136(No. 14-981):2198.

Crane, H. and Dempsey, W. (2016). Edge exchangeable models for network data. arXiv:1603.04571 [math.ST].

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Duman, G. M., El-Sayed, A., Kongar, E., and Gupta, S. M. (2019). An intelligent multiattribute group decision-making approach with preference elicitation for performance evaluation. *IEEE Transactions on Engineering Management*.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM.

Dwork, C., Immorlica, N., Kalai, A. T., and Leiserson, M. (2018a). Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, pages 119–133.

Dwork, C., Immorlica, N., Kalai, A. T., and Leiserson, M. (2018b). Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, pages 119–133.

Ensign, D., Friedler, S. A., Neville, S., Scheidegger, C., and Venkatasubramanian, S. (2018). Runaway feedback loops in predictive policing. In *Conference on Fairness, Accountability and Transparency*, pages 160–171.

for Education Statistics (Ed), N. C. (2013). The nation's report card: Trends in academic progress 2012. NCES 2013-456.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741.

Girolami, M. and Calderhead, B. (2011). Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. (2018). Regularisation of neural networks by enforcing Lipschitz continuity. *arXiv:1804.04368*.

Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Guo, S., Sanner, S., and Bonilla, E. V. (2010). Gaussian process preference elicitation. In *Advances in neural information processing systems*, pages 262–270.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323.

Heidari, H., Ferrari, C., Gummadi, K., and Krause, A. (2018). Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *Advances in Neural Information Processing Systems*, pages 1265–1276.

Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

Ilvento, C. (2019). Metric learning for individual fairness. *arXiv:1906.00250*.

Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. (2016). Fair algorithms for infinite and contextual bandits. *arXiv:1610.09559*.

Jung, C., Kearns, M., Neel, S., Roth, A., Stapleton, L., and Wu, Z. S. (2019). Eliciting and enforcing subjective individual fairness. *arXiv:1905.10660*.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer.

Kamishima, T., Akaho, S., and Sakuma, J. (2011). Fairness-aware learning through regularization approach. In *International Conference on Data Mining Workshops*, pages 643–650. IEEE.

Karrer, B. and Newman, M. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107.

Kearns, M., Roth, A., and Wu, Z. S. (2017). Meritocratic fairness for cross-population selection. In *International Conference on Machine Learning-Volume 70*, pages 1828–1836.

Kemp, C., Tenenbaum, J., Griffiths, T., Yamada, T., and Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *National Conference on Artificial Intelligence (AAAI)*, pages 381–388.

Khannoussi, A., Olteanu, A.-L., Labreuche, C., Narayan, P., Dezan, C., Diguet, J.-P., Petit-Frère, J., and Meyer, P. (2019). Integrating operators preferences into decisions of unmanned aerial vehicles: Multi-layer decision engine and incremental preference elicitation. In *International Conference on Algorithmic DecisionTheory*, pages 49–64. Springer.

Kim, S., Narayanan, S., and Sundaram, S. (2009). Acoustic topic model for audio information retrieval. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.

Lahoti, P., Gummadi, K. P., and Weikum, G. (2019). Operationalizing individual fairness with pairwise fair representations. *arXiv preprint arXiv:1907.01439*.

Larson, J., Mattu, S., Kirchner, L., and Angwin, J. (2016). How we analyzed the COMPAS recidivism algorithm. *ProPublica (5 2016)*, 9.

Leskovec, J., Lang, K., Dasgupta, A., and Mahoney, M. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123.

Lichman, M. (2013). UCI machine learning repository.

Lichtenstein, S. and Slovic, P. (2006). *The construction of preference*. Cambridge University Press.

Lipton, Z., McAuley, J., and Chouldechova, A. (2018). Does mitigating ml's impact disparity require treatment disparity? In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 8125–8135. Curran Associates, Inc.

Lohia, P. K., Ramamurthy, K. N., Bhide, M., Saha, D., Varshney, K. R., and Puri, R. (2019). Bias mitigation post-processing for individual and group fairness. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2847–2851. IEEE.

Louizos, C., Swersky, K., Li, Y., Welling, M., and Zemel, R. (2016). The variational fair autoencoder. In *International Conference on Learning Representations*.

Lum, K. and Isaac, W. (2016). To predict and serve? *Significance*, 13(5):14–19.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. (2018). Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*.

McCallum, A., Corrada-Emmanuel, A., and Wang, X. (2005). The author-recipient-topic model for topic and role discovery in social networks, with application to Enron and academic email. In *Workshop on Link Analysis, Counterterrorism and Security*.

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

Niu, Z., Hua, G., Gao, X., and Tian, Q. (2012). Context aware topic model for scene recognition. In *Computer Vision and Pattern Recognition*, pages 2743–2750.

on Civil Rights, U. C. (2018). Public education funding equity: In an era of increasing concentration of poverty and resegregation.

Peters, M., Saar-Tsechansky, M., Ketter, W., Williamson, S. A., Groot, P., and Heskes, T. (2018). A scalable preference model for autonomous decision-making. *Machine Learning*, 107(6):1039–1068.

Sill, J. (1998). Monotonic networks. In *Advances in Neural Information Processing Systems*, pages 661–667.

Snijders, T. and Nowicki, T. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2007). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Tu, Y., Johri, N., Roth, D., and Hockenmaier, J. (2010). Citation author topic model in expert search. In *ACL International Conference on Computational Linguistics*.

Veitch, V. and Roy, D. (2015). The class of random graphs arising from exchangeable random measures. arXiv:1512.03099 [math.ST].

Vidal Rodeiro, C. and Zanini, N. (2015). The role of the A* grade at A level as a predictor of university performance in the United Kingdom. *Oxford Review of Education*, 41(5):647–670.

Wang, H., Grgic-Hlaca, N., Lahoti, P., Gummadi, K. P., and Weller, A. (2019). An empirical study on learning fairness metrics for compas data with human supervision. *arXiv preprint arXiv:1910.10255*.

Wang, Y. and Wong, G. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19.

Wightman, L. F. and Ramsey, H. (1998). *LSAC national longitudinal bar passage study*. Law School Admission Council.

Williamson, S. (2016). Nonparametric network models for link prediction. *Journal of Machine Learning Research*, 17(202):1–21.

Wood, G. and Zhang, B. (1996). Estimation of the lipschitz constant of a function. *Journal of Global Optimization*, 8(1):91–103.

Xu, D., Yuan, S., Zhang, L., and Wu, X. (2018). Fairgan: Fairness-aware generative adversarial networks. In *IEEE International Conference on Big Data (Big Data)*, pages 570–575. IEEE.

Zafar, M. B., Valera, I., Rodriguez, M., Gummadi, K., and Weller, A. (2017a). From parity to preference-based notions of fairness in classification. In *Advances in Neural Information Processing Systems*, pages 229–239.

Zafar, M. B., Valera, I., Rodriguez, M. G., and Gummadi, K. P. (2017b). Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. (2013). Learning fair representations. In *International Conference on Machine Learning*, pages 325–333.

Zhang, H., Zhang, P., and Hsieh, C.-J. (2019). Recurjac: An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5757–5764.

Zhu, Y., Yan, X., Getoor, L., and Moore, C. (2013). Scalable text and link analysis with mixed-topic link models. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.