Causal Feature Finding

Matthew Laws mdl4@williams.edu
Williams College Computer Science

1. Introduction

When engaging in machine learning, practitioners must consider various factors beyond accuracy when constructing a model. Some key concerns are data availability and computational resources. Firstly, ensuring that there is a sufficient amount of data to train a model so it can accurately predict outcomes is crucial. Secondly, considering the limitations of compute power, especially when access to high-performance hardware like GPUs is limited, is essential for constructing models that can be trained within a reasonable time-frame.[1]

Fortunately, these concerns can be mitigate by using a subset of the available features to train a model. For instance, consider a scenario where a practitioner is worried about the cost of collecting more data for each feature, such as in the case of medical tests. By creating an optimal subset of features, the practitioner can focus on collecting additional data for only that subset, simplifying the data collection process. Similarly, for someone looking to replicate an experiment in a different setting, using a subset of features makes data collection for follow-up experiments easier. Additionally, using a subset of features reduces the number of parameters that need to be optimized in the network, which can significantly improve training time.

In summary, this project demonstrates that deploying causal inference to guide feature selection in machine learning can enhance predictive power.

1.1 Past Work

Past work has looked at different methods for reducing dimensionality of data through a variety of methods. Regularization, especially L1 regularization developed by Tibshirani (1996), is one of the simplest and most widely used ideas to impose lower dimensionality on the data. Regularization, however, is learned during training thus still requires a sufficiently large amount of data. Other research in the field of Causal Inference has aimed to identify causal implications of the outcome for feature selection prior to training Guyon et al. (2007), Yu et al. (2020), and Yu et al. (2022) are examples; however I will try to implement my own framework.

1.2 My Contribution

In this paper I demonstrate the following guarantee: Causal Inference can be used to find a subset of observed features that performs strictly better than a randomly chosen subset.

1. There are more things to think about, but these are motivating examples for this project.

Additionally I provide a framework for conducting this process. This framework allows a practitioner facing constraints in data availability or computational resources to focus on collecting and training only the most relevant data for their model.

2. Preliminaries

2.1 Machine Learning

Machine Learning (ML) is a broad term that describes the use of computers to make predictions based on data. The basic ML pipeline involves taking input data, passing it through an ML model, and obtaining an output, such as a prediction. In order for the model to produce accurate outputs, it needs to be trained. During training, the goal is to find the values for θ that minimize the loss function. Formally:

$$\text{argmin}_\theta \big[L(Y, \hat{Y})\big]$$

Where Y is the vector of true outcomes and $\hat{Y}$ are the predictions given θ. The solution to this problem depends on the model but one way to solve is by treating it as an optimization problem an optimizer.[2] Other structures have different methods of minimizing loss.

2.2 Machine Learning Models

In this project, I explored five distinct machine learning models: Logistic Regression, Decision Trees, Boosted Decision Trees, Bagged Decision Trees, Random Forests, and Neural Networks. A brief description of each is provided in appendix A.1.

2.3 Causal Graphical Models

A causal graphical model illustrates the relationships among observed variables within a system. This model encodes statistical information into a graph, facilitating data analysis. My framework accommodates two types of graphical models: causal Directed Acyclic Graphs (DAGs) and causal Acyclic Directed Mixed Graphs (ADMGs). Other graphical models, such as Partially Directed Acyclic Graphs (PDAGs) and Partial Ancestral Graphs (PAGs), are produced by various discovery algorithms. For more information on PDAGs and PAGs, refer to Appendix A.2.

2.3.1 Directed Acyclic Graphs

DAG is suitable for encoding causal relationships when there is no unmeasured confounding among the observed variables. Unmeasured confounding occurs when an unobserved variable causally influences two or more observed nodes. The process of encoding information into a DAG is relatively straightforward. For each pair of variables X and Y, consider both directions and ask: "Is X a potential cause of Y?" If the answer is yes, a di-

2. Some example optimizers are: Gradient Decent, Stochastic Gradient Decent, Adagrad, RMSProp, and Adam.

rected edge is added from X to Y (X $\rightarrow$ Y). It's important to note that the absence of an edge in a DAG is a stronger assumption than the presence of one, as it asserts there **cannot** be a causal relationship between the two variables. The resulting graph must be acyclic and include all directed edges. While this process typically requires substantial domain knowledge, it can be simplified with causal discovery techniques, as described in Section 2.4.

2.3.2 Acyclic Directed Mixed Graphs

ADMGs extend the concept of DAGs by allowing bidirected edges in addition to directed edges. Bidirected edges are a simple way to encode unmeasured confounding between observed variables as shown in Figure 1. Constructing an ADMG follows a similar process to that of a DAG, but with an additional consideration for potential unmeasured confounding between variables. For each pair of vertices X and Y, one must ask both if X potentially causes Y and if there is potential unmeasured confounding between X and Y.[3] Figure 1 illustrates the latent projection operator for converting a DAG with unmeasured confounding in an ADMG. Here, U represents an unmeasured variable that causes both A and Y, and this relationship is captured by the bidirected edge. As with DAGs, the absence of an edge in an ADMG indicates a stronger assumption than its presence, asserting the definitive absence of a causal relationship. The resulting graph should be acyclic and consist only of directed and bidirected edges.

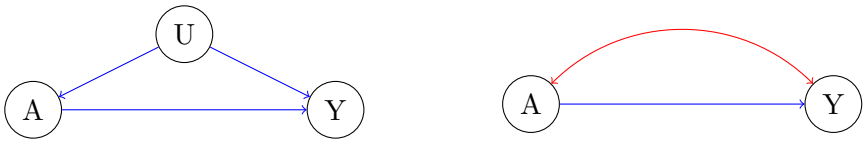


Figure 1: Latent Projection of DAG with unmeasured confounder U into an ADMG.

2.3.3 Intervention Graphs

The fundamental design of intervention graphs is that when some node A is intervened on, all outgoing edges from that node are disconnected from their original source and instead come from the intervention node. This is because the natural state of A no longer has a downstream causal effect; instead, the causal effect arises from the result of the intervention. Furthermore, all downstream nodes from the intervention become potential outcomes, given that A has been intervened on.

An example of an intervention graph is depicted in Figure 2, where the boxed letter represents the intervention. Only the outgoing edges are disconnected because backdoor effects, which are effects that cause A, remain unchanged.

3. Both can be possible.

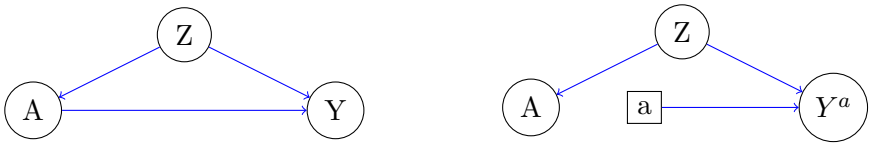


Figure 2: Standard Graph and Intervention Graph after intervening on A.

2.3.4 D-Separation and M-Separation

D-separation and m-separation are critical concepts for determining independence and valid adjustment sets in causal graphical models. D-separation defines three different types of connections between three nodes: a fork ($A \leftarrow B \rightarrow C$); a chain ($A \rightarrow B \rightarrow C$);[4] or a collider ($A \rightarrow B \leftarrow C$). Two nodes X and Y are d-separated if all paths between them are **blocked**.[5] A path can be blocked in two ways. A path is considered blocked if there are any chains or forks that are blocked on the path. A chain or fork is blocked by conditioning on the "middle" variable, B in the examples above. The other condition that blocks a path is a closed collider. Adjusting for nothing, a collider is closed; however, a collider can be "opened" by conditioning on the "colliding" variable (B) or any of its descendants. M-separation is a generalization of d-separation for ADMGs. It follows the same rules as d-separation but extends the definition of a collider to include three types of triples: $A \rightarrow B \leftarrow C$, $A \rightarrow B \leftrightarrow C$, and $A \leftrightarrow B \leftrightarrow C$.

2.4 Causal Discovery

Causal Discovery refers to a collection of algorithms that can uncover the causal structure of data solely from observation. The discovery process relies on three key assumptions: falsification, selection, and faithfulness. Falsification states that if the data determines $X \not\perp\!\!\!\perp Y$ but a graphical model suggests otherwise, the model should be rejected. Faithfulness asserts that there are no spurious independencies in the data. Finally, selection states that if we find $X \perp\!\!\!\perp Y$ then under faithfulness the edge between X and Y is truly absent. Because the data in binary, in this project I employ the Chi-Squared Independence test to determine independence. For more details, refer to McHugh (2013). The specific discovery algorithms used in this project are PC and FCI.

2.4.1 PC Algorithm

The PC algorithm, introduced by Spirtes et al. (2001), is a causal discovery algorithm that takes data as input and outputs a PDAG (section A.2.0.1). The outputted PDAG is guaranteed to be markov equivalent to the true DAG.[6]

The PC algorithm consists of two main phases: skeleton discovery and orientation. In the skeleton discovery phase, the algorithm determines if two variables are independent of each other given a set of other variables (the separating set). It begins with an empty

4. This is reversible.
5. A path is a sequence of edges connecting two nodes with no vertex or edge being repeated.
6. Markov equivalence refers to having the same adjacencies (edges without considering edge type).

separating set and considers all connected variables as potential separators. If two variables are independent given the connecting nodes, there should not be a direct edge between them, and any such edges are removed. This process iterates until the skeleton of the true DAG is obtained.

The next step is the orientation phase which involves 4 rules. The first rule, rule 0, orients all unshielded colliders. An unshielded collider occurs when the skeleton contains a triple, A – B – C, (where there is no edge between A and C) and B is not in the separating set for A and C which means that B must be a collider.[7] After rule 0 is applied, rules 1-3 are applied recursively until no more edges can be oriented. rules 1-3 require previously oriented edges and orient remaining edges in a way that prevent cycles and new independencies from appearing. See figure A.1 for more detailed description of rules 1-3.

2.4.2 FCI Algorithm

The FCI (Fast Causal Inference) Algorithm is similar to the PC algorithm but with one significant difference: while the PC algorithm assumes no unmeasured confounding between observed features, FCI does not make this assumption. Like PC, FCI is guaranteed to identify the correct skeleton for the data. FCI outputs a Partial Ancestral Graph (PAG), as described in section A.2.0.2.

FCI has two main phases: a skeleton discovery phase similar to the PC algorithm,[8] and an orientation phase. In the orientation phase, all edges start as o-bidirected edges and all unshielded colliders are oriented. Then a system of rules is applied repeatedly to orient as many edges as possible. In this algorithm these could lead to edges being oriented as bidirected. Similar to the PC algorithm, the goal of these rules is to avoid introducing cycles and maintain observed independencies. The full rules can be found in the work by Spirtes et al. (2001) on pages 188 and 183.

3. Data

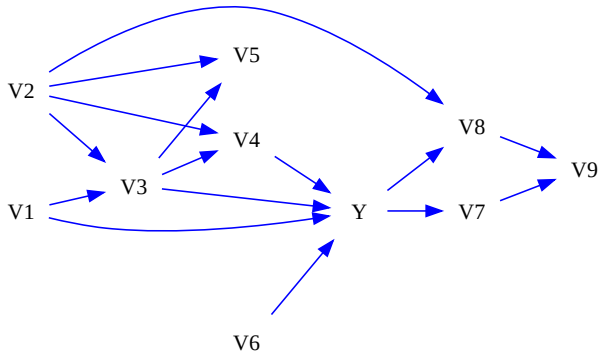
For my project, I used a mix of synthetic and real-world data. Synthetic data demonstrates the methodology's effectiveness when the true causal graph is known and enables analysis of its resilience to distribution shifts. Additionally, I tested the methodology on real data from the National Institute of Diabetes and Digestive and Kidney Diseases (Smith et al., 1988).

3.1 Description

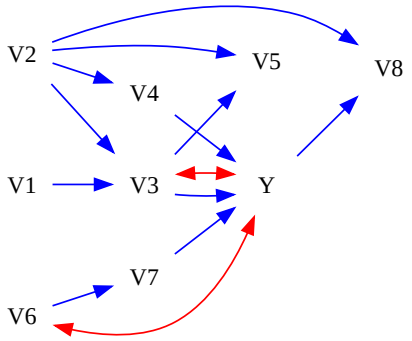
3.1.1 Synthetic Data

For the random data, I generated a DAG over observed variables $\boldsymbol{V} = \{V_1, \ldots, V_9\}$ and some outcome Y given in figure 3a. Each variable's data is generated as a function of its parents. Explicitly the data for some vertex V_i is modeled by $[V_i = f(Pa_{\mathcal{G}}(V_i)) + \text{noise}]$ If

7. otherwise A and C would be dependent through the fork / chain of B.
8. Edges are deleted when nodes are determined to be independent given some separating set.



(a) Causal DAG underlying the synthetic data.



(b) Causal ADMG underlying the mixed synthetic data.

Figure 3: Causal Structures for the Synthetic Data. Also represents the function graph that generates the data.

$Pa_{\mathcal{G}}(V_i) = \varnothing$ then V_i is generated at uniform random on range [0, 1].[9] The data is then binarized by setting values above the column mean to 1 and below to 0.

For distribution shift, I simulated a scenario where the outcome variable Y affects different populations (distributions) differently.[10] To illustrate this, I changed the function f that generates a variable that is a descendant of Y. I chose V_8.[11] I also considered scenarios where the data is non-representative of the wider population. For instance, the real-world dataset was collected only from women, so other genders may not follow the same distribution.[12] To simulate this, I modified the generating function for an upstream variable, V_2, to see if the ideal subset remains resilient.[13]

3.1.2 Synthetic Data with Unmeasured Confounding

To enhance the effectiveness of my framework on real-world data, I developed a method for handling unmeasured confounding. I created a synthetic ADMG with unmeasured confounding between certain variables and the outcome, depicted in Figure 3b. Data corresponding to this ADMG is generated similarly to before, but the unmeasured variables will not be included in the dataset.

3.1.3 Real World Data

The real world data I am using is from a study on 768 patients, all female and of Pima Indian heritage. The dataset includes columns for the outcome (binary: has diabetes or doesn't have diabetes) and 8 observed variables: number of pregnancies, glucose level, blood pressure, skin thickness, insulin level, BMI, age, and diabetes pedigree function.[14]

9. The generating function f was artificially created for each node, some are linear and others are not.
10. for example changing COVID symptoms over time.
11. I just used a random number generator.
12. Gender is generally considered an upstream variable because very few things cause gender.
13. Again I used a random number generator to choose V_2.
14. A measure of predisposition to diabetes that includes measures such as family history.

Using the PC algorithm for causal discovery, I generated a causal DAG over the observed variables, as shown in Figure 4. This DAG allows me to calculate the causal effect that each variable has on the outcome.

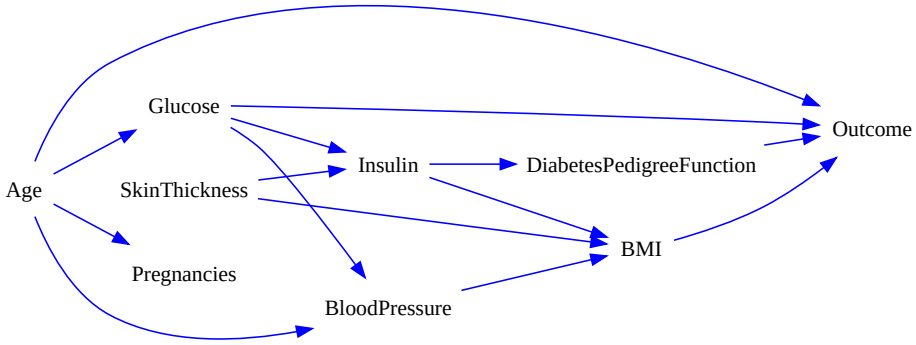


Figure 4: Underlying causal DAG for Diabetes data from PC discovery.

3.1.4 Adaptability

More than specific analysis of a given dataset, this project aims to provide a framework for determining the best causal features for any dataset. The code provides an easy-to-use framework for importing new data / causal graphical models for analysis.

3.2 Obstacles

There are two primary challenges in calculating the causal effects within this framework. Firstly, edge orientation is essential; however, both the PC and FCI algorithms do not guarantee complete orientation. While I provide a program to allow the user orient unoriented edges, this approach can lead to causal misspecification if edges are oriented incorrectly. Unmeasured confounding also presents a significant obstacle. When attempting to determine the effect of a variable V on Y but $V \leftrightarrow Y$ there is never a valid backdoor set because a path from V to Y always exists through the bidirected edge. To address this issue, we must turn to alternative methods for calculating the causal effect such as front-door and instrumental variable as outlined by [Pearl (1995)] and [Wright (1928), Balke and Pearl (2011), and Angrist et al. (1996)]. As illustrated in Figure 3b, there is confounding between V_6 and Y as well as V_3 on Y. I will delve into the calculation of these effects in sections 4.2 and 4.3 respectively.

4. Identification Formulas

In this section, I present the formulas used to calculate the Average Causal Effect (ACE) of each feature on the outcome and provide a metric for ranking these effects. Depending on the feature's position in the graph different evaluation metrics are required to calculate the effect. As our outcome is binary, the predicted ACE will fall between -1 and 1, repre-

senting the increase or decrease in the probability of predicting true when the feature is true.

4.1 Augmented Inverse Probability Weighting (AIPW)

AIPW is a doubly robust formula that predicts both the backdoor adjustment and the Inverse Probability Weighting (IPW) formulas and combines the results. Robins et al. (1994) showed that AIPW will converge to the true ACE if either the backdoor formula or IPW is correct.[15] AIPW is my primary method of predicting the ACE because it works as long as there is no unmeasured confounding between the feature and the outcome.

4.1.1 Backdoor Criterion

Since AIPW builds upon backdoor adjustment, for AIPW to be valid it must satisfy the backdoor criterion. A set Z satisfies the backdoor criterion if, in the causal DAG $\mathcal{G}$, no elements of Z are decedents of the treatment A ($Z \cap De_{\mathcal{G}}(A) = \varnothing$) and $A \perp\!\!\!\perp Y \mid Z$ in the intervention graph $\mathcal{G}^a$. If Z satisfies the backdoor criterion it is said to be a **valid backdoor adjustment set**.

4.1.2 AIPW Functional

$$\textbf{Let } \mathbb{E}[Y^{a_x}] = E\left[\frac{\mathbb{1}(A = a_x)}{p(A = a_x \mid Z)} \cdot (Y - \mathbb{E}[Y \mid A, Z]) + \mathbb{E}[Y \mid A = a_x, Z]\right]$$
$$\textbf{Then } \text{ACE} = E[Y^{a_1}] - E[Y^{a_0}]$$

In the binary case Y^{a_1} refers to the outcome of Y if A was set to 1 by intervention, potentially contrary to fact. Y^{a_0} is the same but if A was set to 0. The AIPW equation can be broken down into a couple parts. First $\mathbb{1}(A = a_x)$ is just an indicator function for if A — the true value of A for this instance — is 1. $p(A = a_x \mid Z)$ is the propensity score for the instance — how likely is it that $A = a_x$ given the conditions (Z). This effect is generally unknown and needs to be estimated. To estimate this value I fit a logistic regression model to the data with the task of predicting A given Z.[16][17] The next part is Y which refers to the actual value of the outcome. Finally there is $\mathbb{E}[Y \mid A, Z]$ and $\mathbb{E}[Y \mid A = a_x, Z]$. Similar to the propensity these values must be estimated. To estimate them I train a new logistic regression fit on the data tasked is to predict Y given the conditions Z and the value of A the treatment. This model can be used to predict $\mathbb{E}[Y \mid A, Z]$ by using it on the original instance, and can also be used to predict $\mathbb{E}[Y \mid A = a_x, Z]$ by generating a modified dataset where all values in A are set to a (1 or 0).

15. We also proved it in class :)
16. An offset term was added to the set Z for better fitting of the model.
17. Often the propensity can get very small and lead to exploding results; however, since the classification was binary I did not find this to be an issue that needed to be handled.

4.2 Front-door Adjustment

Front door adjustment is used to calculate the causal effect of A on Y in the presence of unmeasured confounding between the two variables as seen in figure 5. I implemented an IPW version of front-door adjustment to estimate the ACE.[18] The key idea is to identify a mediator set M that fully mediates the effect of A on Y.[19] Then, by combining the probabilities of M causing Y and A causing M, we can estimate the probability of A causing Y.

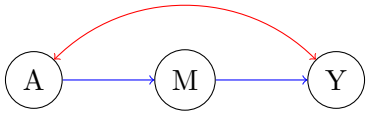


Figure 5: Example of when front-door adjustment is used.

4.2.1 Front-door Criterion

For a front-door adjustment to be valid the front-door criterion must be met. For some mediating set M and some observed confounders C The front door criterion has three rules for calculating the effect of A on Y:

1. All causal paths from A to Y intersect M
2. C is a valid backdoor adjustment set for the effect of A on M
3. $C \cup \{A\}$ is a valid backdoor adjustment set for the effect of M on Y

4.2.2 Front-door IPW Functional

$$\text{ACE} = \mathbb{E}\left[\frac{p(M \mid A = a_1, C)}{p(M \mid A = a, C)} \cdot Y\right] - \mathbb{E}\left[\frac{p(M \mid A = a_0, C)}{p(M \mid A = a, C)} \cdot Y\right]$$

Similar to above, in our binary case $a_1 = 1$ and $a_0 = 0$. As with AIPW the functional can be decomposed into parts. First we have $p(M \mid A = a_1, C)$, $p(M \mid A = a, C)$, and $p(M \mid A = a_0, C)$. These values represent the propensity scores of M given some value of A and confounding C. Since the propensity is not known, it must be estimated. I use a logistic regression classifier to estimate these values, fitting the classifier to model M as a function of A and C on the original data. To estimate $p(M \mid A = a_1, C)$, $p(M \mid A = a, C)$, and $p(M \mid A = a_0, C)$, I create two copies of the data and intervene by setting $A = 1$ in one copy and $A = 0$ in the other copy. Using these copies along with the original data, I predict all three quantities. Finally, we have Y, which is the true value of the outcome for a given instance.

18. There is a non-IPW method; however it is a much more complex formula.
19. Bhattacharya and Nabi (2022) provide a tests to examine the efficacy of the mediator.

4.3 Instrumental Variable Adjustment

Instrumental Variable (IV) adjustment is another method of computing the causal effect of A on Y in the presence of unmeasured confounding between A and Y. IV adjustment uses a variable Z as an instrument to estimate the effect. A standard application of IV is shown in figure 6. In simple terms, IV adjustment calculates the effect of Z on Y (through A) and then removes the effect that Z has on A.

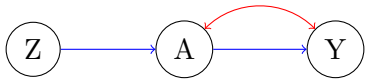


Figure 6: Example of when IV adjustment is used.

4.3.1 IV Criterion

For a IV adjustment to be valid the IV criterion must be met. For some instrument Z and some observed confounders C the IV criterion has three rules for calculating the effect of A on Y:

1. Z only causes Y through A
2. C is a valid backdoor adjustment set for the effect of Z on Y
3. C is a valid backdoor adjustment set for the effect of Z on A

It is also important that Z has a strong causal effect on A otherwise the result will be unstable because of a small denominator.

4.3.2 IV functional

Let $\mathbb{E}[Y^{a_x}] = \sum_{c \in C} p(C = c) \cdot \mathbb{E}[Y \mid A = a_x, C = c]$ (This is just from the backdoor formula)

Then $\text{ACE} = \dfrac{\mathbb{E}[Y^{z_1}] - \mathbb{E}[Y^{z_0}]}{\mathbb{E}[A^{z_1}] - \mathbb{E}[A^{z_0}]}$

Because all of the variables in the system are binarized Y^{z_1} and A^{z_1} are the potential outcomes of Y and A respectively when z is intervened to 1, Y^{z_0}, A^{z_0} are found by setting the z to 0. IV is simply the backdoor formula for the effect of Z on Y where C as a valid backdoor set divided by the effect of Z on A with the same valid backdoor set C. Each of the backdoor adjustment is computed separately and the components can be broken down as follows. $p(Z = z)$ can be modeled using the empirical distribution, but $\mathbb{E}[Y \mid A = a_x, Z = z]$ still needs to be estimated. I fit a logistic regression model to the data and then use it to predict the potential outcomes by intervening on the data.[20]

20. Each backdoor adjustment use the same classifier, the only thing that differs is the intervention (i.e. both parts of the numerator).

4.4 Why Logistic Regression

Choosing the correct machine learning model is important to being able to accurately predict causal effects. Unlike in machine learning a high accuracy is not always the best. striking a balance between bias and variance in paramount. Logistic regression has been shown to be a strong model to be used as the subroutine within these adjustments because it achieves this balance well.

4.5 Causal Ranking Formula

To create an ideal subset of size n, each variable is ranked based on its causal relationship with the outcome Y. Variables are divided into upstream causes $\{U\}$ and downstream effects $\{D\}$ of Y. Using a parameter $\lambda = 0.5$, I find the $\max_n(|U| \cup (\lambda \cdot |D|))$ to computing the ideal subset.[21] Negative causes are considered by using absolute values because negative causes can still be learned. This process yields a subset suitable for machine learning.

4.5.1 Additional Groups

There are two additional groups of variables to the two detailed above. The first is variables A that are not causally related to Y.[22] These variables are assigned a causal score of -1.0 because they are not causally connected. The other group comprises variables whose effects cannot be computed by any of the above methods. These variables are assigned a score of 0.0.

5. Causal Findings

The process of calculating each ACE follows a simple algorithm detailed in Appendix A.3.

5.1 Examples

For clarity I will walk through the process of determining the causal effects of variables with and without unmeasured confounding.

5.1.1 No Unmeasured Confounding

This example will demonstrate the calculation of the effect of Glucose level on the outcome (Diabetes). First the following causal paths are identified: Glucose → Outcome; Glucose → Insulin → DPF → Outcome; and Glucose → Insulin → BMI → Outcome. Then the optimal adjustment set {Age, Skinthickness, BloodPressure} can be calculated as the non-potential outcome parents of all potential outcomes in the after intervening on Glucose.[23] Skinthickness is removed because {Age, BloodPressure} still constitute a valid

21. λ penalizes variables that to not cause Y because changing them would not change Y. Testing showed that $\lambda = 0.5$ may be optional; however it is not guaranteed and left as a hyperparameter.
22. A is not a descendent of Y and Y is not a descendent of A.
23. The potential outcomes are all decedents of Glucose.

backdoor set.[24] This is a valid backdoor set because given the set, A is d-separated from Y^a in the intervention graph and none are decendents of Glucose. AIPW is then used to calculate the ACE, which is found to be 0.322 with 95% confidence intervals from 0.256 to 0.381.[25] Glucose shows a strong causal effect (in fact the strongest), and was about 10% more predictive of the outcome than the average feature.

5.1.2 Unmeasured Confounding

This example demonstrates calculating the effect of V6 on Y in the synthetic ADMG. The causal path is V6 $\rightarrow$ V7 $\rightarrow$ Y. As there is a bidirected edge from V6 to Y, a valid backdoor adjustment set cannot be calculated, thus we begin to look for mediators. V7 intercepts all causal paths from V6 to Y, making it a valid mediator. Then I need to find an adjustment set that completes the front-door criterion. In this case the empty set is sufficient because A and Y are m-separated in the intervention graph when intervening on A, and M and Y are m-separated in the intervention graph given A when intervening on M. Using front-door IPW, I calculated the causal effect to be 0.302 with 95% confidence interval 0.246 to 0.356.

A similar procedure ensues for variables with no mediator but an instrumental variable.

6. Sensitivity Analysis

6.1 Sensitivity to causal assumptions

To ensure the framework can work with causal discovery, certain causal assumptions may be necessary. Since every edge must be oriented, causal assumptions must be imposed on the data. For instance, in the non-mixed diabetes data, orienting the Blood Pressure-Glucose undirected edge posed a challenge due to its situational dependence. Referencing a study by Jalal et al. (2010), which suggests that fructose (metabolized into glucose) could cause high blood pressure, I oriented the edge as Glucose $\rightarrow$ Blood Pressure. However, this orientation is not definitive. Interestingly, FCI suggests a relationship in the opposite direction: Blood Pressure o$\rightarrow$ Glucose. I interpreted this as indicating unmeasured confounding. Alternatively, had I oriented it as Blood Pressure $\rightarrow$ Glucose, Blood Pressure would have been considered an additional cause of Diabetes rather than a non-cause completely changing the causal effect.

Violations of faithfulness or statistical assumptions can lead to causal discovery failing to find the proper graph. For instance, when using the PC discovery method on synthetically generated data, it did not consistently output the correct generating graph. This highlights the challenge of discovering the true underlying structure, which is often unknown in real-world data. Even with discovery, models remain vulnerable to causal misspecification. Additionally, PC's limitation to consider only a subset of possible graphs (DAGs $\subset$ ADMGs) means it may overlook unmeasured confounding, potentially lead-

24. This is the minimal optimal set.
25. The confidence intervals were calculated by computing 200 bootstraps of the data and getting the 2.5% and 97.5% quantiles.

ing to the incorrect declaration of a valid backdoor adjustment set when IV or front-door methods could have be more suitable.

While there is a potential for sensitivity to causal assumptions, being vigilant throughout the process and relying on expert results can greatly reduce the risk.

6.2 Sensitivity to statistical assumptions

Several statistical assumptions are made throughout the pipeline of this process. The main assumption is that relationships in the data can be modeled using logistic regression, which assumes that edges are roughly linear, even though that is not always the case. If a linear model cannot predict the data well, then there will be incorrect estimates for the regression and propensity terms in AIPW, front-door IPW, and IV adjustment, leading to incorrect causal effect estimation. Additionally, if the data goes through the discovery phase, further statistical misspecification is possible. Since discovery relies on the Chi-Squared independence test, statistical misspecification can occur if the assumptions of this test are violated. For example, the test relies on all instances being independent of one another, which is potentially not true in the diabetes dataset given the narrow scope of the participants. While statistical assumptions underlie the foundations of this project, overall most datasets should be robust to these misspecifications.

7. Results

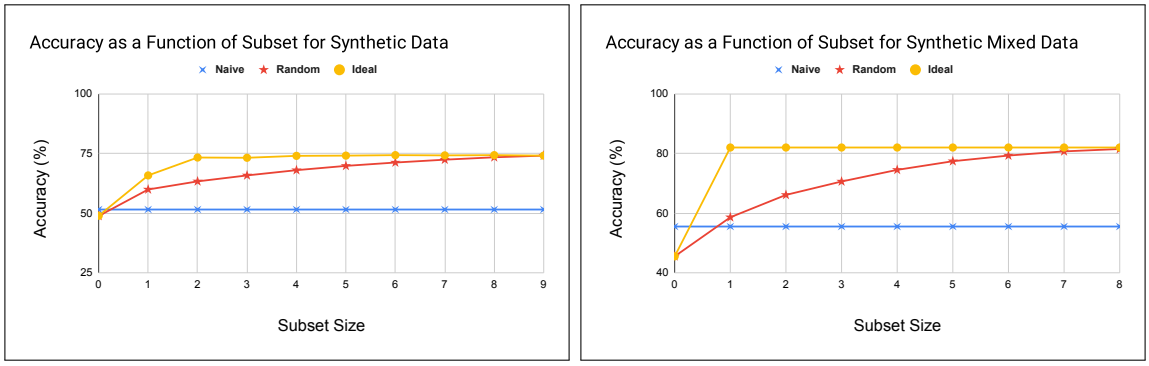
To test the effectiveness of the ideal subset, I trained a series of models using random subsets of size n and calculated the test accuracy. I conducted multiple experiments to explore the strengths and weaknesses of this framework. I used Scikit-learn (Pedregosa et al., 2011), Statsmodel (Seabold and Perktold, 2010), and Torch (Paszke et al., 2019) for machine learning models and Tetrad for causal discovery (Scheines et al., 1998).

7.1 Synthetic Results

I tested the framework's efficacy on synthetic data, both with and without unmeasured confounding. Using different subsets of the features, I trained a Linear Regression model and evaluated its test accuracy. The results are presented in figure 7.

In figure 7, the ideal subset line consistently outperforms the random subset line, demonstrating that the ideal subset has greater predictive power than a random one.[26] This supports the hypothesis that causal inference can enhance feature selection. I also observed that this technique is effective across various Machine Learning models, not just Linear Regression. Notably, it even improved the performance of Neural Network architectures. I evaluated each model on a subset size of 3, comparing random subsets to ideal subsets. The ideal subset consistently outperformed the random subset, as illustrated in figure A.2.

26. The random and ideal lines are identical at subset size of 0 and 9 because these are the points that they are using the same subset (none or all of the features).



(a) Results for Synthetic Data Averaged over 100 trials on independently generated datasets.

(b) Results for a single trial of Mixed Synthetic Data.

Figure 7: Test accuracies from a Linear Regression model trained on different subsets of varying sizes from the synthetic data. **Naive** refers to a classifier that always predicts the mode of the data. **Random** is averaged over all subsets of the given size. **Ideal** is a subset of size n determined using the causal scoring framework.

7.2 Diabetes Results

While demonstrating the methodology on synthetic data is crucial, the ultimate goal is application to real-world datasets. I evaluated the model using the diabetes dataset described in section 3.1.3, with results detailed in figure A.3. The outcomes for the non-mixed data follow those of the synthetic data, further validating the hypothesis; however, the results for mixed data are less definitive, showing both wins and losses at different points. This discrepancy may be explained by the ADMG discovered for the diabetes data, shown in figure A.4, which identifies only two features as causally relevant to predicting the outcome, assigning all other features a causal score of -1.0. With that in mind, while the model can draw on its causal knowledge (up until and including subset size of 2) it does out performs the random subsets. I hypothesize that the model's negative performance can be attributed to its reluctance to train on a variable strongly **correlated** with the outcome due to its method of breaking ties internally.[27] Despite this, the framework demonstrates efficacy on real-world data, provided a robust causal framework is established.

7.3 Distribution Shift

I also theorized that this framework would enhance robustness to distribution shifts. Interestingly this was not the case when tested on the distribution shift data as described in section 3.1.1. The ideal subset did not perform well on either dataset subject to distribution shift, as illustrated in figure A.5. This unexpected outcome suggests that the distribution shift may have increased the causal connections between nodes and the out-

27. Since everything else has a score of -1.0 the order should be random; however it is deterministically random because of the sorting algorithm used. Making this truly random is left as future work.

come, leading to incorrect rankings. Fine-tuning the framework to address distribution shifts is a key area for future work.

8. Future Work

In future work, I aim to expand the project in several exciting directions. Firstly, I plan to implement additional estimation methods that can predict causal effects in scenarios not covered by front-door, IV, and AIPW methods. I also intend to enhance the framework's capability to handle continuous data and address instances of missing data more effectively. Furthermore, I aim to reduce the need for data binarization and enhance the framework's resilience to distribution shifts. Lastly, I want to transition from shipping a Python notebook, to a dedicated framework — making it more user-friendly and accessible, akin to Tetrad.

9. Conclusion

Overall, I consider this project highly successful. It not only presented an intriguing exploration but also provided me with valuable learning experiences. Through this project, I gained a deeper appreciation for the far-reaching impact of causal inference across various domains. I delved into the nuances of automating standard causal processes, such as identifying minimal sets or valid mediators. These processes, often straightforward for a practitioner, require exhaustive search algorithms when automated. Looking ahead, I am eager to further explore how causal inference can enhance machine learning. I aim to extend this project and pursue new avenues where causal inference can revolutionize predictive modeling.

References

Joshua D Angrist, Guido W Imbens, and Donald B Rubin. Identification of causal effects using instrumental variables. *Journal of the American statistical Association*, 91(434): 444–455, 1996.

Alexander Balke and Judea Pearl. Nonparametric bounds on causal effects from partial compliance data. 2011.

George Bebis and Michael Georgiopoulos. Feed-forward neural networks. *Ieee Potentials*, 13(4):27–31, 1994.

Rohit Bhattacharya and Razieh Nabi. On testability of the front-door model via verma constraints. In *Uncertainty in Artificial Intelligence*, pages 202–212. PMLR, 2022.

Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

Leo Breiman. *Classification and regression trees*. Routledge, 2017.

Lidia Ceriani and Paolo Verme. The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini. *The Journal of Economic Inequality*, 10:421–443, 2012.

Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35 (5-6):352–359, 2002.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Corrado Gini. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.]*. Tipogr. di P. Cuppini, 1912.

Isabelle Guyon, Constantin Aliferis, et al. Causal feature selection. In *Computational methods of feature selection*, pages 79–102. Chapman and Hall/CRC, 2007.

Diana I Jalal, Gerard Smits, Richard J Johnson, and Michel Chonchol. Increased fructose associates with elevated blood pressure. *Journal of the American society of nephrology*, 21(9):1543–1549, 2010.

Mary L McHugh. The chi-square test of independence. *Biochemia medica*, 23(2):143–149, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427):846–866, 1994.

Richard Scheines, Peter Spirtes, Clark Glymour, Christopher Meek, and Thomas Richardson. The tetrad project: Constraint based aids to causal model specification. *Multivariate Behavioral Research*, 33(1):65–117, 1998.

Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

Jack W Smith, James E Everhart, WC Dickson, William C Knowler, and Robert Scott Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*, page 261. American Medical Informatics Association, 1988.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Philip Green Wright. *The tariff on animal and vegetable oils*. Number 26. Macmillan, 1928.

Kui Yu, Xianjie Guo, Lin Liu, Jiuyong Li, Hao Wang, Zhaolong Ling, and Xindong Wu. Causality-based feature selection: Methods and evaluations. *ACM Computing Surveys (CSUR)*, 53(5):1–36, 2020.

Kui Yu, Yajing Yang, and Wei Ding. Causal feature selection with missing data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(4):1–24, 2022.

Appendix A.1. A Quick Guide to Machine Learning Models

A.1.1 Logistic Regression

Logistic Regression (LR) is one of the most well known and simplest ML models. The goal of LR is to learn weights $W_1 \dots W_n$ for each feature of the data along with an offset term (W_0). These weights (excluding the offset) are multiplied by the features $f_1 \dots f_n$ of a given instance to achieve a prediction: $W_0 + W_1 \cdot f_1 + \dots + W_n \cdot f_n$. During training the goal is to maximize the probability that the model predicts correctly and during evaluation check if the the prediction is closer to 0 or 1. Dreiseitl and Ohno-Machado (2002) provide a more in depth description.

A.1.2 Decision Trees

Decision Trees (DTs) are also very simple ML models that rely on splitting the data along different linear boundaries. At each layer of the tree the model determines which feature splitting on will best divide the data into groups of 0 and 1. To measure the homogeneity I use the Gini index created by Gini (1912) (a summary in english is provided by Ceriani and Verme (2012)). For each level of the tree we recursively split each leaf seeking to minimize the Gini index over all of the leaves. My DT use 3 layers. Breiman (2017) provides a more in depth description.

A.1.3 Boosted Decision Trees

Boosted Decision Trees build upon DTs by creating an ensemble of trees via boosting. Boosting in essence trains a series of models (in my case 100 decision trees of depth 3) where each model seeks to minimize the prediction error of the previous. Basically the n+1st model focuses on accurately predicting the cases that the nth model struggled with. Friedman (2001) provides more info on boosting.

A.1.4 Bagged Decision Trees

Bagging helps prevent overfitting in over-parameterized models. If an infinite depth decision tree is fit to any training set it can achieve 100% training accuracy because it can split each instance into its own leaf and then predict it correctly. Bagging allows us to reduce the overfitting that infinite depth decision trees faces. Bagging works by bootstrapping the initial data and training multiple (in our case 100 with infinite depth) decision trees and then taking the mode prediction as a result (this would be the mean for continuous data). Breiman (1996) provides more info on bagging.

A.1.5 Random Forests

Random Forests are an extension of bagged decision trees. The only difference is that at each layer only a subset of the features are considers as potential splitting criteria. This

leads more robustness to overfitting. Like bagged decision trees the random forest trains 100 trees with infinite depth. Breiman (2001) provides more info on random forests.[28]

A.1.6 Neural Networks

Neural Networks (NNs) are extensions of regression that allow for hidden layers between the input and the output. Essentially a NN learns new features as combinations of the input features that are predictive of the outcome. It then then uses these features to predict. This allows NNs to fit to data that has a much more complex (non-linear) distribution. Bebis and Georgiopoulos (1994) give a simple overview of NNs.

Appendix A.2. Even More Graphs

A.2.0.1 Partial Directed Acyclic Graphs (PDAGs)

A PDAG is a DAG that allows for undirected edges (—). An undirected edge from X to Y means the edge can either be oriented $X \to Y$ or $Y \to X$. The PC algorithm outputs a PDAG when it cannot decide which way an edge should be oriented. In order to compute the causal effects, all edges should be oriented so the software requires the user to manually direct the edge.

A.2.0.2 Partial Ancestral Graphs(PAGs)

A PAG is a graph that allows for directed, bidirected, o-directed, and o-bidirected edges — o-directed edges are of the form o→ and o-bidirected edges are of the form o–o. Like in ADMGs directed and bidirected edges signify potential causes and potential unmeasured confounding respective. O-directed edges from X o→ Y denote either X → Y, X ↔ Y, or both. O-bidirected edges X o–o Y mean either 1. X → Y, 2. Y → X, 3. X ↔ Y, 1 and 3, or 2 and 3. The FCI algorithm outputs a PAG when it cannot decide which way two nodes should be connected. In order to compute the causal effects, all edges should be oriented as either directed or bidirected so the software requires the user to manually direct the edge.

Appendix A.3. Causal Scoring Algorithm

For each variable A:

1. find all causal paths from A to the outcome.
2. if there are no causal paths then there is no causal effect so we will return a score of -1.0 and continue to next variable.
3. (a) **No unmeasured confounding:** Compute the optimal adjustment set (OAS) by taking the non-potential outcome parents of all potential outcomes in the after intervening on A.

28. This guy did a lot of stuff with decision trees.

   (b) **Unmeasured confounding:** Compute a valid adjustment set (VAS) by taking the parents of A, all X with bidirected paths to A ($X \leftrightarrow \cdots \leftrightarrow$ A) and the parents of all X.[29]

4. Look at each subset of the OAS / VAS from smallest to greatest to find a minimal (optimal, in the case of no unmeasured confounding) adjustment set (MAS)[30]

5. Check if the MAS computed is a valid adjustment set — it should be valid unless there is unmeasured confounding interfering.

6. (a) If the adjustment set is valid use AIPW to compute the causal effect / score and continue to the next variable.

   (b) If it was not valid we must try a different method.[31]

7. Do an exhaustive search to try to find a valid mediator variable and adjustment set that meet the front door criterion.

   (a) If the criterion can be met, use front-door IPW to compute the causal effect / score and continue to the next variable.

   (b) If it was not satisfied we must try our last method.

8. Perform an exhaustive search to try to find a valid instrument and adjustment set that meets the IV criterion.

   (a) If the criterion is met use IV adjustment to calculate the causal effect / score and continue to the next variable.

   (b) If it was not satisfied mark the variable as not computeable and assign a score of 0.0.

Appendix A.4. Additional Figures

29. This is proved to always be a valid adjustment set when there is no bidirected path from A to any of its decedents.
30. this step can be skipped for very large data sets due to time complexity however on data sets with 10-20 features it works well.
31. If there is no unmeasured confounding replace this step with step 8b and remove all steps in between.

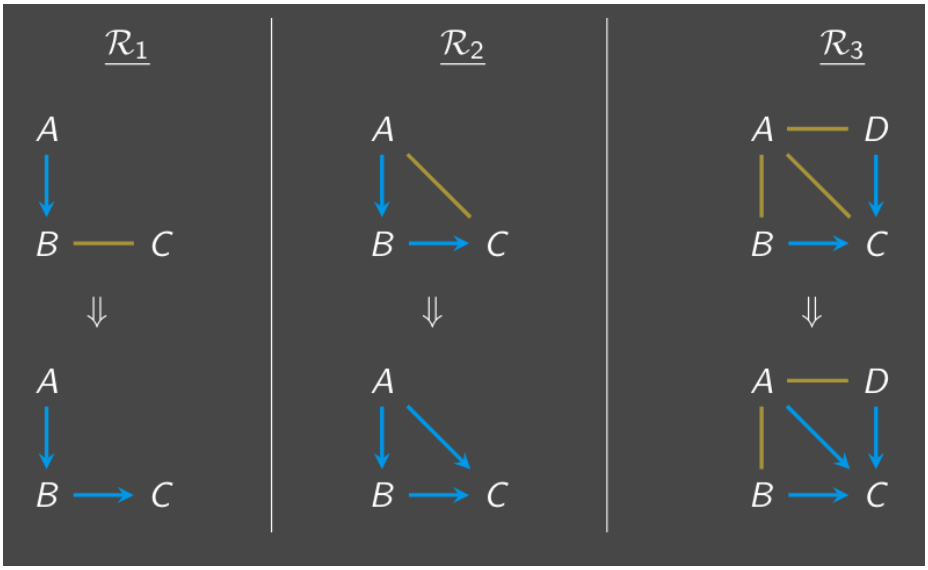


Figure A.1: Rules 1-3 of the PC algorithm. Rule 1 prevents new unshielded colliders from forming. Rule 2 prevents cycles. Rule 3 is necessary because if the edge was oriented the other way that would eventually lead to either a cycle or a new unshielded collider. *Image borrowed from Rohit's CSCI 379 slides.*

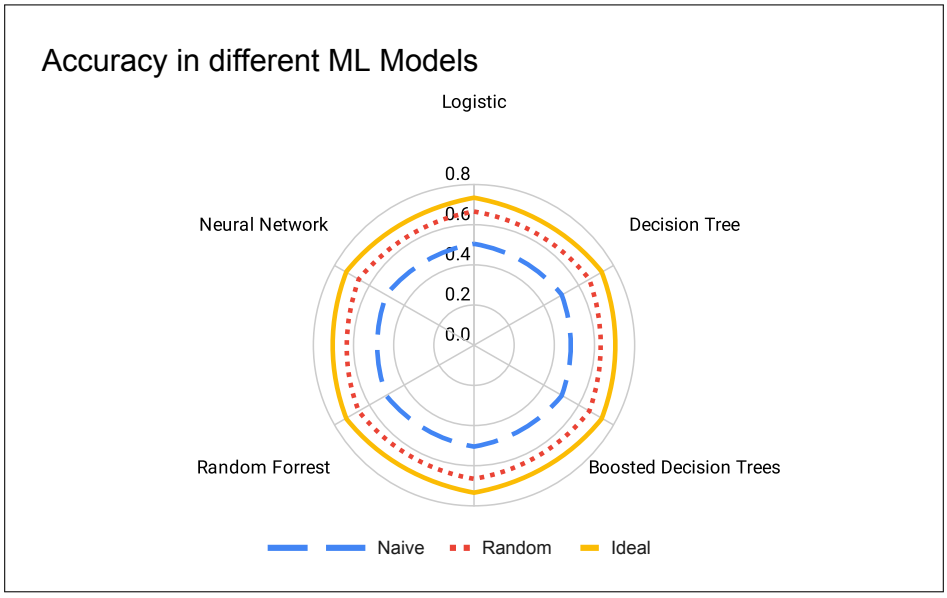
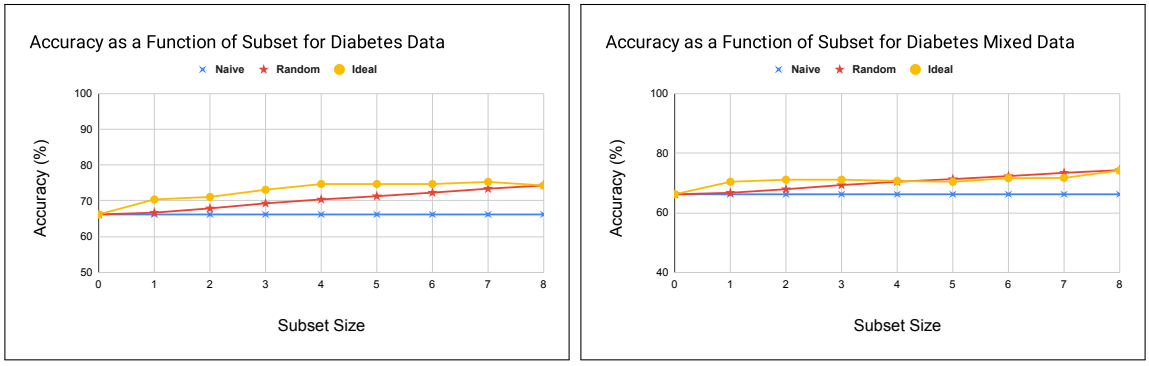


Figure A.2: Accuracy of 5 different machine learning models trained on a subset size of 3. **Naive** refers to a classifier that always predicts the mode of the data. **Random** is averaged over all subsets of size 3. **Ideal** best subset of size 3 determined using the causal scoring framework.



(a) Results for Diabetes Data (b) Results for Mixed Diabetes Data

Figure A.3: Gives test accuracies from a Linear Regression model trained on different subsets of varying sizes from the diabetes data. **Naive** refers to a classifier that always predicts the mode of the data. **Random** is averaged over all subsets of the given size. **Ideal** is a subset of size n determined using the causal scoring framework.

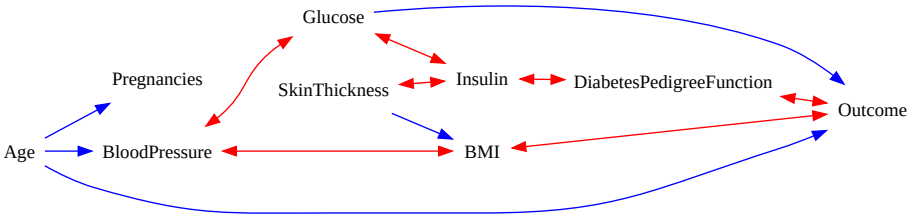


Figure A.4: Underlying causal ADMG for Diabetes data from FCI discovery.

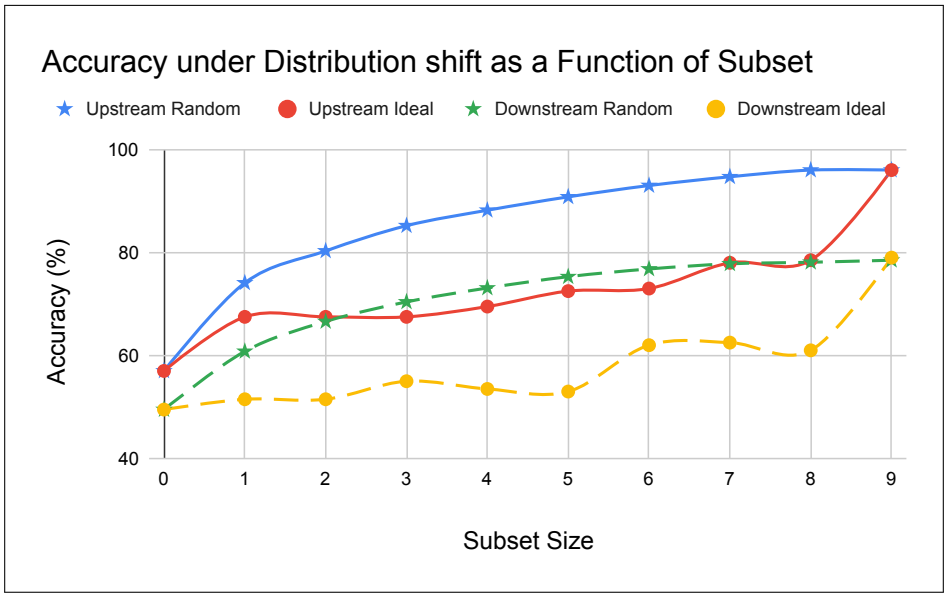


Figure A.5: Accuracy of Logistic Regression model trained on different subsets of features and evaluated on data with an upstream distribution shift and data with a downstream distribution shift.