Combining Sources and Leveraging Contexts

Thesis by
Bijan Mazaheri

In Partial Fulfillment of the Requirements for the
Degree of
Doctorate of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
Defended August 15, 2023

Bijan Mazaheri
ORCID: 0000-0001-9690-8686

ACKNOWLEDGEMENTS

This work was generously funded by the Caltech Computing and Mathematical Sciences Department, a National Science Foundation Graduate Research Fellowship, an Amazon AI4Science Fellowship, and funding from both Prof. Leonard Schulman and Prof. Jehoshua Bruck's labs.

I would like to thank my undergraduate professors at Williams, especially Prof. Duane Bailey, Prof. Keith McPartland, and Prof. Daniel Aalberts, who laid the foundation and frameworks from which I have built my intuition.

I would like to thank my collaborators, Dr. Spencer Gordon, Prof. Yuval Rabani, Dr. Siddharth Jain, Prof. Matthew Cook, Dr. Michaela Hardt, Dr. Atalanti Mastakouri, and Dr. Dominik Janzing, who have helped direct research problems and fill out my understanding of related fields. I would like to thank my advisors, Prof. Jehoshua Bruck and Prof. Leonard Schulman, for their support throughout my time at Caltech.

I thank my running friends, Brandon Wolfe, Jonathan Koch, Adolfo Carvalho, Kevin Horchler, Nick Spector, Chris Myers, and Tony Tomsich, for sharing many miles across California (and Germany) and occasionally enduring small mathematical lectures based on whatever I was fixated on at the time.

I would like to thank my late father, Dr. Mohsen Mazaheri, for inspiring me to pursue a Ph.D. and his continued belief in me. He is dearly missed. In addition, I would like to thank my family, Anna Mazaheri, Nasser Mazaheri, and Said Mazaheri, for their support throughout this process. Finally, I would like to thank my wife, Rebecca, who I met during my first year at Caltech and has been a sounding board for my ideas and shared in my frustrations, sorrows, and triumphs.

ABSTRACT

In this thesis we discuss two levels of knowledge beyond regression and classification. The first involves the identification of exchangeable scenarios or individuals from which causal relationships can be ascertained. We discuss one key difficulty of this task, the "Multi-Source Conundrum," which emerges whenever data is merged from multiple sources. This motivates the "Principle of Limited Latent Classes," an assumption which allows us to introduce new algorithms for deconfounding and causal structure learning.

The second level of knowledge involves the expansion from contextual exchangeability to contextual synthesis. We will study a paradox of nontransitivity that occurs when combining multiple contexts, as well a demonstrating robustness gains from using context-dependent counterfactuals as training features. Through these points, we present contextual synthesis as a new frontier with promise for advances in out-of-distribution robustness, fairness, and privacy.

PUBLISHED CONTENT AND CONTRIBUTIONS

Gordon, Spencer et al. (2023). "Causal inference despite limited global confounding via mixture models". In: *2nd Conference on Causal Learning and Reasoning*.
Bijan Mazaheri participated in the conception of the project, formation of theory, implementation of the experiments, and writing of the manuscript.

Mazaheri, Bijan, Spencer Gordon, et al. (2023). *Causal Discovery under Latent Class Confounding*. arXiv: 2311.07454 [cs.LG].
Bijan Mazaheri participated in the conception of the project, formation of theory, and writing of the manuscript.

Mazaheri, Bijan, Siddharth Jain, Matthew Cook, et al. (2023). *Omitted Labels in Causality: A Study of Paradoxes*. arXiv: 2311.06840 [cs.LG].
Bijan Mazaheri participated in the conception of the project, formation of theory, implementation of the experiments, and writing of the manuscript.

Mazaheri, Bijan, Atalanti Mastakouri, et al. (July 2023). "Causal information splitting: Engineering proxy features for robustness to distribution shifts". In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. Ed. by Robin J. Evans and Ilya Shpitser. Vol. 216. Proceedings of Machine Learning Research. PMLR, pp. 1401–1411. URL: https://proceedings.mlr.press/v216/mazaheri23a.html.
Bijan Mazaheri participated in the conception of the project, formation of theory, implementation of the experiments, and writing of the manuscript.

Mazaheri, Bijan, Siddharth Jain, and Jehoshua Bruck (2021). "Synthesizing New Expertise via Collaboration". In: *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2447–2452. DOI: 10.1109/ISIT45174.2021.9517822.
Bijan Mazaheri participated in the conception of the project, formation of theory, implementation of the experiments, and writing of the manuscript.

Jain, Siddharth et al. (2019). "Short tandem repeats information in tcga is statistically biased by amplification". In: *BioRxiv*, p. 518878.
Bijan Mazaheri participated in the writing of the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Part I

Introduction

Chapter 1

THE HIERARCHY OF KNOWLEDGE

For the past 20 years, researchers have repeatedly demonstrated the predictive capabilities of machine learning (ML) and the generative capabilities of artificial intelligence (AI). While the foundations for these technologies date back to the 1960s, their recent success stems in part from access to large comprehensive datasets (Schmidhuber, 2022).

Knowledge is often limited to the *context* of the data that lead to it, so larger datasets that cover more contexts are generally more powerful than smaller and more specific ones. Unfortunately, broadening the scope of data is not the panacea that this simplistic view promises. For example, consider a broad-reaching survey of alcohol usage, followed by a longitudinal study on the health outcomes of those individuals. For many years, researchers observed what they called a "J-shaped curve," in which optimal mortality rates emerged with one to two alcoholic drinks per day. We now know that this relationship is driven by the influence of patient condition on recommendations for alcohol consumption. Specifically, many people with pre-existing health conditions were advised to avoid alcohol, leading to a significantly less healthy population of teetotalers. When these conditions are adjusted for, the "J" shaped curve disappears, showing a monotonically negative impact of alcohol (T. S. Naimi et al., 2005; Chikritzhs, Fillmore, and Stockwell, 2009). Evidently, comparisons between health recommendations must happen between similarly healthy groups of populations. Such "case-controlled studies" (G. W. Imbens and Rubin, 2015) involve a *constriction* in data-set size in order to isolate a more correct causal relationship.

It is not always possible to extend data in a case-controlled fashion. Within science, data is often collected in sets called "batches," giving rise to "batch effects" that can induce spurious correlations in the aggregate dataset. Extending beyond the laboratory, human-subject studies extend their scope by relaxing entry criteria, giving rise to studies on diverse populations. Historical studies similarly expand by gathering data over extended time-frames. In all of these processes, big data comes hand in hand with *heterogeneity*, which helps some tasks and hurts others. In order to understand the role of data heterogeneity, we must carefully consider the

context that our data comes from and the *target contexts* on which we hope to apply our conclusions. Such generalization highly nuanced, so we will begin by breaking generalization goals into a "hierarchy of knowledge."

1.1 The Hierarchy of Knowledge

We will describe four "levels" of knowledge generalization which are separated by the key challenges involved and the mathematical techniques we use to overcome them. This hierarchy will bear a close resemblance to "Pearl's Causal Ladder," which partitions causality into three "rungs": (1) Associations, (2) Interventions, and (3) Counterfactuals (Judea Pearl, 2009). The hierarchy of knowledge will attempt a similar partitioning, instead separated by the scope of generalization errors.

Level	Name	Tools	Capabilities	How does it fail?
0	Association	Distance, Feature Importance	Anecdotes, Association between examples	Bad data
1	Synthesis	Curve-fitting, Regularization	Generalized prediction, Pattern-detection	Over-fitting, Under-fitting, Insufficient data
2	Transferal	Data re-weighting	Transfer learning, Domain adaptation	Insufficient data overlap
3	Causality	Causal models	Counterfactuals, Universal relationships, Interventions	Incorrect causal modeling, incomplete measurement of the system
4	Wisdom	Combining context-based conclusions	Imagination, Understanding beyond the data	Contradiction

Table 1.1: The hierarchy of knowledge

Level 0: Association All knowledge stems from experiences and events. The simplest form of thought is anecdotal, involving matching new experiences to previous ones. An example of such reasoning would be: "My grandmother is 100 years old and she drinks a glass of wine every night."

Level 0 hinges on the determination of a distance-metric between samples. Principal component analysis (PCA) (KPFRS, 1901) and auto-encoders (Kingma and Welling, 2013) make up crucial tools for this level by helping determine latent features that are necessary to capture the variation between instances. While most inference techniques reach beyond this level into pattern-detection, some elementary techniques like k-nearest neighbors rely only on Level 0 anecdotal evidence.

Level 1: Synthesis With enough data, the first level of knowledge involves pattern-detection for generalization *out of sample*. In Level 1, we seek to separate example-specific information from general trends. These trends can be used for prediction, but their guarantees are limited to the scope of the distribution from which they are drawn.

While Level 0 only requires *correct* data, Level 1's attainability hinges on data *quantity*: More data allows for the generalization of more complex relationships. Out of sample generalization has been well studied, with the most common tool being *regularization* (Sugiyama, 2015).

Level 2: Transferal While Level 1 generalizes patterns from samples to the distribution from which those samples were generated, transferal involves making conclusions about a *different, but related target distribution*. This process is known as domain adaptation (DA). More generally, transfer learning extends this notion to the transfer of tasks (Weiss, Khoshgoftaar, and D. Wang, 2016). The most common approach to domain adaptation involves re-weighting source-data to "look" like a target distribution.

DA is made possible through assumptions on the shared information between p and q. Two such assumptions are *covariate shift* (Shimodaira, 2000) and *label shift* (Schweikert et al., 2008). Both settings involve assuming a constant "label function" $\Pr(Y \mid \mathbf{X})$ when prediction label Y using covariates $\mathbf{X}$. For example, a covariate shift task may involve adapting skin cancer predictions from Europe to Africa — while the function we seek to learn remains constant while the *focus* shifts to darker-skinned populations.

Level 3: Causality Causal inference can be thought of as a special case of DA that involves a shift from "observational" data to an "interventional" target context. Such a setting differs from Level 2 because interventions have a directional impact on a system. For example, going to the beach will not warm the weather. Interventional settings are difficult to predict without further modeling because they often depart from the domain of the data we have at hand. For example, we have likely not seen the climate-impact of forcing many people to go to the beach in the winter. Instead, causality often falls within the realm of *extrapolation*.

At the core of determining the impact of interventions is the comparison between "exchangeable" settings. In order to determine what constitutes exchangeability we must make use of causal assumptions which usually come in the form of a structural causal model (SCM) (Judea Pearl, 2009; G. W. Imbens and Rubin, 2015; Peters, Janzing, and Schölkopf, 2017). "Causal discovery" automates the process of learning these structures (Squires and Uhler, 2022), but is still subject to data-limitations and restrictive assumptions (such as the "faithfullness" assumption, see Uhler et al., 2013).

A primary argument posed by this thesis will be that an additional assumption of inherently low dimensional data, which we will call "the principle of limited latent classes," can extend the power of SCMs beyond canonical examples where exchangeability is easy to define. The essence of this assumption is that the complexity of unmeasured forces does not continue to grow as we expand the scope of our study. When this assumption holds, tools within mixture models can be added to the toolbox for causal inference.

Level 4: Wisdom

The fourth level of knowledge involves understanding the process that lead to our observed data and how those processes affect our conclusions. The most common challenge at this level is that of sampling/survivorship bias, which selectively removes certain data points from our purview.

This thesis will demonstrate that wisdom falls within it's own category of generalization by showing how incorrect conclusions can be drawn from standard causal adjustments in settings with sampling bias. While we will not propose a complete solution to Level 4 generalization, we will present results that suggest approaching the problem using multiple contexts with different sampling biases.

1.2 Summary and Structure

This thesis will begin with an in-depth study of Level 3 knowledge — particularly in the case of multiple batches or populations, which we call "latent global confounding." Within this setting, traditional causal inference is insufficient to identify causal relationships. In Chapter 3, we justify the principal of Limited Latent Classes (LLC) and explain how it can be used to address this insufficiency. Chapter 4 and Chapter 5 will present the first known algorithms for harnessing the LLC to deconfound and learn causal structures this setting.

We will then progress to the study of Level 4 knowledge. Chapter 6 will begin by explaining a paradox that arises among settings with different types of label bias, demonstrating the breakdown of standard causality adjustments and principles. In Chapter 7, we continue to study this paradox and its implications for networks of experts and high-level "decision fusion."

Chapter 6 also introduces techniques for obtaining Level 4 knowledge, namely that auxiliary training tasks can be used as filters of information. In Chapter 8, we see that carefully engineered counterfactual questions and their contradictions with observed

data carry important information for contextually-robust models. Finally, Chapter 9 will discuss the implications of the presented work and the future directions they imply.

This thesis consists of high-level conceptual chapters (Chapters 1, 2, 3, 6, 9) which introduce and discuss the ideas in the low-level technical chapters (Chapters 4, 5, 7, 8). A dependency diagram for these chapters is given in Figure 1.1.

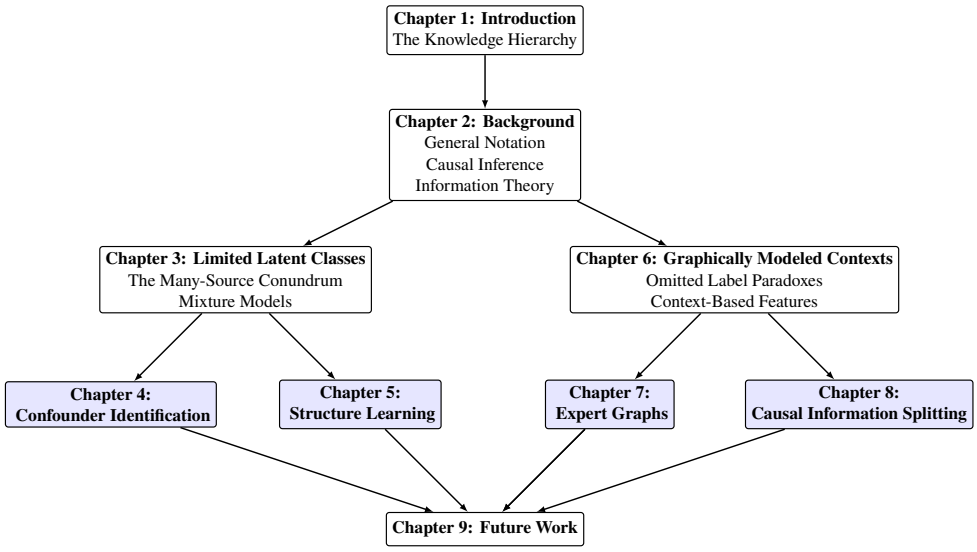


Figure 1.1: A dependency chart of this thesis. Technical chapters, which are adapted from papers, are shaded in blue.

Chapter 2

BACKGROUND

2.1 Notation

The capital Latin alphabet will be used to denote random variables. When referring to sets of these variables, we will use bolded font, e.g. $\mathbf{X} = \{X_1, X_2, \ldots\}$.

The variables discussed in this thesis will generally be discrete. The supports, or "alphabets" of these variables will be denoted using caligraphic script, i.e. $X_i \in \mathcal{X}_i$. The cardinality of these supports will be denoted $|\mathcal{X}|$ and sometimes just referred to as the cardinality of the variable.

Probability distributions on random variables will be denoted $\Pr(\cdot)$. Conditional probabilities will be used frequently, for which we will use a lowercase shorthand to denote assignment. For example, x denotes $X = x$ and $\mathbf{x}$ denotes $\mathbf{X} = \mathbf{x}$. $\Pr(y \mid x) \in [0, 1]$ is a single probability, whereas $\Pr(Y \mid x) \in \triangle^{|\mathcal{Y}|}$ and $\Pr(y \mid X) \in [0, 1]^{|\mathcal{X}|}$ can both be thought of as vectors indexed by the unspecified assignment to Y or X. When these probabilities are parameters of a model, we will use the lowercase Greek alphabet.

The majority of graphs in this thesis will be "structural causal models," which are discussed later in this chapter. Such models are graphs whose vertices are random variables (i.e. capital Latin alphabet). In general, we will try to use $\mathbf{U}$ to denote the set of *unobserved* variables in the system, and $\mathbf{V}$ to denote the *visible* or observed variables. Sometimes, when a specific prediction task is clear, we will also use $\mathbf{X}, Y$ for visible variables, where Y is a label and $\mathbf{X}$ the covariates used to predict that label.

In Chapter 7, a different type of directed graph will be discussed in which the vertices are specific classes of a label variable. Vertices in these graphs are given by the lowercase Latin alphabet because they are assignments to a single variable. Sets of these assignments are denoted using calligraphic script, i.e. $\mathcal{Y} = \{y_1, y_2, \ldots\}$ or lowercase $c = (c_1, c_2, \ldots) \subset \mathcal{Y}$ for paths and cycles.

There will be a few exceptions to these general rules, which will be specified in their relevant chapters.

Graph Notation

To refer to components of a graph, we will use the following operators:

- $\mathbf{PA}(V), \mathbf{CH}(V)$ will refer to the parents and children of a vertex V.
- $\mathbf{AN}(V), \mathbf{DE}(V)$ will refer to the ancestors and descendants of V. $\mathbf{AN}(V) \cup \{V\}$ and $\mathbf{DE}(V) \cup \{V\}$ are denoted using $\mathbf{AN}^+(V), \mathbf{DE}^+(V)$ respectively.
- $\mathbf{MB}(V) = \mathbf{PA}(V) \cup \mathbf{CH}(V) \cup \mathbf{PA}(\mathbf{CH}(V))$ will refer to the Markov boundary of V.[1]
- $\mathbf{NB}_\ell(V)$ refers to the undirected distance ℓ neighborhood of V.

As these operators act on graphs, they can specify the graph structure being used in the superscript, e.g. $\mathbf{PA}^{\mathfrak{G}}(V)$. We will also occasionally write tuples to indicate the intersection of the sets for two vetices, e.g. $\mathbf{CH}(V, W) = \mathbf{CH}(V) \cap \mathbf{CH}(W)$. Finally, these operators can also act on sets to indicate the union of the operation, e.g.

$$\mathbf{PA}(\mathbf{X}) = \bigcup_{X \in \mathbf{X}} \mathbf{PA}(X) \setminus \mathbf{X}. \tag{2.1}$$

Note that our lowercase/uppercase convention can also be applied to operators, e.g. $\mathbf{mb}(V)$ denotes an assignment to $\mathbf{MB}(V)$. Assignments for these operators can also be obtained from a larger set of assignments using a subscript, e.g. $\mathbf{mb}_{\mathbf{c}}(V)$ obtains assignments for $\mathbf{MB}(V) \subseteq \mathbf{C}$ from the assignments of $\mathbf{c}$ to $\mathbf{C}$.

2.2 Causal Inference

The fundamental problem of causal inference Causality encompasses both counterfactual (what could have been) and hypothetical (what could be) statements. If a patient is given treatment and cured, saying that treatment *caused* recovery is equivalent to saying that the patient would not have recovered without treatment (a counterfactual). Similarly, recommending an intervention of treatment carries with it the implication that the patient is better off with treatment than without it.

One framework for understanding causal inference developed by G. W. Imbens and Rubin (2015) involves defining "potential outcomes" of an intervention. The "fundamental problem" of causal statements is that counterfactual and hypothetical outcomes are inherently unobserved. We only have one world and we therefore only

[1] The Markov boundary is the minimal set that d-separates V from all other vertices (Judea Pearl, 2009).

see one potential outcome (e.g. the patient recovers with treatment). We do not see the counterfactual — for all we know, the treatment may have had no effect and the patient would have recovered either way.

We will develop the rest of this section using random variables that represent the potential outcomes of whether the patient receives ($Y^{(1)}$) or does not receive $Y^{(0)}$ treatment. The treatment effect is therefore defined to be the difference between these potential outcomes: $Y^{(1)} - Y^{(0)}$.

Average Treatment Effects One way to access a treatment effect is with a twin. For example, if two identical individuals are given different treatments (e.g. one receives treatment and one receives a control), then we can interpret the outcome of one twin as the counterfactual for the other.

In the absence of identical individuals, we can relax our goal to the identification of an *average treatment effect* (ATE) between two "exchangeable" groups:

$$\text{ATE} := \mathbb{E}[Y^{(1)}] - \mathbb{E}[Y^{(0)}]. \tag{2.2}$$

The simplest way to ensure exchangeability is a randomized controlled trial (RCT), first introduced in Fisher and Wishart (1930). RCTs assign treatments randomly to a population, ensuring that there are no common-causes for the treatment and the outcome.

RCTs are not always possible. It is not ethical to withhold potentially life-saving medication from a sick patient and voters do not generally embrace experimental public policy. In such settings, we must determine how to ascertain causality without active interventions.

Instrumental Variables Natural causes of a treatment can mimic the randomization of RCTs. Such causes are known as instrumental variables and are often used in economics (J. Angrist and G. Imbens, 1995). A famous real-world example involved the use of birthdays to estimate the economic impact of a military draft (J. D. Angrist and A. B. Krueger, 2001). To remove confounding between socioeconomic status and draft eligibility (or perhaps avoidance), economists instead studied the relationship between the birthdays of draft-age individuals and their eventual economic well-being. Birthdays are unrelated to economic status, but functioned as the primary mechanism under which the draft took place. Any relationship between birthday and economic well-being could therefore be attributed to the draft.

Inverse Propensity Weighting Exchangeability gives us insight into how we can mathematically "simulate" the outcome of an RCT. Suppose we have access to *observational* data on patients, some of whom are prescribed a treatment by their doctors. It is common for prescriptions to depend on the *condition* of the patient — sicker patients are more likely to receive treatment while milder cases are monitored but not necessarily treated. Selective prescription therefore drives unexchangeable treated and untreated patient groups — the treated group is on average significantly worse off than the untreated group.

To enforce exchangeability, the health and epidemiological sciences often employ "inverse propensity weighting" (IPW) (G. W. Imbens and Rubin, 2015). Weighting each datapoint by the inverse of the patient's likelihood of receiving their corresponding treatment enforces matching (weighted) distributions of severity between the treated and untreated groups.

IPW techniques rely on assumptions that are typical within epidemiological settings, but do not hold in generality. For example, common *effects* of both treatment and outcome do not induce confounding and therefore need not be a part of the definition of exchangeability. Furthermore, (Judea Pearl, 2009) explains that adjusting for these "collider" variables can actually lead to incorrect calculations of causal effect. The do-calculus has therefore been developed to study causal identifiability with respect to more general graphical models, known as Structural Causal Models, or SCMS (Judea Pearl, 2009).

2.3 Structural Causal Models

Structural Causal Models (Judea Pearl, 2009; Peters, Janzing, and Schölkopf, 2017) graphically model causal systems with arrows representing causation: e.g. $A \to B$ indicates "A causes B." While many dynamical systems may contain cyclic dependencies, restricting theory to acyclic graphs (known as directed acyclic graphs or DAGs) is popular.[2]

SCMs represent a data-generating process, in which each variable is a function of its causes and an independent source of noise. This process implies a factorization of the joint probability distribution,

$$\Pr(\mathbf{v}) = \prod_{v \in \mathbf{v}} \Pr(v \mid \mathbf{pa}_{\mathbf{v}}^{\mathfrak{G}}(v)). \tag{2.3}$$

[2]Cyclic systems can usually be reduced to acyclic graphs by time-indexing the variables (i.e. replacing vertices $\mathbf{V}$ with $\mathbf{V}_1, \ldots, \mathbf{V}_T$) and enforcing the directionality of time by forbidding any "future to past" or "present to present" arrows.

We will call this the "Markov factorization," and the satisfaction of this factorization is referred to causal Markov condition. We sometimes say that $\mathbf{V}$ is "Markovian" in graph $\mathfrak{G}$.

Response Functions

When we have more than one cause of a variable Y, the function determining Y relative to a subset of the causes is indexed by the excluded causes. For example, consider the SCM in Figure 2.2 (a). We can write

$$\Pr(Y = 1 \mid c) = \Gamma_c(X). \tag{2.4}$$

We refer to these indexed functions as **response functions**, for which we reserve the capital Greek alphabet.

"Causal sufficiency" represents the assumption that all of the variables needed to determine a casual relationship are known and measured. Figure 2.2 (b), (c), and (d) show violations of causal sufficiency due to unobserved causes of Y. In these cases, the response functions that govern the values of Y are not known when they are indexed by U. When the distribution of these response functions depends on the covariates, such as in Figure 2.2 (b), (c) and (d), the relationships between the label and the covariates become confounded. The independence between covariates and their response functions in unconfounded systems is referred to as the "principle of independent mechanisms" in Peters, Mooij, et al., 2014.

Independence Properties

For any DAG $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$, we call $\mathbf{P} \subseteq \mathbf{E}$ a **path** if it connects A and B with no repeated vertices. The path is **directed** if it obeys the directions of the edges and **undirected** if it does not. Both directed and undirected paths in a causal DAG can result in dependencies between variables. To understand the conditions for dependence/independence, we will use the concepts of **active** and **inactive** paths, which are defined relative to a conditioning set (Judea Pearl, 2009; Peters, Janzing, and Schölkopf, 2017). Intuitively, whether a path is active or not indicates whether it "carries dependence" between the variables.

In the absence of any conditioning, a path between A to B is active if it is directed or if it is made up of two directed paths from a common cause along that path. In the same unconditioned setting, **inactive paths** are paths that contain a **collider**, i.e. a vertex for which the path has two inward pointing arrows such as $A \rightarrow C \leftarrow B$.

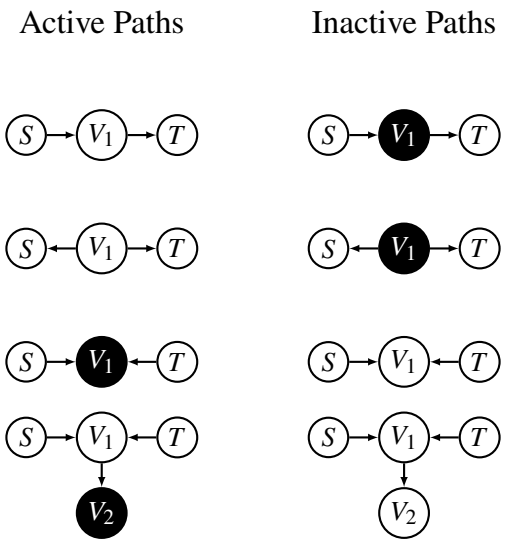


Figure 2.1: Examples of active and inactive paths between S and T. Conditioned vertices are filled in.

When we are given a conditioning set $\mathbf{Z}$, conditional dependencies differ from unconditional ones. Active paths can be **blocked** (thus becoming inactive paths) if some vertex Z along the path between A to B is included in $\mathbf{C}$. Similarly, inactive paths with a collider C can become **unblocked** by including C or some descendant of the collider variable in the conditioning set $\mathbf{Z}$. If two variables A, B contain no active paths (they may contain inactive paths), then we say they are **d-separated** ($A \perp\!\!\!\perp_d B \mid \mathbf{Z}$). If two variables contain at least one active path for a conditioning set $\mathbf{Z}$, we say that they are **d-connected**. See Figure 2.1 for some examples of active and inactive paths.

The data-generating process of structural causal models allows them to be factorized according to the *causal Markov condition*, from which we conclude that d-separation always implies independence. In the opposite direction, it is possible that two d-connected variables by chance exhibit some unexpected statistical independence via cancellation. Under tje assumption of faithfulness (Peter Spirtes, Clark Glymour, Scheines, and David Heckerman, 2000), d-connectedness always implies statistical dependence.

Interventions

Intervening on a "treatment" X has an effect on "outcome" Y that can be calculated by removing all **backdoor paths**, or paths between X and Y which are active but not directed. An example of a backdoor path is given in Figure 2.2 (b). The idea

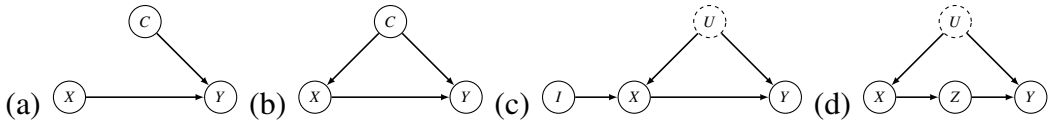


Figure 2.2: (a) C exerts influence on Y, but not on the covariate X. (b) shows both a direct and indirect causal path from X to Y. (c) shows a DAG of an instrumental variable setup. (d) shows an example of a DAG on which the front-door criterion can be applied.

of the backdoor adjustment is to not allow C to vary by changing x, since $X = x$ is driven by an outside intervention rather than the natural variation of C. This fixes the distribution on C to the observed marginal probability distribution despite conditioning on X. Do interventions can therefore be interpreted graphically as shifting from the DAG in Figure 2.2 (b) $\mathfrak{G}$, to the DAG in (a) $\mathfrak{G}_{\overline{X}}$ (i.e. the graph with all incoming edges to X removed).

One way to derive the backdoor adjustment is to modify the observed probability distribution $\Pr(\cdot)$, which is Markovian in $\mathfrak{G}_b$, to the intervened distribution $\Pr^{(a)}(\cdot)$, which is Markovian $\mathfrak{G}_a$.

The "do intervention" in this setting is given by

$$\Pr(y \mid \mathrm{do}(x)) := \Pr^{(a)}(y \mid x) = \frac{\Pr^{(a)}(y, x)}{\Pr^{(a)}(x)} \tag{2.5}$$

By marginalizing over C and applying the Markov factorization for $\mathfrak{G}_a$, we get

$$\Pr(y \mid \mathrm{do}(x)) = \frac{\sum_{c \in \mathcal{C}} \Pr^{(a)}(c) \Pr^{(a)}(x) \Pr^{(a)}(y \mid x, c)}{\Pr^{(a)}(x)} = \sum_{c \in \mathcal{C}} \Pr^{(a)}(c) \Pr^{(a)}(y \mid c, x). \tag{2.6}$$

Finally, we replace all of the probabilities $\Pr^{(a)}(\cdot)$ with the observed probabilities $\Pr(\cdot)$ to get the commonly accepted backdoor adjustment for this setting.

$$\Pr(y \mid \mathrm{do}(x)) = \sum_{c \in \mathcal{C}} \Pr(c) \Pr(y \mid c, x). \tag{2.7}$$

The Frontdoor Criterion

In the absence of the ability to break a backdoor path, we can instead infer the affect of interventions through the front-door criterion (Judea Pearl, 2009), provided that we have access to all directed paths from X to Y and provided that those paths themselves are unconfounded. An example of this setting is given in Figure 2.2(d). Here, we can first infer $\Pr(z \mid \mathrm{do}(x)) = \Pr(z \mid x)$ (for all x) by noting that no backdoor paths

exist between X and Z. We can then compose the effect of this intervention with $\Pr(y \mid \text{do}(z))$ to calculate $\Pr(y \mid \text{do}(x))$:

$$\Pr(y \mid \text{do}(x)) = \sum_{z \in \mathcal{Z}} \Pr(z \mid x) \Pr(y \mid \text{do}(z)). \tag{2.8}$$

$\Pr(y \mid \text{do}(z))$ does contain a backdoor path via $Z \leftarrow X \leftarrow U \rightarrow Y$, but has an observable adjustment set (X) which blocks this path. Hence

$$\Pr(y \mid \text{do}(x)) = \sum_{z \in \mathcal{Z}} \Pr(z \mid x) \sum_{x \in \mathcal{X}} \Pr(x) \Pr(y \mid z, x). \tag{2.9}$$

2.4 Information Theory

This thesis will also use concepts from information theory, with $\mathcal{H}(A)$ indicating the **entropy** of A, $\mathcal{I}(A : B) = \mathcal{H}(A) - \mathcal{H}(A \mid B)$ indicating the **mutual information** between A, B, and $\mathcal{I}(A : B : C) = \mathcal{I}(A : B) - \mathcal{I}(A : B \mid C)$ indicating the **interaction information** between A, B, C.

Lemma 1 (Chain Rule, (Cover, 1999))**.** *For sets of variables* $\mathbf{A}, \mathbf{B}$*, and subset* $\mathbf{B}' \subset \mathbf{B}$

$$\mathcal{I}(\mathbf{A} : \mathbf{B}) = \mathcal{I}(\mathbf{A} : \mathbf{B}') + \mathcal{I}(\mathbf{A} : \mathbf{B} \setminus \mathbf{B}' \mid \mathbf{B}') \tag{2.10}$$

Definition 1 (Cover, 1999)**.** For sets of variables $\mathbf{A}, \mathbf{B}, \mathbf{C}$, the **interaction information** is defined,

$$\mathcal{I}(\mathbf{A} : \mathbf{B} : \mathbf{C}) := \mathcal{I}(\mathbf{A} : \mathbf{B}) - \mathcal{I}(\mathbf{A} : \mathbf{B} \mid \mathbf{C}). \tag{2.11}$$

A key property of interaction information is that it is symmetric to permutations in its three inputs,

$$\mathcal{I}(\mathbf{A} : \mathbf{B} : \mathbf{C}) = \mathcal{H}(\mathbf{A}, \mathbf{B}, \mathbf{C}) + \mathcal{H}(\mathbf{A}) + \mathcal{H}(\mathbf{B}) + \mathcal{H}(\mathbf{C}) - \mathcal{H}(\mathbf{A}, \mathbf{B}) - \mathcal{H}(\mathbf{B}, \mathbf{C}) - \mathcal{H}(\mathbf{C}, \mathbf{A}). \tag{2.12}$$

Another key property is that interaction information can be either positive or negative, differing from mutual information which is non-negative. The following lemmas will describe two common situations in which we can expect positive and negative interaction information.

Lemma 2. *Given three sets of random variables* $\mathbf{A}, \mathbf{B}, \mathbf{C}$ *if* $\mathbf{A} \perp\!\!\!\perp \mathbf{C} \mid \mathbf{B}$ *then* $\mathcal{I}(\mathbf{A} : \mathbf{B} : \mathbf{C}) \geq 0$.

Graphically, Lemma 2 represents a situation where conditioning on **B** d-separates **A** and **C** (i.e. **B** is a *separating set* of **A** and **B**). The symmetry of interaction information means that it is not important which set of variables is the separating set.

The **data processing inequality** uses each "step" of an active path to upper bound the mutual information.

Lemma 3 (Data Processing Inequality (modified from Cover, 1999))**.** *If* $\mathbf{A} \perp\!\!\!\perp \mathbf{C} \mid \mathbf{B}, \mathbf{D}$ *then*

$$\begin{aligned} \mathcal{I}(\mathbf{A} : \mathbf{C} \mid \mathbf{D}) &\leq \min(\mathcal{I}(\mathbf{A} : \mathbf{B} \mid \mathbf{D}), \mathcal{I}(\mathbf{B} : \mathbf{C} \mid \mathbf{D})) \\ &\leq \mathcal{H}(\mathbf{B} \mid \mathbf{D}). \end{aligned} \tag{2.13}$$

Part II

Level 3: Causality

Chapter 3

LIMITED LATENT CLASSES

In this chapter, we introduce the "many-source conundrum," a difficulty that arises when combining data from multiple sources. Though this problem cannot be addressed by classical methods in causal inference, the addition of a new assumption, "the principle of limited latent classes," allows us to employ mixture models. This chapter will provide some musings as to why this assumption is both reasonable and likely and show how it allows mixture models to become a central tool for solving the many-source conundrum and, more generally, handling unobserved, but large-scale confounding.

3.1 The Many-Source Conundrum

The motivation for this problem emerged from using a combined TCGA cancer database (Jain et al., 2021). Our study used the accuracy of ML classifiers as evidence for the presence of signals; if a feature was predictive of cancer-type, we reasoned that it must be related to the development of that cancer. Midway through some exciting results, we discovered that we could distinguish between two types of equipment ("D" and "W" amplification) with correct identification (recalls) of 71% and 86% respectively (Jain et al., 2019). Consequently, our models could leverage this signal to make predictions about any attribute which correlated with this equipment use. Unfortunately, cancer-type was among these equipment-correlated attributes, raising concern about the true origin of our cancer-type prediction accuracy.

We had stumbled apon confounding due to *heterogeneity*, which emerges when equipment or environmental details vary between "batches" of data. This issue extends outside batched experiments, coming into play whenever data spans multiple populations or environments. Counfounding effects sometimes lay dormant, only to awaken during the search for new signals. The equipment-specific signal in TCGA had evaded detection because it was limited to non-coding repeat regions, which are seldom studied in genome-wide association studies (GWAS). This emergence of novel confounding mechanisms is exacerbated by the unprecedented power of modern ML. The novelty of these phenomena conceal them from the intuitions of domain knowledge, making them silent killers of scientific rigor.

Known methods for correcting confounding from heterogeneity involve restricting analysis to homogeneous sub-populations using either case-controlled studies (Schlesselman, 1982) or covariate adjustments (Judea Pearl, 2009). Restriction-based approaches reduce dataset size — limiting the power of many data-demanding machine-learning techniques. This limitation is even more severe in scientific disciplines that rely on batch-based data, upper bounding dataset-size by batch-size. In such settings, the promises of big data become elusive as we fail to grow the size of homogeneous sub-components. We refer to this problem as the "many-source conundrum."

3.2 The Principle of Limited Latent Classes

To handle the many-source conundrum, we will introduce an assumption that allows us to reduce heterogeneity to a limited number of latent classes. This assumption will allow homogeneous components to grow with combined data so long as they are identifiable.

Beyond Graphical Assumptions

Techniques for handing confounding make up the primary task of causal inference. The backdoor and front-door adjustments and the criteria for their application have been synthesized into a general theory of causal identifiability, known as the "do-calculus" (Judea Pearl, 2009). Instrumental variables do not fully identify causal effects, but can be added to the list of techniques for partially determining causal influence.

Traditional approaches to confounding are built on one of two components of a causal model:

1. Observable confounders, for which we can apply the backdoor adjustment.
2. Observable variables which are not affected by the confounder, either along causal paths from the treatment to the effect (for the front door adjustment) or as parents of the treatment (for instrumental variables).

With respect to these classical approaches, the situation presented by the many-source conundrum is grim. If each lab chooses both their equipment (which affects the DNA sequencing output) and the types of cancer they study, then both sequencing output (S) and cancer type (C) are affected by a common cause via the data source (D), graphically modeled as $S \leftarrow D \rightarrow C$. In more complicated systems, all of the

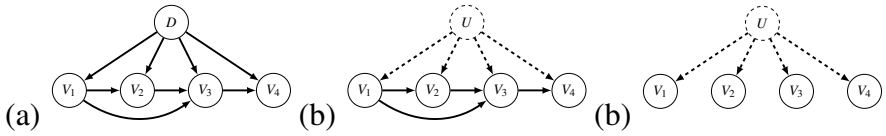


Figure 3.1: (a) A DAG representing the many-source conundrum, with D representing the dataset source. (b) A DAG representing an attempted solution to the many-source conundrum, involving an unobserved latent class that is simpler than D. (c) A DAG representing k-MixProd.

variables in the system could potentially be affected by D, as shown in Figure 3.1 (a). In the absence of any variables unaffected by D, we must apply the backdoor adjustment, giving rise to the many-source conundrum.

The key problem in the many-source conundrum is that the observed class (the dataset source) is too rich. Hence, a reasonable approach is to assume that some simpler unobserved U (often called a **latent class**) is sufficient to control for the confounding induced by heterogeneity. Figure 3.1 (b) shows the causal DAG for this scenario.

Examples of such a U could include private attributes like health status or artificially constructed classes such as individual price sensitivity. Such a U would allow us, in principal, to merge and re-partition data from many sources into homogeneous "unconfounded components." Unfortunately, the existence of a latent class is perhaps even worse than D, because we have now also eliminated the possibility of the backdoor adjustment. Evidently, we must depart from graphical notions of causal identifiability.

Limited Latent Classes

The weakness of the graphical perspective is that it poses no restrictions on the relationships between the variables in the graph. In the context of the many-source conundrum, this worst case analysis ignores the fact that limiting the cardinality of U would also limit U's ability to completely hide the causal dynamics. As it turns out, reality appears to be far from the worst case.

In principal, our backdoor adjustment could apply an even finer partitioning than D, such as assigning each datapoint to its own partition. While a skeptical critic can always assert the inherent uniqueness of every individual (perhaps aided by the cautionary tale from earlier in this chapter), merged datasets have many success stories and science has made measurable progress.

Embedded in our ability to progress to even Level 0 knowledge is the assumption that individual data points share *some* common information that can be synthesized. A consequence of this fact is limited confounding induced by combining datapoints and datasets. For a discrete confounder U, one way of limiting this confounding is by limiting $|\mathcal{U}|$, thereby assuming that the datapoints can be limited to a small number of latent classes. We call this assumption the "principle of Limited Latent Classes" (LLC).

We can hypothesize the reasons why LLC appears to be an inherent property of our world. One observation is that humans and animals are the product of an evolutionary trajectory, so data will inherently have latent classes governed by common ancestors. Clusters also form from large branching effects driven by outside influences, such as migration or a change in environmental conditions. Such events are often driven by low dimensional variables, such as geography and time. Finally, the existence of these latent classes may be a meta-driver for our desire to gather and study the data in the first place. Regardless the source of this phenomenon, it appears to exist and it arms us with the statistical power to solve the many-source conundrum.

3.3 Mixture Models

Under the LLC assumption, we now turn our attention to deconfounding by identifying the probability distribution of U and $\Pr(\mathbf{X}, Y \mid u)$. In the case of discrete U, such a problem can be viewed as identifying a *mixture* of distributions, where each mixture "source" corresponds to a specific assignment u.

There are two problems within mixture models: (1) *Learning* the model, namely, producing any model consistent with (or close to) the observations; (2) *Identifying* the model, namely, producing the true model (or one close to it) up to permutations in the source label. The feasibility of the identification problem hinges on a one-to-one mapping between the observed statistics and the model's parameters. When using the resulting model to deconfound causal relationships, it is imperative that the joint probability distribution with the confounder be identified.

Parametric vs. Graphical Assumptions

Say we observe that the probability density function of $Y \mid X$ forms a bimodal distribution. If we knew the noise in our SCM was additive and Gaussian, it would be possible to infer the response functions $\Gamma_u(X)$ by recovering a mixture of Gaussians. Hence, parametric assumptions can enable the recovery of response functions.

In the absence of parametric assumptions, graphical assumptions can also be used to recover the mixture by harnessing independence properties implied by the causal Markov condition. The simplest case is identifying discrete k-mixtures of product distributions, which we call **k-MixProd**. Such a setting is described by $\mathbf{V} = V_1, \ldots, V_n$ and a latent global confounder or "source" U such that $V_i \perp\!\!\!\perp V_j \mid U$ for all i, j, for which Figure 3.1 (c) shows a causal DAG. The key complexity parameter for identifiability is $k = |\mathcal{U}|$.

This setting well studied (Allman, Matias, and Rhodes, 2009; Jon Feldman, Ryan O'Donnell, and Rocco A Servedio, 2008; Chen and A. Moitra, 2019; S. Gordon, B. H. Mazaheri, et al., 2021). At a high level, the identification algorithms for k-MixProd are primarily method-of-moments approaches that harness information from dependencies in the marginal distribution on X (over U). To achieve varying levels of stability, different approaches require different numbers of independent variables to be affected by U.

Scope Requirements for Identification

Of course, it is possible for U with sufficiently large k to completely control the distribution on $\mathbf{X}$. For example, a cardinality of $k = 2^n$ would be sufficient for binary $X_i \in \{0, 1\}$ to assign each sequence in $\mathbf{X}$ to a latent class in U. Such a powerful U could generate *any* desired probability distribution on $\mathbf{X}$ by simply controlling the probability distribution on U. Limiting k, however, limits the space of marginal probability distributions on $\mathbf{X}$, eventually giving rise to identifiability.

Under a cardinality bound k on the support of U, Allman, Matias, and Rhodes, 2009 showed that $n \geq \Omega(\log(k))$ is sufficient for the generic identification of k-MixProd. In other words, other than a Lebesgue measue 0 set of exceptions, most instances of k-MixProd have a one-to-one correspondence with their observed statistics (the probability distribution on $\mathbf{X}$ marginalized over U) and generating model (up to a set of $k!$ models with permuted labels of U).

For guaranteed identifiability, a lower bound in the case of independently distributed variables was shown to be $2k - 1$ in Li et al., 2015. Tahmasebi, Motahari, and Maddah-Ali, 2018 demonstrated that $n \geq 2k - 1$ in conjunction with a separation condition in the distributions of $X_i \mid U$, is a sufficient upper bound. S. Gordon, B. H. Mazaheri, et al., 2021 recently showed that $n \geq 3k - 1$ is able to give further stability guarantees for the problem. The chapters of this section arbitrarily build results on S. Gordon, B. Mazaheri, Leonard J Schulman, et al., 2020 in Chapter 4

and Allman, Matias, and Rhodes, 2009 in Chapter 5, but our methods easily extend to stronger identifiability conditions with modifications in the sparsity requirements.

The requirements on n imply a duality in the scope of U: while variables that are unaffected by U are useful for the *removal* of U's influence, variables which are affected by U help *identify* that influence. So long as the power of U is limited (i.e. limited cardinality in the case of discrete U), it is likely we will find ourselves in one of the two scenarios.

Mixtures as a Tool

The solutions to k-MixProd make up the main tool which we use for the remainder of this section. In Chapter 4 we explain how to leverage these results in Bayesian networks by solving instances of mixtures of products in conditional probability distributions. Our solution will require knowledge of the DAG on the visible variables, for which we give a structure learning algorithm in Chapter 5.

Chapter 4

CONFOUNDER IDENTIFICATION

This chapter will introduce the first algorithm for identifying mixtures of Bayesian network distributions, which allows for the deconfounding of latent classes. These results are published in S. Gordon, B. Mazaheri, Yuval Rabani, et al., 2023.

4.1 Problem Statement

A Bayesian network is a directed acyclic graph $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$, on a set of $|\mathbf{V}| = n$ random variables. A corresponding Bayesian network distribution (BND) is a probability distribution on the random variables that is Markovian on the graph. That is to say, the joint distribution on the variables can be factored as $\prod_{i=1}^{n} \Pr(V_i = v_i \mid \mathbf{pa}(V_i))$ where $\mathbf{pa}(V_i)$ is the assignment to the parents of V_i. A k-MixBND on $\mathfrak{G}$ is a convex combination, or "mixture", of k BNDs. We represent this situation graphically by a single unobservable random variable U with edges to each of the variable $V \in G$. Here, U is referred to as a "source" variable with range $1, \ldots, k$ and the variables in $\mathfrak{G}$ are referred to as the "observables." The main complexity parameter of the problem is k, representing the number of mixture constituents or "sources."

In this chapter we study the *identification* problem for k-MixBNDs. Specifically, given the graph $\mathfrak{G}$, and given a joint distribution $\Pr(\mathbf{V})$ on the variables (vertices), recover (a) the mixture weights (probability of each source), up to a permutation of the constituents, and (b) for every mixture source and for every vertex V, its conditional distribution given each possible setting to the parents of V. This task identifies the joint probability distribution $\Pr(U, \mathbf{V})$ up to the $k!$ permutations in the label U. Identification will be shown by giving an algorithm that reduces the k-MixBND problem into a series of calls to a k-MixProd oracle. k-MixBND models are not always identifiable, as further discussed in *Assumptions* below. Thus, another contribution of this chapter is to establish a sufficient setting to guarantee identifiability.

Assumptions The following assumptions are used throughout this chapter.

1. *We have access to a k-MixProd oracle requiring N_{mp} variables that are independent within each source.* As different algorithms have different

requirements for the number of independent variables, we will keep our results agnostic to these requirements. The most efficient published algorithm is given in S. Gordon, B. H. Mazaheri, et al., 2021, which requires $N_{\text{mp}} = 3k - 3$ variables and time complexity $\exp(k^2)$. Recent work improves the complexity bound to $\exp(k \log k)$ Spencer L. Gordon et al., 2023.

2. *The observable variables in our BND are binary and discrete.* While a number of papers have focused on continuous or large-alphabet settings, we restrict our focus to the simplest setting of binary, discrete variables.

3. *The mixture is supported on $\leq k$ sources.* If the hidden variable U has unrestricted range (Specifically, range $k = 2^n$ would be enough), the model is rich enough to describe *any* probability distribution on $\mathbf{V}$, making identification impossible. The question is therefore one of trading k against the sample and computational complexity of an algorithm (and the degree of the network).

4. *The underlying Bayesian DAG is sufficiently sparse.* In order to reduce k-MixBND to k-MixProd we need sufficiently many variables that can be separated from each other by conditioning on disjoint *Markov boundaries* (example in Fig. 4.1, definition in Sec. 4.2). As a result, the complexity of the algorithm is exponential in the size of a Markov boundary. Both for complexity and in order to keep n small, a bound on the maximum degree Δ is required. We require $n \geq (\Delta + 1)^4 N_{\text{mp}}$.[1]

5. *The resulting product mixtures are non-degenerate.* Even in mixtures of graphs with sparse structure (in particular the empty graph—the k-MixProd problem), the k-MixBND can be unidentifiable if the mixture components are insufficiently distinct, (e.g., trivially, a mixture of identical sources generates the same statistics as a single source.) Past work has used conditions such as ζ-separation in S. Gordon, B. H. Mazaheri, et al., 2021 to ensure that matrices representing the parameters for each source are well-conditioned. These are not always necessary conditions; characterizing necessary conditions is a difficult question tackled in part in S. L. Gordon and L. J. Schulman, 2022.

6. *The DAG structure representing conditional independence properties within each source, or a common supergraph of these structures, is known.* It is often the case that domain knowledge provides an understanding of the causal DAG.

[1] If the skeleton of $\mathfrak{G}$ happens to be a path, then we only need a milder condition that $n \geq 2N_{\text{mp}}$. For details see S. Gordon, B. Mazaheri, Yuval Rabani, et al., 2023.

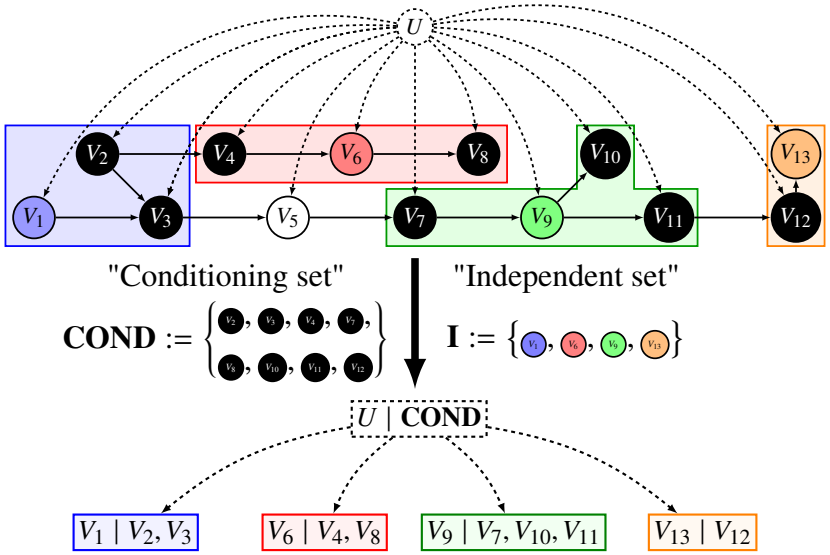


Figure 4.1: The reduction process of conditioning on **COND** to create an instance of k-MixProd. A Bayesian network with four vertices V_1, V_6, V_9, V_{13} and their corresponding disjoint Markov boundaries are indicated.

If the causal DAG is unknown, we are faced with a different problem commonly known as "Causal Discovery" (see C. Glymour, K. Zhang, and P. Spirtes, 2019 for a recent survey.) In the setting of dataset merging, it is likely that the structure can be learned from an individual dataset. An algorithm for learning causal structure in the universal confounding/latent class setting is also given in Chapter 5. Like in Anandkumar, D. Hsu, et al., 2012, the presented algorithm will actually only require a supergraph of the true structure. Hence, some uncertainty in knowledge of the graph can be tolerated. In fact, the algorithm also works even if the different components of the k-MixBND use slightly different causal graphs.

Summary of contributions

Theorem 1. *Our algorithm identifies a k-MixBND distribution with on a graph of maximum degree Δ and of size $n \geq \Omega(N_{MP}\Delta^4)$, using $O(n2^{\Delta^2})$ calls to an oracle for the k-MixProd problem. For an exact statement see Theorem 2.*

The algorithm will be built on the insight that conditioning on a set of Markov boundaries $\mathbf{COND} \subset \mathbf{V}$ of $\mathbf{I} \subset V$ induces *within-source* independence (that is, $V_i \perp\!\!\!\perp V_j \mid U, \mathbf{COND}$ for all $V_i, V_j \in \mathbf{I}$). This describes an instance of k-MixProd for which we can identify the joint probability distribution $\Pr(\mathbf{I}, U \mid \mathbf{COND})$. See

Figure 4.1 for an illustration.

Recovering $\Pr(\mathbf{I}, U \mid \mathbf{COND})$ for some $\mathbf{I} \subset \mathbf{V}$ is insufficient to recover the full joint probability distribution $\Pr(\mathbf{V}, U)$. Hence, we execute a set of $O(n2^{\Delta^2})$ "runs" of a k-MixProd oracle on differing $\mathbf{I}$, $\mathbf{COND}$ and assignments to their Markov boundaries and synthesize information gained from these runes into the joint probability distribution.

The first challenge is to handle symmetries in permutation of the output labels of U by "aligning" the outcomes of these runs. The second challenge is to remove the conditioning of $\mathbf{C}$ from each run. We do this by synthesizing the results of many runs with a procedure we call "Bayesian unzipping." Our key contributions can be summarized by these "alignment" and "unzipping" procedures, as well as the notion of a "good collection of runs" that allows for the successful application of these sub-processes.

Organization The rest of the chapter is organized as follows. In Section 4.1 we outline the literature background of the problem. In Section 4.2 we give some Bayesian network notation. In Section 4.3 we formally develop the notion of a "run," which calls a k-MixProd oracle. In Section 4.4 we explain how the output of the "runs" is combined to get the desired mixture parameters. This section details the processes of alignment and Bayesian unzipping. Section 4.5 explains what is necessary in a group of runs in order for the algorithm to succeed, which provides a framework for defining algorithms in terms of sets of runs.

Background

To our knowledge, the only other attempt at detailing a multiple-run reduction to k-MixProd is Anandkumar, D. Hsu, et al., 2012, which gives an algorithm for mixtures of Markov random fields—i.e., undirected graphical models. As both papers make use of boundary conditioning to induce independence and a form of "alignment," our contribution can be thought of as both an improvement and an extension to the directed graph case. While Anandkumar, D. Hsu, et al., 2012 require a single variable that is independent from the rest of the structure for alignment, our algorithm develops the notion of "good collections of runs" to eliminate this restriction – a contribution which may have implications in the Markov random field setting as well. Additional complications arise for directed graphs because the outputs of the k-MixProd subroutine are conditioned on their Markov boundaries while the desired parameters are only conditioned on their *parents*. Finally, we note

that Anandkumar, D. Hsu, et al., 2012 only guarantees identification of second order marginal probabilities, which is insufficient for causal identification. [2]

Other related work Kivva et al., 2021 contains as a special case a reduction to the k-MixProd problem. Their goal is to learn a causal graphical model with latent variables, but with a very different structure on the visible and latent variables. They allow for a DAG of latent variables with visible children (which is learned as part of their algorithm); on the other hand, they require that there be no causal relations between visible variables. In our work, the structure on the latent variables is trivial (since there is a single latent variable), but the structure on the visible variables is arbitrary. Characterizing identifiability in the generalization of both these settings in which we allow structure on both the visible and latent portion of the graph is a nice problem beyond the scope of this chapter.

Another similarly motivated paper is A. Kumar and G. Sinha, 2021, which studies inference of a certain kind of MixBND, in which the structure of the Bayesian network is known, but the data collected is a mixture over some m unknown interventional distributions. The authors give sufficient conditions for identifiability of the network and of the intervention distributions. At a technical level, the papers are not closely related. k is not a parameter in their work, and instead what is essential is an "exclusion" assumption which says that each variable has some value to which it is not assigned by any of the interventions.

Some other loosely related work includes learning hidden Markov models (D. Hsu, Kakade, and T. Zhang, 2012; Anandkumar, D. J. Hsu, and Kakade, 2012; Sharan et al., 2017), an incomparable line of work to our question, but with somewhat similar motivation. In the same vein, some papers study learning mixtures of Markov chains from observations of random paths through the state space Batu, Guha, and Kannan, 2004; Gupta, R. Kumar, and Vassilvitskii, 2016. These models, too, differ substantially from the models addressed in this chapter, and pose very different challenges. Literature on causal structure learning (P. Spirtes et al., 2000; C. Glymour, K. Zhang, and P. Spirtes, 2019) answers the question of identifying the *presence* of hidden confounders. Fast Causal Inference (FCI) harnesses observed conditional independence to learn causal structure, which can detect the presence of unobserved variables when the known variables are insufficient to explain the observed behavior.

[2] We also mention that Anandkumar, D. Hsu, et al., 2012 introduced the idea of a sparse local separator; if this can be adapted to the directed-graph case one might be able to somewhat relax assumption 4. We do not attempt this in this chapter.

This literature includes the MDAG problem in which the DAG structure may depend upon the hidden variable; see Thiesson et al., 1998 for heuristic approaches to this problem. Other related works study causal inference in the presence of visible "proxy" variables which are influenced by a latent confounder (Miao, Geng, and E. T. Tchetgen, 2018; Kuroki and J. Pearl, 2014; B. Mazaheri, Mastakouri, et al., 2023). This has more recently given rise to attempts at deconfounding using multiple causes (D. Heckerman, 2018; Ranganath and Perotte, 2018; Y. Wang and Blei, 2019). The initial assumptions of Y. Wang and Blei, 2019 were shown to be insufficient for deconfounding in Ogburn, Shpitser, and E. J. T. Tchetgen, 2019. This illustrates the necessity of identifying of the joint probability distribution with the confounder. [3]

Finite mixture models have been the focus of intense research for well over a century, since pioneering work in the late 1800s (Newcomb, 1886; Pearson, 1894). See, e.g., the surveys Everitt and Hand, 1981; Titterington, Smith, and Makov, 1985; Lindsay, 1995; McLachlan, Lee, and Rathnayake, 2019.

4.2 Preliminaries

Markov Boundaries This chapter will repeatedly use the fact that conditioning on the "Markov boundary" of a vertex $\mathbf{MB}(V)$ makes V conditionally independent from everything else in the graph (J. Pearl, 2014).

Definition 2 (Markov Boundary)**.** For a vertex Y in a DAG $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$, the **Markov boundary** of Y, denoted $\mathbf{MB}(Y)$, is defined by

$$\mathbf{MB}(Y) \coloneqq \mathbf{PA}(Y) \cup \mathbf{CH}(Y) \cup \mathbf{PA}(\mathbf{CH}(Y)) \setminus \{Y\}.$$

Lemma 4 (See J. Pearl, 2014)**.** *For any vertex $V \in \mathbf{V}$ and subset $S \subseteq V \setminus (\mathbf{MB}(V) \cup \{V\})$,* $\Pr(V \mid \mathbf{MB}(V), S) = \Pr(V \mid \mathbf{MB}(V))$.

Observation 1. For any $X, Y \in \mathbf{V}$, $X \in \mathbf{MB}(Y) \iff Y \in \mathbf{MB}(X)$

Within-source probabilities It will be easier to write $\Pr_u(v) = \Pr(v \mid u)$ to give the probability distribution within a source.

Finally, here are a few more definitions that will make the upcoming sections simpler.

Definition 3 (Top)**.** We will use $\mathrm{Top}(V)$ to denote $\mathbf{MB}(V) \setminus \mathbf{CH}(V)$.

[3] Thanks to Betsy Ogburn for her thoughts on this topic.

Definition 4 (Depth of a vertex)**.** Given a DAG $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$ and any vertex $V \in \mathbf{V}$, let $d_{\mathfrak{G}}(V)$ be the **depth** of V in $\mathfrak{G}$, i.e. the length of the shortest path from a degree-0 vertex in $\mathfrak{G}$. When $\mathfrak{G}$ is clear from context, we'll omit the subscript.

Definition 5. We'll introduce a parameter $\gamma(G)$ which will appear in the complexity of the identification procedure, which is defined by $\gamma(G) \coloneqq \max_{V \in \mathbf{V}} |\mathbf{MB}(V)|$.

4.3 Applying a k-MixProd run

Our algorithm will induce instances of k-MixProd through post-selected conditioning. A significant portion of this paper will be accounting for multiple calls (or "runs") of a k-MixProd oracle and explaining how their results can be combined.

Describing runs

We will need to keep track of two crucial elements of each "run" of a k-MixProd oracle.

1. Which variables $\in \mathbf{V}$ are passed to our k-MixProd oracle as independent variables (the **independent set**).
2. Which variables $\in \mathbf{V}$ we have conditioned on (the **conditioning set**) and what values we have post-selected these variables to take.

A sufficient conditioning set to induce within-source independence among the independent set is the union of their Markov boundaries. This will be further refined in Subsection 4.5.

Definition 6 (Run)**.** A **run** over a graph $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$ is a tuple $a = (\mathbf{I}^a, f^a)$ where $\mathbf{I}^a \subseteq \mathbf{V}$ are variables that we will d-separate (within each source) by conditioning on assignments to the set

$$\mathbf{COND}^a \coloneqq \bigcup_{I \in \mathbf{I}^a} \mathbf{MB}(I).$$

The value of the assignment is given by $f^a : \mathbf{COND}^a \to \{0, 1\}$. We'll call $\mathbf{I}^a$ the **independent set** for a, and $\mathbf{COND}^a$ the **conditioning set**.

We will restrict our attention to *well-formed runs*, i.e. runs for which $\mathbf{I}^a \cap \mathbf{COND}^a = \emptyset$.

Definition 7. An individual run $a = (\mathbf{I}^a, f^a)$ is N**-independent** if $|\mathbf{I}^a| \geq N$.

Superscript notation We'll write $\mathbf{mb}^a(V)$, $\mathbf{pa}^a(V)$, $\mathbf{ch}^a(V)$ to refer to the assignment to the Markov boundary of V, parents of V, and children of V as set by run a.[4] In a similar spirit, we'll occasionally write v^0 to denote the assignment $V = 0$.

Definition 8 (Distribution induced by a run)**.** For any well-formed run a, the induced distribution on the variables in $\mathbf{I}^a$ is denoted by

$$\Pr^a(\cdot) = \Pr(\cdot \mid \mathbf{cond}^a),$$

where $\mathbf{cond}^a$ is the assignment to $\mathbf{COND}^a$ in keeping with our conventions.

The outputs of applying a k-MixProd oracle to $\Pr^a(\mathbf{I}^a)$ are a matrix $\mathbf{M}^a \in [0,1]^{|\mathbf{I}^a| \times k}$ and a vector of mixture weights, $\pi^a \in [0,1]^k$ (satisfying $\sum_u \pi_u^a = 1$) given by

$$\begin{aligned} M_{i,u}^a &:= \Pr^a(X_i^a = 1 \mid U_a = u) \\ &= \Pr(X_i = 1 \mid U^a = u, \mathbf{COND}^a), && \forall X_i \in \mathbf{I}^a, u \in [k] \\ \pi_u^a &:= \Pr^a(U^a = u) = \Pr(U_a = u \mid \mathbf{COND}^a), && \forall j \in [k] \end{aligned} \tag{4.1}$$

where U^a is the mixture source distributed over $[k]$ according to $\Pr(U \mid \mathbf{COND}^a)$. Note that because mixtures are invariant to permutations of mixture component labels, we cannot guarantee correspondence between the labels for the source variables from different runs. Hence the labels of U^a map to an unknown permutation of the labels in U. Alignment of these labels is handled in Section 4.4.

4.4 Combining Runs

A single run of the k-MixProd oracle will not contain sufficient information to learn the parameters of the k-MixBND problem. Instead we must synthesize information across *multiple* runs.

Aligning source labels across different runs

Each run of the k-MixProd oracle will return $\Pr^a(V \mid U_a = u)$ for some arbitrary permutation U_a of the variable. We need to align all of the outputs to the same permutation of the source, U. If the runs overlap on at least one variable with the same mixture probabilities, we can use that "alignment variable" to identify which source corresponds to which set of parameters. In our setup, we will guarantee these alignment variables exist by ensuring that runs have shared vertices in their independent sets whose Markov boundaries have identical assignments.

[4] Any quantities parameterized by a run will take the parameter as a superscript.

Definition 9. $X \in \mathbf{V}$ is **separated** if for all $u_i \neq u_j \in [k]$, $\Pr_{u_i}(x) \neq \Pr_{u_j}(x)$.

Definition 10 (Aligned runs)**.** A pair of runs a, b over independent sets $\mathbf{I}^a, \mathbf{I}^b$ is **alignable** if there exists a separated $X \in \mathbf{I}^{(a)} \cap \mathbf{I}^{(b)}$ such that $\Pr_u^a(V^{(ab)}) = \Pr_u^b(V^{(ab)})$ for all $u \in [k]$. We'll call any such random variable X an **alignment variable**, and use $\mathbf{AV}(a, b)$ to denote the set of all alignment variables for $\Pr^a$ and $\Pr^b$. We sometimes say a and b are "aligned at" X.

Definition 11 (Alignment spanning tree)**.** We say a set of ℓ runs is alignable if there exists an undirected spanning tree over the graph with vertices $a_1, \ldots a_\ell$ and an edge $\{a_i, a_j\}$ whenever $\mathbf{AV}(a_i, a_j) \neq \emptyset$. We call this the **alignment spanning tree**.

The alignment step will take the output from alignable runs and permute the mixture labels until the parameters match along each alignment variable. Pseudocode for this procedure is given in Algorithm 1.

Algorithm 1: Alignment

Input: A set of runs $\mathcal{A} = \{a_0, \ldots, a_\ell\}$ with outputs M^a, π^a for each $a \in \mathcal{A}$. In addition, we have a spanning tree $\mathcal{T} = (\mathcal{A}, \mathbf{E})$ and alignment variables $\mathbf{AV}(a_i, a_j) \subseteq \mathbf{I}^{a^{(i)}} \cap \mathbf{I}^{a^{(j)}}$.
Output: $\Pr_u^a(\mathbf{I}^a)$ and $\Pr^a(u)$ for each $a \in \mathcal{A}$ and $u \in [k]$.
Let $\Pr^{a_0}(u) \leftarrow \pi_u^{a_0}$ and $\Pr_u^{a_0}(\mathbf{I}^{a_0}) \leftarrow M_{-,u}^{a_0}$ for an arbitrary choice of a_0
for *each edge* (a_i, a_j) *along a breadth first traversal of* $\mathcal{T}$ *from* a_0 **do**
  Choose some $X_{\mathbf{AV}} \in \mathbf{AV}(a_i, a_j)$.
  Let q, r give the indices for the alignment variable, i.e. $X_{\mathbf{AV}} = X_q^{a_i}$ and $X_{\mathbf{AV}} = X_r^{a_j}$.
  Find σ, the permutation on the sources that minimizes $\left\| M_{q,-}^{a_i} - \sigma M_{r,-}^{a_j} \right\|_\infty$.
  Assign $\Pr^{a_j}(U) \leftarrow \sigma\pi^{a_j}$ and $\Pr_u^{a_j}(\mathbf{I}^a) \leftarrow \sigma M^{a_j}$.
end

Bayesian unzipping: recovering parameters per source

Recall that our algorithm conditions on Markov boundaries to induce independent variables. Hence, after aligning the sources in runs of the k-MixProd oracle we will have access to $\Pr_u(Y \mid \mathbf{MB}(Y))$ for each $Y \in V$. Our goal is to obtain $\Pr_u(\mathbf{V})$, which is described by the parameters $\Pr_u(Y \mid \mathbf{PA}(Y))$ for each $Y \in \mathbf{V}$. Note that

$$\Pr_u(y^1 \mid \mathbf{mb}(Y)) = \frac{\Pr_u(y^1, \mathbf{mb}(Y))}{\Pr_u(y^1, \mathbf{mb}(Y)) + \Pr_u(y^0, \mathbf{mb}(Y))}. \tag{4.2}$$

The terms in this fraction are all of the same form and can be factored according to the DAG into

$$\Pr_u(y, \mathbf{mb}^a(Y)) = \Pr_u(\mathbf{top}(Y)) \Pr_u(y \mid \mathbf{pa}^a(Y)) \underbrace{\prod_{V \in \mathbf{CH}(Y)} \Pr_u(v^a \mid f^a(\mathbf{PA}(V) \setminus \{Y\}), y)}_{\Pr_u(\mathbf{ch}^a(Y) \mid \mathbf{top}^a(Y), y)}.$$

See Figure 4.2 for a concrete example of this decomposition. After substituting this factorization into Equation (4.2) we see that $\Pr_u(\mathbf{top}(Y))$ appears in both the numerator and denominator because it is independent of the assignment to Y. Simplification leaves only the following terms:

1. $\Pr_u(y^0 \mid \mathbf{pa}^a(Y))$ and $\Pr_u(y^1 \mid \mathbf{pa}^a(Y))$, which must sum to 1.
2. $\Pr_u(\mathbf{ch}^a(Y) \mid \mathbf{top}^a(Y), y^0)$ and $\Pr_u(\mathbf{ch}^a(Y) \mid \mathbf{top}^a(Y), y^1)$ which are both the product of the desired parameters of variables later in the topological ordering. We can ensure we have access to these terms by solving for the parameters of $V \in \mathbf{V}$ in a reverse-topological ordering.[5]

We can substitute $1 - \Pr_u(y^1 \mid \mathbf{pa}^a(Y))$ for $\Pr_u(y^0 \mid \mathbf{pa}^a(Y))$ in the expanded version of Equation (4.2) to obtain a single equation with only $\Pr_u(y^1 \mid \mathbf{pa}^a(Y))$ as an unknown, which we can then solve. The pseudocode for this process is given in Algorithm 2.

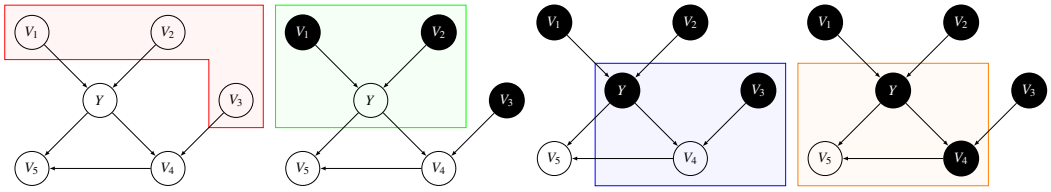


Figure 4.2: We can decompose $\Pr_u(v_1, v_2, v_3, y, v_4, v_5) = \Pr_u(v_1, v_2, v_3) \Pr_u(y \mid v_1, v_2) \Pr_u(v_4 \mid y, v_3) \Pr_u(v_5 \mid y, v_4)$. U and any other variables in the graph are omitted for clarity.

Recovering the distribution on sources

Now consider some arbitrary run a with conditioning $\mathbf{cond}^a$. Since $\Pr_u(\mathbf{V}) = \prod_{V \in \mathbf{V}} \Pr_u(V \mid \mathbf{PA}(V))$, knowing $\Pr_u(V \mid \mathbf{PA}(V))$ grants us full access to the within-source probability distribution $\Pr_u(\mathbf{V})$ after Bayesian unzipping. From this we can

[5]We will want to ensure that we only need to unzip parameters from vertices of a bounded depth, which bounds the iterations of this step. Details on how this is done appear in Section 4.5.

Algorithm 2: Bayesian Unzipping

Input: A collection of runs $\mathcal{A}$ of size at most $2^{O(\Delta^2)}$ and their aligned output. For each V_i and assignment to its parents $\mathbf{pa}(V_i)$ there must be some run with V_i in its independent set with parents conditioned to $\mathbf{pa}(V_i)$.

Output: $\widetilde{\Pr}_u(Y \mid \mathbf{PA}(Y))$

Fix a topological ordering on the vertices in $\mathbf{V}$, $\langle X_1, X_2, \ldots, X_n \rangle$; **for** $i = n, n-1, \ldots, 1$ **do**
  for *each assignment* $\mathbf{pa}(X_i)$ *to* $\mathbf{PA}(X_i)$ **do**
    Let a be a run in $\mathcal{A}$ with $\mathbf{pa}^a(X_i) = \mathbf{pa}(X_i)$.;
    for $u = 1, \ldots, k$ **do**
      for $b = 0, 1$ **do**
        if X_i *is a bottom vertex* **then**
          Set $\widetilde{\Pr}_u(x_i^b \mid \mathbf{pa}^a(X_i)) \leftarrow \widetilde{\Pr}_u^a(x_i^b)$.;
        else
          Set $\widetilde{\Pr}_u(x_i^b \mid \mathbf{pa}(X_i)) \leftarrow \frac{\widetilde{\Pr}_u^a(x_i^b)\widetilde{\Pr}_u(\mathbf{ch}^a(X_i) \mid \mathbf{top}^a(X_i), x_i^{1-b})}{\widetilde{\Pr}_u^a(x_i^b)\widetilde{\Pr}_u(\mathbf{ch}^a(X_i) \mid \mathbf{top}^a(X_i), x_i^{1-b}) + \widetilde{\Pr}_u^a(x_i^{1-b})\widetilde{\Pr}_u(\mathbf{ch}^a(X_i) \mid \mathbf{top}^a(X_i), x_i^b)}$;
        end
      end
    end
  end
end

we obtain $\Pr_u(\mathbf{cond}^a) = \Pr(\mathbf{cond}^a \mid u)$. The k-MixProd oracle will also return $\Pr^a(U) = \Pr(U \mid \mathbf{cond}^a)$ when run on a (after source alignment). Finally, $\Pr(\mathbf{cond}^a)$ is directly observable. Combining these terms in Bayes' rule lets us compute the distribution on U (under the assumption of positivity),

$$\Pr(u) = \frac{\Pr(u \mid \mathbf{cond}^a)\Pr(\mathbf{cond}^a)}{\Pr(\mathbf{cond}^a \mid u)}.$$

Outline of the combination process

Combining a set of runs $\mathcal{A}$ has four steps.

1. Use a k-MixProd oracle on $\Pr(\mathbf{I}^a \mid \mathbf{COND}^a)$ for each run $a \in \mathcal{A}$ to compute $P(V \mid \mathbf{MB}(V), U_a = u)$ for all variables $V \in \mathbf{V}$.

2. Align the parameters obtained from the previous step to ensure that U means the same thing across different runs, giving $\Pr_u(V \mid \mathbf{MB}(Y))$.

3. Recover $\Pr_u(V \mid \mathbf{PA}(V))$ for each vertex $V \in \mathbf{V}$ via Bayesian unzipping.

4. Compute $\Pr(U)$ by applying Bayes' law.

The full procedure appears as Algorithm 3.

Algorithm 3: The k-MixBND algorithm

Input: A good collection of runs $\mathcal{A}$ of size at most $2^{O(\Delta^2)}$.
Output: $\widetilde{\Pr}_u(Y \mid \mathbf{PA}(Y))$ and $\widetilde{\Pr}(U)$.
Estimate $\widetilde{\Pr}^a(\mathbf{I}^a)$ for all runs $a \in \mathcal{A}$. ;
for *each run* $a \in \mathcal{A}$ **do**
  Set $(M^a, \pi^a) \leftarrow$ LearnProductMixture($\Pr^a(\mathbf{I}^a)$);
end
Run Algorithm 1 to align the sources in the output of all the runs in $\mathcal{A}$.;
Run Algorithm 2 to unzip the parameters.;
Fix any run $a \in \mathcal{A}$.;
for $u = 1, \ldots, k$ **do**
  Set $\widetilde{\Pr}(u) \leftarrow \frac{\widetilde{\Pr}(u|\mathbf{cond}^a)\widetilde{\Pr}(\mathbf{cond}^a)}{\widetilde{\Pr}_u(\mathbf{cond}^a)}$.;
end

4.5 Collections of runs

With the main concepts of source alignment and Bayesian unzipping now defined, our algorithm will primarily consist of finding a good collection of these runs so that these subroutines can be successfully applied to recover the k-MixBND mixture.

Observation 2. Two runs a, b are aligned at $X \in \mathbf{V}$ if and only if

1. $X \in \mathbf{I}^a \cap \mathbf{I}^b$,
2. $\mathbf{mb}^a(X) = \mathbf{mb}^b(X)$, i.e, $f^a(\mathbf{MB}(X)) = f^b(\mathbf{MB}(X))$, and
3. X is separated given $\mathbf{mb}^a(X)$ (equivalently, given $\mathbf{mb}^b(X)$).

Definition 12. A collection of runs $\mathcal{A}$ **covers** $X \in \mathbf{V}$ if for every assignment $\mathbf{pa}(X)$ to $\mathbf{PA}(X)$ there exists a run $a \in \mathcal{A}$ with $X \in \mathbf{I}^a$ and $\mathbf{pa}(X) = \mathbf{pa}^a(X)$.

Definition 13 (A good collection of runs)**.** A collection of well-formed runs $\mathcal{A}$ is *good* if it is (i) alignable via an alignment spanning tree, (ii) every run is N_{mp}-independent, and (iii) the collection covers every vertex in $\mathbf{V}$.

The following is our main result on good collections of runs:

Theorem 2. *Given a graph with max degree Δ satisfying $n \geq N_{mp} \times O(\Delta^4)$, we can find a set of centers $\mathbf{X} = \{X_1, \ldots, X_{N_{mp}}\} \subseteq \mathbf{V}$ of size N_{mp} and depth at most $3N_{mp}$, such that Algorithm 4 succeeds in finding a good collection of runs $\mathcal{A}$ of size $O(2^{\Delta^2} n)$.*

While any good collection of runs will suffice for our algorithm, Theorem 2 represents conditions under which we can provably obtain such a collection of runs.

A generic good collection of runs

To prove Theorem 2, we will sketch the construction of a good collection of runs before giving its details. To ensure alignment is possible, we will construct a set of *central runs*, $\mathcal{A}_C$ which we can align to each other and which all other runs will be alignable to.

Definition 14 (Centers, Central Runs)**.** A set of vertices $\mathbf{X} = \{X_1, \ldots, X_{N_{\text{mp}}}\} \subseteq \mathbf{V}$ will be called **centers** if the Markov boundaries of the vertices in $\mathbf{X}$ are disjoint. Given a set of centers $\mathbf{X}$, a run a is called a **central run** if $\mathbf{I}^a = \mathbf{X}$.

To build these central runs, we will start with a set of N_{mp} vertices

$$\mathbf{X} = \{X_1, X_2, \ldots X_{N_{\text{mp}}}\}$$

with *disjoint* Markov boundaries and a maximum depth of $3N_{\text{mb}}$, whose existence is implied by our degree bounds. An example of four such vertices is given in Figure 4.1.

First, we fix a run a_0 with $\mathbf{I}^{a_0} = \mathbf{X}$ and $\mathbf{mb}^{a_0}(\mathbf{X})$ being chosen arbitrarily where $\mathbf{MB}(\mathbf{X}) \coloneqq \cup_{X_i \in \mathbf{X}} \mathbf{MB}(X_i)$. We will refer to this assignment $\mathbf{mb}^{a_0}(\mathbf{X})$ as the *default* assignment. Each run in $a \in \mathcal{A}_C$ will have the same independent set $\mathbf{I}^a = \mathbf{X}$ and will agree with a_0 on the assignment to all of the conditioning set other than the Markov boundary of some $X_i \in \mathbf{X}$, i.e.

$$f^a(V) = f^{a_0}(V) \quad \forall V \in \mathbf{MB}(\mathbf{X}) \setminus \mathbf{MB}(X_i). \tag{4.3}$$

The central runs will span over all assignments $\mathbf{mb}(X_i)$ to $\mathbf{MB}(X_i)$ for each $X_i \in \mathbf{X}$. We'll write each such run as $a_0[\mathbf{mb}(X_i)]$.

Definition 15. $\mathcal{A}_C \coloneqq \{a_0\} \cup \left\{a_0[\mathbf{mb}(X_i)] : i \in [3k-3], \mathbf{mb}(X_i) \in \{0,1\}^{\mathbf{MB}(X_i)}\right\}$.

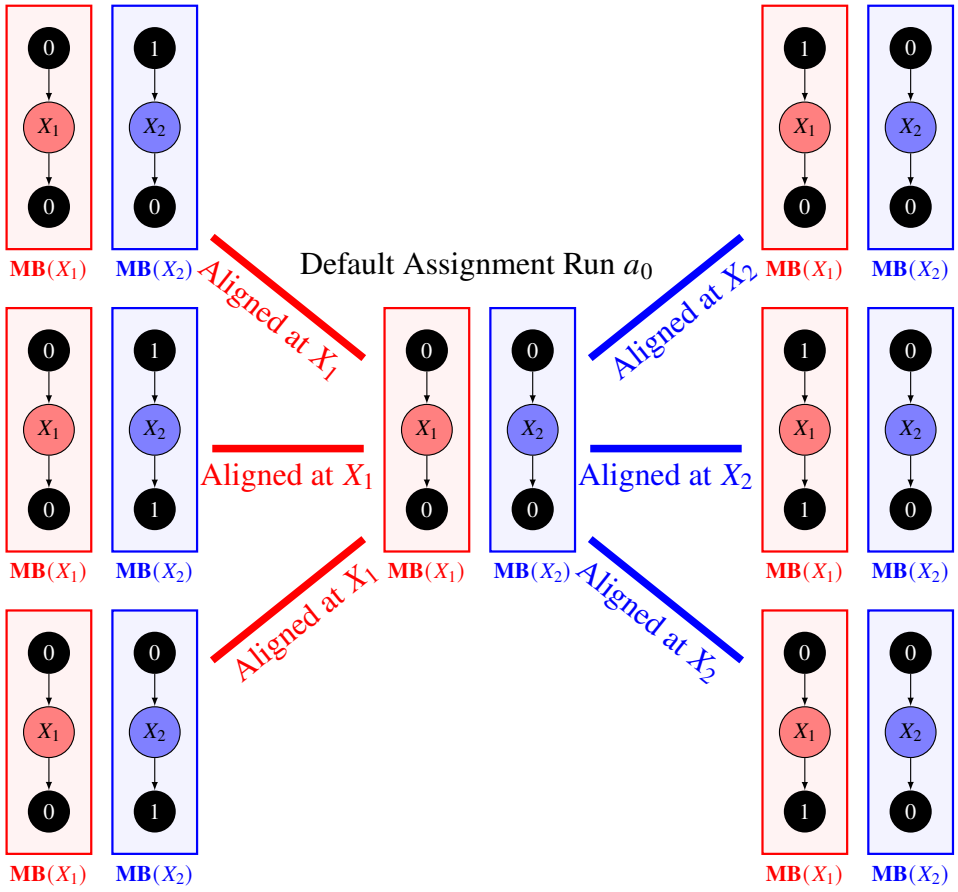


Figure 4.3: An alignment spanning tree of the default assignment a_0 ($\mathbf{COND}^{a_0}$ arbitrarily assigns all Markov boundaries to 0) and six other central runs. The runs on the left cover all possible assignments to $\mathbf{MB}(X_2) \in \{(0,0),(0,1),(1,0)\}$, while maintaining the default assignment to $\mathbf{MB}(X_1)$ to allow alignment with a_0. The right runs similarly cover all possible assignments to $\mathbf{MB}(X_1)$, aligned at X_2.

See Figure 4.3 for an example of a set of central runs and a visualization of how they are alignable.

The central runs provide a backbone for easily guaranteeing alignment. The runs in $\mathcal{A}_Y$ made up of the following two types of perturbations to the independent set (with $\mathbf{COND}^a$ always defined as the union of the Markov boundaries of the independent set, as in Definition 6):

1. For each $Y \in \mathbf{MB}(X_i)$ for some $X_i \in \mathbf{X}$, we exclude X_i from the independent set to form $\mathbf{I}^{a_Y} = \mathbf{X} \cup \{Y\} \setminus \{X_i\}$.[6]

2. For each $Y \notin \mathbf{MB}(\mathbf{X}) \cap \mathbf{X}$, $\mathbf{I}^a = \mathbf{X} \cup \{Y\}$.

[6]This is a well-formed run since $Y \in \mathbf{MB}(X_i) \implies X_j \notin \mathbf{MB}(Y)$ for any $j \neq i$ by Observation 1.

For either independence set we will form $2^{|\mathbf{PA}(Y)|}$ runs each associated with a single assignment to $\mathbf{pa}^a(Y)$, with the remaining variables in $\mathbf{COND}^a \cap \mathbf{COND}^{a_0}$ conditioned on their defaults given by f^{a_0}. Any other assignments to variables in $\mathbf{COND}^a$ can be chosen arbitrarily.

Algorithm 4 gives the proceedure for explicitly constructing a good set of runs from N_{MP} centers.

Algorithm 4: Building a good collection of runs

Input: Vertices $\mathbf{X} = \{X_1, \ldots, X_{3k-3}\} \subseteq \mathbf{V}$ having disjoint Markov boundaries with maximum depth $3N_{\text{mp}}$.
Output: A good collection of runs $\mathcal{A}$.
Let a_0 be a run with $\mathbf{I}^{a_0} = \{X_1, \ldots, X_{3k-3}\}$ and $\mathbf{COND}^{a_0}$ chosen arbitrarily.;
Set $\mathcal{A} \leftarrow \{a_0\}$.;
for $i = 1, \ldots, 3k-3$ **do**
  $\mathcal{A} \leftarrow \mathcal{A} \cup \left\{a_0[\mathbf{mb}(X_i)] : \mathbf{mb}(X_i) \in \{0,1\}^{\mathbf{MB}(X_i)}\right\}$.;
  for $Y \in \mathbf{V} \setminus \mathbf{X}$ **do**
    if $Y \in \mathbf{MB}(X)$ *for some* $X \in \mathcal{X}$ **then**
      $\mathbf{I}^a \leftarrow \mathbf{X} \cup \{Y\} \setminus \{X_i\}$;
    end
    else
      $\mathbf{I}^a \leftarrow \mathbf{X} \cup \{Y\}$;
    end
    for $\mathbf{pa}(Y) \in \{0,1\}^{|\mathbf{PA}(Y)|}$ **do**
      if $Y \in \mathbf{AN}(\mathbf{I}^a - Y)$ **then**
        $\mathbf{COND}^a \leftarrow \mathbf{MB}(\mathbf{I}^a)$;
      end
      else
        $\mathbf{COND}^a \leftarrow \mathbf{MB}(\mathbf{I}^a \setminus \{Y\}) \cup \mathbf{PA}(Y)$;
      end
      $\mathbf{COND}^a \leftarrow \mathbf{MB}(\mathbf{I}^a)$;
      $f^a(\mathbf{PA}(Y)) \leftarrow \mathbf{pa}(Y)$;
      $\mathsf{Defaults} \leftarrow \mathbf{COND}^{a_0} \cap \mathbf{COND}^a \setminus \mathbf{PA}(Y)$;
      $f^a(\mathsf{Defaults}) \leftarrow f^{a_0}(\mathsf{Defaults})$;
      $f^a(\mathbf{COND}^a \setminus \mathbf{COND}^{a_0} \setminus \mathbf{PA}(Y))$ are chosen arbitrarily.;
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{a\}$, where a is given by $a = (\mathbf{I}^a, f^a)$.;
    end
  end
end

Lemma 5. *Given a set* $\mathbf{X} = \left\{X_1, \ldots, X_{N_{mp}}\right\} \subseteq \mathbf{V}$ *of* N_{mp} *variables with depth at most* $3N_{mp}$ *and disjoint Markov boundaries, Algorithm 4 finds a good collection of*

runs $\mathcal{A}$, satisfying $|\mathcal{A}| = O(2^\gamma n)$ where $\gamma = \max_{V \in \mathbf{V}} |\mathbf{MB}(V)|$.

Claim 1. By construction, $\mathcal{A}_C$ covers $\mathbf{X}$ and is alignable.

Claim 2. A_Y covers $\mathbf{V} \setminus \mathbf{X}$ and each run in $\mathcal{A}_Y$ can be aligned with some run in $\mathcal{A}_C$.

Proof. The fact that $\mathcal{A}_Y$ covers $\mathbf{V} \setminus \mathbf{X}$ follows immediately from $\mathcal{A}_Y$ containing a run assigning for independent variable $Y \in \mathbf{V} \setminus \mathbf{X}$ each possible assignment $\mathbf{pa}(Y)$. Fix any run $a \in \mathcal{A}_Y$ with $\mathbf{I}^a = \mathbf{X} - X_i + Y$ or $\mathbf{I}^a = \mathbf{X} + Y$ depending on whether Y overlaps with $\mathbf{MB}(\mathcal{X})$. Now if $\mathbf{MB}(Y) \cap \mathbf{MB}(X_j) = \emptyset$ for any $j \neq i$, a and a_0 are aligned at X_j. If instead $\mathbf{MB}(Y) \cap \mathbf{MB}(X_j) \neq \emptyset$ for all $j \neq i$, pick any j and consider the central run $a_0[\mathbf{mb}^a(X_j)] \in \mathcal{A}_C$. Clearly, a and $a_0[\mathbf{mb}^a(X_j)]$ are aligned at X_j. In either case, we've aligned a to a run in $\mathcal{A}_C$. □

Claim 3. Every run $a \in \mathcal{A}$ is at least N_{mp}-independent.

Proof of Lemma 5. This follows immediately from Claims 1, 2, and 3 □

Degree bounds

We can ensure that N_{mp} centers can be found on certain degree-bounded graphs, which in turn bound γ. Let Δ_{in} upper bound on the in-degree of any vertex in $\mathfrak{G}$ and let Δ_{out} upper bound the out-degree. Then

$$\gamma \leq \Delta_{\text{in}} + \Delta_{\text{out}} + \Delta_{\text{out}}(\Delta_{\text{in}} - 1) = \Delta_{\text{in}} + \Delta_{\text{out}}\Delta_{\text{in}}.$$

If we have a bound, Δ, on the degree of the undirected skeleton of $\mathfrak{G}$, we get that

$$\gamma \leq \Delta(\Delta - 1) = \Delta^2 - \Delta.$$

Corollary 1. If either of the following conditions hold, we can find N_{mp} centers for $\mathfrak{G}$ with depth at most $3N_{\text{mp}}$:

1. $n \geq N_{\text{mp}}(\Delta_{\text{in}}^2 + 2\Delta_{\text{out}}\Delta_{\text{in}} + \Delta_{\text{out}}^2\Delta_{\text{in}}^2 - \Delta_{\text{in}} - \Delta_{\text{out}} + 1) = N_{\text{mp}} \cdot O(\Delta_{\text{out}}^2\Delta_{\text{in}}^2)$.
2. $n \geq N_{\text{mp}}(\Delta^4 - 2\Delta^3 + \Delta + 1) = N_{\text{mp}} \cdot O(\Delta^4)$.

Proof of Lemma 2. This follows immediately from Corollary 1 and Lemma 5. □

Limiting the depth of unzipping

As currently given, our algorithm may require Bayesian unzipping steps equal to the depth of the graph. We can bound accumulated errors from this process by limiting the depth of the vertices that need to be unzipped.

Recall that the goal of the conditioning set of each run is to d-separate each of the vertices in the independent set. For a topological ordering on the independence set, notice that we need not condition on the descendants of the deepest vertices in order to d-separate them from the others. Conveniently, avoiding conditioning on these vertices descendants leaves the output of the k-MixProd oracle in the desired form. We call these deepest vertices "bottom" vertices.

Definition 16 (Bottom vertices in a run)**.** Given vertices $\mathbf{I}$, we'll define the set of bottom vertices of the run to be the subset $\mathbf{BOT}(\mathbf{I}) \subseteq \mathbf{I}$ of vertices with maximal depth among the vertices in $\mathbf{I}$. That is $d(B) = \max_{I \in \mathbf{I}} d(I)$ for all $B \in \mathbf{BOT}(\mathbf{I})$.

We can now update the conditioning sets for our definition of runs:

$$\mathbf{COND}^a := \bigcup_{I \in \mathbf{I}^a \setminus \mathbf{BOT}(\mathbf{I}^a)} \mathbf{MB}(I) \bigcup_{B \in \mathbf{BOT}(\mathbf{I}^a)} \mathbf{PA}(B) \tag{4.4}$$

We append two additional requirements for a good collection of runs (Definition 13).

- no vertex appears both as a bottom vertex and a non-bottom vertex, and
- every non-bottom vertex has depth at most $3N_{\text{mp}}$.

Note that because we only need independent sets of size N_{mp}, it is trivial to limit the depth of our non-bottom vertices to $3N_{\text{mp}}$.

4.6 Conclusion

We have developed the first algorithm for identifying the parameters of k-MixBND mixtures. This algorithm allows us to access the probability distributions within each source — equivalent to the probability distribution conditioned on a universal and unobserved confounder. With access to this conditional distribution, the confounding of U can be adjusted for, opening up the opportunity for causal inference despite universal confounding.

The algorithm presented here is intended as an identifiability result. We hope this algorithm and framework can serve as a springboard for understanding how solutions to the k-MixProd and k-MixBND problems are intimately related.

The alignment process is highly nontrivial, explaining why papers such as Anandkumar, D. Hsu, et al., 2012 made crude assumptions (such as a conveniently independent variable) to simplify this process. The formal development of a notion of a run, while tedious, will allow for further improvements to give better "good sets of runs." Graph-specific sets of runs can also be optimized further, as demonstrated in the published version of this chapter, S. Gordon, B. Mazaheri, Yuval Rabani, et al., 2023.

Chapter 5

STRUCTURE LEARNING UNDER GLOBAL CONFOUNDING

This chapter will present an algorithm for causal discovery under latent global confounding using the LLC assumption. This work is not yet published.

5.1 Introduction

Structural Causal Models Many modern approaches to studying causal systems use structural causal models (SCMs) to graphically model causal relationships in a directed acyclic graph (DAG) (Judea Pearl, 2009; Peters, Janzing, and Schölkopf, 2017). In an SCM, $A \to B$ indicates "A has a direct effect on B." These graphical models provide a systematic way of determining covariate adjustments to indentify the effects of interventions.

"Causal discovery" is the task of recovering a causal DAG from data. The simplest algorithms for causal discovery make use of a correspondence between the conditional independence of the observed variables and graphical properties of the underlying SCM. These "d-separation rules"[1] give graphical criteria for independence and dependence under an assumption known as faithfulness. For example, $A \not\perp\!\!\!\perp B$ and $A \perp\!\!\!\perp C \mid B$ is sufficient to conclude that there is no arrow between A and C. We say that B is a "separating set" for A, C and conclude that there are three possible structures: $A \to B \to C$, $A \leftarrow B \leftarrow C$ or $A \leftarrow B \to C$. Together, we say these three structures form a "Markov equivalence class," which can be described using a CPDAG, i.e. the undirected skeleton of the true graph and partial orientation of the edges.

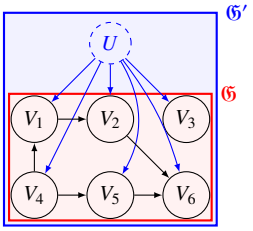


Figure 5.1: The goal is to learn the graph structure $\mathfrak{G}$ *without* observing U.

[1] See Judea Pearl (2009) for a review of d-separation or Chapter 2 for a summary of important results.

Problem Statement Suppose we augment a DAG $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$ with an unobserved U that has k "latent classes," each of which leaves a distinctive signal on all the observed variables $\mathbf{V}$. The result is a mixture-model whose DAG, $\mathfrak{G}' = (\mathbf{V} \cup \{U\}, \mathbf{E}')$ includes additional arrows from the "universal confounder" to every $V \in \mathbf{V}$. See Figure 5.1 for an example. We will refer to $\mathfrak{G}$ as the observed sub-graph and we will use $\Delta := \max_{V \in \mathbf{V}} \deg^{\mathfrak{G}}(V)$ to denote its maximum (in plus out) degree.

The goal is to to uncover the observed sub-graph's structure $\mathfrak{G}$ up to a Markov equivalence class (i.e. a CPDAG) using statistics gained from the "observed probability distribution," i.e. $\Pr(\mathbf{V})$ marginalized over U. Notice that U confounds all pairs of variables in $\mathbf{V}$, which requires it to be in *every* separating set. This means that the correspondence between conditional independence and d-separation is no longer adequate to make any structural deductions about the graph.

Assumptions We will assume (1) that the distribution is faithful[2] with respect to $\mathfrak{G}'$, and (2) that U is discrete with a known number of latent classes k. k represents the "complexity" of unobserved confounding, as more latent classes are capable of exerting "more complex" signals. Since our deductions rely upon each value of U having distinctive effects upon the observables, the most difficult case of the problem is Bernoulli observable variables. Compositions of such variables are adequate to express any finite-range variables. Hence we focus entirely upon the case of Bernoulli observables.

Causal discovery with unobserved confounding The PC algorithm was the first causal discovery algorithm to make use of conditional independence, outlined in (P. Spirtes et al., 2000). Many causal discovery algorithms have been developed since, summarized in Squires and Uhler (2023). A number of these algorithms address the presence of latent confounding using one of two assumptions that "limit" U, neither of which apply to our setting.

The first type of assumption involves limiting the *degree* of latent confounding. For example, the FCI algorithm can detect the presence of unobserved confounders that act on *only two* observed variables (Peter Spirtes, Clark Glymour, Scheines, Peter Spirtes, et al., 1993; Peter Spirtes, 2001). Richardson and Peter Spirtes (2002)'s seminal work introduced ancestral graphs for the general study of this

[2]The precise assumption is a slight extension of faithfulness to the mixture setting, discussed later.

setting. Unfortunately, this assumption cannot be applied to a global or "pervasive" confounder.

The second type of assumption is on the parametric nature of the structural equations. For example, Frot, Nandy, and Maathuis, 2019 was able to show superior performance in settings with large-degree confounders with linear relationships and additive sub-Gaussian or elliptical noise. Other settings include linear structural equations with non-Gaussian additive noise (Cai et al., 2023) and non-linear structural equations within a finite-dimensional Hilbert space (Agrawal et al., 2023). These approaches are incapable of addressing *discrete* U, $\mathbf{V}$, which have no restriction on their noise or structural equations.

Motivation

Heterogeneous Data Whenever data spans multiple populations, environments, or laboratories, the relationships within are subject to "global" or "pervasive" confounding (S. Gordon, B. Mazaheri, Yuval Rabani, et al., 2023). In these settings, true causal relationships become entangled with spurious indicators of population (or lab source, etc.) membership. While these data-sources are sometimes observed, naive adjustments for global confounding require conditioning on the data-source, which limits the power of a joined dataset. Recoverying a smaller subset of latent "classes" of data-sources is therefore essential to the problem of data-fusion (Castanedo, 2013). This task involves learning a mixture model, ie. uncovering "mixture weights" $\Pr(U)$ and "source distributions" $\Pr(\mathbf{V} \mid U)$.

Mixture Models Most work on mixture models deals with parametric distributions. The simplest example is a Gaussian mixture, which relies on the parametric assumption of Gaussianity to recover clusters of points. Without this assumption, there is no way to tell the difference between a mixture of two mono-modal (one hump) distributions versus a single bi-modal (two humps) distribution.

Discrete data poses an interesting challenge because categorical distributions are non-parametric. Notice that the statistics gathered from a single unbiased coin are exactly the same as those gathered from two biased coins with the same average bias (e.g. $1/3$ and $2/3$), so long as a we are limited to a *single sample* from each choice of coin. Almost all of the research in this setting leverages an assumption of mutually independent observed variables within the source distributions (Cryan, L. Goldberg, and P. Goldberg, 2001; Freund and Mansour, 1999; Chaudhuri and Rao, 2008; J. Feldman, R. O'Donnell, and R. A. Servedio, 2008; Y. Rabani, L. J. Schulman, and

Swamy, 2014; Chen and A. Moitra, 2019; S. Gordon, B. H. Mazaheri, et al., 2021; Spencer L. Gordon et al., 2023) (see also the earlier seminal work of Kearns et al. (1994)).

k-MixProd amounts to a graphical assumption, i.e. that the causal model within each source is an empty graph. Early work by Anandkumar, D. Hsu, et al. (2012) and S. Gordon, B. Mazaheri, Yuval Rabani, et al. (2023) broadened this class of independence assumptions for mixture identifiability, exploring Markov random fields and Bayesian networks respectively.

A key assumption of (S. Gordon, B. Mazaheri, Yuval Rabani, et al., 2023) and k-MixProd is that the Bayesian DAG structure within each mixture component is *known*. To date there is no result showing if and when the within-source DAG structure can be identified from the observed data of a mixture model. If we are capable of recovering graphical structure in this setting, then we can use that structure to recover the mixture without ever making parametric or structural assumptions.

Other Related Work Other work has investigated a different setting in which *different DAG structures* are mixed (Saeed, Panigrahi, and Uhler, 2020). This work relies on the preservation of some local conditional independence properties to learn a "union graph."

Main Result

This chapter will give the first proof of the identifiability of causal structures within the discrete mixture setting by developing the first known algorithm for this problem.

Theorem 3. *Consider* $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$ *with* $\Omega(\log(k)\Delta^3)$ *vertices, mixture source* $U \in \{1, \ldots, k\}$ *and degree bound* Δ. $\mathfrak{G}$ *is generically*[3] *identifiable up to its Markov equivalence class.*

Using an oracle that can solve k-MixProd in time τ and an oracle that can solve for non-negative rank in time ρ we give an algorithm that runs in time $|\mathbf{V}|^{\mathcal{O}(\Delta^2 \log(k))}\,\rho + \mathcal{O}(k\,|\mathbf{E}|\,2^{\Delta^2})\tau$.

The non-negative rank of a $k+1 \times k+1$ matrix (as used for our algorithm) can be solved in time $k^{\mathcal{O}(k^2)}$ (Ankur Moitra, 2016). In the absence of non-negative rank tests, Anandkumar, D. Hsu, et al., 2012 demonstrated that regular rank tests generally work well in place of non-negative rank tests in practice. We will develop a hypothesis

[3] with Lebesgue measure 1 in the space of observed moments.

test for matrix-rank that requires $\mathcal{O}(k^6)$ operations to invert a $(k+1)^2 \times (k+1)^2$ covariance matrix of the estimated elements of our $k+1 \times k+1$ matrix.

Our solution also requires solving k-MixProd on 3 variables of cardinality $\mathcal{O}(k)$, which corresponds to decomposing a $\mathcal{O}(k) \times \mathcal{O}(k) \times \mathcal{O}(k)$ tensor into rank 1 components. When the rank of such a tensor is known to be linear in k, this decomposition can be solved in $O(k^{6.05})$ (J. Ding et al., 2022), though S. Gordon, B. H. Mazaheri, et al., 2021 showed that the problem generally suffers from instability, with sample complexity exponential in k. This step may be considered optional, as it is used to refine a small number of "false-positive" adjacencies confined to a provably small subset of the DAG.

Our algorithm will build on two key ideas. The first of these ideas is that the rank of a matrix formed by the joint probabilities of two variables contains information about the graphical structure on those variables under latent class confounding. We observe that the joint probability distribution of two independent variables of cardinality ℓ forms an $\ell \times \ell$ matrix that can be written as a rank 1 outer-product. Hence, marginalizing over a k-mixture gives us a linear combination of these rank 1 matrices, which will generically be rank k.

Of course, finite data only affords a stochastic perturbation of the true matrix of joint probabilities. A naive approach to testing the rank of the underlying joint probability distribution involves thresholding singular values as in Anandkumar, D. Hsu, et al. (2012). Such an approach is unstable, with many practical difficulties associated with selecting the correct threshold. Instead, we modify a statistical test based on Ratsimalahelo (2001) to obtain a hypothesis test for the rank of an estimated joint probability matrix. We demonstrate the superiority of this approach and note that this test may be of fundamental use in other mixture settings.

The idea of using matrix rank is limited by the cardinality of our variables — the joint probability distribution of two binary observables forms a 2×2 matrix whose rank is no larger than 2. This problem leads to the second idea: We can "coarsen" variables by joining sets of variables. These "super variables" take on values in the Cartesian product of their components' alphabets. For example, three Bernouli variables can be combined into a single variable of cardinality 8. This chapter develops the notion of testing rank on these coarsened variables for causal discovery.

Outline of Algorithm

The PC-algorithm works in two phases. The first phase begins with a complete graph (i.e. all possible edges) and uses conditional independence tests to find non-adjacencies and "separating sets" that d-separate them. When a non-adjacency is found, the corresponding edge is removed and its separating set is stored. The second phase uses the separating sets from the first phase to orient immoralities (as well as further propagation of edge orientation via Meek rules).

Our algorithm will mirror the PC Algorithm, but will split the first phase into two parts, yielding three total phases. Phase I will again begin with a complete graph and remove edges between variables when we find evidence of non-adjacency (this time using rank tests).

Phase I will only test independence on groupings of variables, so its termination will not guarantee that we have discovered all possible non-adjacencies. Instead, a provably small subset of the graph will have *false positive* edges. In Phase II, we will make use of the structure we have uncovered so far to induce instances of k-MixProd within conditional probability distributions. A k-MixProd solver will then identify the joint probability distribution between subsets of $\mathbf{V}$ and the latent class U. Access to this joint probability distribution enables a search for separating sets that include U, meaning that the rest of the structure can be resolved using traditional conditional independence tests (following the standard steps of the PC algorithm).

Phase III mirrors the last phase of the PC algorithm: identifying immoralities using non-adjacencies and separating sets, and then propagating orientations according to Meek rules. This phase is no different from the PC algorithm, so we will not discuss this phase in detail.

5.2 Additional Background

Judea Pearl, 1988 uses structural causal models to justify the *local Markov condition*, which means that d-separation always implies independence and allows DAG structures to be factorized. It is possible that two d-connected variables by chance exhibit some unexpected *statistical* independence. This complication is often assumed away using "faithfulness" (Peter Spirtes, Clark Glymour, Scheines, and David Heckerman, 2000), which ensures that d-connectedness implies statistical dependence. Together, the local Markov condition and faithfulness give a correspondence between statistical dependence and the graphical conditions of the causal DAG which can be leveraged for causal structure learning.

The following fact will be useful when building separating sets.

Lemma 6 (Judea Pearl, 2009)**.** *If vertices V_i, V_j are nonadjacent in $\mathfrak{G}$, either $\mathbf{PA}^{\mathfrak{G}}(V_i)$ or $\mathbf{PA}^{\mathfrak{G}}(V_j)$ are a valid separating set for V_i, V_j.*

We will often want to bound the cardinality of separating sets relative to the degree bound of the graph (Δ). When dealing with a separating set between two vertices V_i, V_j, Lemma 6 implies a simple upper bound of Δ. Separating sets for *sets* (or coarsenings) of vertices are significantly more complicated because conditioning may d-separate some pairs of vertices while d-connecting others.

To unify the treatment of separating sets, we will make use of **moral graphs**, which can be thought of as undirected equivalents of DAGs (Lauritzen et al., 1990). We will denote the moral graph of $\mathfrak{G}$ as $\mathfrak{G}^{(m)}$. To transform $\mathfrak{G}$ into $\mathfrak{G}^{(m)}$, we add edges between all immoralities, i.e. nonadjacent vertices with a common child, sometimes called an unsheilded collider. After this, we change all directed edges to undirected edges.

A very useful fact from Lauritzen et al., 1990 (also Eq. 1 in Acid and De Campos, 1996) is that all separating sets $\mathbf{C} \subseteq \mathbf{V}$ for $\mathbf{S}, \mathbf{S}' \subseteq \mathbf{V}$ in $\mathfrak{G}$ are also separating sets in $(\mathfrak{G}[\mathbf{AN}^+(\mathbf{S} \cup \mathbf{S}' \cup \mathbf{C})])^{(m)}$. This transforms complicated active path analysis into simple connectedness arguments on undirected moral graphs (of special subgraphs of $\mathfrak{G}$). A convenient consequence of this transformation, which we will use throughout the paper, is Lemma 7.

Lemma 7. *If $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$ has degree bound Δ, then the size of a separating set between $\mathbf{S}, \mathbf{S}' \subseteq \mathbf{V}$ is no larger than* $\min(|\mathbf{S}|, |\mathbf{S}'|)\Delta^2$.

Lemma 7 is a consequence of the maximum increase in the degree of the moral graph.

Proof. A key observation is that the moral graph of a subgraph of $\mathfrak{G}$ has no additional edges relative to $\mathfrak{G}^{(m)}$. That is, if $\mathfrak{G}[\mathbf{W}] = (\mathbf{W}, \mathbf{F})$ is a subgraph of $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$ then the corresponding edge-sets of the moral graphs obey $\mathbf{F}^{(m)} \subseteq \mathbf{E}^{(m)}$ because adding vertices cannot have invalidated any previously contained immoralities.

Abbreviate $(\mathfrak{G}[\mathbf{AN}^+(\mathbf{S} \cup \mathbf{S}' \cup \mathbf{C})])^{(m)}$ as $\mathfrak{G}_{\mathbf{C}}^{(m)}$. Even though we do not know what $\mathbf{C}$ is, we know that the 1-neighborhood of $\mathbf{S}$ in $\mathfrak{G}_{\mathbf{C}}^{(m)}$ suffices as a separating set. $\mathfrak{G}^{(m)}$ has all of the edges of $\mathfrak{G}_{\mathbf{C}}^{(m)}$, so

$$\mathbf{NB}_1^{\mathfrak{G}_{\mathbf{C}}^{(m)}}(\mathbf{S}) \subseteq \mathbf{NB}_1^{\mathfrak{G}^{(m)}}(\mathbf{S}). \tag{5.1}$$

Note that $\mathbf{NB}_1^{\mathfrak{G}^{(m)}}(\mathbf{S})$ is not necessarily a separating set for $\mathbf{S}, \mathbf{S}'$ in $\mathfrak{G}^{(m)}$, in fact it may include some vertices in $\mathbf{S}$ itself. However, the size of the separating set is bounded by $\left|\mathbf{NB}_1^{\mathfrak{G}^{(m)}}(\mathbf{S})\right|$, which is no larger than $|\mathbf{S}|\,\Delta^2$. As we chose $\mathbf{S}$ arbitrarily, this bound also holds for $\mathbf{S}'$. □

Another helpful result from moral graphs is Lemma 8, which will help us when we prove the existence of separating sets.

Lemma 8 (Corollary of Theorem 1 in Acid and De Campos, 1996)**.** *For DAG $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$, and $\mathbf{S}, \mathbf{S}' \subseteq \mathbf{V}$, separating sets $\mathbf{S}, \mathbf{S}$ in $(\mathfrak{G}[\mathbf{AN}^+(\mathbf{S} \cup \mathbf{S}')])^{(m)}$ are also separating sets in $\mathfrak{G}$.*

5.3 Rank Tests

This section will develop "rank tests" which will serve as a replacement for conditional independence tests as a test for d-separation or d-connectedness.

Checking for k-Mixture Independence

To determine non-adjacency, we will take advantage of a signature U leaves on the marginal probability distributions of variables which are independent conditional on U. First, we interpret the marginal probability distribution as a matrix.

Definition 17. Given two discrete variables $X, Y \in \mathbf{V}$ each with $|X| = |Y| = m$, define the "probability matrix" $\mathbf{M}[X, Y] \in [0, 1]^{m \times m}$ to be

$$\mathbf{M}[X, Y]_{x,y} := \Pr(x, y), \tag{5.2}$$

where x, y both range from $1, \ldots, m$. Similarly, for $\mathbf{C} \subseteq \mathbf{V}$, define

$$\mathbf{M}[X, Y \mid \mathbf{c}]_{x,y} := \Pr(x, y \mid \mathbf{c}). \tag{5.3}$$

We now notice that we can decompose the probability matrix into a linear combination of conditional probability matrices for each source, for which Lemma 9 gives an upper bound on rank.

Lemma 9. *Given a mixture of Bayesian network distributions that are Markovian in $\mathfrak{G}$, if $X \perp\!\!\!\perp_d^{\mathfrak{G}} Y \mid \mathbf{C}$, then for all $\mathbf{c}$, $\mathrm{rk}_+(\mathbf{M}[X, Y \mid \mathbf{c}]) \leq k$* [4].

[4] $\mathrm{rk}_+(\mathbf{M})$ will denote the non-negative rank of matrix $\mathbf{M}$.

Proof. We can decompose $\mathbf{M}[X, Y \mid \mathbf{c}]$ as follows:

$$\mathbf{M}[X, Y \mid \mathbf{c}] = \sum_u \Pr(u)\mathbf{M}[X, Y \mid \mathbf{c}, u]. \tag{5.4}$$

$X \perp\!\!\!\perp Y \mid \mathbf{C}, U$, so $\mathbf{M}[X, Y \mid \mathbf{c}, u]$ can be written as the outer product of two vectors describing the probabilities of each variable. Therefore, we conclude that $\mathrm{rk}_+(\mathbf{M}[X, Y \mid \mathbf{c}, u]) = 1$. If $U \in [k]$, then $\mathrm{rk}_+(\mathbf{M}[X, Y \mid \mathbf{c}]) \leq k$. □

Having shown that d-separation in $\mathfrak{G}$ upper bounds the rank of probability matrices, we now seek a lower bound on the rank in the case of d-connectedness. This will require a "faithfulness-like" assumption that the dependence exerts some noticeable effect between the two variables. Lemma 10 shows that such a condition holds generically.

Lemma 10. *Given a mixture of Bayesian network distributions, each of which is faithful to* $\mathfrak{G}$*. If* $X \not\perp\!\!\!\perp_d^{\mathfrak{G}} Y \mid \mathbf{C}$ *and* $|X| = n$, $|Y| = m$ *with* $n, m > k$*, then for all* $\mathbf{c}$*,* $\mathrm{rk}_+(\mathbf{M}[X, Y \mid \mathbf{c}]) > k$ *with Lebesgue measure* 1.

The proof of Lemma 10 (see Section 5.8) involves applying faithfulness to each component of the decomposition in Equation 5.4. Lemma 9 and Lemma 10 provide a generically necessary and sufficient condition for detecting $V_i \perp\!\!\!\perp_d^{\mathfrak{G}} V_j$.

Lemma 11 (Rank Test)**.** *For* V_i, V_j *with cardinality* $> k$*,* $V_i \perp\!\!\!\perp_d^{\mathfrak{G}} V_j \mid \mathbf{C}$ *if and only if (generically)* $\mathrm{rk}_+(\mathbf{M}[V_i, V_j \mid \mathbf{C}]) \leq k$.

Hypothesis test for rank

We can build a hypothesis test for the null-hypothesis that $\mathbf{A} \coloneqq \mathbf{M}[V_i, V_j \mid \mathbf{C}]$ is has rank $\leq k$. First, decompose $\mathbf{A}$ according to the SVD $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{U}^\top$. If $\mathbf{A} \in \mathbb{R}^{m \times m}$ is rank k, we will have $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{k \times m}$. When this decomposition is done on the empirical matrix, $\hat{\mathbf{A}}$, we define $\mathbf{U}_2$ and $\mathbf{V}_2$ to be the $k+1$th to mth "extra" rows of $\mathbf{U}$ and $\mathbf{V}$ respectively, and $\mathbf{L}$ to be the diagonal matrix with the $k+1$th through mth singular values. Vectorize $\mathbf{L}$ to $\hat{l}$ by stacking the columns of $\mathbf{L}$. Let Σ be the covariance matrix of the entries of $\mathbf{A}$ with similar stacking, i.e.

$$\Sigma_{i+jm, i'+j'm} \coloneqq \mathrm{Cov}(\mathbf{A}_{ij}, \mathbf{A}_{i'j'}). \tag{5.5}$$

Now, with $\hat{\Sigma}^\dagger$ indicating the Moore-Penrose pseudoinverse of $\hat{\Sigma}$ and $\otimes$ indicating the Kronecker product, define

$$\hat{\mathbf{Q}}^\dagger \coloneqq (\mathbf{V}_2^\top \otimes \mathbf{U}_2^\top)\hat{\Sigma}^\dagger(\mathbf{V}_2 \otimes \mathbf{U}_2). \tag{5.6}$$

According to Ratsimalahelo (2001), if f is the rank of Σ, then $N\hat{l}^\top \hat{\mathbf{Q}}^\dagger \hat{l}$ converges to χ^2_f under N samples.

We can use this test statistic with a simplified $\hat{\Sigma}$ by noting that $\hat{\mathbf{A}}$ for a single data-point is 0 for all but a single entry, which is 1. This means that the variance of an entry is given by the variance of a Bernoulli random variable and the covariance is given by the expanding into three cases: both are 0, the first entry is 1, the second entry is 1. This simplifies to

$$\mathrm{Cov}(\mathbf{A}_{ij}, \mathbf{A}_{i'j'}) = \begin{cases} \mathbf{A}_{ij}(1 - \mathbf{A}_{ij}) & \text{if } i = i', j = j' \\ -\mathbf{A}_{ij}\mathbf{A}_{i'j'} & \text{otherwise.} \end{cases} \tag{5.7}$$

Using this estimate for $\hat{\Sigma}$ gives us a hypothesis test that is specifically designed to implement the rank test from this section. This approach is tested and compared to thresholding the k + 1th singular value in Section 5.7.

5.4 Algorithm Phase I

We will now outline the first two phases of our algorithm, leaving out the final phase of orienting edges with respect to Meek's rules. The first phase of our algorithm involves coarsening sets of variables and applying rank tests. We then analyze the output of the first phase $\mathfrak{G}_1$ which has removed *most but not all* of the missing edges. We define FP edges as these "leftovers" and show that they are contained within a subset of bounded size. This fact allows us to resolve all of the non-adjacencies in $\mathfrak{G}$ by setting up instances of k-MixProd in Phase II.

Coarsened Rank Tests

Lemma 11 allows for a simple generalization of classical structure-learning algorithms (such as the PC algorithm) provided that our probability matrix $\mathbf{M}[X, Y]$ is large enough. Unfortunately, categorical variables ranging over smaller alphabets (such as the binary alphabets addressed by this chapter) do not contain sufficient information to detect non-adjacency in cases of larger k. We resolve this problem by coarsening sets of small-alphabet (binary) variables into supervariables of larger cardinality.

Definition 18. Consider DAG $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$, $V_i, V_j \in \mathbf{V}$, and sets $\mathbf{S}_i, \mathbf{S}_j \subseteq \mathbf{V} \setminus \{V_i, V_j\}$. We call the ordered pair $(\mathbf{S}_i^+, \mathbf{S}_j^+) = (\mathbf{S}_i \cup \{V_i\}, \mathbf{S}_j \cup \{V_j\})$ an **independence preserving augmentation (IPA)** of (V_i, V_j) if, for some **IPA conditioning set** $\mathbf{C} \subset \mathbf{V}$,

$$\mathbf{S}_i^+ \perp\!\!\!\perp_d^{\mathfrak{G}} \mathbf{S}_j^+ \mid \mathbf{C}.$$

The creation of supervariables allows us to use conditional rank tests in the place of conditional independence tests. This leads to a modified version of the PC algorithm that searches over pairs of supervariable coarsenings instead of pairs of vertices, given in Algorithm 5.

Algorithm 5: Phase I

Input: The marginal probability distribution $\Pr(\mathbf{V})$, marginalized over U.
Output: An undirected graph $\mathfrak{G}_1 = (\mathbf{V}, \mathbf{E}_1)$ and a separating set $\mathbf{C}_{ij}$ for each detected non-adjacency.

Begin with a complete undirected graph $\mathfrak{G}_1 = (\mathbf{V}, \mathbf{V} \times \mathbf{V})$ and $d_{\max} \leftarrow |\mathbf{V}| - 1$.
for $\ell = 0$ *to* $\ell = d_{\max}$ **do**
  for $\mathbf{C} \subset \mathbf{V}$ *and* $|\mathbf{C}| = \ell$ **do**
    for $\mathbf{S}, \mathbf{S}' \subseteq \mathbf{V} \setminus \mathbf{C}$, *with* $|\mathbf{S}| = |\mathbf{S}'| = \lceil \lg(k) \rceil + 1$ **do**
      if *arbitrary assignment* $\mathbf{c}$ *has* $\mathrm{rk}_+(\mathbf{M}[\mathbf{S}, \mathbf{S}' \mid \mathbf{c}]) \leq k$ **then**
        Remove all edges between $\mathbf{S}$ and $\mathbf{S}'$ in $\mathfrak{G}_1$.
        $\mathbf{C}_{i,j} \leftarrow \mathbf{C}$ for each $V_i \in \mathbf{S}, V_j \in \mathbf{S}'$
        Update $d_{\max}$ to the maximum degree of $\mathfrak{G}_1$.
      end
    end
  end
end

Lemma 12. *Algorithm 5 utilizes* $|\mathbf{V}|^{\mathcal{O}(\Delta^2 \log(k))}$ *non-negative rank tests.*

Proof. Lemma 7 tells us that the maximum size of a separating set, is $\alpha := (\lceil \lg(k) \rceil + 1)\Delta^2$, so we need to check $\binom{|\mathbf{V}|}{\alpha} + \binom{|\mathbf{V}|}{\alpha - 1} + \ldots + \binom{|\mathbf{V}|}{1}$ possible separating sets, which is $|\mathbf{V}|^{\mathcal{O}(\Delta^2 \log(k))}$. We must iterate over all possible supervariables for each separating sets, which is upper bounded by $\binom{|\mathbf{V}|}{2(\lceil \lg(k) \rceil + 1)}$, which is $|\mathbf{V}|^{\mathcal{O}(\log(k))}$. □

FP Edges

Phase I of our algorithm removes an edge between two non-adjacent variables through a rank test so long as there exists an IPA for the non-adjacency. Not all non-adjacencies will contain an IPA, so the adjacency graph $\mathfrak{G}_1$ contains a *superset* of the true adjacencies.

Definition 19. $\mathbf{E}_1 \setminus \mathbf{E}$ are **false positive (FP)** edges.

To see why FP edges exist, we will introduce the notion of immoral descendants.

Definition 20 (Immoral Descendants)**.** For non-adjacent V_i, V_j the **immoral descendants** of V_i and V_j are their co-children (often called immoralities (Judea Pearl, 2009)) and those children's descendants.

$$\mathbf{FB}(V_i, V_j) := \mathbf{CH}(V_i, V_j) \cup \mathbf{DE}(\mathbf{CH}(V_i, V_j)) \tag{5.8}$$

Observation 3. Any set $\mathbf{C}$ such that $V_i \perp\!\!\!\perp_d V_j \mid \mathbf{C}$ must be disjoint from $\mathbf{FB}(V_i, V_j)$.

Clearly, an IPA conditioning set $\mathbf{C}$ will need to avoid the immoral descendants $\mathbf{FB}(V_i, V_j)$ in order to preserve $V_i \perp\!\!\!\perp V_j \mid \mathbf{C}$. Lemma 13 will show that $\mathbf{S}_i, \mathbf{S}_j$ must also avoid $\mathbf{FB}(V_i, V_j)$.

Lemma 13. *All IPAs for V_i, V_j are disjoint from the $\mathbf{FB}(V_i, V_j)$. That is, $\mathbf{FB}(V_i, V_j) \cap \mathbf{S}_i^+ = \emptyset$ and $\mathbf{FB}(V_i, V_j) \cap \mathbf{S}_j^+ = \emptyset$ for all IPAs $(\mathbf{S}_i^+, \mathbf{S}_j^+)$ of (V_i, V_j).*

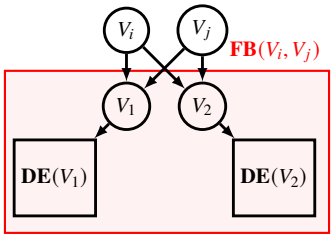


Figure 5.2: An illustration of an FP edge after Phase I due to a large set of immoral descendants. The population variable U is omitted to avoid clutter. While V_i and V_j are d-separated by $\mathcal{C} = \emptyset$ no IPA can be made because all of the leftover vertices are immoral descendants.

This illustrates that FP edges can occur as pairs of vertices with too many immoral descendants, leaving no vertices to form IPAs (shown in Figure 5.2).

Using non-descendants to form IPAs

Notice that the immoral descendants of a pair of vertices are always descendants of both V_i and V_j. To simplify our analysis, we focus on the existence of IPAs that are disjoint from the entire set of descendants. This allows us to show that FP edges only occur between "early vertices."

Definition 21. We define the early vertices,

$$\mathbf{H} := \{V \in \mathbf{V} \text{ s.t. } \left|\overline{\mathbf{DE}}(V)\right| < (2 + \Delta^2)(\lceil \lg(k) \rceil + 1)\}.$$

Lemma 14. *After Phase I (Algorithm 5), all of the false positive edges lie within the early vertices. More formally, $\mathbf{E}_1 \setminus \mathbf{E} \subseteq \mathbf{H} \times \mathbf{H}$.*

Observation 4. $|\mathbf{H}| \leq (2 + \Delta^2)\lceil \lg(k) \rceil$ and the maximum degree of $\mathfrak{G}_1$ is bounded by $|\mathbf{H}|$.

5.5 Phase II: Handle FP Edges

Recall that the marginal probability distribution cannot use independence tests to discover non-adjacency because latent variable U confounds all of the independence properties. An important observation is that the within-source distribution $\Pr(\mathbf{V} \mid u)$ would not suffer from this limitation because it would allow us to condition on separating sets that include unobserved U.

Phase II will make use of this observation by selecting subsets of variables $\mathbf{T} \subseteq \mathbf{V}$ on which to obtain $\Pr(\mathbf{T} \mid u)$ using techniques from discrete mixture models. We will then apply regular conditional independence tests on the recovered $\Pr(\mathbf{T} \mid u)$ to detect FP edges. We will use a separate $\mathbf{T}_{ij} \ni V_i, V_j$ coarsening to verify each edge $(V_i, V_j) \in \mathbf{E}_1$, though this process can likely be optimized further.

The primary result on mixture model identifiability is given by Allman, Matias, and Rhodes, 2009 as a direct consequence of a result by Kruskal, 1977.

Lemma 15 (Allman, Matias, and Rhodes, 2009)**.** *Consider the discrete mixture source $U \in \{1, \ldots, k\}$ and discrete variables X_1, X_2, X_3 with cardinality $\kappa_1, \kappa_2, \kappa_3$ respectively and $X_1 \perp\!\!\!\perp X_2 \perp\!\!\!\perp X_3 \mid U$. The mixture is generically identifiable (with Lebesgue measure 1 on the parameter space) if*

$$\min(\kappa_1, k) + \min(\kappa_2, k) + \min(\kappa_3, k) \geq 2k + 2.$$

We can again use coarsening to form X_i with large enough κ_i. The conditions for identifiability are therefore quite mild — Phase I only needs to uncover enough sparsity to d-separate three sufficiently large independent coarsenings, one of which will be $\mathbf{T}_{ij}$. Conveniently, the constrained nature of our FP edges means that the graph $\mathfrak{G}_1$ is sufficiently sparse to allow the construction of this setting.

$\mathbf{T}_{ij}$ must be designed to include enough information to discover a nonadjaceny between V_i, V_j. In other words, we need to ensure that $\mathbf{T}_{ij}$ contains a separating set $\mathbf{C} \subset \mathbf{T}_{ij}$ such that $V_i \perp\!\!\!\perp_d^{\mathfrak{G}} V_j \mid \mathbf{C}$. It turns out that augmenting V_i, V_j with their distance-1 neighborhood is enough to guarantee this requirement.

Definition 22. Given vertices V_i, V_j, let $\mathbf{T}_{ij}$ be the set containing V_i, V_j and all vertices that are distance 1 in $\mathfrak{G}_1$ from V_i or V_j.

Lemma 16. *If vertices V_i, V_j are nonadjacent, the set $\mathbf{T}_{ij}$ contains a valid separating set $\mathbf{C} \subseteq \mathbf{T}_{ij}$ such that $V_i \perp\!\!\!\perp_d V_j \mid \mathbf{C}$.*

Lemma 16 guarantees that the conditional probability distribution $\Pr(\mathbf{T}_{ij} \mid u)$ has sufficient information to verify or falsify the adjacency of V_i and V_j.

5.6 Utilizing $\mathfrak{G}_1$ to set up k-MixProd

The rest of the construction of the k-MixProd instances is left to Section 5.6. Generally, it involves ensuring that the recovered $\mathfrak{G}_1$ from Phase I is sparse enough to d-separate all $\mathbf{T}_{ij}$ from two other supervariables of sufficient cardinality. The procedure is outlined by Algorithm 6 and then Algorithm 7 performs the actual correction with the statistics recovered from k-MixProd oracles. Lemma 17 summarizes the results proved in Section 5.6.

Lemma 17. *Phase II requires $\Omega(\Delta^3 \log(k))$ vertices and solves k-MixProd no more than $\mathcal{O}(k\,|\mathbf{E}|\,2^{\Delta^2})$ times.*

The first step to recovering $\Pr(\mathbf{T}_{ij} \mid u)$ will be to select some $\mathbf{Z}_{ij}$ and recover $\Pr(\mathbf{T}_{ij} \mid u, \mathbf{z}_{ij})$ using instances of k-MixProd induced on the conditional probability distribution $\Pr(\mathbf{V} \mid \mathbf{z}_{ij})$. Recall that k-MixProd requires three independent variables of sufficient cardinality. Hence, we must find $\mathbf{X}_1, \mathbf{X}_2, \mathbf{T}_{ij}$ which are sufficiently large, and d-separated from each other by $\mathbf{Z}_{ij}$ in $\mathfrak{G}$. See Figure 5.3 for an example.

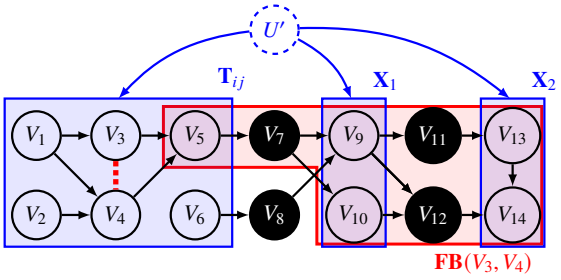


Figure 5.3: The given graph has an FP edge between V_3 and V_4, indicated by a dashed line, caused by a large set of immoral descendants (shown in red). Conditioning on V_7, V_{11}, V_{12} creates an instance of k-MixProd on $\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2$. Notice that V_7, V_{11}, V_{12} are all in $\mathbf{FB}(V_3, V_4)$, which means that the $\Pr(\mathbf{T}_{ij} \mid V_7, V_{11}, V_{12}, u')$ recovered by k-MixProd will not be sufficient for detecting the FP edge. This obstacle will be solved in Subsection 5.6.

Of course we have access to $\mathfrak{G}_1$, not $\mathfrak{G}$. $\mathfrak{G}_1$ contains no orientations[5] and may

[5]It is, in principle, possible to orient immoralities within $\mathfrak{G}_1$ at this stage, but this gives no complexity improvements.

contain extra false-positive edges. We will need to build a conditioning set $\mathbf{Z}_{ij}$ that achieves a guaranteed instance of k-MixProd nonetheless. While Markov boundaries cannot be computed without $\mathfrak{G}$, we can easily use $\mathfrak{G}_1$ to find a superset that *contains* the Markov bounary of a given vertex.

Lemma 18. *The* 2*-neighborhood of* $\mathbf{X} \subseteq \mathbf{V}$ *in* $\mathfrak{G}_1$ *contains* $\mathbf{MB}^{\mathfrak{G}}(\mathbf{X})$.

Proof. The distance between $X \in \mathbf{X}$, and $V \in \mathbf{V}$ in $\mathfrak{G}_1$ is less than or equal to the distance in $\mathfrak{G}$, because $\mathbf{E}_1 \supseteq \mathbf{E}$. This means that the 2-neighborhood of $\mathbf{X}$ in $\mathfrak{G}_1$ includes the 2-neighborhood of $\mathbf{X}$ in $\mathfrak{G}$. Furthermore because all vertices in $\mathbf{MB}^{\mathfrak{G}}(\mathbf{X})$ are distance ≤ 2 from at least one $X \in \mathbf{X}$, we have that $\mathbf{MB}^{\mathfrak{G}}(\mathbf{X})$ is contained in the 2-neighborhood of $\mathbf{X}$ in $\mathfrak{G}_1$. □

The sparsity of $\mathfrak{G}_1$ will dictate the number of necessary vertices to successfully set up k-MixProd. We will want to limit the size of $\mathbf{Z}_{ij}$ as much as possible. Fortunately, the bounds on the size of $\mathbf{H}$ mean that *most* of $\mathfrak{G}_1$ is degree bounded by Δ. We will avoid large degree vertices in $\mathbf{H}$ by strategically selecting $\mathbf{X}_1, \mathbf{X}_2$ with the smallest 2-neighborhoods. $\mathbf{Z}_{ij} = \mathbf{MB}(\mathbf{X}_1) \cup \mathbf{MB}(\mathbf{X}_2)$ will be sufficient to d-separate all three vertices, so we need not worry about potentially large degree in $\mathbf{T}_{ij}$. This process is described by Algorithm 6.

Algorithm 6: Formation of k-MixProd Instances

Input: Two vertices $V_i, V_j \in \mathbf{V}$ and $\mathfrak{G}_1 = (\mathbf{V}, \mathbf{E}_1)$ from the output of Algorithm 5.
Output: $\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2$ and $\mathbf{Z}_{ij}$ such that $\mathbf{T}_{ij} \perp\!\!\!\perp_d \mathbf{X}_1 \perp\!\!\!\perp_d \mathbf{X}_2 \mid \mathbf{Z}_{ij}$.
Let $\mathbf{T}_{ij} = \{V_i, V_j\} \cup \mathbf{NB}_1^{\mathfrak{G}_1}(V_i) \cup \mathbf{NB}_1^{\mathfrak{G}_1}(V_j)$.
$\mathbf{V}' \leftarrow \mathbf{V} \setminus (\mathbf{T}_{ij} \cup \mathbf{NB}_2(\mathbf{T}_{ij}))$
$\mathbf{X}_1, \mathbf{X}_2$ all begin as empty sets.
while $2^{|\mathbf{X}_1|} + 2^{|\mathbf{X}_2|} < 2k + 2 - \min(k, 2^{|\mathbf{T}_{ij}|})$ **do**
  Add $V \in \mathbf{V}' \setminus (\mathbf{NB}_2(\mathbf{X}_2) \cup \mathbf{X}_2)$ to $\mathbf{X}_1$.
  Add $V \in \mathbf{V}' \setminus (\mathbf{NB}_2(\mathbf{X}_1) \cup \mathbf{X}_1)$ to $\mathbf{X}_2$.
end
$\mathbf{Z}_{ij} \leftarrow \mathbf{NB}_2(\mathbf{X}_1) \cup \mathbf{NB}_2(\mathbf{X}_2)$.

Lemma 19. *Algorithm 6 terminates successfully (without running out of vertices in* $\mathbf{V}'$*) with* $\Omega(\Delta^3 \log(k))$ *vertices.*

Proof. The algorithm designates vertices in $\mathbf{V}$ into the following sets and succeeds so long as those sets are disjoint.

1. $\mathbf{X}_1$ and $\mathbf{X}_2$
2. $\mathbf{NB}_2(\mathbf{X}_1), \mathbf{NB}_2(\mathbf{X}_2)$
3. $\mathbf{T}_{ij}$
4. $\mathbf{NB}_1(\mathbf{T}_{ij})$

$\left|\mathbf{T}_{ij}\right| \geq 2$ and $k \geq 2$, so the number of vertices added to $\mathbf{X}_1, \mathbf{X}_2$ in the loop of Algorithm 6 is at most $\lceil \lg(2k + 2 - 2) \rceil < \lg(k) + 2$. To bound the 2-neighborhood, we notice that we cannot easily apply our degree bound of Δ because $\mathbf{H}$ could be a clique in $\mathfrak{G}_1$ (from FP edges). Instead, we bound

$$|\mathbf{NB}_2(\mathbf{X}_1) \cup \mathbf{NB}_2(\mathbf{X}_2) \cup \mathbf{X}_1 \cup \mathbf{X}_2| \leq \Delta^2 |\mathbf{X}_1 \cup \mathbf{X}_2| + \Delta |\mathbf{H}| \tag{5.9}$$

because the distance 1 neighborhood could include all of $\mathbf{H}$ but all additional neighborhoods are bounded by Δ. $|\mathbf{H}|$ is $\mathcal{O}(\Delta^2 \log(k))$ by Observation 4, so this bound is $\mathcal{O}(\Delta^3 \log(k))$.

The size of $\mathbf{T}_{ij}$ is the largest when including V_i or V_j in $\mathbf{H}$, for which $\mathbf{NB}_1(V_i)$ could be all of $\mathbf{H}$ and $\mathbf{NB}(V_i)$ then necessarily falls outside of $\mathbf{H}$. This worst case gives

$$\left|\mathbf{T}_{ij}\right| \leq |\mathbf{H}| + \Delta^2, \tag{5.10}$$

which is $\mathcal{O}(\Delta^2 \log(k))$. Expanding to the 1 neighborhood picks up another factor of Δ, bringing us again to $\mathcal{O}(\Delta^3 \log(k))$.

□

Aligning multiple k-MixProd runs

k-MixProd distributions are symmetric with respect to the $k!$ permutations on the label of their source. For this reason, there is no guarantee that multiple calls to a k-MixProd solver will return the same permutation of source labels.

To solve this, S. Gordon, B. Mazaheri, Yuval Rabani, et al., 2023 noticed that any two solutions to k-MixProd problems that share the same conditional probability distribution for at least one "alignment variable" can be "aligned" by permuting the source labels until the distributions on that variable match up. We will only need alignment along runs for different assignments to each $\mathbf{Z}_{ij}$, used in the next section. Explicitly, two assignments $\mathbf{z}_{ij}$ and $\mathbf{z}'_{ij}$, need least one $\mathbf{X}^* \in \{\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2\}$ such that $\mathbf{mb}_{\mathbf{z}_{ij}}(\mathbf{X}^*)$ and $\mathbf{mb}_{\mathbf{z}'_{ij}}(\mathbf{X}^*)$ are the same, in order for alignability to be satisfied.

To align sets of k-MixProd results which are not all pairwise alignable, Chapter 4 introduced the concept of an "alignable set of runs" for which chains of alignable pairs create allow alignability. We re-use this idea in the following lemma.

Lemma 20. *The set of k-MixProd instances on the same $\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2$ with all possible assignments $\mathbf{z}_{ij}$ to $\mathbf{Z}_{ij}$ is alignable.*

Proof. Any two runs with assignments $\mathbf{z}_{ij}$ and $\mathbf{z}'_{ij}$ that differ in their assignment to only one variable are alignable. Therefore, any two non-alignable runs can be aligned using a chain of Hamming-distance one alignments. □

Recovering the unconditioned within-source distribution

After all our calls to the k-MixProd oracle, we have access to $\Pr(\mathbf{T}_{ij} \mid u, \mathbf{z}_{ij})$ and $\Pr(u \mid \mathbf{z}_{ij})$ for every assignment $\mathbf{z}_{ij}$ and u. $\Pr(\mathbf{T}_{ij} \mid u, \mathbf{z}_{ij})$ is insufficient to determine the adjacency of V_i, V_j because $\mathbf{Z}_{ij}$ may contain vertices in the immoral descendants of V_i, V_j, prohibiting the discovery of a separating set within $\mathbf{T}_{ij}$.

Instead, we must recover $\Pr(\mathbf{T}_{ij} \mid u)$, which is not conditioned on $\mathbf{Z}_{ij}$. To do this, we can apply the law of total probability over all possible assignments to $\mathbf{Z}_{ij}$.

$$\Pr(\mathbf{T}_{ij} \mid u) = \sum_{\mathbf{z}_{ij}} \Pr(\mathbf{z}_{ij} \mid u) \Pr(\mathbf{T}_{ij} \mid \mathbf{z}_{ij}, u) \tag{5.11}$$

We can obtain $\Pr(\mathbf{z}_{ij} \mid u)$ by using Bayes rule on the k-MixProd output, $\Pr(u \mid \mathbf{z}_{ij})$.

$$\Pr(\mathbf{z}_{ij} \mid u) = \frac{\Pr(u \mid \mathbf{z}_{ij}) \Pr(\mathbf{z}_{ij})}{\Pr(u)}. \tag{5.12}$$

$\Pr(\mathbf{z}_{ij})$ can be obtained by counting the frequency of $\mathbf{z}_{ij}$ in the data. In addition, $\Pr(u) = \sum_{\mathbf{z}_{ij}} \Pr(\mathbf{z}_{ij}) \Pr(u \mid \mathbf{z}_{ij})$ is computable by the law of total probability after the runs for each assignment $\mathbf{z}_{ij}$, have been aligned. Equivalently, we can normalize such that $\sum_{\mathbf{z}_{ij}} \Pr(\mathbf{z}_{ij} \mid u) = 1$.

Lemma 21. *We can compute $\Pr(\mathbf{T}_{ij} \mid u)$ using known quantities,*

$$\Pr(\mathbf{T}_{ij} \mid u) = \frac{\sum_{\mathbf{z}_{ij}} \Pr(u \mid \mathbf{z}_{ij}) \Pr(\mathbf{z}_{ij}) \Pr(\mathbf{T}_{ij} \mid \mathbf{z}_{ij}, u)}{\sum_{\mathbf{z}_{ij}} \Pr(\mathbf{z}_{ij}) \Pr(u \mid \mathbf{z}_{ij})}.$$

$\Pr(\mathbf{T}_{ij} \mid u)$ is a completely deconfounded distribution on which we can run the PC-algorithm. The full procedure is given in Algorithm 7, in which we use Algorithm 6

Algorithm 7: Phase II: Detection and correction of FP edges.

Input: $\Pr(\mathbf{V})$ marginalized over U, a black box solver for k-MixProd, and $\mathfrak{G}_1 = (\mathbf{V}, \mathbf{E}_1)$ from the output of Algorithm 5.
Output: $\mathfrak{G}_2 = (\mathbf{V}, \mathbf{E}_2)$, an undirected skeleton of $\mathfrak{G}$ and separating sets for nonadjacenceis (vertices not in $\mathbf{E}_2$) .
Start with $\mathbf{E}_2 \leftarrow \mathbf{E}_1$.
for *each* $\{V_i, V_j\} \in \mathbf{E}_1$ **do**
  Retrieve $\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2, \mathbf{Z}_{ij}$ from Algorithm 6.
  for *each assignment* $\mathbf{z}_{ij}$ **do**
    Run the k-MixProd solver on $\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2$ on $\Pr(\mathbf{V} \mid \mathbf{z}_{ij})$.
  end
  Perform alignment of the $2^{\mathbf{z}_{ij}}$ runs to retrieve $\Pr(\mathbf{T}_{ij} \mid \mathbf{Z}_{ij}, U)$.
  Calculate $\Pr(\mathbf{T}_{ij} \mid u)$ for every u using Lemma 21.
  Run PC or any other structure learning algorithm on $\Pr(\mathbf{T}_{ij} \mid u)$ to find a separating set $\mathbf{C}_{ij}$ (or verify adjacency) for V_i, V_j. If $V_i \perp\!\!\!\perp V_j \mid \mathbf{C}_{ij}, u$ for all u, remove $\{V_i, V_j\}$ from $\mathbf{E}_2$ and store $\mathbf{C}_{ij}$.
end

followed by alignment and Lemma 21 in order to remove all of the false-positive edges from $\mathfrak{G}_1$.

Lemma 22. *Algorithm 7 requires solving k-MixProd $\mathcal{O}(k\,|\mathbf{E}|\,2^{\Delta^2})$ times.*

Proof. This algorithm requires running k-MixProd for every possible assignment to the conditioning set $\mathbf{D}_{ij}$, for which we have $\left|\mathbf{D}_{ij}\right| \leq (\lg(k) + 2)\Delta^2$ total binary variables. This gives an upper bound of $2k2^{\Delta^2}$ runs of k-MixProd for each edge. □

5.7 Empirical Results

The algorithm is successful when enough data is gathered, as proved by our theoretical results. We now employ three empirical tests to show the superiority of our derived hypothesis-based rank test as well as investigate the sensitivity of Phase I.

Structural Equation Setup

SCMs are made up of a graphical structure and accompanying structural equations. We focus our tests primarily on varying the graphical structure, using a standard set of structural equations on these graphs.Our U are generated using a fair coin ($k = 2$), and all other vertices are Bernoulli random variables with bias p_V determined by V's parents (including U):

$$p_V = \frac{1 + \sum_{W \in \mathbf{PA}^{\mathfrak{G}'}(V)} W}{\left|\mathbf{PA}^{\mathfrak{G}'}(V)\right| + 2}. \tag{5.13}$$

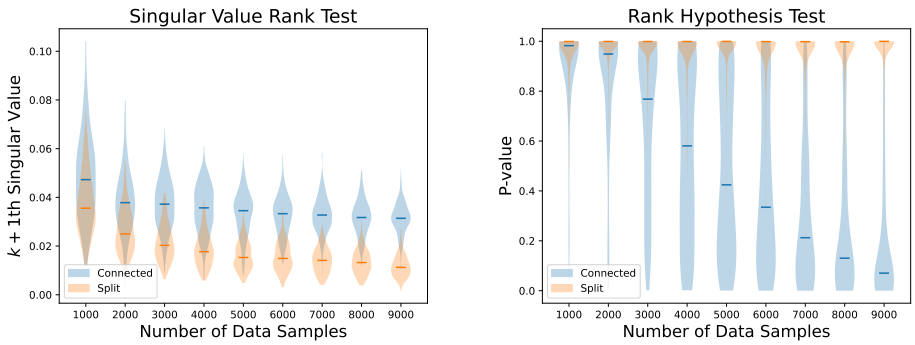


Figure 5.4: The results of Test 1.

Structural equations of this form have a reasonable strength between vertices that is decreased relative to in-degree.

Test 1: Rank Hypothesis Test vs. Singular Values

We begin with a comparison of the test developed in Section 5.3 to a naive thresholding of singular values as in Anandkumar, D. Hsu, et al. (2012). To study the differences between these tests, we generate data from two observed subgraphs.

1. "Connected" $\mathfrak{G}^c$: $V_1 \to V_2 \to V_3 \to V_4$
2. "Split" $\mathfrak{G}^s$: $V_1 \to V_2 \qquad V_3 \to V_4$

If we coarsen our vertices into $\mathbf{S}_i^+ = \{V_1, V_2\}$ and $\mathbf{S}_j^+ = \{V_3, V_4\}$, then these two DAGs differ in that $\mathbf{S}_i^+ \not\perp\!\!\!\perp_d^{\mathfrak{G}^c} \mathbf{S}_j^+$ and $\mathbf{S}_i^+ \perp\!\!\!\perp_d^{\mathfrak{G}^s} \mathbf{S}_j^+$. We note that this is a challenging test, as the connection between the two partitions is only driven by the $X_2 \to X_3$ arrow.

We varied the number of samples from these distributions from 1000 to 9000 and studied the distributions of the two reported p-values and k + 1th singular values across 200 runs. The results are reported in Figure 5.4, showing that the hypothesis test has significant difference in p-values beyond 4000 samples (relative to the difference in the k + 1th singular values).

The hypothesis test does appear to remove edges more aggressively in the low-data regime — i.e. the blue curves overlap with much of the orange curve in the low-data regime. However, it is worth noting that it is almost impossible to choose a threshold for the $k+1$th singular value ahead of time, whereas hypothesis tests give a meaningful interpretation to significance.

Test 2: Recovering a "Y" Graph

The second test is a simple recovery test on a 7 vertex graph in which two disconnected parents form a child, with a series of descendants. We sample 10,000 data points and test with a p-value of .0005 — i.e. we remove edges when we get a p-value of .9995 in our rank test (a value that is motivated by knock-on effects and the concentration of the "split" graph results in Test 1). We report our results from 100 different tests (and implicitly show the tested graph) in Figure 5.5.

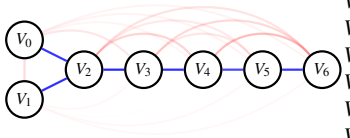
This test illustrates a few things that are not revealed in our theoretical results. The first is that edges which are far-enough apart often appear independent even before conditioning on their separating sets, presumably due to their weak dependence. In these cases our algorithm will "incorrectly" remove edges between vertices too early, but still give the correct result (as with any other independence-based algorithm).

As is the case with most causal discovery algorithms, the frequency of false-positive edges tends to increase with the size of the separating set between the vertices. Vertices with a large separating set require a rank tests for each assignment to that separating set, leading to more "accidentally" dependence. These "knock-on effects" are often handled using p-value adjustments, suggesting that a smaller p-value thresholds would serve a similar purpose for our algorithm.

It is worth emphasizing this effect is dependent on the separating set for the IPA rather than the two vertices themselves. For the graph tested in Figure 5.5 conditioning on V_5 is sufficient to induce independence between V_4 and V_6 in the unconfounded setting. In the presence of a global counfounder, however, we require an additional vertex to be coarsened with V_6 and independent from V_4. As V_6 has no descendants, we must obtain this vertex from $V_0, \ldots, V_2$, requiring an additional vertex to be conditioned on. For this reason, false-positive edges are especially likely to occur at the *end* of our chain.

Test 3: Many Graphs with Varying Density

In our third test, we explore the role that graph density plays in accurately detecting graph adjacency. For this test, we sample random Erdös-Renyi undirected graph structures on 7 vertices and orient them according to a random permutation of the edges. We vary the probability of edge-occurrence in our graphs from .1 to .9 in .1 increments, sampling 20 graph structures for each. Among these graphs we draw 10,000 datapoints and study the role of maximum in-degree and total number of edges on the percentage of correctly recovered edges (p-value .0005 again). The



	V_0	V_1	V_2	V_3	V_4	V_5	V_6
V_0	0.0	0.14	0.79	0.1	0.04	0.05	0.05
V_1	0.14	0.0	0.81	0.03	0.01	0.03	0.03
V_2	0.79	0.81	0.0	0.85	0.17	0.09	0.11
V_3	0.1	0.03	0.85	0.0	0.82	0.1	0.15
V_4	0.04	0.01	0.17	0.82	0.0	0.72	0.26
V_5	0.05	0.03	0.09	0.1	0.72	0.0	0.79
V_6	0.05	0.03	0.11	0.15	0.26	0.79	0.0

Figure 5.5: Results from Test 2. In blue, we show correctly returned edges. In red, we show edges that were returned which are **not** in the true model. The opacity of the lines show the percentage of the time that the edge was returned (ideally, we would want faint red lines and strong blue lines). To the right of the graph, we show a table of the frequency of returning the edge colored according to the same scheme.

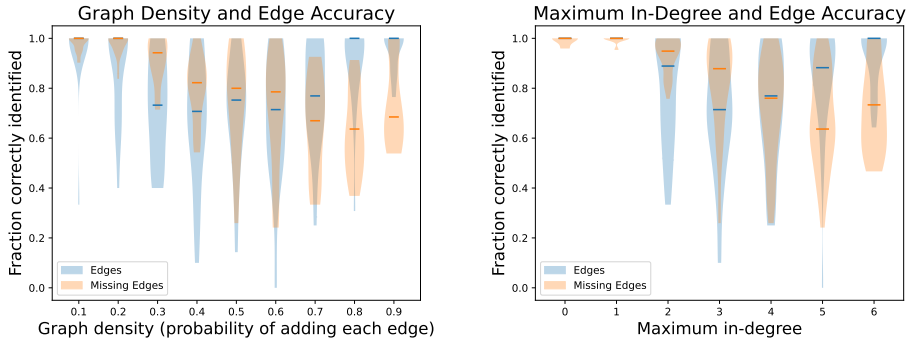


Figure 5.6: The results of Test 3. Horizontal ticks represent the median accuracy for recovering edges (blue) or lack of edges (orange). A violin plot is also shown, representing the density of results over 20 iterations at each p (probability of adding an edge).

results are given in Figure 5.6.

For $p = 0.1$ to $p = 0.3$ the medians of both true positive and true negative edge reconstruction are at 100%, with the distributions showing the occasional error. As the density of the graph increases, we fail to detect edges (lower blue marks), and incorrectly return edges where there are none (lower orange marks). At high densities, our accuracy for detecting true edges returns to higher levels, but at the cost of occasionally adding false positive edges. The second figure shows that these false positive edges may be due to the larger in-degree of more dense networks. We see very good recovery for networks with limited in-degree, and significantly more error with larger in-degrees.

5.8 Deferred Proofs

Proof of Lemma 14

To prove this lemma we first define the entire set of descendants $\mathbf{D}_{ij} \supseteq \mathbf{FB}(V_i, V_j)$.

Definition 23. Let the set $\mathbf{D}_{ij} := \mathbf{DE}(V_i, V_j)$ be the descendants of both vertices and $\mathbf{A}_{ij} := \mathbf{V} \setminus \mathbf{D}_{ij}$.

Recall that a separating set for any two sets of variables exists as a separating set in the moral graph of their ancestors (Lemma 8). By restricting our focus to $\mathbf{S}_i, \mathbf{S}_j \subseteq \mathbf{A}_{ij}$ will also have $\mathbf{AN}^+(\mathbf{S}_i^+, \mathbf{S}_j^+) \subseteq \mathbf{A}_{ij}$ which guarantees that our separating sets will not overlap with $\mathbf{FB}(V_i, V_j)$. Lemma 23 will tell us how large we need $\mathbf{A}_{ij}$ to be in order to be guaranteed an IPA.

Lemma 23. *An IPA* $\mathbf{S}_i^+, \mathbf{S}_j^+$ *for* $V_i, V_j \in \mathbf{V}$ *exists so long as* $\left|\mathbf{A}_{ij}\right| \geq (2+\Delta^2)(\lceil \lg(k) \rceil + 1)$.

Proof. We can form $\mathbf{S}_i^+$ out of V_i and $\lceil \lg(k) \rceil$ arbitrary other vertices from $\mathbf{A}_{ij}$. Now, let the separating set be $\mathbf{C} := \mathbf{MB}^{\mathfrak{G}[\mathbf{A}ij]}(\mathbf{S}_i^+)$ and note that $\mathbf{C}$ d-separates $\mathbf{S}_i^+$ from all other elements of $\mathbf{A}_{ij}$. Since we know $|C| \leq \Delta^2(\lceil \lg(k) \rceil + 1)$, we have at least $\lceil \lg(k) \rceil$ vertices in $\mathbf{A}_{ij}$ left to join with V_j and make $\mathbf{S}_j^+$. □

We are now ready to prove Lemma 14.

Proof. A convenient consequence of Lemma 23 is that it guarantees the existance of IPAs everywhere except within a small subset of vertices. Let $\overline{\mathbf{DE}}(V) := \mathbf{V} \setminus \mathbf{DE}(V)$ be the "non-descendants" of V. Note that $\mathbf{A}_{ij} = \overline{\mathbf{DE}}(V_i) \cup \overline{\mathbf{DE}}(V_j)$. This implies that

$$\left|\mathbf{A}_{ij}\right| \geq \max(\left|\overline{\mathbf{DE}}(V_i)\right|, \left|\overline{\mathbf{DE}}(V_j)\right|). \tag{5.14}$$

Hence, so long as at least one vertex has enough non-descendants, $\mathbf{A}_{ij}$ will be large enough to form an IPA. This set of vertices with enough non-descendants corresponds to the complement of the early vertices. □

Proof of Lemma 10

Proof. We will drop the conditioning on $\mathbf{c}$ in this proof for simplicity. Consider the sum

$$\sigma_j := \sum_{i=1}^{j} \Pr(u_i)\mathbf{M}[X, Y \mid u_i], \tag{5.15}$$

and note that $\sigma_k = \mathbf{M}[X, Y]$. Faithfulness with respect to $\mathfrak{G}'$ tells us that there is some assignment, which we call u_1 wlog, such that $X \not\perp\!\!\!\perp Y \mid u_1$. Hence $\mathrm{rk}_+(\sigma_1) > 1$.

Now, we show inductively that $\mathrm{rk}_+(\sigma_i) = \mathrm{rk}_+(\sigma_{i-1}) + 1$ is a measure 1 event for $i = 1, \ldots, k$. Denote $\mathbf{M}[X, Y \mid u_i] = v_i w_i^\top$ with column space v_i drawn from a

subspace with non-zero measure on $\mathbb{R}^n$. The column space of σ_{i-1} is rank $\leq i-1 < m$, so it has measure zero on $\mathbb{R}^m$. Hence, v_i being in the column space of σ_{i-1} is a measure 0 event. We conclude that $\text{rk}_+(\sigma_{i-1} + \mathbf{M}^{u_i}[X, Y]) = \text{rk}_+(\sigma_{i-1}) + 1$ with measure 1. Inducting on i gives $\text{rk}_+(\sigma_k) > k$ with measure 1. □

Proof of Lemma 13

Proof. Suppose for contradiction that some vertex $B \in \mathbf{FB}(V_i, V_j) \cap \mathbf{S}_j^+$. $B \in \mathbf{FB}(V_i, V_j)$ implies that there is a directed path $\mathbf{P} \subseteq \mathbf{FB}(V_i, V_j)$ from V_i to B. By the definition of an IPA, $B \in \mathbf{S}_j^+$ means that there must be some $\mathbf{C}$ with $B \perp\!\!\!\perp_d V_i \mid \mathbf{C}$. We conclude that $\mathbf{C}$ must contain some $C \in \mathbf{P}$ in order to block $\mathbf{P}$ from being an active path. However, this also means that $C \in \mathbf{FB}(V_i, V_j)$, which contradicts Observation 3. The same argument holds for $B \in \mathbf{S}_i^+$. □

Proof of Lemma 16

Proof. $\mathbf{T}_{ij}$ contains both $\mathbf{PA}(V_i)$ and $\mathbf{PA}(V_j)$, so Lemma 6 tells us that we contain a separating set. □

Part III

Level 4: Wisdom

Chapter 6

GRAPHICALLY MODELED CONTEXTS

Recall that the goal of Level 4 knowledge is to infer universal relationships that are invariant among possible worlds. When seeking to generalize models outside of their domain, it is natural to focus on shifts in data due to interventions and changes in exogenous variables. It is not immediately obvious what role dataset-specific biases play in universality. The focus of Part III will be to argue that sampling bias and restricted contexts play an essential role in Level 4 knowledge in the following ways:

1. Context must be understood when adjusting for shifts in distributions, or else conclusions develop paradoxes.

2. Context can be used to isolate information based on its causal functions.

6.1 The Domain Expertise Paradox

We will begin our discussion of Level 4 knowledge by introducing a paradox that arises when the context of data is misunderstood. The context we will specifically explore is that of domain expertise, in which experts or ML models are capable of classifying *only a subset* of the total possible classes. We will also refer to this as an "omitted label context." For example, "dogs vs. cats" is omitted label context, but "dogs vs. non-dogs" is not. While the relative probabilities of classes within this subset are maintained, data from all other labels are unobserved. More precisely, $p(y^*_1)/p(y^*_2) = q(y^*_1)/q(y^*_2)$ for $y*_1, y*_2 \in \mathcal{Y}^*$, but $p(y') = 0$ if $y' \notin \mathcal{Y}^*$. Within the scope of this paper, we will restrict our focus to $|\mathcal{Y}^*| = 2$.

Omitted label contexts are motivated by a few real-life scenarios within medicine and epidemiology. The first is "immortal time bias" (Suissa, 2008), which famously reversed the perceived risks of postmenopausal hormone treatment. While initial observational studies suggested this treatment could decrease in cardiovascular issues (Grodstein, Joann E Manson, and Stampfer, 2006), a followup clinical study eventually showed the opposite (Michels and JoAnn E Manson, 2003).[1] This discrepancy can be attributed to the observational study's focus on *current users*

[1] Grodstein, Joann E Manson, and Stampfer (2006) was initially published before Michels and JoAnn E Manson (2003), but later updated.

of the therapy (Hernán, Alonso, et al., 2008). More specifically, the backtracking nature of the observational study excluded a group of vulnerable women who had not survived treatment long enough to participate – i.e. all participants were "immortal" from the inception of their treatment to the beginning of the study. Exclusion of an outcome (in this case, death before the study) constitutes an example of an omitted label context.

Omitted label contexts are also extremely common in the study of rare conditions. For example, a census genome sequencing of the US population would be an impractical and financially infeasible task. Instead, databases like TCGA (Tomczak, Czerwińska, and Wiznerowicz, 2015) allow focused access to patients with specific (and often rare) cancers. In study designs, investigators may opt for an omitted label context or induce further label shift by working with a uniform distribution on the labels of interest.

Omitted labels are a form of sampling bias – a topic that has been studied in detail within the causal inference literature (Bareinboim and Tian, 2015; Correa, Tian, and Bareinboim, 2019). Bareinboim and Tian (2015) calls a causal effect "recoverable" if it can be computed in the presence of a selection mechanism. An important difficulty within omitted label contexts is that they are what we will call "irreversible." Zero-probability labels cannot be "weighted-up" to transform the distribution to the that of the general population. With respect to covariate adjustments, this leads to incompatible quantities that make the causal effects unrecoverable.

6.2 Simpson's Paradox

We will ease into our dissection of errors in causal quantities by discussing Simpson's paradox. This discussion will rely on hypothetical observational data on a treatment T and it's outcome Y, given in Table 6.1 (a). The example is motivated by the effect of illness severity on the probability of treatment prescription. Patients improve with treatment within both severe and mild cases, but treatment is primarily given to more severe illnesses that have a lower overall rate of improvement. As a result, treated patients have lower rates of improvement than untreated patients.

The driver for Simpson's paradox is the difference in severity between those who did and did not receive treatment, and the effect of this severity on patient outcomes. That is to say the treatment and control groups are not exchangeable. A natural solution to this is to reweight the rows of our table to exchangeability by emphasizing the severely ill patients who did not receive treatment and the mildly ill patients who did receive

T	X	$y^{(0)}$	$y^{(1)}$
$t^{(0)}$	$x^{(0)}$	3	7
$t^{(0)}$	$x^{(1)}$	1	0
$t^{(1)}$	$x^{(0)}$	0	1
$t^{(1)}$	$x^{(1)}$	7	3

(a) A specification of counts for Simpson's Paradox.

T	X	$y^{(0)}$	$y^{(1)}$	$y^{(2)}$
$t^{(0)}$	$x^{(0)}$	3	7	0
$t^{(0)}$	$x^{(1)}$	1	0	99
$t^{(1)}$	$x^{(0)}$	0	1	99
$t^{(1)}$	$x^{(1)}$	7	3	0

(b) An augmentation of (a) with a third column that shifts the distribution of X.

T	X	$y^{(0)}$	$y^{(1)}$	$y^{(2)}$
$t^{(1)}$	$x^{(0)}$	2	1	0
$t^{(1)}$	$x^{(1)}$	0	2	1
$t^{(1)}$	$x^{(2)}$	1	0	2
$t^{(0)}$	$x^{(0)}$	0	1	2
$t^{(0)}$	$x^{(1)}$	2	0	1
$t^{(0)}$	$x^{(2)}$	1	2	0

(c) A specification of counts that mimics Condorcet's paradox.

Table 6.1: Three tables discussed in this paper.

treatment. This is accomplished by reweighting datapoints (t, x, y) according to the inverse propensity of receiving the treatment that they got, $w(t, x, y) = 1/\Pr(t \mid x)$, sometimes referred to as "Inverse Propensity weighting" (IPW) (G. W. Imbens and Rubin, 2015). For Table 6.1 (a), this corresponds to weighting up the second and third rows by a factor of 10. When this reweighting is interpreted as a synthetic study on 40 participants (20 treated and 20 control, each with a 10 : 10 split on severity), the new apparent treatment effect is $13/20 - 7/20 = 30\%$.

An alternative perspective is that the causal effect of the treatment lies in the outcome changes *within each severity group*. By separately considering the severe and mild patients, we can average outcomes according to the marginal probability distribution of severity. Following this intuition, the "backdoor adjustment" (Judea Pearl, 2009) calculates the probability distribution of $Y = y^{(i)}$ under an intervention of $T = t^{(j)}$:

$$\Pr(y^{(i)} \mid \mathrm{do}(t^{(j)})) \coloneqq \sum_x \Pr(x) \Pr(y^{(i)} \mid x, t^{(j)}).$$

The difference between the two possible interventions gives the "average treatment effect" (ATE)

$$\mathrm{ATE} = \Pr(y^{(1)} \mid \mathrm{do}(t^{(1)})) - \Pr(y^{(1)} \mid \mathrm{do}(t^{(0)})) = \frac{1/1 + 3/10}{2} - \frac{7/10 + 0/1}{2} = .3.$$

Notice that the marginal probability distribution of X is uniform, corresponding to an equal weighting of the $x^{(0)}$ and $x^{(1)}$ rows in Table 6.1 (a). In fact, both IPW and backdoor approaches result in the same weightings of the rows of the table because $\Pr(t, x)/\Pr(t \mid x) = \Pr(x)$.

Simpson's paradox has been the subject of a long list of works for which it would be impossible to do a full justice to. Judea Pearl (2022) and Hernán, Clayton, and

Keiding (2011) describe Simpson's paradox as "solved" by causal modeling because the confounding role of X tells the researcher how to proceed, namely that they must separately consider outcome changes for each assignment of x. We will focus on one key takeaway: the choice of how to re-weight sub-cases (rows of our table) plays a key role in the conclusion of a study, sometimes reversing the apparent relationship (as in Simpson's Paradox).

An important observation that there is a geometry to the way in which these errors occur. Notice that the reversal in Table 6.1 (a) would be maximized by further increasing the probability of rows 1 and 4, e.g. by changing the 3, 7 counts to 300, 700. This reweighting strengthens the dependence of T on X, resulting in an unadjusted treatment effect that approaches $\Pr(y^{(1)} \mid t^{(1)}, x^{(1)}) - \Pr(y^{(1)} \mid t^{(0)}, x^{(0)}) = .4$. While we will not dive further into the geometry of Simpson's paradox, the existence of this structure stands as motivation for the structures we will study in Chapter 7.

6.3 Omitted Label Contexts

Now that we understand the potential effects of reweighting distributions on covariates, we will move our focus to the study of omitted label contexts. Recall that these contexts involves the removal of some labels while preserving the relative probabilities of the non-removed labels. This removal can shift the apparent distribution of any variable that is associated with Y, including both treatment T and covariates X.

Causality within Omitted Label Contexts

Consider a second hypothetical dataset that augments Table 6.1 (a) with an additional column, shown in Table 6.1 (b). We will focus on the context that excludes the deceased ($y^{(2)}$) label, meaning that the observed dataset is equivalent to Table 6.1 (a), which we recall has a .3 ATE on the outcome of $y^{(1)}$. Although the full context has the exact same (uniform) marginal probability distribution on X, we see a reversal of the ATE on $y^{(1)}$:

$$\text{ATE} = \frac{1/100 + 3/10}{2} - \frac{7/10 + 0/100}{2} = -.195. \tag{6.1}$$

The correct adjustment comes down to a loss of datapoints. The goal is to shift to exchangeable treatment and control distributions in the *overall population*, which involves weights $w(t^{(0)}, x^{(0)}, y) = w(t^{(1)}, x^{(1)}, y) = 10$ and $w(t^{(0)}, x^{(1)}, y) = w(t^{(1)}, x^{(0)}, y) = 1$, or any other rescaling. Notice that this reweighting differs from the reweighting suggested by IPW and the backdoor adjustment in the left table. Instead of scaling up the $(t^{(1)}, x^{(0)})$ and $(t^{(0)}, x^{(1)})$ rows to make up for a bias towards

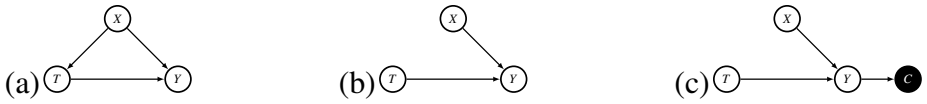


Figure 6.1: (a) A causal DAG depicting confounding from a common cause X. (b) The causal DAG that "severs" $X \to T$ by reweighting for exchangeability. (c) The causal DAG depicting the effect of a omitted label context C which has been conditioned on.

giving treatment to more severe cases, the correct reweighting does the opposite, resulting in a seemingly less exchangeable distribution.

This effect can be understood by graphically modeling the selection bias as in Bareinboim and Tian (2015), shown in Figure 6.1. Figure 6.1 (a) shows the graph describing a confounding variable X causing both T and Y. The goal of IPW and the backdoor adjustment is to reweigh the distribution to fit the DAG in Figure 6.1 (b), i.e. the distribution of X is exchangeable in both $t^{(0)}$ and $t^{(1)}$ or equivalently $X \perp\!\!\!\perp T$. Figure 6.1 (c) shows the effect of restricting the labels of Y within a dataset context (such as with omitted label contexts), which involves conditioning on a child of Y. X and T are not d-separated[2], because conditioning on a variable that is causally downstream of both X and T can induce a spurious correlation.

The effect we have outlined darkens the outlook for causal inference in omitted label contexts: We are no longer guided by the principle of exchangeability in our observed data and finding the correct adjustment requires knowledge of the context-induced distribution $\Pr(X, T, Y \mid y \in \{y^{(i)}, y^{(i)}\})$. This distribution cannot be obtained without extending the study to all labels.

To progress to Level 4 knowledge, now investigate what can be learned from many conclusions under *different* omitted label contexts. Such a network is not a replacement for a single study on all of the potential labels, but is a realistic setting for patching omitted label bias. We study networks of conclusions that ignore the limitations of their omitted labels and perform standard adjustments. We will see that these networks have limitations, much like the limitations to Simpson's paradox.

6.4 Networks of Contexts

Before we discuss the structures within networks of omitted label contexts, we will introduce another paradox from social choice theory, known as the Condorcet Paradox (Nicolas et al., 1785). We will see that this paradox and its structure are

[2] see Judea Pearl (2009) for the full rules of d-separation

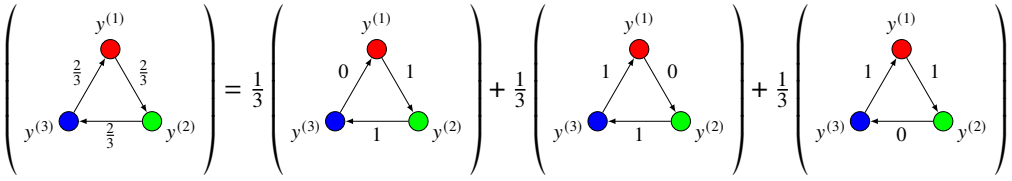


Figure 6.2: The Condorcet paradox as an aggregation of rankings.

deeply related to the networks we will study.

The Condorcet Paradox

The Condorcet paradox works as follows: three voters each have preferences $y^{(0)} \to y^{(1)} \to y^{(2)}$, $y^{(1)} \to y^{(2)} \to y^{(0)}$, and $y^{(2)} \to y^{(0)} \to y^{(1)}$, with $a \to b$ indicating a preference of a over b. The key to these preferences is that the order has been rotated three times, meaning that each candidate is preferred to its successor mod 2. That is, $y^{(i)} \to y^{(i+1 \mod 2)}$) in two out of the three voters. The result is an aggregate cycle of preference $y^{(0)} \to y^{(1)} \to y^{(2)} \to y^{(0)}$ with frequencies of $2/3$ voters for each edge.

This paradox can be generalized into what we will call an "aggregation of rankings" (AR) – a complete directed-graph[3] on the set of labels $\mathcal{Y}$ with weights on each $y^{(i)} \to y^{(j)}$ corresponding to the fraction of voters who prefer $y^{(i)}$ to $y^{(j)}$. AR structures are a convex combination of total orderings (i.e. graphs with edge weights of 0 or 1), with component weights corresponding to the fraction of voters carrying each total ordering. See Figure 6.2 for an illustration of this perspective for the Condorcet paradox. As a result, the space occupied by all possible AR structures is known as the "linear ordering polytope," which has been the subject of extensive study (P. C. Fishburn, 1992; Alon, 2002).

The preferences of voters in the Condorcet paradox can be embedded into a table of frequencies, with each voter becoming a specific value for covariate X. Table 6.1 (c) demonstrates this using the counts $2 > 1 > 0$ to induce high, medium, and low preference. Notice that the order of preferences for each x in the $t^{(1)}$ half of the table (first three rows) exactly correspond to the order of preferences given by the voters in the Condorcet paradox, starting with $y^{(0)} \to y^{(1)} \to y^{(2)}$ for $x^{(0)}$ and cycling the order with the other values of X.

[3]These graphs are always complete, but we use graph terminology as in B. Mazaheri, Jain, and Bruck (2021) in order to reference properties that are dependent on cycles.

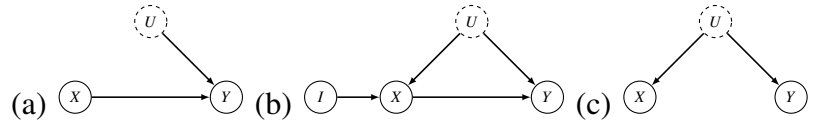


Figure 6.3: (a) U exerts unobserved influence on Y, but not on the covariate X, meaning an auxiliary training task predicting Y using X can remove the effect of U. (b) shows a DAG of an instrumental variable setup. (c) shows an unobserved active path for which an auxiliary training task predicting Y using X can capture and isolate information about U.

The $t^{(0)}$ half of the table complements the $t^{(1)}$ half so that the counts for $(t^{(0)}, x^{(i)}, y^{(j)})$ and $(t^{(1)}, x^{(i)}, y^{(j)})$ always sum to three. As a result, restricting our table to any two columns still yields a uniform probability distribution on X, i.e. $\Pr(x^{(0)} \mid y \in \{y^{(i)}, y^{(j)}\}) = \Pr(x^{(1)} \mid y \in \{y^{(i)}, y^{(j)}\}) = \Pr(x^{(2)} \mid y \in \{y^{(i)}, y^{(j)}\})$. This is the distribution that a naive study would average over when applying a backdoor adjustment, meaning that

$$\begin{aligned} \Pr(y^{(0)} \mid \mathrm{do}(t^{(1)}), y \in \{y^{(0)}, y^{(1)}\} &= \frac{2/3 + 0/2 + 1/1}{3} = 5/9, \\ \Pr(y^{(0)} \mid \mathrm{do}(t^{(0)}), y \in \{y^{(0)}, y^{(1)}\} &= \frac{0/3 + 2/2 + 1/3}{3} = 4/9. \end{aligned} \tag{6.2}$$

The calculations in 6.2 conclude that the ATE on $y^{(0)}$ in the $y \in \{y^{(0)}, y^{1)}\}$ context is $+1/9$. These calculations are the same for the ATE on $y^{(1)}$ for $y \in \{y^{(1)}, y^{(2)}\}$ and the ATE on $y^{(2)}$ for $y \in \{y^{(2)}, y^{(0)}\}$ due to the cyclic shifting of columns. Hence, the studies separately conclude that the treatment increases the relative frequency of all three labels, which is clearly impossible.

The embedding of the Condorcet paradox into causal conclusions implies a correspondence between aggregations of rankings and backdoor adjustments[4]. This correspondence will lead to the study of "expert graphs" in Chapter 7.

6.5 Context-Based Features

The paradoxes we have presented establish that exchangeability (as it is understood for causal inference) is an insufficient criterion for understanding data contexts – especially sampling bias and omitted label contexts. Chapter 8 will discuss context-based (counterfactual) features, which represent specifically engineered scenarios in which prediction tasks contain desirable information.

Training Tasks as Filters

To filter information, we will draw on and expand upon insight from instrumental variables. Consider Figure 6.3 (a) and notice that the scope of U is limited to just Y and therefore opens up no new causal pathways between variables in the observed system. As a result, we can interpret U as an additional source of independent noise for Y. Such unobserved influence cannot ever be incorporated in a model for Y with the data at hand. This distinction is related to **epistemic uncertainty** (i.e. "knowable" randomness) versus and aleatoric ("unknowable" randomness) (Hacking, 2001).

Variation due to independent noise (whether or not the uncertainty is epistemic or aleatoric) can be removed using prediction tasks. For example, in Figure 6.3 (a) if I were to train a model $\hat{Y}(\cdot)$ which predicts Y using X, the output of such a model $\hat{Y}(X)$ would be independent of U by virtue of it being a function of X, which is also independent of U. This can also be thought of as retrieving a response function that is indexed and marginalized over the unobserved noise,

$$\hat{Y}(X) \approx \mathbb{E}_U(\Gamma_u(X)). \tag{6.3}$$

Such an observation is useful even when the scope of U extends beyond Y. For example, consider Figure 6.3 (b), which graphically models the conditions of an instrumental variable. U confounds both X and Y, but $\hat{X}(I)$ is independent of U by virtue of I's independence from U. Therefore, $\hat{X}(I)$ can be used to isolate X's impact on Y from its correlation via the U backdoor path.

The effectiveness of this strategy is, of course, dependent on the unaffected variable's (i.e. I's) ability to accurately predict the cause we wish to study (i.e. X). This can be quantified using information theory. For example, in Figure 6.3 (b), $I \perp\!\!\!\perp Y \mid X, U$, which lets us apply the data processing inequality:

$$\mathcal{I}(\hat{X}(I) : Y) \leq \mathcal{I}(\hat{X}(I) : X, U) = \mathcal{I}(\hat{X}(I) : X) \leq \mathcal{I}(I : X). \tag{6.4}$$

Hence, we cannot determine causal relationships that exceed the predictive power of our instrumental variable (I) on our cause (X).

Controlling Filtration with Active Paths

The key to further engineering these training tasks involves studying the active paths between the label and its covariates. This is formalized by Lemma 24.

[4] or any other case-based weighting

Lemma 24 (Unobserved common-cause information)**.** *Given a causal DAG* $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$*, for any* $U, V_i, V_j \in \mathbf{V}$ *where* U *d-separates* V_i, V_j *(i.e.* $V_i \perp\!\!\!\perp V_j \mid U$ *by the causal Markov condition), we have that* $\mathcal{J}(V_i, V_j : U) \geq \mathcal{J}(V_i : V_j)$.

Proof. A visualization of this proof is given in Figure 6.4. Colors are added to the equations in the proof to match this figure. Begin with the definition of mutual information:

$$\mathcal{J}(V_i, V_j : U) \coloneqq \mathcal{H}(V_i, V_j) - \mathcal{H}(V_i, V_j \mid U). \tag{6.5}$$

We can expand the joint entropy of both terms as follows,

$$\mathcal{H}(V_i, V_j) = \mathcal{H}(V_i \mid V_j) + \mathcal{H}(V_j \mid V_i) + \mathcal{J}(V_i : V_j) \tag{6.6}$$

$$\mathcal{H}(V_i, V_j \mid U) = \mathcal{H}(V_i \mid V_j, U) + \mathcal{H}(V_j \mid V_i, U) + \underbrace{\mathcal{J}(V_i : V_j \mid U)}_{=0 \text{ because } V_i \perp\!\!\!\perp_d^{\mathcal{G}} V_j \mid U}. \tag{6.7}$$

Together, Equations 6.6 and 6.7 give:

$$\begin{aligned}
\mathcal{J}(V_i, V_j : U) =& \mathcal{H}(V_i \mid V_j, U) + \mathcal{H}(V_j \mid V_i) + \mathcal{J}(V_i : V_j) \\
& - \mathcal{H}(V_i \mid U, V_j) - \mathcal{H}(V_j \mid U, V_i) \\
=& \mathcal{J}(V_i : U \mid V_j) + \mathcal{J}(V_j : U \mid V_i) + \mathcal{J}(V_i : V_j) \\
\geq& \mathcal{J}(V_i : V_j).
\end{aligned}$$

□

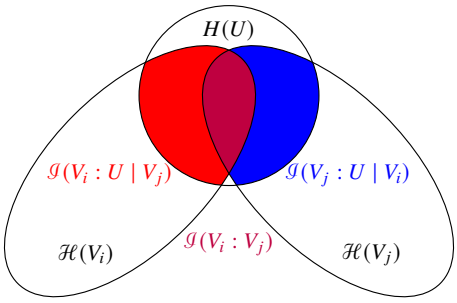


Figure 6.4: A visual proof for Lemma 24.

Now, consider the DAG in Figure 6.3 (c). We can conclude from Lemma 24 that

$$\mathcal{J}(X, Y : U) \geq \mathcal{J}(X : Y). \tag{6.8}$$

Similarly, if we apply the lemma to $\hat{Y}(X)$,

$$\mathcal{J}(\hat{Y}(X), Y : U) \geq \mathcal{J}(\hat{Y}(X) : Y). \tag{6.9}$$

Later in Chapter 8 we will argue that $\mathcal{I}(\hat{Y}(X) : Y) \approx \mathcal{I}(X : Y)$, if the model is trained "ideally." Generally, this indicates that a variable Y and its prediction from X captures information about the variables along active paths from X to Y.

With this insight comes the potential to isolate information by activating and deactivating different paths through conditioning. The predictors within these contexts become estimates of counterfactuals. Chapter 8 uses this idea to isolate robustness-improving information. Path-specific information has also been used in fairness contexts (Chiappa, 2019; Dutta, Venkatesh, et al., 2020; Dutta and Hamman, 2023), but with restrictive sufficiency constraints. A broader interpretation of training tasks as information filters may therefore also provide advances in these areas.

Chapter 7

EXPERT GRAPHS

In this section we give a detailed study of the paradox presented in Chapter 6. This work has been partially published in B. Mazaheri, Jain, and Bruck, 2021, with unpublished components in B. Mazaheri, Jain, Cook, et al., 2023.

7.1 Motivation

AI has made considerable progress towards methods for training machine learning models, but privacy laws and data ownership severely limit many consumers' access to the data necessary for these techniques. In the absence of high quality data, many practitioners rely on pre-trained third-party classifiers and regressors. In order to fully harness these "off the shelf" products, knowledge from different training tasks is put together to address new goals. This process is generally known as **decision fusion** (Castanedo, 2013).

As we have seen in Chapter 6, intuition often fails in this setting. Given an A vs. B classifier that prefers A and a B vs. C classifier that prefers B, one might assume that A is preferable to C. This assumption of *transitivity* is incorrect for any set of classifiers with decision boundaries that do not meet at a single point (see Figure 7.1). In more than 2 dimensions it is even worse; the three $(n-1)$-dimensional class boundary manifolds need to be aligned on an entire $(n-2)$-dimensional manifold. Barring perfect high-dimensional classifiers, nontransitivity is bound to occur.

Nontransitivity has been explored in two related settings: probability theory and voter preferences. In probability, sets of dice with nontransitive winning probabilities (such as A beats B beats C beats A) have been a source of considerable interest (Savage Jr, 1994). Voting theory has studied the *Condorcet paradox*, where pairwise elections of candidates yield nontransitive preferences. The "linear ordering polytope" generalization corresponds to all possible pairwise election networks in a population of ranked preferences (Alon, 2002; P. C. Fishburn, 1992; McGarvey, 1953; Cohen and Falmagne, 1990; Saari, 2000), which we call *Aggegations of Rankings* (ARs). Related work focuses on the "Condorcet domain," which studies conditions necessary for transitivity (Saari, 2009; P. Fishburn, 1996; Monjardet, 2006). Such previous work has only focused on complete graphs of all pairwise elections.

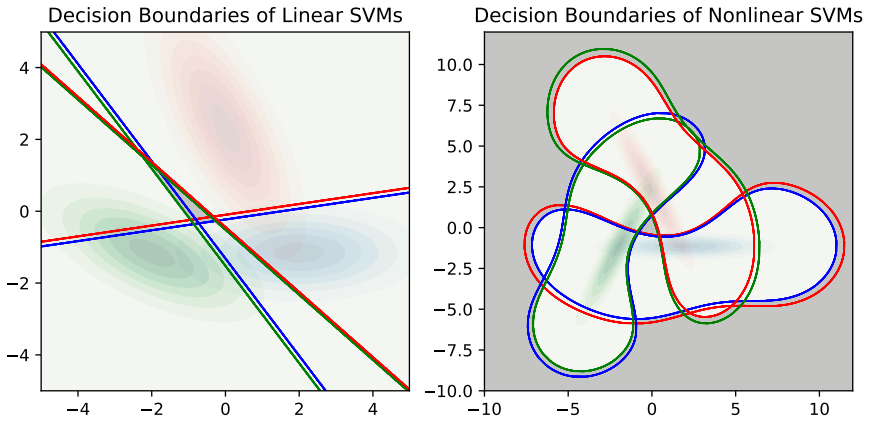


Figure 7.1: Examples of decision boundaries from classifiers trained on pairs of differently colored Gaussians. Regions of nontransitivity are shaded in grey. In the case on the right, this includes both the center and the outer region of all outliers. The classifiers on the left are trained with sklearn's linear SVM, and on the right they are trained with sklearn's nonlinear kernel SVM.

The level of nontransitivity is limited in these frameworks. For example, it is possible to construct a population of voters where $2/3$ prefer A to B to C to A, but impossible to create such a cycle with 100% preference. This effect is known as the *triangle inequality* or, as we call it, the *curl condition*. Other properties of this framework have also been studied (P. C. Fishburn, 1992; Reinelt, 1985; Grötschel, Jünger, and Reinelt, 1985; Gilboa, 1990).

This chapter will present an investigation into the parallels between ARs, machine learning classifiers and, more generally, human experts, particularly with respect to the paradoxes discussed in Chapter 6. These *expert graphs* expand our understanding of nontransitivity and empower further cross-pollination between these fields, such as bounding unknown edges and protocols for deciding on a transitive ordering in a nontransitive region.

Additional Notation

The following notations are used throughout the chapter.

- $[\ell]$ is used to denote the set $\{1, 2, \ldots, \ell\}$ for any $\ell \in \mathbb{N}$.
- $\mathbb{1}[c]$ will be used for an indicator function which is 1 if condition c is met and 0 otherwise.
- $\mathbf{1}_\ell$ denotes an all 1 vector of size ℓ.

- $\triangle^\ell$ will be used to denote vectors of length ℓ which are probability distributions. That is, $\lambda \in \triangle^\ell$ iff $\lambda \in [0,1]^\ell$ and $\mathbf{1}^\top \lambda = 1$.
- We use $\prec, \succ, \preceq, \succeq$ to denote element-wise inequality. For example, we say $\mathbf{w} \succeq \mathbf{v}$ if $w_i \geq v_i \; \forall \, i \in [\ell]$.
- We will use $\mathrm{Co}(S)$ to denote the open convex hull of S, $\overline{\mathrm{Co}}(S)$ to denote the closed convex hull, and $\mathrm{Bo}(\cdot)$ to denote the boundary.
- While graphs in other chapters have vertices that are random variables, the vertices in this chapter represent assignments to Y. Hence, they appear as lowercase letters. Sets and ordered lists of labels are denoted using calligraphic font.

7.2 Aggregations of Rankings and Soft Rankings

When setting up the paradox of nontransitivity presented in Chapter 6, Table 6.1 (c) used counts of $2, 1, 0$ to induce preference between labels in each row. As this system is effectively cardinal, these preferences differ from those in the Condorcet paradox in that they can be any frequencies between $[0, 1]$. For this reason, we refer to the induced preferences in each row of our tables as a "soft ranking." We will now be formal about both aggregations of rankings (ARs) and aggregations of soft rankings (ASRs).

Definition 24 (Ranking)**.** A ranking of $\mathcal{Y}$ is a function $A : \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$ generated by a total ordering. We use $A(y^{(i)}, y^{(j)}) = 1$ to denote preference $y^{(i)} \to y^{(j)}$ and $A(y^{(i)}, y^{(j)}) = 1$ for $y^{(i)} \leftarrow y^{(j)}$.

Definition 25 (Aggregation of Rankings (AR))**.** An aggregation of rankings is specified by a set of rankings $\mathbf{A}$ and a corresponding weight function $\alpha \in \triangle^{|\mathbf{A}|}$ (indexed by $A \in \mathbf{A}$).

Definition 26 (Aggregate Preference)**.** An aggregation preference in an AR between $y^{(i)}, y^{(j)} \in \mathcal{Y}$ is defined to be

$$R_{\mathbf{A},\alpha}(y^{(i)}, y^{(j)}) \coloneqq \sum_{A \in \mathbf{A}} \alpha_A A(y^{(i)}, y^{(j)}).$$

Corresponding to rankings, ARs, and aggregate preferences $R_{\mathbf{A},\alpha}$, we will have soft rankings, ASRs, and aggregate probabilities $F_{\mathbf{B},\beta}$.

Definition 27 (Soft Rankings)**.** A soft ranking on $\mathcal{Y}$ is a function $B : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ generated by a categorical probability distribution on $\mathcal{Y}$, $\mathbf{p} \in \triangle^{|\mathcal{Y}|}$:

$$B(y^{(i)}, y^{(j)}) \coloneqq \frac{p_i}{p_i + p_j}.$$

Definition 28 (Aggregation of Soft Rankings (ASR))**.** An aggregation of soft rankings is specified by a set of soft rankings $\mathbf{B}$ and a corresponding weight function $\beta \in \triangle^{|\mathbf{B}|}$ (indexed by $B \in \mathbf{B}$).

Definition 29 (Aggregate Probability)**.** An aggregate probability in an ASR between $y^{(i)}, y^{(j)} \in \mathcal{Y}$ is

$$F_{\mathbf{B},\beta}(y^{(i)}, y^{(j)}) \coloneqq \sum_{B \in \mathbf{B}} \beta_B B(y^{(i)}, y^{(j)}).$$

Observation 5. Suppose the probability distribution for a covariate adjustment on $\mathbf{X}$ (e.g. X in our previous examples), $\Pr(\mathbf{X} \mid Y \in \{y^{(i)}, y^{(j)}\})$ is the same for all pairs of labels $\{y^{(i)}, y^{(j)}\}$. The treatment effects computed for each label pair then correspond to the difference between the aggregate probabilities in two ASRs with a B for each assignment of $\mathbf{X} = \mathbf{x}$ and $\beta_B = \Pr(\mathbf{x} \mid Y \in \{y^{(i)}, y^{(j)}\})$.

We will now show that ARs and ASRs on the same cardinality $|\mathcal{Y}| = n$ can hold the exact same vectors of weights. To make this statement precise, we will denote $\mathcal{A}$ as the set of $\{0, 1\}^{n(n-1)}$ vectors associated with the output values of some A in a total ordering and $\mathrm{Co}(\mathcal{A})$ as its convex hull. Note that $\mathrm{Co}(\mathcal{A})$ is the space of possible vectors of aggregate preferences $R(y^{(i)}, y^{(j)})$. Similarly denote $\mathcal{B}$ as the set of $[0, 1]^{n(n-1)}$ vectors generated by some categorical distribution and note its convex hull $\mathrm{Co}(\mathcal{B})$ is the space of possible aggregate probability vectors.

Theorem 4. *$Co(\mathcal{A})$ and $\overline{Co}(\mathcal{B})$ are the same.*

It is not difficult to see how soft rankings can be made "harder" by simply increasing the relative difference in counts. That is, replacing $2, 1, 0$ in Table 6.1 (c) with $100, 1, 0$ more closely simulates an absolute preference. Showing that any set of aggregate probabilities from an ASR can be realized with aggregate preferences from an AR is less obvious. We will prove this direction by using the probability table in an ASR to directly construct a corresponding AR.

Probabilities can Emulate Preferences

We will begin with the simpler direction, given by Lemma 25.

Lemma 25. $Co(\mathcal{A}) \subset \overline{Co}(\mathcal{B})$.

To prove Lemma 25, we will first show that for every $A \in \mathcal{A}$, there exists a $B \in \mathcal{B}$ which is arbitrarily close to it. We will then make use of the following more general lemma.

Lemma 26. *Consider a set of vectors $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_t\}$ and with $\mathbf{v}_i \in \mathbb{R}^m$ for all i. If we have $\tilde{\mathcal{V}}$ such that for every $\varepsilon > 0$ and $\mathbf{v} \in V$, there exists $\tilde{\mathbf{v}} \in \tilde{\mathcal{V}}$ such that $\|\tilde{\mathbf{v}} - \mathbf{v}\|_2 < \varepsilon$, then $Co(\mathcal{V}) \subseteq \overline{Co}(\tilde{\mathcal{V}})$.*

Convex hulls of finite sets in $\mathbb{R}^\ell$ are *convex* polytopes, which can be expressed as an intersection of h halfspaces indexed by f with $\{\mathbf{x} : \mathbf{a^{(f)}}^\top \mathbf{x} x < b^{(f)}\}$ (Grünbaum et al., 1967). Vectors $\mathbf{a^{(f)}}^\top$ can be combined as row-vectors of a matrix, A, so that any convex polytope can be expressed as

$$\{x : A\mathbf{x} \prec \mathbf{b}\} = \left\{ \mathbf{x} : \begin{pmatrix} (\mathbf{a^{(1)}})^\top \\ \vdots \\ (\mathbf{a^{(h)}})^\top \end{pmatrix} \mathbf{x} \prec \begin{pmatrix} b^{(1)} \\ \vdots \\ b^{(h)} \end{pmatrix} \right\}. \tag{7.1}$$

For convenience, the vectors $\mathbf{a^{(f)}}, \tilde{\mathbf{a}}^{(\mathbf{f})}$ are assumed to be unit vectors throughout.

The idea behind the proof will be to analyze the movement of the boundaries of the polytope defined by $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ and corresponding polytope defined by the "perturbed points" $\tilde{\mathcal{V}} = \{\tilde{\mathbf{v}}_1, \ldots \tilde{\mathbf{v}}_m\}$. They key is to show that a point that is far enough from the boundary of $\mathrm{Co}(\mathcal{V}$ will also be within $\mathrm{Co}(\tilde{\mathcal{V}})$, given by Lemma 27. This required distance from the boundary will be relative to the amount by which the perturbed points have moved. As we make the perturbation arbitrarily small (i.e. $\varepsilon \to 0$, all points in the interior of the polytope will be included.

Lemma 27. *Let*

$$Co(\mathcal{V}) = \{\mathbf{x} : A\mathbf{x} \prec \mathbf{b}\}$$
$$Co(\tilde{\mathcal{V}}) = \{\mathbf{x} : \tilde{A}\mathbf{x} \prec \tilde{\mathbf{b}}\}$$

as given by Equation 7.1. If $A\mathbf{x} \prec \mathbf{b} - \varepsilon \mathbf{1}_\ell$ and $\|\mathbf{v_i} - \tilde{\mathbf{v}}_\mathbf{i}\|_2 < \varepsilon \ \forall i$, then $\tilde{A}\mathbf{x} \prec \tilde{\mathbf{b}}$.

To prove Lemma 27, we will need to show that the boundaries of the polytopes do not move too much. We will do this using Lemma 28, which bounds how far $\mathrm{Bo}(\mathrm{Co}(V))$ can be from $\mathrm{Bo}(\mathrm{Co}(\tilde{\mathcal{V}}))$ along a single "face."

Definition 30. Choose $f \in [h]$. Define:

$$W^{(f)} = \{\mathbf{w} : (\mathbf{a^{(f)}})^\top w = b^{(f)}, \mathbf{w} \in V\}$$
$$\tilde{W}^{(f)} = \{\tilde{\mathbf{v}}_\mathbf{i} : \mathbf{v_i} \in W^{(f)}\}$$

We restrict the size of $\left|W^{(f)}\right| = \ell$, which is the number of points needed to define a halfspace in $\mathbb{R}^\ell$. This can be done by allowing for multiple identical $\mathbf{a_f}, b_f$ combinations corresponding to all size ℓ subsets of the v_i along the boundary.

Note that $\mathrm{Co}(W^{(f)})$ describes a "face" of the polytope $\mathrm{Co}(\mathcal{V})$ indexed by f which is perpendicular to $\mathbf{a^{(f)}}$. $\mathrm{Co}(\tilde{W}^{(f)})$ describes the perturbed face.

Lemma 28. *Choose $f, g \in [h]$ arbitrarily and let $W^{(f)} = \{\mathbf{w_1^{(f)}}, \ldots, \mathbf{w_\ell^{(f)}}\}$ and $\tilde{W}^{(f)} = \{\tilde{\mathbf{w}}_\mathbf{1}^{(\mathbf{f})}, \ldots, \tilde{\mathbf{w}}_\ell^{(\mathbf{f})}\}$. For every $\mathbf{m^{(f)}} \in \overline{Co}(W^{(f)})$, we have $(\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \mathbf{m^{(f)}} < \tilde{b}^{(g)} + \varepsilon$.*

Proof. Because $m \in \overline{\mathrm{Co}}(W^{(f)})$, there is some $\lambda \in \triangle_\ell$ with

$$\mathbf{m^{(f)}} = \sum_{i=1}^{\ell} \lambda_i \mathbf{w_i^{(f)}} \in \overline{\mathrm{Co}}(W^{(f)}) \tag{7.2}$$

Consider also

$$\tilde{\mathbf{m}}^{(\mathbf{f})} = \sum_{i=1}^{\ell} \lambda_i \tilde{\mathbf{w}}_\mathbf{i}^{(\mathbf{f})} \in \overline{\mathrm{Co}}(\tilde{W}^{(f)}) \tag{7.3}$$

Note that the norm of the difference between these two vectors is bounded:

$$\begin{aligned} \left\|\mathbf{m^{(f)}} - \tilde{\mathbf{m}}^{(\mathbf{f})}\right\|_2 &= \left\|\sum_{i=1}^{\ell} \lambda_i (\mathbf{w_i^{(f)}} - \tilde{\mathbf{w}}_\mathbf{i}^{(\mathbf{f})})\right\|_2 \\ &\leq \sum_{i=1}^{\ell} \lambda_i \underbrace{\left\|\mathbf{w_i^{(f)}} - \tilde{\mathbf{w}}_\mathbf{i}^{(\mathbf{f})}\right\|_2}_{<\varepsilon} < \varepsilon \end{aligned} \tag{7.4}$$

Also note that because $\tilde{\mathbf{m}}^{(\mathbf{f})} \in \overline{\mathrm{Co}}(\tilde{W}^{(f)}) \subseteq \overline{\mathrm{Co}}(\tilde{\mathcal{V}})$, we have that $(\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \tilde{\mathbf{m}}^{(\mathbf{f})} \leq \tilde{b}^{(g)}$. Now, a simple application of Cauchy-Schwartz gives:

$$\begin{aligned} (\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \mathbf{m^{(f)}} &= (\tilde{\mathbf{a}}^{(\mathbf{g})})^\top (\tilde{\mathbf{m}}^{(\mathbf{f})} + (\mathbf{m^{(f)}} - \tilde{\mathbf{m}}^{(\mathbf{f})})) \\ &= \underbrace{(\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \tilde{\mathbf{m}}^{(\mathbf{f})}}_{\leq \tilde{b}^{(g)}} + (\tilde{\mathbf{a}}^{(\mathbf{g})})^\top (\mathbf{m^{(f)}} - \tilde{\mathbf{m}}^{(\mathbf{f})}) \\ &\leq \tilde{b}^{(g)} + \left\|\tilde{\mathbf{a}}^{(\mathbf{g})}\right\|_2 \left\|\mathbf{m^{(f)}} - \tilde{\mathbf{m}}^{(\mathbf{f})}\right\|_2 \\ &< \tilde{b}^{(g)} + \varepsilon \end{aligned} \tag{7.5}$$

□

With this, we are now ready to prove Lemma 27.

Proof. Choose an arbitrary face $g \in [h]$. Recall we have $\mathbf{x} \in \text{Co}(V)$ with $(\mathbf{a}^{(\mathbf{g})})^\top \mathbf{x} < b - \varepsilon$ and we wish to show $(\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \mathbf{x} < \tilde{b}^{(g)}$.

Let $\mathbf{m}_{\mathbf{x}}^{(\mathbf{f})}$ be the result of extending $\tilde{\mathbf{a}}^{(\mathbf{g})}$ from $\mathbf{x}$ to $\text{Bo}(V)$. This must hit some face with $(\mathbf{a}^{(\mathbf{f})})^\top \mathbf{m}_{\mathbf{x}}^{(\mathbf{f})} = b^{(f)}$, so $\mathbf{m}_{\mathbf{x}}^{(\mathbf{f})} \in \text{Co}(W^{(f)})$. That is, find β such that

$$\mathbf{m}_{\mathbf{x}}^{(\mathbf{f})} = \beta \tilde{\mathbf{a}}^{(\mathbf{g})} + \mathbf{x} \in \text{Co}(W^{(f)}) \tag{7.6}$$

First, lets bound β. Notice that because $\mathbf{m}_{\mathbf{x}}^{(\mathbf{f})} \in \text{Co}(W^{(f)})$, we have

$$\begin{aligned} (\mathbf{a}^{(\mathbf{f})})^\top \mathbf{m}_{\mathbf{x}}^{(\mathbf{f})} &= (\mathbf{a}^{(\mathbf{f})})^\top \left(\sum_{i=1}^{\ell} \lambda_i \mathbf{w}_{\mathbf{i}}^{(\mathbf{f})} \right) \\ &= \sum_{i=1}^{\ell} \lambda_i (\mathbf{a}^{(\mathbf{f})})^\top \mathbf{w}_{\mathbf{i}}^{(\mathbf{f})} = b^{(f)} \end{aligned} \tag{7.7}$$

So, we have

$$b^{(f)} = (\mathbf{a}^{(\mathbf{f})})^\top \mathbf{m}_{\mathbf{x}}^{(\mathbf{f})} = \beta \underbrace{(\mathbf{a}^{(\mathbf{f})})^\top \tilde{\mathbf{a}}^{(\mathbf{g})}}_{\leq 1} + \underbrace{(\mathbf{a}^{(\mathbf{f})})^\top \mathbf{x}}_{< b^{(f)} - \varepsilon} \Rightarrow \varepsilon < \beta \tag{7.8}$$

Now, apply Lemma 28

$$\begin{aligned} (\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \mathbf{m}_{\mathbf{x}}^{(\mathbf{f})} &< \tilde{b}^{(g)} + \varepsilon \\ (\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \mathbf{x} + (\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \tilde{\mathbf{a}}^{(\mathbf{g})} \beta &< \tilde{b}^{(g)} + \varepsilon \\ (\tilde{\mathbf{a}}^{(\mathbf{g})})^\top \mathbf{x} &< \tilde{b}^{(g)}. \end{aligned} \tag{7.9}$$

Face $g \in [h]$ was chosen arbitrarily, so this holds for all halfspaces in the convex polytope. Hence, we have $A\mathbf{x} < \mathbf{b}$. □

We now give the proof for Lemma 25.

Proof. For a given A from a total ordering, we will show how to find a probability vector $\mathbf{p}$ that generates a B with values that are arbitrarily close to the 0, 1 values of A. As already alluded to, this will involve blowing up the ratios of the probabilities in $\mathbf{p}$.

Let $y^{(0)} \to \ldots \to y^{(n-1)}$ be the ordering specified without loss of generality. Let the ith element of $\mathbf{p}$ be ε^i / z, where $z = \sum_{j=1}^{n} \varepsilon^j$ is simply a normalization factor so that $\mathbf{p}$ remains in the simplex. Notice that this assignment gives us

$$B(y^{(i)}, y^{(j)}) = \frac{1}{1 + \varepsilon^{j-i}}$$

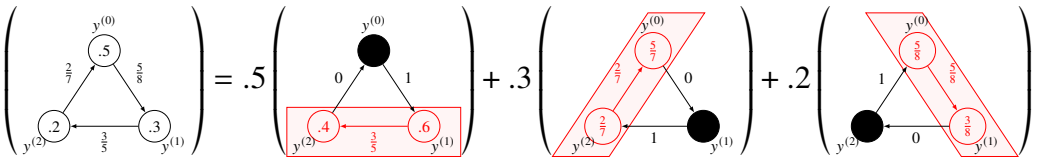


Figure 7.2: A demonstration of the inductive step in the proof for Lemma 29. The weights on the LHS are the aggregate probabilities $(y^{(i)}, y^{(j)})$ that we wish to generate, while the numbers within each vertex $y^{(i)}$ specify p_i. The weights of the graphs on the RHS are given by Equation 7.11, with adjusted (re-normalized) probabilities $p^{[-k]}$ specified within the vertices. Three subgraphs are highlighted in red, which represent the smaller sets of labels which can be decomposed according to the inductive hypothesis.

for all $j > i$. Our goal is $B(y^{(i)}, y^{(j)})$ arbitrarily close to $A(y^{(i)}, y^{(j)}) = 1$ for $j > i$. Setting $\varepsilon > 0$ arbitrarily close to 0 achieves this. Finally, we can apply Lemma 26 to complete our proof. □

Preferences can Emulate Probabilities

We will continue with the more difficult direction, given by Lemma 29.

Lemma 29. $Co(\mathcal{B}) \subset Co(\mathcal{A})$.

We will prove this direction by showing that every possible instance of B is in $\text{Co}(\mathcal{A})$. Convexity of $\text{Co}(\mathcal{A})$ will then complete the proof.

Let $\mathbf{A}^{(i)} \subset \mathbf{A}$ denote the set of rankings for which $y^{(i)}$ is the "first choice." Equivalently, $\mathbf{A}^{(i)}$ is defined such that we have $A(y^{(i)}, y^{(j)}) = 1$ for all $j \neq i$ and $A \in \mathbf{A}^{(i)}$. We extend this notation to multiple indices, with $\mathbf{A}^{(ij)}$ encoding $y^{(i)}$ as first choice and $y^{(j)}$ as second choice. If the number of rankings that satisfy the restriction is singular, then we remove the bold, e.g. $A^{(ij)}$.

Proof. We will induct on the number of labels n. The inductive hypothesis is that any $B \in \mathcal{B}$ generated by a categorical distribution $\mathbf{p} \in \triangle^n$ over n labels can be expressed as an AR $\mathbf{A}$, α. This can easily be shown for the base case of $n = 2$ by assigning $\alpha_{A^{(01)}} = p_0$ and $\alpha_{A^{(10)}} = p_1$.

Now, assuming the inductive hypothesis to be correct for all B on n labels, we will show how to construct an AR for a B on $n + 1$ labels.

First, expand $R_{\mathbf{A},\alpha}$, which we have not yet specified, into aggregate rankings on $\mathbf{A}^{(0)}, \ldots \mathbf{A}^{(n)}$,

$$R_{\mathbf{A},\alpha} = \sum_{k=0}^{n} p_k R_{\mathbf{A}^{(k)},\alpha^{(k)}}. \tag{7.10}$$

Now, consider choosing some label $y^{(k)}$ and constructing a new $\mathbf{p}^{[-k]} \in \triangle^n$ by setting $p_k^{[-k]} \leftarrow 0$ and all other $p_i^{[-k]} \leftarrow p_i/(1-p_k)$. Notice that $B(y^{(i)}, y^{(j)})$ is invariant to scaling p_i, p_j (if scaled together). Therefore, this new $\mathbf{p}^{[-k]}$ implies a $B^{[-k]}(y^{(i)}, y^{(j)})$ that matches $B(y^{(i)}, y^{(j)})$ for all $i, j \neq k$. $B^{[-k]}$ also satisfies the requirements for the inductive hypothesis, so we can assume there is a set of rankings $\mathbf{A}^{[-k]}$ and corresponding $\alpha^{[-k]}$ that forms an AR for which $R_{\mathbf{A}^{[-k]},\alpha^{[-k]}} = B^{[-k]}$.

Each $A^{[-k]} \in \mathbf{A}^{[-k]}$ can now be augmented with a first-choice preference of $y^{(k)}$ to generate the set $\mathbf{A}^{(k)}$ with corresponding $\alpha^{(k)} = \alpha^{[-k]}$. Using this assignment, we have that

$$R_{\mathbf{A}^{(k)},\alpha^{(k)}}(y^{(i)}, y^{(j)}) = \begin{cases} B(y^{(i)}, y^{(j)}) & \text{if } i, j \neq k \\ 1 & \text{if } i = k \\ 0 & \text{if } j = k \end{cases} \tag{7.11}$$

Applying Equation 7.11 to Equation 7.10 gives,

$$\begin{aligned} R_{\mathbf{A},\alpha}(y^{(i)}, y^{(j)}) &= p_i + \sum_{k \neq i,j} p_k B(y^{(i)}, y^{(j)}) \\ &= (p_i + p_j) B(y^{(i)}, y^{(j)}) + (1 - p_i - p_j) B(y^{(i)}, y^{(j)}) \\ &= B(y^{(i)}, y^{(j)}). \end{aligned}$$

We chose i, j wlog, so we have constructed an AR which emulates the the soft ranking B. This completes the inductive proof. As stated earlier, convexity of $\mathrm{Co}(\mathscr{A})$ gives the desired result. □

As the proof for Lemma 29 is rather complicated, Figure 7.2 illustrates an example inductive step.

7.3 Curl and the curl condition

Not all assignments of weights to edges are realizable ARs/ASRs. Understanding these restrictions can help us bound the outputs of untrained classifiers (a task addressed in Section 7.5), which corresponds to quantifying allowed weights for nonexistent edges. A necessary condition for ARs (and therefore also ASRs) is the "triangle inequality" (P. C. Fishburn, 1992).

Lemma 30 (Triangle Inequality)**.** *Given an AR, $R_{\mathbf{A},\alpha}$, and any three $y^{(i)}, y^{(j)}, y^{(k)}$, with $i \neq j \neq k$, we have $R_{\mathbf{A},\alpha}(y^{(i)}, y^{(k)}) \leq R_{\mathbf{A},\alpha}(y^{(i)}, y^{(j)}) + A(y^{(j)}, y^{(k)})$.*

Observation 6. We can rewrite the triangle inequality in terms of the weights along a cycle. Using $R_{\mathbf{A},\alpha}(y^{(a)}, y^{(b)}) = 1 - R_{\mathbf{A},\alpha}(y^{(b)}, y^{(a)})$ on the triangle inequality we get

$$R_{\mathbf{A},\alpha}(y^{(i)}, y^{(j)}) + R_{\mathbf{A},\alpha}(y^{(j)}, y^{(k)}) + R_{\mathbf{A},\alpha}(y^{(k)}, y^{(i)}) \geq 1, \tag{7.12}$$

$$R_{\mathbf{A},\alpha}(y^{(i)}, y^{(j)}) + R_{\mathbf{A},\alpha}(y^{(j)}, y^{(k)}) + R_{\mathbf{A},\alpha}(y^{(k)}, y^{(i)}) \leq 2. \tag{7.13}$$

From Observation 6, we see that the triangle inequality is in fact a statement about nontransitivity along 3-cycles. To generalize this notion to larger cycles, we define the curl for a function F, which may be a ranking, soft ranking, or aggregation of rankings or soft rankings.

Definition 31 (Curl)**.** Given any function F on pairs of labels $\mathcal{Y}$ and a cycle $c = (c_1, c_2, \ldots, c_\ell)$ on those labels, we define the **curl**:

$$\text{Curl}(F, c) \coloneqq F(y^{(c_\ell)}, y^{(c_1)}) + \sum_{i=1}^{(\ell-1)} F(y^{(c_i)}, y^{(c_{i+1})}). \tag{7.14}$$

This generalization of the triangle inequality to larger cycles is called the "curl condition."

Definition 32 (Curl Condition)**.** Given a ranking, soft ranking, AR, or ASR F, the *curl condition* is satisfied if, for all cycles c of length ℓ, we have $1 < \text{Curl}(F, c) < \ell - 1$. If the curl condition is satisfied, we say that F is "curl consistent."

The curl condition is obeyed for all rankings because of acyclicity. ARs are linear combinations of rankings and therefore also follow the curl condition. Finally, the equivalence between ARs and ASRs given in Theorem 4 means that the curl condition must also hold for ASRs. Therefore, the curl condition is necesarry for all of the structures we have discussed in this Chapter.

7.4 When the curl condition is sufficient

It is known that the curl condition is insufficient to describe all possible ARs (i.e. there are examples of curl-consistent F which cannot be achieved using an AR). However, the condition *is* sufficient in certain sparse settings. Such sparse settings will require a more graphical description of ARs and ASRs which we will call linear ordering graphs (LOGs) and Expert Graphs (EGs) respectively.

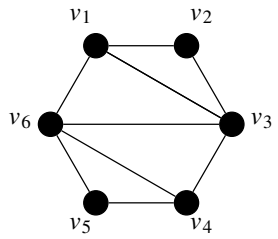


Figure 7.3: An example of a triangulated cycle.

Definition 33. A **linear ordering graph** (LOG) $G = (\mathcal{Y}, \mathcal{E}, R_{\mathbf{A},\alpha})$ is a directed weighted graph with the weight on edge $(y^{(i)}, y^{(j)}) \in \mathcal{E}$ given by $R_{\mathbf{A},\alpha}(y^{(i)}, y^{(() j)})$ from an AR.

Definition 34. An **expert graph** (EG) $G = (\mathcal{Y}, \mathcal{E}, F_{\mathbf{B},\beta})$ is a directed weighted graph with the weight on edge $(y^{(i)}, y^{(() j)}) \in \mathcal{E}$ given by $F_{\mathbf{B},\beta}$ from an ASR.

Past work has shown that the curl condition (triangle inequality) is sufficient on graphs with less than 6 vertices (Alon, 2002), a special example of which is a 3-vertex clique (a "triangle"). This result can be composed to prove sufficiency for a class of graphs we call "triangulated cycles." We conjecture that the curl condition is sufficient to describe *all* planar LOGs and expert graphs.

Triangulated Cycles

We will now explore a class of graphs corresponding to a cycle that is triangulated with chords.

Definition 35. A **triangulated cycle** is a planar graph with vertices $\mathcal{V} = \{v_1, \ldots, v_\ell\}$ and cyclic edges $\mathcal{E} = \mathcal{E}_C \cup \mathcal{E}_T$ where $\mathcal{E}_C$ represents the edges around the cycle and $\mathcal{E}_T$ represents the chords. More specifically, $\mathcal{E}_C = \{(v_1, v_2), (v_2, v_3), \ldots, (v_n, v_1)\}$ and $\mathcal{E}_T$ follows:

1. For all $v_i, v_{i+1} \in V$, there exists $v_{t(i,i+1)} \in V$ such that $(v_i, v_{t(i,i+1)}) \in \mathcal{E}_T$ and $(v_{i+1}, v_{t(i,i+1)}) \in \mathcal{E}_T$.
2. There is no $(v_i, v_j) \in \mathcal{E}$ and $(v_a, v_b) \in \mathcal{E}$, such that the edges "cross": $i < a < j < b$ or $a < i < b < j$.

We begin with a lemma that helps us restrict our focus to the smaller components of the cycle.

Lemma 31. *Consider a weighted digraph or EG/LOG $G = (V, E, F)$ and two cycles, $a = (x, y, a_1, \ldots, a_{\ell_a - 2})$ and $b = (y, x, b_1, \ldots, b_{\ell_b - 2})$ which overlap at a single edge (x, y), consider the "outer-cycle" $c = (a_1, \ldots, a_{\ell_a - 2}, b_1, \ldots, b_{\ell_b - 2})$. If a and b follow the curl condition, then so does c.*

Proof. Note that because $F(x, y) = 1 - F(y, x)$, we have

$$\text{Curl}(F, a) + \text{Curl}(F, b) - 1 = \text{Curl}(F, c)$$

If the curl condition follows for both a and b, then

$$\begin{aligned} &\text{Curl}(F, c) \geq 1 \\ &\text{Curl}(F, c) \leq \ell_a + \ell_b - 1. \end{aligned}$$

□

To prove sufficiency of the curl condition for a class of graphs, we will use Theorem 4, which equivalence between expert graphs and LOGs. That is, by showing that a class of DAGs with weights satisfying the curl condition can be decomposed into a linear combination of ranking graphs, we can conclude that it can be achieved as a LOG and therefore also be achieved as an expert graph. To apply this process, we will merge decompositions of sub-graphs, for which we will need the concept of "graph merging."

Definition 36. Consider two LOGs $G_1 = (\mathcal{Y}_1, \mathcal{E}_1, F_1)$ and $G_2 = (\mathcal{Y}_2, \mathcal{E}_2, F_2)$. If for all shared edges $e \in \mathcal{E}_1 \cap \mathcal{E}_2$ we have $F_1(e) = F_2(e)$, then we can *merge* the graphs to get $G_1 \cup G_2 = (\mathcal{Y}_1 \cup \mathcal{Y}_2, \mathcal{E}_1 \cup, \mathcal{E}_2, F)$ with

$$F(e) \coloneqq \begin{cases} F_1(e) & \text{if } e \in \mathcal{E}_1 \\ F_2(e) & \text{if } e \in \mathcal{E}_2 \end{cases}.$$

Lemma 32. *Consider two LOGs $G_1 = (\mathcal{Y}_2, \mathcal{E}_1, F_1)$ and $G_2 = (\mathcal{Y}_2, \mathcal{E}_2, F_2)$. If G_1 and G_2 share a single edge $\mathcal{E}_1 \cap \mathcal{E}_2 = e^*$ for which $F_1(e^*) = F_2(e^*)$, then $G_1 \cup G_2$ is also a LOG.*

Proof. By the definition of LOGs, G_1 and G_2 must have decompositions into weighted digraphs with binary weightings and no cycles. For G_1, sort the decompositions into rankings with $F_1(e) = 0$, which define two new LOGs and a weight $w^{(1)}$:

$$G_1 = w^{(1)} G_1^{F_1(e^*)=0} + (1 - w^{(1)}) G_1^{F_1(e^*)=1}. \tag{7.15}$$

Do the same for G_2,

$$G_2 = w^{(2)} G_2^{F_2(e^*)=0} + (1 - w^{(2)}) G_2^{F_2(e^*)=1}. \tag{7.16}$$

Note that $G_1^{F_2(e^*)=1} \cup G_2^{F_2(e^*)=1}$ is a LOG and can be merged because e^* is the only shared edge and has the same weight in both graphs. Now, note that because $F_1(e^*) = F_2(e^*)$, we have $w \coloneqq w^{(1)} = w^{(2)}$. This gives,

$$G_1 \cup G_2 = w(G_1^{F_2(e^*)=0} \cup G_2^{F_2(e^*)=0}) + (1 - w)(G_1^{F_2(e^*)=1} \cup G_2^{F_2(e^*)=1}). \qquad \square$$

We are now ready to prove that the curl condition is sufficient to describe all expert graphs on triangulated cycles.

Theorem 5. *Any curl consistent directed graph on a triangulated cycle can be achieved as a LOG (and also an expert graph).*

Proof. We note that any cycle triangulation can be generated by merging a smaller cycle triangulation and a cycle of length 3 (a triangle) which share a single edge, which exactly matches the setting in Lemma 32.

The base case of this recursive process is a triangle, for which we have already argued that a decomposition exists. The proof then follows from using Lemma 32 as an inductive step. $\square$

A decomposition for a dense graph also works as a decomposition on a sparser graph that has only a subset of the dense graph's edges. Therefore, the curl condition is also sufficient for any graph on a subset of the edges in a triangulated cycle (including a cycle of length > 3).

7.5 Synthetic experts

Bounding missing edges

The definition of expert graphs is given on digraphs that are not necessarily complete. If we are given an expert graph with missing edge-weights, many possible ASRs or ARs could generate the given edge-weights. Each of these possible models gives different values on the *missing* edge-weights. Hence, in the absence of a relevant expert, synthetic experts will bound the space of possible edge-weights. To give these bounds, we will use the curl condition.[1]

[1] We have shown the curl condition to be necessary, but not sufficient to describe the space of possible expert graphs (via equivalence with LOGs). As a result, stronger bounds may be possible.

For this section, we introduce new notation in which a bar indicates a reversal of a path or cycle. For example, if $c = (c_1, c_2, c_3)$ then $\overline{c} = (c_3, c_2, c_1)$.

Definition 37. Let $\mathcal{b} = (b_1, b_2, \cdots, b_\ell)$ denote an ordered set of vertices/classes in expert graph $G = (\mathcal{Y}, \mathcal{E}, F(\cdot))$ with $(b_i, b_{i+1}) \in \mathcal{E}\ \forall\ i \in [\ell - 1]$. The **weight** of a path is given by:

$$\text{Weight}(F, \mathcal{b}) \coloneqq \sum_{i=1}^{\ell-1} F(b_i, b_{i+1}). \tag{7.17}$$

The path weight is very similar to the curl, but is defined on a path that does not loop back to its starting point.

Lemma 33. *Consider a curl consistent graph $G = (\mathcal{Y}, \mathcal{E}, F(\cdot))$, for which $(s, t) \notin \mathcal{E}$. Let $\mathcal{B}^{(st)}$ be the set of paths in G beginning at vertex s and ending at vertex t:*

$$\mathcal{B}^{(st)} \coloneqq \{\mathcal{b}\ :\ b_1 = s, b_{|\mathcal{b}|} = t\}.$$

Then,

$$1 - \min_{\overline{\mathcal{b}} \in \mathcal{B}^{(ts)}} \text{Weight}(F, \overline{\mathcal{b}}) < F(s, t) < \min_{\mathcal{b} \in \mathcal{B}^{(st)}} \text{Weight}(F, \mathcal{b}). \tag{7.18}$$

Proof. Consider the ℓ-length cycle $c = (s, \ldots, t)$. The curl condition gives

$$1 < \text{Curl}(F, \overline{c}) < \ell - 1. \tag{7.19}$$

Notice that, for the path version of c, $\mathcal{b} = (s, \ldots, t)$

$$\text{Curl}(F, \overline{c}) = \text{Weight}(F, \overline{\mathcal{b}}) + F(s, t).$$

Substituting this into Equation 7.19 gives

$$1 - \text{Weight}(F, \overline{\mathcal{b}}) < F(s, t) < \ell - 1 - \text{Weight}(F, \overline{\mathcal{b}}).$$

Now, by applying $F(b_i, b_j) = 1 - F(b_j, b_i)$, we can simplify the upper bound to get

$$1 - \text{Weight}(F, \overline{\mathcal{b}}) < F(s, t) < \text{Weight}(F, \mathcal{b}). \tag{7.20}$$

Recall that $\mathcal{b}$ and $\overline{\mathcal{b}}$ were chosen without loss of generality. Obtaining the optimal bounds over all paths yields the desired result. □

A number of shortest-path algorithms for weighted directed graphs, such as Floyd-Warshall (Cormen et al., 2022), can be applied to find the best bounds given by Lemma 33. An example of this process is illustrated in Figure 7.4.

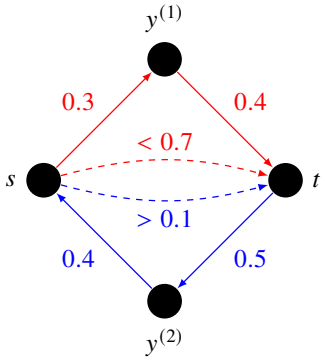


Figure 7.4: The shortest path from $s \to t$, given by $s \to y^{(1)} \to t$ gives $f_d(s,t) < 0.7$. The shortest path from t to s, given by $t \to y^{(2)} \to S$ gives $f_d(s,t) > 1 - .9 = .1$. Together, we get $F(s,t) \in (0.1, 0.7)$.

ζ-accurate synthetic experts

We can restrict the possible edge-weights of a synthetic expert to arbitrary precision.

Definition 38. For $\zeta > 0$ call the synthetic expert between $y^{(i)}$ and $y^{(j)}$ a **ζ-accurate synthetic expert** if for some lower bound $L \in [0,1]$, $f(y^{(i)}, y^{(j)}) \in (L, L+\zeta)$.

Lemma 34. *Given $G = (\mathcal{Y}, \mathcal{E}, F(\cdot))$, the synthetic expert between s and t is ζ-accurate if and only if there exists a cycle $c = (c_1, \ldots, c_\ell)$ with $(s,t) = (c_i, c_{i+1})$ for some i and* $\text{Curl}(F,c) \leq 1 + \zeta$.

Proof. Choose $s, t \in \mathcal{V}$ wlog and denote the shortest path:

$$\mathcal{s}^{(st)} = \arg\min_{\mathcal{b} \in \mathcal{B}^{(st)}} \text{Weight}(F, \mathcal{b}). \tag{7.21}$$

Recall from Lemma 33 that

$$1 - \text{Weight}(F, \mathcal{s}^{(ts)}) < f(s,t) < \text{Weight}(F, \mathcal{s}^{(st)}).$$

Let $L = 1 - W(F, \mathcal{s}^{(ts)})$. The desired gap between bounds is $\text{Weight}(F, \mathcal{s}^{(st)}) - \left(1 - \text{Weight}(F, \mathcal{s}^{(ts)})\right) \leq \zeta$, which is true if and only if

$$\text{Weight}(F, \mathcal{s}^{(ts)}) + \text{Weight}(F, \mathcal{s}^{(st)}) \leq 1 + \zeta.$$

Now consider the cycle $c^{(sts)} = \mathcal{s}^{(st)}\mathcal{s}^{(ts)}$, which is the shortest cycle through s and t. We have

$$\text{Curl}(G, c^{(sts)}) = \text{Weight}(F, \mathcal{s}^{(ts)}) + \text{Weight}(F, \mathcal{s}^{(st)}) \leq 1 + \zeta.$$

□

ζ-accurate synthetic experts describe convergence to fully determining a missing opinion without access to the relevant expert. A natural question is whether any curl consistent graph can be formed entirely using $\zeta-$ accurate synthetic experts. Lemma 35 shows that any curl consistent network of expert opinions can be formed using ζ-accurate synthetic experts.

Lemma 35. *Given a curl consistent $G = (\mathcal{Y}, \mathcal{E}, F)$, there exists another weighted digraph $G' = (\mathcal{Y} \cup \mathcal{Y}', \mathcal{E}', F')$ with $\mathcal{Y}' \cap \mathcal{Y} = \emptyset$ and $\mathcal{E} \cap \mathcal{E}' = \emptyset$ such that*

1. *All synthetic experts $e \in \mathcal{E}'$ in G' are ζ-accurate. That is, we have $F(e) \in (L^{(e)}, L^{(e)} + \zeta)$ for some lower bound $L^{(e)}$.*
2. *G' is also curl consistent.*

Proof. We give a constructive proof, for which an example is given in Figure 7.5. Begin with the set of classes $\mathcal{Y}$ and $\mathcal{E}' = \emptyset$. Now, for each $e = (y^{(i)}, y^{(j)}) \in \mathcal{E}$, do the following:

1. Add a vertex $y^{(ij)}$ to $\mathcal{Y}'$ and assign values to $F'(y^{(i)}, y^{(ij)})$ and $F'(y^{(ij)}, y^{(j)})$ to create a path from $y^{(i)} \to y^{(ij)} \to y^{(j)}$ of weight $F(e) + \frac{\zeta}{2}$.
2. Similarly add a vertex $y^{(ji)}$ to create a path from $y^{(j)} \to y^{(ji)} \to y^{(i)}$ of weight $1 - F(e) + \frac{\zeta}{2}$.

By Lemma 34, e is now a ζ-accurate synthetic classifier in G' that includes value $F(e)$.

If this process created a cycle c' in G' with $\text{Curl}(F', c') \leq 1$, then there must also be a cycle through G with $\text{Curl}(F, c) \leq 1$, which is a contradiction. Thus, we know our constructed G' is curl consistent. □

Feasibility of networks of synthetic experts

The bounds provided in Lemma 33 interact with each other when many experts are being synthesized on the same graph. For example, the bounds obtained for each edge may not be attainable simultaneously while remaining curl consistent (see Figure 7.6).

While the bounds for synthetic experts may not be simultaneously attainable, each value is individually attainable, though fixing a value for a single synthetic expert will induce new bounds on the leftover synthetic experts.

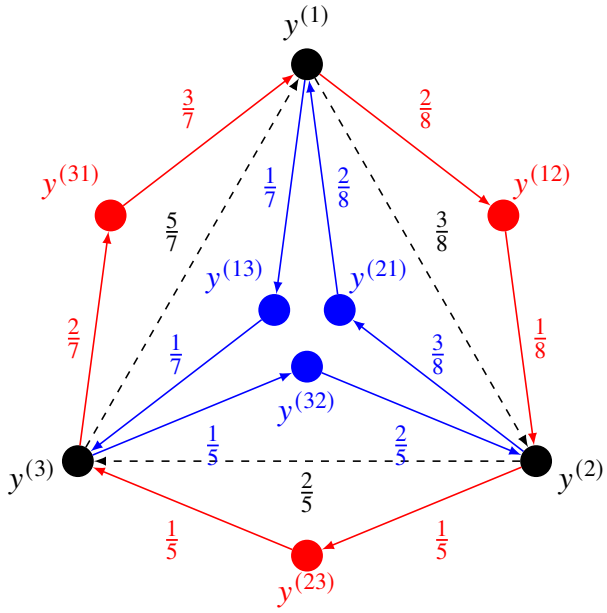


Figure 7.5: An example of how to create ζ-accurate synthetic experts that make up any curl consistent graph by adding additional paths. All edge weights given in color have a $+\frac{\zeta}{4}$ added to their weight that has been omitted to reduce clutter. Here, the cycle on $y^{(1)}, y^{(2)}, y^{(3)}$ is created by: (1) Adding shortest paths (shown in red on the other part of the cycle) with total weight equal to the desired edge weight $f(e) + \frac{\zeta}{2}$. (2) Adding shortest reverse paths (shown in blue on the inner part of the cycle) with total weight equal to $1 - f(e) + \frac{\zeta}{2}$.

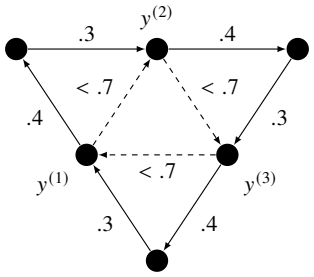


Figure 7.6: An example of how we cannot always achieve any combination of synthetic expert bounds. Here, a choice of just under .7 for all the synthetic experts (given in dashed lines) would violate the curl condition on cycle $y^{(1)} \to y^{(2)} \to y^{(3)} \to y^{(1)}$.

7.6 Conclusion

This chapter explores *networks of classifiers* trained on different tasks. We build intuition for understanding how these classifiers can give nontransitive results and develop a framework inspired by nontransitivity in voting theory. We proved equivalence between voting theory ARs and ASRs. We use this equivalence to further our understanding of necessary and sufficient conditions on the weights of expert graph/LOG edges, allowing us to quantify what levels of nontransitivity are allowed. We conjecture that the curl condition is sufficient to describe all possible *planar* LOGs/expert graphs. From insights into the role of composition in the emergence of nontransitivity, it may be possible to optimize networks of classifiers to minimize their nontransitive regions.

Chapter 8

CAUSAL INFORMATION SPLITTING

In this section, we present a specific setup in which counterfactual features obtained from auxiliary training tasks can be used to isolate Level 3 knowledge and improve robustness to contextual shifts. This work is published in B. Mazaheri, Mastakouri, et al., 2023.

8.1 Introduction

The principle assumption when building any (not necessarily causal) prediction model is access to relevant data for the task at hand. When predicting label Y from inputs $\mathbf{X}$, this assumption reads that the data is drawn from a (training) probability distribution $\mathbf{X}, Y$ that is identical to the distribution that will generate its use-cases (target distribution).

Unfortunately, the dynamic nature of real-world systems makes obtaining perfectly relevant data difficult. Data-gathering mechanisms can introduce sampling bias, yielding distorted training data. Even in the absence of sampling biases, populations, environments, and interventions give rise to distribution shifts in their own right. For example, Zech et al., 2018 found that convolutional neural networks to detect pneumonia from chest radiographs often relied on site-specific features, including the metallic tokens indicating laterality and image processing techniques. This resulted in poor generalization across sites. Understanding these inter-site breakdowns in performance is essential to safety-critical domains such as healthcare.

Generalization and Invariant Sets The first attempts at handling dissociation between training and target distributions involved gathering unlabeled samples of the testing distribution. Within domain generalization (DG), covariate shift handles a shift in the distribution of $\mathbf{X}$ (Shimodaira, 2000) and label shift handles a shifting $\Pr(Y)$ (Schweikert et al., 2008). DG often assumes a stationary label function $\Pr(Y \mid \mathbf{X})$, which is extremely limiting in real-life applications. To address these limitations, one can assume the label function is stationary for a *subset* of the covariates in $\mathbf{X}$, called an **invariant set** in Muandet, Balduzzi, and Schölkopf, 2013 and Rojas-Carulla et al., 2018.

One approach to finding invariant sets has been to capture shifting information from a collection of datasets (Rojas-Carulla et al., 2018; Magliacane et al., 2018). Such techniques require access to a comprehensive set of datasets that represent all possible shiftings. A causal perspective developed in Storkey et al., 2009 and Judea Pearl and Bareinboim, 2011 instead uses graphical modeling via **selection diagrams** to model shifting mechanisms. This approach requires access to multiple datasets to learn these mechanisms, but does not require that those datasets span the entire space of possible shifting. Such approaches also allow the use of domain expert knowledge when building selection diagrams. A detailed comparison of stability in the causal and anticausal scenario is given in Schölkopf et al., 2012.

Contributions The causal perspective to distribution shift is obscured when we lack direct measurements of the causes and effects of Y. Such settings arise from noisy measurements, privacy concerns, as well as abstract concepts that cannot be easily quantified (such as "work ethic" or "interests"). Instead, we will focus on a setting where we only measure *proxies* for the causes and effects of Y, see Fig. 8.1 for an example. All of these proxies are descendants of $\mathbf{U}$ — a case which is common in medicine, where the measured variables are often blood markers (or other tests) that are indicative of an underlying condition.

The proxy setting is difficult to address in standard framework. While previous approaches to partially observed systems suggest restricting model inputs to those on stable paths (Subbaswamy and Saria, 2018), no observed proxies satisfy this condition in our setting. That is, the proxies of the unobserved causes are insufficient to fully block the environmental shifts of those causes.

We will use concepts from causal inference and information theory to define and study the **Proxy-based Environmental Robustness** (PER) problem. Our framework will demonstrate that perfection is indeed the enemy of good – some variables (although with an unstable relationship to the target) should still be included as features to build a model with improved stability.

A primary goal of this paper will be to distinguish between proxies that are "helpful" or "hurtful" for stability - a property that they inherit from their parents (of which they are proxies). The stability of these unobserved variables depends on their causal structure, which is unobserved. We will present a strategy for feature selection based on properties that propagate from the underlying causal structure to its observed proxies. Specifically, we will build on the observation that post-selecting on a single

value of the prediction label Y induces a special independence pattern, which the proxies also inherit. We use this to classify proxies from partial knowledge of a few "seeds" — a technique we call **proxy bootstrapping**.

It is possible that some proxy variables will contain information about both stable and unstable hidden variables. We call these **ambiguous proxies** because it is unclear whether they will improve or worsen the model's transportability. Inspired by node splitting (Subbaswamy and Saria, 2018), we introduce a method we call **causal information splitting (CIS)**, which can improve stability of our models at no cost (and even some benefit) to the distribution shift robustness. CIS isolates stabilizing information using auxiliary prediction tasks that answer counterfactual questions about the covariates. While theoretical guarantees require a number of assumptions, we demonstrate the surprising ability of CIS to separate stabilizing information from ambiguous variables on synthetic data experiments with relaxed assumptions. Furthermore, we utilize CIS to enhance a prediction task on U.S. Census data that was strongly affected by the COVID-19 pandemic. While plenty of experiments have confirmed that techniques for robust models do not consistently provide benefits over empirical risk minimization (Gulrajani and Lopez-Paz, 2021), our proposed technique provides benefits for an income prediction task in the majority of tested states.

8.2 Related Work

There is an increasing body of work on domain generalization, see Quinonero-Candela et al., 2008 for an overview. While we focus on proactively modeling shifts, work on invariant risk minimization (Arjovsky et al., 2019; Bellot and Schaar, 2020) has approached this problem when given access to the shifted data on which the models will be used. Recent work further generalizes to unseen environments constituting mixtures (Sagawa et al., 2019) and affine combinations (D. Krueger et al., 2021). Data from multiple environments can also be used for causal discovery (Peters, Bühlmann, and Meinshausen, 2016b; Heinze-Deml, Peters, and Meinshausen, 2018; Peters, Bühlmann, and Meinshausen, 2016a).

Another line of work seeks robustness to small adversarial changes in the input that should not change the output (with attacks, e.g. Croce and Hein, 2020 and defenses, e.g. A. Sinha, Namkoong, and Duchi, 2018). Moving from small changes to potentially bigger interventions, work on counterfactual robustness and invariance, introduces additional regularization terms (Veitch et al., 2021; Quinzan et al., 2022).

Our work differs by allowing for interventions that change the label.

We do not address the tradeoffs associated with robustness and model accuracy in this paper. Such tradeoffs are a natural consequence of restricting the input information for our model, since unstable information is still useful in unperturbed cases. This problem is generally addressed by Oberst et al., 2021; Rothenhäusler et al., 2021 via regularization.

The information theoretic decomposition presented in this paper is deeply related to the Partial Information Decomposition, which has been a subject of growing interest in information theory (Bertschinger et al., 2014; Banerjee et al., 2018; Williams and Beer, 2010; Gurushankar, Venkatesh, and Grover, 2022; Venkatesh and Schamberg, 2022). The use of the PID in causal structures was pioneered by Dutta, Venkatesh, et al., 2020 for fairness (Dutta and Hamman, 2023). Our method of extracting counterfactual features using auxiliary training tasks removes information from proxies that is unique to bad features, even though those features are not observed.

8.3 Background

General Notation Uppercase letters denote random variables, while lowercase letters denote assignments to those random variables. Bold letters denote sets/vectors. The paper will use concepts from information theory, with $\mathcal{H}(A)$ indicating the **entropy** of A, $\mathcal{I}(A : B)$ indicating the **mutual information** between A, B, and $\mathcal{I}(A : B : C)$ indicating the **interaction information** between A, B, C.

Causal Graphical Models Graphically modeling distribution shift makes use of causal DAGs. For a causal DAG $\mathfrak{G} = (\mathbf{V}, \mathbf{E})$, the joint probability distribution factorizes according to the local Markov condition,

$$\Pr(\mathbf{v}) = \prod_{v \in \mathbf{v}} \Pr(v \mid \mathbf{pa}_{\mathbf{v}}^{\mathfrak{G}}(V)).$$

$\mathbf{PA}^{\mathfrak{G}}(V), \mathbf{CH}^{\mathfrak{G}}(V)$ denote the parents and children of V in $\mathfrak{G}$. Following the uppercase/lowercase convention, $\mathbf{pa}_{\mathbf{v}}(V)$ is an assignment to $\mathbf{PA}(V)$ using the values in $\mathbf{v}$.[1] $\mathbf{DE}^{\mathfrak{G}}(V)$ and $\mathbf{AN}^{\mathfrak{G}}(V)$ denote the descendants and ancestors respectively. $\mathbf{FM}(V) = \mathbf{PA}(V) \cup \mathbf{CH}(V)$ denotes the "family."

We will rely on the concepts of ***d*-separation** and **active paths** to discuss the independence properties of Bayesian networks, which are indtroduced in Chapter 2. See Judea Pearl, 2009 for a more extensive study.

[1] $\mathbf{PA}(V) \subseteq \mathbf{V}$

Active Path Notation In addition to using $A \perp\!\!\!\perp_d B \mid C$ to indicate d-separation conditioned on C, we will develop a notation to refer to sets of variables that act as "switches" for d-separation. $\mathbf{A} \multimap \mathbf{C} \multimap \mathbf{B}$ means that we have both $\mathbf{A} \not\perp\!\!\!\perp_d \mathbf{B}$ and $\mathbf{A} \perp\!\!\!\perp_d \mathbf{B} \mid \mathbf{C}$. Conversely, we have $\mathbf{A} \multimap \mathbf{C} \multimap \mathbf{B}$ if $\mathbf{A} \perp\!\!\!\perp_d \mathbf{B}$, but $\mathbf{A} \not\perp\!\!\!\perp_d \mathbf{B} \mid \mathbf{C}$ (i.e. conditioning on $\mathbf{C}$ renders $\mathbf{A}$ and $\mathbf{B}$ d-connected).

Graphically Modeling Distribution Shift Borrowing terms from Magliacane et al., 2018, we will begin with a graphical model $\mathfrak{G} = (\mathbf{V} \cup \mathbf{U})$, calling $\mathbf{U} \cup \mathbf{V}$ the **system variables** with (un-)observed variables. In addition, we are also given a set of context variables $\mathbf{M}$, which model the mechanisms that shift our distribution. The augmentation of $\mathfrak{G}$ with $\mathbf{M}$ gives what we call the **distribution shift diagram** (DSD), $\mathfrak{G}^+ = (\mathbf{V} \cup \mathbf{U} \cup \mathbf{M}, \mathbf{E} \cup \mathbf{E_M})$, for which $\mathfrak{G}$ is a subgraph, with additional vertices $\mathbf{M}$ introducing shifts along $\mathbf{E_M}$. $\mathbf{X} \subseteq \mathbf{V}$ such that $\Pr(Y \mid \mathbf{X}) = \Pr(Y \mid \mathbf{X}, \mathbf{M})$ is called an "invariant set" because it blocks all possible influence from the mechanisms of the dataset shift. Judea Pearl and Bareinboim, 2011 shows this framework is capable of modeling sampling bias and population shift.

8.4 Setting

This paper will consider the **Proxy-based Environmental Robustness (PER)** setting. PER focuses on the role of proxy variables in feature selection by assuming *all* of the causes and effects $\mathbf{U} = \mathbf{FM}(Y)$ are unobserved.[2] We are given access to a list of "visible proxy variables" $\mathbf{V} \setminus \{Y\}$ which are descendants of at least one $U \in \mathbf{U}$. Hence, $\mathbf{V}$ can be thought of as the union of overlapping subsets $\mathbf{CH}(U)$ for each $U \in \mathbf{U}$.

We will assume that there are no edges directly within $\mathbf{U}$ or within $\mathbf{V}$, which we call **systemic sparsity**. See Figure 8.1 for an example of this setting. This assumption enforces two useful independence properties: (1) $V_i \perp\!\!\!\perp V_j \mid U$ for $V_i, V_j \in \mathbf{CH}(U)$ and (2) $U_i \perp\!\!\!\perp U_j \mid Y$ for $U_i \neq U_j \in \mathbf{U}$. Systemic sparsity guarantees that a discoverable causal structure exists within the unobserved variables and simplifies the interactions between the proxies.

We will build our theory on distribution shift diagrams $\mathfrak{G}^+ = (\mathbf{V} \cup \mathbf{U} \cup \mathbf{M}, \mathbf{E} \cup \mathbf{E_M})$ with one $M_i \in \mathbf{M}$ connected to a corresponding $U_i \in \mathbf{U}$. Each M_i models a different shifting mechanism for each unobserved cause and effect of Y. It is common to

[2] This assumption is not necessary but allows us to focus on more difficult questions that have not been answered by previous work. Namely, direct causes and effects can be visible or have perfect proxies without changing the results of the paper.

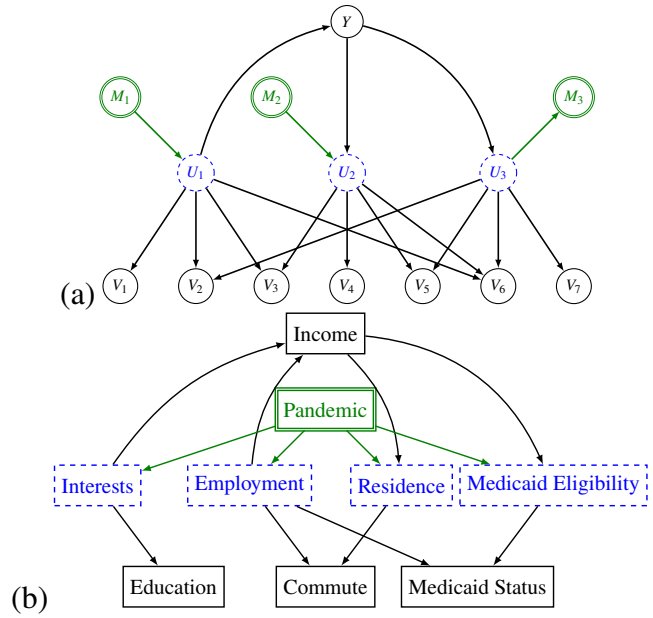


Figure 8.1: Examples of the $\mathfrak{G}^+$ considered for the paper. (a) shows a generic setup where U_1 is a hidden cause of Y, and U_2, U_3 are hidden effects. (b) shows a *plausible* model explaining the success of our real-data experiment in Section 8.7.

assume there is no direct shifting mechanism acting on Y — which comes without loss of generality since such a mechanism can be thought of as another unobserved cause (Judea Pearl and Bareinboim, 2011; Peters, Bühlmann, and Meinshausen, 2016b).

In this setting, a perfect invariant set $\mathbf{X}$ in which $Y \perp\!\!\!\perp_d \mathbf{M} \mid \mathbf{X}$ does not exist. PER will instead seek to minimize the influence of the context variables on our label function. Borrowing concepts from information theory, the task in PER corresponds to finding a set of features $\mathbf{X}$ that minimizes the conditional mutual information between the label and the environment. We call this quantity, $\mathcal{I}(Y : \mathbf{M} \mid \mathbf{X})$, the **context sensitivity**. To allow for feature engineering, we define these features to be the output of a function, $\mathbf{X} = F(\mathbf{V} \setminus \{Y\})$ which can capture higher-level representations of $\mathbf{V} \setminus \{Y\}$.

Challenges in PER The PER setting is difficult to address using existing methods. Building a model on the causes $\mathbf{PA}(Y)$ as in Schölkopf et al., 2012 is impossible because all of the causes are unobserved. Furthermore, finding a separating set as in Magliacane et al., 2018; Judea Pearl and Bareinboim, 2011 is also impossible for the same reason. Proxies can contain combinations of both stable and unstable information when they are connected to multiple $U \in \mathbf{U}$. Introduced in Subbaswamy

and Saria, 2018, "node splitting" requires knowledge of the structural equations that govern a vertex to remove unstable information from ambiguous variables, which can only be learned if the causes of the split node are observed. This requirement limits node splitting's power in the proxy setting.

Invertible Dropout Functions

We will demonstrate the failure of existing approaches in this setting using a counterexample built on structural equations models with cleanly interpretable entropic relationships. This construction will show the cost of restricting features to those with stable paths to the prediction variable Y, and serve as a framework for understanding the problem in general. For a discussion of relaxations, see Sec. 8.8 and for a demonstration that our method can work in real-world settings (where the assumption does not hold), see Sec.8.7.

Our restricted structural equations give edges from A to B described by an invertible function with "dropout" noise,

$$B^{(A)}(A) = \begin{cases} \mathcal{T}_{A,B}(A) & \text{with probability } \alpha_{A,B} \\ \phi & \text{with probability } 1 - \alpha \end{cases}. \tag{8.1}$$

$\mathcal{T}_{A,B}(\cdot)$ is a function that is invertible, with $\mathcal{T}_{A,B}(\phi) = \phi$. The probability that information from the parent is preserved is given by $\alpha_{A,B} \in [0, 1]$. We will refer to $B^{(A)}(A) \neq \phi$ as "transmission," and $\alpha_{A,B}$ as the "probability of transmission."[3] ϕ, called "null", is a value that represents the dropout, or the failure of the edge to "transmit".

The structural equation for a vertex B given its parents is a deterministic function of these $B^{(A)}$,

$$B = \mathcal{T}_B(\{B^{(A)}(A) \text{ for } A \in \mathbf{PA}(B)\}), \tag{8.2}$$

where $\mathcal{T}_B$ is not necessarily an invertible function.

For functions with many children, the probability that at least one of their children transmits is

$$\alpha_{A,\mathbf{CH}(A)} := 1 - \prod_{B \in \mathbf{CH}(A)} (1 - \alpha_{A,B}). \tag{8.3}$$

[3]The direction of the edge for these $\alpha_{A,B}$ will sometimes be arbitrary, in which case the ordering of the vertices is unimportant.

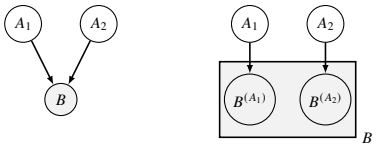


Figure 8.2: A diagram showing separability.

Separability and Faithfulness If $\mathcal{T}_B$ is invertible, we say that B is a separable variable, which means that a child B with more than one parent can be split into separate disconnected vertices $B^{(A)}$ for $A \in \mathbf{PA}(B)$, each with the structural equation given by Equation 8.1 (See Figure 8.2). Separable variables make up a special violation of faithfulness in that conditioning on separable colliders no longer opens up active paths, illustrated by Lemma 36.

Lemma 36 (Separability violates faithfulness)**.** *If* $U_1 \multimap V \multimap U_2$ *and* V *is separable, then* $U_1 \not\perp\!\!\!\perp_d U_2 \mid V$*, but* $U_1 \perp\!\!\!\perp U_2 \mid V$*.*

The proof follows from the definition of mutual information and the fact that $U_1 \perp\!\!\!\perp U_2 \mid V$.

Our setting will rely on the assumption of faithfulness of the sub-graph on the $\mathbf{U} \cup \{Y\}$ vertices for proxy bootstrapping, as is the case for algorithms attempting any degree of structure learning. Specifically, we will require that any active path between two proxies V_i, V_j that does not travel through any other vertices in $\mathbf{V}$ must imply statistical dependence (we call this "partial faithfulness"). When we move to causal information splitting, we will allow *specific* violations of faithfulness that come from separable proxies $\mathbf{V}$ in order to illustrate an ideal use-case of our method. This does not contradict partial faithfulness.

Transmitting Active Paths A convenient aspect of these structural equations is that α_{AB} controls the mutual information between A and its child $B^{(A)}$,

$$\begin{aligned}\mathcal{I}(A : B^{(A)}) =& \mathcal{H}(A) - \mathcal{H}(A \mid B^{(A)}) \\ =& \mathcal{H}(A) - \Pr(B^{(A)} = \phi)\mathcal{H}(A \mid B^{(A)} = \phi) \\ & - \Pr(B^{(A)} \neq \phi)\mathcal{H}(A \mid B^{(A)} \neq \phi).\end{aligned}$$

An important insight is that $\mathcal{H}(A \mid B^{(A)} = \phi) = 0$ and $\mathcal{H}(A \mid B^{(A)} \neq \phi) = \mathcal{H}(A)$. Applying this gives

$$\mathcal{I}(A : B^{(A)}) = \mathcal{H}(A) - (1 - \alpha_{A,B})\mathcal{H}(A) = \alpha_{A,B}\mathcal{H}(A).$$

This aspect generalizes to active single paths. For a length-2 path $A \to B \to C$, $\mathcal{I}(A : C) = \mathcal{I}(A : C^{(B)}) = \mathcal{H}(A) - \mathcal{H}(A \mid C^{(B)})$. Again, we can break up $\mathcal{H}(A \mid C^{(B)})$ into $\mathcal{H}(A \mid C^{(B)} = \phi) = 0$ and $\mathcal{H}(A \mid C^{(B)} \neq \phi) = \mathcal{H}(A)$. Hence, reasoning about mutual information reduces to the task of determining the probability that one of the endpoints is null. In our setup, the dropout events of different edges are independent events. Hence, $\mathcal{I}(A : C) = \alpha_{A,B}\alpha_{B,C}\mathcal{H}(A)$.

Conditioning adds an additional complication. Notice that transmitting active paths can "transfer" a conditioning. That is, $\mathcal{H}(A \mid x) = 0$ when there is only one active path between A and X (or X to A) and it transmits. In the next section, we will study two cases that emerge in the PER problem: colliders and non-colliders.

8.5 Context Sensitivity

We quantify robustness through the dependence on environmental mechanisms and the label function.

Definition 39 (Context sensitivity)**.** Context sensitivity of a mechanism $M \in \mathbf{M}$ is defined as $\mathcal{I}(Y : M \mid \mathbf{X})$.

If $\mathbf{X}$ d-separates $\mathbf{M}$ from Y, the context sensitivity is 0 and training on $\mathbf{X}$ to predict Y yields a model that is robust across environments $\mathbf{M}$.

We are usually most concerned with the success of our prediction models, something that is limited by the "relevance", $\mathcal{I}(Y : \mathbf{X})$, of our input. This concept is related to context sensitivity, and we can rewrite the sensitivity in terms of the expected relevance across environments.

$$\begin{aligned}\mathcal{I}(Y : M \mid \mathbf{X}) &= \mathcal{I}(Y : M) - \mathcal{I}(Y : M : \mathbf{X}) \\ &= \mathcal{I}(Y : M) - \mathcal{I}(Y : \mathbf{X}) + \mathcal{I}(Y : \mathbf{X} \mid M).\end{aligned}$$

Redundancy

Recall that in our setting we assume that all direct causes and effects are unobserved. This unobserved set of parents gives rise to an invariant set $\mathbf{S} \subseteq \mathbf{U}$[4]. We seek to identify a subset of visible proxies $\mathbf{X} \subseteq \mathbf{V}$ to extract information about $\mathbf{S}$.

Definition 40. For a specific U, we call $\mathcal{I}(U : \mathbf{X}) = \mathcal{H}(U) - \mathcal{H}(U \mid \mathbf{X})$ the **redundancy** between U and $\mathbf{X}$.

[4]The Markov boundary of Y would also give an invariant set, but could include vertices in $\mathbf{M}$ that are parents of effects of Y.

Lemma 37. *In the dropout function setting, let* $\mathbf{CH_X}(U) := \mathbf{CH}(U) \cap \mathbf{X}$.

$$\mathcal{J}(U : \mathbf{X}) = \alpha_{U,\mathbf{CH_X}(U)}\mathcal{H}(U).$$

Redundancy in the dropout function setting is controlled by our choice of $\mathbf{X}$ via $\alpha_{U,\mathbf{CH_X}(U)}$, the probability of transmission to at least one child.

Our graphical assumptions ensure that only one potential active path exists between each $M \in \mathbf{M}$ and Y — hence each vertex acts as either a collider or a non-collider in the interaction of M and Y (and does not do both). We now demonstrate that redundancy with stable (non-collider) variables generally improves our context sensitivity, whereas redundancy with unstable (collider) variables worsens it.

"Good" U If M_i and Y do not form a collider at $U_i \in \mathbf{U}$, we say $U_i \in \mathbf{U}^{\text{GOOD}}$. From d-separation, we have that $M_i \perp\!\!\!\perp_d Y \mid U_i$ for all $U_i \in \mathbf{U}^{\text{GOOD}}$. For an example, $\mathbf{U}^{\text{GOOD}} = \{U_1, U_3\}$ in Figure 8.1. Let $\mathbf{CH_X}(U_i) = \mathbf{CH}(U_i) \cap \mathbf{X}$.

Lemma 38 (Redundancy with $\mathbf{U}^{\text{GOOD}}$)**.** *In the dropout function setting, for some* $U_i \in \mathbf{U}$*, if corresponding* $M_i \rightsquigarrow U_i \rightsquigarrow Y$*, then*

$$\mathcal{J}(M_i : Y \mid \mathbf{X}) = \alpha_{M_i,U_i}(1 - \alpha_{U_i,\mathbf{CH_X}(U_i)})\alpha_{U_i,Y}\mathcal{H}(M_i).$$

Lemma 38 comes from multiplying the probability of transmission of each edge along the path M_i, U_i, Y. We also pick up a term requiring that the $U_i, \mathbf{X}$ edges do not transmit, in which case conditioning on $\mathbf{X}$ would reduce the entropy of U to nothing and close off the path.

"Bad" U The inclusion of $\mathbf{CH}(U_i)$ in $\mathbf{X}$ could open up active paths via colliders of the form $M_i \rightarrow U_i \leftarrow Y$. We call the set of these variables $\mathbf{U}^{\text{BAD}}$. For an example, $\mathbf{U}^{\text{BAD}} = \{U_2\}$ in Figure 8.1.

Lemma 39 (Redundancy with $\mathbf{U}^{\text{BAD}}$)**.** *In the dropout function setting,* $U_i \in \mathbf{U}, \mathbf{X} \subseteq \mathbf{V}$*, if* $M_i \rightsquigarrow U_i \rightsquigarrow Y$ *then*

$$\mathcal{J}(M_i : Y \mid \mathbf{X}) = \alpha_{U_i,\mathbf{CH}_X(U_i)}\mathcal{J}(M_i : Y \mid U_i)$$

Lemma 39 demonstrates that there are still proxies for which inclusion hurts our model's robustness. Similar concepts can be demonstrated via upper bounds when we allow arbitrary sets of structural equations (see B. Mazaheri, Mastakouri, et al., 2023). Optimizing these upper bounds does not give a guarantee of optimality, but can still point towards a general improvement.

Feature Selection Implications

The proxy graphical setup requires $\mathbf{X} \multimap \mathbf{U} \multimap Y$, meaning the relevance of our input is upper bounded by the redundancy with U, $\mathcal{I}(\mathbf{X} : Y) \leq \mathcal{I}(\mathbf{U} : \mathbf{X})$.

Lemma 38 shows that proxies of $\mathbf{U}^{\text{GOOD}}$ help build accurate and universal models, while Lemma 39 shows that proxies of $\mathbf{U}^{\text{BAD}}$ can trade universality for domain-specific accuracy. Of course, proxies need not lie neatly in these two classes — many proxies contain a combination of universally-relevant and domain-relevant features. This suggests multiple classes of proxy variables.

Definition 41.

$$\mathbf{V}^{\text{GOOD}} := \mathbf{CH}(\mathbf{U}^{\text{GOOD}}) \setminus \mathbf{CH}(\mathbf{U}^{\text{BAD}}) \tag{8.4}$$

$$\mathbf{V}^{\text{BAD}} := \mathbf{CH}(\mathbf{U}^{\text{BAD}}) \setminus \mathbf{CH}(\mathbf{U}^{\text{GOOD}}) \tag{8.5}$$

$$\mathbf{V}^{\text{AMBIG}} := \mathbf{CH}(\mathbf{U}^{\text{BAD}}) \cap \mathbf{CH}(\mathbf{U}^{\text{GOOD}}) \tag{8.6}$$

The behavior of $\mathbf{V}^{\text{GOOD}}$ in the dropout function setting shows how restricting models to invariant features fails; a high redundancy with $\mathbf{U}^{\text{GOOD}}$ is beneficial for the context sensitivity even though the paths from the proxies are unstable. Inclusion of $\mathbf{V}^{\text{GOOD}}$ in $\mathbf{X}$ improves context sensitivity even though $\mathbf{V}^{\text{GOOD}}$ is not made up of direct causes (as suggested by Schölkopf et al., 2012) or invariant features (as suggested by Magliacane et al., 2018 and (Subbaswamy and Saria, 2018)).

For feature selection, an obvious strategy is to choose $\mathbf{X} = \mathbf{V}^{\text{GOOD}}$, avoid $\mathbf{V}^{\text{BAD}}$, and potentially try using some elements in $\mathbf{V}^{\text{AMBIG}}$. In the next section we will explore how we can use non-invertible functions to transform these $\mathbf{V}^{\text{AMBIG}}$ into $\mathbf{V}^{\text{GOOD}}$.

Proxy Bootstrapping

Given the robustness implications of the different classes of V, their partitioning into good, bad, and ambiguous partitions will be important. We will now demonstrate how to harness partial information to determine these partitions and classify proxies. This step is optional if the role of each proxy is already understood (as is the case when the DAG is known). The results in this subsection will *only* require the graphical assumptions of the PER setting — i.e. systemic sparsity, partial faithfulness, and an independent shifting mechanism M_i for each $U_i \in \mathbf{U}$.

We begin with an observation about the independence structure of the conditional probability distribution on Y.

Lemma 40 (Linking related proxies)**.** *Within the graphical constraints of PER, if* $V_i \not\perp\!\!\!\perp_d V_j \mid Y$*, then either they have a shared parent (*$\mathbf{PA}(V_i) \cap \mathbf{PA}(V_j) \neq \emptyset$*) or they both have at least one parent that is a cause of* Y *(i.e.* $\mathbf{PA}(V_i) \cap \mathbf{PA}(Y) \neq \emptyset$ *and* $\mathbf{PA}(V_j) \cap \mathbf{PA}(Y) \neq \emptyset$*).*

Definition 42. For a DSD $\mathfrak{G}^+ = \{\mathbf{V} \cup \mathbf{U} \cup \mathbf{M}, \mathbf{E}\}$, define the dependence graph $\mathfrak{G}_Y = (\mathbf{V}, \mathbf{E}_Y)$ to be an undirected graph with edges $(V_i, V_j) \in \mathbf{E}_Y$ iff $V_i \not\perp\!\!\!\perp_d V_j \mid Y$.

Lemma 40 tells us that $\mathfrak{G}_Y$ will have a clique on the sets $\mathbf{CH}^{\mathfrak{G}}(U)$ for $U \in \mathbf{U}$. Furthermore, conditioning on Y links its causes, so $\mathfrak{G}_Y$ has one large clique on $\mathbf{CH}^{\mathfrak{G}}(\mathbf{PA}(Y))$. This clique structure can be utilized to enhance partial knowledge of $\mathbf{CH}(\mathbf{U}^{\text{GOOD}})$ and $\mathbf{CH}(\mathbf{U}^{\text{BAD}})$. In this sense, "birds of a feather flock together" — information about each clique's proxies can be a determined from understanding a single member of that clique.

Lemma 41 (Information about seed proxies spreads)**.** *If* $V_i \in \mathbf{V}^{\text{GOOD}}$ *then all neighbors of* $V_j \in \mathbf{NB}^{\mathfrak{G}_Y}(V_i)$ *are not in* $\mathbf{V}^{\text{BAD}}$ *- i.e.* $V_j \in \mathbf{V}^{\text{GOOD}} \cap \mathbf{V}^{\text{AMBIG}}$*. If* $V_i \in \mathbf{V}^{\text{BAD}}$ *then all neighbors of* $V_j \in \mathbf{NB}^{\mathfrak{G}_Y}(V_i)$ *are not in* $\mathbf{V}^{\text{GOOD}}$ *- i.e.* $V_j \in \mathbf{V}^{\text{BAD}} \cap \mathbf{V}^{\text{AMBIG}}$.

Lemma 41 suggests an algorithm for bootstrapping the sets $\mathbf{V}^{\text{GOOD}}, \mathbf{V}^{\text{BAD}}, \mathbf{V}^{\text{AMBIG}}$ from a set of "seed" vertices $\mathbf{V}^* \subseteq \mathbf{V}$ with known set memberships.

1. Construct $\mathfrak{G}_Y$ according to Definition 42 using conditional independence tests.
2. For each $V^* \in \mathbf{V}^*$, if $V^* \in \mathbf{V}^{\text{GOOD}}$ then add a "good" label to $\mathbf{NB}(V^*)$. If $V^* \in \mathbf{V}^{\text{BAD}}$ then add a "bad" label to $\mathbf{NB}(V^*)$.
3. All $V \in \mathbf{V} \setminus \mathbf{V}^*$ with both "good" and "bad" labels receive an "ambigious" label instead.

Theorem 6 (Proxy bootstrapping works)**.** *Upon termination of proxy bootstrapping all vertices with a single label are correctly described if :*

1. *Partial faithfulness holds.*
2. $\mathbf{V}^*$ *has at least one* $V^* \in \mathbf{V}^* \cap \mathbf{CH}(U)$ *for each* $U \in \mathbf{U}^{\text{GOOD}} \cap \mathbf{CH}(Y)$.
3. $\mathbf{V}^*$ *has at least one* $V^* \in \mathbf{V}^* \cap \mathbf{CH}(\mathbf{PA}(Y))$.
4. $\mathbf{V}^*$ *has at least one* $V^* \in \mathbf{V}^*$ *for each* $U \in \mathbf{U}^{\text{BAD}}$.

8.6 Causal Information Splitting

This section will expand our theory into **feature engineering**, which allows us to build inputs on functions of $\mathbf{V}$. A main takeaway from Section 8.5 was that we should build models using proxies for $\mathbf{U}^{\text{GOOD}}$ and avoid using features that are proxies for $\mathbf{U}^{\text{BAD}}$. The extension of this to engineered features is to build a model on *functions* of proxies for which the output of those functions *is* related to $\mathbf{U}^{\text{GOOD}}$ and *not* related to $\mathbf{U}^{\text{BAD}}$. We present two lemmas to formalize this notion.

Let $\widetilde{\mathbf{CH}}_{\mathbf{X}}(U_i)$ be the children or functions of children of U_i in $\mathbf{X}$. Lemma 42 shows that building models with more redundancy with $\mathbf{U}^{\text{GOOD}}$ (i.e. lower $\mathcal{H}(U_i \mid \widetilde{\mathbf{CH}}_{\mathbf{X}}(U_i))$ improves our context sensitivity in the dropout function setting.[5]

Lemma 42 (Engineering redundancy for $\mathbf{U}^{\text{GOOD}}$)**.** *In the dropout function setting, if* $U_i \in \mathbf{U}^{\text{GOOD}}$ *then*

$$\mathcal{J}(M_i : Y \mid \mathbf{X}) = \alpha_{M_i,U_i}\alpha_{U_i,Y}\mathcal{H}(U_i \mid \widetilde{\mathbf{CH}}_{\mathbf{X}}(U_i)).$$

Of course, even good proxies are related to $\mathbf{U}^{\text{BAD}}$ through their connection to Y, so $\mathbf{X} \perp\!\!\!\perp \mathbf{U}^{\text{BAD}}$ is impossible. Instead, Lemma 43 tells us that if we avoid redundancy with $\mathbf{U}^{\text{BAD}}$ after conditioning on Y, we do not pick up any context sensitivity from the associated shifting mechanisms.

Lemma 43 (Avoiding redundancy with $\mathbf{U}^{\text{BAD}}$)**.** *For some* $U_i \in \mathbf{U}^{\text{BAD}}$*, if we maintain* $\mathcal{J}(U_i : \mathbf{X} \mid Y) = 0$*, then* $\mathcal{J}(M_i : Y \mid \mathbf{X}) = 0$*.*

Recall that ambiguous proxies contain information about both $\mathbf{U}^{\text{GOOD}}$ and $\mathbf{U}^{\text{BAD}}$. The inclusion of an ambiguous proxy V_A improves context sensitivity because of its redundancy with $\mathbf{U}^{\text{GOOD}}$ via Lemma 42. This section will develop a technique for filtering V_A into $F(V_A)$, which will satisfy the conditions in Lemma 43. To do this, we will require separability.

Separable Ambigious Proxies Consider the setup in Figure 8.3, where $V_G \in \mathbf{V}^{\text{GOOD}}$, $V_B \in \mathbf{V}^{\text{BAD}}$, and $V_A \in \mathbf{V}^{\text{AMBIG}}$. V_A is generated by invertible $\mathcal{T}_A$, making it a **separable ambiguous proxy (SAP)**.[6] Splitting V_A into components allows us to isolate the origins of its ambiguity — the mixing of good information from $V_A^{(G)}$ and bad information from $V_A^{(B)}$.

[5] Appendix C shows that redundancy with $\mathbf{U}^{\text{GOOD}}$ lowers an upper bound on context sensitivity in more general cases

[6] While we may still be able to gain useful information from non-separable proxies, the tradeoffs are difficult to quantify and hence beyond the scope of this paper.

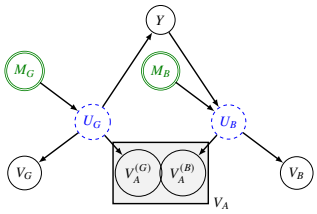


Figure 8.3: $V_G \in \mathbf{V}^{\text{GOOD}}$, $V_B \in \mathbf{V}^{\text{BAD}}$. $V_A \in \mathbf{V}^{\text{AMBIG}}$ is a linear transformation of two components, $V_A^{(G)}, V_A^{(B)}$, which are good and bad respectively.

Isolation Functions

We would like to isolate $V_A^{(G)}$ from V_A to avoid paying the penalty for $V_A^{(B)}$. We will do this using **isolation functions**.

Definition 43. We define an **isolation function** of V_i on V_A, with optional conditioning on y, to be

$$\begin{aligned} &F_{\text{ISO}(V_i)}(V_A \mid y) := \arg\min_F \mathcal{H}(F(V_A \mid y)) \\ &\text{such that} \mathcal{I}(F(V_A) : V_i \mid y) = \mathcal{I}(V_A : V_i \mid y). \end{aligned} \tag{8.7}$$

$F_{\text{ISO}(V_i)}(V_A \mid Y)$ gives a vector of functions with an entry for each $y \in Y$.

Note that isolation functions are sufficient statistics for V_i (Cover, 1999). Isolation involves maintaining the information about V_i while removing excess noise.

Recall from Lemma 43 that in order to avoid worsening context sensitivity, we want to ensure $\mathcal{I}(F(V_A) : \mathbf{U}^{\text{BAD}} \mid Y) = 0$. Isolation functions on SAPs are well designed for this purpose, because they enforce the independence properties of the isolated vertex on their outputs. In order to achieve $\mathcal{I}(F(V_A) : \mathbf{U}^{\text{BAD}} \mid Y) = 0$ while preserving as much information about $\mathbf{U}^{\text{GOOD}}$ as possible, an optimal isolation function would be to isolate $\mathbf{U}^{\text{GOOD}}$ using $F_{\text{ISO}(\mathbf{U}^{\text{GOOD}})}(V_A \mid Y)$.

Of course, we do not have access to $\mathbf{U}^{\text{GOOD}}$, so our next best option is to isolate $\mathbf{V}^{\text{GOOD}}$ using $F_{\text{ISO}(\mathbf{V}^{\text{GOOD}})}(V_A \mid Y)$, since $\mathbf{U}^{\text{BAD}} \perp\!\!\!\perp \mathbf{V}^{\text{GOOD}} \mid Y$. Lemma 44 shows that the output of $F_{\text{ISO}(V_G)}(V_A \mid Y)$ behaves like a good proxy if $V_G \in \mathbf{V}^{\text{GOOD}}$ and V_A is a SAP.

Lemma 44 (Isolating $\mathbf{V}^{\text{GOOD}}$ behaves like $\mathbf{V}^{\text{GOOD}}$)**.** *For $V_G \in \mathbf{V}^{\text{GOOD}}$ and $U_B \in \mathbf{U}^{\text{BAD}}$ and an isolation function $F_{\text{ISO}(V_G)}(V_A \mid Y)$,*

$$\mathcal{I}(U_B : F_{\text{ISO}(V_G)}(V_A \mid Y) \mid Y) = 0.$$

The benefit from $F_{\text{ISO}(V_G)}(V_A \mid Y)$'s information about $\mathbf{U}^{\text{GOOD}}$ is difficult to quantify for use with Lemma 42, but lower bounds can be obtained (B. Mazaheri, Mastakouri, et al., 2023).

Even without a quantification of improvement, Theorem 7 shows that isolation functions can avoid worsening the context sensitivity, while certain conditions can guarantee relevance gains for predicting Y.

Theorem 7 (CIS costs and benefits)**.** *Consider $V_G \in \mathbf{V}^{\text{GOOD}}$ and $V_A \in \mathbf{V}^{\text{AMBIG}}$ where V_A is a SAP. Also consider the isolation function $F_{\text{ISO}(V_G)}(V_A \mid Y)$. We will compare the context sensitivity of inputs $\mathbf{X} := \{V_G\}$ and $\mathbf{X}^+ := \{V_G, F_{\text{ISO}(V_G)}(V_A \mid Y))\}$. We claim that $\mathcal{I}(M : Y \mid \mathbf{X}^+) \leq \mathcal{I}(M : Y \mid \mathbf{X})$ for all $M \in \mathbf{M}$. Furthermore, if*

$$\mathcal{I}(F_{\text{ISO}(V_G)}(V_A \mid Y) : V_G) < \mathcal{I}(F_{\text{ISO}(V_G)}(V_A \mid Y) \quad : V_G \mid Y), \tag{8.8}$$

then the relevance improves: $\mathcal{I}(Y : \mathbf{X}^+) > \mathcal{I}(Y : \mathbf{X})$.

Theorem 7 tells us that using an isolation function helps when the function is more predictive of the isolated variable in the post-selected Y distribution than it is in the full distribution. This condition is sufficient but loose because it does not take into account direct effects from $\mathcal{I}(Y : F_{\text{ISO}(V_G)}(V_A \mid Y))$ (for which we have no guaranteed bounds). The proof is given in Appendix E.

Auxiliary Training Tasks

In the infinite sample regime, consider an "optimal" model $F(\cdot)$ that predicts V_i using input V_A. Optimal models should utilize all of the information available for prediction in their inputs, meaning $I(F(V_A) : V_i) = \mathcal{I}(V_A : V_i)$. Information theoretically, minimizing $\mathcal{H}(F_{V_i}(V_A))$ corresponds to reducing the outputs of $F_{V_i}(V_i)$ to equivalence classes wherein $\Pr(V_A \mid F_{V_i}(V_i) = f)$ is constant. This minimization corresponds to ensuring $F_{V_i}(V_i)$ does not over-fit to the empirical values of V_A using noise that is orthogonal to $\mathbf{PA}(V_A)$.

Auxiliary training tasks can therefore be used in place of isolation functions: we can get an approximate isolation function, $\tilde{F}_{\text{ISO}(V_i)}(V_A)$, by training a model to predict V_i using input V_A. We do not give any theoretical results beyond intuition for this interpretation, but will support our claims with experiments in the next section.

Equation 8.8 in Theorem 7 also has a nice interpretation within the training context — the accuracy of the predictor must degrade when moving from the post-selected data

to the full dataset. More precisely, the conditions for improvement now translate to

$$\min_F \mathbb{E}[\text{Error}(F(V_A), V_G)] > \sum_y \Pr(y) \min_F (\mathbb{E}[\text{Error}(F(V_A), V_G) \mid y]), \tag{8.9}$$

which can easily be checked on our training data.

Suggested Overall Procedure

We propose the following procedure for building robust (low context-sensitivity) models in the PER problem.

1. Partition the data into constant $Y = y$ and determine cliques of dependence.
2. Using domain knowledge, identify seeds in $\mathbf{V}^{\text{GOOD}}$, $\mathbf{V}^{\text{BAD}}$ for proxy bootstrapping (Sec. 8.5).
3. Perform CIS on $\mathbf{V}^{\text{AMBIG}}$ (Sec. 8.6).
4. Build a prediction model for Y using $\mathbf{V}^{\text{GOOD}}$ and the CIS-engineered $\mathbf{V}^{\text{AMBIG}}$.

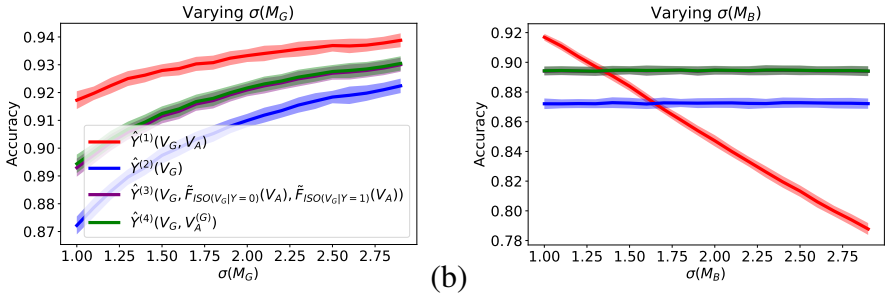
8.7 Experiments

We will now demonstrate the effectiveness of these methods on synthetic and real world data. Full code for both of these experiments is available at `https://zenodo.org/badge/latestdoi/651823136`.

Experiments on Synthetic Data

We generate data for the DAG in Figure 8.3 based on normal distributions, see details of the setup in Section 8.10. We vary the standard deviations of normally distributed M_G and M_B. The training data is drawn from $\sigma(M_G) = \sigma(M_B) = 1$, while the testing data varies both quantities and thus the influence of the context. We measure the accuracy of our feature engineering based on CIS, $\hat{Y}^{(3)}(V_G, \tilde{F}_{\text{ISO}(V_G)}(V_A))$, that utilizes the auxiliary task approximation to isolate V_A's predictive information about V_G. We compare it to $\hat{Y}^{(1)}(V_G, V_A)$ trained on $\mathbf{V}^{\text{GOOD}} \cup \mathbf{V}^{\text{AMBIG}}$ and $\hat{Y}^{(2)}(V_G)$ trained on only $\mathbf{V}^{\text{GOOD}}$. For a theoretical limit of CIS we also compare to $\hat{Y}^{(4)}(V_G, V_A^{(G)})$ although access to $V_A^{(G)}$ is usually not possible.

Results When comparing feature selection approaches, we observe in Figure 8.4 that including V_A results in higher accuracy of $\hat{Y}^{(1)}$ over $\hat{Y}^{(2)}$ when the shift acts on $\mathbf{U}^{\text{GOOD}}$ (a) or is small for $\mathbf{U}^{\text{BAD}}$ (b). However, the accuracy of $\hat{Y}^{(2)}$ deteriorates with bigger shifts in $\mathbf{U}^{\text{BAD}}$.



(a) (b)

Figure 8.4: Results from our experiments on synthetic data. Single standard deviation confidence intervals are shaded in the corresponding colors.

Our proposed method based on causal information splitting offers a middle ground. $\hat{Y}^{(3)}$ is able to maintain the same robustness as $\hat{Y}^{(2)}$ while taking advantage of some of the gains enjoyed by $\hat{Y}^{(1)}$ in (a). In fact, $\hat{Y}^{(3)}$ performs very similarly to $\hat{Y}^{(4)}$, which had a-priori knowledge of the SAP components and used only $V_A^{(G)}$. These improvements were achieved despite not meeting the sufficient condition for increasing relevance in Theorem 7.

Experiments on Census Data

We use US Census data processed through folktables F. Ding et al., 2021 to predict whether the income of a person exceeds 50k following Dua and Graff, 2017. To test out-of-domain generalization, prediction models were built on 2019 pre-pandemic data and evaluated on 2021 data during the pandemic.[7] As model inputs, we consider commute time (coded as JWMNP in the dataset), a flag whether the person received Medicaid, Medical Assistance, or any kind of government-assistance plan for those with low incomes or a disability (coded as HINS4) and education level (SCHL). This small feature set was purposefully selected to see a starker effect of including/excluding individual features, including a feature with relatively stable predictive power (education level) and two features heavily affected by the pandemic through increased work-from-home and medicaid's continuous enrollment provision.

Our auxiliary task from Sec. 8.6, referred to as *engineered features*, does not use HINS4 and JWMNP directly as input features to predict the income level. Instead it uses HINS4 and JWMNP to train two models predicting the education-level: One trained on examples with high income and one trained on examples with low income. These predictions based on HINS4 and JWMNP together with the actual education-level serve as input features to the final model. We compare the model

[7]We ignored the experimental release of 2020 data to ensure a starker distribution shift.

built on these engineered features to ones using all three features directly (*all features*) or using just the stable education feature (*limited features*).

We use logistic regression from sklearn with l1 regularization to build models based on the different feature sets that the three methods created. l1 regularization yielded better generalization than l2 regularization.

Table 8.1: Comparison of out-of-domain (2021) performance via mean of accuracy.

State	All Features	Engineered Features	Limited Features
CA	**0.712** ± 0.0011	**0.711** ± 0.0014	0.692 ± 0.0014
FL	**0.683** ± 0.0012	0.678 ± 0.0018	0.680 ± 0.0013
GA	0.689 ± 0.0025	**0.707** ± 0.0055	**0.709** ± 0.0029
IL	0.662 ± 0.0026	**0.689** ± 0.0033	0.684 ± 0.0019
NY	**0.707** ± 0.0022	**0.702** ± 0.0025	0.687 ± 0.0080
NC	**0.691** ± 0.0031	**0.684** ± 0.0034	**0.683** ± 0.0030
OH	0.689 ± 0.0022	**0.703** ± 0.0040	**0.696** ± 0.0029
PA	0.672 ± 0.0017	**0.695** ± 0.0023	0.688 ± 0.0022
TX	0.690 ± 0.0029	**0.712** ± 0.0028	**0.712** ± 0.0027
avg	0.688	**0.698**	0.692

Results Table 8.1 reports the mean and standard deviation of accuracies for 10 different test splits. For the F1 scores of the same experiment, see Section 8.10. Using all features leads to the best in-domain performance (see Section 8.10), but not necessarily the best out-of-domain performance. Dropping the ambiguous features hurts predictive power in limited feature models, but helps with robustness varies across the states: these limited models even perform better on 2021 data. Our proposed feature engineering using CIS achieves the best of both worlds, with the best mean out-of-domain accuracy of 0.698. It also achieves close to the best out-of-domain accuracy for 8 out of 9 states.

8.8 Discussion

In this paper we studied the challenging problem of building models that are robust to distribution shift when causes and effects of the target variable are unmeasured. Among the observed noisy proxies, we showed how to perform feature selection based on conditional independence tests and knowledge about some seed nodes.

After bootstrapping, we often have a significant number of ambiguous proxies, which have components that are both helpful and hurtful to our model's robustness. Through CIS, however, we showed how to isolate robust predictive power from these ambiguous proxies using auxiliary learning tasks. We proved that including these engineered features safely increases robustness in our setting, while also improving

accuracy. In our experiments on real census data under shifts due to the pandemic, we showed that the engineered features provided benefits for most states over using the ambiguous features directly or completely ignoring them. While our theoretical framework is involved, these experiments demonstrate improvements outside of our assumptions.

Relaxation of Assumptions A number of our assumptions can be softened. One softening of systemic sparsity would involve allowing edges within $\mathbf{U}$ so long as their dependence is relatively weak. Such a relaxation would involve using mutual information (or correlation) thresholds instead of independence tests. Sparsity assumptions may also be relaxed by building on ideas from mixtures of DAG structures from S. Gordon, B. Mazaheri, Yuval Rabani, et al. (2023).

The strongest assumption is that of separable ambiguous proxies. Under a softening of the separability assumption, we cannot guarantee that we have isolated only robust information from our ambiguous proxy — some unstable information associated with $\mathbf{U}^{\text{BAD}}$ may slip through. However, degrees of separability may still guarantee the benefit of the engineered feature.

While separability corresponds to invertability with linear functions, there are many examples of nonlinear that are separable. For example, when the effects of two causes have significantly different magnitudes they can be easily disentangled, such as fine and hyper-fine structures in atomic energy levels. Work on data fission (Leiner et al., 2022) may provide valuable insights to help understand the degrees of separability for different choices of functions.

8.9 Deferred Proofs

Proof of Lemma 36

Proof. We can rewrite the conditional mutual information making use of $U_1 \perp\!\!\!\perp U_2$ as follows.

$$\begin{aligned}
\mathcal{J}(U_1 : U_2 \mid V) &= -\mathcal{J}(U_1 : U_2 : V) \\
&= -\mathcal{J}(U_1 : V) + \mathcal{J}(U_1 : V \mid U_2) \\
&= -\mathcal{J}(U_1 : V^{(U_1)}) + \mathcal{J}(U_1 : V^{(U_2)} \mid U_2) \\
&= -\mathcal{J}(U_1 : V^{(U_1)}) + \mathcal{J}(U_1 : V^{(U_1)}) = 0.
\end{aligned}$$

□

Proof of Lemma 37

Proof. $\mathcal{J}(U : \mathbf{X}) = \mathcal{H}(U) - \mathcal{H}(U \mid \mathbf{X})$. If at least one child of U is conditioned on and transmits (i.e. in $V \in \mathbf{CH}(U) \cap \mathbf{X}$ and $v_{\mathbf{x}} \neq \phi$), then $\mathcal{H}(U \mid \mathbf{x}) = 0$. Otherwise, $\mathcal{H}(U \mid \mathbf{X}) = \mathcal{H}(U)$ because all $X \in \mathbf{X} \setminus \mathbf{CH}(U)$ with active paths to U must go through colliders in $\mathbf{CH}(U)$ — all of which are not in $\mathbf{X}$ or not transmitting. □

Proof of Lemma 38

Proof. $\mathcal{J}(M_i : Y \mid \mathbf{X}) = \sum_{\mathbf{X}} \mathcal{J}(M_i : Y \mid \mathbf{X})$ is zero unless (M_i, U_i) and (U_i, Y) both transmit. Furthermore, the DPI gives $\mathcal{J}(M_i : Y \mid \mathbf{X}) \leq \mathcal{H}(U_i \mid \mathbf{X})$, which is zero if any of the edges from $U_i \to X$ for $X \in \mathbf{X}$ transmit. □

Proof of Lemma 39

Proof. $\mathcal{J}(M_i : Y \mid \mathbf{X}) = \sum_{\mathbf{X}} \mathcal{J}(M_i : Y \mid \mathbf{X})$ is zero unless both (M_i, U_i) and (U_i, Y) transmit and at least one of (U_i, X) transmits for $X \in \mathbf{X}$, in which case $\mathcal{J}(M_i : Y \mid \mathbf{x}) = \mathcal{J}(M_i : Y \mid U_i)$. □

Proof of Lemma 40

Proof. ($\Rightarrow$) We will prove this with the contrapositive. If there is no shared parent between V_i and V_j, then all active paths must go through Y. However, because at least one of V_i, V_j is not connected to a cause, all paths between V_i and V_j cannot have a collider at Y (through two causes). This means conditioning on Y blocks the remaining paths.

($\Leftarrow$) If there is a shared parent U between V_i and V_j, then $V_i \leftarrow U \rightarrow V_j$ is an active path, d-connecting the two vertices. If V_i and V_j each have corresponding parents $U_i, U_j \in \mathbf{PA}(Y)$, then $V_i \leftarrow U_i \rightarrow Y \leftarrow U_j \rightarrow V_j$ is an active path conditioned on Y. □

Proof of Lemma 41

Proof. The proof follows from Lemma 40. Adjacent edges in $\mathfrak{G}_Y$ either indicate shared parents or that both vertices have a (potentially different) cause of Y as their parent.

If V_i and V_j share a parent, then $\mathbf{PA}(V_i) \subseteq \mathbf{U}^{\text{GOOD}}$ implies $\mathbf{U}^{\text{GOOD}} \cap \mathbf{PA}(V_j) \neq \emptyset$, so V_j has at least one "good" parent. The symmetric argument holds for $\mathbf{PA}(V_i) \subseteq \mathbf{U}^{\text{BAD}}$.

If V_i and V_j both have at least one causal parent, then we know $V_i, V_j \notin \mathbf{V}^{\text{BAD}}$. We know both vertices have at least one good U as a parent, so it trivially follows that both are either in $\mathbf{V}^{\text{GOOD}}$ or $\mathbf{V}^{\text{AMBIG}}$. □

Proof of Theorem 6

Proof. The only potential conditioned collider is Y, which is not allowed to be separable by partial faithfulness. Hence, partial faithfulness guarantees that we can construct $\mathfrak{G}_Y$ from conditional independence tests because d-connection implies dependence between V.

The requirements on $\mathbf{V}^*$ given by the theorem ensure that every $V \in \mathbf{V}$ has at least one label from an adjacency to a $V^* \in \mathbf{V}^*$ in $\mathfrak{G}_Y$.

The algorithm adds "good" labels to all vertices with a known good parent and "bad" labels to all vertices with a known bad parent. Therefore, all "ambigious" vertices are correctly labeled.

We now only need to guarantee that that the "good" and "bad" vertices are not ambiguous. If V were ambiguous, it would be connected to a U of the opposite label (i.e. a "good" vertex would be connected to a bad U). Such a U would have at least one $V^* \in \mathbf{V}^* \cap \mathbf{CH}(U)$ which would be adjacent to V and have given V the label of U, a contradiction. □

Proof of Lemma 42

Proof. $\mathcal{I}(M_i : Y \mid \mathbf{X}) = \sum_{\mathbf{x} \in \mathbf{X}} \Pr(\mathbf{x}) \mathcal{I}(M_i : Y \mid \mathbf{x})$. Now, we have

$$\mathcal{I}(M_i : Y \mid \mathbf{x}) = \mathcal{H}(M_i \mid \mathbf{x}) = \mathcal{H}(U_i \mid \mathbf{x})$$

if both (M_i, U_i) and (U_i, Y_i) edges transmit, which occurs with probability $\alpha_{M_i,U_i}\alpha_{U_i,Y}$. Pulling this coefficient outside of the sum gives $\mathcal{I}(M_i : Y \mid \mathbf{X}) = \alpha_{M_i,U_i}\alpha_{U_i,Y}\mathcal{H}(U_i \mid \mathbf{x})$.

□

Proof of Lemma 43

Proof. $U_i \in \mathbf{U}^{\text{BAD}}$ means $M_i \dashrightarrow U_i \dashleftarrow Y$, so $\mathcal{I}(M_i : Y) = 0$.

$$\begin{aligned} \mathcal{I}(M_i : Y \mid \mathbf{X}) &= -\mathcal{I}(M_i : Y : \mathbf{X}) \\ &\leq \mathcal{I}(M_i : \mathbf{X} \mid Y) \\ &\leq \mathcal{I}(U_i : \mathbf{X} \mid Y) = 0 \end{aligned} \tag{8.10}$$

The final inequality comes from the data processing inequality. □

Proof of Lemma 44

Proof. Consider the function $F_C(V_A) = F_C(G, B)) = F_{\mathrm{ISO}(V_G)}(G)$. By definition,

$$\mathcal{J}(F_{\mathrm{ISO}(V_G)}(G) : V_G \mid Y) = \mathcal{J}(G : V_G \mid Y) \tag{8.11}$$
$$= \mathcal{J}(V_A : V_G \mid Y). \tag{8.12}$$

Hence, F_C is in the feasible set of the optimization function defining isolation functions. Furthermore, $F_C(V_A)$ is only a function of G and $G \perp\!\!\!\perp U_B \mid Y$, so we can also conclude that $F_C(V_A) \perp\!\!\!\perp U_B$. This means

$$\begin{aligned}\mathcal{H}(F_C(V_A) \mid Y) &= \mathcal{H}(F_C(V_A) \mid U_B, Y) \\ &\leq \mathcal{H}(F_{\mathrm{ISO}(V_G)}(V_A) \mid U_B, Y) \\ &\leq \mathcal{H}(F_{\mathrm{ISO}(V_G)}(V_A) \mid Y) - \mathcal{J}(F_{\mathrm{ISO}(V_G)}(V_A) : U_B \mid Y).\end{aligned}$$

Hence, if $\mathcal{J}(F_{\mathrm{ISO}(V_G)}(V_A) : U_B \mid Y) > 0$, then $\mathcal{H}(F_C(V_A) \mid Y) < \mathcal{H}(F_{\mathrm{ISO}(V_G)}(V_A) \mid Y)$, contradicting the minimality of $F_{\mathrm{ISO}(V_G)}(V_A)$. □

Proof of Theorem 7

Proof. To shorten some equations, we will use

$$\mathbf{F}(V_A) := F_{\mathrm{ISO}(V_G)}(V_A \mid Y).$$

We first show that Equation 8.8 is sufficient for an improvement in relevance. We can expand the relevance of $\mathbf{X}^+$ as follows:

$$\begin{aligned}\mathcal{J}(Y : \mathbf{X}^+) =& \mathcal{J}(Y : \mathbf{F}(V_A)) + \mathcal{J}(Y : V_G \mid \mathbf{F}(V_A)) \\ &\geq \mathcal{J}(Y : V_G \mid \mathbf{F}(V_A)) \\ &\geq \mathcal{J}(Y : V_G) - \mathcal{J}(Y : V_G : \mathbf{F}(V_A)).\end{aligned} \tag{8.13}$$

So, for guaranteed improvement in relevance ($\mathcal{J}(Y : \mathbf{X}^+) > \mathcal{J}(Y : V_G)$), we need negative $\mathcal{J}(Y : V_G : \mathbf{F}(V_A)) < 0$. Expanding,

$$\mathcal{J}(Y : V_G : \mathbf{F}(V_A)) = \mathcal{J}(\mathbf{F}(V_A) : V_G) - \mathcal{J}(\mathbf{F}(V_A) : V_G \mid Y). \tag{8.14}$$

Thus, Equation 8.8 gives us the exact condition needed for negative interaction information, guaranteeing improvement.

We can show that the context sensitivity is no worse by separately considering the context sensitivity with $\mathbf{PA}(\mathbf{U}^{\text{BAD}})$ and $\mathbf{PA}(\mathbf{U}^{\text{GOOD}})$. We begin with $\mathbf{U}^{\text{BAD}}$. Applying Lemma 44,

$$\mathcal{J}((V_G, \mathbf{F}(V_A)) : \mathbf{U}^{\text{BAD}} \mid Y) = 0, \tag{8.15}$$

which satisfies the conditions for Lemma 43 to ensure us that $\mathcal{J}(\mathbf{PA}(\mathbf{U}^{\text{BAD}}) : Y \mid \mathbf{X}^+) = 0$.

Now, consider an arbitrary for $M_G = \mathbf{PA}(U_G) \in \mathbf{PA}(\mathbf{U}^{\text{GOOD}})$. Lemma 42 tells us that

$$\mathcal{J}(M_G : Y \mid \mathbf{X}^+) = \alpha_{M_G:U_G}\alpha_{U_G,Y}H(U_G \mid \widetilde{\mathbf{CH}}_{\mathbf{X}^+}(U_G)).$$

We then observe that $H(U_G \mid \widetilde{\mathbf{CH}}_{\mathbf{X}^+}(U_G)) \leq H(U_G \mid \mathbf{X})$ because entropy is submodular, which leads us to conclude

$$\mathcal{J}(M_G : Y \mid \mathbf{X}^+) \leq \mathcal{J}(M_G : Y \mid \mathbf{X}).$$

This completes the proof. □

8.10 Experimental Details

Synthetic experimental setup

M_G and M_B are drawn from normal distributions with mean 0 and variable standard deviations. All other vertices (other than Y) are the average of their parents plus additional Gaussian noise $N(0, .2)$. $T_A \in \mathbb{R}^2$ is generated by applying a rotation matrix to $(T_A^{(G)}, T_A^B)^T$[8]. Y indicates whether its parents sum to a positive number with a 5% probability of flipping randomly.

F1 scores for real world experiment

We give the F1 scores for the experiment described in Section 8.7 in Table 8.2.

Real world experiment in-domain performance

Here we provide the results of the in-domain accuracy for the experiment described in Sec. 8.7. Recall, that we use US Census data and consider distributions shifts across time as suggested by F. Ding et al., 2021. Table 8.3 shows the accuracy on 2019 data on a held-out dataset (separate from the training split). We repeated the experiment 10 times on different training/testing splits and report the mean and standard deviation of the accuracy for the largest states in the U.S. As expected, using all features has the most predictive power for in-domain tasks.

[8]Many rotations were tried in our experiments with identical results, so we display results from a 45 degree rotation.

Table 8.2: Comparison of out-of-domain (2021) performance on predicting high income via F1 scores.

State	All Features	Engineered Features	Limited Features
CA	**0.684**	**0.683**	0.676
FL	**0.459**	0.388	0.388
GA	0.541	**0.626**	**0.624**
IL	0.563	**0.630**	**0.628**
NY	**0.688**	**0.690**	0.662
NC	**0.475**	0.410	0.410
OH	0.519	**0.581**	**0.580**
PA	0.531	**0.608**	**0.606**
TX	0.554	**0.619**	**0.619**
avg	0.557	**0.582**	0.577

Table 8.3: Comparison of in-domain (2019) performance on predicting high income via Accuracies.

State	All Features	Engineered Features	Limited Features
CA	**0.713** ± 0.0010	**0.710** ± 0.0012	0.691 ± 0.0011
FL	**0.700** ± 0.0014	0.693 ± 0.0020	0.694 ± 0.0017
GA	**0.708** ± 0.0025	**0.708** ± 0.0036	**0.707** ± 0.0036
IL	**0.689** ± 0.0023	**0.690** ± 0.0039	**0.685** ± 0.0021
NY	**0.705** ± 0.0024	0.698 ± 0.0022	0.687 ± 0.0076
NC	**0.713** ± 0.0020	0.703 ± 0.0049	0.700 ± 0.0028
OH	**0.717** ± 0.0029	**0.716** ± 0.0042	**0.712** ± 0.0033
PA	**0.702** ± 0.0028	**0.701** ± 0.0027	0.695 ± 0.0026
TX	**0.708** ± 0.0019	**0.705** ± 0.0025	**0.706** ± 0.0022
avg	**0.706**	0.703	0.697

Part IV

Discussion

Chapter 9

RETHINKING DIMENSIONALITY AND ERRORS

Through a detailed discussion of Level 3 and 4 knowledge, this thesis has introduced a number of important ideas on which future work can be built.

9.1 Diverse Data

One recurring lesson is the nuanced problems associated with data from multiple sources, contexts, and populations. While the many-source conundrum can be addressed using the LLC and mixture models, significant additional work is required to handle the analogous settings with continuous variables.

Further study is also needed to understand the effects of sampling bias on subsets of variables. A natural complement to the universal confounder is a universal "sink" which applies different levels of sampling bias to the entire system. With an analogous limit to the power of this sampling bias, we can study the identifiability of the relationships and causal structure.

9.2 The Necessity (and Blessing?) of Dimensionality

High dimensional models often overfit, leading to what is known as the "curse of dimensionality." This thesis presents two algorithms for a previously unsolved setting of causal identification and causal discovery. The main requirements for these algorithms are a minimum span of confounding influence relative to the sparsity of the graph — i.e. the unobserved U must point to enough variables to be identifiable. This implicitly *requires* a high dimensional setting.

The necessity of dimensionality may go further. Response functions which are indexed by latent variables, such as $\Gamma_u(X)$, are generally unidentifiable because we cannot draw multiple samples from the same response function without resampling u. This information deficiency parallels the need for "exchangable" samples for Level 3 knowledge.

To navigate heterogeneous data, our approach in Chapter 4 utilizes the observation that each additional observed variable contains new information about the heterogeneity. Concretely, this corresponds to creating "synthetic copies" of variables, which use linear combinations of new vertices to mimic the statistical properties of a disjoint

vertex. These synthetic copies allow us to simulate multiple samplings from the same distribution with constant U. The details of this "method of synthetic bits" are given in S. Gordon, B. H. Mazaheri, et al., 2021.

The need for exponential sample complexity in the k-MixProd algorithm partially derives from restricting ourselves to a linear number of independent vertices. However, with a quadratic number of independent vertices it is possible to form a quadratic number of synthetic copies of each vertex. At this threshold, we expect to gain enough information to begin to label each data-point with a specific U, significantly simplifying the problem to the identification of a categorical distribution on U. As a result, it is possible that access to higher-dimensional data contains the cure to the (often prohibitive) data demands of deconfounding and causal inference.

9.3 The Information Theoretical Value of Errors

Causal structures represent conditional independence properties. These structures fail when these independence properties are violated, either due to an incorrect model or the presence of unobserved confounding. As we saw in Chapters 4 and 5, the breakdown of these properties provides key information that is used to learn the effects of confounding.

A similar phenomenon is observed at Level 4 in Chapter 8, where pairing a variable with its (potentially incorrect) prediction can help isolate information from variables along active paths, thereby enhancing stability to distribution shift. Again, we see that *errors* play a key roll in augmenting knowledge.

In an field that has placed an increasing emphasis on incremental improvements to accuracy, it can be easy to forget that science is built on the *breakdown* of models. This thesis treats statistical models more broadly than the naïve tasks for which they have been trained by harnessing their failures. Following down this path may yield exciting new tools to ascend the hierarchy of knowledge.

BIBLIOGRAPHY

Acid, Silvia and Luis M De Campos (1996). "An algorithm for finding minimum d-separating sets in belief networks". In: *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pp. 3–10.

Agrawal, Raj et al. (July 2023). "The DeCAMFounder: nonlinear causal discovery in the presence of hidden variables". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology*, qkad071. ISSN: 1369-7412. DOI: 10.1093/jrsssb/qkad071. eprint: https://academic.oup.com/jrsssb/advance-article-pdf/doi/10.1093/jrsssb/qkad071/50874769/qkad071.pdf. URL: https://doi.org/10.1093/jrsssb/qkad071.

Allman, E. S., C. Matias, and J. A. Rhodes (2009). "Identifiability of parameters in latent structure models with many observed variables". In: *Ann. Statist.* 37.6A, pp. 3099–3132. DOI: 10.1214/09-AOS689.

Alon, Noga (2002). "Voting paradoxes and digraphs realizations". In: *Advances in Applied Mathematics* 29.1, pp. 126–135.

Anandkumar, A., D. Hsu, et al. (2012). "Learning high-dimensional mixtures of graphical models". In: *arXiv preprint arXiv:1203.0697*.

Anandkumar, A., D. J. Hsu, and S. M. Kakade (2012). "A Method of Moments for Mixture Models and Hidden Markov Models". In: *Proc. 25th Ann. Conf. on Learning Theory - COLT*. Vol. 23. JMLR Proceedings, pp. 33.1–33.34. URL: http://proceedings.mlr.press/v23/anandkumar12/anandkumar12.pdf.

Angrist, Joshua and Guido Imbens (1995). *Identification and estimation of local average treatment effects*.

Angrist, Joshua D and Alan B Krueger (2001). "Instrumental variables and the search for identification: From supply and demand to natural experiments". In: *Journal of Economic perspectives* 15.4, pp. 69–85.

Arjovsky, Martin et al. (2019). "Invariant risk minimization". In: *arXiv preprint arXiv:1907.02893*.

Banerjee, Pradeep Kr et al. (2018). "Unique informations and deficiencies". In: *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, pp. 32–38.

Bareinboim, Elias and Jin Tian (2015). "Recovering causal effects from selection bias". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1.

Batu, T., S. Guha, and S. Kannan (2004). "Inferring Mixtures of Markov Chains". In: *Proc. 17th Conf. on Learning Theory*, pp. 186–199. DOI: 10.1007/978-3-540-27819-1_13.

Bellot, Alexis and Mihaela van der Schaar (2020). "Generalization and invariances in the presence of unobserved confounding". In: *arXiv preprint arXiv:2007.10653* 4.

Bertschinger, Nils et al. (2014). "Quantifying unique information". In: *Entropy* 16.4, pp. 2161–2183.

Cai, Ruichu et al. (2023). "Causal Discovery with Latent Confounders Based on Higher-Order Cumulants". In: *arXiv preprint arXiv:2305.19582*.

Castanedo, Federico (2013). "A review of data fusion techniques". In: *The scientific world journal* 2013.

Chaudhuri, K. and S. Rao (2008). "Learning Mixtures of Product Distributions Using Correlations and Independence". In: *Proc. 21st Ann. Conf. on Learning Theory - COLT*. Omnipress, pp. 9–20. URL: http://colt2008.cs.helsinki.fi/papers/7-Chaudhuri.pdf.

Chen, S. and A. Moitra (2019). "Beyond the low-degree algorithm: mixtures of subcubes and their applications". In: *Proc. 51st Ann. ACM Symp. on Theory of Computing*, pp. 869–880. DOI: 10.1145/3313276.3316375.

Chiappa, Silvia (2019). "Path-specific counterfactual fairness". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01, pp. 7801–7808.

Chikritzhs, Tanya, Kaye Fillmore, and Tim Stockwell (2009). "A healthy dose of scepticism: four good reasons to think again about protective effects of alcohol on coronary heart disease". In: *Drug and alcohol review* 28.4, pp. 441–444.

Cohen, Michael and Jean-Claude Falmagne (1990). "Random utility representation of binary choice probabilities: a new class of necessary conditions". In: *Journal of Mathematical Psychology* 34.1, pp. 88–94.

Cormen, Thomas H et al. (2022). *Introduction to algorithms*. MIT press.

Correa, Juan D, Jin Tian, and Elias Bareinboim (2019). "Identification of causal effects in the presence of selection bias". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 2744–2751.

Cover, Thomas M (1999). *Elements of information theory*. John Wiley & Sons.

Croce, Francesco and Matthias Hein (2020). "Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-Free Attacks". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML.

Cryan, M., L. Goldberg, and P. Goldberg (2001). "Evolutionary trees can be learned in polynomial time in the two state general Markov model". In: *SIAM J. Comput.* 31.2, pp. 375–397. DOI: 10.1137/S0097539798342496.

Ding, Frances et al. (2021). "Retiring Adult: New Datasets for Fair Machine Learning". In: *Advances in Neural Information Processing Systems* 34.

Ding, Jingqiu et al. (2022). "Fast algorithm for overcomplete order-3 tensor decomposition". In: *Conference on Learning Theory*. PMLR, pp. 3741–3799.

Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository*. URL: http://archive.ics.uci.edu/ml.

Dutta, Sanghamitra and Faisal Hamman (2023). "A Review of Partial Information Decomposition in Algorithmic Fairness and Explainability". In: *Entropy* 25.5, p. 795.

Dutta, Sanghamitra, Praveen Venkatesh, et al. (2020). "An information-theoretic quantification of discrimination with exempt features". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 3825–3833.

Everitt, B. S. and D. J. Hand (1981). "Mixtures of discrete distributions". In: *Finite Mixture Distributions*. Dordrecht: Springer Netherlands, pp. 89–105.

Feldman, J., R. O'Donnell, and R. A. Servedio (2008). "Learning Mixtures of Product Distributions over Discrete Domains". In: *SIAM J. Comput.* 37.5, pp. 1536–1564. DOI: 10.1137/060670705.

Feldman, Jon, Ryan O'Donnell, and Rocco A Servedio (2008). "Learning mixtures of product distributions over discrete domains". In: *SIAM Journal on Computing* 37.5, pp. 1536–1564.

Fishburn, Peter (1996). "Acyclic sets of linear orders". In: *Social choice and Welfare* 14.1, pp. 113–124.

Fishburn, Peter C (1992). "Induced binary probabilities and the linear ordering polytope: A status report". In: *Mathematical Social Sciences* 23.1, pp. 67–80.

Fisher, Ronald Aylmer and John Wishart (1930). "The arrangement of field experiments and the statistical reduction of the results". In.

Freund, Y. and Y. Mansour (July 1999). "Estimating a mixture of two product distributions". In: *Proc. 12th Ann. Conf. on Computational Learning Theory*, pp. 53–62. DOI: 10.1145/307400.307412.

Frot, Benjamin, Preetam Nandy, and Marloes H Maathuis (2019). "Robust causal structure learning with some hidden variables". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 81.3, pp. 459–487.

Gilboa, Itzhak (1990). "A necessary but insufficient condition for the stochastic binary choice problem". In: *Journal of Mathematical Psychology* 34.4, pp. 371–392.

Glymour, C., K. Zhang, and P. Spirtes (2019). "Review of Causal Discovery Methods Based on Graphical Models". In: *Frontiers in Genetics* 10, p. 524. DOI: 10.3389/fgene.2019.00524.

Gordon, S. L. and L. J. Schulman (2022). "Hadamard extensions and the identification of mixtures of product distributions". In: *IEEE Transactions on Information Theory*. to appear.

Gordon, Spencer, Bijan Mazaheri, Yuval Rabani, et al. (2023). "Causal inference despite limited global confounding via mixture models". In: *2nd Conference on Causal Learning and Reasoning*.

Gordon, Spencer, Bijan Mazaheri, Leonard J Schulman, et al. (2020). "The sparse Hausdorff moment problem, with application to topic models". In: *arXiv preprint arXiv:2007.08101*.

Gordon, Spencer, Bijan H Mazaheri, et al. (2021). "Source identification for mixtures of product distributions". In: *Conference on Learning Theory*. PMLR, pp. 2193–2216.

Gordon, Spencer L. et al. (2023). *Identification of Mixtures of Discrete Product Distributions in Near-Optimal Sample and Time Complexity*. arXiv: 2309.13993 [cs.LG].

Grodstein, Francine, Joann E Manson, and Meir J Stampfer (2006). "Hormone therapy and coronary heart disease: the role of time since menopause and age at hormone initiation". In: *Journal of Women's Health* 15.1, pp. 35–44.

Grötschel, Martin, Michael Jünger, and Gerhard Reinelt (1985). "Facets of the linear ordering polytope". In: *Mathematical programming* 33.1, pp. 43–60.

Grünbaum, Branko et al. (1967). *Convex polytopes*. Vol. 16. Springer.

Gulrajani, Ishaan and David Lopez-Paz (2021). "In Search of Lost Domain Generalization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=lQdXeXDoWtI.

Gupta, R., R. Kumar, and S. Vassilvitskii (2016). "On Mixtures of Markov Chains". In: *Advances in Neural Information Processing Systems*. Vol. 29. URL: https://proceedings.neurips.cc/paper/2016/file/8b5700012be65c9da25f49408d959ca0-Paper.pdf.

Gurushankar, Keerthana, Praveen Venkatesh, and Pulkit Grover (2022). "Extracting unique information through markov relations". In: *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, pp. 1–6.

Hacking, Ian (2001). *An introduction to probability and inductive logic*. Cambridge university press.

Heckerman, D. (2018). "Accounting for hidden common causes when inferring cause and effect from observational data". (NIPS 2017 causal inference workshop). URL: https://arxiv.org/abs/1801.00727.

Heinze-Deml, Christina, Jonas Peters, and Nicolai Meinshausen (2018). "Invariant causal prediction for nonlinear models". In: *Journal of Causal Inference* 6.2.

Hernán, Miguel A, Alvaro Alonso, et al. (2008). "Observational studies analyzed like randomized experiments: an application to postmenopausal hormone therapy and coronary heart disease". In: *Epidemiology* 19.6, pp. 766–779.

Hernán, Miguel A, David Clayton, and Niels Keiding (Mar. 2011). "The Simpson's paradox unraveled". In: *International Journal of Epidemiology* 40.3, pp. 780–785. ISSN: 0300-5771. DOI: 10.1093/ije/dyr041. eprint: https://academic.oup.com/ije/article-pdf/40/3/780/18481353/dyr041.pdf. URL: https://doi.org/10.1093/ije/dyr041.

Hsu, D., S. M. Kakade, and T. Zhang (Sept. 2012). "A Spectral Algorithm for Learning Hidden Markov Models". In: *J. Comput. Syst. Sci.* 78.5, pp. 1460–1480. DOI: 10.1016/j.jcss.2011.12.025.

Imbens, G. W. and D. B. Rubin (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Jain, Siddharth et al. (2019). "Short tandem repeats information in tcga is statistically biased by amplification". In: *BioRxiv*, p. 518878.

– (2021). "Glioblastoma signature in the DNA of blood-derived cells". In: *Plos one* 16.9, e0256831.

Kearns, M. et al. (1994). "On the learnability of discrete distributions". In: *Proc. 26th Ann. ACM Symp. on Theory of Computing*, pp. 273–282. DOI: 10.1145/195058.195155.

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Kivva, Bohdan et al. (2021). "Learning latent causal graphs via mixture oracles". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer et al. URL: https://openreview.net/forum?id=f9mSLa07Ncc.

KPFRS, LIII (1901). "On lines and planes of closest fit to systems of points in space". In: *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (SIGMOD)*, p. 19.

Krueger, David et al. (2021). "Out-of-distribution generalization via risk extrapolation (rex)". In: *International Conference on Machine Learning*. PMLR, pp. 5815–5826.

Kruskal, Joseph B (1977). "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics". In: *Linear algebra and its applications* 18.2, pp. 95–138.

Kumar, A. and G. Sinha (July 2021). "Disentangling mixtures of unknown causal interventions". In: *Proc. Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*. Ed. by C. de Campos and M. H. Maathuis. Vol. 161. Proc. Machine Learning Research. PMLR, pp. 2093–2102. URL: https://proceedings.mlr.press/v161/kumar21a.html.

Kuroki, M. and J. Pearl (2014). "Measurement bias and effect restoration in causal inference". In: *Biometrika* 101.2, pp. 423–437. DOI: 10.1093/biomet/ast066.

Lauritzen, Steffen L et al. (1990). "Independence properties of directed Markov fields". In: *Networks* 20.5, pp. 491–505.

Leiner, James et al. (2022). “Data fission: splitting a single data point”. In: *arXiv preprint arXiv:2112.11079*.

Li, J. et al. (2015). “Learning Arbitrary Statistical Mixtures of Discrete Distributions”. In: *Proc. 47th Ann. ACM Symp. on Theory of Computing*, pp. 743–752. DOI: 10.1145/2746539.2746584.

Lindsay, B. G. (1995). *Mixture models: theory, geometry and applications*, pp. i–163.

Magliacane, Sara et al. (2018). “Domain Adaptation by Using Causal Inference to Predict Invariant Conditional Distributions”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS’18. Montréal, Canada: Curran Associates Inc., pp. 10869–10879.

Mazaheri, Bijan, Siddharth Jain, and Jehoshua Bruck (2021). “Synthesizing New Expertise via Collaboration”. In: *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2447–2452. DOI: 10.1109/ISIT45174.2021.9517822.

Mazaheri, Bijan, Siddharth Jain, Matthew Cook, et al. (2023). *Omitted Labels in Causality: A Study of Paradoxes*. arXiv: 2311.06840 [cs.LG].

Mazaheri, Bijan, Atalanti Mastakouri, et al. (July 2023). “Causal information splitting: Engineering proxy features for robustness to distribution shifts”. In: *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. Ed. by Robin J. Evans and Ilya Shpitser. Vol. 216. Proceedings of Machine Learning Research. PMLR, pp. 1401–1411. URL: https://proceedings.mlr.press/v216/mazaheri23a.html.

McGarvey, David C (1953). “A theorem on the construction of voting paradoxes”. In: *Econometrica: Journal of the Econometric Society*, pp. 608–610.

McLachlan, G. J., S. X. Lee, and S. I. Rathnayake (2019). “Finite Mixture Models”. In: *Annual Review of Statistics and Its Application* 6.1, pp. 355–378. DOI: 10.1146/annurev-statistics-031017-100325.

Miao, W., Z. Geng, and E. T. Tchetgen (2018). “Identifying Causal Effects With Proxy Variables of an Unmeasured Confounder”. In: *Biometrika* 105.4, pp. 987–993. DOI: 10.1093/biomet/asy038.

Michels, Karin B and JoAnn E Manson (2003). *Postmenopausal hormone therapy: a reversal of fortune*.

Moitra, Ankur (2016). “An almost optimal algorithm for computing nonnegative rank”. In: *SIAM Journal on Computing* 45.1, pp. 156–173.

Monjardet, Bernard (2006). “Condorcet domains and distributive lattices”. In.

Muandet, Krikamol, David Balduzzi, and Bernhard Schölkopf (2013). “Domain generalization via invariant feature representation”. In: *International conference on machine learning*. PMLR, pp. 10–18.

Naimi, Timothy S et al. (2005). “Cardiovascular risk factors and confounders among nondrinking and moderate-drinking US adults”. In: 28.4, pp. 369–373.

Newcomb, S. (1886). “A Generalized Theory of the Combination of Observations so as to Obtain the Best Result”. In: *American Journal of Mathematics* 8.4, pp. 343–366.

Nicolas, Jean Antoine et al. (1785). *Essai sur l’application de l’analyse à la probabilité des decisions rendues à la pluralité des voix. Par m. le marquis de Condorcet,...* de l’Imprimerie Royale.

Oberst, Michael et al. (2021). “Regularizing towards causal invariance: Linear models with proxies”. In: *International Conference on Machine Learning*. PMLR, pp. 8260–8270.

Ogburn, Elizabeth L, Ilya Shpitser, and Eric J Tchetgen Tchetgen (2019). “Comment on “blessings of multiple causes””. In: *Journal of the American Statistical Association* 114.528, pp. 1611–1615.

Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Pearl, Judea (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann.

– (2009). *Causality*. Cambridge university press.

– (2022). “Comment: understanding Simpson’s paradox”. In: *Probabilistic and causal inference: The works of judea Pearl*, pp. 399–412.

Pearl, Judea and Elias Bareinboim (2011). “Transportability of causal and statistical relations: A formal approach”. In: *Twenty-fifth AAAI conference on artificial intelligence*.

Pearson, K. (1894). “Contributions to the mathematical theory of evolution III”. In: *Philosophical Transactions of the Royal Society of London (A.)* 185, pp. 71–110.

Peters, Jonas, Peter Bühlmann, and Nicolai Meinshausen (2016a). “Causal inference by using invariant prediction: identification and confidence intervals”. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012.

– (2016b). “Causal inference by using invariant prediction: identification and confidence intervals”. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 78.5, pp. 947–1012. ISSN: 13697412, 14679868. URL: http://www.jstor.org/stable/44682904 (visited on 02/13/2023).

Peters, Jonas, Dominik Janzing, and Bernhard Schölkopf (2017). *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

Peters, Jonas, Joris M Mooij, et al. (2014). “Causal discovery with continuous additive noise models”. In.

Quinonero-Candela, Joaquin et al. (2008). *Dataset shift in machine learning*. Mit Press.

Quinzan, Francesco et al. (2022). *Learning Counterfactually Invariant Predictors*. DOI: 10.48550/ARXIV.2207.09768. URL: https://arxiv.org/abs/2207.09768.

Rabani, Y., L. J. Schulman, and C. Swamy (2014). "Learning mixtures of arbitrary distributions over large discrete domains". In: *Proc. 5th Conf. on Innovations in Theoretical Computer Science*, pp. 207–224. DOI: 10.1145/2554797.2554818.

Ranganath, R. and A. Perotte (2018). "Multiple causal inference with latent confounding". URL: https://arxiv.org/abs/1805.08273.

Ratsimalahelo, Zaka (2001). *Rank test based on matrix perturbation theory*. Tech. rep. EERI Research Paper Series.

Reinelt, Gerhard (1985). *The linear ordering problem: algorithms and applications*. Vol. 8. Heldermann.

Richardson, Thomas and Peter Spirtes (2002). "Ancestral graph Markov models". In: *The Annals of Statistics* 30.4, pp. 962–1030.

Rojas-Carulla, Mateo et al. (2018). "Invariant models for causal transfer learning". In: *The Journal of Machine Learning Research* 19.1, pp. 1309–1342.

Rothenhäusler, Dominik et al. (2021). "Anchor regression: Heterogeneous data meet causality". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 83.2, pp. 215–246.

Saari, Donald G (2000). "Mathematical structure of voting paradoxes". In: *Economic Theory* 15.1, pp. 1–53.

– (2009). "Condorcet domains: A geometric perspective". In: *The Mathematics of Preference, Choice and Order*. Springer, pp. 161–182.

Saeed, Basil, Snigdha Panigrahi, and Caroline Uhler (2020). "Causal structure discovery from distributions arising from mixtures of dags". In: *International Conference on Machine Learning*. PMLR, pp. 8336–8345.

Sagawa, Shiori et al. (2019). "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization". In: *arXiv preprint arXiv:1911.08731*.

Savage Jr, Richard P (1994). "The paradox of nontransitive dice". In: *The American Mathematical Monthly* 101.5, pp. 429–436.

Schlesselman, James J (1982). *Case-control studies: design, conduct, analysis*. Vol. 2. Oxford university press.

Schmidhuber, Juergen (2022). "Annotated History of Modern AI and Deep Learning". In: *arXiv preprint arXiv:2212.11279*.

Schölkopf, Bernhard et al. (2012). "On causal and anticausal learning". In: *arXiv preprint arXiv:1206.6471*.

Schweikert, Gabriele et al. (2008). "An empirical analysis of domain adaptation algorithms for genomic sequence analysis". In: *Advances in neural information processing systems* 21.

Sharan, V. et al. (2017). "Learning Overcomplete HMMs". In: *Advances in Neural Information Processing Systems*, pp. 940–949. URL: https://arxiv.org/abs/1711.02309.

Shimodaira, Hidetoshi (2000). "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of statistical planning and inference* 90.2, pp. 227–244.

Sinha, Aman, Hongseok Namkoong, and John Duchi (2018). "Certifiable Distributional Robustness with Principled Adversarial Training". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Hk6kPgZA-.

Spirtes, P. et al. (2000). "Constructing Bayesian network models of gene expression networks from microarray data".

Spirtes, Peter (2001). "An anytime algorithm for causal inference". In: *International Workshop on Artificial Intelligence and Statistics*. PMLR, pp. 278–285.

Spirtes, Peter, Clark Glymour, Richard Scheines, and David Heckerman (2000). *Causation, prediction, and search*. MIT press.

Spirtes, Peter, Clark Glymour, Richard Scheines, Peter Spirtes, et al. (1993). "Discovery algorithms for causally sufficient structures". In: *Causation, prediction, and search*, pp. 103–162.

Squires, Chandler and Caroline Uhler (2022). "Causal structure learning: A combinatorial perspective". In: *Foundations of Computational Mathematics*, pp. 1–35.

– (2023). "Causal structure learning: A combinatorial perspective". In: *Foundations of Computational Mathematics* 23.5, pp. 1781–1815.

Storkey, Amos et al. (2009). "When training and test sets are different: characterizing learning transfer". In: *Dataset shift in machine learning* 30, pp. 3–28.

Subbaswamy, Adarsh and Suchi Saria (2018). "Counterfactual Normalization: Proactively Addressing Dataset Shift Using Causal Mechanisms." In: *UAI*, pp. 947–957.

Sugiyama, Masashi (2015). *Introduction to statistical machine learning*. Morgan Kaufmann.

Suissa, Samy (2008). "Immortal time bias in pharmacoepidemiology". In: *American journal of epidemiology* 167.4, pp. 492–499.

Tahmasebi, B., S. A. Motahari, and M. A. Maddah-Ali (2018). "On the Identifiability of Finite Mixtures of Finite Product Measures". (Also in "On the identifiability of parameters in the population stratification problem: A worst-case analysis," Proc. ISIT'18 pp. 1051-1055). URL: https://arxiv.org/abs/1807.05444.

Thiesson, B. et al. (1998). "Learning Mixtures of DAG Models". In: *Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, pp. 504–513. DOI: 10.5555/2074094.2074154.

Titterington, D. M., A. F. M. Smith, and U. E. Makov (1985). *Statistical Analysis of Finite Mixture Distributions*. John Wiley and Sons, Inc.

Tomczak, Katarzyna, Patrycja Czerwińska, and Maciej Wiznerowicz (2015). "Review The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge". In: *Contemporary Oncology/Współczesna Onkologia* 2015.1, pp. 68–77.

Uhler, Caroline et al. (2013). "Geometry of the faithfulness assumption in causal inference". In: *The Annals of Statistics*, pp. 436–463.

Veitch, Victor et al. (2021). "Counterfactual Invariance to Spurious Correlations in Text Classification". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer et al. URL: https://openreview.net/forum?id=BdKxQp0iBi8.

Venkatesh, Praveen and Gabriel Schamberg (2022). "Partial information decomposition via deficiency for multivariate gaussians". In: *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 2892–2897.

Wang, Y. and D. M. Blei (2019). "The Blessings of Multiple Causes". In: *Journal of the American Statistical Association* 114.528, pp. 1574–1596. DOI: 10.1080/01621459.2019.1686987.

Weiss, Karl, Taghi M Khoshgoftaar, and DingDing Wang (2016). "A survey of transfer learning". In: *Journal of Big data* 3.1, pp. 1–40.

Williams, Paul L and Randall D Beer (2010). "Nonnegative decomposition of multivariate information". In: *arXiv preprint arXiv:1004.2515*.

Zech, John R. et al. (Nov. 2018). "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study". In: *PLOS Medicine* 15.11, pp. 1–17. DOI: 10.1371/journal.pmed.1002683. URL: https://doi.org/10.1371/journal.pmed.1002683.