Minimal-Cost Intervention for Fairness

Yiyou Chen, Yizuo Chen, Siyan Dong, Haiying Huang
University of California, Los Angeles
{gerry99, yizuo.chen, siyand, hhaiying}@ucla.edu

Abstract

We aim to provide a simple, novel, and unified treatment for removing potential discrimination in a given causal model through deleting edges (a.k.a conditional intervention). The reduced model, which is then learned from data, is guaranteed to be bias-free. To represent the trade-off between fairness and accuracy, we associate each edge with a cost and define the task of **Minimal-Cost Intervention**, that is to find the cheapest edge set which would satisfy two goals after being deleted: 1) block every problematic path from protected attributes $\mathbf{P}$ (and its proxies) to target Y; 2) preserve dependence of the target on some essential features $\mathbf{U} \in \mathbf{X}$. First, we prove that Minimal-Cost Intervention is *NP-complete* when $\mathbf{U} \neq \emptyset$. Next, we develop an exact solver for this problem by encoding the constraints for fairness and preserving features into a W-MAXSAT problem. Finally, we develop a polytime approximate algorithm for this problem and evaluate its performance on some benchmark Bayesian Networks.

1 Introduction

Machine learning algorithms have been widely used in decision-making tasks in the human-based areas such as insurance [13], self-driving cars [4], criminal-justice forecast [1]. Recently, fairness has raised researchers' attention as unregulated ML systems may cause discrimination in their decisions. For example, a biased ML system may predict the cost of insurance based on attributes such as gender and race. Proposed in 2017, [11] offers a way to obtain fairness via causal interventions, where numerical constraints are derived to remove dependencies between discriminative variables (proxies) and some target variable.

In this work, we offer an alternative way to ensure fairness by removing direct cause-effect relationships that are represented as directed edges in a casual graph, which is licensed by causal interventions from [12], [9], [2]. In addition to a causal structure, we also assign costs for each edge to represent the cost of removal (intervention). The definition of costs is unrestricted; it may be preferences, mutual information, costs of intervention, etc.. In our *minimal-cost intervention* problem, we consider three types of variables: proxy variables, target variable, and essential variables, corresponding to variables that introduce discrimination such as visual features or names, variable that we want to predict such as car insurance cost, and variables whose dependencies with the target need to be preserved perpetually such as driving skill. Our goal is to minimize the total cost of edge removals while preserving the dependency between essential variables and the target.

In this paper, we study two versions we start with a brief background on fairness and causal interventions. We then show that a reduced version of the problem with no essential variable is equivalent to a minimal cut problem. We next prove that the problem with essential variable is *NP-complete* and can be formulated as a *weighted MaxSAT* (W-MAXSAT) problem. We then propose an approximate algorithm that provide a valid intervention edge set but may not give a minimal cost. We wrap up with experimental results that show the effectiveness of our algorithms.

2 Background

2.1 Fairness: from Observation to Causality

What does it mean for a model to be biased? This is a highly-debated question and in past years, researchers have proposed many different criterion. For example, let $\mathbf{X}$ be all features, A be the protect attributes, and Y be the target in the model. *Demographic parity* [3] requires Y to be independent of A, i.e. $\mathbb{P}(Y|A) = \mathbb{P}(Y)$. *Equality odds* [8] relaxes the condition and only requires Y to be independent of A given the true label Y', i.e. $\mathbb{P}(Y|A, Y') = \mathbb{P}(Y|Y')$, which means every social group $a \in \mathcal{A}$ will have the same misclassification rate. Nevertheless, recent studies [11] show that these observational criteria are inherently flawed: they treat the model as a black box and ignore the *causal* relationship among $\mathbf{X}, A, Y$ and proxies $\mathbf{P}$ for A. This can lead to paradoxical conclusion illustrated by Simpson's paradox as shown in [11].

Consider a causal model for the Berkeley admission process in Figure 1. When examining college-wise, men have a significantly higher admission rate than woman. Thus, a demographic parity test is tempting to conclude there is indeed a gender bias here. However, when examining individual departments, women are slightly favored over men in most departments. The explanation is that more women tend to apply for competitive departments. More specifically, variable G exerts influence on Y through two paths $G \to Y$ and $G \to D \to Y$. We want to quantify the former $G \to Y$, while the latter is acceptable since the path is blocked by a "resolving" variable like department choice D. This cannot be achieved without prior knowledge, i.e. causal graph of the data-generating process. With this motivation, [11] proposes two new criterion that measures discrimination with respect to a causal graph:

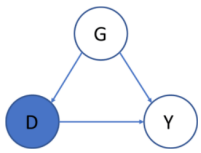

Figure 1: G is gender, D is department choice, and Y is admission result. D is a *resolving* variable

Let $\mathcal{G}$ be a causal graph. Let $\mathbf{R}$ be a set of *resolving* variables, which are safe to be included into decision process such as department choice. Let A be a protected variable. Let $\mathbf{P}$ be a set of *proxy* variables, which are indicative of A such as names.

Definition 1 (Unresolved discrimination)**.** *Decision Y exhibits Unresolved discrimination if there exists a direct path from A to Y in $\mathcal{G}$ that is not blocked by resolving variables in $\mathbf{R}$ and $Y \notin \mathbf{R}$.*

Definition 2 (Potential Proxy discrimination)**.** *Decision Y exhibits potential proxy discrimination if there exists a direct path from A to Y in $\mathcal{G}$ that is blocked only by proxy variables in $\mathbf{P}$ and $Y \notin \mathbf{P}$.*

A direct path from A to Y is safe when it is blocked by resolving variables. However, it is problematic if the direct path is unblocked, or blocked only by proxies $\mathbf{P}$ in which case a third test is needed:

Definition 3 (Proxy discrimination)**.** *A decision Y exhibit no proxy discrimination for proxy $\mathbf{P}$ if for all $\mathbf{p}_1, \mathbf{p}_2 \in \mathbf{P}$,*

$$\mathbb{P}(Y|\ do(\mathbf{P} = \mathbf{p}_1)) = \mathbb{P}(Y|\ do(\mathbf{P} = \mathbf{p}_2))$$

Note that the above definition is equivalent to saying that the target Y is independent of proxies $\mathbf{P}$ when we intervene on $\mathbf{P}$, i.e. $(Y \perp\!\!\!\perp \mathbf{P})_{\mathcal{I}(\mathbf{P})}$.

2.2 Interventions for Debiasing

[11] propose some preliminary methods for removing discrimination by doing interventions. We use X, $\texttt{Pa}(X)$, $\mathbf{Y}$, θ_X, $\Theta_{\mathbf{Y}}$, $\mathcal{M}(\mathcal{G}, \Theta)$ to denote a single variable, parents for variable X, a set of variables, parameters for variable X, parameter set for variables $\mathbf{Y}$, a causal model with causal structure (DAG) $\mathcal{G}$ and parameter set Θ, respectively.

Definition 4 (parametric intervention)**.** *Consider a causal model $\mathcal{M}(\mathcal{G}, \Theta)$. A parametric intervention on variables $\mathbf{Y}$ modifies Θ_Y to a new set of parameters $\Theta'_{\mathbf{Y}}$. The intervention results in a new model $\mathcal{M}'(\mathcal{G}, \Theta')$.*

The atomic intervention in [12] and conditional intervention in [9] involve changes on graph structures.

Definition 5 (atomic intervention)**.** *Consider a causal model $\mathcal{M}(\mathcal{G}, \Theta)$. An atomic intervention on variables $\mathbf{Y}$ sets each $X \in \mathbf{Y}$ to some deterministic value $C(X)$. Graphically, we remove all edges between $\texttt{Pa}(X)$ and X. The intervention results in a new model $\mathcal{M}'(\mathcal{G}', \Theta')$.*

Definition 6 (conditional intervention)**.** *Consider a causal model $\mathcal{M}(\mathcal{G}, \Theta)$. A conditional intervention on variables $\mathbf{Y}$ makes each $X \in \mathbf{Y}$'s parents to be some non-descendants of X $\mathbf{Z}$. Graphically, we set the parents of X to be $\mathbf{Z}$ instead of $\texttt{Pa}(X)$. The intervention results in a new model $\mathcal{M}'(\mathcal{G}', \Theta')$.*

It follows that the removal of edges in a causal model is licensed by both conditional intervention where new parents are subsets of original parents and parametric interventions where local CPTs are modified to adapt the conditional independence. We will denote an atomic intervention on variable set $\mathbf{X}$ as $\mathcal{I}(\mathbf{X})$ and conditional intervention that removes edge set $\mathbf{Z}$ as $\mathcal{I}_e(\mathbf{Z})$. In reality, interventions often involve policies and subjective judgements and thus are associated with costs. Let the intervention edge set, denoted $\mathbf{Z}$, be the set of removal edges, then we want to find a minimal-cost intervention edge set $\mathbf{Z}$ in $\mathcal{G}$ such that the proxies become independent to the target after removing edges in $\mathbf{Z}$.

3 Minimal-cost intervention on Causal Models

3.1 Problem Statement

Given a causal graph $\mathcal{G}$, there may exist different intervention edge set $\mathbf{Z}$ that block targets T from proxies $\mathbf{P}$. Since each edge is associated with a cost, we want to find $\mathbf{Z}$ such that the total cost is minimized. Moreover, we want to preserve the dependencies between some important features $\mathbf{U} \subseteq \mathbf{X}$ with the target after the interventions. Motivated by these two concerns, we propose a new problem setting: 1) can we find an intervention edge set $\mathbf{Z}$ that block target from proxies but preserve the dependencies between target and some important features? 2) if such intervention edge set exist, how to find $\mathbf{Z}$ that leads to minimum cost?

The *Minimal-Cost Intervention* problem we propose is characterized by a tuple $(\mathcal{G}, T, \mathbf{P}, \mathbf{U})$ where

- $\mathcal{G}$: a causal graph of data-generating process. Each edge e in $\mathcal{G}$ has a cost $\varphi(e)$.
- T: a target variable that we want to debias.
- $\mathbf{P}$: a set of protected variables and proxies that introduce bias.
- $\mathbf{U}$: a set of essential variables whose dependencies with the target are preserved after intervention $\mathcal{I}_e(\mathbf{Z})$ on $\mathcal{G}$.

Our goal is to find an optimal intervention edge set $\mathbf{Z}^*$ that optimizes the following

$$\begin{aligned} &\min_{\mathbf{Z}} \sum_{e \in \mathbf{Z}} \varphi(e) \\ &s.t.\ (T \perp\!\!\!\perp \mathbf{P})_{\mathcal{I}_e(\mathbf{Z}), \mathcal{I}(\mathbf{P})} \text{ and } \forall U \in \mathbf{U}.\ (T \not\perp\!\!\!\perp U)_{\mathcal{I}_e(\mathbf{Z})} \end{aligned} \tag{1}$$

3.2 More on Dependency

By keeping dependency between two nodes X and Y, we mean X is not *d-separated* from Y [5] according to the graphical structure $\mathcal{G}$. We provide some useful lemmas here for quickly checking such dependence. For simplicity, let $\texttt{Anc}(X)$ denote the ancestors of variable X (include X) and $\texttt{Path}(X, Y)$ denote a simple path from variable X to Y.

Theorem 1. *Without conditional variables, variable X and Y are dependent iff they have a common ancestor in $\mathcal{G}$, i.e. $X \not\perp\!\!\!\perp Y$ iff $\texttt{Anc}(X) \cap \texttt{Anc}(Y) \neq \emptyset$.*

Proof. Suppose $\texttt{Anc}(X) \cap \texttt{Anc}(Y) \neq \emptyset$, then exists $U \in \texttt{Anc}(X) \cap \texttt{Anc}(Y)$ such that there exist $\texttt{Path}(U, X)$ and $\texttt{Path}(U, Y)$. We immediately have $X \not\perp\!\!\!\perp Y$.

Suppose X and Y are dependent, then there exists an unblocked (dependency) path $\texttt{Path}(X, Y)$ [5]. If we check the valves in the path, there is no $A \to B \leftarrow C$ on the path since it will block the dependency. The path has to be $X \leftarrow \cdots \leftarrow Y$, $X \to \cdots \to Y$, or $X \leftarrow \cdots \leftarrow U \to \cdots \to Y$. In either case, we have $\texttt{Anc}(X) \cap \texttt{Anc}(Y) \neq \emptyset$. □

Corollary 1. *Without conditional variables, variable X and Y are dependent after an intervention on X iff there exists a direct path from X to Y in $\mathcal{G}$, i.e. $(X \not\perp\!\!\!\perp Y)_{\mathcal{I}(X)}$ iff $\texttt{Path}(X, Y)$ exists.*

Proof. After the intervention on X, X is disconnected with all its ancestors. Therefore, $\texttt{Anc}(X) = \{X\}$. From Yheorem 1, $X \not\perp\!\!\!\perp Y$ iff $X \in \texttt{Anc}(Y)$, which is equivalent to the existence of $\texttt{Path}(X, Y)$. □

4 Minimum Cut for $\mathbf{U} = \emptyset$

When no preserved node is introduced (i.e. $\mathbf{U} = 0$), the goal is to remove the dependency among the proxies and the target variable after intervening the proxies. Given a causal graph $\mathcal{G}$ associated with a cost for each edge, there is no dependency between a proxy node $p \in \mathbf{P}$ and the target T after proxy intervention on p if there is no directed path from p to T by Theorem 1. Therefore, a minimum cut algorithm naturally follows as a creation of a bipartite graph on $\mathcal{G}$ with p and T contained in different sub-graphs. Suppose we have multiple proxies, i.e. $|\mathbf{P}| > 1$, we simply add a dummy proxy node p' and directed edges (p', p) for all $p \in \mathbf{P}$ with edge weights set to ∞. One may verify that the minimum cut of p' and T gives the exact solution to the original problem.

The minimum cuts can be found using graph algorithms that have been well developed such as Edmonds–Karp algorithm and Ford–Fulkerson algorithm. The removal of edges is licensed by conditional intervention defined in Definition 6. The modification of $\mathcal{G}$ then leads to a removal of proxy discrimination introduced by $\mathbf{P}$.

5 NP-Completeness for $\mathbf{U} \neq \emptyset$

Minimal-Cost Intervention problem clearly belongs to NP when $\mathbf{U} \neq \emptyset$ since we can check the dependencies among essential variables, proxies, and the target variable within polynomial time with an algorithm like *Dependent Set* (Algorithm 2). To prove NP-hardness, we show the following polynomial-time reductions:

$$\textbf{SET-COVER} \leq_P \textbf{DAG-CPMC} \leq_P \textbf{DAG-OMC} \leq_P \textbf{MIN-COST INTERVENTION}$$

We precisely define each class of problems:
SET-COVER: given a set of elements P and a set of subsets T with weights $W : T \to \mathbb{Z}$, find a subset $K \subset T$ such that $\bigcup K = P$ while $\sum_K W(K)$ is minimized.
DAG-CPMC (DAG-Connectivity Preserving Min-Cut): given a source s_1, a destination t, and a third node s_2, find the minimum weighted cut on s_1 and t so that the connectivity of s_1 and s_2 is preserved after the cut.
DAG-OMC (DAG-Orphan Min-Cut): given a source s, a destinition t, and a third orphan node u (no incoming edges), find the minimum weighted cut on s and t so that the connectivity of u and t is preserved after the cut.
MIN-COST INTERVENTION: given a set of proxies $\mathbf{P}$, a target variable T, and a set of essential variables $\mathbf{U}$, find the minimum weighted cut on $\mathbf{P}$ and T so that the dependency between $\mathbf{U}$ and T is preserved.

Please see Appendix 2 for the complete proof.

6 Constrained Optimization with W-MAXSAT

Equation 1 is a constrained optimization problem that is NP-complete to solve as shown in Section 5. We next offer an exact inference algorithm that finds an optimal intervention edge set $\mathbf{Z}^*$ by formulating the optimization problem as a W-MAXSAT problem.

We first write constraints in boolean logic form. From Corollary 1 and Theorem 1, $(T \perp\!\!\!\perp \mathbf{P})_{\mathcal{I}_e(\mathbf{Z}), \mathcal{I}(\mathbf{P})}$ iff there is no directed path from $\mathbf{P}$ to T after the removal of edge set $\mathbf{Z}$, and $(T \not\perp\!\!\!\perp \mathbf{U})_{\mathcal{I}_e(\mathbf{Z})}$ iff $\texttt{Anc}(T) \cap \texttt{Anc}(U) \neq \emptyset$ for all $U \in \mathbf{U}$. Both cases require a search of directed paths between two variables.

Note that Algorithm 1 in Appendix B takes polynomial time complexity but exponential space complexity. To find all directed paths from $\mathbf{P}$ to T we simply call Algorithm 1 for each $P \in \mathbf{P}$ and

T. To find all unblocked dependency paths between some $U \in \mathbf{U}$ and T, we simply loop through all nodes X in $\mathcal{G}$ and enumerate all possible $\texttt{Path}(X, U)$ and $\texttt{Path}(X, T)$. The implementation is shown in Algorithm 2 in Appendix B.2.

From now on, let $\texttt{pset}(P, T)$ denote the set of all directed paths from $P \in \mathbf{P}$ to T and $\texttt{uset}(U, T)$ denote the set of all unblocked dependency paths between $U \in \mathbf{U}$ and T. Moreover, we assume that each path is represented as a union of edges in $\mathcal{G}$.

We next introduce the encoding of the constrained optimization problem to the W-MAXSAT problem. From [5], W-MAXSAT problem is defined as follows.

Definition 7 (W-MAXSAT)**.** *Consider a CNF $\alpha_1, \ldots, \alpha_m$ with weights for each clauses $w(\alpha_1), \ldots, w(\alpha_m)$ over Boolean variables $X_1, \ldots, X_n$. We want to find a truth assignment $x_1, \ldots, x_n$ that maximizes the following weight*

$$Wt(x_1, \ldots, x_n) = \sum_{x_1, \ldots, x_n \vDash \alpha_i} w(\alpha_i)$$

Recall that for each edge $e \in \mathcal{G}$, $\varphi(e)$ is the cost of removing e from $\mathcal{G}$. Let M be a very large number that represents ∞. then we can encode Equation 1 into a W-MAXSAT problem in following way.

- For each edge $e \in \mathcal{G}$, we assign indicator I_e and $I_{\bar{e}}$ to represent whether or not e is removed from $\mathcal{G}$. When $I_e = \texttt{True}$, edge e is kept; when $I_e = \texttt{False}$, edge e is removed. We can write a CNF using indicators only
$$\Pi_1 = \bigwedge_{e \in \mathcal{G}} I_e$$
We assign weight $w(I_e) = \varphi(e)$ for each indicator in the first cluster.
- We next a logical expression that satisfies $(T \perp\!\!\!\perp \mathbf{P})_{\mathcal{I}_e(\mathbf{Z}), \mathcal{I}(\mathbf{P})}$. We want every directed path from $\mathbf{P}$ to T to be disconnected, i.e. at least one edge must be removed on each directed path from $\mathbf{P}$ to T. We can therefore write the following expression
$$\Pi_2 = \bigwedge_{P \in \mathbf{P}} \bigwedge_{p \in \texttt{pset}(P,T)} \bigvee_{e_i \in p} \neg I_{e_i}$$
Note that the logic expression is already in CNF form. Since we want the constraint to always be satisfied, we assign $w(\alpha_i) = M$ for each clause $\alpha_i \in \Pi_2$.
- We can then write a logical expression that satisfies $(T \not\perp\!\!\!\perp \mathbf{U})_{\mathcal{I}_e(\mathbf{Z})}$. For each $U \in \mathbf{U}$, we want at least one unblocked dependency path between U and T. Therefore, there exists at least one path in $\texttt{uset}(U, T)$ where none of the edges are removed. we can write the following expression
$$\Pi_3 = \bigwedge_{U \in \mathbf{U}} \bigvee_{p \in \texttt{uset}(U,T)} \bigwedge_{e_i \in p} I_{e_i}$$
Note that Π_3 is not in CNF form, but simple rules can be applied to convert Π_3 into CNF form Π_3'. Again, since we want the constraint to always be satisfied, we assign $w(\beta_i) = M$ for each clause $\beta_i \in \Pi_3'$.

It is evident that if we solve above W-MAXSAT problem, we have all rules in Π_2 and Π_3 satisfied. The optimal truth assignment on the indicators is also guaranteed to maximize $\sum_{e \in \mathcal{G}} \varphi(e)$, which is equivalent to minimizing $\sum_{e \in \mathbf{Z}} \varphi(e)$ where $\mathbf{Z}$ is an intervention edge set consistent with the edge indicators, i.e. $e \in \mathbf{Z}$ iff $I_e = \texttt{True}$.

7 Approximation Algorithm

In Section 7, we have discussed an exact exponential time solution W-MAXSAT to perform minimal-cost intervention with $\mathbf{U} \neq \emptyset$. Here, we try to come up an approximation algorithm with heuristics that runs in polynomial time. Note that to make sure two variables u, v have dependencies after intervention, we only need to make sure there exist a path from $k \in \texttt{Anc}(u)$ to u and a path from k to v after the minimal weighted cut. To preserve such a path after the cut, we set the edge cost on the path to ∞. Our approximation algorithm follows from finding a heuristic to evaluate the importance of paths, from which k is chosen. Since edges with larger costs are less likely to be cut, possible heuristics naturally follow:

- Maximal min cost: given u, v, find $k \in \texttt{Anc}(u)$ and paths $\texttt{Path}(k, u)$, $\texttt{Path}(k, v)$ that maximize the minimal edge cost in $\texttt{Path}(k, u) \cup \texttt{Path}(k, v)$.
- Maximal total cost: given u, v, find $k \in \texttt{Anc}(u)$ and paths $\texttt{Path}(k, u)$, $\texttt{Path}(k, v)$ that maximize the total edge cost in $\texttt{Path}(k, u) \cup \texttt{Path}(k, v)$.

The approximation algorithm consists of four steps:
1. Given a causal graph $(\mathcal{G}, T, \mathbf{P}, \mathbf{U})$, create a new undirected graph $\mathcal{G}' = \mathcal{G}|_{\texttt{Anc}(T) \cup \texttt{Anc}(\mathbf{U})}$, a subgraph of $\mathcal{G}'$ on $\texttt{Anc}(T) \cup \texttt{Anc}(\mathbf{U})$ with edge costs $\varphi(e') = \varphi(e) - \max_{e \in E} \varphi(e) - 1$. This makes sure all edges have negative weights so finding largest-weight path in step 2 won't cause infinite loop.
2. For each preserved node $u \in \mathbf{U}$, find the optimal paths $\texttt{Path}(k, u)$ and $\texttt{Path}(k, T)$ on $\mathcal{G}'$ using Bellman-ford or Dijkstra algorithm, optimizing a heuristic such as "maximal min cost" or "maximal total cost."
3. For each preserved node $u \in \mathbf{U}$, set costs of edges in $\mathcal{G} \cap \texttt{Path}(u, T)$ to ∞.
4. Run the minimal weighted cut algorithm on $\mathcal{G}$ as in Section 4.

We provide the pseudocode for step 2 in Appendix B.3.

8 Experimental Results

We evaluate the performance of the minimal cut algorithm from Section 4 and the approximate algorithm from Section 7 by conducting experiment on a simple *insurance* model [1]. The insurance is designed to evaluate car insurance risks. The model contains 27 variables and 52 edges and the graph structure is shown in Figure 4. We also assigned costs to each edge based on our personal preferences. Note that those costs are not fixed and modifications can be made based on use cases. In this experiment, we consider variable *SocialEcon* be a proxy $\mathbf{P}$, *PropCost* as the target T, and *DrivingSkill* as a essential variable $\mathbf{U}$. In other words, we do not want *SocialEcon* to cause bias on our prediction of *PropCost*. However, we should always allow *DrivingSkill* to affect *PropCost*. Our algorithm needs to find a minimal-cost intervention edge set $\mathbf{Z}$ that removes the discrimination on *PropCost* caused by *SocialEcon* while the dependency between *PropCost* and *DriveQuality*.

We first sampled 5,000 examples from the true distribution provided by the bnlearn repository. We then run parameter learning and inferences on different causal structures [2]. The parameter learning procedure follows Maximum Likelihood Estimates (MLE) and learns the Conditional Probability Tables (CPTs) that maximizes the likelihood given the observed data instances. We then then run probabilistic inference on the learned BN model via variable elimination [10], etc.. In this example, we are particularly interested in the query $\mathbb{P}(T|\mathbf{U})$, or $\mathbb{P}(\textit{PropCost}|\textit{DrivingSkill})$; we will, however, also record $\mathbb{P}(\textit{PropCost}|\textit{CarValue})$ to get a sense of how much effect on probabilistic inference is caused by edge removals.

We first run parameter learning and probability queries on the original graph (without any interventions). The results should provide a brief view of the dependencies among our interested variables. The results are shown in Table 1.

PropCost	*DrivingSkill*			*CarValue*				
values	SubStandard	Normal	Expert	5K	10K	20K	50K	100K
Thousand	0.36	0.53	0.51	0.49	0.48	0.48	0.45	0.31
TenThou	0.32	0.31	0.31	0.33	0.33	0.27	0.29	0.32
HundredThou	0.22	0.11	0.12	0.12	0.13	0.17	0.18	0.24
Million	0.09	0.05	0.05	0.05	0.06	0.07	0.08	0.13

Table 1: Each entry represents a conditional probability on *PropCost*.

We next assumes that we do not have essential variable and only removes the bias brought by *SocialEcon*. The minimal-cost intervention edge set $\mathbf{Z}$ can be found by applying a minimal-cut

[1] The insurance model is selected from bnlearn repository `https://www.bnlearn.com/bnrepository`. Variables are defined at `https://www.bnlearn.com/documentation/man/insurance.html`

[2] We used bnlearn library for Python `https://pypi.org/project/bnlearn` for parameter learning and inferences.

algorithm[3] as shown in Section 4. The diagram is shown in Figure 5 with red marks on removed edges. It is evident that after removing the edges, *PropCost* becomes independent of *SocioEcon* when we intervene on *SocioEcon*, which implies that no bias is caused by *SocioEcon*. The results for probability queries are recorded in Table 2. Unfortunately, the edge removals in fact caused PropCost to be independent of *DrivingSkill*, which is not what we want. This will be fixed with the approximate algorithm shown in Section 7.

PropCost	*DrivingSkill*			*CarValue*				
values	SubStandard	Normal	Expert	5K	10K	20K	50K	100K
Thousand	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
TenThou	0.30	0.30	0.30	0.30	0.30	0.30	0.30	0.30
HundredThou	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Million	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06

Table 2: Probability queries without considering the dependency between *DrivingSkill* and *PropCost*.

In order to guarantee the dependency between target and essential variable, we apply the approximate algorithm shown in Section 7. The algorithm returns a sound intervention edge set that may not necessarily be minimal. See Figure 6 for a diagram that shows the intervention edge set. In this case, *SocioEcon* does not add bias to *PropCost* while *PropCost* depends on *DrivingSkill*, which shows the soundness of the algorithm. Nevertheless, the solution may not be optimal; finding a minimal-cost intervention edge set that keeps certain dependencies is NP-complete as shown in Section 5. Both *DrivingSkill* and *CarValue* are dependent with *PropCost* as shown in Table 3. If we compare Table 1 and 3, we see very small change on probability queries. On the other hand, Table 2 presents a very different condition distribution from the other two tables.

PropCost	*DrivingSkill*			*CarValue*				
values	SubStandard	Normal	Expert	5K	10K	20K	50K	100K
Thousand	0.36	0.53	0.51	0.49	0.48	0.49	0.45	0.31
TenThou	0.32	0.31	0.31	0.34	0.33	0.27	0.29	0.32
HundredThou	0.22	0.11	0.12	0.12	0.13	0.17	0.18	0.24
Million	0.10	0.05	0.05	0.05	0.05	0.07	0.08	0.13

Table 3: Probability queries with dependency between *DrivingSkill* and *PropCost*.

9 Conclusion

In this paper, we studied methods to debias both with and without essential variables. Our experimental results support our theoretical development, which shows the sufficiency of achieving fairness via conditional interventions (edge deletions). Our technique translates the task of debiasing to graph-based problems which are widely studied and more adaptive. This gives a novel perspective on debiasing and lights up future directions: in addition to introducing costs on conditional intervention as we studied, one may also study cost models and the robustness of our method.

10 Github link

Github link to our project: https://github.com/arthurhaiying/CS260-Causal-Fairness.git

References

[1] Richard Berk. *Criminal justice forecasts of risk: A machine learning approach*. Springer Science & Business Media, 2012.

[2] Biao Qin. "Differential Semantics of Intervention in Bayesian Networks". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

[3] We used NetworkX library [7] for implementation of minimum cut algorithm.

[3] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. "Building classifiers with independency constraints". In: *2009 IEEE International Conference on Data Mining Workshops*. IEEE. 2009, pp. 13–18.

[4] Mike Daily et al. "Self-driving cars". In: *Computer* 50.12 (2017), pp. 18–23.

[5] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[6] Qi Duan and Jinhui Xu. *On the Connectivity Preserving Minimum Cut Problem*. 2013. arXiv: 1309.6689 [cs.DS].

[7] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

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

[9] Juan D. Correa and Elias Bareinboim. "A Calculus for Stochastic Interventions: Causal Effect Identification and Surrogate Experiments". In: *34th AAAI Conference on Artificial Intelligence*. 2020.

[10] Nevin Lianwen Zhang and David Poole. "A simple approach to Bayesian network computations". In: *the Tenth Biennial Canadian Artificial Intelligence Conference (AI-94)*. 1994, pp. 171–178.

[11] Niki Kilbertus et al. "Avoiding Discrimination through Causal Reasoning". In: *Advances in Neural Information Processing Systems*. 2017.

[12] Judea Pearl. *Causality: Models, Reasoning and Inference*. 2nd ed. 2009.

[13] Riya Roy and K Thomas George. "Detecting insurance claims fraud using machine learning techniques". In: *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE. 2017, pp. 1–6.

A Appendix: Minimal-Cost Intervention is NP-Hard

Theorem 2.

SET-COVER $\leq_P$ ***DAG-CPMC*** $\leq_P$ ***DAG-OMC*** $\leq_P$ ***MIN-COST INTERVENTION***.

Proof. **SET-COVER**$\leq_P$ **DAG-CPMC**: this is a slight modification of the reduction used in [6] which applies to general graphs. In the following illustration, we consider $S = \{x_1, x_2, x_3\}$ and $T = \{A_1, A_2, A_3\}$ where $A_1 = \{x_1, x_3\}, A_2 = \{x_2, x_3\}, A_3 = \{x_1, x_2\}$.

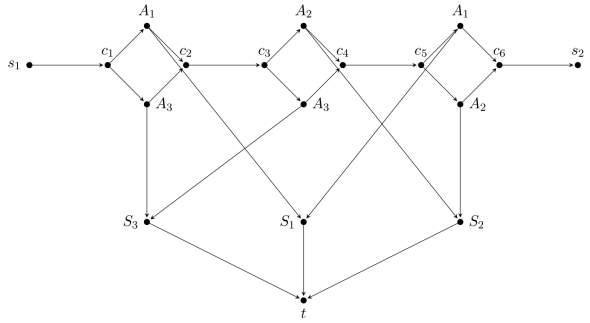

Figure 2: **SET-COVER**$\leq_P$ **DAG-CPMC**

For each variable $x \in P$, we create a gadget like $\{c_1, A_1, c_2, A_3\}$ where c_{2i-1}, c_{2i} are dummy nodes and A_j are the sets where each variable belongs to. We add a direct edge from c_{2i-1} to each A_j with weight 1, as well as a edge from each A_j to c_{2i} with weight 1. We then add direct edges $(s_1, c_1), (c_2, c_3), ..., (c_{2i}, c_{2i+1}), ..., (c_{2|P|}, s_2)$ with weights ∞. Next, we connect all set variables A_j to a single S_j for each j with a direct edges of weights ∞. Finally, we connect each S_j to t with direct edges of weights $W(A_j)|P||T|$. Note that there's a one-to-one correspondence between the set of (S_j, t) in the cut and the set cover, and the monotonicity is guaranteed by the weight assignments. In addition, the entire cut is uniquely determined once the cut among (S_j, t) is determined. Therefore, we showed

$$\textbf{SET-COVER}(P, T) = S \iff \textbf{DAG-CPMC}(s_1, s_2, t) = CUT|_{S \subset \{(S_j, t)\}}$$

which concludes **SET-COVER**$\leq_P$ **DAG-CPMC**.

SET-COVER$\leq_P$ **DAG-CPMC**$\leq_P$**DAG-OMC**: after a polynomial-time reduction a graph problem as shown in figure 2 to set cover, we can further reduce it from a **DAG-OMC** problem within polynomial time by reversing the edge directions (shown in figure 3):

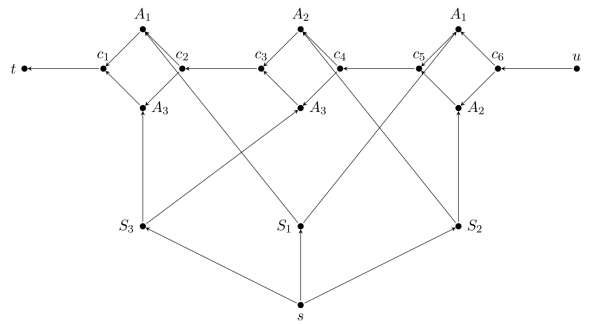

Figure 3: **SET-COVER**$\leq_P$ **DAG-CPMC**$\leq_P$**DAG-OMC**

To preserve the connectivity between s_1 and s_2 after the cut is equivalent to preserve the connectivity between t and u where u is the orphan node. It's also easy to see that two problems give the exact

same minimum weighted cut. Therefore,

$$\textbf{DAG-CPMC}(s_1, s_2, t) \iff \textbf{DAG-OMC}(s, t, u)$$

which concludes **SET-COVER**$\leq_P$ **DAG-CPMC**$\leq_P$**DAG-OMC**.

SET-COVER $\leq_P$ **DAG-CPMC** $\leq_P$ **DAG-OMC** $\leq_P$ **MIN-COST INTERVENTION**: we notice that a **DAG-OMC** problem that can reduce to set cover is a sub-case of **MIN-COST INTERVENTION** where $|\mathbf{P}| = |\mathbf{U}| = 1$. Moreover, when $U \in \mathbf{U} = \{U\}$ has no parents, it has dependency with T if and only if there's a directed path from U to T. Combining the results from Section 4, there's an equivalence relation

$$\textbf{DAG-OMC}(s, t, u) \iff \textbf{MIN-COST INTERVENTION}(\mathbf{P} = \{P\}, T, \mathbf{U} = \{U\})$$

which concludes the proof of

$$\textbf{SET-COVER} \leq_P \textbf{DAG-CPMC} \leq_P \textbf{DAG-OMC} \leq_P \textbf{MIN-COST INTERVENTION}.$$

Therefore, **MIN-COST INTERVENTION** is NP-Complete. □

B Pseudocode for Algorithms

B.1 Finding Directed Paths

Algorithm 1 Search for all directed paths between two nodes in $\mathcal{G}$

procedure SEARCH_DIRECTED_PATH($u :=$ start node, $v :=$ end node, $adj[.] :=$ array where $adj[x]$ returns the set of children of x in $\mathcal{G}$, $\Gamma :=$ cache where $\Gamma[x]$ records the set of all directed paths between x and v)
    if $u = v$ **then**
        return $\{\{v\}\}$
    end if
    if $\Gamma[u]$ is defined **then**
        return $\Gamma[u]$
    end if
    $\Gamma[u] = \emptyset$
    for each children $c_i \in adj[u]$ **do**
        for each path p in SEARCH_DIRECTED_PATH(c_i, v, adj) **do**
            $p' = \{u\} \cup p$
            $\Gamma[u] = \Gamma[u] \cup \{p'\}$
        end for
    end for
    return $\Gamma[u]$
end procedure

B.2 Finding Unblocked Dependency Paths

Algorithm 2 Search for all unblocked dependency paths between two nodes in $\mathcal{G}$

procedure SEARCH_UNBLOCKED_PATH($U \coloneqq$ a preserved node, $T \coloneqq$ target node, $\mathbf{V} \coloneqq$ all nodes in $\mathcal{G}$, $\Sigma \coloneqq$ set of unblocked dependency paths between U and T)
    $\Sigma = \emptyset$
    for each $X \in \mathbf{V}$ **do**
        $\Sigma_1 = \text{SEARCH\_DIRECTED\_PATH}(X, U, adj, \emptyset)$
        $\Sigma_2 = \text{SEARCH\_DIRECTED\_PATH}(X, T, adj, \emptyset)$
        for each $p_1 \in \Sigma_1$ **do**
            for each $p_2 \in \Sigma_2$ **do**
                $p' = p_1 \cup p_2$
                $\Sigma = \Sigma \cup \{p'\}$
            end for
        end for
    end for
end procedure

B.3 Step 2 of Approximate Algorithm

Algorithm 3 Longest path from u to Y on $\mathcal{G}'$ using Bellman-Ford with "maximal min cost."

procedure BELLMAN-FORD-MAX-MIN-COST($\mathcal{G}', u, T$)
    $\Sigma(V) = \emptyset$
    queue $\mathcal{Q} = \{u\}$.
    while $\mathcal{Q} \neq \emptyset$ **do**
        $v = \text{front}(\mathcal{Q})$
        $\mathcal{Q} = \mathcal{Q} \backslash \{v\}$
        for each $p \in \texttt{Pa}(v)$ **do**
            if $\min(\texttt{Weight}(v), \varphi(v, p)) > \texttt{Weight}(p)$ **then**
                $\texttt{Weight}(p) = \min(\texttt{Weight}(v), \varphi(v, p))$
                $\Sigma(p) = \Sigma(v) \cup (p, v)$
                $\mathcal{Q} = \mathcal{Q} \cup \{p\}$
            end if
        end for
        for each $c \in \texttt{Children}(v)$ **do**
            if $\min(\texttt{Weight}(v), \varphi(v, c)) > \texttt{Weight}(c)$ **then**
                $\texttt{Weight}(c) = \min(\texttt{Weight}(v), \varphi(v, c))$
                $\Sigma(c) = \Sigma(v) \cup (v, c)$
                $\mathcal{Q} = \mathcal{Q} \cup \{c\}$
            end if
        end for
    end while
    return $\Sigma(T)$
end procedure

C Appendix: Figures

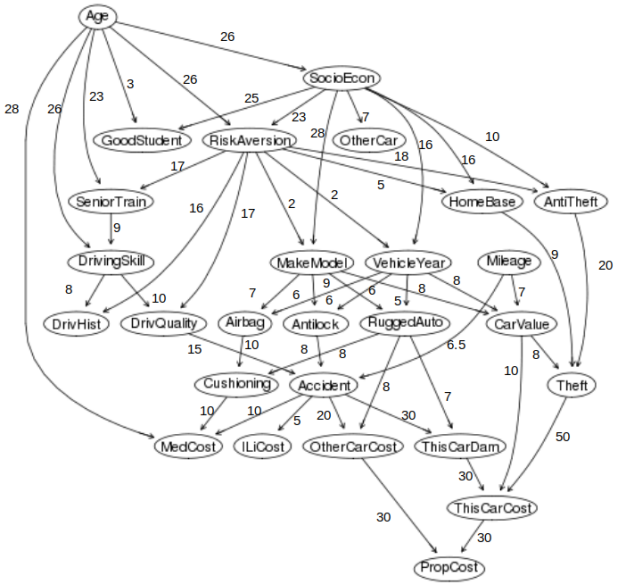

Figure 4: Insurance BN diagram

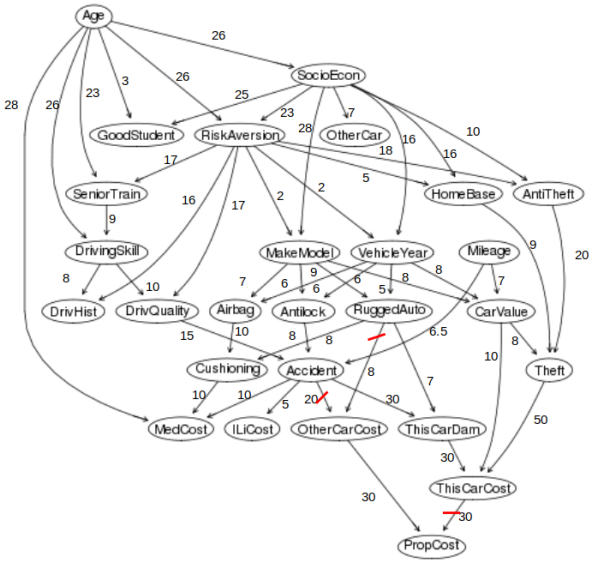

Figure 5: BN Diagram with red marks on removed edges.

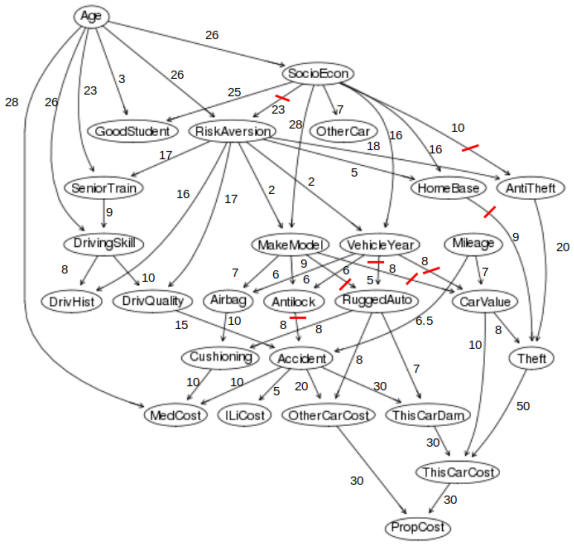

Figure 6: BN Diagram with red marks on removed edges.