universität wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

„Measurement Dependence Inducing Latent Causal Models“

verfasst von / submitted by

Alex Markham

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Doktor der technischen Wissenschaften (Dr. techn.)

Wien, 2021 / Vienna, 2021

Studienkennzahl laut Studienblatt / degree programme code as it appears on the student record sheet:	UA 786 880
Dissertationsgebiet laut Studienblatt / degree programme as it appears on the student record sheet:	Informatik
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. -Ing. Moritz Grosse-Wentrup

Abstract

In this dissertation, we reconsider some of the common assumptions used in causal inference methods, ranging from foundational ideas, such as the notion of a data set as being sampled from a single causal model instead of being from a complex causal system, to more method-specific assumptions, such as linearity of causal relationships or (non)Gaussianity of causal variables. In doing so, we make several contributions to the field of causal inference: we define a new class of causal model; we derive new causal inference methods using these models; and we explore both the aspects of these models that are captured by other existing models as well as the ways in which these models and their equivalence classes subsume other existing models. The mathematical thread connecting our different contributions is the use of undirected graphs, which facilitates novel graph theoretic, algebraic, geometric, and statistical perspectives on causal methods. Though our main focus is theoretical, we also provide a causal inference software package and demonstrate our methods with various real-world applications.

Kurzfassung

In dieser Dissertation überdenken wir einige der gängigen Annahmen, die in Methoden der kausalen Inferenz verwendet werden. Dies reicht von grundlegenden Ideen, wie z.B. der Vorstellung, dass ein Datensatz aus einem einzelnen Kausalmodell statt aus einem komplexen Kausalsystem entnommen wird, bis hin zu eher methodenspezifischen Annahmen, wie z.B. der Linearität von Kausalbeziehungen oder der (Nicht-)Gauß'schen Verteilung von kausalen Variablen. Dabei leisten wir mehrere Beiträge zum Gebiet der kausalen Inferenz: wir definieren eine neue Klasse von Kausalmodellen; wir leiten neue kausale Inferenzmethoden unter Verwendung dieser Modelle ab und wir untersuchen sowohl die Aspekte dieser Modelle, die von anderen existierenden Modellen erfasst werden, als auch die Art und Weise, in der diese Modelle und ihre Äquivalenzklassen andere existierende Modelle subsumieren. Der mathematische Faden, der unsere verschiedenen Beiträge verbindet, ist die Verwendung ungerichteter Graphen, die neue graphentheoretische, algebraische, geometrische und statistische Perspektiven auf kausale Methoden ermöglicht. Obwohl unser Hauptaugenmerk auf der Theorie liegt, stellen wir auch ein Softwarepaket zur kausalen Inferenz zur Verfügung und demonstrieren unsere Methoden mit verschiedenen realen Anwendungen.

Contents

Publication List

This cumulative doctoral dissertation is a compilation and superordinate analysis of the following works, each of which has been published or is currently under review at peer-revewied conferences:

- Markham, A. and Grosse-Wentrup, M. (2020). Measurement dependence inducing latent causal models. In Peters, J. and Sontag, D., editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 590–599. PMLR

- Markham, A., Chivukula, A., and Grosse-Wentrup, M. (2020). `MeDIL`: A python package for causal modelling. In Jaeger, M. and Nielsen, T. D., editors, *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, pages 621–624. PMLR

- Markham, A. and Grosse-Wentrup, M. (2021). A Distance Covariance-based Kernel for Nonlinear Causal Clustering in Heterogeneous Populations. In *under review but available as arXiv e-print*, page arXiv:2106.03480

1. Preamble

Most generally, the aim of this dissertation is to develop new causal inference methods that complement existing methods. Thus, we hope to contribute to the robustness and variety of available causal inference methods, facilitating different applied scientists having more flexibility in choosing methods that are better suited for their respective applications.

We do this by critically revisiting the foundations and some common assumptions of causal inference. In particular we consider the three-part research question:

Q.1 To what extent can we learn a model of a causal system *while observing only some parts of that system,*

Q.2 especially when the population from which the model is being learned is *the result of not one but multiple causal systems,*

Q.3 and *without assuming a particular distribution*, such as a Gaussian one.

Answering these questions and providing algorithms for causal inference in these scenarios will be helpful for applications in a variety of fields, such as genetics and cognitive science. For example, Q.1 can arise from a typical brain-computer interface experiment: these result in behavioral (e.g., hand movement) and neuroimaging (e.g., electroencephalography, measuring electrical activity of the brain) data that, due to physical constraints of measurement devices compared to the complexity of the brain and nervous system, leaves parts of the cognitive and behavioral system unobserved (Tan and Nijholt, 2010). Q.2 can also arise in such a setting, because though different brains share general the same general organization, they are nevertheless highly individualized (Naselaris et al., 2021). Furthermore, it is hypothesized that neuronal populations can encode a multitude of different probability distributions (Ma et al., 2006), hence Q.3. However, instead of focusing on particular applications like the preceding example, this dissertation focuses on the theoretical and computational aspects of these questions.

Before addressing these questions in more detail, some review is helpful. The rest of this chapter is organized as follows: Section 1.1 is devoted to summarizing the background knowledge needed to properly contextualize and understand the preceding research questions—we do this by briefly tracing the history and development of the scientific understanding of causality in Section 1.1, arriving at the precise mathematical account that is currently so popular.This allows us to revisit our research questions in more detail in Section 1.2, discussing them in the context of existing work. Chapter 2 consists of three papers that form the basis of this cumulative dissertation. Each of these papers (one per section) addresses a concrete part of the preceding research questions and makes a novel theoretical or practical contribution to the field of causal inference: Section 2.1 introduces a new class of latent causal models (i.e., models partially observed systems) and an algorithm for learning their causal structure without assuming any specific distribution; Section 2.2 presents a software package implementing these as well as other causal inference

algorithms and demonstrates how to use the package; and Section 2.3 derives a kernel that can be used as a statistically consistent test of whether samples come from different causal structures and that connects our new class of causal models to more general machine learning research in clustering and kernel methods.

In Chapter 3, we summarize the main contributions of the preceding papers, before providing a superordinate analysis of them in Section 3.2, detailing precisely how they are linked and how they work together to address the preceding research questions: namely, they are connected mathematically by their novel use of undirected graphs for causal inference. Finally, we conclude in Section 3.3 with some intuitions about promising directions for future work, most notably by exploring connections to recent work in the field of algebraic statistics.

1.1. Causality Background

Beyond the history itself being presented, the main point of this section is that the notion of causality used today in the field of causal inference (and in the rest of this dissertation) has a precise mathematical definition along with the corresponding limitations that this entails. Though this definition has been developed to capture some common intuitions and shortcomings of other definitions, it nevertheless does not entirely subsume all other understandings of causality and should not be seen as the panacea for all problems encountered in the study of artificial intelligence or science more generally.

1.1.1. Ancient causality

The concept of *causality* has an uncapturably vast scholarly history, featuring prominently in some of the oldest surviving texts of ancient philosophical and scientific thought. Its history can be traced[1] back to Aristotle, e.g., in Book II Part 3 of *Physics*, dating back to the 4th century BCE. Aristotle's account of causality is noteworthy for being one of the earliest and most comprehensive, and it was likely known to, if not a direct influence on, many of the (western) philosophers and scientists who have since studied causality. To summarize very briefly, Aristotle posited four kinds of cause: the *efficient* cause is most similar to the notion used today in causal inference (and natural language) and referred to the agent or object responsible for the effect, while the *material*, *formal*, and *final* causes were broader and included notions ranging from physical composition of objects to the intentions or goals of actions. While the details of Aristotle's theory of causation, lacking modern mathematical formalism, bear little resemblance to theories of causation used in machine learning today, his writings nevertheless set the precedent of equating scientific explanations with causal explanations, providing an authoritative source for later scholars emphasizing the importance of causal thinking in scientific inquiry.

Though often neglected in discussions about the history of causality and philosophy of science, ancient Indian philosophy contains many interesting developments. Causality was one of the central topics both in the more religious works of the Vedic Period (as far back as the 8th century BCE) as well as the natural philosophy of the Cārvāka, Ājīvika, and subsequent schools, dating back to the 6th century BCE. These are noteworthy

[1] For example, see Pearl (2009, Chapter 5), Pearl and Mackenzie (2018, Chapter 1), and especially Hulswit (2004).

for being earlier than Aristotle and having an analogous influence on later scholars in Eastern philosophy more generally. Despite their age, some of the ideas are surprisingly contemporary, for example: whereas Aristotle's understanding of causality (like that of some Vedic philosophers) placed special importance on agents and intention, philosophers in the Cārvāka school thought of the world as fundamentally material and therefore thought of causality as one of the ways material objects interact through their own inherent properties without any need of agents or intentions—see Sarvepalli Radhakrishnan (1957, Chapter VII) and Kalupahana and Deutsch (1975, Chapter II) for more details and examples.

For the purposes of this dissertation, these ancient accounts of causality serve to illustrate that though there are common threads connecting causal inference today with ideas about causality from over two millennia ago, there are nevertheless different possible notions of causality that should not be conflated and that are not necessarily captured by current definitions.

1.1.2. Granger causality

One notion of causality used today, especially in time series analysis in fields like econometrics, is that introduced by Granger (1969). Granger causality has two defining aspects: (i) predictiveness and (ii) temporal precedence (Granger, 1980). Informally, this means we say A causes B if (i) knowing A allows us to better predict B and (ii) A happens before B. This definition has a significant drawback, in that it does not rule out spurious correlations, making it weaker than what is often meant when discussing a causal relationship. For example, analyzing a data set consisting of yearly doctorate degrees awarded in computer science (D), precipitation in Kansas (K), and precipitation in Mississippi (M), all between the years 1996 and 2009, shows that D is better predicted using both K and M rather than only using M (Vigen, a,b). However, barring some contrived chain of events, rainfall in Kansas while I write this dissertation is clearly not going to cause my (or anyone else's) defense to be successful. Nevertheless, Granger causality is computationally easy to test for, which has led to many applications and extensions. See Eichler (2012) for a discussion of these extensions as well as a comparison to other definitions of causality.

1.1.3. Rubin causal model framework

Another notion of causality used today is the Rubin causal model (Rubin, 1974, 2005), built upon the potential outcomes framework (Neyman, 1923), and thus also sometimes called the Neyman-Rubin framework. Establishing causation in the RCM requires finding a measurable effect of some intervention, according to the maxim "no causation without manipulation". Thus, it is important that, when asking the question "Does A cause B?", A must be something that (at least in principle) can be manipulated or intervened on.

In the ideal case, one sets up an experiment, with a clear distinction between the treatment variables (those that will be intervened on) and the outcome variables (those that will be measured). Units (these could be subjects in a clinical drug trial, plots of land in testing a new fertilizer, etc.) are then randomly assigned to one of two groups, and the intervention is carried out for one group but not the other. Measuring the outcomes for the non-intervention group establishes a sort of baseline, which can then be compared to the outcomes of the intervention group. The resulting difference between the outcomes of the two groups is termed the *causal effect* of the intervention. Importantly, this random

assignment of units into either the intervention or non-intervention groups helps ensure that the units in the different groups are distinguished only by the intervention itself, as opposed to other distinctions that could lead to a difference in the outcomes of the two groups; this in turn ensures that any measured causal effect is actually causal and not merely a spurious correlation.

For example, imagine we want to test whether a certain kind of fertilizer increases crop yields. Thus, we have treatment variable F for whether or not a plot of land receives fertilizer and the outcome variable Y for the weight of the harvested fruit divided by the area of the plot. By randomly assigning plots to either receive fertilizer or not, we ensure that the growing conditions (such as sunlight, available water, etc.) vary in similar ways between the two different groups, as opposed to plots in one group having generally better conditions, which could invalidate any conclusions about the causal effect of fertilizer.

However, even if possible in principle, intervention is often practically infeasible. In these cases, when data is purely observational and not experimental, causal inference is still possible, but it becomes more complicated. As in the experimental case, there is a clear distinction between treatment and outcome variables. However, because there is no intervention, units cannot be randomly divided into two groups with their treatment variable values assigned accordingly—instead there is some non-random assignment mechanism. Thus, instead of simply taking the difference between the outcomes of the two treatment groups to calculate the causal effect, a procedure must be employed to control for the non-random assignment mechanism and its effects, which can introduce a spurious correlation or hide a causal relationship between the treatment and outcome variables. Such a procedure is called matching and involves relying on other observed variables to pick a subset of the units that fulfill certain properties one would expect if the assignment mechanism had been random, thus resulting in the *ignorablility* of the assignment mechanism (Rubin, 1973; Rosenbaum and Rubin, 1983; Dehejia and Wahba, 1999).

Returning to the fertilizer example, imagine instead of setting up an experiment we are given a bunch of observational data containing the variables F and Y. Using our domain knowledge that plants take in nutrients (some of which are provided by fertilizer) and convert them into plant matter (some of which is fruit), as well as the fact that F is directly manipulable while Y is not, we can say F is the treatment variable and Y is the outcome variable. Importantly, notice that the direction of the causal relation (if a relation exists) from F to Y is determined by domain knowledge and the manipulability of the cause, as opposed to some purely mathematical procedure. This aspect of the RCM has its advantages but also its drawbacks: on the one hand, it helps ensure that our results are causally plausible (unlike the rainfall in Kansas causing a successful dissertation defense example found using Granger causality (Section 1.1.2)); on the other hand, it makes it difficult to use the RCM in non-experimental cases where such domain knowledge is lacking or where both causal directions seem plausible, and furthermore, it thus relies on the accuracy of our domain knowledge. With the treatment and outcome variables established, we now must make up for the fact that the data is non-experimental by controlling for any other possible distinctions between those plots of land that received fertilizer and those that did not. Again this is only possible with domain knowledge: we must come up with other possible causes of Y, such as soil quality, sun light, water availability, etc., and acquire data for these variables and use it to perform matching. Without this matching, or if we do not control for all the right variables, any detected causal relation between F and Y may merely be spurious or even result in the wrong causal conclusion. For example, if

fertilizer was only used to help make up for otherwise very poor growing conditions, and we naively examine F and Y, it may look like using fertilizer results in worse crop yields, when in reality it improved crop yields in those poor conditions but was not enough to surpass the good growing conditions in which fertilizer was deemed unnecessary. This erroneous conclusion could be avoided by employing matching and therefore only comparing units with similar soil conditions.

A key idea of the RCM in both of the above cases is that the causal effect is with respect to groups or populations, as opposed to individual units, making it possible to circumvent what Holland et al. (1985) calls *the fundamental problem of causal inference*: no single unit can be both intervened on and not intervened on at the same time, making it impossible to see the unit-level causal effect of an intervention. Nevertheless, the causal effect for a population of units *can* be learned, because the population can be split into subpopulations, so that one subpopulation is intervened on while the other is not. Thus, an important assumption here is *structural homogeneity*, i.e., that despite being composed of unique units, the population nevertheless shares a common causal (and therefore statistical) structure—this is motivated by "population" as opposed to "typological" thinking (Xie, 2013).

See Holland et al. (1985) for a more formal overview of the RCM as well as a discussion of its history and relation to other ideas about causality. The RCM framework is mathematically subsumed by the graphical causal models discussed in the next section (Pearl, 2009, Ch. 7.4.4) (cf. Gelman, 2009), however they emphasize different aspects of causality: the RCM focuses on quantifying the causal effect of interventions, while causal graphs can additionally be used more abstractly, allowing one focus on the broader causal structure. We will revisit this comparison between Rubin and graphical causal models in sections 1.2 and 3.2, and this comparison provides important context for understanding our research questions and the overarching themes connecting our different publications.

1.1.4. Causal graphical models

Current definitions of causality tend to focus on probabilistic graphical models, with the earliest example being the path diagrams introduced by Wright (1921). These diagrams contain nodes connected by arrows, building upon previous work on the *product-moment correlation coefficient* and its associated partial correlation coefficient (Pearson, 1895; Galton, 1889; Isserlis, 1914)[2], allowing for a systematic graphical representation of the

[2]Though a digression, it is important to note that these methods were developed as part of eugenics research. For example, Karl Pearson (after whom Pearson correlation is named), in his retirement speech (Filon et al., 1934), reflected on his career at University College London and as editor of the journal Biometrika:

> "The climax culminated in Galton's [Pearson's mentor] preaching of Eugenics, and his foundation of the Eugenics Professorship. Did I say 'culmination'? No, that lies rather in the future, perhaps with Reichskanzler Hitler and his proposals to regenerate the German people. In Germany a vast experiment is in hand, and some of you may live to see its results. If it fails it will not be for want of enthusiasm, but rather because the Germans are only just starting the study of mathematical statistics in the modern sense!"

Here, as always, it is worth emphasizing that no amount of mathematical sophistication can ever justify racism or genocide. See Horkheimer (1972); Sim (2004) for an introduction to the field of critical theory, which analyzes and critiques the positivism and scientism underlying Pearson's ideas; and see Crenshaw et al. (1995) for an introduction to critical race theory, which argues that social problems result from the structure and organization of society (as opposed to resulting from the maladapted

structures underlying correlation coefficients between variables. In these diagrams, the correlation coefficient between two variables is taken to be the result of all paths connecting them, leading Wright to define the *path coefficient* to measure the direct influence along each individual path—and, with appropriate domain knowledge, this influence can be interpreted as a causal relation. Thus, unlike other statisticians of the time, Wright responded to the warning "correlation does not imply causation" not by ignoring questions of causation but instead by using correlation to come up with new mathematical definitions that (along with domain knowledge) *do* allow one to reason about causes by using estimated correlation coefficients (Wright, 1934, 1923) (cf. Niles, 1922, 1923).

Wright's path diagrams eventually gave rise to the structural equation models (SEMs) and their representation as directed acyclic graphs (DAGs), as well as various techniques for path analysis that are now ubiquitous in the social sciences (Westland, 2015), but his diagrams are the first example of some important ideas used for structure learning in today's causal graphical models. Namely, he provided a rudimentary framework facilitating the use of patterns of correlations to reason about causal structure, by translating between the language of probability theory (in which random variables are related to each other by correlation coefficients) and the language of graphs (in which nodes are related by arrows).

Two main developments were responsible for refining Wright's rudimentary framework: (i) the introduction of the do-calculus (Pearl, 1995), extending the language of probability theory to better capture causal relations, making it possible to have genuinely causal formal probabilistic models, and (ii) the introduction of d-separation (Verma and Pearl, 1988), which not only ensures the causal interpretation of these formal models but also makes it possible to abstract away from the probabilistic details of these models and focus on the causal structure more generally.

Functional causal models

The do-calculus (Pearl, 2012), consisting of the do$(\cdot)$ operator for representing an intervention or manipulation of a random variable along with inference rules for computing the causal effects of interventions (cf. Spirtes et al., 2000, Manipulation Theorem and Theorem 7.1), extends the language of probability theory so that it can describe causal instead of merely associative relationships. These inference rules lead to the same results that can be found using the RCM in both the experimental (interventional) setting and, using the *back-door criterion* (compare to Rosenbaum and Rubin (1983)'s conditions for ignorability), the observational setting. However, the added formalism facilitates extending SEMs (which, recall, are linear and Gaussian) to fully nonparametric functional causal models (FCMs). Thus, the do-calculus provides the key semantic component for extending Wright's framework so that it has a justifiably causal interpretation.

Pearl (2009, Ch. 1.4) offers a more detailed and formal introduction to FCMs. For the purposes of the current discussion, the most interesting aspect of FCMs is not the functional model itself or its use estimating causal effects but rather its corresponding DAG and more abstract representation of causal structure.

genes of "lesser races", like Pearson and his associates thought).

Causal structure learning

The rules of the do-calculus are stated in terms of how interventions affect the structure of probabilistic independence among random variables. Thus, the ability of DAGs to capture causal structure is due to the correspondence between d-separation (d for direction), which is a property of variables (nodes) in a directed graph, and probabilistic independence.

Definition 1 (d-separation) Nodes A and B are d-separated given a (possibly empty) set of nodes $\mathbf{Z}$ if there is no *active path* between them. A path between A and B, a sequence of nodes connected by arrows, is active iff

1. for every collider Q on the path, Q or one of its descendants is in $\mathbf{Z}$ (e.g., in the path $A \to Q \leftarrow B$, Q is a collider, and in $Q \to \ldots \to D$, D is a descendant of Q)
2. and every non-collider on the path is not in $\mathbf{Z}$.

(See (Neapolitan et al., 2004, Ch. 2.1) for a formal presentation and examples.)

This is the key syntactical component needed to refine Wright's rudimentary framework. Like independence, d-separation is symmetric, but its definition in terms of paths of directed arrows make it amenable to representing the asymmetric nature of causation. Thus, d-separation facilitates the implicit encoding of probabilistic dependence between variables in the form of a DAG, so that causal relationships can be represented explicitly and intuitively.

An important assumption underlying the causal interpretation of Wright's path diagrams is that correlations (insofar as they reflect probabilistic dependence) are ultimately due to causal relations. Though not in the context of path diagrams, Reichenbach (1956) articulated and argued for this assumption, using it to give a formal probabilistic characterization of causality:

Common cause assumption (Reichenbach, 1956, p. 157) If random variables A and B are probabilistically dependent, then either

1. A is a cause of B,
2. B is a cause of A, or
3. there exists a C that is a cause of them both.

Compared to path analysis in Wright's framework using linear correlation coefficients, the common cause assumption (CCA) focuses more generally on probabilistic dependence, as well as explicitly assuming causal relations as the explanation for these dependencies. The CCA is a basic assumption for many methods of learning a FCM from a data set, especially when combined with the assumption of *causal sufficiency*, i.e., the assumption that, for a given data set, for all variables A and B that are dependent but do not satisfy 1. or 2. of the CCA, there is a variable C satisfying 3. also included in the data set (Pearl, 2009, p. 30). These two assumptions together are known as the *causal Markov assumption*, which guarantees that the causal relationships (and therefore probabilistic dependencies) between all the variables can be represented as a DAG. Using d-separation, it can (as in (Pearl, 2009, Theorem 1.2.4)) be stated as:

Causal Markov assumption: Given a set of random variables $\mathbf{V}$ whose causal structure is represented as the DAG $\mathcal{G}$, if $V_i, V_j \in \mathbf{V}$ are d-separated in $\mathcal{G}$ (with conditioning set $\mathbf{C} \subset \mathbf{V}$), then V_i and V_j are probabilistically independent given $\mathbf{C}$, i.e., then $V_i \perp\!\!\!\perp V_j \mid \mathbf{C}$.

The converse of the CMA, which often accompanies it, can be stated similarly:

Causal faithfulness assumption (CFA): Given a set of random variables $\mathbf{V}$ whose causal structure can be represented as a DAG $\mathcal{G}$, if $V_i \perp\!\!\!\perp V_j \mid \mathbf{C}$, then all V_i and V_j are d-separated in $\mathcal{G}$ (conditioned on $\mathbf{C}$).

Under these assumption, the task of learning the causal structure among a set of variables becomes a matter of using statistical methods to estimate which random variables are probabilistically dependent (and thus not d-separated in the DAG) and then figuring out which of the three possible causal relationships in the CCA is responsible for each pair of dependent variables (i.e., figuring out how to direct the edges along the paths).

One way of figuring this out is by doing interventions, reminiscent of the RCM framework. The graphical implications of the do-calculus, by comparing d-separations in the graph before and after performing an intervention, make it possible to learn the directions of all arrows in the DAG.

More interestingly, it can be done (to an extent) from purely observational data. Even without comparing before and after interventions, the definition d-separation makes it possible to orient some edges in the DAG, namely those that form colliders and subsequently those whose direction is required to ensure acyclicity in the DAG. Two classic algorithms for this are the PC and IC algorithms (Spirtes and Glymour, 1991; Verma and Pearl, 1990).

The do-calculus and d-separation thus provide the mathematical tools needed, so that given the CMA and CFA, causal structure learning (even from purely observational data!) in the form of a DAG is possible. Readers familiar with the field of mathematical logic may find a helpful analogy here: just as Gödel's completeness theorem establishes a correspondence between syntax and semantics in first-order logic, the causal Markov and faithfulness assumptions establish a correspondence between the graphical syntax of DAGs and the causal semantics of FCMs (understood formally, as an extension of the language of probability theory, and excluding unmeasured confounders as well as selection variables). A similar correspondence (though based on different assumptions) also exists for the more general case of ancestral graphs, which can represent unmeasured confounders and selection bias.

Generalized causal structure learning

Unlike DAGs, which contain only directed edges (like $\rightarrow$), ancestral graphs (AGs) contain three edge types. These edge types, along with the extension of d-separation to m-separation and correspondingly modified causal Markov and faithfulness assumptions allow for much more expressive causal graphs than is possible with DAGs.

Because DAGs assume causal sufficiency, they are not able to represent causal relationships among sets of variables in violation of this assumption, i.e, learning a DAG for a set of such variables will result in incorrect causal conclusions. In contrast, AGs do not assume causal sufficiency, instead using their added expressiveness to represent the presence of latent variables with edge types not found in DAGs. AGs are thus able to represent causal

models over arbitrary sets of variables, ensuring correct causal conclusions in cases where DAGs would fail.

See (Richardson and Spirtes, 2002) for a detailed formal definition of AGs, (Richardson and Spirtes, 2003) for a thorough discussion of how to interpret the different edge types, (Ali et al., 2009) for a definition of Markov equivalence in AGs, and (Spirtes et al., 1999) for a learning algorithm.

Finally, it is important to emphasize that, the preceding historical overview aside, this dissertation uses 'cause' and related words as technical terms, with narrow, formal definitions based on the do-calculus, d-separation, the CMA, and the CFA. Though such a technical term attempts to capture some important aspects of causality, it of course also ignores some other important aspects of causality as understood in more general philosophical or natural language contexts as well as other technical contexts with different formal definitions.

1.2. Research questions revisited

With the foundations of causal inference laid out, we can now more explicitly describe our research questions.

Q.1, having to do with learning a causal model while only observing parts of the system can be reframed in terms of causal inference under a modification of the CMA. We address this question in Section 2.1, replacing the assumption of causal sufficiency with the assumption of *strong causal insufficiency*, i.e., the assumption that none of the observed variables cause one another and thus that their dependence is induced by unobserved latent variables. In doing so, we find undirected graphs, as opposed to DAGs, to be especially helpful, and we provide a novel causal semantics for them. This should not be seen as a weaker or stronger assumption than that of causal sufficiency but rather just a different assumption suited for different tasks—there are some causal relations representable in our model that are not representable with DAGs and vice versa, though the relations in both are representable with AGs. For example, in applications such as psychiatric diagnostic questionnaires, the data consists of measurements of symptoms but not the causes themselves, making strong causal insufficiency more reasonable than sufficiency. Furthermore, this hints at another connection, namely, to the a priori distinction in the RCM between variables that are possible causes and those that are effects.

Q.2, having to do with learning these models from populations resulting from not one but multiple causal systems, can be stated more clearly in terms of the structural homogeneity assumption. We address this in Section 2.3 by introducing a measure of structural (graphical) similarity, making it possible to determine if the assumption is violated for a given data set, and if so, to split the data set into clusters which individually do satisfy the assumption.

Q.3, concerning learning these models without assuming any particular distribution, is addressed in each section of Chapter 2 by focusing on the generality of FCMs and their corresponding graphical representation of probabilistic independence as opposed to the Gaussianity assumption of SEMs.

Notice that in each of these cases, we address questions that draw from the RCM framework but by making use of and extending the graphical framework facilitated by the FCM. Thus, though the FCM mathematically subsumes the RCM, the less abstract

conception of causality in the RCM leads to the emphasis of (1) the importance of distinguishing between possible causal mechanisms and measurements of their effects, as well as (2) the population as opposed to the typological perspective and the importance of considering homogeneity in causal learning tasks. At the same time, the more abstract graphical conception of causality in the FCM framework makes it possible to (1) discover causal relations, even when lacking the a priori knowledge needed to hypothesize different causes and effects, as well as (2) characterize the relevant population homogeneity in explicit terms of causal structure.

2. Publications

This chapter consists of the three conference papers that form the basis for this cumulative dissertation. Each paper is thus somewhat self-contained but still assumes familiarity with certain topics, as is appropriate for the respective conference communities. The synopses additionally contain the current publication/review status and complete bibliographic information of the paper as well as an explicit statement of my contributions (as opposed to those of my co-authors).

2.1. MeDIL Causal Models

2.1.1. Synopsis

We introduce a new class of causal model (called MCMs) for scenarios in which a scientist only has access to the observed effects of unobserved causes. These MCMs provide a causal semantics for the graph theoretic edge clique cover (ECC) problem, which facilitates learning a minimal MCM from measurement data by representing its pairwise probabilistic independence relations as an undirected graph and then finding the minimal ECC over that graph. We demonstrate learning and interpreting a minMCM for a real-world psychometric data set.

Complete bibliographic information

Markham, A. and Grosse-Wentrup, M. (2020). Measurement dependence inducing latent causal models. In Peters, J. and Sontag, D., editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 590–599. PMLR

My contribution

- connected the causal inference problem to the graph theoretic problem
- derived the theoretical results
- implemented the algorithm, applied it to data, and produced all plots
- wrote the paper (with help, especially in Section 5.2)

Measurement Dependence Inducing Latent Causal Models

Alex Markham[1] **and Moritz Grosse-Wentrup**[1 2 3]
[1]Research Group Neuroinformatics, Faculty of Computer Science, University of Vienna
[2]Research Platform Data Science @ Uni Vienna
[3]Vienna Cognitive Science Hub

Abstract

We consider the task of causal structure learning over measurement dependence inducing latent (MeDIL) causal models. We show that this task can be framed in terms of the graph theoretic problem of finding edge clique covers, resulting in an algorithm for returning minimal MeDIL causal models (minMCMs). This algorithm is non-parametric, requiring no assumptions about linearity or Gaussianity. Furthermore, despite rather weak assumptions about the class of MeDIL causal models, we show that *minimality* in minMCMs implies some rather specific and interesting properties. By establishing MeDIL causal models as a semantics for edge clique covers, we also provide a starting point for future work further connecting causal structure learning to developments in graph theory and network science.

1 INTRODUCTION

Despite the many theoretical and practical difficulties, establishing and understanding causal relationships remains one of the fundamental goals of scientific research. Consequently, many different approaches have been developed, with applications spanning a diverse range of fields, e.g., from epidemiology to psychometrics to neuroimaging (Parascandola, 2001; Hoover, 2006; Seth et al., 2015). Some of the most well-known approaches include Granger causality (Granger, 1969) for time-series data, the Rubin causal model and potential outcomes framework (Holland, 1986) for randomized controlled trials, and functional causal models and the representation of their causal structure as directed acyclic graphs (Pearl, 2000; Spirtes et al., 2000). The last of these, the directed acyclic graph (DAG), provides the context for our approach to causal structure learning.

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

Roughly speaking, causal structure learning (CSL) typically focuses on identifying which variables are directly causally related and how these *direct causal relations form a structure* over which indirect causal relations exist. One way of characterizing CSL algorithms is according to which of the three following assumptions they rely on: (i) the *causal Markov assumption*, which says the random variables are (conditionally) independent (denoted by $\perp\!\!\!\perp$) if the corresponding vertices in the DAG are d-separated (denoted by $\perp$); (ii) the *causal faithfulness assumption*, which says the vertices in the DAG are d-separated if the corresponding random variables are (conditionally) independent; and (iii) the *causal sufficiency* of the set of variables, i.e. that there are no unobserved or latent common causes. The basic approach to CSL—namely the original constraint-based IC and PC algorithms (Verma and Pearl, 1990; Spirtes and Glymour, 1991)—rely on all three, while many of the algorithms developed in the 30 years since (as we will see in Section 1.1) relax these assumptions.

Considering applications of CSL to, for example, psychometrics and neuroimaging, the assumption of causal sufficiency seems implausible. For a data set consisting solely of answers to a depression diagnostic questionnaire or of voxel intensities in calcium imaging recordings (with random variables corresponding respectively to the questions or voxels), we think it is relatively uncontroversial to claim that not only are the random variables not causally sufficient, but indeed *every* dependence relation among them is induced by unobserved latent variables (respectively either cognitive processes related to, e.g., depression, or calcium signaling in cellular tissue, plus other confounders). In fields and applications such as these—where interventions are often difficult or unfeasible, and where the goal is to reason about underlying causes based on their measurable effects—a more tailored causal modeling framework may prove insightful. Thus, the main difference between the traditional approach outlined above and the one we present in this paper is that

we assume a strong causal *in*sufficiency of the random variables being modeled and therefore are able to represent a different (but not entirely disjoint) class of causal structures than is possible with DAGs in the traditional approach.

The rest of the paper is organized as follows: We begin by reviewing related work, emphasizing points of departure. In Section 2 we define measurement dependence inducing latent (MeDIL) causal models to be the class of latent measurement models in which measurement variables can only be effects (and not causes—contrary to the definition of measurement models explored by others), making no further assumptions about linearity or parametrizations of the distributions. We then introduce the notions of observational consistency and minimality, allowing us to, for a given (estimated) distribution of measurement variables, construct a minimal MeDIL causal model (minMCM). Then, in Section 3, by framing minMCMs as edge clique covers (ECCs) of the undirected dependency graph over measurement variables, we note how two notions of minimality emerge. Subsequently, despite our nonrestrictive assumptions and notion of minimality in minMCMs, we are able to prove (i) that a minMCM lower bounds the number of latent variables or the number of functional causal relations (depending on which notion of minimality is used), (ii) that the latent variables of the minMCM are all pairwise independent, and (iii) that (somewhat surprisingly) the minMCM can have more latent causes than measured variables. In Section 4 we describe an algorithm for learning minMCMs from only *unconditional* (in)dependencies. Finally, we demonstrate our approach with an application to a psychometric data set in Section 5, before concluding with a discussion of promising directions for future work.

1.1 RELATED WORK

Elaborating on the basic approach mentioned above, CSL without latents amounts to finding an *essential graph* (Andersson et al., 1997), a mixed graph with directed and undirected edges, which represents the Markov equivalence containing the true DAG. The essential graph is typically found by using either a score- or constraint-based approach. Score-based methods find an essential graph by directly optimizing a score of how well it fits the data samples (Chickering, 2002).Constraint-based methods take a set of conditional independence relations as input (which must be estimated or acquired somehow before applying the algorithm), and these relations constitute a set of constraints on the possible d-separations, which the output essential graph satisfies (Verma and Pearl, 1990; Spirtes and Glymour, 1991). Our approach in this paper is more closely related to constraint-based methods, especially their extensions to latent variable models.

Extensions of CSL to causal models including latent variables (i.e., relaxing the causal sufficiency assumption), such as the FCI algorithm and its variants (Spirtes et al., 1999), correspondingly extend the search space from essential graphs to partial ancestral graphs (PAGs), which have an additional three edge types (so five total), allowing them to represent the extended Markov equivalence class containing dependencies induced by latent variables.

In these terms, our latent CSL algorithm *is not* searching for a PAG. As we explain in sections 2 and 3, by making use of the strong causal insufficiency in this application space, we can directly represent the conditional independence constraints that form the input for our algorithm as an undirected dependence graph (UDG). This UDG is essentially a PAG with only bidirected edges. Or, put another way, it is a modified Markov random field (Kindermann and Snell, 1980) where the conditional independence relations are determined from the undirected edges by using strong causal insufficiency (see Proposition 6) instead of the Markov property, thereby allowing the UDG to represent latent induced dependence (which Markov random fields are usually incapable of representing).

With the conditional independence constraints input in the form of a UDG over measurement variables, our algorithm essentially adds the latent causes and directed edges necessary to construct the minimally causally sufficient DAG containing latent and measurement variables. Thus, instead of doing CSL in the presence of latent variables as is the case with FCI and similar algorithms, we *use CSL to reason about latent variables*.

Our approach is more related in this respect to other work on measurement models (Silva and Scheines, 2005; Silva et al., 2006; Kummerfeld et al., 2014; Kummerfeld and Ramsey, 2016). However, these other approaches utilize properties of the covariance matrix of the measurement variables, such as vanishing tetrad constraints, while we utilize graph theoretic properties of the UDG representation of conditional independencies. This results in connections between our approach and causal feature learning (Chalupka et al., 2016) and causal consistency and abstraction (Rubenstein et al., 2017; Beckers and Halpern, 2019), which will be discussed more with respect to future work in Section 6.2. Another closely related approach is factor analysis, especially when framed in terms of using the topology of a Bayesian network of observed variables to reason about hidden factors (Martin and VanLehn, 1994), with the main difference being our goal of a minimally causally sufficient DAG as opposed to a statistically convenient (but not necessarily as causally relevant) factor model.

Overall, our approach has several points of overlap in terms of motivations and formal methods in existing CSL, measurement model, and factor analysis approaches. However, we address the problem from a different perspective, utilizing the causal insufficiency property of our application space and graph theoretic edge clique cover methods to produce a novel algorithm.

2 MINIMAL MEDIL CAUSAL MODELS

We begin with a formal definition of *measurement dependence inducing latent (MeDIL) causal models*, before discussing the notion of observational consistency and its implications about minimality in such models.

We use functional causal models (FCMs) to describe causal relations in complex systems.

Definition 1 (Functional Causal Model)**.** A *functional causal model* is a triple $\mathcal{M} = \langle \mathbf{V}, \mathbf{F}, \boldsymbol{\epsilon} \rangle$, where

- $\mathbf{V}$ is the set of (endogenous) random variables,
- $\mathbf{F}$ is a set of functions defining each endogenous variable as a function of its direct causes (i.e., parents or $\mathrm{pa}()$) and its corresponding exogenous random variable, so that for each $V_i \in \mathbf{V}$, we have $V_i := f_i(\mathrm{pa}(V_i), \epsilon_i)$. Furthermore, $\mathbf{F}$ is constrained such that no V_i is a direct cause of itself or any of its causes, removing the possibility of causal cycles.
- $\boldsymbol{\epsilon}$ defines a joint probability distribution over the exogenous (or noise) variables, with a corresponding $\epsilon_i \in \boldsymbol{\epsilon}$ for each $V_i \in \mathbf{V}$, and with ϵ_i being independent with ϵ_j for each $\epsilon_i, \epsilon_j \in \boldsymbol{\epsilon}$

□

In particular, we are interested in latent CSL over measurement variables, so it is advantageous to move from the general FCM definition to a specifically structural/graphical definition that conceptually differentiates the set of endogenous variables into causally effective latent variables and their observed measurements, leading to the idea of MeDIL causal models:

Definition 2 (Measurement Dependence Inducing Latent Causal Model (MCM))**.** A graphical MCM is a DAG, given by the triple $\mathcal{G} = \langle \mathbf{L}, \mathbf{M}, \mathbf{E} \rangle$. $\mathbf{L}$ and $\mathbf{M}$ are disjoint sets of vertices, while $\mathbf{E}$ is a set of directed edges between these vertices, subject to the following constraints:

1. all vertices in $\mathbf{M}$ have in-degree of at least 1 and out-degree of 0
2. all vertices in $\mathbf{L}$ have out-degree of at least 1
3. $\mathbf{E}$ contains no cycles

□

There are no further constraints as to the variety of distributions and functional causal relations that MCMs can represent, i.e., they are non-parametric and their arrows can represent arbitrary functional relations between variables. The formal constraints 1. and 2. in Definition 2 are to ensure that MCMs are applicable to settings in which we can explicitly separate into disjoint sets the measured effect variables $\mathbf{M}$ whose probabilistic dependencies must therefore be mediated by latent causes $\mathbf{L}$.

However, the explicit separation of cause and effect and the corresponding latent structure in MCMs introduces its own difficulties for inference. Namely, many latent models are consistent with a given probability distribution over observed effects, making the task of inferring a single latent model ill-posed. In order to help explain this consistency of different latent models and illustrate our strategy for restricting the problem so that inference is well-posed, consider the following definition and example.

Definition 3 (Observational Consistency)**.** A MCM is *observationally consistent* with a probability distribution over measurement variables if it is capable of inducing the pairwise dependencies (which can estimated from samples) of that distribution. This can be seen as a weakening of the notion of observational equivalence corresponding to our extension from DAGs containing only observed variables to the notion of MCMs.[1]

Example 4 (Observational Consistency)**.** Suppose we have data consisting of peoples' answers to a questionnaire with four questions designed to measure depression and stress. We assume that the answer to one question cannot cause the answer to another and therefore that the observed answers as well as any observed association between answers are the result of latent causes, such as depression or stress. Define random variables $\mathbf{M} = \{M_1, M_2, M_3, M_4\}$ corresponding to answers to the four questions, and let them have only the following two pairwise independencies:

$$M_1 \perp\!\!\!\perp M_4 \quad \text{and} \quad M_2 \perp\!\!\!\perp M_4$$

The pairwise dependency structure between variables in $\mathbf{M}$ is shown in Figure 1(a), and three observationally consistent MCMs are shown in 1(b), 1(c), 1(d). As this example demonstrates, multiple latent models can give rise to the same set of observed dependencies.

□

[1] *observational* or *Markov equivalence* (Pearl, 2000, pp. 16–20) means two DAGs have the same skeletons and colliders, while observational consistency means that two MCMs have the same undirected dependency graphs over measurement variables (e.g., Figure 1)

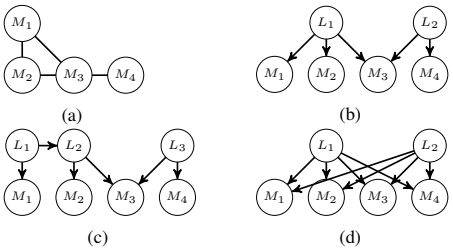


Figure 1: (a) undirected dependency graph over $\mathbf{M}$—notice two missing edges corresponding to independencies; (b) minimal MCM over $\mathbf{M}$; (c) non-minimal MCM observationally consistent with $\mathbf{M}$; (d) MCM corresponding to ICA or FA

We address this problem by employing Ockham's razor to pick a *minimal MCM (minMCM)* (e.g., Figure 1(b)).

Definition 5 (Minimal MeDIL causal model (minMCM))**.** A *minMCM* for a set of measurement variables $\mathbf{M}$ is any least expressive (i.e., minimal) MCM that is observationally consistent with $\mathbf{M}$. As Pearl and Verma (1995) note, a latent causal model's expressive power can be measured by the (in)dependencies it induces over the measured variables, with more dependencies corresponding to more expressive power. In our case, criteria can be given for minimality in modified terms of the causal faithfulness and causal Markov assumptions:

1. in addition to being observationally consistent with its set of measurements, a minMCM must graphically induce the measurements without violating faithfulness; the notion of faithfulness used here is concerned with conditional independencies only over measurements and not all variables in the MCM, so we call it *measurement-faithfulness*; note that Figure 1(b) is faithful to the conditional independencies in Example 4 while Figure 1(d) is not—the MCM in Figure 1(b) is minimal while that in 1(d) is not

2. considering arbitrary subsets of the latents, $\mathbf{Z} \subseteq \mathbf{L}$, there are as few d-separations of the form $M_i \not\perp\!\!\!\perp M_j \mid \mathbf{Z}$ as (faithfully) possible, i.e., such d-separations only exist in an minMCM if implied by the (in)dependencies and causal insufficiency of the distribution only over measurement variables; we call this *measurement-Markov* since it says the only d-separations in the minMCM are those implied by measurement-faithfulness[2]; note that Figure 1(c) does not satisfy this □

[2] just as is the case with the usual causal faithfulness and Markov conditions

Learning a minMCM for a data set only requires considering the *unconditional* independence relations among its variables, unlike the other methods mentioned in Section 1.1. This follows from Proposition 6.

Proposition 6. *In a MCM, the set of unconditional (in)dependencies over measurement variables fully determines the set of conditional (in)dependencies over measurement variables.*

Proof. The Causal Markov and Causal Faithfulness assumptions (CMA and CFA, respectively) imply that two variables are probabilistically independent if and only if they are d-separated (allowing us to use independence/d-separation and $\perp\!\!\!\perp$ / $\perp$ interchangeably). Recall from Definition 2 that all dependence relations (and therefore, by the CMA and CFA, d-connections) between measurement variables are mediated by latent variables. Hence, all measurement variables have out-degree 0, and so any measurement variable in a path between two other measurement variables must be a collider and any dependent measurement variables must share at least one latent parent. This means that the set of unconditional (in)dependencies over measurement variables fully determines the set of conditional (in)dependencies as follows: for all $M_i, M_j, M_k \in \mathbf{M}$,

- $M_i \not\perp\!\!\!\perp M_j \implies M_i \not\perp\!\!\!\perp M_j \mid M_k$
- $M_i \perp\!\!\!\perp M_j \implies$
$$\begin{cases} M_i \perp\!\!\!\perp M_j \mid M_k, & \text{if } M_i \perp\!\!\!\perp M_k \text{ or } M_j \perp\!\!\!\perp M_k \\ M_i \not\perp\!\!\!\perp M_j \mid M_k, & \text{otherwise} \end{cases}$$

□

As we will see in Section 4, even though estimating conditional independencies is not required for our method, doing so nevertheless can help determine whether any of the assumptions have been violated.

3 MINIMAL MEDIL CAUSAL MODELS AS EDGE CLIQUE COVERINGS

We can now present our main insight:

Proposition 7. *The problem of finding a minMCM for a set of measurement variables can be framed as the graph theoretical problem of finding a minimum edge clique covering (ECC)*[3] *(Erdős et al., 1966; Gramm et al., 2009; Ennis et al., 2012) over the corresponding undirected dependency graph of the measurement variables.*

[3] A minimum ECC over an undirected graph is a collection of cliques that exactly covers its edges, where an edge $E = (V_i, V_j)$ is covered by clique C iff $V_i, V_j \in C$.

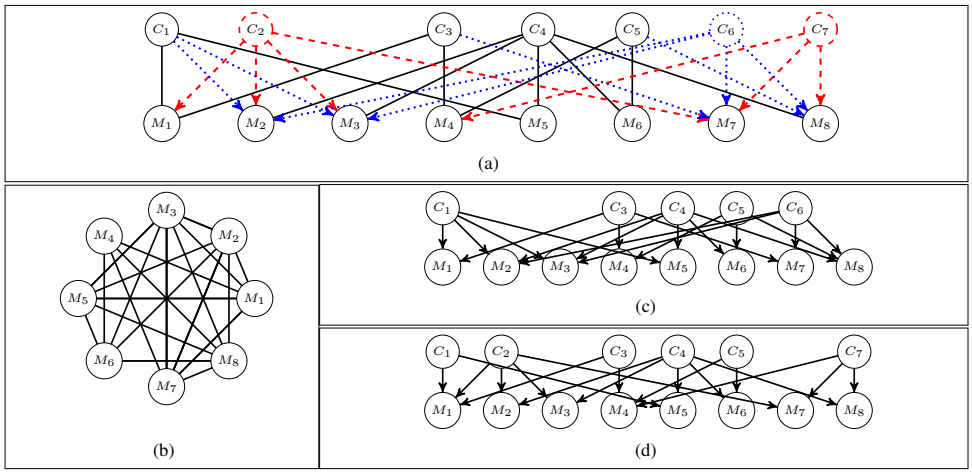


Figure 2: (a) MCM, where each C_i corresponds to a maximal clique in $\mathrm{D}(\mathbf{M})$—dashed red edges/vertices are redundant for vertex-minimality while blue dotted edges/vertices are redundant for edge-minimality; (b) $\mathrm{D}(\mathbf{M})$—undirected dependency graph of $\mathbf{M} = \{M_1, \ldots, M_8\}$; (c) vertex-minimal minMCM of $\mathrm{D}(\mathbf{M})$; (d) edge-minimal minMCM of $\mathrm{D}(\mathbf{M})$

Proof. For a given set of measurement variables $\mathbf{M}$, denote the *undirected dependency graph* as $\mathrm{D}(\mathbf{M})$, e.g., Figure 1(a), where an edge represents dependence and the lack of an edge represents independence. Proposition 6 tells us that $\mathrm{D}(\mathbf{M})$, though it only encodes unconditional (in)dependencies, contains all necessary information for characterizing observationally consistent MCMs. Consider the MCM $\mathcal{G} = \langle \mathbf{L}, \mathbf{M}, \mathbf{E} \rangle$ constructed from a set of cliques $\mathbf{C}$ comprising a minimum ECC over $\mathrm{D}(\mathbf{M})$ using the following procedure: (i) posit a latent $L_C \in \mathbf{L}$ iff $C \in \mathbf{C}$ and (ii) posit a directed edge $E \in \mathbf{E}$ from the latent L_C to the measurement variable M iff $M \in C$. In other words, $\mathbf{G}$ is a MCM with measurement variables $\mathbf{M}$, one latent for each clique in the minimum ECC over $\mathrm{D}(\mathbf{M})$, and an edge from each latent to exactly the measurement variables in the corresponding clique.

Note that $\mathbf{G}$ is not only observationally consistent with $\mathrm{D}(\mathbf{M})$ but also captures its independencies and is thus faithful, satisfying criterion 1. of Definition 5. Furthermore, the construction of $\mathbf{G}$ from a minimum ECC ensures that latents are only posited when necessitated by the dependencies between measurements, satisfying criterion 2. of Definition 5. Thus, $\mathbf{G}$ is an minMCM for $\mathrm{D}(\mathbf{M})$.

□

A minimum ECC can be minimal in two related but distinct ways: the original and more well-studied approach seeks the smallest number of cliques needed to cover all edges (this is equivalent to the *intersection number* (Erdős et al., 1966)), while another justifiable approach is to seek an ECC requiring the fewest assignments of vertices to cliques. The corresponding interpretation for minMCMs is vertex-minimal (fewer cliques imply fewer latents imply fewer total vertices) and edge-minimal (fewer assignments of measurement vertices to cliques implies fewer directed edges from latent to measurement vertices), resulting in Proposition 8. There are some undirected dependency graphs for which the vertex-minimal and edge-minimal minMCMs are identical, such as figures 1 and 3, but this identity does not hold generally (Ennis et al., 2012) (see Figure 2). In either approach to minimality, the resulting minMCM induces the same set of dependencies over measurement variables and thus has the same expressive power (w.r.t. the measurement variables). We thus see no straightforwardly principled way of picking one approach over the other, and so we present both in hopes that practitioners will use whichever one (or both) they judge most sensible/interesting for their particular application.

Regardless of which notion of minimality is used, minMCMs have some interesting properties. First, they lower bound (i) the number of causal concepts or (ii) the number of functional causal relations that are required to model measurements of a complex system at any level of granularity (Proposition 8). Second, minMCMs contain no

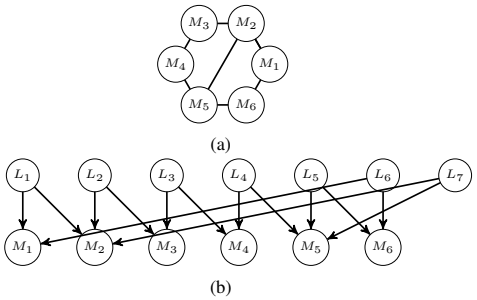


Figure 3: (a) example $\mathrm{D}(\mathbf{M})$ for which the minMCM (b) has 6 measurement variables and 7 latent variables

causal links between the latent variables (Proposition 9). Finally, in contrast to factor analysis, a minMCM may require more latent than measurement variables (Proposition 10).

Proposition 8. *For a given set of unconditional pairwise dependencies among measurement variables* $\mathbf{M}$*, a minMCM gives a lower bound on the number of latent variables or edges (depending on the measure of minimality is used) required in any (faithful and observationally consistent) MCM.*

Proof. This is a direct consequence of the construction of minMCMs from either the clique-minimum or assignment-minimum ECC of $\mathrm{D}(\mathbf{M})$, as described in Proposition 7. □

Proposition 9. *In a minMCM, each latent variable is d-separated from every other latent variable.*

Proof. Intuitively, this is a result of the definition of a minMCM being minimal in the sense of least expressive (and thus having as few latents or edges): if two latent variables are d-connected, then the dependencies among measurement variables that they induce could also instead be induced by a single latent variable (which also results in fewer edges). A minMCM has no redundant latent variables or edges and therefore no d-connected latent variables. For example, note in that the MCMs in figures 1(b) and 1(c) induce the same d-separations over the measurement variables, but that 1(b) with its d-separated latents has the fewer latents and fewer total edges. More formally, this follows directly from procedure for constructing an minMCM in Proposition 7 and Algorithm 1. □

Proposition 10. *There exist minMCMs containing more latent than measurement variables.*

Proof. This follows from the graph theoretical characterization of minMCMs: there are at least as many latent variables as the intersection number of $\mathrm{D}(\mathbf{M})$, which in a graph with n vertices is (non-trivially) upper bounded by $\frac{n^2}{4}$ (Erdős et al., 1966). A simple example can be found when $\mathrm{D}(\mathbf{M})$ is as in Figure 3, resulting in $n = 6$ nodes and an intersection number of $i = 7$. □

4 A minMCM-FINDING ALGORITHM AND ITS COMPLEXITY

The procedure in the proof of Proposition 7 for constructing a minMCM from an undirected dependency graph leads directly to Algorithm 1.

Algorithm 1: constructing a minimal MeDIL causal model (minMCM)

Input : undirected dependency graph, $\mathrm{D}(\mathbf{M})$, over the measurement variables $\mathbf{M}$

Output : vertex-minimal or assignment-minimal MCM $\mathcal{G}$ over $\mathbf{M}$

1 initialize edgeless graph with a vertex for each $M \in \mathbf{M}$;
2 find a clique-minimum or assignment-minimum edge clique cover of $\mathrm{D}(\mathbf{M})$, using the algorithm in Fig. 3 of (Gramm et al., 2009) or the algorithm `FIND-AM` of (Ennis et al., 2012), respectively;
3 **for** *each clique* C *in the cover* **do**
4 | add vertex L with edges directed to each $M \in C$;
5 **end**

Notice that Line 2 in Algorithm 1 is to find a minimum ECC of $\mathrm{D}(\mathbf{M})$. Nearly all of the computational complexity of Algorithm 1 comes from this step, which is known to be an NP-hard problem, and so the choice of an efficient ECC-finding algorithm and implementation is especially important.

In case a clique-minimum ECC (and therefore vertex-minimum minMCM) is preferred, (Gramm et al., 2009) provides an exact algorithm. The exact algorithm finds an ECC in $\mathcal{O}(f(2^k) + n^4)$ time, where k is the number of cliques in the ECC and n is the number of vertices in the undirected graph, and is thus fixed-parameter tractable. Furthermore, (Cygan et al., 2016) gives a lower bound on the complexity of the clique-minimum ECC problem and argues that the algorithm is probably optimal. Gramm et al. (2009) also provide a free/libre implementation of their algorithm, though it has not been maintained for some time and does not easily run on most modern machines.

In case an assignment-minimum ECC (and therefore edge-minimum minMCM) is preferred, (Ennis et al., 2012) provides an exact algorithm. Though they do not offer an analyses of its complexity, it is essentially a backtracking algorithm based on (Bron and Kerbosch, 1973)'s maximal clique finding algorithm, which has time complexity of$\mathcal{O}(3^{n/3})$, and so this assignment-minimum ECC finding algorithm has an even larger complexity.

As far as we are aware, no other implementations of the clique-minimum or assignment-minimum ECC finding algorithms exist. To remedy this, we have implemented and released these and a few other related causal inference tools as a free/libre Python package at https://medil.causal.dev. Already Gramm et al. (2009) and Ennis et al. (2012) showed that their algorithms perform in a reasonable amount of time on moderately sized graphs, e.g., returning a solution containing 100 cliques in a matter of minutes. Unsurprisingly, given the hardware advancements of the past decade, our implementation performs even better, e.g. finding the 614 clique solution to the 61 node graph presented in the next section in only 39 seconds using an Intel Core i7-8700K CPU.

5 APPLICATION

In this section we demonstrate the necessary steps to get from a raw data set to a minMCM output from our algorithm. We then hint at how this output can be analyzed and suggest some conclusions that can be drawn from it. Note that our contribution in this paper is theoretical, and the point of the following application is to make some of our theoretical claims and the potential use cases more concrete.

5.1 THE DATA AND PREVIOUS ANALYSES

The *Stress, Religious Coping, and Depression* data set[4] was collected by Bongjae Lee from the University of Pittsburgh in 2003. There were 127 participants answering a total of 61 questions: 21 designed to measure stress, 20 for religious coping, and 20 for depression—see (Silva and Scheines, 2005) for the full questionnaire. This data has been analyzed by several other measurement model methods (Silva and Scheines, 2005; Silva et al., 2006; Kummerfeld et al., 2014; Kummerfeld and Ramsey, 2016), and their findings (which largely agree with each other) can be briefly summarized as follows: (i) in contrast to the design goal, most of the measurement variables are "impure" in that they are caused by multiple latent variables; (ii) they are able to find some subsets (ranging in number from three to nine) of "pure" measurement variables that passed their significance tests and some of which suggest a model similar to what Lee hypothesized containing three latent variables—the first of which causes only measurement variables of stress, the second only depression, and the third only coping; (iii) most of their models scoring the highest significance are more complex models than Lee's model (the most complex containing eight latents (Silva and Scheines, 2005)).

[4]We would like to thank David Danks and especially Joseph Ramsey at Carnegie Mellon University for providing us with a copy.

5.2 ANALYSIS USING minMCMS

Notice that the input to Algorithm 1 is an undirected dependency graph, while in practice one does not have direct knowledge of the (in)dependencies themselves but only samples of the measurement variables. It is therefore necessary to first estimate the independencies before applying this algorithm. Because the algorithm is agnostic to the test statistic, it is not constrained to linear methods such as Pearson correlation (for which "$X \perp\!\!\!\perp Y \implies \mathrm{corr}(X, Y) = 0$" but not the converse) but can leverage the power of nonlinear independence tests (Gretton et al., 2005; Székely et al., 2007). We used the distance correlation (Székely et al., 2007) as our test statistic (with the property "$X \perp\!\!\!\perp Y \iff \mathrm{dCorr}(X, Y) = 0$") and performed 1000 random permutations of the measurement variables to sample from the null-distribution (Dwass, 1957). The p-value for each pair was then calculated as the proportion of the permutation tests in which the absolute distance correlation of the pair of variables with permuted samples exceeded that of the original pair. Finally, independence between two variables was concluded if the distance correlation between them was less than 0.1 and the corresponding p-value was greater than 0.1.[5]

The binary-valued 61×61 matrix corresponding to the estimated independencies, with a 0 for independence and 1 for dependence thus forms the adjacency matrix for the UDG that is input for Algorithm 1. We decided to find a latent-minimal minMCM, and the result has 614 latent variables. It is thus too complex to be legibly displayed here, so we instead present figures 4 and 5 to facilitate analysis of the results.

Looking at the histogram in Figure 4(a), we find a median indegree (i.e., number of latent causes) of the measurement variables of 27, but with one in particular, M_{30}, having 425. The item in the questionnaire corresponding to M_{30} was the ninth in the set designed to measure depression, and it asked participants how frequently the event "I thought my life had been a failure" occurred in the preceding week. Semantically, it makes sense that this item would have many more latent causes than the

[5]As one would expect, using a nonlinear measure of dependence allows us to detect more dependencies: we found almost 31% of the over 1500 estimated nonlinear pairwise dependencies (i.e., edges in the UDG) to be undetectable using the linear Pearson correlation.

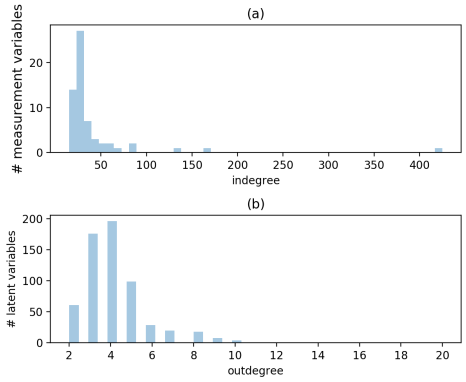


Figure 4: histograms showing (a) indegree of the measurement variables and (b) the outdegree of the latent variables

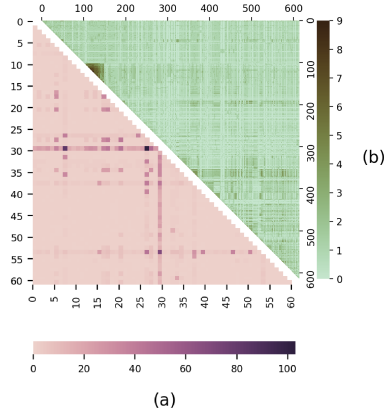


Figure 5: heatmaps showing (a) the number of latent variables each pair of the 61 measurement variables have in common and (b) the number of measurement variables each pair of the 614 latent variables have in common

other items, because its scope is much larger, requiring reflection on the participants' entire life up to that point instead of just during the week in question, as is the case for other depression items, such as "I enjoyed life" (M_{37}, 24 latent causes) and "I felt sad" (M_{39}, 25 latent causes). Furthermore, looking at Figure 5(a), showing the number of latents each pair of measurement variables share, we see that M_{30} shares a relatively high amount of latent causes with the other measurement variables (median of 21), while for M_{37} and M_{39} the median of shared latent causes is one. Our analysis thus agrees with the previous analyses described in Section 5.1 insofar as we also find many "impure" measurement variables, but extends their insights by differentiating between measurement variables that are best considered a general or mixed measurement (M_{30}) and those that, even though they are also impure, span different subsets of the latent space (M_{37} and M_{39}).

Looking at the outdegree (i.e., the number of measurement variables a latent causes) in Figure 4(b) we find a median of four and a range from 2 to 20. The number of measurements shared by each pair of latent variables reveals further structure (Figure 5(b)). In particular, the incidence matrix representation of the latents corresponding to the block structure between approximately L_{105}–L_{145} reveals seven measurement variables that these latents mostly have in common, corresponding to four stress and three depression items. On the other side, 41% (roughly 74k) pairs of variables do not share any measurement variables. Such insights may be used to simplify models, e.g. by removing measurement variables that induce multiple latents, or to build subsets of "pure" measurement variables, in the sense that the resulting measurement subsets are caused by disjoint sets of latents[6].

Finally, we note that there is more structure to be explored in the minMCM and figures 4 and 5 , but that is beyond our present scope. Note that the type of structure analyzed here emerges only when considering an ECC (i.e. patterns in the UDG, which is an abstraction of the correlation matrix) and not from the correlation matrix itself—analogous to higher-moment statistics or higher-order logic.

Our findings are not inconsistent with previous analyses of this data set, as can be seen by their agreement with points (i) and (iii) in Section 5.1, and should rather be seen as complementary. More generally our algorithm and corresponding analyses do not subsume existing methods but rather provide a novel perspective that allows us to focus on otherwise unutilized structure in measurement data, which in addition to helping to model the data also aids in, e.g., assessing and revising questionnaires and instruments.

6 DISCUSSION

Having in the preceding sections presented our minMCM finding algorithm, its supporting theory, and a demonstration application, we now conclude with two main directions for future work: the first direction is primarily concerned with applications of Algorithm 1 in its current state or requiring only minor modifications, while the second is primarily concerned with significantly extending

[6]Note that this is a bit different from the notion of "pure" used in the other measurement literature

Algorithm 1 and with developing new methods based on insights gleaned during its development.

6.1 FUTURE APPLICATIONS AND MINOR MODIFICATIONS

Being constraint-based, the Algorithm 4 relies on estimated independencies. Thus, errors in the inference of minMCMs come not from Algorithm 1 itself but rather from the estimation of independencies that it (along with many other causal inference methods) requires as input. In this regard, a single incorrectly estimated independence can in the (unlikely) worst case[7]result in incorrectly doubling or halving the number of estimated latents or edges. In any case, as mentioned at the end of Section 2, further estimates of conditional independencies can help corroborate or refute the estimated unconditional independencies. More detailed examination is needed to make this more theoretically precise as well as to determine how much of a problem this is likely to pose for real data.

One final caveat for interpreting minMCMs is that, for complex graphs, there can be multiple minimum ECCs (for both types of minimality), each with the same minimum number of cliques or assignments. Thus, while using a minMCM to reason about the minimum number of edges or latents is always valid, stronger conclusions may require that the graph $\mathrm{D}(\mathbf{M})$ admits only one minMCM (which is simple enough to test) or that further assumptions or background knowledge are used to justify one minMCM over other observationally consistent ones. To this end, the (non-minimal) MCM corresponding to maximal cliques (e.g., Figure 2(a)) may be especially interesting, because it contains all observationally consistent MCMs (including the minMCMs in 2(c) and 2(d)).

Another promising aspect of our approach for future work is its extensibility, which results from establishing MeDIL causal models as a causal semantics for edge clique covers. Though we have so far focused on minimal ECCs, a MCM corresponding to *any* ECC for a given UDG is guaranteed to be measurement-faithful and causally sufficient (though not minimal or measurement-Markov) for the corresponding distribution of measurement variables. Using a different class of ECCs simply requires a different algorithm to be used in Line 2 of Algorithm 1. Just as we expressed simplicity of the causal model in terms of the number of latents (or edges) in the MCM and therefore the number of cliques (or assignments) in the ECC, *any* property of a causal model that can be expressed in terms of properties of an ECC can be used to repurpose an ECC-finding algorithm for the desired CSL task. For example, developments in network science (Conte et al., 2019) make it possible for ECC-based causal analysis of very large graphs, even containing up to millions of nodes.

[7]This is when the inclusion/exclusion of a single edge in an $n \geq 3$ vertex undirected dependency graph makes the difference between the graph having $2(n-2)$ maximal cliques that are all edges and $n-2$ maximal cliques that are all triangles. Fortunately, such precarious structures are easy to detect and can be removed by picking different sets of measurements.

6.2 EXTENSIONS AND FURTHER DEVELOPMENTS

Because Algorithm 1 returns a causally sufficient DAG, it should be possible to actually learn a corresponding fully specified functional causal model using, e.g., some version of nonlinear ICA or variational autoencoders (Khemakhem et al., 2019) that has been modified to take into account the conditional independence structure. This could potentially lead to the development of a causal, non-parametric generalization of factor analysis (Martin and VanLehn, 1994) which would still be interestingly different from similar existing work (Hoyer et al., 2008; Kummerfeld and Ramsey, 2016). Furthermore, since learning such a FCM would require the data set and not just its CI relations, it would be straight-forward to make a score-based adaptation of Algorithm 1 inspired by (Elidan et al., 2001), where cliques are picked according to maximizing a scoring criterion instead of (possibly misestimated) CI relations. This would help overcome the potential pitfall mentioned in Section 6.1.

Additionally, notice that formally, (though not semantically) *every* DAG is a MCM: any given DAG $\mathcal{G}$ can be partitioned into sink nodes $\mathbf{S}$ and non-sink nodes $\mathbf{N}$, in which case it is observationally consistent with respect to $\mathbf{S}$ to any other DAG $\mathcal{H}$ whose (sub)set of sink nodes $\mathbf{S}'$ has the same UDG as $\mathbf{S}$. This allows for some of the theory developed in sections 2 and 3 to be easily repurposed to characterizing subset-Markov equivalence classes for DAGs with different sets of variables, as long as they have some subset of sink nodes $\mathbf{S} = \mathbf{S}'$ in common. This may help connect causal coarsening (Chalupka et al., 2016) with causally consistent transformations between micro- and macro-models (Rubenstein et al., 2017) and causal abstraction (Beckers and Halpern, 2019).

References

Andersson, S. A., Madigan, D., and Perlman, Michael D, e. a. (1997). A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541.

Beckers, S. and Halpern, J. Y. (2019). Abstracting causal models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2678–2685.

Bron, C. and Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577.

Chalupka, K., Eberhardt, F., and Perona, P. (2016). Causal feature learning: an overview. *Behaviormetrika*, 44(1):137–164.

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554.

Conte, A., Grossi, R., and Marino, A. (2019). Large-scale clique cover of real-world networks. *Information and Computation*, 270:104464.

Cygan, M., Pilipczuk, M., and Pilipczuk, M. (2016). Known algorithms for edge clique cover are probably optimal. *SIAM Journal on Computing*, 45(1):67–83.

Dwass, M. (1957). Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, 28(1):181–187.

Elidan, G., Lotner, N., Friedman, N., and Koller, D. (2001). Discovering hidden variables: A structure-based approach. In *Advances in Neural Information Processing Systems*, pages 479–485.

Ennis, J. M., Fayle, C. M., and Ennis, D. M. (2012). Assignment-minimum clique coverings. *Journal of Experimental Algorithmics*, 17(1):1.1.

Erdős, P., Goodman, A. W., and Pósa, L. (1966). The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18:106–112.

Gramm, J., Guo, J., Hüffner, F., and Niedermeier, R. (2009). Data reduction and exact algorithms for clique cover. *Journal of Experimental Algorithmics*, 13:2.2.

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. *Algorithmic Learning Theory*, page 63–77.

Holland, P. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960.

Hoover, K. D. (2006). Causality in economics and econometrics. *SSRN Electronic Journal*.

Hoyer, P. O., Shimizu, S., Kerminen, A. J., and Palviainen, M. (2008). Estimation of causal effects using linear non-gaussian causal models with hidden variables. *International Journal of Approximate Reasoning*, 49(2):362–378.

Khemakhem, I., Kingma, D. P., and Hyvärinen, A. (2019). Variational autoencoders and nonlinear ica: A unifying framework. *arXiv preprint arXiv:1907.04809*.

Kindermann, R. and Snell, J. L. (1980). Ii. markov fields on graphs. *Contemporary Mathematics*, page 24–33.

Kummerfeld, E. and Ramsey, J. (2016). Causal clustering for 1-factor measurement models. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1655–1664. ACM.

Kummerfeld, E., Ramsey, J., Yang, R., Spirtes, P., and Scheines, R. (2014). Causal clustering for 2-factor measurement models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 34–49. Springer.

Martin, J. and VanLehn, K. (1994). Discrete factor analysis: Learning hidden variables in bayesian networks (technical report no. lrdc-onr-94-1). *LRDC, University of Pittsburgh: Pittsburgh, Pennsylvania*.

Parascandola, M. (2001). Causation in epidemiology. *Journal of Epidemiology & Community Health*, 55(12):905–912.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Pearl, J. and Verma, T. (1995). A theory of inferred causation. In *Studies in Logic and the Foundations of Mathematics*, volume 134, pages 789–811. Elsevier.

Rubenstein, P., Weichwald, S., Bongers, S., Mooij, J., Janzing, D., Grosse-Wentrup, M., and Schölkopf, B. (2017). Causal consistency of structural equation models. In *Proceedings of the Thirty-Third Annual Conference on Uncertainty in Artificial Intelligence (UAI 2017)*, page ID11.

Seth, A. K., Barrett, A. B., and Barnett, L. (2015). Granger causality analysis in neuroscience and neuroimaging. *Journal of Neuroscience*, 35(8):3293–3297.

Silva, R. and Scheines, R. (2005). Generalized measurement models. Technical report, Carnegie Mellon University.

Silva, R., Scheines, R., Glymour, C., and Spirtes, P. (2006). Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(Feb):191–246.

Spirtes, P. and Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT Press.

Spirtes, P., Meek, C., and Richardson, T. (1999). An algorithm for causal inference in the presence of latent variables and selection bias. *Computation, causation, and discovery*, 21:1–252.

Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794.

Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 255–270. Elsevier Science Inc.

2.2. The MeDIL Python Package

2.2.1. Synopsis

We present the MeDIL Python package, summarizing the three main steps to go from a raw data set to a MeDIL functional causal model: (1) independence testing, (2) structure learning, and (3) function learning. We describe the three different submodules for carrying out each of these steps (including using one of them to generate simulated data sets), as well as a fourth submodule for producing various plots, all with accompanying example code. Complete documentation and source code can be found at `https://medil.causal.dev`.

Complete bibliographic information

Markham, A., Chivukula, A., and Grosse-Wentrup, M. (2020). MeDIL: A python package for causal modelling. In Jaeger, M. and Nielsen, T. D., editors, *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, pages 621–624. PMLR

My contribution

- writing the software package (with help on the `functional_MCM` submodule)
- writing the paper

MeDIL: A Python Package for Causal Modelling

Alex Markham ALEX.MARKHAM@CAUSAL.DEV
Research Group Neuroinformatics, Faculty of Computer Science, University of Vienna

Aditya Chivukula
Department of Statistics, Ludwig Maximilian University of Munich

Moritz Grosse-Wentrup
Research Group Neuroinformatics, Faculty of Computer Science, University of Vienna
Research Platform Data Science @ Uni Vienna
Vienna Cognitive Science Hub

Abstract

We present the MeDIL Python package for causal modelling. Its current features focus on (i) nonlinear unconditional pairwise independence testing, (ii) constraint-based causal structure learning, and (iii) learning the corresponding functional causal models (FCMs), all for the class of measurement dependence inducing latent (MeDIL) causal models. MeDIL causal models and therefore the MeDIL software package are especially suited for analyzing data from fields such as psychometric, epidemiology, etc. that rely on questionnaire or survey data.

Keywords: causal modelling; Python; structure learning; latent variable model; nonlinear independence; edge clique cover; generative adversarial network.

1. Introduction

Markham and Grosse-Wentrup (2020) introduce *measurement dependence inducing latent* (MeDIL) causal models. These models have disjoint sets of (unobserved) latent variables and (observed) measurement variables. In order for a set of random variables to be considered measurement variables, it must satisfy the assumption of *strong causal insufficiency*, i.e., none of the measurement variables may (even indirectly) cause one another—thus, any probabilistic dependence between them must be mediated by latent causes. The assumption of strong causal insufficiency is especially applicable in settings such as psychometric instrument questionnaires, and MeDIL causal models can, for example, be thought of as a causally interpretable factor analysis.

Graphically, MeDIL causal models (MCMs) are represented as directed acyclic graphs with disjoint sets of vertices representing the latent and measurement variables, where the measurement variables are represented as sink vertices (i.e., have no outgoing edges). These MCMs can be inferred by sampling a set of measurement variables as follows:

1. perform (nonlinear) independence tests on samples to generate undirected dependency graph (UDG) over measurement variables
2. perform causal structure learning by applying an edge clique cover finding algorithm to the UDG, resulting in a graphical MCM
3. use generative adversarial networks to learn a functional MCM (i.e., learn the functional relations corresponding to edges in the to the graphical MCM)

See (Markham and Grosse-Wentrup, 2020) for more details, supporting theory, and related work for steps 1 and 2, and see (Chivukula et al., 2020) for those of step 3.

2. Features

MeDIL is a free/libre software package written in Python (Van Rossum and Drake, 2009) and makes extensive use of NumPy (Oliphant, 2006), which is required for all three submodules. For installation instructions, documentation, and examples, visit https://medil.causal.dev

We begin with all the necessary import statements and generating the sample data set:

```
# for making sample data
import numpy as np
from medil.examples import triangle_MCM
from medil.functional_MCM import gaussian_mixture_sampler
from medil.functional_MCM import MeDILCausalModel  # also used in step 3

# for step 1
from medil.independence_testing import hypothesis_test, dependencies, distance

# for step 2
from medil.ecc_algorithms import find_clique_min_cover as find_cm

# for step 3
from pytorch_lightning import Trainer
from medil.functional_MCM import uniform_sampler, GAN

# for visualization
import medil.visualize as vis
from medil.independence_testing import distance


# make sample data
num_latent, num_observed = triangle_MCM.shape

decoder = MeDILCausalModel(biadj_mat=triangle_MCM)
sampler = gaussian_mixture_sampler(num_latent)

input_sample, output_sample = decoder.sample(sampler, num_samples=10000)
np.save("measurement_data", output_sample)
```

2.1 Independence Testing

The independence_testing submodule performs permutation-based hypothesis testing using nonlinear distance correlation from the dcor package Carreño (2020).

```
# step 1: estimate UDG
p_vals, null_corr = hypothesis_test(output_sample.T, num_resamples=100)
```

```
dep_graph = dependencies(null_corr, 0.1, p_vals, 0.1)
# dep_graph is adjacency matrix of the estimated UDG
```

However, any other preferred way of acquiring the (unconditional) pairwise dependencies can be used, and the resulting UDG can be plugged directly into step 2.

2.2 Causal Structure Learning

The `ecc_algorithms` submodule provides an implementation of an algorithm for finding a clique-minimal edge clique cover of a given UDG. The result is an biadjacency matrix that provides the minimal number of latent variables and their connections to measurement variables. It only uses NumPy, though part of the implementation contains code adapted from NetworkX (Hagberg et al., 2008) for finding maximal cliques.

```
# step 2: learn graphical MCM
learned_biadj_mat = find_cm(dep_graph)
```

2.3 FCM Learning

Given a set of measurement samples and the causal structure learned in step 2, the `functional_MCM` submodule uses generative adversarial networks (GANs) with the maximum mean discrepancy (MMD) loss to learn a nonlinear functional causal model. It defaults to using a uniform distribution for latent variables and a normal distribution for the exogenous variables, but any prior can be specified. The GANs are built using PyTorch Lightning (Falcon, 2019).

```
# step 3: learn functional MCM
num_latent, num_observed = learned_biadj_mat.shape

decoder = MeDILCausalModel(biadj_mat=learned_biadj_mat)
sampler = uniform_sampler(num_latent)

minMCM = GAN("measurement_data.npy", decoder,
 ↪  latent_sampler=sampler, batch_size=100)
trainer = Trainer(min_epochs=1000)
trainer.fit(minMCM)
```

2.4 Visualizing and Evaluating Results

The `visualize` submodule uses Matplotlib (Hunter, 2007).

```
# confirm given and learned causal structures match
vis.show_dag(triangle_MCM)
vis.show_dag(learned_biadj_mat)

# compare plots of distance correlation values for given and learned
 ↪  MCMs
```

```
generated_sample = decoder.sample(sampler, 1000)[1].detach().numpy()
generated_dcor_mat = distance(generated_sample.T)

vis.show_obs_dcor_mat(null_corr, print_val=True)
vis.show_obs_dcor_mat(generated_dcor_mat, print_val=True)

# get params for learned functional MCM; replace '0' with any 'i' in {0, ..., 5} to get params for any corresponding M_i
print(decoder.observed["0"].causal_function)
```

3. Future Development

Immediate further development will consist of (1) integrating other measures of independence, such as the Hilbert-Schmidt Independence Criterion, and (2) implementing/integrating other exact and heuristic edge clique cover finding algorithms, e.g., for minimizing the number of functions in the MCM instead of the number of latents, or other partial or heuristic solutions for use on very large networks. Future development will depend on the direction of our theoretical causality work, but is likely to include clustering samples coming from mixtures of MCMs, and learning causally consistent transformations between micro- and macro-models.

References

C. R. Carreño. dcor: Distance correlation and related e-statistics in Python, 2020. URL https://dcor.readthedocs.io/.

A. Chivukula, A. Markham, B. Bischl, and M. Grosse-Wentrup. Learning MeDIL causal models using generative neural networks, 2020. URL https://causal.dev/files/chivukula_thesis.pdf. masters thesis.

W. Falcon. Pytorch lightning. *GitHub. See: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, and J. Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3): 90–95, 2007.

A. Markham and M. Grosse-Wentrup. Measurement dependence inducing latent causal models. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020. ISSN 2640-3498. URL http://www.auai.org/uai2020/proceedings/244_main_paper.pdf.

T. E. Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.

2.3. Causal Clustering

2.3.1. Synopsis

We consider the problem of clustering within structurally heterogeneous populations, i.e., those in which different samples are generated by different causal structures. We derive the *dependence contribution kernel* and prove the corresponding kernel space is isometric to a space of causal graphical models, meaning the kernel allows us to measure the similarity/distance between the generating causal structures of different samples without first having to explicitly learn their causal structures. This kernel can be used in a wide variety of existing clustering algorithms, facilitating a flexible and extensible solution to the clustering problem in structurally heterogeneous populations. We demonstrate the dependence contribution kernel on a real-world data set by clustering genes according to their latent transcription factor networks. A Python implementation of the kernel as well as the full source code for the real-world application can be found respectively at `https://causal.dev/code/dep_con_kernel.py` and `https://causal.dev/code/fibroblast_clustering.py` or in Sections A.1 and A.2 of the Appendix.

Complete bibliographic information

Markham, A. and Grosse-Wentrup, M. (2021). A Distance Covariance-based Kernel for Nonlinear Causal Clustering in Heterogeneous Populations. In *under review but available as arXiv e-print*, page arXiv:2106.03480

My contribution

- connected the covariance-based clustering problem to kernel methods and graph embeddings
- derived the theoretical results
- implemented the kernel and clustering algorithm, applied it to the data, and produced all plots
- wrote the paper (with help, especially in the introduction)

A Distance Covariance-based Kernel for Nonlinear Causal Clustering in Heterogeneous Populations

Alex Markham
Research Group Neuroinformatics,
Faculty of Computer Science,
University of Vienna.
alex.markham@causal.dev

Moritz Grosse-Wentrup
Research Group Neuroinformatics,
Faculty of Computer Science,
University of Vienna;
Research Platform Data Science @ Uni Vienna;
Vienna Cognitive Science Hub.

Abstract

We consider the problem of causal structure learning in the setting of heterogeneous populations, i.e., populations in which a single causal structure does not adequately represent all population members, as is common in biological and social sciences. To this end, we introduce a distance covariance-based kernel designed specifically to measure the similarity between the underlying nonlinear causal structures of different samples. This kernel enables us to perform clustering to identify the homogeneous subpopulations. Indeed, we prove the corresponding feature map is a statistically consistent estimator of nonlinear independence structure, rendering the kernel itself a statistical test for the hypothesis that sets of samples come from different generating causal structures. We can then use existing methods to learn a causal structure for each of these subpopulations. We demonstrate using our kernel for causal clustering with an application in genetics, allowing us to reason about the latent transcription factor networks regulating measured gene expression levels.

1 Introduction

Learning causal relationships from observational and experimental data is one of the fundamental goals of scientific research, and causal inference methods are thus used in a wide variety of fields. The resulting variety of applications nevertheless share some common difficulties, such as causal inference from complex time-series data (Eichler, 2012) or the underlying causal structure being obscured by unmeasured confounders (Greenland et al., 1999). Another common difficulty, especially for applications in the biological and social sciences, is causal inference from heterogeneous populations (Xie, 2013; Brand and Thomas, 2013)—addressing this difficulty is our main motivation.

In general terms, we understand a heterogeneous population to be one whose members are not adequately described by a single model but rather better described by a collection of models. Within our context of causal structure learning, this means a population is heterogeneous if some samples are generated by different causal structures—we call this *structural* heterogeneity. We note that there are other kinds of heterogeneity, such as that in samples generated by different joint distributions over the same causal structure, which are not the scope of this work.

A specific example of structural heterogeneity can be found in genetics: causal methods are used to learn the structure of gene regulatory networks (Emmert-Streib et al., 2012), and gene expression data from a single recording or experiment may include thousands of genes, many of which are involved in entirely different networks (Liu, 2015); thus, attempting to learn a single causal structure for all of the genes will obscure the fact that different sets of them have different structures.

Preprint. Under review.

The bulk of our work in this paper, and our main contribution, is to introduce the *dependence contribution kernel*, which facilitates a flexible and easily extensible approach to causal clustering: first perform clustering to identify structurally homogeneous subsets of samples, and then proceed with the actual learning task on each cluster. We prove that our kernel is a statistically consistent estimator of the similarity of the causal structures underlying different samples and can thus be used to find clusters that minimize structural heterogeneity for causal structure learning tasks. Furthermore, the kernel is derived from the distance covariance (Székely et al., 2007), imbuing it with the ability to detect nonlinear dependence. It can easily be used in a wide array of clustering algorithms, such as k-means, DBSCAN, spectral clustering, or any other method that analogously makes use of a similarity (or distance) measure between samples (Filippone et al., 2008).

The rest of the paper is organized as follows: We finish this section by discussing some of the most relevant related work from the causal inference and statistics literature. All of Section 2 is devoted to the theory underlying our dependence contribution kernel, including a comparison of the familiar product-moment covariance with the distance covariance (Section 2.1), defining an equivalence class of causal models with a convenient representation in the kernel space (Section 2.2), and the actual definition of our kernel and proofs of its relevant properties (Section 2.3). Next, in Section 3, we demonstrate causal clustering with the kernel on a heterogeneous gene expression data set, finding structurally homogeneous clusters for which we then learn latent causal measurement models, allowing us to reason about the different transcription factor networks responsible for regulating the measured gene expression levels. Finally, we conclude in Section 4 mentioning possible future work.

1.1 Related Work

Causal inference in heterogeneous populations sometimes refers to data-fusion (Bareinboim and Pearl, 2016), i.e., combining known homogeneous subpopulations and performing causal inference on the resulting heterogeneous population, or similarly, it can refer to meta-learning using known subpopulations (Sharma et al., 2019). Other times, it refers to estimating heterogeneous treatment effects (Xie et al., 2012; Athey and Imbens, 2015). However, in our case, the subpopulations are not known and we rather consider the problem of learning which samples come from which subpopulation, and these are differentiated according to structure instead of treatment effect.

Previous work on causal clustering has focused more on the causal modeling aspect, using stronger assumptions about the underlying structures to learn more detailed models. For example, Kummerfeld et al. (2014); Kummerfeld and Ramsey (2016) focus on causal clustering in measurement models, with the goal of clustering different features together to study their latent causal structure, based on tetrad constraints within the linear product-moment covariance matrix. Huang and Zhang (2019) define a class of causal models facilitating mechanism-based clustering, learning causal models both for clusters of samples as well as a shared one for all samples, assuming the underlying structures are linear non-Gaussian. Saeed et al. (2020) characterize distributions arising from mixtures of directed acyclic graph (DAG) causal models (i.e., causal models without latent or selection variables), trying to learn both the component DAGs and a representation of how they are mixed. All of these approaches, like most causal inference methods, make specific (and for some applications, restrictive) assumptions about the underlying distributions or causal structures.

In contrast, our method is not tied to specific distributional assumptions such as linearity or (non)Gaussianity—we assume there are enough samples for statistical inference, as well as the usual causal Markov and faithfulness assumptions. For the first step, we cluster samples together if they (implicitly, in the kernel space) have similar nonlinear independence structures. For the second step, causal structure learning, any existing method (along with its corresponding assumptions) can in principle be used. In our gene expression data application (Section 3), the measurement dependence inducing latent (MeDIL) causal model framework (Markham and Grosse-Wentrup, 2020), which assumes the data consists of measurement variables that are causally connected only through latent variables, seems appropriate, however other applications can easily use other methods. For example, component and mixture DAGs (Saeed et al., 2020) can be better learned when one first knows which samples come from which component—clustering with our kernel ensures samples in different clusters come from different DAGs, and so using their method instead of the MeDIL framework would be a natural choice for applications in which a DAG (without any latents) is more appropriate.

Finally, there is some work from the statistics literature that sounds superficially similar to our distance covariance-based kernel but is conceptually quite different. Namely, another well-known

measure of nonlinear independence, the Hilbert-Schmidt Independence Criterion (Gretton et al., 2005, 2008), is part of a class of reproducing kernel Hilbert space- (RKHS-) based dependence measures that Sejdinovic et al. (2013) show is equivalent to distance-based measures such as the distance covariance. Our dependence contribution kernel, unlike these, is not a dependence measure between features—it rather uses the distance covariance to measure the similarity of samples based on patterns in the dependence structure of their features, and is rather more like a graph embedding (Cai et al., 2018).

2 Theory

2.1 Product-moment Covariance, Distance Covariance, and Dependence Contribution

Though there is more to causal relationships than probabilistic dependence, causal inference methods based on graphical models ultimately rely on at least implicitly learning conditional independence (CI) relations. CI relations can be estimated in many ways, with different dependence measures and tests each having their own theoretical guarantees and being better suited for distributions of various different kinds of data (e.g., categorical, discrete, or continuous) and with various kinds of relationships (e.g., linear, monotonic nonlinear, arbitrary nonlinear) and with different testing assumptions (see Tjøstheim et al., 2018, for a comprehensive overview).

A widely used measure of dependence is the *product-moment covariance*, often just called covariance, which is defined for two zero-mean random variables X_1 and X_2 as the scalar value $\mathrm{cov}(X_1, X_2) = \mathrm{E}[X_1 X_2]$. This can be extended from a pair of random variables to every pair of variables in a random vector, thus returning a matrix instead of a scalar. The covariance matrix for a vector of zero-mean random variables $\mathbf{X} = (X_1, \ldots, X_m)$ can be estimated from a set $S \in \mathbb{R}^{n,m}$ of n samples as $\hat{\Sigma}_{\mathbf{X}} = \frac{1}{n} S^\top S$, and the j, j'-th value of $\hat{\Sigma}_{\mathbf{X}}$ is thus the estimate $\hat{\mathrm{cov}}(X_j, X'_j)$.

Two random variables being probabilistically independent (denoted $\perp\!\!\!\perp$) implies that their product-moment covariance is zero, i.e., $X_j \perp\!\!\!\perp X_{j'} \implies \mathrm{cov}(X_j, X_{j'}) = 0$ (importantly, the inverse of this does not hold). Thus, the estimated product-moment covariance can be used in statistical hypothesis testing for probabilistic independence (Wasserman, 2013, Ch. 10): X_j and $X_{j'}$ are assumed to be independent if and only if $\hat{\mathrm{cov}}(X_j, X_{j'})$ is sufficiently close to 0. However, this method has an important problem: the product-moment covariance is only a valid test statistic against *linear* dependence.

Székely et al. (2007) introduce the *distance covariance* to remedy this problem: random variables are probabilistically independent if and only if their distance covariance is zero, i.e., $X_j \perp\!\!\!\perp X_{j'} \iff \mathrm{dCov}(X_j, X_{j'}) = 0$, resulting in the estimated distance covariance being a valid test statistic against all types of dependence. The distance covariance is related to the product-moment covariance by $\mathrm{dCov}^2(X_j, X_{j'}) = \mathrm{cov}(|X_j - X'_j|, |X_{j'} - X'_{j'}|) - 2\mathrm{cov}(|X_j - X'_j|, |X_{j'} - X''_{j'}|)$, where $(X'_j, X'_{j'})$ and $(X''_j, X''_{j'})$ are independent and identically distributed (iid) copies of $(X_j, X_{j'})$ (Székely and Rizzo, 2014). The key intuition here is that the distances (e.g., $|X_j - X'_j|$) constitute a nonlinear projection, so that using the linear product-moment covariance in this projected space allows for the detection of nonlinear dependence in the original space.

Note that dCov is typically defined to be a scalar value when taken between two arbitrary-dimensional random vectors, but our restricted presentation of it above in terms of random variables is to make it more obviously analogous to the product-moment covariance between random variables. Thus, corresponding to $\hat{\Sigma}_{\mathbf{X}}$ for random vectors, we define the following:

Definition 1 Let $S \in \mathbb{R}^{n,m}$ be a set of n samples from the vector of random variables $\mathbf{X} = (X_1, \ldots, X_m)$. For each $j \in \{1, \ldots, m\}$ and $i, i' \in \{1, \ldots, n\}$, define the pairwise distance matrix D^j, with values given by $D^j_{i,i'} := |S_{i,j} - S_{i',j}|$. Now define the corresponding doubly-centered matrices $C^j_{i,i'} := D^j_{i,i'} - \bar{D^j}_{i,\cdot} - \bar{D^j}_{\cdot,i'} + \bar{D^j}_{\cdot,\cdot}$, where putting a bar over the matrix and replacing an index i or i' with $\cdot$ denotes taking the mean over that index. Define the matrix $L \in \mathbb{R}^{n^2,m}$ so that each column is a flattened doubly-centered distance matrix, $L := (\mathrm{vec}(C^1), \ldots, \mathrm{vec}(C^m))$, where $\mathrm{vec}(C^j)$ denotes "flattening" matrix C^j into a column vector. Finally, the estimated *distance covariance matrix* over sample S is defined as

$$\hat{\Delta}_{\mathbf{X}} := \frac{1}{n^2} L^\top L.$$

Analogous to $\hat{\Sigma}_{\mathbf{X}}$, the j, j'-th entry of $\hat{\Delta}_{\mathbf{X}}$ corresponds to $\widehat{\mathrm{dCov}}^2(X_j, X_{j'})$—indeed it is mathematically equivalent to computing each pairwise distance covariance value and then manually filling in the matrix. The novelty of our Definition 1 is in finding a matrix of pairwise values instead of a single value for the distance covariance between random vectors, which helps provide an intuition for our next definition:

Definition 2 Let $S \in \mathbb{R}^{n,m}$ be a set of n samples from the vector of random variables $\mathbf{X} = (X_1, \ldots, X_m)$; note that we consistently use indices $i, i' \in \{1, \ldots, n\}$ and $j, j' \in \{1, \ldots, m\}$. Let $D \in \mathbb{R}^{n,n,m}$ denote the 3-dimensional array of stacked pairwise distance matrices defined by $D_{i,i',j} := |S_{i,j} - S_{i',j}|$, and use $C \in \mathbb{R}^{n,n,m}$ to denote these same distance matrices after being doubly-centered, i.e., $C_{i,i',j} := D_{i,i',j} - \bar{D}_{i,\cdot,j} - \bar{D}_{\cdot,i',j} + \bar{D}_{\cdot,\cdot,j}$, where replacing an index i or i' with $\cdot$ denotes the entire (lower-dimensional) subarray over that index, and writing a bar, $\bar{D}$, denotes taking the mean over that subarray. Then standardize the doubly-centered distances to get $Z_{i,i',j} := \frac{C_{i,i',j}}{\bar{D}_{\cdot,\cdot,j}}$. Finally, the *dependence contribution map*, $\varphi : \mathbb{R}^m \to \mathbb{R}^{m,m}$, is defined as

$$\varphi(S_{i,\cdot}) := Z_{i,\cdot,\cdot}^{\top} Z_{i,\cdot,\cdot} - \mathcal{T}(\alpha),$$

where $\mathcal{T}(\alpha) \in \mathbb{R}^{m,m}$ is a matrix of scaled critical values corresponding to a given significance level α with zeros along the diagonal, i.e., $\mathcal{T}(\alpha)_{j,j'} = \begin{cases} 0, & \text{if } j = j' \\ \frac{1}{n}\chi^2_{1-\alpha}(1), & \text{otherwise} \end{cases}$, with $\chi^2_{1-\alpha}(1)$ being the $1-\alpha$ quantile of the chi-square distribution with 1 degree of freedom.

Notice the similarity between Definitions 2 and 1: if we set $\mathcal{T}(\alpha)$ to be a matrix of 0s and forgo standardization (i.e., use C instead of Z), then $\frac{1}{n^2}\sum_{i=1}^{n} \varphi(S_{i,\cdot}) = \hat{\Delta}_{\mathbf{X}}$. Now, the differences: $\hat{\Delta}_{\mathbf{X}}$ is a single matrix computed over an entire set of samples, whereas φ is a map that projects each given sample to a new feature space; each entry of $\hat{\Delta}_{\mathbf{X}}$ is simply a distance covariance value, whereas each entry of the sum of $\varphi(S_{i,\cdot})$ over i, by using standardization (using Z instead of C) and subtracting a critical value, corresponds to the result of using a distance covariance value in a statistical hypothesis test for independence—indeed:

Lemma 3 Let $S \in \mathbb{R}^{n,m}$ be a set of n iid samples from random variables $X_1, \ldots, X_m$ with finite first moments. For a given significance level α, under the null hypothesis of $X_j \perp\!\!\!\perp X_{j'}$, the test

$$\text{reject } h_\emptyset \text{ if } \quad \big(\sum_{i=1}^{n} \varphi(S_{i,\cdot})\big)_{j,j'} > 0$$

is statistically consistent against all types of dependence.

Proof. This follows from (Székely and Rizzo, 2009, Theorem 5 and Corollary 2) and how φ is defined to correspond to the difference between distance covariance and critical values. □

These differences between $\hat{\Delta}_{\mathbf{X}}$ and φ serve two important purposes: first, they ensure φ maps to a Hilbert space so that our Definition 9 is a corresponding kernel function (Schölkopf et al., 2001); and second, as the name "dependence contribution map" suggests, they ensure $\varphi(S_{i,\cdot})$ is informative not just about distance covariance but about nonlinear dependence and about how the inclusion of sample $S_{i,\cdot}$ in a set of samples S contributes to the dependence patterns estimated from S— this is the key intuition behind how our kernel function is used to learn structurally homogeneous sample subsets, as explicated in the following sections.

2.2 Causal Graphs in Kernel Space

In general, a full causal structure can only be learned with sufficient data about the effects of interventions, and thus causal structure learning from purely observational data is usually possible only up to an equivalence class of causal graphs (Spirtes et al., 2000; Pearl, 2009). For example, the classic PC and IC algorithms, under the assumptions of no selection bias and no confounding by latent variables, do not necessarily return a fully-specified DAG but instead return a mixed graph, containing possibly directed and undirected edges, representing the Markov equivalence class (Spirtes and Glymour, 1991; Pearl and Verma, 1995).

We now define a set of equivalence classes for ancestral graphs (AGs), which—unlike causal DAGs—do not assume the absence of selection bias and latent confounders (Richardson et al., 2002):

Definition 4 Consider an arbitrary ancestral graph $\mathcal{A}$ with the set of vertices $V^{\mathcal{A}}$ and edge function $E^{\mathcal{A}}$, and denote the set of unconditional m-connection statements entailed by their corresponding unique maximal ancestral graph as $M^{\mathcal{A}} = \{(j, j') : j \not\perp\!\!\!\perp_m j' \mid \emptyset\} \subseteq V^{\mathcal{A}} \times V^{\mathcal{A}}$. For any ancestral graph $\mathcal{A}'$ such that $V^{\mathcal{A}'} = V^{\mathcal{A}}$, define the *unconditional equivalence* relation denoted by '$\sim_{\mathrm{U}}$' as

$$\mathcal{A} \sim_{\mathrm{U}} \mathcal{A}' \quad \text{if and only if} \quad M^{\mathcal{A}} = M^{\mathcal{A}'}.$$

Lemma 5 This lemma has two parts: (i) the relation $\sim_{\mathrm{U}}$ is an equivalence relation over the set of ancestral graphs $\mathbb{A}$; (ii) for an arbitrary ancestral graph $\mathcal{A} \in \mathbb{A}$, the bidirected graph $\mathcal{U}^{\mathcal{A}} = (V^{\mathcal{A}}, E^{\mathcal{U}})$, where $E^{\mathcal{U}}$ maps all pairs $(j, j') \in M^{\mathcal{A}}$ to the bidirected edge symbol '$\leftrightarrow$', is a unique *representative* of the equivalence class $[\mathcal{A}]$.

Proof. For (i), recall that an equivalence relation is any relation satisfying reflexivity, symmetry, and transitivity (Devlin, 2003), all of which are satisfied by $\sim_{\mathrm{U}}$ because of its correspondence to the relation '$=$' between sets. Thus, to prove (ii), it suffices to show that the map $s : \mathbb{A}/\!\sim_{\mathrm{U}} \to \mathbb{A}, [\mathcal{A}] \mapsto \mathcal{U}^{A}$ is injective (i.e, that it is a *section*) and that $[s([\mathcal{A}])] = [A]$ (Mac Lane, 2013). The key to the proof is the observation that $\mathcal{U}^{\mathcal{A}}$, because it contains only bidirected edges, is maximal and therefore entails exactly the unconditional m-separation statements $M^{\mathcal{A}}$, thus by (i) we have $\mathcal{U}^{\mathcal{A}} \sim_{\mathrm{U}} \mathcal{A}$ or equivalently $\mathcal{U}^{\mathcal{A}} \in [\mathcal{A}]$ or equivalently $[\mathcal{U}^{\mathcal{A}}] = [\mathcal{A}]$. Let $\mathcal{A}, \mathcal{A}'$ be arbitrary AGs, and assume $s([\mathcal{A}]) = s([\mathcal{A}'])$. Then by definition of s we have $\mathcal{U}^{\mathcal{A}} = \mathcal{U}^{\mathcal{A}'}$, and by the observation above, $\mathcal{U}^{\mathcal{A}} \in [\mathcal{A}']$ and thus $[\mathcal{A}] = [\mathcal{A}']$, making s injective. And finally, by the definition of s and also by the observation above, $[s([\mathcal{A}])] = [\mathcal{U}^{A}] = [A]$, completing the proof. □

This equivalence relation and its representatives has some important but perhaps subtle properties. First, it is different from Markov equivalence over AGs (which is characterized by partial ancestral graphs, PAGs) (Zhang, 2007)—it uses only unconditional m-separation while PAGs are learned from conditional m-separation statements. Second, because all DAGs are AGs, $\sim_{\mathrm{U}}$ is also an equivalence relation over DAGs. Third, being a representative means that every equivalence class includes exactly one fully bidirected graph (along with other equivalent AGs). Fourth, because each representative is formed by considering m-connected paths, $\mathcal{U}^{\mathcal{A}}$ is not equivalent to what would be generated by some "edge-wise" procedure, such as simply replacing every edge in a PAG/AG/DAG/Markov random field/moralized DAG with bidirected edges—note that Markham and Grosse-Wentrup (2020) also explore fully bidirected ancestral graphs, however they explore these graphs not as equivalence classes of AGs but rather as specific measurement models. Finally, its most important property is that it facilitates Theorem 8, for which we first need a few more definitions.

Definition 6 Given arbitrary ancestral graphs $\mathcal{A}, \mathcal{A}' \in \mathbb{A}$ over the same set of vertices, define the *Hamming similarity product*, denoted '$\bullet$' as

$$\bullet : \mathbb{A} \times \mathbb{A} \to \mathbb{A} \quad \text{and} \quad \mathcal{A} \bullet \mathcal{A}' \mapsto \mathcal{H},$$

where $\mathcal{H} = (V^{\mathcal{A}}, E^{\mathcal{H}})$ and the function $E^{\mathcal{H}}(j, j') =$ '$\leftrightarrow$' if and only if $E^{\mathcal{A}}(j, j') = E^{\mathcal{A}'}(j, j')$.

In words, the Hamming similarity product between two ancestral graphs returns a fully bidirected graph, with edges only where the two graphs have the same edge type. Now, shifting from ancestral graphs to real-valued square matrices:

Definition 7 Let '$\sim_{\mathrm{O}}$' denote the *orthant equivalence* relation ('orthant' is the generalization of 'quadrant' from $\mathbb{R}^2$ to arbitrarily higher dimensions) in square real matrices, i.e., for matrices $Y, Y' \in \mathbb{R}^{m,m}$ and with the element-wise function $\operatorname{sign}(Y)_{j,j'} = \begin{cases} 1, & \text{if } Y_{j,j'} > 0 \text{ or } j = j' \\ -1, & \text{otherwise} \end{cases}$,

$$Y \sim_{\mathrm{O}} Y' \quad \text{if and only if} \quad \operatorname{sign}(Y)_{j,j'} = \operatorname{sign}(Y')_{j,j'} \text{ for all } j, j'.$$

Theorem 8 Let a be the map from the set of unconditional equivalence classes over ancestral graphs with m vertices, $\mathbb{A}^m/\!\sim_{\mathrm{U}} = \mathbb{U}^m$, to the set of orthant equivalence classes over the image of φ, i.e., $m \times m$ symmetric real matrices with positive diagonal entries, $\varphi(\mathbb{R}^m)/\!\sim_{\mathrm{O}} = \mathbb{O}^m$, defined by $a : \mathcal{U} \mapsto O$, where $O_{j,j'} = \begin{cases} 1, & \text{if } E^{\mathcal{U}}(j, j') = \text{'}\leftrightarrow\text{'} \text{ or } j = j' \\ -1, & \text{otherwise} \end{cases}$. Then a is a group isomorphism between $(\mathbb{U}^m, \bullet)$ and $(\mathbb{O}^m, \odot)$, where '$\odot$' denotes the element-wise product.

Proof. First, note that $(\mathbb{U}^m, \bullet)$ is indeed a group, satisfying the three group axioms (Artin, 2011): the representative of its identity element is the fully connected bidirected graph over m vertices, $\mathcal{U}^1$;

each element is its own inverse; and $\bullet$ is associative. Likewise, $(\mathbb{O}^m, \odot)$ is a group with identity element $[\mathbb{1}^{m,m}]$, each element its own inverse, and the associative element-wise product operator.

Now, to show the two groups are isomorphic, it suffices to show (i) that a is bijective and (ii) that for arbitrary $\mathcal{U}, \mathcal{U}' \in \mathbb{U}^m$, $a(\mathcal{U}) \odot a(\mathcal{U}') = a(\mathcal{U} \bullet \mathcal{U}')$. For (i) notice that if $U \neq U'$, then there must be at least one pair of vertices j, j' such that $E^{\mathcal{U}}(j, j') \neq E^{\mathcal{U}'}(j, j')$ and thus clearly $O_{j,j'} \neq O'_{j,j'}$, so a in injective. Furthermore, notice that every distinct $O \in \mathbb{O}^m$ is the image of some graph $\mathcal{U}$, so a is also surjective. For (ii), for every $j, j' \in \{1, \dots, m\}$, the definitions of a, $\odot$, and $\bullet$ ensure $a(\mathcal{U})_{j,j'} \odot a(\mathcal{U}')_{j,j'} = 1 \iff E^{\mathcal{U}}(j, j') = E^{\mathcal{U}'}(j, j') \iff 1 = a(\mathcal{U} \bullet \mathcal{U}')$, completing the proof. □

In abstract less formal terms, [but still referring back to specific formal ideas]: there are two different spaces, [that of ancestral graphs and that of real square matrices]; we propose a way of transforming each space, [taking the quotient by its respective equivalence class]; then we describe a way of comparing members within each space, [using the respective products]; this induces a specific structure within each space, [that defined by each respective group]; and finally we show these structures are the same, [i.e., there is a group isomorphism between them].

For causal inference, which (often, but not necessarily) amounts to taking several samples in real space and inferring a single corresponding member in the space of ancestral graphs (or, more often, its quotient set by some equivalence relation), Theorem 8 means we can compare the different graphs of different sample sets without having to first move to the ancestral graph space.

Finally, notice the space of real square matrices is not a typical sample space but rather precisely (a superspace of) the space that our dependence contribution map φ (Definition 2) maps samples to—this means that mapping samples with φ allows us to make use of the group isomorphism. Though this already provides an intuition for why using φ would help with causal clustering, explicitly mapping each sample with it would be unnecessarily computationally expensive, and we are ultimately interested in morphisms between *metric spaces* (not just groups) of samples and graphs. To address this, we thus now move on to defining a kernel for φ.

2.3 The Dependence Contribution Kernel

Definition 9 Let $S, Z, \mathcal{T}$, and φ be as in Definition 2. We define the *dependence contribution kernel* using the Frobenius (denoted by the subscript $_{\mathrm{F}}$) inner product and norm:

$$\kappa(S_{i,\cdot}, S_{i',\cdot}) = \frac{\langle \varphi(S_{i,\cdot}),\, \varphi(S_{i',\cdot}) \rangle_{\mathrm{F}}}{\|\varphi(S_{i,\cdot})\|_{\mathrm{F}} \, \|\varphi(S_{i',\cdot})\|_{\mathrm{F}}}$$

A more convenient expression for applying the kernel to a data set is obtained by first defining a helper kernel, γ along with vec from Definition 1:

$$\begin{aligned}\gamma(S_{i,\cdot}, S_{i',\cdot}) &= \langle \varphi(S_{i,\cdot}),\, \varphi(S_{i',\cdot}) \rangle_{\mathrm{F}} \\ &= \left(\left(\mathrm{vec}(Z_{i,\cdot})^\top \mathrm{vec}(Z_{i',\cdot})\right)^2 - Z_{i,\cdot}\mathcal{T}Z_{i,\cdot}^\top - Z_{i',\cdot}\mathcal{T}Z_{i',\cdot}^\top + \|\mathcal{T}\|_2^2\right.\end{aligned}$$

This allows us to write

$$\kappa(s, s') = \frac{\gamma(S_{i,\cdot}, S_{i',\cdot})}{\gamma(S_{i,\cdot}, S_{i,\cdot})^{\frac{1}{2}} \gamma(S_{i',\cdot}, S_{i',\cdot})^{\frac{1}{2}}}$$

Finally, note that κ can be readily implemented on an entire set of samples, returning an entire Gram (kernel) matrix instead of a scalar value, by replacing the matrix operations above with tensor operations and specifying the correct axes along which summation occurs—an open source Python implementation can be found at `https://causal.dev/code/dep_con_kernel.py`.

A proper distance metric can also be obtained from this kernel through function composition: $\arccos \circ \kappa$. The key idea behind the kernel is that it is the cosine similarity in the space that φ maps to, meaning for arbitrary sample points x, x' it evaluates to $\cos(\theta)$, where θ is the angle between $\varphi(x)$ and $\varphi(x')$. In this space, θ represents the dissimilarity of the *dependence patterns* underlying x and x', without being biased by the possibly different magnitudes of $\varphi(x)$ and $\varphi(x')$ due to differing *variances*. Indeed, it can be used as a statistical test of whether samples come from different dependence structures and therefore causal models:

Theorem 10 Let $S \in \mathbb{R}^{n,m}$, $S' \in \mathbb{R}^{n',m}$ be sets of n, n' iid samples drawn respectively from the random variables $X = (X_1, \ldots, X_m)$ and $X' = (X'_1, \ldots, X'_m)$ with finite first moments. Then,

$$\sum_{i=1}^{n}\sum_{i'=1}^{n'} \kappa(S_{i,\cdot}, S'_{i',\cdot}) < 0 \implies \exists j, j' \in \{1, \ldots, m\} \text{ such that } \mathcal{I}(X_j, X_{j'}, \emptyset) \neq \mathcal{I}(X'_j, X'_{j'}, \emptyset).$$

Proof. Through Slutsky's Theorem (see Takeshi, 1985, Theorem 3.2.7) and the continuous mapping theorem (see Van der Vaart, 2000, Theorem 2.3), the consistency of φ (Lemma 3) guarantees the consistency of κ. Because the numerator of κ is a Frobenius inner product of φ,

$$\sum_{i=1}^{n}\sum_{i'=1}^{n'} \kappa(S_{i,\cdot}, S'_{i',\cdot}) \propto \sum_{i=1}^{n}\sum_{i'=1}^{n'}\sum_{j=1}^{m}\sum_{j'=1}^{m} \varphi(S_{i,\cdot})_{j,j'} \varphi(S'_{i',\cdot})_{j,j'}.$$

Thus, in order for $\sum_{i,i'} \kappa(S_{i,\cdot}, S'_{i',\cdot}) < 0$, there must be a j and j' for which $\varphi(S_{i,\cdot})_{j,j'} > 0$ but $\varphi(S'_{i',\cdot})_{j,j'} < 0$ (or vice versa), and thus the hypothesis test in Lemma 3 would reject the null hypothesis that $X_j \perp\!\!\!\perp X_{j'}$ but fail to reject that $X'_j \perp\!\!\!\perp X'_{j'}$. □

Corollary 11 Due to the relationship between independence structure and causal structure, an immediate of result of Theorem 10 is that $\sum_{i,i} \kappa(S_{i,\cdot}, S'_{i',\cdot}) < 0$ implies X and X' have different causal structures.

Theorem 12 Let d be the distance measure between unconditional equivalence classes of ancestral graphs over m vertices, $d(\mathcal{U},\mathcal{U}') = m^2 - |\{(j,j') : E^{\mathcal{U}\bullet\mathcal{U}'}(j,j') = \text{`}\leftrightarrow\text{'}\}| - m$. For given sample sets S, S' (i.e., real $n \times m$ matrices), use $\bar{\varphi}(S)$ to denote the mean of the sample in kernel space, $\sum_i \varphi(S_{i,\cdot})$, and say $S \sim_{\mathrm{K}} S'$ if and only if $\bar{\varphi}(S) \sim_{\mathrm{O}} \bar{\varphi}(S')$; denote the corresponding quotient set by this equivalence class as $\mathbb{R}^{n,m}/\sim_{\mathrm{K}} = \mathbb{K}^{n,m}$ and a representative from each equivalence class as $Q \in [S]$. Let δ be the distance between sets of samples in $\mathbb{K}$ defined as $\delta(Q, Q') = m^2 - \frac{1}{2n^2}\sum_{i,i'} \gamma(Q_{i,\cdot}, Q'_{i,\cdot})$. Let $b : \mathbb{U}^m \to \mathbb{K}^{n,m}, b : \mathcal{U} \mapsto \Omega$, where Ω is the unique element in $\mathbb{K}$ such that $\operatorname{sign}(\bar{\varphi}(\Omega)) = a(\mathcal{U})$. Then b is a distance-preserving map (i.e., an isometry) from the metric space $(\mathbb{U}^m, d)$ to $(\mathbb{K}^{n,m}, \delta)$.

Proof. Notice that $(\mathbb{U}^m, d)$ is indeed a metric space (Choudhary, 1993, Ch. 2): $d(\mathcal{U},\mathcal{U}') = 0$ iff $\mathcal{U}^{-1} \bullet \mathcal{U}'$ is the empty graph, which happens iff $\mathcal{U} = \mathcal{U}'$; the symmetry of d follows from the symmetry $\bullet$; and for subadditivity of d, observe that for vertices j, j' in arbitrary 2-vertex graphs $\mathcal{U},\mathcal{U}',\mathcal{U}$" we have either $d(\mathcal{U},\mathcal{U}") = 2$, in which case $d(\mathcal{U},\mathcal{U}') + d(\mathcal{U}',\mathcal{U}") = 4$, or we have $d(\mathcal{U},\mathcal{U}") = 0$, in which case $d(\mathcal{U},\mathcal{U}') + d(\mathcal{U}',\mathcal{U}")$ is either 0 or 4—in both cases $d(\mathcal{U},\mathcal{U}") \leq d(\mathcal{U},\mathcal{U}') + d(\mathcal{U}',\mathcal{U}")$; this easily extends to graphs of arbitrary numbers of vertices. Likewise, $(\mathbb{K}^{n,m}, \delta)$ is a metric space: $\delta(Q,Q') = 0 \iff \frac{1}{2n^2}\sum_{i,i'} \gamma(Q_{i,\cdot}, Q'_{i,\cdot}) = m^2 \iff \bar{\varphi}(Q)_{j,j'} = \bar{\varphi}(Q)_{j,j'}$, for all j, j', so iff $Q = Q'$; symmetry and subadditivity of δ follow from the symmetry and subadditivity of γ.

Finally, to show b is an isometry, we must show (i) that it is bijective and (ii) that for all $\mathcal{U},\mathcal{U}' \in \mathbf{U}^m$, $d(\mathcal{U},\mathcal{U}') = \delta(b(\mathcal{U}), b(\mathcal{U}'))$. For (i), observe that by the group isomorphism a and definition of b, we have $\mathcal{U} \neq \mathcal{U}' \implies a(\mathcal{U}) \neq a(\mathcal{U}') \implies Q \neq Q' \implies b(\mathcal{U}) \neq b(\mathcal{U}')$ and so b is injective. Also observe that because $\mathbb{K}$ is exactly the set of representatives of orthant equivalence classes of sample sets in kernel space, then for every $Q \in \mathbb{K}$, there exists a $\mathcal{U}$ such that $b(\mathcal{U}) = Q$, and so b is surjective.

For (ii), isomorphism a and the relation between element-wise product and Frobenius inner product allow us to write $d(\mathcal{U},\mathcal{U}') = m^2 - \sum_{j,j'}(O \odot O')_{j,j'} = m^2 - \langle O, O'\rangle_{\mathrm{F}}$. Substituting O, O' with their corresponding Ω, Ω', and because the Frobenius inner product is a sesquilinear form, we can write $d(\mathcal{U},\mathcal{U}') = m^2 - \frac{1}{n^2}\sum_{i,i'}\langle \varphi(\Omega_{i,\cdot}), \varphi(\Omega'_{i,\cdot})\rangle_{\mathrm{F}}$, which by Definition 10 finally gives us that $d(\mathcal{U},\mathcal{U}') = \delta(\Omega, \Omega')$, completing the proof. □

In less formal terms, Theorem 12 shows how the space of unconditional equivalence classes of ancestral graph corresponds to the space of real matrices, which is a common space for samples to lie in. More specifically, it shows how the structure defined by distances between graphs is the same as the structure defined by distances between sets of samples and how this sample distance is related to our kernel κ. Note that this is much stronger than Theorem 10: not only can κ tell us that two sets of samples come from different causal models, it gives a measure of just how different the causal models are, in terms of their differing unconditional nonlinear independencies/m-separation statements.

To summarize, we began by defining φ (Definition 2), which maps a given data set into a new higher-dimensional feature space. This feature space corresponds to a space of causal graphical

models, such that samples which are similar in the new feature space must come from similar causal models (Theorem 8). Our main contribution then is to propose the dependence contribution kernel κ (Definition 9).This kernel κ is guaranteed not only to tell us that two sets of samples come from different causal models (Theorem 10 and Corollary 11) but furthermore exactly how different the causal models are (Theorem 12), all without the computational expense of explicitly projecting samples or learning causal models. Thus, κ is well-suited for addressing the causal clustering problem and ensures that resulting clusters will be structurally homogeneous so that subsequent causal structure learning will be more informative.

3 Application

We use kernel k-means with our dependence contribution kernel to cluster a gene expression data set and then use the measurement dependence inducing latent (MeDIL) causal model framework for structure learning within each cluster (Markham and Grosse-Wentrup, 2020). The goal of causal clustering here is to reason about the different latent transcription factor (TF) networks governing gene expression (see Verny et al., 2017; Hackett et al., 2020, for other latent causal model approaches to learning TF networks). The original data set comes from Iyer (1999) and can be found at `genome-www.stanford.edu/serum/data/fig2clusterdata.txt`, with subsequent analysis by Dhillon et al. (2003, 2004). All of the Python code for our analysis is open source and available at `https://causal.dev/code/fibroblast_clustering.py`.

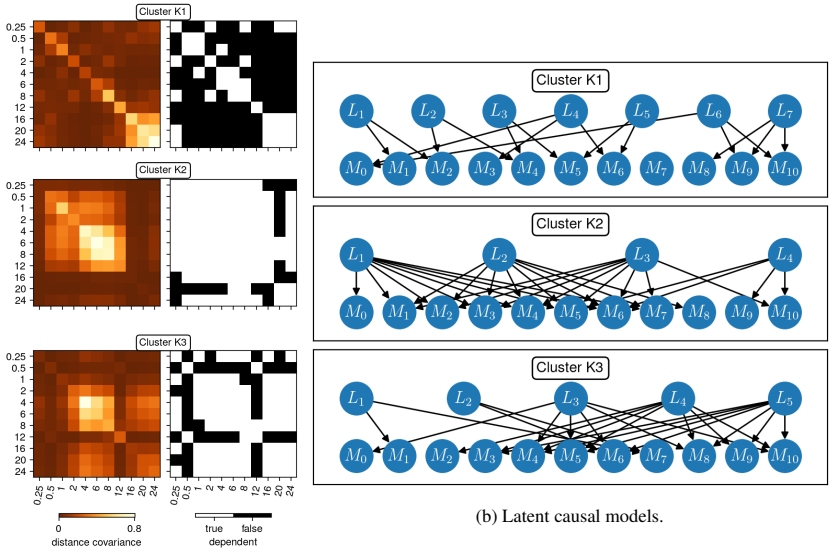
The data consists of the measured gene expression levels of 517 different genes from human fibroblast cells in response to serum exposure, measured at 11 different time points, i.e., there are 517 samples and 11 different features. In genetics applications, it is not unusual to consider genes to be samples and expression (over time) to be features—indeed the three previous analyses of this data all have this approach—and the intuition is simply that we wish to cluster genes based on patterns in their expression levels over time, in order to identify subsets of genes that are controlled by the same gene regulatory network. Also notice that such data exemplifies the structurally heterogeneous populations discussed in Section 1: different genes can of course be regulated by different TFs, and so we can better represent the data by first clustering it into subpopulations that are more homogeneous and then performing causal structure learning on each subpopulation.

For clustering, we used $k = 6$, which we found by looking at both the Variance Ratio Criterion (Caliński and Harabasz, 1974) and the Silhouette Coefficients (Rousseeuw, 1987), computed with the scikit-learn machine learning toolbox (Pedregosa et al., 2011). We implemented (unweighted) kernel k-means ourselves, using the pseudocode given by Dhillon et al. (2004), with initial mean points drawn uniformly at random from the sample set, and with significance level $\alpha = 0.1$ for the kernel parameter $\mathcal{T}(\alpha)$. We then used the `MeDIL` (Markham et al., 2020) package to learn the dependence structure and latent causal models for each cluster.

Figure 1 shows an example of our results for three of the six gene clusters: Figure 1a shows their distance covariance heatmaps and estimated nonlinear dependence structure with significance level $\alpha = 0.1$ (so the axes are the 11 different features, i.e. the time, in hours, at which gene expression level was measured), while Figure 1b shows their corresponding causal structures, with measurement variables M_0–M_{10} for each of the features and learned latent variables L for different posited TFs.

The results show a clear difference in causal structure for the different clusters and allow us to reason about the latent TFs regulating genes in different clusters: notice that the latents in cluster K1 each cause only two or three measurement variables that tend to be close together—e.g., L_1 causes M_1 and M_2, indicating the TF corresponding to L_1 is "short-acting", only affecting gene expression from 30 minutes (M_1) to 1 hour (M_2) after serum exposure; in contrast, the latents in cluster K3 each cause between two and seven measurement variables that tend to be more spread out—e.g., L_1 causes M_1 and M_7, indicating the corresponding TF is more complicated, "long-acting" but not continuously so, affecting gene expression 30 minutes (M_1) and 12 hours (M_7) after serum exposure, but independently of gene expression in the time between.

Our results are especially noteworthy compared what happens if one ignores the heterogeneity of the data and learns a causal structure for the entire data set without first clustering with our kernel into structurally homogeneous subpopulations: in that case, all of the measurement variables are dependent, with a single latent causing all of them, and no meaningful conclusions can be drawn



(a) Dependence structures.

(b) Latent causal models.

Figure 1: Results of dependence contribution kernel clustering with significance level $\alpha = 0.1$.

about how unmeasured transcription factors regulate measured gene expression, i.e., the heterogeneity obscures the underlying causal structures.

In summary, our causal clustering analysis reveals which subpopulations (clusters) of genes have similar latent TF networks as well as how the TF networks differ between clusters—information that is obscured when analyzing the structurally heterogeneous data set as a whole.

4 Discussion

We address the problem of causal clustering—that is, finding the different causal structures underlying a structurally heterogeneous data set. Our main contribution is to develop the *dependence contribution kernel* and prove its suitability for the causal clustering task. This allows us to first use the kernel with existing clustering methods, such as kernel k-means or DBSCAN, to identify homogeneous subpopulations. Then we use existing causal structure learning methods on each subpopulation. The kernel guarantees that each subpopulation is more structurally homogeneous and therefore the resulting causal structures better capture the causal structures within the data than if a single model were learned for the entire heterogeneous population.

Furthermore, we prove several interesting theoretical properties of our kernel, including (i) that it can be used as a statistical test for the hypothesis that two sets of samples come from different causal structures, as well as (ii) how it induces a metric space that is isometric to the one defined by Hamming distance between ancestral graphs, i.e., comparing sets of samples with our kernel is equivalent to first estimating the causal graphs of the different sets and then comparing those graphs. Beyond the practical applications of our kernel, as shown by our application in reasoning about latent transcription factor networks that regulate gene expression, this work also draws from and suggests further fruitful connections between a variety of fields, including causal inference, kernel methods, and algebraic statistics.

Acknowledgements

We thank Anja Meunier and Liam Solus for helpful discussions and comments on a previous draft.

References

Artin, M. (2011). *Algebra*. Pearson Prentice Hall.

Athey, S. and Imbens, G. W. (2015). Machine learning methods for estimating heterogeneous causal effects. *Stat*, 1050(5):1–26.

Bareinboim, E. and Pearl, J. (2016). Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352.

Brand, J. E. and Thomas, J. S. (2013). Causal effect heterogeneity. In *Handbook of Causal Analysis for Social Research*, pages 189–213. Springer.

Cai, H., Zheng, V. W., and Chang, K. C.-C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.

Choudhary, B. (1993). *The Elements of Complex Analysis*. New Age International.

Devlin, K. (2003). *Sets, functions, and logic: An introduction to abstract mathematics*. CRC Press.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means, spectral clustering and normalized cuts. *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '04*.

Dhillon, I. S., Marcotte, E. M., and Roshan, U. (2003). Diametrical clustering for identifying anti-correlated gene clusters. *Bioinformatics*, 19(13):1612–1619.

Eichler, M. (2012). Causal inference in time series analysis. *Wiley Series in Probability and Statistics*, page 327–354.

Emmert-Streib, F., Glazko, G., Gökmen, A., and De Matos Simoes, R. (2012). Statistical inference and reverse engineering of gene regulatory networks from observational expression data. *Frontiers in Genetics*, 3:8.

Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190.

Greenland, S., Pearl, J., and Robins, J. M. (1999). Confounding and collapsibility in causal inference. *Statistical Science*, 14(1).

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. *Algorithmic Learning Theory*, pages 63–77.

Gretton, A., Fukumizu, K., Teo, C., Song, L., Schölkopf, B., and Smola, A. (2008). A kernel statistical test for independence. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems 20*, pages 585–592. MIT Press.

Hackett, S. R., Baltz, E. A., Coram, M., Wranik, B. J., Kim, G., Baker, A., Fan, M., Hendrickson, D. G., Berndl, M., and McIsaac, R. S. (2020). Learning causal networks using inducible transcription factors and transcriptome-wide time series. *Molecular Systems Biology*, 16(3):e9174.

Huang, B. and Zhang, K. (2019). Specific and shared causal relation modeling and mechanism-based clustering. *Advances in Neural Information Processing Systems (NeurIPS)*.

Iyer, V. R. (1999). The transcriptional program in the response of human fibroblasts to serum. *Science*, 283(5398):83–87.

Kummerfeld, E. and Ramsey, J. (2016). Causal clustering for 1-factor measurement models. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1655–1664. ACM.

Kummerfeld, E., Ramsey, J., Yang, R., Spirtes, P., and Scheines, R. (2014). Causal clustering for 2-factor measurement models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 34–49. Springer.

Liu, Z.-P. (2015). Reverse engineering of genome-wide gene regulatory networks from gene expression data. *Current Genomics*, 16(1):3–22.

Mac Lane, S. (2013). *Categories for the working mathematician*, volume 5. Springer Science & Business Media.

Markham, A., Chivukula, A., and Grosse-Wentrup, M. (2020). `MeDIL`: A Python package for causal modelling. In *Proceedings of the 10th International Conference on Probabilistic Graphical Models (PGM)*. PMLR.

Markham, A. and Grosse-Wentrup, M. (2020). Measurement dependence inducing latent causal models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 590–599. PMLR.

Pearl, J. (2009). *Causality*. Cambridge University Press.

Pearl, J. and Verma, T. (1995). A theory of inferred causation. In *Studies in Logic and the Foundations of Mathematics*, volume 134, pages 789–811. Elsevier.

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

Richardson, T., Spirtes, P., et al. (2002). Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Saeed, B., Panigrahi, S., and Uhler, C. (2020). Causal structure discovery from distributions arising from mixtures of dags. In *International Conference on Machine Learning*, pages 8336–8345. PMLR.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426. Springer.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, pages 2263–2291.

Sharma, A., Gupta, G., Prasad, R., Chatterjee, A., Vig, L., and Shroff, G. (2019). MetaCI: Meta-learning for causal inference in a heterogeneous population. *CoRR*, abs/1912.03960.

Spirtes, P. and Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT Press.

Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794.

Székely, G. J. and Rizzo, M. L. (2009). Brownian distance covariance. *The Annals of Applied Statistics*, 3(4):1236–1265.

Székely, G. J. and Rizzo, M. L. (2014). Partial distance correlation with methods for dissimilarities. *The Annals of Statistics*, 42(6):2382–2412.

Takeshi, A. (1985). *Advanced econometrics*, volume 1. Harvard university press.

Tjøstheim, D., Otneim, H., and Støve, B. (2018). Statistical dependence: Beyond pearson's ρ. *arXiv preprint arXiv:1809.10455*.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press.

Verny, L., Sella, N., Affeldt, S., Singh, P. P., and Isambert, H. (2017). Learning causal networks with latent variables from multivariate information in genomic data. *PLoS computational biology*, 13(10):e1005662.

Wasserman, L. (2013). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

Xie, Y. (2013). Population heterogeneity and causal inference. *Proceedings of the National Academy of Sciences*, 110(16):6262–6268.

Xie, Y., Brand, J. E., and Jann, B. (2012). Estimating heterogeneous treatment effects with observational data. *Sociological Methodology*, 42(1):314–347.

Zhang, J. (2007). A characterization of markov equivalence classes for directed acyclic graphs with latent variables. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

3. Discussion

The three publications having been presented and their individual contributions made clear, we now discuss how they form integral parts of a coherent research program. We begin in Section 3.1 by returning once again to our research questions to see explicitly how they are answered by the publications, the limitations of these answers, and what new questions these answers spawn. Then, in Section 3.2, we discuss the ways in which the different publications are (perhaps non-obviously) linked, focusing in particular on their mathematical connections via undirected graphs as well as their shared conception of causality, motivated by specific aspects of the Neyman-Rubin framework along with the generality and expressiveness of the causal graphical model framework. Next, we present a number of promising but partially formed ideas for future research in 3.3, before finally concluding in Section 3.4.

3.1. Research questions answered

Q.1: To what extent can we learn causal models under strong causal insufficiency?

Ultimately, under the assumption of strong causal insufficiency, a causal model can be learned for a given data set up to observational consistency (Section 2.1, Definition 3). This is in line with the usual results of constraint-based causal structure learning being possible only up to the Markov equivalence class (MEC). Whereas in the case of causal sufficiency (or weak insufficiency), when the true causal model over observed variables is a DAG (or AG) and the MEC can be represented by a mixed graph (or PAG), in our strongly insufficient case, the true causal model over only observed variables can be described as an undirected graph (UG) with each edge representing dependence induced by a latent variable, and no two different UGs are Markov equivalent. However, we are interested not only in a causal model over the observed variables, which in the strongly insufficiency case are only effects, but also in their latent causes. Thus, we are interested not only in learning a UG but in learning a MCM with its explicit representation of latent variables and their causal relations to the observed effects.

Though every UG is its own MEC (over measurement variables), each UG represents an infinite equivalence class of observationally consistent MCMs, corresponding to all the possible ways of adding and connecting new latent variables without changing the dependencies they induce among the measurement variables. Thus, further constraints are needed to pick a specific MCM from the infinite class of MCMs observationally consistent with a given data set—for this, we turn to minimality (Section 2.1, Definition 5). In doing so, there are at least two reasonable notions of minimality, vertex-minimal and edge-minimal, respectively minimizing the number latent variables or edges from latents to measurements.

Whichever notion is chosen, the additional minimality assumption facilitates learning an ECC over the UG and thus a minMCM, a graphical causal model. This model can be

further refined into a FCM, learning specific distributions for the latents and functions describing their causal relations to the effects—code for this was developed and presented in Section 2.2, subsection 2.3), using generative adversarial networks as described in (Chivukula et al., 2020), a master's thesis extending Section 2.1.

In this way, UGs, minMCMs, and their corresponding FCMs provide an answer to Q.1. One limitation to this approach is the computational complexity of finding a minimal ECC required to specify a minMCM, which is known to be NP-complete. However, this does not seem to pose a practical problem even for graphs containing up to hundreds of measurement variables, which already surpasses the scale of human understanding and is thus more than adequate for the sorts of applications this method is best suited for. More specifically, we envision our approach being most useful to practitioners wanting to reason about the possible latent causes of their observed variables, possibly also refining their set of measurement variables, for example consisting of questions forming a diagnostic instrument, similarly to how factor analysis[1] is used (Thompson, 2004).

Unlike factor analysis, a purely statistical method for finding a number of factors capable of capturing the variation of a (usually larger) number of observed variables, a functional minMCM provides a minimal set of latent variables capable of causally inducing the dependencies among the observed effects. Whereas factor analysis is purely statistical (though often presented in covertly causal terms, such as "finding underlying factors to explain the observed variables"), MCMs explicitly represent causal structure and thereby facilitate causal reasoning. Interestingly, while the number of latent factors in a factor analysis is always less than or equal to the number of observed variables, dependence patterns among measurement variables in the minMCM case may require a greater number of latent causes. This renders minMCMs capable of detecting highly confounded questionnaires, i.e., using minMCMs as opposed to factor analysis makes it explicit when the number of latent causes must be greater than the number of observed effects, helping practitioners to identify when questionnaires capture too broad a range of latent causes as opposed to only the few specific causes of interest.

Unlike many other causal inference methods, such as using the PC algorithms to find a DAG, MCMs do not facilitate discovering new, direct causal relations among observed variables. Rather, the focus of the MCM is on identifying effects that share common causes. Thus, an interesting further question remains as to how MCMs could be used in an experimental (i.e., with interventions) setting to actually identify latents—we discuss some possibilities for this in Section 3.3.1.

[1] It is important to note that factor analysis, like the early statistical methods mentioned in Section 1.1.4, also has its roots in eugenics and racism. Factor analysis was introduced by Charles Spearman, who developed it while trying to measure human intelligence (Spearman, 1904). Spearman was involved in the eugenics movement (Wintroub, 2020) and stated himself in (Spearman, 1927, p. 8) that

> "an accurate measurement of everyone's intelligence would seem to herald the feasibility of selecting the better endowed persons for admission into citizenship—and even for the right of having offspring."

This sentiment, the use of intelligence tests, and the subsequent use of race as a proxy for intelligence (Stubblefield, 2007) led to the widespread forced sterilizations of people perceived as having mental disabilities, indigenous populations, and ethnic minorities throughout the United States, Canada, the United Kingdom, Australia, Sweden, Germany, and other countries (Pegoraro, 2015; Tydén, 2010; Garton, 2010; Black, 2012).

Q.2: . . . from heterogeneous populations?

Our approach to learning causal models from a heterogeneous population is to first partition it into homogeneous subpopulations and then proceed as usual with causal inference. We do this by defining a measure of causal structural homogeneity in the form of the dependence contribution kernel. This kernel works by implicitly projecting samples into a high-dimensional space in which (nonlinear) causal ancestral graphs have been embedded, so that distance between points in this space is isometric to the distance between their generating causal ancestral graphs. Thus, using this kernel to compute similarity or distance in standard existing clustering methods allows them to find clusters of samples generated by the same causal structure.

One limitation to this approach is that, by first partitioning the samples and then independently learning a model for each individual partition, the sample size and therefore statistical power of the learned models is less than what could otherwise be attained by, for each individual causal relation, using all samples sharing it (even if some samples sharing that causal relation differ in other causal relations, i.e., the generating graphs of the different partitions have some but not all edges in common).

Another possible limitation of this approach is that distance in the kernel space is isometric to distance between unconditional equivalence classes of AGs as opposed to fully-specified AGs that encode conditional independencies. However, interestingly, preliminary results from our simulation study shows that kernel k-means clustering with the dependence contribution kernel is indeed able to distinguish between individual AGs, even those belonging to the same UEC, though this is perhaps not too surprising considering samples generated by two random AGs in the same UEC are still likely to have detectably different distance correlation matrices (see Section 3.3.2 and Figure 3.1).

Q.3: . . . without assuming any particular distribution?

The methods developed to address Q.1 and Q.2 do so without assuming any particular distribution for the random variables in the causal models and without constraints (e.g., linearity) on the functions used to represent the causal relations. This is possible primarily because of (i) our focus on causal graphical models, which characterize probability distributions only in terms of patterns of conditional probabilistic independence, as opposed to more specific constraints or parameters, and thus (ii) our use of distance correlation, a general (nonlinear) measure of probabilistic dependence.

On the one hand, the generality of our approach makes it broadly applicable and less likely to lead to faulty causal reasoning as the result of unsatisfied distributional assumptions. On the other hand, distance covariance is more costly to compute than the more common product-moment covariance and our models are less detailed than is possible with parametric methods whose narrower assumptions are met.

3.2. Linking MeDIL causal models and the dep-con kernel

Though three separate manuscripts, the ideas underlying the three publications in Chapter 2 are connected in a variety of ways, some more obvious than others.

Their most obvious connection is demonstrated by the application in Section 2.3, where we use the kernel developed there to find clusters in a data set (presumably) satisfying the

assumption of strong causal insufficiency but which is structurally heterogeneous. The homogeneous clusters found thus admit an interpretation using minMCMs (Section 2.1), which we learn using the `MeDIL` software package (Section 2.2). Recall that (i) minMCMs are determined purely by their unconditional independence relations over the measurement variables and (ii) the dep-con kernel is defined with respect to unconditional equivalence classes over AGs. Under strong insufficiency and minimality assumptions, each equivalence class over AGs can be reduced to a single minMCM, i.e., all other AGs besides the minMCM in each class either contain variables that cause others (violating strong insufficiency) or they are more expressive in that they can induce a wider array of probability distributions (violating minimality). Thus, in this and similar applications, the dep-con kernel allows for even stronger interpretations as Corollary 11 (Section 2.3) becomes an "if and only if" instead of merely "if" in the DAG or more general AG case.

Their less obvious connections can be roughly divided into those that are mathematical and those that are conceptual, which we address respectively below.

3.2.1. Mathematically, in terms of undirected graphs

Whereas other causal methods typically use directed or mixed graphs, we primarily use undirected graphs. To better understand how we use UGs and how this relates to other methods, it is helpful to first review some properties of DAGs, AGs, their equivalence classes, and other undirected graphs. For more formal presentation of these topics and other graphical models, see (Koller and Friedman, 2009; Sadeghi and Lauritzen, 2014).

DAGs

Directed acyclic graphs of course contain only directed edges, but the essential graph corresponding to Markov equivalence classes of DAGs is a mixed graph possibly containing both directed and undirected edges. Undirected edges in essential graphs should be seen as a directed edge whose direction is unknown, i.e., they represent an immediate causal relation between two variables but where more information (such as by doing interventions) is needed to determine which is the cause and which is the effect in the relationship.

Another context in which undirected edges can arise when considering DAGs is in the process[2] for transforming a DAG representation of a probability distribution to an UG representation. In this case, the structure of the UG represents a more general class of probability distributions than that entailed by the structure of the DAG, including those that lead to causal conclusions in violation of the causal faithfulness and Markov assumptions. Hence, the edges of the UG lack a consistent causal interpretation and this process is used not as part of causal inference but merely as a way of rewriting the factorization of a known probability distribution.

AGs

Ancestral graphs are a kind of mixed graph capable of representing causal relations even in the presence of selection bias and confounding, and correspondingly have three edge types: directed, undirected, and bi-directed. Undirected edges in AGs thus represent the presence

[2] unfortunately known as "moralization", because it involves "marrying" (adding an edge between) two nodes if they share a child (both have directed edges to a common third node)

of selection bias obscuring the independence between the variables, i.e., they implicitly represent a third variable that is a collider between the two and that is being conditioned on as part of the data selection process to render the connected nodes dependent, so there is in actuality no causal relation between the two and no shared common cause.

Markov random fields

Whereas the previous two subsections dealt with interpreting undirected edges in mixed graphs, we now consider the most common fully undirected graph, known as a Markov random field (MRF) (Kindermann and Snell, 1980). Analogous to how d-separation in DAGs corresponds to probabilistic independence (through the CMA), separation statements in a MRF can encode conditional independence statements (through the Markov properties). However, no causal assumptions are used and there is no consistent causal interpretation of the edges, e.g., as to whether the encoded in/dependencies are the result of an immediate causal relationship (of perhaps unknown direction, as in the essential graph case) or a common confounding cause.

UDGs and UECs

In contrast to undirected edges in these other cases, undirected edges in the UDGs (undirected dependency graphs) of Section 2.1 and in the UECs (unconditional equivalence classes) of Section 2.3 always correspond to unconditional independence.

In the case of UDGs, the further assumptions of strong insufficiency and minimality lead to the full characterization of *conditional* independencies and the consistent causal interpretation of undirected edges as corresponding to a latent cause (Propositions 6 and 7, Section 2.1). This facilitates a causal interpretation of edge clique covers defined over UDGs and allows for the explicit representation of latents in the corresponding minMCM. Note that this interpretation of undirected edges in UDGs is similar to the interpretation of bidirected edges in AGs.

Indeed, relaxing the strong insufficiency and minimality assumptions and replacing undirected edges of the UDG with bidirected edges results in the UEC. This representation using undirected edges allows for a convenient embedding of the UEC into the kernel space: there being only one edge type, absence or presence of an edge can respectively be made to correspond to negative or nonnegative real numbers, leading to the isomorphism between Hamming similarity in graphs and element-wise product in real matrices and hence the isometry between the space of ancestral graphs and the space of real-valued samples (Theorems 8 and 12, Section 2.3).

Thus, in both the UDG and UEC, undirected graphs are given a novel uses and interpretations for causal inference through their representation of unconditional dependence. Their undirected edges differ in interpretation from those in DAG essential graphs, AGs, and MRFs, and they facilitate the use of edge clique covers and graph embeddings for causal inference.

3.2.2. Conceptually, in terms of foundational issues in causality

Both MCMs and the dep-con kernel address questions that can be traced back to the RCM (Rubin causal model) framework but by making use of and extending the graphical framework associated with FCMs (functional causal models). Though the FCM mathematically

subsumes the RCM, the less abstract conception of causality in the RCM leads to the emphasis of (1) the importance of distinguishing between possible causal mechanisms and measurements of their effects, as well as (2) the population as opposed to the typological perspective and the importance of considering homogeneity in causal learning tasks. At the same time, the more abstract graphical conception of causality in the FCM framework makes it possible (1) to discover causal relations, even when lacking the a priori knowledge needed to hypothesize different causes and effects, as well as (2) to characterize the relevant population homogeneity in explicit terms of causal structure.

In this way, the abstraction afforded by graphical methods facilitate the intuitive representation of causal relationships and a broader range of causal inference methods than is possible with the RCM, however this abstraction perhaps also makes it easier to overlook the assumptions that are critical for the sound use of such methods. It is thus important not only to be aware of these assumptions but also to make a variety of methods for different possible assumptions and applications, and it is in this light that we present (i) MCMs and the (ii) dep-con kernel: (i) compared to the (weak) causal insufficiency permitted in ancestral graphs, our insufficiency assumption is mathematically stronger (i.e., ancestral graphs are capable of representing all of the independencies among measurement variables in a UDG or MCM) but should rather be seen as just a different assumption suited for a different task (e.g., questionnaire data and similar measurement models in which causes are not directly observed) and with correspondingly different abilities (e.g., the explicit representation of latent causes over cliques of observed variables); (ii) by first using the dep-con kernel in clustering, other existing causal inference methods can then be used to learn more accurate models whose assumptions of structural homogeneity is thus satisfied.

3.3. Future Work

Though the MeDIL causal models and dependence contribution kernel of Sections 2.1 and 2.3 are more theoretical and focus on foundational issues, they hint at a number of interesting applications, extensions, and further developments, most interestingly with connections to recent work on causal consistency and algebraic statistics, which we speculate about in the following sections.

3.3.1. Extensions and applications of MCMs

MCMs for other kinds of ECCs

One of the main contributions of MCMs is in establishing a causal semantics for ECCs (edge clique covers) of undirected dependency graphs. In Section 2.1 we introduce a minimality constraint, making it possible to learn a minimal MCM from the infinite class of observationally consistent MCMs. However, the semantics holds for cliques in UDGs generally, not just minimal ECCs, hence there are other ways of selecting from this infinite class, namely by using a constraint other than minimality. For example, the exact ECC solution is NP-complete, scaling poorly to large numbers of nodes and cliques in the minECC solution, however much faster heuristic solutions exist, allowing for approximately minimal MCMs to be efficiently found for large scale real-world networks containing hundreds of thousands of nodes and edges Conte et al. (2020); Abdullah et al.

(2020). Another possibility is not to look for full ECCs but rather only partial ECCs (Agarwal and Mazumdar, 2016), e.g., connecting certain variables of interest, or looking for the maximal clique in the graph, representing the latent cause responsible for the most effects (for example Ding et al. (2008) give an algorithm for efficiently finding the maximal clique in large graphs).

MCMs with interventions

MCMs can be seen as the result of taking the RCM's cause/effect distinction seriously in the context of causal structure learning from observational data. Recall (Section 1.1.3) that learning a RCM from observational data has two important steps: (1) posit which variables are hypothesized to be causes and which are to be effects, and (2) collect or use other variables to perform matching to control for possible confounders. Analogously, learning a MCM involves: (1) identify measurement variables thought to be the effects of some latent causes (e.g., a psychiatric diagnostic instrument for depression) (2) use MCM to learn (e.g., minimal) causal structure connecting measurement variables, which helps guide the search for relevant intervention targets and latent identification.

Searching for intervention targets could take the form of, for example, prescribing a drug suspected to result in symptoms detectable among the measurement variables. Latent identification could take the form of an iterative process: For example, comparing the UDGs before and after an intervention, different independence patterns would indicate some latent structure interventionally consistent with both UDGs. Furthermore, by looking for changes in the actual post-intervention distributions, it is possible to associate latents in the minMCM with the performed intervention thus helping to identify the latent. This could be repeated until understanding of the (identified) latent structure is achieved or until the available interventions are exhausted.

MCMs for causal consistency

Chalupka et al. (2014) proves the causal coarsening theorem and provide an algorithm for constructing causal macro variables from (noncausal) observed micro-variables. This can be seen as finding a new representation for low-level, detailed data in terms of observed data, transforming it into a higher-level causal representation in terms of variables related to each other causally. Rubenstein et al. (2017) builds upon this, providing a definition and algorithm for causally-consistent transformations between micro- and macro-level SEMs (structural equation models, linear Gaussian FCMs). Beckers et al. (2020) builds upon this further, providing definitions, algorithms, and use-cases for various related kinds of causal abstraction (i.e., micro- to macro-level transformations between causal models).

All of these approaches focus on transformations with respect to a specific distributions (e.g., a fully specified SEM) and could perhaps benefit from a more graphical perspective. The key insight into how MCMs help provide this perspective is hinted at in the last paragraph of Section 2.1: mathematically, (but not semantically) *every* DAG is a MCM—any given DAG $\mathcal{G}$ can be partitioned into sink nodes $\mathbf{S}$ and non-sink nodes $\mathbf{N}$, in which case it is observationally consistent with respect to $\mathbf{S}$ to any other DAG $\mathcal{H}$ whose (sub)set of sink nodes $\mathbf{S}'$ has the same UDG as $\mathbf{S}$. This allows for much of the theory underlying MCMs to be easily repurposed to characterizing subset-Markov equivalence classes for DAGs with different sets of variables, as long as they have some subset of sink nodes $\mathbf{S} = \mathbf{S}'$

in common. Thus, our idea of graphical causal consistency is that two different causal models are causally consistent with respect to some shared measurement variables/effects when they are part of the same equivalence class of MCMs and causal consistency is a type of equivalence relation over graphs of different sizes. For example, given an arbitrary DAG, one can construct a UDG over the sink nodes (effects) of interest. Finding a minMCM over these nodes results in a new, most macro-level DAG capable of inducing probability distributions that are observationally consistent with those of the original DAG. Further investigation of this could focus on (i) extending this to the interventional setting (which is an important part of what makes the causal consistency of the above mentioned papers actually causal,) (ii) specifically describing how the distributions of the non-sink nodes in the original graph are related to those in the macro-model, and (iii) considering cases when the nodes of interest are not only sink nodes.

3.3.2. Extensions and applications of the dep-con kernel

Before considering various extensions and applications of the dep-con kernel κ, it is helpful to develop a better intuition for the kernel space and its properties, which we conceptually split into four parts: (1) the unnormalized kernel space, i.e., the space that φ maps to, the kernel space for γ before it is normalized into κ; (2) the role of the α parameter and correspondingly $\mathcal{T}(\alpha)$; (3) the kernel space of κ, the normalized γ; and finally (4) the one dimensional space that κ maps to

(1) The image of φ is related to the pairwise distance covariance matrix between features (say we have n of them), so it is symmetric, and we are interested in the covariances (so not the variances along the diagonal) and what they tell us about dependencies, thus it suffices to build an intuition for the $\binom{n}{2}$ subspace of the upper (or lower) triangular matrix without the diagonal. Importantly, notice that there are $2^{\binom{n}{2}}$ orthants of this space, the same as the number of possible undirected graphs over n vertices.

(2) The α parameter corresponds to a statistical significance level in a distance covariance test for independence, and $\mathcal{T}(\alpha)$ is the corresponding critical value, i.e., the point in the space of distance covariances that the probability specified it α gets mapped to. By subtracting $\mathcal{T}(\alpha)$ from the distance covariance computed by φ, we thus shift points lying in its image so that their mean in each dimension corresponds to whether the pair of features corresponding to that dimension would fail an independence test—i.e., the mean is greater than or equal to 0 if and only if the the null hypothesis of independence would be rejected. Not only does this allow the kernel space to encode unconditional independence—i.e., the 0 in each dimension is a threshold or sort of decision boundary between those points that contribute to a test result of independence and those contributing to dependence—but by encoding it as the sign of each dimension, the product of different points is also meaningful: the product of two points is positive if and only if they have the same result for the independence test.

(3) Normalizing this space to produce κ can be thought of as projecting all points in the space to the surface of the unit hypersphere. In doing so, we ignore the scaling of the critical-value-shifted-covariance values[3], so that they points in the space better correspond to dependence rather that shifted covariance. Also not that, being on the surface of

[3] though perhaps in a biased way—it may be better to instead base the kernel on distance correlation, dividing each feature by its distance variance, but doing so would add significantly to mathematical and computational complexity of the kernel

a hypersphere, distance between points should also be calculated along this surface, or equivalently, in terms of the angle between the points.

(4) Finally, the actual kernel value for two samples, lying in the one-dimensional space that κ maps to, provides a summed measure of similarity of the samples in each dimension. That is, for each dimension, we take the product of the two samples, thus increasing the summed measure when the samples have the same dependence and decreasing it when they differ (this is a result of the last clause of paragraph (2) above).

Error rates

Because κ is so explicitly related to statistical hypothesis testing, it should be possible to compute error rates, e.g., on the statement given in Theorem 10, Section 2.3, especially for thresholds other than 0. For example, consider the element-wise product of two points in the kernel space (note that summing the resulting matrix would lead to the actual κ value): the probability of the sign (measuring the similarity of the two points in that dependence dimension) being wrong is $2\alpha(1-\alpha) + 2\beta(1-\beta)$, corresponding to two times (because there are two samples) the probability of a type I error in the test for one sample but not the other plus two times the probability of a type II error for one sample but not the other. Next, one must figure out how this error rate for one element of the matrix combines with the other elements (the kernel sums these elements) along with the actual values of the element (the whole matrix is normed), and the result should be the error rate for Theorem 10. Similarly, it should be possible to compute error rates for other applications of the dep-con kernel.

In the case of clustering, simulation results[4] give us a hint of its performance compared to other kernels in clustering, with an error rate of around 0.17, as shown in Figure 3.1.

Other causal kernels

The theory and intuitions underlying the dep-con kernel rely on algebraic (e.g., the group isomorphisms used to establish the isometry of Theorem 12) and geometric (e.g., distance between points in the kernel space being arc length) insights. Thus, it may be of interest to and could benefit from the field of algebraic statistics, which uses methods from algebraic geometry to address questions in statistics (Sullivant, 2018). For example, for discrete random variables, conditional independence constraints can be encoded as vanishing constraints on quadratic polynomial equations in the joint probability distribution. This could be of particular interest for finding analogous constraints in the dep-con kernel space, extending it to incorporate conditional independencies instead of only unconditional ones.

Another potential way of extending the dep-con kernel to encode conditional independencies is through the use of non-redundant clustering methods[5] (Mautz et al., 2018; Niu et al., 2010), for example using spectral clustering and kernel k-means, both of which are can easily be used with our kernel. These methods find multiple different non-redundant clusterings of a given data set by defining some measure of redundancy and then simultaneously learning several subspaces that minimize the redundancy and then learning a clustering within each subspace. These subspaces can also be described in terms of conditioning on

[4]Special thanks to Richeek Das, our enthusiastic and talented undergraduate research intern for helping to run these simulations and produce the plots for Figures 3.1 and 3.2!

[5]Thanks to Lukas Miklautz for this idea.

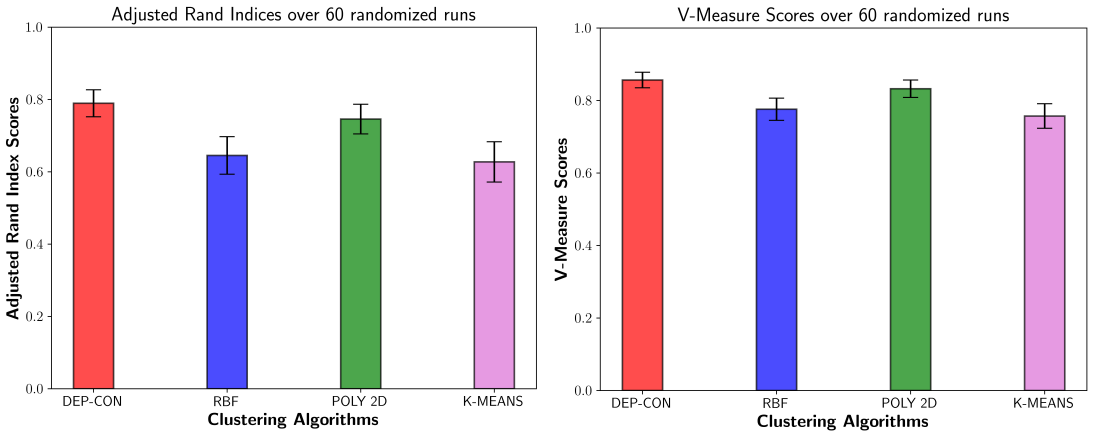


Figure 3.1.: Six random DAGs containing ten nodes each were randomly generated and parameterized as linear, Gaussian SEMs from which samples where drawn. Clustering was then performed using k-means, with the dep-con kernel, the radial basis function kernel, the degree-2 polynomial kernel (on normed data, thus computing the product-moment correlation), and finally with no kernel. Scores were averaged over 60 different repetitions of the above procedure, with error bars showing the standard deviation. On the left is the unsupervised score (e.g., what we used to pick k and other parameters for each clustering method on each data set) and on the right is the supervised score, showing how accurate the methods were in clustering samples from the same SEM together.

sets of features. Thus, it should be possible to use these subspaces along with the dep-con kernel to identify conditional as opposed to only unconditional independencies.

As opposed to extending the dep-con kernel, another option would be to use it as inspiration for other causally-interpretable kernels. The dep-con kernel works by implicitly embedding samples onto the surface of a hypersphere in a way that encodes independence relations. There may be other embeddings that also lend themselves to causal interpretations. For example, Studený et al. (2010) introduces the characteristic imset, an algebraic representation of Markov equivalence classes of DAGs as binary vectors (note that the dep-con kernel space essentially dose this for unconditional equivalence classes), as well as the characteristic imset polytope, whose vertices correspond to different MECs. A kernel capable of implicitly embedding samples into such a polytope would provide a measure of similarity between Markov equivalence classes of DAGs in the same way that the dep-con kernel measures similarity between unconditional equivalence classes over AGs.

Use in other kernel methods

Though we have so far discussed the dep-con kernel for use in clustering, it can also be used in other kernel methods or as a general measure of causal structural similarity of samples. For example, it can be used in kernel PCA to provide a better (in terms of causal structure) low-dimensional representation of data than usual PCA, as shown in Figure 3.2.

Another use for the kernel is in unsupervised clustering validation, similar to the Calinski-Harabasz or Davies-Bouldin indices Caliński and Harabasz (1974); Halkidi et al.

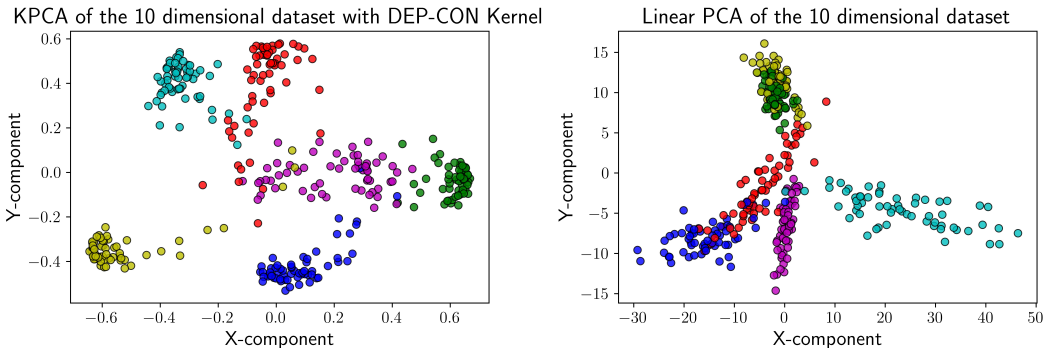


Figure 3.2.: A single data set was generated as described in Figure 3.1. On the left are the first two principal components resulting from the dep-con kernel PCA, and on the right are the first two for usual PCA. The dep-con kernel leads to clearer separation of samples from different DAGs in the low-dimensional representation.

(2001). For clusters C_1, C_2, clusterings minimizing $\kappa(C_1, C_2)$ while maximizing $\kappa(C_1, C_1)$ and $\kappa(C_2, C_2)$ would lead to heterogeneity between-clusters and structural homogeneity within-clusters, so using results in a causally-interpretable evaluation of clusterings.

Finally, the kernel could also be used in classification tasks for brain-computer interfaces—for example Xu et al. (2021) shows how the distance covariance matrix (essentially our kernel space) can be used as a drop-in replacement for existing methods using the product-moment covariance, and the dep-con kernel can be used to preform classification tasks more efficiently than explicit projection to the kernel space.

3.4. Conclusion

We began this dissertation with the three-part question: (Q.1) To what extent can we learn a model of a causal system *while observing only some parts of that system,* (Q.2) especially when the population from which the model is being learned is *the result of not one but multiple causal systems*, and (Q.3) *without assuming a particular distribution*, such as a Gaussian one.

In answer to this question: (Q.1) we have shown that causal inference of partially-observed systems, and more specifically under the assumption of strong causal insufficiency, is in general possible up to observational equivalence over the observed variables; we have shown that additionally assuming vertex- or edge-minimality allows learning a minMCM, which gives a lower bound respectively on either the number of latent variables or causal relations present in any observationally consistent MCM and thus a minimal causal structure; we have provided a software package capable of, in addition to learning these causal structures, using GANs to learn FCMs corresponding to these structures; (Q.2) we have shown that the distance between generating causal structures can be measured directly from their generated samples, so that samples close together in this sense are guaranteed to come from the same causal structure, thus facilitating the application of existing inference methods separating the samples and then learning each generating structure; and finally, (Q.3) we have done this all without assuming any particular distribution or parametrization,

making use of only general independence measures, thus placing no constraints (such as linearity) on the functional form of the causal relations.

We have thus contributed to the field of causal inference by (i) defining a new class of causal model for measurement variables with latent confounders, methods for learning these models, (ii) defining a kernel, provably capable of measuring distance between samples based on how different their generating causal models are (and recall, this applies more generally to ancestral graph causal models, not just our newly defined MeDIL causal models), which can be applied in a wide variety of causal inference tasks, from clustering, to statistical hypothesis testing, to structure learning, to feature visualization, etc., as well as (iii) providing a free and open source software package making these methods widely available for other researchers to apply, study, and modify. We have demonstrated our methods on psychometric data and gene expression data, making clear what assumptions (such as strong causal insufficiency, or structural heterogeneity) data must satisfy in order for our methods to be applicable. These contributions lay the foundation for a variety of different applications and extensions of statistical and causal inference methods, both in applied fields, such as brain-computer interfaces, as well as theoretical fields like algebraic statistics.

In making these contributions, we focused mathematically on undirected graphs, exploring various ways in which they can represent equivalence classes of causal models, how some of their existing known properties (such as edge clique covers) can be given a novel causal semantics, and how the structure defined by distances between these graphs can be embedded into a real-valued vector space so that we can use samples to implicitly and efficiently reason about their generating graphs.

Causally, we drew inspiration from the Neyman-Rubin framework, making use of (i) its clear distinction between variables that are possible causes and those that are their effects and (ii) its foundation in population as opposed to typological thinking, as well as from the graphical causal model framework developed by Spirtes, Pearl, and others, including its focus on graphs and the way it allows reasoning about causal models more abstractly in terms of their causal structure based on patterns of probabilistic independence, even in the absence of interventional data. Furthermore, and most of all, we hope our work here hints at possibilities for a new, richer understanding of causality, contributing to a shift beyond the heavily assumption-laden (e.g., linear, Gaussian, sufficient FCMs) effort to find a single "true" causal model for a data set and rather toward a broader, more intricate understanding in which a given data set is seen as but one limited perspective of a more complex causal system, invariably admitting many different causal models that are not only interesting in themselves but also in their relation to each other and for the more complete picture their combined consideration can provide of the underlying causal system and of causality more generally, i.e., of how we fundamentally interact with and understand the world of which we are a part.

Bibliography

Abdullah, W., Hossain, S., and Khan, M. (2020). A sparse matrix approach for covering large complex networks by cliques.

Agarwal, A. and Mazumdar, A. (2016). Local partial clique covers for index coding. *arXiv preprint arXiv:1603.02366*.

Ali, R. A., Richardson, T. S., and Spirtes, P. (2009). Markov equivalence for ancestral graphs. *The Annals of Statistics*, 37(5B):2808–2837.

Beckers, S., Eberhardt, F., and Halpern, J. Y. (2020). Approximate causal abstractions. In *Uncertainty in Artificial Intelligence*, pages 606–615. PMLR.

Black, E. (2012). *War against the weak: Eugenics and America's campaign to create a master race*. Dialog Press.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Chalupka, K., Perona, P., and Eberhardt, F. (2014). Visual causal feature learning. *arXiv preprint arXiv:1412.2309*.

Chivukula, A., Markham, A., Bischl, B., and Grosse-Wentrup, M. (2020). Learning MeDIL causal models using generative neural networks. masters thesis.

Conte, A., Grossi, R., and Marino, A. (2020). Large-scale clique cover of real-world networks. *Information and Computation*, 270:104464.

Crenshaw, K., Gotanda, N., Peller, G., and Thomas, K. (1995). *Critical race theory: The Key Writings that formed the Movement*. The New Press.

Dehejia, R. H. and Wahba, S. (1999). Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American statistical Association*, 94(448):1053–1062.

Ding, C., Li, T., and Jordan, M. I. (2008). Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *2008 Eighth IEEE International Conference on Data Mining*, pages 183–192. IEEE.

Eichler, M. (2012). Causal inference in time series analysis. In Berzuini, C., Dawid, P., and Bernardinell, L., editors, *Causality: Statistical perspectives and applications*, pages 327–352. John Wiley & Sons.

Filon, L. N. G., Yule, G. U., Westergaard, H., and Greenwood, M. (1934). Speeches delivered at a dinner held in university college, london in honour of professor karl pearson 23 april 1934.

Galton, F. (1889). I. co-relations and their measurement, chiefly from anthropometric data. *Proceedings of the Royal Society of London*, 45(273-279):135–145.

Garton, S. (2010). Eugenics in australia and new zealand: Laboratories of racial science. *The Oxford handbook of the history of eugenics*, pages 243–257.

Gelman, A. (2009). Resolving disputes between j. pearl and d. rubin on causal inference. `https://statmodeling.stat.columbia.edu/2009/07/05/disputes_about/`. Accessed: 2021-06-03.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438.

Granger, C. W. (1980). Testing for causality: a personal viewpoint. *Journal of Economic Dynamics and control*, 2:329–352.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of intelligent information systems*, 17(2):107–145.

Holland, P. W., Glymour, C., and Granger, C. (1985). Statistics and causal inference*. *ETS Research Report Series*, 1985(2):i–72.

Horkheimer, M. (1972). Traditional and critical theory. *Critical theory: Selected essays*.

Hulswit, M. (2004). A short history of causation. *SEED Journal (Semiotics, Evolution, Energy, and Development)*, 4(3):16–42.

Isserlis, L. (1914). On the partial correlation ratio. *Biometrika*, 10(2/3):391–411.

Kalupahana, D. J. and Deutsch, E. (1975). *Causality: The central philosophy of Buddhism*. University Press of Hawaii Honolulu.

Kindermann, R. and Snell, J. L. (1980). Ii. markov fields on graphs. *Contemporary Mathematics*, page 24–33.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11):1432–1438.

Markham, A., Chivukula, A., and Grosse-Wentrup, M. (2020). `MeDIL`: A python package for causal modelling. In Jaeger, M. and Nielsen, T. D., editors, *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, pages 621–624. PMLR.

Markham, A. and Grosse-Wentrup, M. (2020). Measurement dependence inducing latent causal models. In Peters, J. and Sontag, D., editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 590–599. PMLR.

Markham, A. and Grosse-Wentrup, M. (2021). A Distance Covariance-based Kernel for Nonlinear Causal Clustering in Heterogeneous Populations. In *under review but available as arXiv e-print*, page arXiv:2106.03480.

Mautz, D., Ye, W., Plant, C., and Böhm, C. (2018). Discovering non-redundant k-means clusterings in optimal subspaces. *Kdd'18: Proceedings of the 24Th Acm Sigkdd International Conference on Knowledge Discovery & Data Mining*, pages 1973–1982.

Naselaris, T., Allen, E., and Kay, K. (2021). Extensive sampling for complete models of individual brains. *Current Opinion in Behavioral Sciences*, 40:45–51.

Neapolitan, R. E. et al. (2004). *Learning bayesian networks*, volume 38. Pearson Prentice Hall Upper Saddle River, NJ.

Neyman, J. (1923). Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes. *Roczniki Nauk Rolniczych*, 10:1–51.

Niles, H. E. (1922). Correlation, causation and wright's theory of" path coefficients". *Genetics*, 7(3):258.

Niles, H. E. (1923). The method of path coefficients an answer to wright. *Genetics*, 8(3):256.

Niu, D., Dy, J. G., and Jordan, M. I. (2010). Multiple non-redundant spectral clustering views. In *ICML*.

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

Pearl, J. (2009). *Causality*. Cambridge university press.

Pearl, J. (2012). The do-calculus revisited. *arXiv preprint arXiv:1210.4852*.

Pearl, J. and Mackenzie, D. (2018). *The book of why: the new science of cause and effect*. Basic books.

Pearson, K. (1895). Vii. note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, 58(347-352):240–242.

Pegoraro, L. (2015). Second-rate victims: The forced sterilization of indigenous peoples in the usa and canada. *Settler Colonial Studies*, 5(2):161–173.

Reichenbach, H. (1956). *The direction of time*. University of California Press.

Richardson, T. and Spirtes, P. (2002). Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030.

Richardson, T. S. and Spirtes, P. (2003). Causal inference via ancestral graph models. *Oxford Statistical Science Series*, 1(27):83–105.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Rubenstein, P., Weichwald, S., Bongers, S., Mooij, J., Janzing, D., Grosse-Wentrup, M., and Schölkopf, B. (2017). Causal consistency of structural equation models. In *Proceedings of the Thirty-Third Annual Conference on Uncertainty in Artificial Intelligence (UAI 2017)*.

Rubin, D. B. (1973). Matching to remove bias in observational studies. *Biometrics*, pages 159–183.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331.

Sadeghi, K. and Lauritzen, S. (2014). Markov properties for mixed graphs. *Bernoulli*, 20(2):676–696.

Sarvepalli Radhakrishnan, S. (1957). *A Source Book in Indian Philosophy*. Princeton University Press.

Sim, S. (2004). *Introducing critical theory: A graphic guide*. Icon Books.

Spearman, C. (1904). " general intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292.

Spearman, C. E. (1927). *The abilities of man*, volume 6. Macmillan New York.

Spirtes, P. and Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. (2000). *Causation, prediction, and search*. MIT press.

Spirtes, P., Meek, C., and Richardson, T. (1999). An algorithm for causal inference in the presence of latent variables and selection bias. *Computation, causation, and discovery*, 21:1–252.

Stubblefield, A. (2007). "beyond the pale": Tainted whiteness, cognitive disability, and eugenic sterilization. *Hypatia*, 22(2):162–181.

Studený, M., Hemmecke, R., and Lindner, S. (2010). Characteristic imset: a simple algebraic representative of a bayesian network structure. In *Proceedings of the 5th European workshop on probabilistic graphical models*, pages 257–264. HIIT Publications.

Sullivant, S. (2018). *Algebraic statistics*, volume 194. American Mathematical Soc.

Tan, D. S. and Nijholt, A. (2010). *Brain-computer interfaces*. Springer-Verlag London Limited.

Thompson, B. (2004). Exploratory and confirmatory factor analysis: Understanding concepts and applications.

Tydén, M. (2010). The scandinavian states: reformed eugenics applied. *Bashford and Levine*.

Verma, T. and Pearl, J. (1988). *Influence diagrams and d-separation*. UCLA, Computer Science Department.

Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 255–270.

Vigen, T. Spurious correlations: Computer science doctorates awarded vs. precipitation in Kansas. `http://tylervigen.com/view_correlation?id=95475`. Accessed: 2021-06-02.

Vigen, T. Spurious correlations: Computer science doctorates awarded vs. precipitation in Mississippi. `http://tylervigen.com/view_correlation?id=95476`. Accessed: 2021-06-02.

Westland, J. C. (2015). From paths to networks: The evolving science of networks. In *Structural equation models*, pages 161–172. Springer.

Wintroub, M. (2020). Sordid genealogies: a conjectural history of cambridge analytica's eugenic roots. *Humanities and Social Sciences Communications*, 7(1):1–16.

Wright, S. (1921). Correlation and causation. *J. agric. Res.*, 20:557–580.

Wright, S. (1923). The theory of path coefficients a reply to niles's criticism. *Genetics*, 8(3):239.

Wright, S. (1934). The method of path coefficients. *The annals of mathematical statistics*, 5(3):161–215.

Xie, Y. (2013). Population heterogeneity and causal inference. *Proceedings of the National Academy of Sciences*, 110(16):6262–6268.

Xu, J., Markham, A., Meunier, A., Raggam, P., and Grosse-Wentrup, M. (2021). Distance covariance: A nonlinear extension of riemannian geometry for EEG-based brain-computer interfacing. In *2021 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE.

A. Appendix

A.1. Dependence Contribution Kernel Implementation in Python

```python
## Copyleft 2021, Alex Markham, see https://medil.causal.dev/license.html
# Tested with versions:
# python: 3.9.5
# numpy: 1.20.3
# scipy: 1.6.3
import numpy as np
from numpy import linalg as LA
from scipy.spatial.distance import pdist, squareform
from scipy.stats import chi2


def dep_contrib_kernel(X, alpha=None):
    num_samps, num_feats = X.shape
    thresh = np.eye(num_feats)
    if alpha is not None:
        thresh[thresh == 0] = (
            chi2(1).ppf(1 - alpha) / num_samps
        )  # critical value corresponding to alpha
        thresh[thresh == 1] = 0
    Z = np.zeros((num_feats, num_samps, num_samps))
    for j in range(num_feats):
        n = num_samps
        t = np.tile
        D = squareform(pdist(X[:, j].reshape(-1, 1), "cityblock"))
        D_bar = d.mean()
        D -= (
            t(d.mean(0), (n, 1)) + t(d.mean(1), (n, 1)).T - t(D_bar, (n,
            ↪  n))
        )  # doubly centered
        Z[j] = D / (D_bar)  # standardized
    F = Z.reshape(num_feats * num_samps, num_samps)
    left = np.tensordot(Z, thresh, axes=([0], [0]))
    left_right = np.tensordot(left, Z, axes=([2, 1], [0, 1]))
    gamma = (F.T @ F) ** 2 - 2 * (left_right) + LA.norm(thresh)  # helper
    ↪  kernel

    diag = np.diag(gamma)
    kappa = gamma / np.sqrt(np.outer(diag, diag))  # cosine similarity
    kappa[kappa > 1] = 1  # correct numerical errors
    return kappa
```

A.2. Dependence Contribution Kernel Application

```python
## Copyleft 2021, Alex Markham, see https://medil.causal.dev/license.html
# Tested with versions:
# python: 3.9.5
# requests: 2.25.1
# numpy: 1.20.3
# scipy: 1.6.3
# medil: 0.6.0
# matplotlib: 3.4.2
# networkx: 2.5
import requests, os
import numpy as np
from numpy import linalg as LA
from scipy.spatial.distance import pdist, squareform
from scipy.stats import chi2
from medil.ecc_algorithms import find_clique_min_cover as find_cm
import matplotlib.pyplot as plt
import networkx as nx


## Load data into Python, and download first if necessary

path = "/home/alex/projects/clustering/temp/"  # "./"  # change if desired
if not os.path.exists(path + "data.txt"):
    url = "http://genome-www.stanford.edu/serum/data/fig2clusterdata.txt"
    r = requests.get(url)
    with open(path + "data.txt", "w") as f:
        f.write(r.text)
cols = np.arange(5, 16)
data = np.loadtxt(
    path + "data.txt",
    skiprows=2,
    usecols=cols,
    delimiter="\t",
)


## Perform clustering


def dep_contrib_kernel(X, alpha=0.1):
    num_samps, num_feats = X.shape
    thresh = np.eye(num_feats)
    if alpha is not None:
        thresh[thresh == 0] = (
            chi2(1).ppf(1 - alpha) / num_samps
        )  # critical value corresponding to alpha
        thresh[thresh == 1] = 0
    Z = np.zeros((num_feats, num_samps, num_samps))
    for j in range(num_feats):
        n = num_samps
```

```
        t = np.tile
        D = squareform(pdist(X[:, j].reshape(-1, 1), "cityblock"))
        D_bar = D.mean()
        D -= (
            t(D.mean(0), (n, 1)) + t(D.mean(1), (n, 1)).T - t(D_bar, (n,
            ↪  n))
        )  # doubly centered
        Z[j] = D / (D_bar)  # standardized
    F = Z.reshape(num_feats * num_samps, num_samps)
    left = np.tensordot(Z, thresh, axes=([0], [0]))
    left_right = np.tensordot(left, Z, axes=([2, 1], [0, 1]))
    gamma = (F.T @ F) ** 2 - 2 * (left_right) + LA.norm(thresh)  # helper
    ↪  kernel

    diag = np.diag(gamma)
    kappa = gamma / np.sqrt(np.outer(diag, diag))  # cosine similarity
    kappa[kappa > 1] = 1  # correct numerical errors
    return kappa, gamma


def kernel_k_means(data, num_clus=6, kernel=dep_contrib_kernel,
↪  max_iters=100):
    num_samps, num_feats = data.shape
    rng = np.random.default_rng(1312)
    init = rng.choice(
        num_samps, num_clus, replace=False
    )  # choose initial clusters using Forgy method
    inner_prods, _ = kernel(data)
    left = np.tile(np.diag(inner_prods)[:, np.newaxis], (1, num_clus))
    distances = (
        left
        - 2 * inner_prods[:, init]
        + np.tile(inner_prods[init, init], (num_samps, 1))
    )
    # use law of cosines to get angle instead of Euc dist
    # clip corrects for numerical error, e.g. 1.0000004 instead of 1.0
    arc_distances = np.arccos(np.clip((1 - (distances ** 2 / 2)), -1, 1))
    labels = np.argmin(arc_distances, axis=1)
    for itr in range(max_iters):
        # compute kernel distance using ||x - mu|| = k(x,x) -
        ↪  2k(x,mu).mean() + k(mu,mu).mean() = left - 2*middle + right
        ip_clus = np.tile(inner_prods, (num_clus, 1, 1))

        m_idx = np.fromiter(
            (j for c in range(num_clus) for i in labels for j in labels ==
            ↪  c),
            bool,
            num_clus * num_samps ** 2,
        )
        m_idx = m_idx.reshape(num_clus, num_samps, num_samps)
        counts = np.fromiter(
```

```
            ((labels == label).sum() for label in range(num_clus)), int,
            ↪  num_clus
        )
        # counts = m_idx[:, 0, :].sum(1)
        ip_clus[~m_idx] = 0
        middle = ip_clus.sum(2).T / counts  # sum/ counts, because 0s
        ↪  through off mean

        r_idx = np.fromiter(
            (
                (i and j)
                for c in range(num_clus)
                for i in labels == c
                for j in labels == c
            ),
            bool,
            num_clus * num_samps ** 2,
        )
        r_idx = r_idx.reshape(num_clus, num_samps, num_samps)
        ip_clus[~r_idx] = 0
        right = ip_clus.sum((1, 2)) / (counts ** 2)

        distances = left - 2 * middle + right
        # law of cosines
        arc_distances = np.arccos(np.clip((1 - (distances ** 2 / 2)), -1,
        ↪  1))
         new_labels = np.argmin(arc_distances, axis=1)
        if (labels == new_labels).all():
            print("converged")
            break
        print("iteration {} with cluster sizes {}".format(itr, counts))
        labels = new_labels
    return labels


cluster_labels = kernel_k_means(data)


## Generate plots


def make_heatmaps_and_dags(path, data, labels):
    with open(path + "data.txt") as f:
        first_line = f.readline()
    x = first_line.split("\t")[5:16]
    x[0:2] = ["0.25", "0.5"]
    x[2:] = [time[:-2] for time in x[2:]]

    ## Dcov
    def compute_d_cov(X):
        num_samps, num_feats = X.shape
        dists = np.zeros((num_feats, num_samps ** 2))
```

```
146         d_bars = np.zeros(num_feats)
147         # compute doubly centered distance matrix for every feature:
148         for feat_idx in range(num_feats):
149             n = num_samps
150             t = np.tile
151             # raw distance matrix:
152             d = squareform(pdist(X[:, feat_idx].reshape(-1, 1),
                ↪  "cityblock"))
153             # doubly centered:
154             d_bar = d.mean()
155             d -= t(d.mean(0), (n, 1)) + t(d.mean(1), (n, 1)).T - t(d_bar,
                ↪  (n, n))
156             dd = d.flatten()
157             dists[feat_idx] = dd / n
158             d_bars[feat_idx] = d_bar
159         return dists @ dists.T, d_bars
160
161     plt.rcParams.update(
162         {
163             "text.usetex": True,
164             "font.family": "sans-serif",
165             "font.sans-serif": ["Helvetica"],
166         }
167     )
168
169     fig, axs = plt.subplots(
170         4, 2, figsize=(4, 9.5), sharex=True, sharey=True,
            ↪  constrained_layout=True
171     )
172     alpha = 0.1
173     crit = chi2(1).ppf(1 - alpha)
174     counts = np.append(517, np.bincount(cluster_labels))
175     ims = dict()
176     deps = dict()
177     covs = dict()
178     tests = dict()
179     for r in range(2):
180         for c in range(4):
181             if c == 0:
182                 cov, d_bars = compute_d_cov(data)
183             if c > 0:
184                 c += 2
185                 cov, d_bars = compute_d_cov(data[cluster_labels == c - 1])
186             covs[c] = cov
187             if r == 1:
188                 dep = np.zeros_like(cov)
189                 test = counts[c] * cov / np.outer(d_bars, d_bars)
190                 dep[test > crit] = 1
191                 deps[c] = dep
192                 tests[c] = test
193             c -= 2 if c > 0 else 0
194             ax = axs[c, r]
```

```
            cmap = "YlOrBr_r" if r == 0 else "binary"
            im = cov if r == 0 else -dep
            ims[(r + 1) * (c + 1)] = ax.imshow(im, cmap=cmap)
            ax.set_yticks(np.arange(len(x)))
            ax.set_yticklabels(x)
            ax.set_xticks(np.arange(len(x)))
            ax.set_xticklabels(x, rotation=80)
    # fig.text(0.53, -0.03, "Time (hours)", ha="center", va="center")
    # fig.text(-0.03, 0.5, "Time (hours)", ha="center", va="center",
    ↪  rotation="vertical")
    box = dict(facecolor="none", edgecolor="black", boxstyle="round")
    fig.text(0.54, 0.99, "Unclustered data", ha="center", va="center",
    ↪  bbox=box)
    fig.text(0.54, 0.765, "Cluster K1", ha="center", va="center", bbox=box)
    fig.text(0.54, 0.54, "Cluster K2", ha="center", va="center", bbox=box)
    fig.text(0.54, 0.295, "Cluster K3", ha="center", va="center", bbox=box)
    # fig.text(
    #     0.49,
    #     0.50,
    #     r"Same custers as above, but with values thresholded to 0 or 1,
    ↪  using $\alpha=0.1$",
    #     ha="center",
    #     va="center",
    #     bbox=box,
    # )
    cbar = fig.colorbar(
        ims[1],
        ax=axs[3, 0],
        location="bottom",
        shrink=0.6,
        label="distance covariance",
        ticks=[0.005, 0.8],
    )
    cbar.ax.set_xticklabels(["0", "0.8"])

    cbar2 = fig.colorbar(
        ims[4],
        ax=axs[3, 1],
        location="bottom",
        shrink=0.6,
        label=r"dependent",  # , $\alpha = 0.1$",
        ticks=[-0.75, -0.25],
        # boundaries=[0, 0.5, 1],
        values=[-1, 0],
    )
    cbar2.ax.set_xticklabels(["true", "false"])
    # fig.text(0.95, 0.3, "Fail to reject", ha="center", va="center",
    ↪  bbox=box, rotation=-90)
    # plt.tight_layout()
    # fig.text(0.935, 0.247, "0", ha="center", va="center")
    plt.savefig(path + "heatmaps.png", dpi=200, bbox_inches="tight")
    fig.clf()
```

```
243
244     def plot_dag(biadj_mat, ax):
245         num_latent, num_obs = biadj_mat.shape
246         pos_dict = {}
247         latent_pos_dict = {
248             idx: (val, 1) for idx, val in enumerate(np.linspace(0, 1,
                ↪  num_latent))
249         }
250         obs_pos_dict = {
251             idx + num_latent: (val, 0)
252             for idx, val in enumerate(np.linspace(0, 1, num_obs))
253         }
254
255         pos_dict.update(latent_pos_dict)
256         pos_dict.update(obs_pos_dict)
257         # print(pos_dict)
258
259         node_color = []
260         node_color.extend(num_latent * [0])
261         node_color.extend(num_obs * [1])
262
263         full_adj_mat = get_dag_from_biadj(biadj_mat)
264
265         G = nx.DiGraph(full_adj_mat)
266
267         nx.draw_networkx(G, pos=pos_dict, with_labels=False, ax=ax,
            ↪  node_size=500)
268         nx.draw_networkx_labels(
269             G,
270             pos=latent_pos_dict,
271             labels={idx: "$L_{{{}}}$".format(idx + 1) for idx in
                ↪  range(num_latent)},
272             font_color="w",
273             ax=ax,
274         )
275         nx.draw_networkx_labels(
276             G,
277             pos=obs_pos_dict,
278             labels={
279                 idx + num_latent: "$M_{{{}}}$".format(idx) for idx in
                    ↪  range(num_obs)
280             },
281             font_color="w",
282             ax=ax,
283         )
284         nx.draw_networkx_nodes(
285             G, pos=pos_dict, node_color=node_color, ax=ax, node_size=0
286         )
287         # nx.draw_networkx(G, pos=pos_dict, arrows=True, with_labels=False)
288         ax.set_xlim(-0.1, 1.1)
289         ax.set_ylim(-0.5, 1.85)
290
```

```
    def get_dag_from_biadj(biadj_mat):
        num_latent, num_obs = biadj_mat.shape
        dag_adj_mat = np.zeros((num_latent + num_obs, num_latent +
        ↪  num_obs))
        dag_adj_mat[:num_latent, num_latent:] = biadj_mat
        return dag_adj_mat

    biadj_mats = dict()
    for key in deps.keys():
        dep = deps[key]
        biadj_mats[key] = find_cm(deps[key])

    fig, axs = plt.subplots(4, 1, figsize=(5, 5.5),
    ↪  constrained_layout=True)
    for idx, key in enumerate((0, 3, 4, 5)):
        ax = axs[idx]
        b_mat = biadj_mats[key]
        plot_dag(b_mat[np.lexsort(b_mat.T)], ax)

        box = dict(facecolor="none", edgecolor="black", boxstyle="round")
        fig.text(0.5, 0.96, "Unclustered data", ha="center", va="center",
        ↪  bbox=box)
        fig.text(0.5, 0.71, "Cluster K1", ha="center", va="center",
        ↪  bbox=box)
        fig.text(0.5, 0.46, "Cluster K2", ha="center", va="center",
        ↪  bbox=box)
        fig.text(0.5, 0.21, "Cluster K3", ha="center", va="center",
        ↪  bbox=box)
        plt.savefig(path + "dags.png", dpi=200, bbox_inches="tight")


make_heatmaps_and_dags(path, data, cluster_labels)
```