Bauhaus-Universität Weimar
Faculty of Media
Degree Programme Computer Science for Digital Media

Conditional Independence Test for Categorical Data towards Causal Discovery and Application in Bibliometrics

Master's Thesis

Sagar Nagaraj Simha

1. Referee: Prof. Dr. Jakob Runge
2. Referee: Prof. Dr. Benno Stein
3. Referee: Dr. Magdalena Anna Wolska

Submission date: October 12, 2022

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Jena, Germany, October 12, 2022

.....

Sagar Nagaraj Simha

Abstract

This work describes the design and evaluation of a conditional independence test for categorical data towards causal discovery. Conditional independence (CI) testing is at the crux of a causal discovery framework. Constraint-based causal discovery algorithm suite such as PC (Spirtes et al. [2000]), PCMCI (Runge et al. [2019]), PCMCIplus (Runge [2020b]) and LPCMCI (Gerhardus and Runge [2020]), all use CI tests to infer causal relations from purely observational data. The theoretical performance of these algorithms is usually measured by assuming an 'oracle' CI which readily has the knowledge of dependence or independence. However, in practice this is not available and the performance of the causal discovery algorithm heavily depends on the performance of the CI test under different distributions of the data. Therefore the design of a calibrated CI test is paramount. Here, particular focus is on the case of categorical data and a test is presented based on conditional mutual information (CMI) combined with a local permutation scheme - CMISymbPerm.

The test is formulated as a hypothesis test of independence - $X \perp\!\!\!\perp Y \mid Z$ while any or all of the variables may be multivariate. The test is evaluated using a Bayesian Network with link strength as a data generating process and numerical experiments are run over different parameter configurations from 50 upto 2000 samples, number of symbols upto 6 and dimensions of Z up to 4. The experiments demonstrates that the test reliably approximates the true null distribution. Numerical experiments also include the comparison of CMISymbPerm with G^2 test statistic which approximates the null as a χ^2 distribution. Results show that CMISymbPerm and G^2 converge in type I error for large samples. The permutation scheme grows $\mathcal{O}(c^n)$ in time complexity with number of samples as compared to G^2 $\mathcal{O}(1)$. CMISymbPerm should be preferred for lower sample sizes, larger dimensions and higher number of symbols, while G^2 should be preferred for larger sample sizes or when time is a constraint. The PC algorithm with partial correlation test is then applied on Open Academic Graph 2.1 (OAG [2020]) to investigate causal links in Bibliometrics data of continuous variables.

The work on categorical CI testing is heavily based on CMIknn (Runge [2018]), a non-parametric test for continuous data. All the methods implemented are contributed as part of the package TIGRAMITE [1].

[1] https://github.com/jakobrunge/tigramite

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

I would like to thank the members of Causal Inference group at Deutsches Zentrum für Luft- und Raumfahrt (DLR), Jena, Germany for all the discussions and constructive suggestions at various stages of the thesis, particularly my supervisor Prof. Dr. Jakob Runge, Dr. Andreas Gerhardus, Nicolas-Domenic Reiter and Tom Hochsprung. I would also like to thank Dr. Magdalena Anna Wolska and Prof. Dr. Benno Stein at Bauhaus Universität Weimar for the co-supervision. I am particularly thankful to my family and friends for their unwavering support. I am also grateful for all the resources and infrastructure provided for this work by DLR, Jena and WEBIS, Bauhaus Universität Weimar.

Chapter 1

Introduction

1.1 Motivation

The field of causality formally studies the notion of causation and causal relationships amongst phenomena. Causal inference, specifically observational causal inference concerns with discovering and inferring causal relationships in the data generated by a phenomena of interest. The notion of causation goes a step beyond correlation [1], which so far has been the foundational principle behind most machine learning algorithms. But, as with the popular phrase 'correlation does not imply causation', it is important to differentiate flow of association (correlation) and causation when modeling a phenomenon from observational data. Analysis using purely correlation based estimates may lead to biases as the differences may be due to a third factor (confounder) or the sampling itself may be biased or inconsistent (selection bias). Causal inference provides the necessary scientific framework to perform such analyses and infer causal structures that generated the data, and consequently - how actions, interventions and treatments affect outcomes of interest. These ideas are formally studied under Causal Discovery and Cause-Effect estimation respectively. This has been widely applied in numerous fields. In epidemiology - how effective a vaccine is in curing a disease or if a new drug is indeed the cause of a side effect and not a pre-existing medical condition. In advertising - if a change in website's functionality and color scheme brought more customer engagement. In economics - if a new economic policy with a target at middle class households increased their purchasing power. Causal inference enables the discovery of such key insights.

[1] A more precise term is 'association' since it is a general term that defines all statistical dependencies. Here, the terms are interchangeably used.

The importance of causal inference is particularly evident in the case of Simpson's paradox - a statistical phenomenon where an association between two variables in a population emerges, disappears or reverses when the population is divided into subpopulations. Consider the example in Table 1.1. Say, a decision has to be made on prescribing a treatment (A/B). The data in Table 1.1 contains mortality rate (Y) based on specific treatment (T) given to a population with the severity of the condition (C). The goal is to make a decision on choice of T that reduces Y considering C. If the total effect is looked at, the decision to choose Treatment A is obvious considering lower mortality rate. But, when the data is looked at within subpopulations, the choice of Treatment B is more obvious. This is caused by unequal weighting of the data within different sample sizes. This can be well explained by looking into the two different causal structures in Figures 1.1 and 1.2, both generating the same data in the Table 1.1. In Figure 1.1, the treatment B is chosen since the condition influences both the mortality rate and the treatment, while in Figure 1.2, the treatment influences both the outcome and the condition (say longer wait times with Treatment B worsens the condition), in which case Treatment A is preferred. Therefore, the right decision highly depends on the underlying causal structure and not on the observational data alone. This analysis is particularly important in the field of epidemiology where the cost of a decision involves risks to human lives (Neal [2020]) .

Previously such a causal analysis was done by means of a randomized controlled trial, proposed by Wright [1921]. Here, a group is randomly split into control and treatment groups where only the latter is subjected to some treatment (also called intervention). Then, measurements are taken and if the measurements are significantly different between the two groups, then one can isolate that the treatment is indeed the cause of the measured changes in the variable. But these studies may be expensive, unethical, time consuming and in many cases infeasible - for example, in the case when the group has to be re-treated going back in time. Pearl [1988] revolutionized the field by introducing formal mathematical notions to causality (Pearl [2009]) which made such analyses possible circumventing the issues mentioned before. The concept of Structural Causal Models together with the ideas form Probabilistic Graphical Models paved way for the suite of algorithms in Causal Discovery and Cause-Effect estimation.

1.1.1 Summary

The suite of causal discovery algorithms reconstruct the underlying casual structure as a Directed Acyclic Graph (DAG) which entails a joint probability distribution. The algorithms are built on assumptions, specifically - Markov

Table 1.1: Simpson's paradox

	Condition		
	Mild	Severe	Total
Treatment A	15% (210/1400)	30% (30/100)	**16%** (240/1500)
Treatment B	**10%** (5/50)	**20%** (100/500)	19% (105/550)

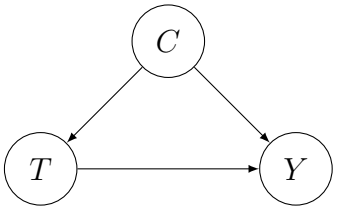

Figure 1.1: Causal Structure 1

equivalence, faithfulness and minimality (Peters et al. [2017]). Under the assumption of faithfulness, finding sets of random variables that are independent of each other given a conditioning set is equivalent to being d-separated in the DAG. The d-separated variables are then used to construct a causal graph by removing links and using orientation rules. Hence, an essential part of finding d-separated variables is testing independence. This is then formulated as a hypothesis test of conditional independence (CI) - $X \perp\!\!\!\perp Y \mid Z$. The CI test needs to be well calibrated in order to then calibrate the estimated causal graph. In this endeavor, this work proposes the design of a CI test - CMISymbPerm for categorical random variables and evaluates it using Bayesian Network as a SCM that generates ground truth data. We also compare with other well established method such as G^2 test statistic. Finally, we apply the causal discovery algorithm in discovering causal links in Open Academic Graph.

In the next section we introduce formally categorical data, hypothesis testing, evaluation metrics used as well as the PC algorithm. Chapter 2 describes the synthetic data generating process, specifically structural causal models which encodes the one way causal relationships amongst variables. We then

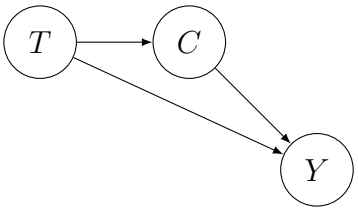

Figure 1.2: Causal Structure 2

formally define Bayesian Networks and Causal Bayesian Networks with important distinctions between them. We then introduce link Strength - η as a parameter which controls the effect size in case of dependence and also automatically generates conditional probability tables which helps expert elicitation.

In Chapter 3 we define the CMISymbPerm - An information theoretic dependence measure with a permutation scheme for generating the null distribution and the G^2 test statistic. In Chapter 4 we evaluate the performance of permutation scheme in null distribution approximation. The chapter also includes a comparison of the proposed CI test with G^2 test statistic. In Chapter 5, we look into Bibliometrics, a study of scientific publications. Here, we apply the causal discovery algorithm on data from Open Academic Graph 2.1 to discover causal links amongst features like h-index, number of publications, number of citations and so on. Finally, in chapter 6 we conclude with a brief on our findings and provide an outlook into further work.

1.1.2 Categorical data

A categorical random variable is a random variable which can take on values from a predefined set of categories or symbols. Each sample in a categorical data then belongs to a category that the random variable can take. There are no relationships between the categories except that they can be taken up by a random variable. Examples for a categorical random variable are different countries in the world, or different blood types. In the rest of the work, we use the word categories and symbols interchangeably.

1.1.3 Hypothesis testing and errors

In statistical test theory, a hypothesis test helps infer whether the data under consideration supports a particular hypothesis. The probabilistic decisions help make inferences about the population that the samples were derived from. Hypothesis test is foundational for many fields - communications, signal processing, psychology, economics, biology. A statistical error (Wikipedia [2022c]) is part of hypothesis testing. The test compares two hypotheses, the null hypothesis, denoted by H_0 and the alternate hypothesis, denoted by H_1. This is conceptually similar to the judgement in a court trial. The null hypothesis corresponds to the position of the defendant: is presumed to be innocent until proven guilty. So, the null hypothesis is presumed to be true until the data provide convincing evidence against it. The alternative hypothesis corresponds to the position against the defendant. In the hypothesis test of conditional independence, the null and alternate are:

		Actual	
		P(H_1 is true)	**N(H_0 is true)**
Detected	P'	True Positive (1-β)	False Positive (α)
	N'	False Negative (β)	True Negative (1-α)

Table 1.2: Confusion matrix

$$H_0 : X \perp\!\!\!\perp Y \mid Z$$
$$H_1 : X \not\perp\!\!\!\perp Y \mid Z$$

H_0 is the null hypothesis that X and Y are independent conditioned on Z, and H_1 is the alternate hypothesis that X and Y are not independent conditioned on Z, implying dependence.

If the result of the test corresponds with reality, then a correct decision has been made. However, if the result of the test does not correspond with reality, then an error has occurred. There are two situations in which the decision is wrong. The null hypothesis may be true but we reject H_0. On the other hand, the alternate hypothesis H_1 may be true, whereas we do not reject H_0. These are formally defined as type I error and type II errors.

A positive result "true" corresponds to rejecting the null hypothesis, while a negative result corresponds to failing to reject the null hypothesis (or accepting the null hypothesis). Thus, "false" means the conclusion drawn is incorrect. Consequently, a type I error is equivalent to a false positive, and a type II error is equivalent to a false negative. In terms of the hypothesis test of independence, the type I error is then wrongly concluding conditional independence, and type II error is incorrectly detecting conditional independence. The type I error is usually denoted by α, also called the significance level, while the type II error is denoted by β. The power of the test, or probability of a 'true positive', i.e, correctly rejects H_0 when the alternate hypothesis H_1 is true is then $1 - \beta$. Consequently, $1 - \alpha$ is then the probability of a 'true negative', i.e, correctly not rejecting the null hypothesis. The descriptions are summarized in the confusion matrix 1.2.

Given that the null hypothesis is true, when the probability of obtaining a test statistic as extreme as the one obtained is lower than α, then the test is said to be **calibrated**. This probability under the null is then defined as p-value.

1.1.4 Metrics for evaluation

A major design principle behind statistical hypothesis testing is that we try to control type I error rate. When we fix α we are attempting to ensure that when we repeat the test over many samples, at most α of true null hypotheses are incorrectly rejected. Thus, a controlled hypothesis test has a type-I error at most α. In other words, the probability of making a type I error is represented by α, which is the p-value below which the null hypothesis is rejected. A secondary goal of hypothesis testing is to minimise β - the type II error rate. Since power is defined as $1 - \beta$, we look to maximise the power of the test which is equivalent to minimising the type II errors. Therefore, in order to quantify the performance of the hypothesis test at α, we look at evaluation metrics that provide a way to quantify the trade offs between the type I error and the power of the test given a significance value or threshold. We look at the evaluation metric - True Positive Rate (TPR) and False Positive Rate (FPR) (Navarro [2015]).

TPR and FPR Both metrics are derived from the confusion matrix in table 1.2. The true positive rate (TPR) is the total number of correctly detected positive results out of all the positive samples performed in a test, also called the sensitivity.

$$TPR = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

The false positive rate (FPR) is the total number of incorrectly detected positive results out of all the negative samples performed in a test, also denoted by 1 - specificity.

$$FPR = \frac{False\ Positive}{False\ Positive + True\ Negative}$$

Both the metrics evaluate with respect to a given significance value α over varying parameters of sample size, number of symbols and dimensions of the conditioning set.

1.2 Causal Discovery from Observational Data

Causal Discovery or structure learning or structure identification is a technique that infers the causal relationships amongst random variables from observational data. These causal relationships are represented by Directed Acyclic Graphs. We specifically focus on constraint-based causal discovery algorithm - The PC algorithm (Spirtes et al. [2000]). The is built on the principle that the structure of the causal graph imposes constraints in the observational distribution, e.g. conditional independencies. Detecting these constraints can then be used to infer the causal graph. In order to build this algorithm, we first define some required graphical ideas and necessary assumptions (Gerhardus [2021], Neal [2020]).

d-separation In order to define d-separation, we first define what a 'blocked path' is in graphical models. A path between nodes X and Y is blocked by a (potentially empty) conditioning set Z if either of the following is true:

1. Along the path, there is a chain $\cdots \to W \to \cdots$ or a fork $\cdots \leftarrow W \to \cdots$, where W is conditioned on ($W \in Z$).

2. There is a collider W on the path that is not conditioned on ($W \notin Z$) and none of its descendants are conditioned on ($\mathrm{de}(W) \nsubseteq Z$).

The unblocked path is then a path that is not blocked where association flows.

Definition 1 (d-separation) *Two sets of nodes X and Y are d-separated by a set of nodes Z if all of the paths between any node in X and any node in Y are blocked by Z.*

If all the paths between X and Y are blocked, then we say that X and Y are d-separated. And if there exists at least one path between X and Y that is unblocked, then we say that X and Y are d-connected. The notation $X \bowtie Y \mid Z$ is used to denote d-separation in a graph G.

Assumption 1 (Data generation by a Structural Causal Model) *We assume that a structural causal model $\mathfrak{C}$ exists that generates the observational distribution P as well as interventional distributions.*

Assumption 2 (Acyclicity) *The structural causal model $\mathfrak{C}$ that generates the data is Acyclic.*

Assumption 3 (Markov Assumption) *Given that P, a joint probability distribution over X and Y is Markov with respect to a graph G, if X and Y are d-separated in G conditioned on Z, then X and Y are independent in P conditioned on Z. This is denoted as, $X \bowtie Y \mid Z \implies X \perp\!\!\!\perp Y \mid Z$*

Assumption 4 (Faithfulness) $X \perp\!\!\!\perp Y \mid Z \implies X \bowtie Y \mid Z$. *This assumption helps infer d-separation in the graph from independencies in the distribution and consequently infer causal graphs.*

Assumption 5 (Causal Sufficiency) *There are no unobserved confounders for any variables in the graph.*

Under the above assumptions, we can partially identify the causal graph, CPDAG (Completed Partially Directed Acyclic Graph). Different graphs within the CPDAG can entail the same probability distribution. These graphs are said to be Markov equivalent. Two structural qualities that we can use to distinguish graphs from each other are skeletons and colliders. A graph's skeleton is the structure one gets when all of its directed edges are replaced by undirected edges. Two graphs are Markov equivalent if and only if they have the same skeleton and same colliders. A PC algorithm learns the CPDAG.

1.2.1 The PC causal discovery algorithm

A PC algorithm starts with a complete undirected graph, where all the nodes are adjacent to each other. In the first phase, it uses conditional independence tests of the form ($X \perp\!\!\!\perp Y \mid Z$) to identify the skeleton. In the second phase, the colliders are identified and oriented. The third phase builds on identified colliders to further orient the other edges using logical rules. The three phases of a PC algorithm are shown in algorithms 1, 2 and 3 (Runge [2020a]).

Algorithm 1 The PC causal discovery algorithm - Skeleton discovery

PC algorithm Phase 1/3: Skeleton discovery.
Require: Conditional independence information among variables in **X**.

Form complete graph $\mathfrak{C}$ with edges o-o; define adj(X^j) for all X^j in $\mathfrak{C}$.
p = 0

while any adjacent pair (X^i, X^j) in $\mathfrak{C}$ satisfy |adj(X^j) \X^i| $\geq p$ **do**

  Select new ordered and adjacent pair (X^i, X^j) with |adj(X^j) \X^i| $\geq p$

  while (X^i, X^j) are adjacent and not all S $\subset$ *adj*(X^j) \X^i with |S| = p have been considered **do**

    Choose new S $\subset$ *adj*(X^j) \X^i with |S| = p

    if $X \perp\!\!\!\perp Y \mid S$ **then**

      Delete X^i o-o X^j from $\mathfrak{C}$

      Store sepset(X^i, X^j) = S

    end if

  end while

  p = p + 1

  Compute adj(X^j) for X^j in $\mathfrak{C}$

end while
return $\mathfrak{C}$, sepset

Algorithm 2 The PC causal discovery algorithm - Orienting colliders

PC algorithm Phase 2/3: Orienting colliders.
Require: Skeleton $\mathfrak{C}$, sepset.

for All Unshielded triples X^i o-o X^k o-o X^j with non-adjacent $(X^i, X^j$ in $\mathfrak{C}$ **do**

  if $X^k \notin$ *sepset*(X$^i, X^j$) **then**

    Orient X^i o-o X^k o-o X^j as $X^i \rightarrow X^k \leftarrow X^j$

  end if

end for
return $\mathfrak{C}$, sepset

Algorithm 3 The PC causal discovery algorithm - Apply orientation rules R1-R3

PC algorithm Phase 3/3: Apply orientation rules R1-R3.
Require: Partially oriented $\mathfrak{C}$. Exhaustively (repeat until no further orientations can be made) apply following rules

1. Orient $X^i \to X^k$ o-o X^j as $X^i \to X^k \to X^j$ whenever (X^i, X^j) are non-adjacent
2. Given $X^i \to X^k \to X^j$ and X^i o-o X^j : Orient X^i o-o X^j as $X^i \to X^j$
3. Orient X^i o-o X^j as $X^i \to X^j$ whenever there are two chains X^i o-o $X^k \to X^j$ and X^i o-o $X^l \to X^j$ such that X^k and X^l are non-adjacent

return $\mathfrak{C}$, sepset

1.3 Related Work

Here, we survey some of the approaches towards categorical data generating process (DGP) and CI testing. In DGP [2022], the authors generate categorical variables by transforming the continuous variables with an internal covariance structure. Here, the numerical values are min-max scaled and treated as probabilities. These values are then used to draw from multinomial distributions. The likelihood of a category depends on the variance of a continuous variable. However, this data discretization may lead to information loss. This method also assumes that the continuous dependent variables are all readily available. This can be a limitation when generating categorical time series. In Huegle et al. [2022], the authors introduce a mixed additive noise model as a data generating process. The model can generate mixed discrete-continuous as well as nonlinear data. They formalize it as a functional causal model in order to introduce the notion of causal relationships amongst variables. The categorical DGP involves a simple modulo over the number of symbols. This can be quite limiting to generate varying distributions according to complexity of domains. Log-linear models and Markov Chains are other modeling techniques to generate multivariate categorical data. But, they do not provide the mechanism to encode one way causal relationships which the Bayesian Networks help address. The extension to Dynamic Bayesian Network can also generate causal models for time series.

In Tsagris [2017a], the author proposes a conditional independence test of two variables with categorical data using Poisson log-linear models. The author considers the G^2 statistic as test of conditional independence with a time

optimization on generating contingency tables for observed and expected frequencies using a Poisson log-linear model. In Tsamardinos and Borboudakis [2010a], the authors take a similar approach as ours to compare permutation based exact test with the G^2 asymptotic test. They propose semi-parametric permutation tests that improves the time complexity of a conventional permutation test. They also evaluate the improvement in learning the Bayesian network that models the observed data. Our approach in CMISymbPerm includes the information theoretic based dependence measure along with the permutation scheme.

Chapter 2

Data generating process

This section defines a formal model towards synthetic data generation. This is intended as a ground truth model to benchmark the designed conditional independence test. Here, particular focus is on the generation of categorical data. A number of standard categorical data models (statistical models) are surveyed that can be interpreted as a causal model (assuming causal assumptions and causal mechanisms Peters et al. [2017]) - modulo, log linear, Markov Chain etc. This additionally requires the formal definition of structural causal models which provide the framework to encode causal relationships leveraging the above models within functional causal models (Pearl [2009]).

A causal model differs from the classic probabilistic model in that it encodes causal relationships amongst variables, allowing for interventions. The emphasis is on the ability to generate observational data that follows complex distributional models based on expert knowledge and intuition. Particular focus here is on Bayesian Network (Koller and Friedman [2009]), a probabilistic graphical model which provides flexibility to elicit intricate expert knowledge through conditional probability tables (CPTs). A causal Bayesian Network is then a case of Bayesian Network that additionally encodes statistical and causal assumptions by allowing for interventions. It inherits most of the properties from a Bayesian Network. This is explained in the later section on Bayesian Networks in detail.

2.1 Structural Causal Models

A formal method of describing and encoding causal relationships are Structural Causal Models (SCMs) or Structural Equation Models (SEMs) (Peters et al. [2017]). An SCM also describes the data generating process. The SCM entails a joint distribution over all the observables. A structural causal model $\mathfrak{C} := (S, \mathrm{P}_N)$ consists of a collection S of d (structural) assignments

$X_j := f_j(PA_j, N_j), j = 1, 2, ...d,$, where $PA_j \subseteq X_1, ..., X_d$ are called parents of X_j, and a joint distribution $P_N = P_{N_1}, ..., P_{N_d}$ over the noise variables are jointly independent.

The variables in PA_j cause X_j through a functional mechanism (Pearl [2009]) f_j while N_j accounts for all the background factors outside the model. The joint independence of the noise variables is sufficient to describe the causal relationship among all variables.

A Causal graph is then $G(\mathfrak{C})$, a directed graph with set of nodes X and directed edges from each variable in PA_j to X_j for all j, representing causal influences. Structural Causal Models can help answer counterfactual statements and cause-effect estimation.

2.2 Bayesian Network - A categorical data model

A Bayesian Network (BN) is a graphical model that represents a set of variables and their conditional dependencies via a Directed Acyclic Graph (DAG) (Koller and Friedman [2009]). In a DAG G, each node represents an event or a random variable and each directed edge between nodes encodes the causal influence of one variable over the other. The dependencies are quantified by conditional probabilities for each node given its parents in the network. A Bayesian Network typically models associations through a joint probability distribution - the probability of every possible event as defined by the values of the parents. The joint distribution of p nodes, where each node x_i has parents pa_i is,

$$P(x_1, x_2, ..., x_p) = \prod_i P(x_i|pa_i)$$

The Bayesian Network achieves compactness in complexity by factoring the joint distribution into local, conditional distributions for each variable, given its parents. This is the property of the Local Markov Assumption,

Assumption 6 (Local Markov Assumption) *Given its parents in the DAG, a node X is independent of all its non-descendants.*

In the example in figure 2.1 below, the local Markov assumption factorizes the joint probability distribution as,

$$P(x_1, x_2, x_3, x_4) = P(x_1)P(x_2|x_1)P(x_3|x_2, x_1)P(x_4|x_3)$$

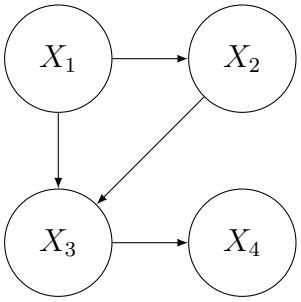

Figure 2.1: An example of a Bayesian Network with 4 random variables. X_3 has parents X_1 and X_2.

2.2.1 Causal Bayesian Networks

Bayesian Networks are often used to represent causal relationships, but this need not be the case. A directed edge from nodes X_A to X_B does not require that X_B be causally dependent on X_A. The two Bayesian Networks, x $\rightarrow$ y $\rightarrow$ z and x $\leftarrow$ y $\leftarrow$ z are equivalent, imposing the same conditional independence requirements. Only changing or intervening on y reveals the two graphs. Therefore, a causal Bayesian Network has the requirement that the relationships be causal. A variable is actively caused to be in a given state and the probability density function changes to that obtained when the links from the parents of that node are removed, and setting the variable to the caused value. This is formally described in Pearl [2009] as below.

Let $P(v)$ be a probability distribution on a set V of variables, and let $P_x(v)$ denote the distribution resulting from the intervention $do(X = x)$ that sets a subset X of variables to constants x. Denote by P_* the set of all interventional distributions $P_x(v)$, $X \subseteq V$, including $P(v)$, which represents no intervention (i.e, $X = null$). A DAG G is said to be a causal Bayesian Network compatible with P_* if and only if the following three conditions hold for every $P_x \in P_*$:

1. $P_x(v)$ is Markov relative to G
2. $P_x(v_i) = 1$ for all $V_i \in X$ whenever v_i is consistent with $X = x$;
3. $P_x(v_i|pa_i) = P(v_i|pa_i)$ for all $V_i \notin X$ whenever pa_i is consistent with $X = x$, i.e each $P(v_i|pa_i)$ remains invariant to interventions not involving V_i.

The above definition imposes constraints on the interventional space P_* that permit us to encode the vast space economically, in the form of a single Bayesian Network G. These constraints enable us to compute the distribution $P_x(v)$ resulting from any intervention $do(X = x)$ as a truncated factorization,

$$P_x(v) = \prod_{i|V_i \notin X} P(v_i|pa_i)$$

for all v consistent with x, which follows from and implies conditions 1-3, thus justifying the family deletion procedure on G.

2.3 Link Strength

A Bayesian Network encodes the causal beliefs of an expert's knowledge by means of Conditional Probability Tables (CPTs). CPTs relate the state distributions of the child nodes to that of their parent nodes by means of entries for all combinations of parents and children. The causal links can be encoded by eliciting probabilities of the CPT relating each parent and child. But, the size of the CPT grows exponentially both with the number of parents and the number of symbols for each child. If a child node has p parent nodes, and both the child node and the parent node have m symbols each, the total number of entries in the CPT is m^{p+1}. Even for moderate dimensions, say p=3 and m=5, the size of the CPT is already 625 entries. Eliciting probabilities for relatively large networks is then manually cumbersome. Therefore there is a need to automate generation of CPTs. One approach is to generate random stochastic matrices as CPT entries. Although they encode independence, the dependence reflects in the measured strength only over large samples. This is especially true when the number of symbols is higher requiring larger samples for better representation of all symbols. This also has the drawback that it is hard to control the effect size especially in presence of a confounder. There could be an analytical approximation that quantifies the effect with this method which is considered under future work.

The strength of the causal link can be parameterized providing an easier way for the expert to encode causal beliefs. This section adapts the ideas in Kokkonen et al. [2005] and proposes a single parameter link strength, η that combines the ideas of Varis and Kuikka [1997] and the generalised Noisy-Or model of Srinivas [1993] to derive the entire CPTs. η defined here ranges between 0 to 1 as compared to -1 to $+1$ originally defined in Kokkonen et al. [2005] and is given per link with the assumption that all the symbols are equally likely. The complexity of the elicitation reduces from m^{p+1} to p. η provides the means to control the effect size in order to test the effectiveness of the CI test, especially in the case of dependence. Also, link strengths can be particularly useful in eliciting causal beliefs where the knowledge of strengths are only known relative to each other.

The following properties are desired in the relationship between link strength parameters and the resulting derived CPTs

1. $\eta = 0$ denotes the case when the child is independent of the parent.
2. $\eta_1 = \eta_2$ denotes both parents have equal effect on the child node.
3. When all $\eta = 0$, the CPT is non-informative, i.e. all probabilities are equal to the inverse of the number of symbols in the child node.

Generalized Noisy-Or model with link strength

The generalized noisy-or model, in figure 2.2 derives the conditional probabilities relating the symbols of the parent and the child nodes, where each node may have arbitrary number of symbols. However this work assumes that all parents and children have m symbols (this can be extended to arbitrary number of symbols in future work). The symbol of each of n parents u_i in $[u_i]_{i=1,2,...,n}$ is passed through a line failure probability matrix $P(u_i^{'}|u_i)$ represented by N_i in the schematic, each being a square matrix of size = m x m, producing $[u_i^{'}]_{i=1,2,...,n}$. The $u^{'}$ is then passed through F that does a weighted average over the symbol indices of $u_i^{'}$ to produce $P(x|u)$, where x is the child node.

The $P(u^{'}|u)$ and F can be parameterized by aid of a single parameter link strength η and is the only parameter required for each link. η is defined between 0 to 1 where 1 means strong influence while 0 means no influence. η relates to $P(u^{'}|u)$, F and $P(x|u)$ as,

$$P(u_i^{'}(r)|u_i(c)) = \begin{cases} \frac{1}{m} + \eta_i \left(1 - \frac{1}{m}\right) & \text{if } r = c \\ \\ \frac{1}{m-1}\left[1 - \frac{1}{m} - \eta_i \left(1 - \frac{1}{m}\right)\right] & \text{if } r \neq c \end{cases} \tag{2.1}$$

When the η increases from 0 to 1, the diagonals of the CPT increases from 1/m to 1, and the remaining probability mass is distributed evenly over the off-diagonal elements of the CPT. This is from the intuition that the parent is likely to continue to retain its symbol with probability 1/m.

And the F as,

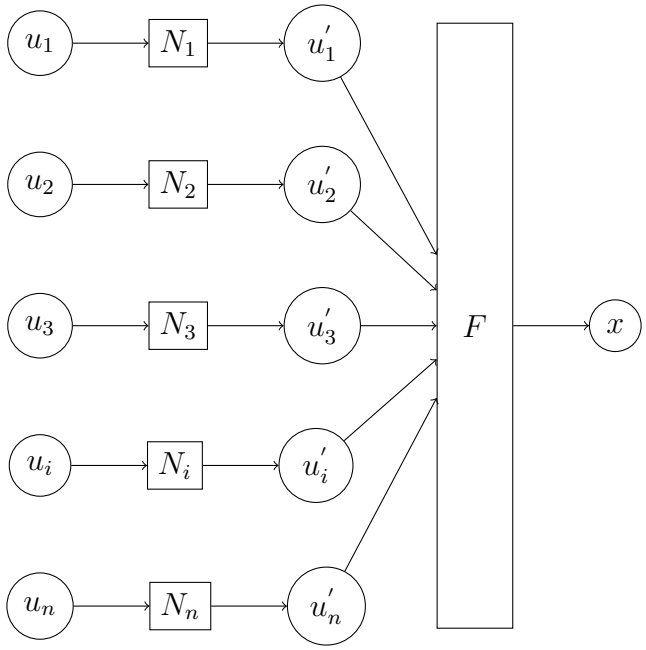

Figure 2.2: Schematic of the generalized Noisy-Or model.

$$F(u^{'}) = x\left(ceiling\left(\frac{1}{\sum_i \eta_i}\sum_i [\eta_i I(u_i^{'})]\right)\right) \tag{2.2}$$

$F(\cdot)$ denotes the symbol of child node x, $I(u_i^{'})$ is the numerical index of the symbol $u_i^{'}$, m is the number of symbols of the node. $ceiling(.)$ is a roundup function to the closest index. $x(j)$ denotes the j^{th} symbol of the child node x.

The symbols of the child node x relate to that of the parent nodes u,

$$P(x|u) = \sum_{u^{'}:x=F(u^{'})} P(u^{'}|u) = \sum_{u^{'}:x=F(u^{'})} \prod_{u^{'}} P(u_i^{'}|u_i) \tag{2.3}$$

This means that we go through the search space of all permutations of $(u, u^{'})$ over m symbols and sum the joint probabilities of each permutation that results in a category in x. The Algorithm to generate $P(x|u)$ for a child x given $u_{i=1,2,\ldots,n}$ parents is described in algorithm 4.

Some changes from the Kokkonen et al. [2005] and issues to be considered in future work

1. The η originally proposed is between -1 to 1, however this work only considers between 0 to 1. Since the main intention of link strength was to control the effect size in a CI test, η between 0 to 1 suffices.

2. The ceiling($\cdot$) function in equation 2.2 is introduced in order to convert the float argument into an integer index of the symbol. This will introduce a bias towards the higher index.

Algorithm 4 Derive the CPT $P(x|u)$

Require: Given a child node x and parents $u_{i=1,2,\ldots,n}$ with link strengths $\eta_{i=1,2,\ldots,n}$ and each node having m symbols each

1. For each parent u_i, generate $P(u_i^{'}|u_i)$ given m and η_i using equation 2.1
2. Generate all $L = m^{2p}$ permutations of $(u, u^{'})$
3. For each permutation $perm$ in L of $(u, u^{'})$, find $x_{perm} = F(u^{'})$
4. Compute the joint probability $P(u^{'}|u)$ for each x_{perm} with $\prod_{perm} P(u_i^{'}|u_i)$
5. Compute the total probability $P(x|u) = \sum_{x_{perm}} P(u^{'}|u)$ for each symbol of x

---

Chapter 3

Conditional Independence Test

Conditional independence (CI) test concerns with the problem of testing independence between random variables X and Y accounting for the confounding random variable Z, where all the variables are categorical and are either uni-variate or multivariate in nature. We look at the general definition of CI where the conditional probability mass function factorizes: $p(X, Y|Z) = p(X|Z)\ p(Y|Z)$. The CI test is usually formulated as a hypothesis test with the null hypothesis representing independence and the alternate hypothesis representing dependence,

$$H_0 : X \perp\!\!\!\perp Y \mid Z \tag{3.1}$$

$$H_1 : X \not\perp\!\!\!\perp Y \mid Z \tag{3.2}$$

Consider the case with n i.i.d tuples (X_i, Y_i, Z_i), defined in a high-dimensional space $(X_i \times Y_i \times Z_i)$ in $\mathbb{Z}^{d_x} \times \mathbb{Z}^{d_y} \times \mathbb{Z}^{d_z}$. Conditional independence test statistic $T : X \times Y \times Z \to \mathbb{R}$ summarizes the evidence in the observational data with respect to the $H_0 : X \perp\!\!\!\perp Y|Z$ with a real-valued scalar which is a dependence measure. Its value from observed data, compared to a defined threshold (significance level - α) under H_0 then determines a decision of whether or not to reject the null hypothesis H_0. Hypotheses tests can fail in two ways:

1. Type I error : rejecting H_0 when it is true.
2. Type II error : not rejecting H_0 when it is false.

A significance value, α (usually at 0.05) provides the acceptable threshold for type I error. When the type-I error is within α, the test is said to be calibrated.

3.1 Asymptotic and Exact tests

In hypothesis testing, one generally has two broad choices: an asymptotic test or an exact test. In asymptotic tests, the properties of the estimator and tests are evaluated approximately for large sample sizes (under the limit of samples $n \to \infty$). A p-value that is then calculated using an approximation to the true distribution is called an asymptotic p-value. In exact tests, if the null hypothesis is true, then all assumptions made during the derivation of the distribution of the test statistic are met (Wikipedia [2022a]). Thus, a p-value calculated using the true distribution is called an exact p-value. Using an exact test provides a significance test that maintains the type I error rate of the test at the desired significance level α of the test. In contrast, an approximate test maintains the desired type I error only approximately. This approximation may be made close to α by making the sample size sufficiently large. Thus, for large sample sizes, the exact and asymptotic p-values are very similar, while for smaller sample sizes they can be quite different and can lead to different conclusions about the hypothesis of interest.

There are different types of categorical data tests available that can be classified under the above two cases. Here, we name a few.

- Asymptotic tests

  Likelihood ratio test - G^2 test

  Chi-square test (χ^2)

  Multinomial test

- Exact tests

  Permutation test

  Fisher exact test

  Barnard's exact test

  Boschloo's test

We choose one test from each case - G^2 test from asymptotic class and Permutation test from the exact class and evaluate the performance with respect to a Bayesian Network used as a data generating process. We are particularly interested in understanding the type I error control over number of symbols, n_{symbs} and dimensions of Z, D_z.

The G^2 is a general likelihood ratio test in the asymptotic class. Although the χ^2 test is most commonly used, G^2 test approximates to the theoretical chi-squared distribution better. It is being increasingly used for goodness-of-fit

tests as well since it is less sensitive to small cell frequencies compared to χ^2 test.

Permutation tests are a general class of re-randomization based exact tests. They exist for any test statistic, regardless of whether or not its distribution is known. Thus, one is always free to choose the statistic which best discriminates between hypothesis - null and alternative. The only assumption that permutation tests make is that the labels are exchangeable. They are usually more computationally expensive as compared to an asymptotic method.

3.2 The Likelihood Ratio test

(Wasserman [2010]) The likelihood describes the extent to which the samples provide support for any particular parameter value of a distribution. Higher support corresponds to a higher value of a likelihood. The likelihood-ratio test assesses the goodness of fit of two competing statistical models based on the ratio of their likelihoods, specifically one found by maximization over the entire parameter space and another found after imposing some constraint. If the constraint (i.e., the null hypothesis) is supported by the observed data, the two likelihoods should not differ by more than sampling error. Thus the likelihood-ratio test tests whether this ratio is significantly different from one, or equivalently whether its natural logarithm is significantly different from zero.

Suppose that we have a statistical model with parameter space Θ. A null hypothesis is that the parameter θ is in a specified subset $\tilde{\Theta}$ of Θ. The alternative hypothesis is then that θ is in the compliment of $\tilde{\Theta}$, i.e $\Theta \backslash \tilde{\Theta}$, denoted by $\tilde{\Theta}^c$. The likelihood ratio test statistic for the null hypothesis $H_0 : \theta \in \tilde{\Theta}$ is given by,

$$\lambda_{LR} = -2 \ln \frac{sup_{\theta \in \tilde{\Theta}} L(\theta)}{sup_{\theta \in \Theta} L(\theta)} = 2 \ln \frac{L(\theta)}{L(\tilde{\theta})} \tag{3.3}$$

a ratio of the maximum likelihood estimate (MLE) and the MLE when θ is restricted to lie in $\tilde{\Theta}$

3.2.1 G^2 test

G^2 test of independence is a likelihood ratio test. It is also known as the log-likelihood ratio test, or the G-test. It belongs to the class of asymptotic tests of independence. We can derive the general formula for the G^2 test from the log-likelihood ratio in equation 3.3, where the underlying model is a multinomial model (Wikipedia [2022b]). Suppose we have random variables

$X = (X_1, X_2, ..., X_n)$ and we consider a sample $x = (x_1, x_2, ..., x_n)$ where each x_i is the number of times an object i was observed, i.e, the entries of a contingency table. The probability mass function for the multinomial distribution is given by,

$$P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \frac{N!}{\prod_{i=1}^{n} x_i!} \prod_{i=1}^{n} \theta_i^{x_i} \tag{3.4}$$

where, x_i are non-negative integers such that $N = \sum_{i=1}^{n} x_i$ is the total number of objects observed. θ_i are constants with $\theta_i > 0$ and $\sum_{i=1}^{n} \theta_i = 1$. This would be the likelihood function of $L(\hat{\theta})$ and $L(\tilde{\theta})$.

Then, the test statistic in equation 3.3 is then,

$$2 \ln \left(\frac{L(\hat{\theta}|x)}{L(\tilde{\theta}|x)} \right) = 2 \ln \left(\frac{\prod_{i=1}^{n} \hat{\theta}_i^{\,x_i}}{\prod_{i=1}^{n} \tilde{\theta}_i^{\,x_i}} \right) \tag{3.5}$$

For a multinomial model, the MLE of $\hat{\theta}_i$ given some data is defined by $\hat{\theta}_i = \frac{x_i}{n}$ and we may represent each null hypothesis parameter $\tilde{\theta}_i$ as $\tilde{\theta}_i = \frac{e_i}{n}$

This in equation 3.5 leads to

$$2 \ln \prod_{i=1}^{n} \left(\frac{x_i}{e_i} \right)^{x_i} = 2 \sum_{i=1}^{n} x_i \ln \left(\frac{x_i}{e_i} \right) \tag{3.6}$$

(Tsamardinos and Borboudakis [2010a]) The general formula for G^2 in case of test of independence between X and Y is

$$G^2 = 2 \sum_{xy} N_{xy} \ln \left(\frac{N_{xy}}{E_{xy}} \right) \tag{3.7}$$

N_{xy} = observed frequencies of $X = x, Y = y$.
N_x = marginal total of $X = x$.
N_y = marginal total of $Y = y$.
N = Total sample size.
$E_{xy} = \frac{N_x N_y}{N}$ = Expected frequencies under the assumption of independence.

Two likelihoods are estimated - the likelihood of the observed frequencies under a multinomial distribution, and the likelihood if it is assumed that row and column classifications are independent. Twice the natural logarithm of this ratio is equal to G^2. Given the null distribution that the observed frequencies result from random sampling from a distribution with the given expected frequencies, the distribution of G^2 is asymptotically distributed as chi-square

(χ^2) distribution with $f = (|X| - 1)(|Y| - 1)$ degrees of freedom. In practice, the observed and the expected frequencies are derived by constructing a contingency table. The G^2 is preferred over χ^2 because it is less sensitive to small cell frequencies (Finkler [2010]).

G^2 test of conditional independence The above formula can be easily extended to the case of conditional independence, $X \perp\!\!\!\perp Y \mid Z$, where Z can be multivariate $(Z_1, ..., Z_k)$. The test statistic is given by,

$$G^2 = 2 \sum_{xyz} N_{xyz} \ln \left(\frac{N_{xyz}}{E_{xyz}} \right) \tag{3.8}$$

N_{xyz} = observed frequencies of $X = x, Y = y, Z = z$, where $z = (z_1, ..., z_k)$.
N_{xz} = marginal total of $X = x, Z = z$.
N_{yz} = marginal total of $Y = y, Z = z$.
N = Total sample size.
$E_{xyz} = \dfrac{N_{xz} N_{yz}}{N_z}$ = Expected frequencies under the assumption of independence.

The G^2 components over each $Z = z$ are added together to arrive at the final statistic value. The degrees of freedom of the asymptotic χ^2 distribution is then $f = (|X| - 1)(|Y| - 1) \prod_{i=1}^{k} |Z_i|$, where |X| denotes the number of symbols that X can take.

The Chi-squared test is an approximation of the log-likelihood ratio on which G^2 tests are based on. $G^2 \approx \chi^2$ when the observed frequencies are close to the expected frequencies. When the difference is large, the χ^2 approximation begins to break down. Here, the effects of outliers in data will be more pronounced, and this explains why χ^2 tests fail in situations with little data. For samples with reasonable size ($\tilde{1}000$), the G^2 and χ^2 tests will lead to the same conclusions. However, the approximation to the theoretical χ^2 distribution for the G^2 test is better that the Pearson's chi-square test. It is not hard to show that G^2 can also be expressed in terms of mutual information.

3.2.2 Sparse contingency tables

A contingency table records the multivariate distribution of two or more discrete random variables in an N-dimensional matrix. Each cell represents a joint outcome of all the variables and has an associated joint probability. We assume that the underlying population for this contingency table is described by a multinomial distribution. The G^2 test analyses the deviation of these observed frequencies from the expected frequencies to produce a test statistic

that captures association between random variables and consequently independence or dependence through a hypothesis test. One common issue is sparsity (or zeros) in a contingency table. Sparsity can be classified into two types - structural zeros and random zeros. A G^2 test is valid only when the expected frequencies for each cell are at least 5. Therefore, we need to account for sparsity by applying corrections in order to use the test reliably.

A structural zero or a fixed zero is when it is impossible to observe values for certain combinations of the variables. This can be a constraint from the underlying physical process or structural causal model that produced the data. A random zero or sampling zero is when the cell probability is positive but yet no observations are present. This can be due to sampling variations or sparse samplings as in the case of sensor data from field. Random zeros are typical when sample sizes are low and/or when the dimension of the data is very high and therefore not all cells in a contingency table are filled. The G^2 test is not valid in the case of structural zeros. Henceforth, assuming that there are no structural zeros, we look into the correction when random zeros occur.

Correction A common heuristic applied is to reduce the degrees of freedom by one for each expected cell count that is zero in a contingency table. This comes from the intuition that if the expected cell count for a cell is zero, then at least one of the marginals corresponding to that cell in the observed table must be zero. This is only possible when all the entries in the observed table corresponding to that marginal are zero. Therefore, this reduction in degrees of freedom would adjust the shape of the χ^2 null distribution by shifting towards zero and avoid over acceptance of the null hypothesis H_0 where $H_0 : X \perp\!\!\!\perp Y|Z$ and therefore avoid inflated false negatives. This however only helps in improving the power of the test. This is the heuristic used in our work. Say random variable X has n symbols and random variable Y has m symbols each. The degree of freedom $dof = (n-1)*(m-1)$. If a marginal corresponding to a category in Y is zero, then the dof is reduced as $dof = (n-1)*(m-1-1)$.

The above heuristic was suggested by Ku [1963], Bishop et al. [2007] and is followed by Peter Spirtes and Boomsma [1996], Kalisch et al. [2012] and by Tsamardinos and Borboudakis [2010a]. Although Neapolitan [2003] also quotes the same heuristic referring to Peter Spirtes and Boomsma [1996], the example (10.39 on page 601) seems to follow a slightly different variant of reducing the degree of freedom by one for every observed zero count. This falsely implies that the corresponding expected count is zero and would lead to over rejection. Bishop et al. [2007] also note that an exact general rule for calculating the reduction of degrees of freedom given cells with zero entries seems not to be known. Baker et al. [1985] argue against reduction of degrees of freedom. They argue that the expected value is a population parameter which

is unaffected by the random zeros. Brzezińska [2015] surveys other heuristics including a modification of the Yates's correction for continuity (Yates [1934]) and variants of adding small values to each cell. There is also a need for consideration when the frequencies are very small but non zero, producing biased estimates.

3.3 CMISymbPerm - Conditional Mutual Information based permutation test

Testing for conditional independence is particularly challenging when analytical expressions for null distribution are not available or are approximated only for large samples as in G^2. Here, we propose a design of an exact CI test based on Conditional Mutual Information (CMI) combined with a local permutation scheme for generating the null distribution - CMISymbPerm. The test statistic used for categorical data is a CMI based on bincount histogram for contingency table generation while the local permutation scheme is intended to generate the null distribution accounting for Z. We later run experiments comparing CMISymbPerm with G^2 test statistic, where G^2 assumes the null to be χ^2 distributed.

Within a causal discovery algorithm, H_0 true signifies independence and consequently no causal link between the variables, and H_0 being false signifies dependence under further assumptions. Hence, performance of the CI test has a direct impact on that of the causal discovery algorithm. These algorithms also often make use of the test statistic's value, for example to sort the order in which the conditions are tested. The CMI estimate here readily allows for an interpretation in terms of the relative importance of one condition over another (Runge et al. [2019]).

3.3.1 Conditional Mutual Information

Conditional Mutual Information (CMI) is the expected value of the mutual information between two random variables X and Y, given the third Z with a joint probability distribution $p(x, y, z)$, defined as

$$\begin{aligned} I_{(X;Y|Z)} &= \sum_{x,y,z} p(x,y,z)\, log\left(\frac{p(x,y|z)}{p(x|z)p(y|z)}\right) \\ &= H(X|Z) - H(X|YZ) \\ &= H(XZ) + H(YZ) - H(XYZ) - H(Z) \end{aligned} \tag{3.9}$$

where H denotes the Shannon entropy assuming that the densities $p(\cdot)$ exist. This measure is then used to test the hypothesis in (3.1). The CMI $I_{X;Y|Z} = 0$ if $X \perp\!\!\!\perp Y|Z$ given that the densities are well defined. Here, p(.) is approximated through contingency tables formed using bincount histogram. This is a frequentist approach and hence the approximated $p(\cdot)$ depends on the number of samples.

3.3.2 Permutation scheme for null distribution approximation

Permutation test (Wasserman [2010]) A permutation test is a non-parametric method that involves two or more samples. This is an exact test, meaning that it is not based on large sample approximations and hence works for lower samples as well. Consider the two sample test. Here, we explain the concept of permutation test using the two sample test. The null hypothesis is that all samples come from the same distribution. Suppose that $X_1, ..., X_m \sim F_X$ and $Y_1, ..., Y_n \sim F_Y$ are two independent samples and H_0 is the hypothesis that the two samples are identically distributed.

$$H_0 : F_X = F_Y$$
$$H_1 : F_X \neq F_Y$$

Let $T(x_1, x_2, ..., x_m, y_1, y_2, ..., y_n)$ be some test statistic. Consider forming all $N!$ permutations of the data $X_1, ..., X_m, Y_1, ..., Y_n$ where N = m + n. For each permutation, compute the test statistic $T = T_1, ...T_{N!}$. Under the null hypothesis, each of these values are equally likely. This null distribution P_0 is uniformly distributed and is called the permutation distribution of T. If t_{obs} is the observed test statistic, the p-value is then

$$p - value = P_0(T > t_{obs}) = \frac{1}{N!} \sum_{j=1}^{N} I(T_j > t_{obs})$$

Since it is not usually practical to calculate all $N!$ permutations, we approximate the p-value by sampling randomly B permutations (or surrogates) from the set of all permutations. The fraction of the times $T_j > t_{obs}$ among these surrogates approximate the p-value. This method is also called surrogate data testing and is show in Algorithm 5.

This principle is adapted to the case of hypothesis test of conditional independence. This is an effective approach in case of unconditional independence, $X \perp\!\!\!\perp Y$ where the independence is simulated by randomly permuting all x-values in the data. In the case of conditional independence $X \perp\!\!\!\perp Y \mid Z$,

Algorithm 5 A general algorithm for permutation test

1. Compute the observed test statistic $t_{obs} = T(X_1, ..., X_m, Y_1, ..., Y_n)$.
2. Randomly permute the data. Compute the statistic again using the permuted data.
3. Repeat the previous step B times and let $T_1, T_2, ...T_B$ denote the resulting values.
4. The approximate p-value is

$$\frac{1}{B}\sum_{j=1}^{B} I(T_j > t_{obs})$$

we need to preserve the respective dependencies of X and Y with Z before permuting them. In the case of discrete data, this can be done by creating subspaces by grouping the samples within multivariate Z, called neighbors, and then randomly permuting the x_is within the neighbors having the same Z_i. Furthermore, a list keeps track of the already 'used' indices of neighbors while permuting within the groups of Z_i in order to achieve (approximately) sampling without replacement. This is the principle behind CMISymbPerm. This (1) preserves the marginals (2) helps in ensuring that the dependence is not re-preserved (in case of $X \not\perp\!\!\!\perp Y|Z$) in some of the samples if neighbors are reused for permuting. However, this cannot always be ensured - for example, when there is only one neighbor present for many samples. The performance of this permutation scheme in generating the H_0 particularly depends on the number of neighbors available for permuting, which in turn depends on the dimensionality of Z, number of samples and number of symbols of each variable. Their interdependencies are shown in the experiments section.

Algorithm 6 Permutation based conditional independence test - CMISymbPerm

Require: Data $\{x_i, y_i, z_i\}_{i=1}^{n}$, number of permutation surrogates B, CMI estimator $\hat{I}(x; y|z)$

Estimate $\hat{I}(x; y|z)$ of $\{x_i, y_i, z_i\}_{i=1}^{T}$

Compute list of neighbors for each sample point i: $N_i = \{l \in 1, 2, ..., n\} : z_l = z_i$ in the subspace of z. List for each sample is of varying length $k_i = [1, T]$.

for all $b \in 1, 2, ..., B$ **do** Initialize empty list U = {} of used neighbors

  Initialize empty array x^* of length T

  Shuffle lists N_i separately for each i

  Create random permutation π of $\{1, 2, ..., T\}$

  for all $i \in \pi$ **do**

    $j = N_i(0)$

    $m = 0$

    for $j \in U$ and $m < k_i - 1$ **do**

      $m = m + 1$

      $j = N_i(m)$

    end for

    $x_i^* = x_j$

    Add j to U

  end for

  Compute $\hat{I}_b = \hat{I}(x^*; y|z)$ of $\{x_i^*, y_i, z_i\}_{i=1}^{T}$

end for

Compute p-value by $p = \frac{1}{B} \sum_{b=1}^{B} [\hat{I}_b \geq \hat{I}(x; y|z)]$

return p and test statistic value $\hat{I}(x; y|z)$

As described in Algorithm 2, the null distribution is estimated by applying the CMI estimator on the permuted surrogates and the p-value is derived as the fraction of surrogate CMIs larger or equal than the CMI of the original data. The CMI estimator holds for arbitrary dimensions of X, Y, Z. The local permutation scheme can be used to jointly permute multivariate X.

Chapter 4

Experiments and results

Here we discuss the experimental setup and the results. First we describe the experiments that map the relation of the designed link strength with the dependence measure (conditional mutual information). We then evaluate the performance of the two conditional independence tests with two models using Bayesian Networks. Here, we discuss first the performance of the permutation scheme in CMISymbPerm in generating the null distribution and its dependence over number of symbols n_{symbs} and number of samples N which in turn influence the number of neighbors available to permute from. We then compare the performances of CMISymbPerm and G^2 test statistics in error controls and time complexity. We investigate results over different parameters - number of samples N, number of symbols n_{symbs} and number of dimensions of Z, D_z. We compare the tests over the control of type I error at α, power of the test $1 - \beta$, as well as the time complexity of the method. All the experiments are run on the computer cluster.

True underlying models Two Bayesian Network models are used in the data generating process. The experiments denote the two models with the parameter c.

- $X \perp\!\!\!\perp Y \mid Z$, denoted as $c = 0$.
- $X \not\perp\!\!\!\perp Y \mid Z$, denoted as $c = 1$.

4.1 Link strength and CMI

We look at link strength η defined in chapter 2 and dependence measure (CMI) defined in chapter 3 for two Bayesian Networks (c = 0 and c = 1). We perform a visual analysis to evaluate that η for a link reflects appropriately in the

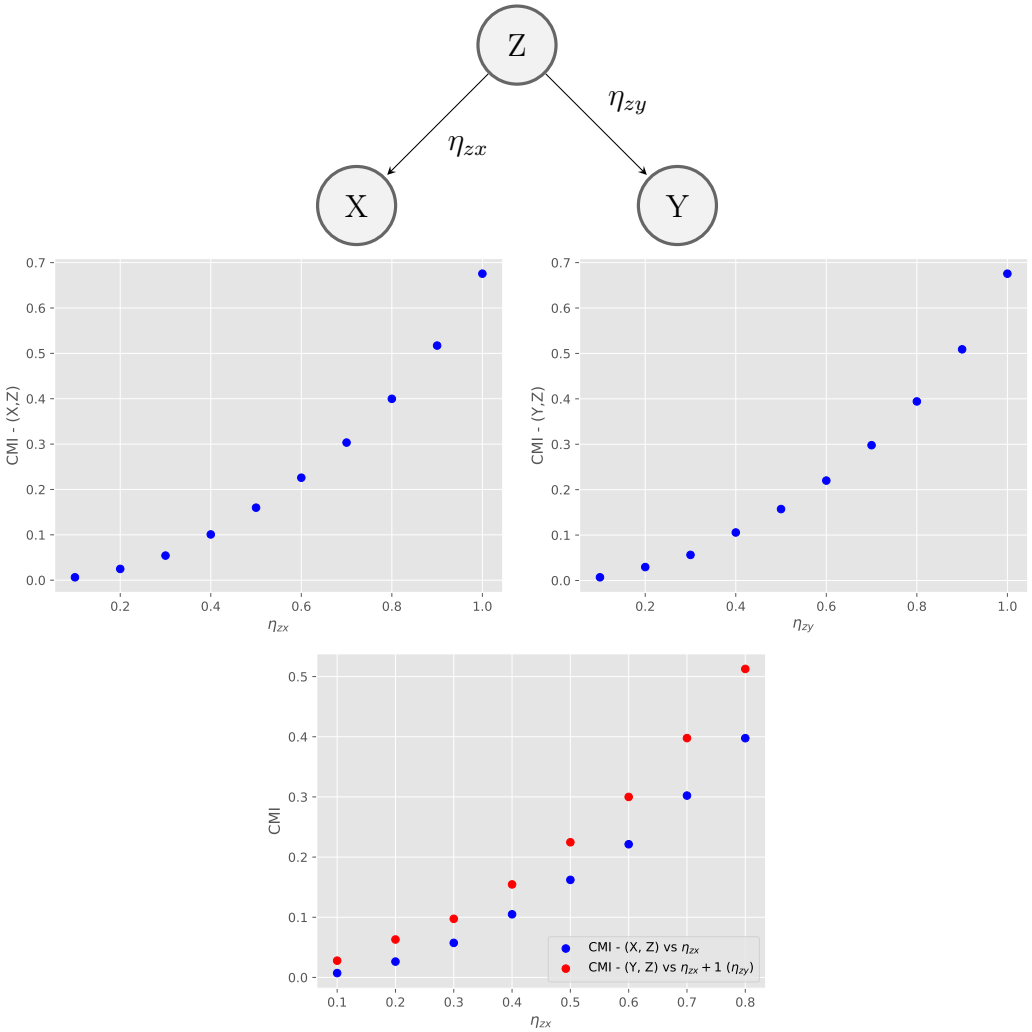

Figure 4.1: Model 1 : $X \perp\!\!\!\perp Y \mid Z$

CMI under different conditioning sets. Although an analytical formula can be derived that quantifies the relation, an empirical evaluation suffices for our needs which is to automatically elicit conditional probability tables and to control the relative effect sizes. We look at deriving an analytical formula in future work.

The model 1 in figure 4.1 shows the case $c = 0$. The first two scatter plots show a nonlinear increase in CMI over increasing link strengths η_{zx} and η_{zy} in range [0, 1] respectively. The third scatter plot shows when η_{zy}=$\eta_{zx} + 0.1$.

The model 2 in figure 4.2 shows the case $c = 1$. We make the following observations,

- In the first two plots, we fix $\eta_{xy} = 0.7$, and test the effect of Z on X and

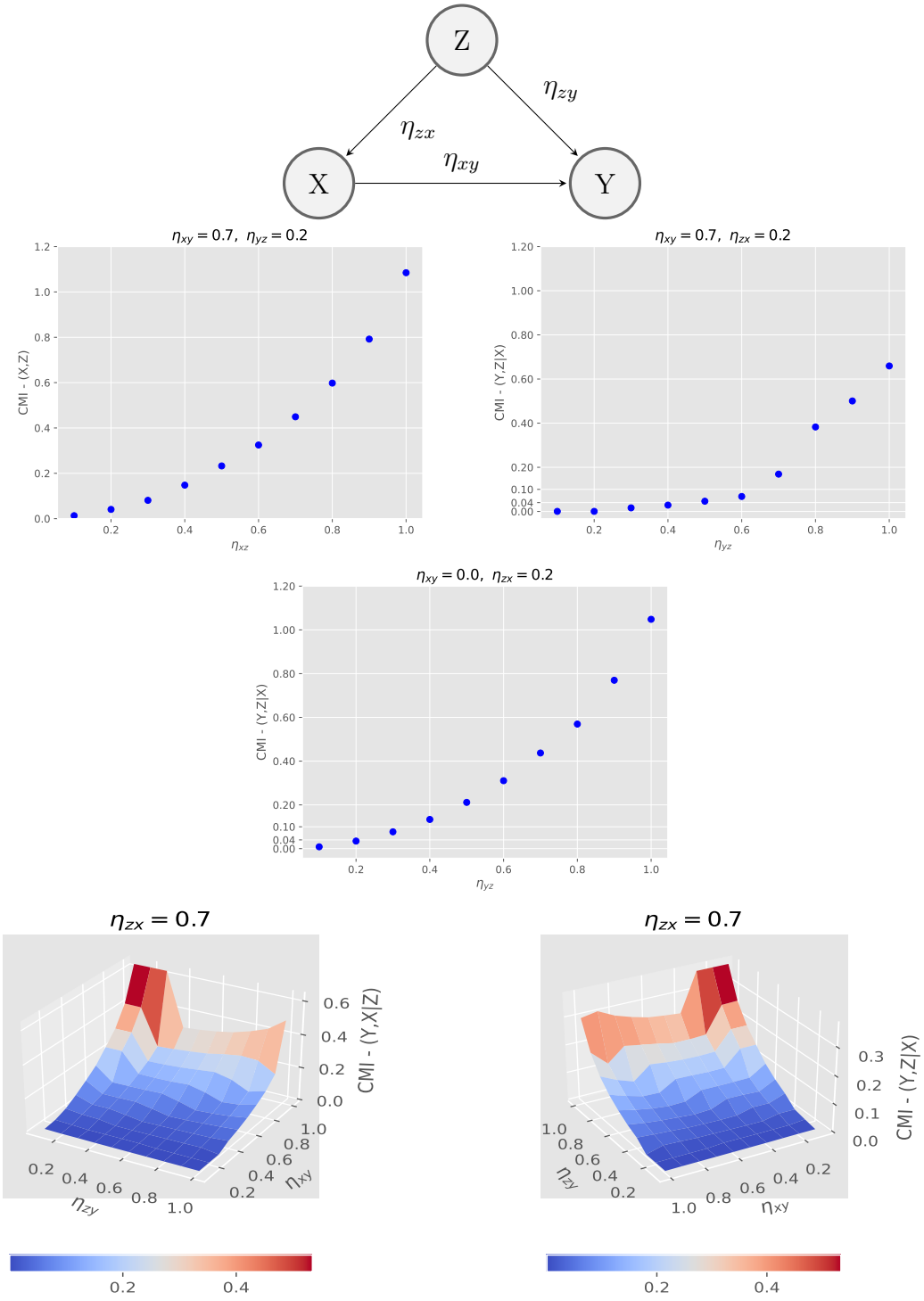

Figure 4.2: Model 2 : $X \not\perp\!\!\!\perp Y \mid Z$

on Y. The first plot shows increase in CMI(X, Z) over increasing η_{zx}. The link η_{xy} and η_{zy} has no effect on CMI(X, Z) as intended.

- The second plot shows CMI(Y, Z | X). The effect of Z on Y is prominent only after $\eta_{zy} > 0.7$. This is because, since $\eta_{xy} = 0.7$ and $\eta_{zx} = 0.2$, conditioning on X also conditions parts of Z. Hence, up until $\eta_{zy} = 0.7$, Z has an effect on Y through X which gets conditioned out, explaining the lower CMI levels.

- The third plot shows when $\eta_{xy} = 0$. This is equivalent to an absent link between X and Y. Thus, plot of CMI(Y, Z | X) over η_{yz} is equivalent to CMI(Y, Z).

The two 3D surface plots show the effect on Y from X and Z individually. We fix the link $\eta_{zx} = 0.7$, a relatively strong link, and vary the other two links η_{zy} and η_{xy} over [0, 1]. The 3D surface plots visualizes the CMI over these two link strengths. The nonlinear relationships of CMI over any of the two dimensions of η can be observed.

- The first 3D plot shows CMI(Y, X | Z). For a fixed $\eta_{zx} = 0.7$, we make the below observations.

  - When η_{zy} is low, most of the effect on Y is from X, and hence the CMI(Y, X | Z) increases non-linearly with increasing η_{xy}. This is represented by the part of the graph with steep rise to red.
  - When η_{zy} increases, since Z also causes X (through a fixed strong η_{zx}), conditioning on Z also conditions out parts of X. This reflects as overall decreasing CMI(Y, X | Z) over increasing η_{zy}. This is shown by the red to orange part of the graph.
  - The effect of Z conditioned out through X on Y also reflects for when η_{zy} is high and η_{xy} varies, which explains the very shallow increase from blue to orange in the graph.
  - When η_{xy} is low, since we look at CMI(Y, X | Z), most of the effect of Z on Y is conditioned out and there are no effects of Z on Y through X at low η_{xy}. This reflects as constant very low CMI(Y, X | Z) over all η_{zy}, represented by the dark blue.

The second 3D plot shows CMI(Y, Z | X). The observations here are similar to the ones made for CMI(Y, X | Z). All the observations over varying link strengths for the two models are as expected from the design of link strengths.

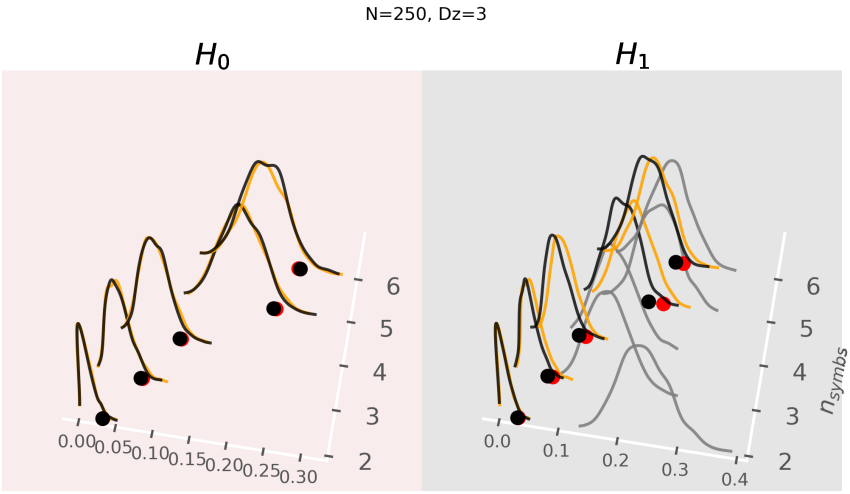

Figure 4.3: H_0 approximated by CMISymbPerm

4.2 CMISymbPerm

4.2.1 CMISymbPerm in approximating the null distribution

Figure 4.3 shows the performance of CMISymbPerm in approximating the null distribution. The plot shows the Gaussian approximations of the CMI test statistic over B=1000 surrogates, $n_{symbs} = 3$ symbols and $T = 250$ samples. The orange curves represent the true H_0 and the grey curves represent the true H_1, derived from the Bayesian Network models c = 0 and c = 1 respectively. The black curves represent the permuted null distributions. While in the pink tile, the H_0 is permuted from true H_0, in the white tile the H_0 is permuted from true H_1. The goodness-of-fit of the black curves with the orange curves show the effectiveness of the permutation scheme. The orange curves in the white tile is shown to compare how close the approximation of permuted H_0 is, as compared to the true H_0 when permuting from H_1. The 95% quantiles are represented by the red (for true H_0) and black dots (for permuted H_0) respectively. The Appendix figure A.1 shows the grid of plots with each column showing varying number of samples = [50, 2000] while each row shows varying dimensions of Z, D_z=[1, 4].

The permutation test highly depends on the number of neighbors available within the subspace of Z for permutation. More the number of neighbors, higher the choices to permute from and closer to achieving sampling without replacement ensuring that the ties between X and Y are broken. Therefore, high samples (> 250), lower number of symbols (< 3) and lower dimensions of Z

(< 3) are ideal for the permuted null to approximate the true null distribution. The figure in 4.3 shows the case corresponding to $D_z = 3$ and $T = 250$. The approximation of the null distribution improves with number of samples, as can be seen in the appendix figures A.1.

4.2.2 Performance of CMISymbPerm test over n_{symbs}, N and D_z

Figure 4.4 shows the error controls at different n_{symbs} and N. Each row shows the 3D plots of false positive rate (FPR) and true positive rate (TPR) over N and n_{symbs} for a particular D_z in [1, 4]. The left column shows FPR while the right shows TPR. The plots of FPR shows that the permutation scheme is largely effective in type I error control at α (shown by the grey frame at 0.05), especially for $N > 250$ across n_{symbs} and D_z. The FPR shoots up at $N < 100$, $n_{symbs} > 3$ and $D_z > 2$. This is because the permutation scheme is highly dependent on the number of neighbors available. Lesser the number of samples and more number of symbols leads to lower number of neighbors to permute from. This approximates the null farther from the true null leading to misclassification in the hypothesis test. This reflects in FPR and TPR. One can see across plots that increased FPR translates to reduced TPR (or power). Increased sample size improves error control. This is especially true when the $D_z > 3$, where the combination of symbols in Z are well represented by higher samples. TPR starts to improve beyond $N > 100$ for $D_z < 3$ even for $n_{symbs} > 4$, while larger $D_z = 4$ requires $N > 1000$ for higher $n_{symbs} > 4$.

4.3 Comparison of CMISymbPerm and G^2 in CI test

4.3.1 Over N and D_z with fixed n_{symbs}

Figure 4.5 shows the grid of plots comparing CMISymbPerm with G^2 test statistics. The comparison is done in terms of false positive rate (FPR), true positive rate (TPR) and time complexity. For a fixed $n_{symbs} = 3$, each row shows the plots over varying N for a particular D_z. The D_z is varied from [1, 4]. The figures for different n_{symbs} is shown in Appendix A.2, A.3, A.4, A.5. As seen in the previous plot, CMISymbPerm controls the FPR at α across D_z, while G^2 performs poorly in comparison. G^2 is particularly poor for lower samples of $N < 250$. This is because the χ^2 distribution derived from dof does not approximate the true null well at these parameter ranges which leads to G^2 over rejecting H_0 while permutation scheme with 1000 permutations

is data adaptive approximating the null better. This over rejection reflects in higher TPR. The FPR of the both the tests converge only at large sample sizes $N > 1000$, but only for lower dimensions of $D_z < 3$. However, for $D_z > 3$, CMISymbPerm still performs better. CMISymbPerm is comparable to G^2 in power for $D_z < 3$. The TPR of both tests converge when the effective sample size grows, especially for $D_z > 3$.

The performance of CMISymbPerm comes at a significant cost in time complexity mainly due to permutation scheme used for null distribution generation. Since we look for neighbors in subspace of Z for each sample, the test exponentially grows in time complexity with sample size N. A minimum of $N = 100$ samples are required for the tests to work reliably.

Figures A.2, A.3, A.4 and A.5, show these comparisons for $n_{symbs} = [2, 4, 5, 6]$. Across all parameters, CMISymbPerm consistently performs at FPR control at α as compared to G^2. The two tests are only comparable in FPR at $n_{symbs} <= 3$, $D_z <= 2$ when $N > 100$, and in case of $n_{symbs} = 4$, $D_z <= 2$ but for $N > 500$. Therefore G^2 could be the choice of test in these ranges since the time complexity is $\mathcal{O}(n) = 1$. Over $n_{symbs} > 3$ and $D_z > 2$, CMISymbPerm is most reliable across N. In terms of power, CMISymbPerm matches to G^2 only at approximately $N > 500$. CMISymbPerm is extremely conservative in rejection of H_0 as compared to G^2. G^2 can also be preferred when time is a significant factor or when the number of conditioning sets in a causal discovery algorithm is very high. In all other cases, CMISymbPerm could be the choice for a better calibrated test.

4.3.2 Over D_z and n_{symbs} with fixed N

Figure 4.6 shows the performance over D_z for $N = 250$ and $n_{symbs} = [2, 3, 4]$. Here, we particularly choose standard sample size. CMISymbPerm is consistent across D_z, while G^2 quickly blows up in FPR with increasing D_z. The TPR for both the tests is comparable until $D_z = 3$ and dips at higher D_z with G^2 less as compared to CMISymbPerm (as also seen in previous figure). This improves as the effective sample sizes over higher D_z and n_{symbs} increase.

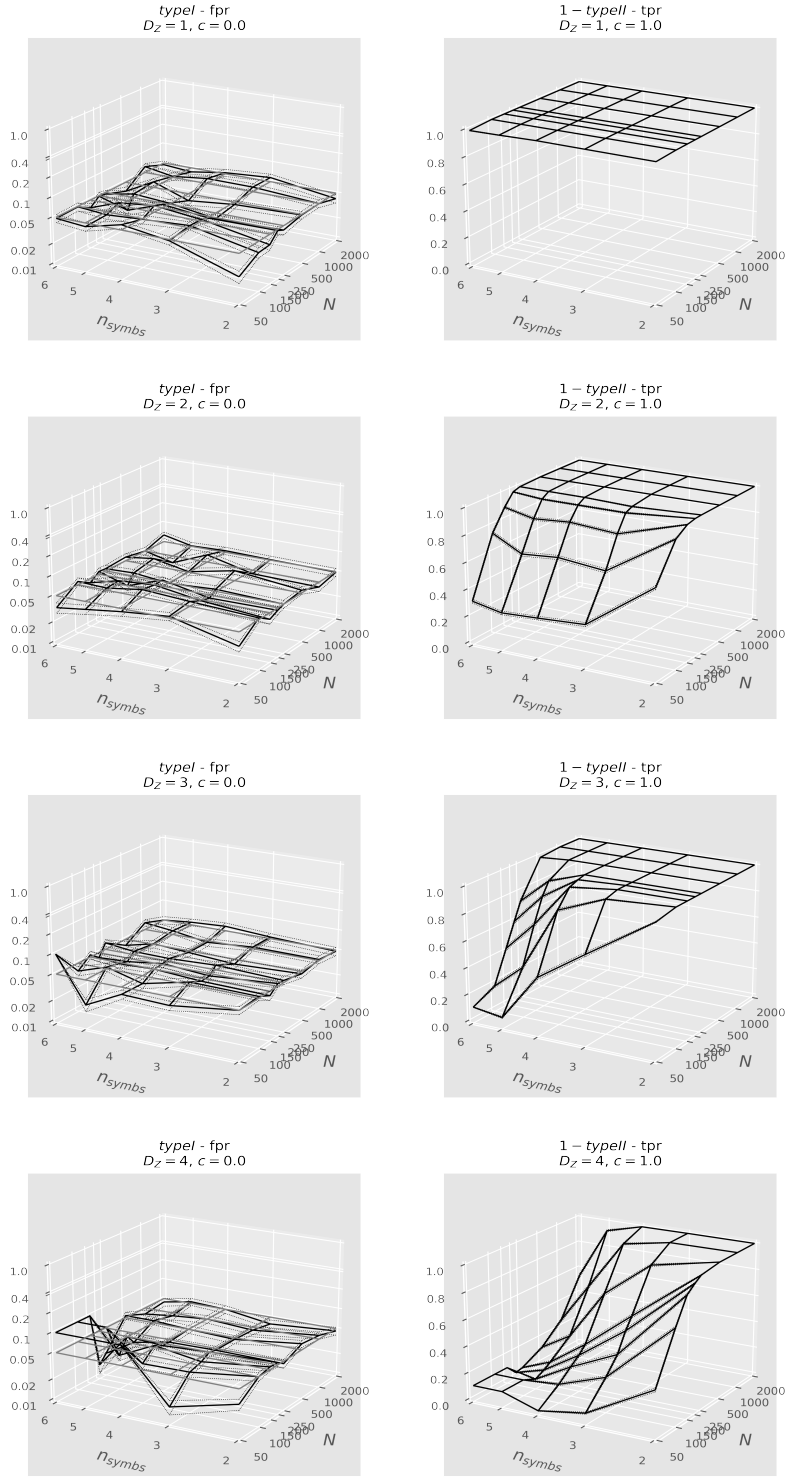

Figure 4.4: False Positive Rate and True Positive Rate

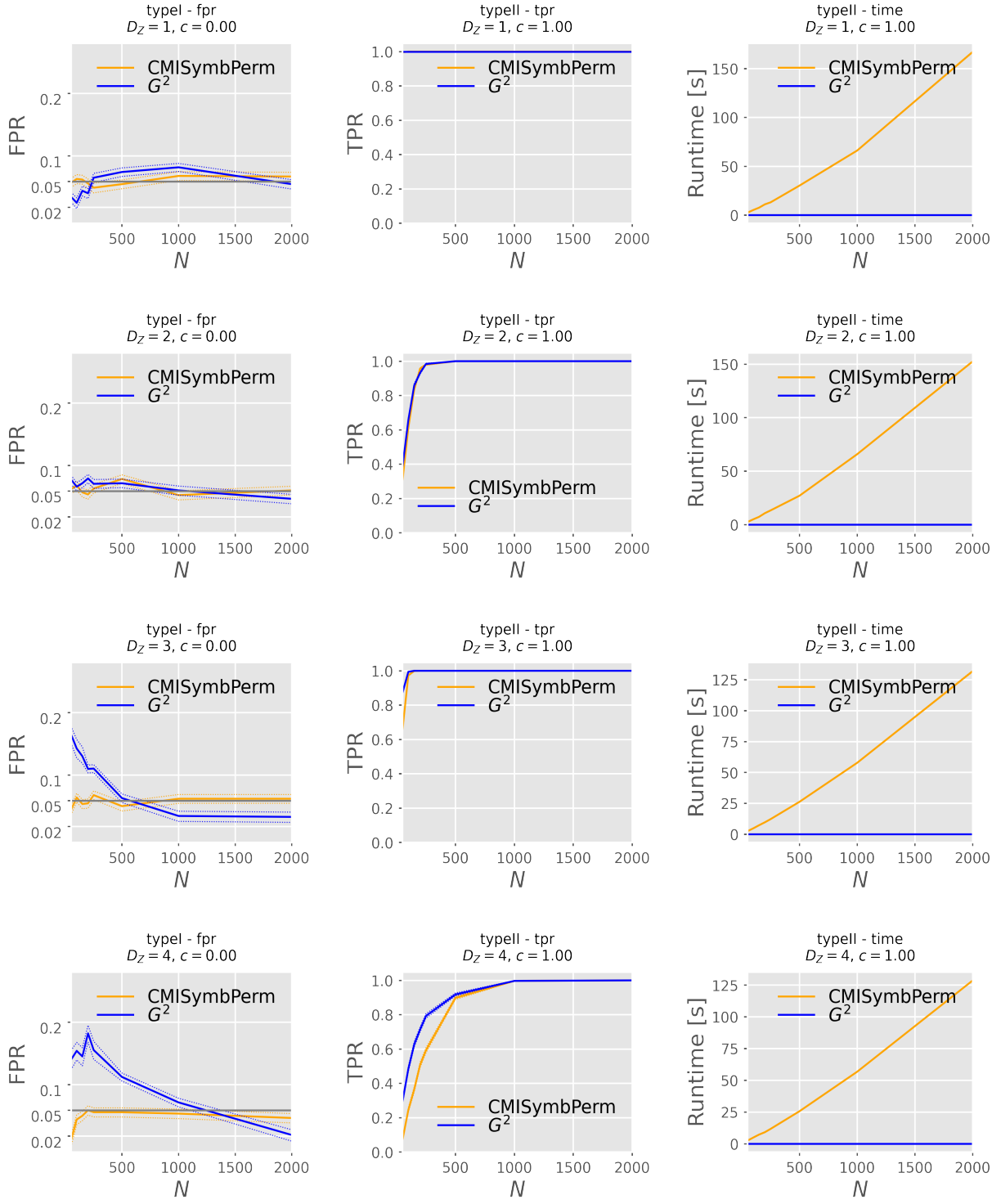

Figure 4.5: Over N and D_z with $n_{symbs} = 3$

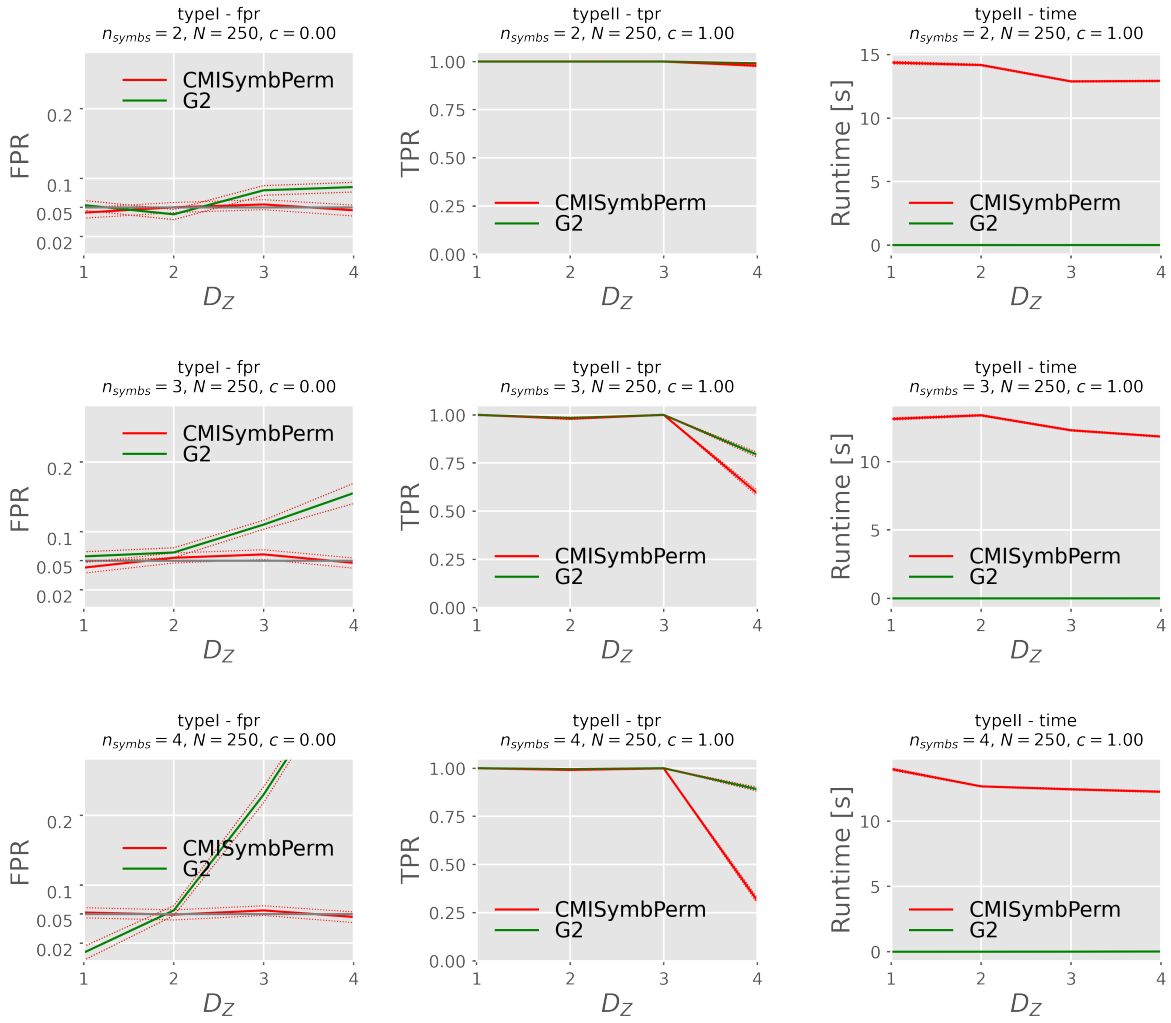

Figure 4.6: Over D_z with $N = 250$ and $n_{symbs} = 2, 3, 4$

Chapter 5

Causal Discovery - Bibliometrics

This analysis is part of a project on social network analysis of researchers. One of the tasks in the project is to analyze publication characteristics of researchers. This requires access to dataset of publications. The aim was initially to parse web profiles of researchers and attempt to categorize them using bibliometrics. Extracting unstructured content from web is complex. Hence, in this work, we restrict to structured data, specifically Open Academic Graph 2.1, a knowledge graph with publications curated from 1800s to the present. We parse this knowledge graph to curate data for each researcher, and perform exploratory data analyses including causal analyses. The dataset contains around 240 million papers by 243 million authors published in around 53 thousand venues across 25 thousand affiliations, all amounting to approximately 500 GB dataset. We indexed this large dataset onto Elasticsearch [2015] (ES), a scalable search and analytics engine hosted at Webis, Bauhaus University Weimar. ES helps manage such large datasets as indices over computer clusters through APIs. We leverage ES's tool 'Kibana' to visually explore the dataset through full Query DSL (Domain Specific Language) based on JSON. We then use these DSL queries along with pandas (McKinney et al. [2010]) to curate a set of 11 useful features for approximately 4800 researchers. The original intention was to look into subset of categorical data to apply CMISymbPerm. But data curation phase yielded that continuous valued features were more reliable in the dataset.

In this work, we curate the features with a pivot on the 'position' of a researcher (professor/associate professor/assistant professor) and analyze the bibliometric features that influence a particular category of researcher. We employ stratified sampling strategies to curate a total of 10 features for about 4800 researchers with around 1600 samples for each category in position. We then build a multinomial logistic regression classifier that can predict the classification labels in position based on the features. We further analyze the model

to list the dominant features using the method of *permutation feature importance*. We then use PC algorithm described in Algorithms 1, 2, 3, a prominent causal discovery algorithm to build causal graphs with the same features over all researchers as well as for each subgroup in position. We then compare the features from the two approaches and understand the intra-dependencies through causal graphs. The causal analysis reasons out confounders, mediators and selection biases that are not apparent from a logistic regression model.

5.1 Open Academic Graph 2.1

The Open Academic Graph (OAG) [1] was generated by linking two large academic graphs: Microsoft Academic Graph (MAG) and ArnetMiner (AMiner). The dataset collects OAG 2.1 generated in 2020. The two large graphs are both evolving and here, the MAG July 2020 and AMiner October 2020 snapshots are considered. We leverage a mix of features across the two datasets identified by linked IDs provided within the dataset.

5.1.1 Data curation

We leverage ES and pandas to curate the data. We first identify the features relevant to each researcher (author schema). We use the field 'author' to refer to a researcher/academician and represents a sample point. The features indexed are,

1. *id* - Author id
2. *name* - Author name
3. *orgs* - Author affiliations
4. *org* - Author organization
5. $last\_known\_aff\_id$ - Last known affiliation ID
6. *position* - Author position
7. n_pubs - Number of author publications
8. $n\_citation$ - Author citation count
9. h_index - Author h-index

[1] https://www.aminer.cn/oag-2-1 (Sinha et al. [2015], Tang et al. [2008], Zhang et al. [2019])

10. *pubs.i* - Author paper ID
11. *pubs.r* - Author order in the paper

We use ES's 'match_phrase' query to only query a subset of the authors that are in position = {professor, associate professor, assistant professor} and have a non-empty field in the rest of the feature list above. Through affiliation ID, we get the location (latitude, longitude) of the organization that the author is affiliated to. We then derive a list of publication related features through the list of paper IDs for each author. After this phase, following removal of outliers we now have non-empty fields for 8000 professors, 2300 associate professors and 1600 assistant professors. We now perform stratified sampling (using random seeds) to obtain datasets having 1600 samples for each position. The total samples amounts to 4800 with 96% of the samples having features within the expected range. The final 11 features for our analysis are,

1. *position* - Author position
2. *n_pubs* - Number of author publications
3. *n_citation* - Author citation count
4. *h_index* - Author h-index
5. *latitude* - Latitude of the affiliation
6. *longitude* - Longitude of the affiliation
7. *avg_pos_authorship* - Average position of authorship across papers
8. *years_active* - Years active in publishing
9. *avg_collaborations* - Average collaborations over papers
10. *avg_citations* - Average citations per paper
11. *lang_count* - Number of languages published in.

5.2 Multinomial logistic regression on *position*

We now build a multinomial logistic regression model with the above 10 features on position using scikit-learn. We intend to build a model that can classify and consequently predict a researcher's position in academia based on the 10 features listed. A multinomial logistic regression predicts the probability of

a category membership on a dependent variable based on multiple independent variables. Here, the independent variables are the continuous valued features and the dependent variable is the field 'position' with three categories. The algorithm uses the one-vs-rest (OvR) scheme where the analysis breaks the dependent variable into a series of comparisons between two categories. We use the cross-entropy loss which is a measure from the field of information theory. It builds upon entropy and generally computes the difference between two probability distributions given a set of events. The input features are min-max scaled and fed to the logistic regression which fits a model. The fit model returns a mean accuracy score by predicting using the model, the input training set on the classifying label. For example, a single run on the above dataset returned 0.263. It means that about 26% of the labels were predicted accurately.

A single run may result in a noisy estimate of model performance. The fit classification models are evaluated using a repeated stratified k-fold cross validation. Different splits of the data may result in different performance. A k-fold cross validation procedure divides the dataset into k non-overlapping folds. Each of the k folds will be used as test set while the rest are used as training set. A total of k models are then fit and evaluated on k test sets. Finally, the mean and the standard deviation of the scores are reported. A repeated k-fold cross validation procedure with k=10 splits, repeated 1000 times yielded a mean accuracy of 0.582 with a standard deviation of 0.02.

Permutation feature importance We now inspect the model to analyse the dominant features of the fitted estimator. The 'permutation feature importance' provides a model inspection technique that is model agnostic. It is defined to be the decrease in a model score when a single feature value is randomly shuffled. The principle is similar to the permutation scheme defined in Algorithm 5 in Chapter 3. The method breaks the relationship between the feature and the dependent variable. The amount of drop in the model score (R^2) is then a quantification of the dependence of the model on the feature. Permutation does not reflect on the predictive power of a feature but only on how important the feature is for a particular model. Here, the technique is applied on the model fit on training data.

The parameter n_repeats=30 sets the number of times a feature is randomly shuffled and returns a sample of feature importances. The dominant features after 1000 repetitions are in Table 5.1.

The top five dominant features remain invariant over different stratified samples. One can reason out the influence of these features on the tenureship of an academician. Longer active years in publishing, higher h-index and publications are common notions related to professorship or promotion in academia.

Feature	Mean score	Standard deviation
years_active	0.132	0.006
n_pubs	0.036	0.005
h_index	0.014	0.003
n_citation	0.011	0.004
longitude	0.010	0.003
avg_citations	0.006	0.002

Table 5.1: Permutation feature importance

But, location (specifically longitude) of a researcher seem to be a significant feature. We would like to understand this further along with a detailed view on the rest of the features.

One can draw parallels between this method and the method of cause and effect estimation where the permutation scheme acts as an intervention on features. We look at this principle formally through the theoretical framework of causal inference, especially constraint-based causal discovery framework such as PC (Glymour et al. [2019]) implemented in TIGRAMITE [2].

5.3 Causal discovery

The PC algorithm uses conditional independence testing to identify d-separated paths. The choice of the CI test highly depends on the nature of data and its interdependencies within variables. All the fields except 'position' are continuous valued. Although some of the fields such as h_index, n_pubs, n_citations etc, can only take integer values, there are no upper bound on the values that the variables can take. Hence, we treat them as continuous valued variables. We analyze the 4800 samples over all positions as well as the subgroups of 1600 samples each within position.

We first perform a pairwise scatter plot to understand the nature of dependencies. The top figure in 5.1 shows the pairwise scatter plot for 10 features listed above (without position). The features are min-max scaled and represented by numbered ids. One can see over the individual plots, that almost all features are linearly dependent, except longitude(3) and latitude(4) that are

[2]https://github.com/jakobrunge/tigramite

non-linearly dependent with most variables. Figure 5.2 shows the Pearson's pairwise correlation matrix. We read this figure and the scatter plot together and make the following observations.

1. h_index has a significant positive correlation with n_pubs and n_citations. This is as expected from the definition of h_index (Bornmann and Daniel [2007] - number of publications h having a citation value h and above. The correlation of h_index with years_active and lang_count has to be investigated if they indeed have a direct causal path or are mediated/confounded by other variables.

2. n_citations and n_pubs has significant correlation to each other. They are also correlated to years_active, avg_citations and lang_count. The high correlation with lang_count points to the previous point of likely mediation to h_index via n_pubs and n_citations.

3. Location (longitude, latitude) is loosely negatively correlated with h_index and citation parameters. This needs to be investigated further.

4. years_active also is correlated to lang_count. One can reason that it is likely for a researcher to publish in more languages over time in academia.

5. avg_colaborations is correlated to n_citations. avg_citations is strongly correlated only to n_citations but loosely to longitude.

6. The figure 5.2 shows the spread of the the locations in the sample. We can note that the samples are mostly from American, European and Asian continents. This perhaps may introduce selection bias and is important to note while analyzing the causal links. Plots of locations over stratified samples within position are also similar.

7. Another possible source of selection bias is in data preparation since samples are based on the pivot on position.

We choose the ParCorr CI test that implements the linear partial correlation. The null distribution is assumed to be Student's t distribution. We now generate a causal graph for the entire 4800 samples (all authors) as well as individual causal graphs for each subgroup in position (1600 each). The parameter pc_alpha is the only tunable parameter of interest. It controls the false positive rate in the CI test. A lower pc_alpha=0.005 makes the test less likely to reject the null hypothesis ($X \not\perp\!\!\!\perp Y \mid Z$), i.e. more likely that a weak link would be removed between any two variables in the causal graph. Since we perform a stratified sampling to feed into CI test, we perform bootstrap with 1000 iterations generating multiple causal graphs for each of the 4 cases.

We then summarize the results and generate the summarized causal graphs in Figure 5.3. The directed arrows '–>' denote the direct causal path. 'o-o' denotes that the collider and orientation rules could not be applied in algorithm 3 (Markov equivalence). 'x-x' denotes that the directionality is undecided due to conflicting orientation rules. The auto correlation of a node is denoted by auto-MCI, represented by the node color. The cross correlation between any two nodes is denoted by cross-MCI, represented in the graph as link color. The presence of a link denotes that the link was frequently present across bootstraps. The cross-MCI is the mean of the test statistic values and the link width is the frequency of a link over bootstraps. The plot for pc_alpha=0.05 is in Appendix in A.6.

5.3.1 Causal interpretations

For all the authors with the causal graph in figure (a) 5.3 and the correlation matrix in 5.1, we note some important observations.

1. There is no direct link between n_pubs and n_citations contrary to 0.56 in correlation matrix. These were confounded by other features. This points to the notion that publishing more does not lead to more citations. Publishing more does not also lead to more average citations directly.

2. h_index is analytically derived from n_pubs and n_citations. This is reflected as two strong directed arrows towards h_index in the causal graph.

3. h_index has no link to avg_citations. This is as expected since h_index accounts only those publications having a citation equal to or above a value. However, the correlation matrix shows a correlation of 0.25.

4. The longitude of the affiliation has a direct link to average citations. Authors from certain affiliations may be cited more which in turn leads to higher citations and higher h_index. Thus, we can observe a causal pathway from longitude of the affiliation –> average citations –> citations –> h-index. This is a mediated path amongst the dominant features in the classifier.

5. More collaborations lead to more citations which then has a path to h_index. More collaborations also lead up to higher authorship position. This is perhaps representative of authors in mentorship roles over years. However, this does not directly lead to more citations.

6. Publishing in multiple languages over years lead to more publications and consequently to h_index. But, does not have direct link to citations, But a correlation of 0.43 can be seen in the correlation matrix.

7. years_active was the most dominant feature as seen in 5.1. The correlation matrix shows the variable highly correlated to h_index, n_pubs and n_citations. But the causal graphs reveal that it only has a direct path to n_pubs and h_index but not to n_citations.

8. These observations may point to years_active, n_pubs, h_index and n_citations being the only direct factors towards classifying a position. A further encoding of position into numerical variable will help understand this further.

In figures 5.3 (b), (c), (d), the causal graphs for each category in position are different. This can be interpreted as varying publishing behaviors. This points to variance amongst the features across categories which might explain the 0.58 mean score of the logistic classifier.

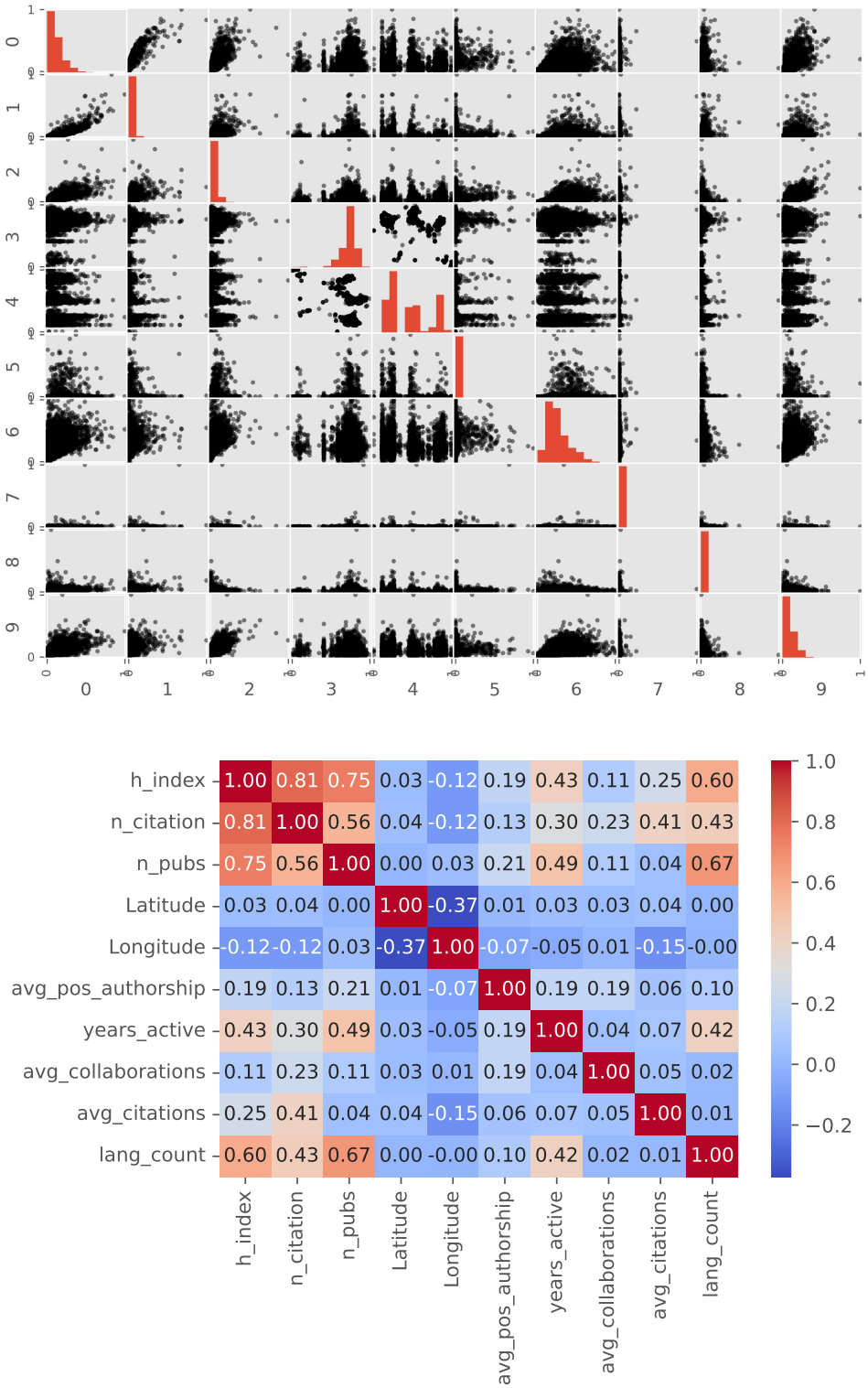

Figure 5.1: Top : Scatter plot of features (Numbered ids correspond to feature names in the bottom plot). Bottom: Correlation matrix

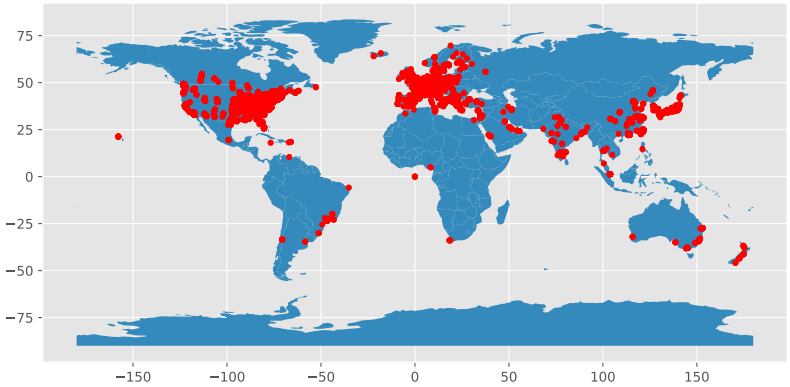

Figure 5.2: Locations of all authors' affiliations in the sample.

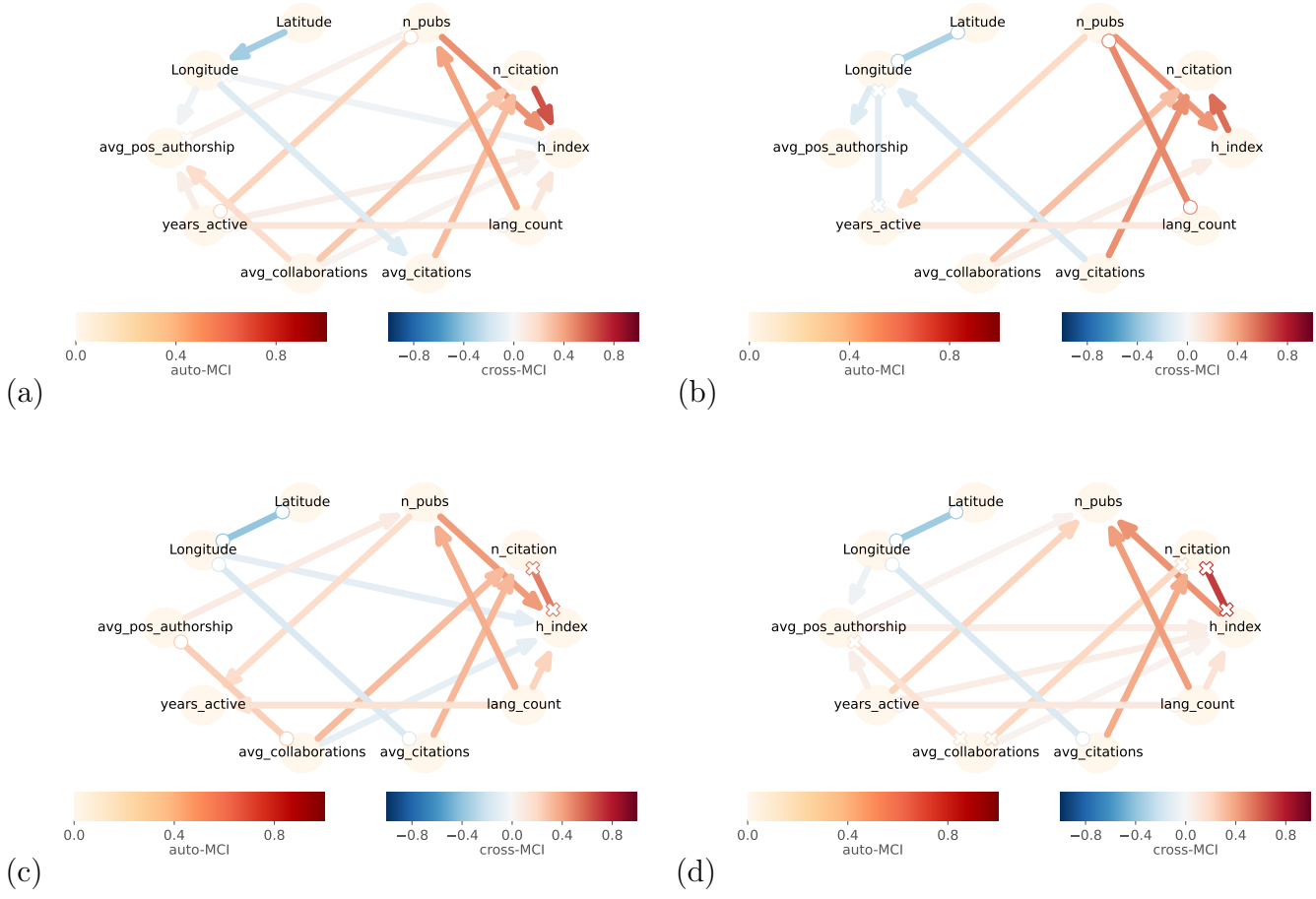

Figure 5.3: pc_alpha=0.005 (a) All authors (b) Professors (c) Associate professors (d) Assistant professors

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Conditional independence testing (CI) is the crucial part of a causal discovery pipeline. In a typical causal discovery framework (The PC algorithm, for example), under the assumptions of sufficiency, faithfulness and markov condition, the skeleton is discovered from a bipartite graph by employing conditional independence tests on multiple sets of variables and consequently eliminating links that are conditionally independent. The skeleton discovered is then passed onto the orientation phase in order to arrive at the CPDAG. Therefore, the design of an appropriate CI test with bounds on type I and type II errors are vital towards causal discovery. We consider the case of categorical data for conditional independence testing. This is formulated as a hypothesis test of independence $X \perp\!\!\!\perp Y \mid Z$. We design CMISymbPerm, a conditional mutual information based permutation scheme which belongs to the class of exact tests. We then evaluated the designed test using Bayesian Network (with a link strength parameter) as a ground truth data generating process. The evaluation is done in terms of FPR and TPR, along with a measure of time complexity.

The null distribution approximation in CMISymbPerm is dependent on number of samples N, number of symbols n_{symbs} and dimensions of Z D_z. We run experiments for $N = 50$ to 2000, $n_{symbs} = 2$ to 6 and $D_z = 1$ to 4 to find the range of these parameters where the CI test is best calibrated at a significance level α. We find that for samples $N > 250$, CMISymbPerm approximates the true null distribution well across n_{symbs} and D_z. We then compare CMISymbPerm with G^2 test statistic, a likelihood ratio based asymptotic test of independence. It uses a χ^2 distribution to approximate the null based on estimated degrees of freedom. The main advantage of this method is that that time complexity is $\mathcal{O}(1)$ as compared to $\mathcal{O}(c^n)$ for CMISymbPerm. However, the approximation is only true for large samples of N. Numerical

experiments show that CMISymbPerm consistently performs in FPR control at α as compared to G^2. The two tests are only comparable at lower dimensions of data and higher sample sizes. We conclude the range of parameters where CMISymbPerm and G^2 can be preferred from experiments.

We then perform causal analysis on Bibliometrics data from Open Academic Graph (2.1). We parse 4800 samples over three positions in academia from 240 million data points for 10 numerical features. We first use multinomial logistic regression to build a classifier that can predict a researcher's position. We find the dominant features using the method of permutation feature importance. The dominant features include years active in academia, number of publications, h-index, number of citations, longitude of the affiliation and average citations.

We then perform causal analysis using PC algorithm coupled with a partial correlation test to find causal pathways over the 10 variables. We build causal graphs for all authors as well as for authors within subgroups of position. We find no direct link between n_pubs and n_citations and seem to be confounded by other variables. We find a causal pathway from longitude of the affiliation to h-index over average citations and n_citations. This is a stronger mediated path as compared to the direct path between location and h-index. Years active in publishing and citations are also confounded by other variables. This analysis then pointed to only years active, number of publications, h-index and number of citations to be the important parameters in classifying position.

6.2 Future Work

An evaluation of the causal discovery algorithm with CMISymbPerm would be an immediate consequence of this work. Further, the ideas from CMIknn and CMISymb can be used to build a CI test for the case of mixed data (arbitrary mix of continuous and discrete variables).

The link strength parameter as well as dependence measure estimates assume that the number of symbols are same for all nodes. Further work is to relax this condition for arbitrary number of symbols. An analytical formula leading link strength to CMI will help experiment with relative effect size in the CI test better.

In case of permutation test, the exponential time complexity is mainly due to the surrogate data testing. Parallelizing these over multiple cores of CPU or over multiple nodes in a cluster can significantly speed up the null distribution approximation. The time complexity within a surrogate is dependent on finding the number of neighbors to permute from, i.e. finding the subspace of Z. This grows for higher samples and dimensions of Z. Variants of KD trees for

categorical data can bring the time complexity to at least $\mathcal{O}(NlogN)$. Approximate permutation schemes in Tsamardinos and Borboudakis [2010b] can be explored further as well. Another significant time consumption is in the computation of contingency tables based on bincount histogram. KD Tree based methods can help here as well. Contingency table prediction using Poisson log-linear models (Tsagris [2017b]] could improve the time factor.

The calibration of CMISymbPerm over dimensions higher than 4 is important for larger conditioning sets in large graphs. The CI test was evaluated with only two models $X \perp\!\!\!\perp Y \mid Z$ and $X \not\perp\!\!\!\perp Y \mid Z$. More canonical models that are typical in a causal graph have to be evaluated. Another approach could be evaluation over random graphs generated from random Bayesian Networks.

The analyses in Bibliometrics were done for 4800 samples for three categories in position. These were generated using simple queries for respective positions in ES. More complex queries that incorporate natural language processing functionalities available within ES can increase sample size significantly. This would improve both the classifier as well as causal discovery phases which would eliminate weak links over smaller samples as well as detect stronger associations. Improving queries also allows the expansion of the dataset to more categories in position across academia. Further, encoding of 'position' into a continuous feature can be included within causal analysis which would help understand the causal graphs for subgroups in Figures 5.3 (b), (c) and (d) better. The causal graphs generated were fixed at significance level pc_alpha=0.005 and 0.05 which is quite conservative in accepting a link. Analysis over varying pc_alpha = [0.005, 0.01, 0.05, 0.1] could provide more insights.

Appendix A

Appendix

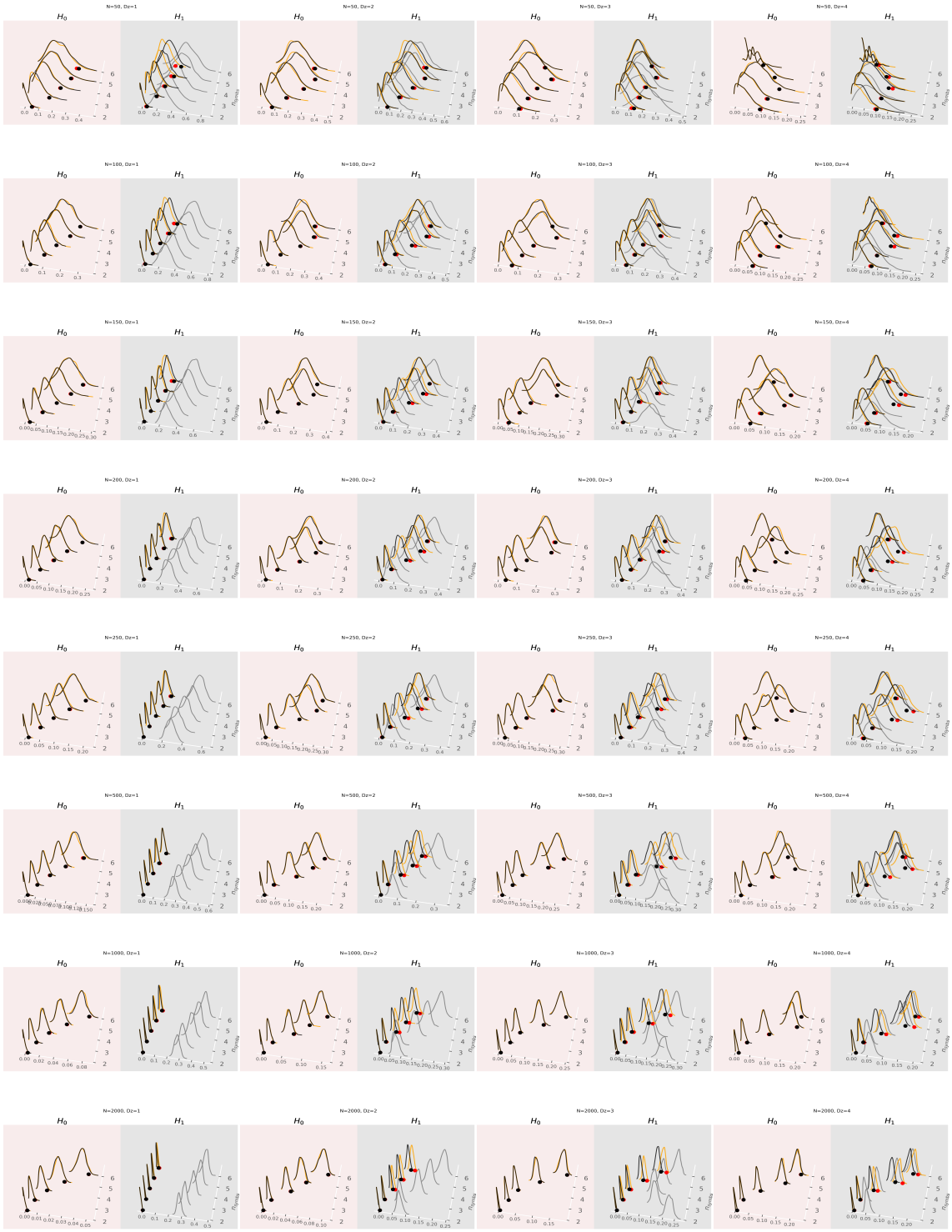

Figure A.1: H_0 approximated by CMISymbPerm

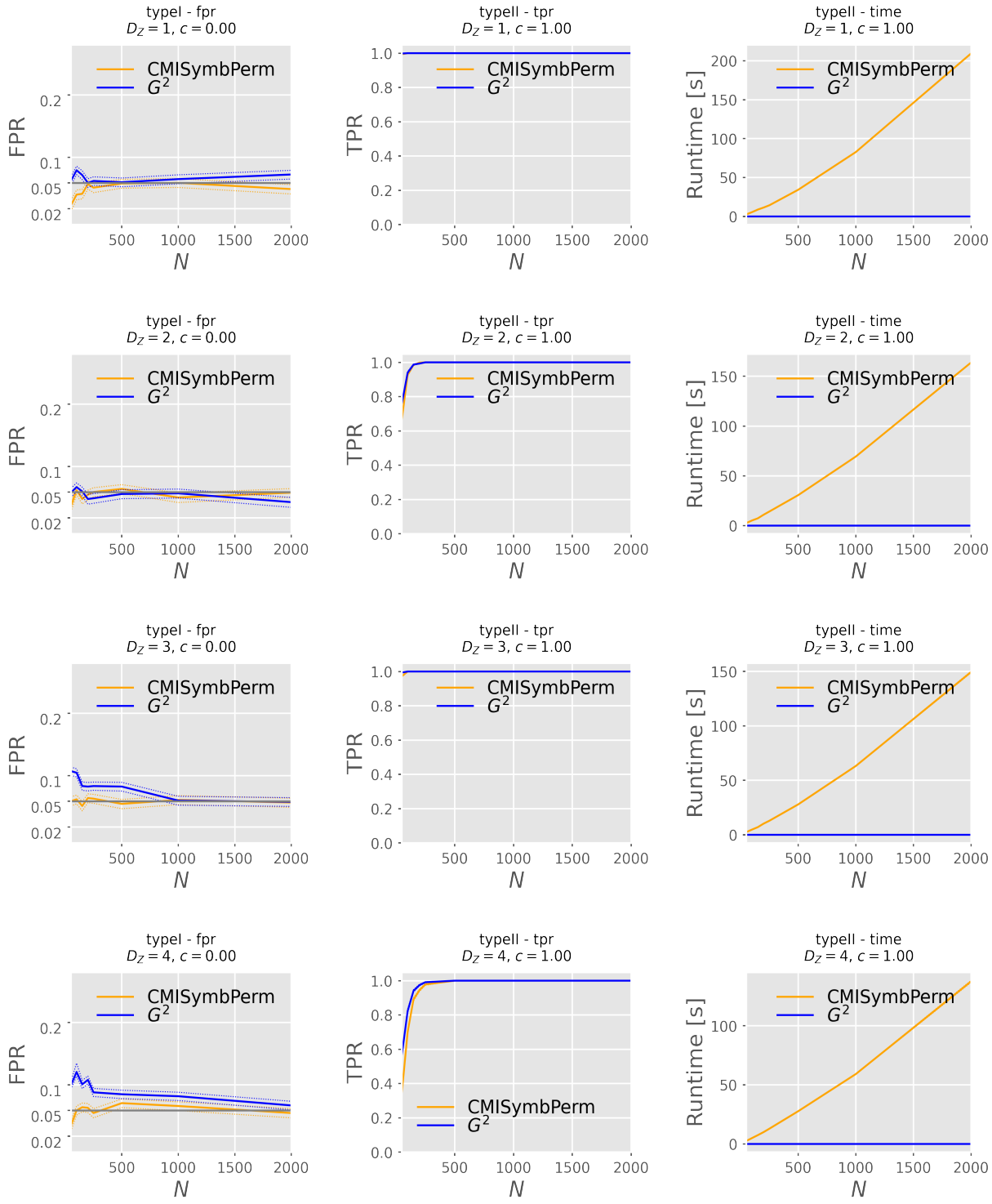

Figure A.2: Over N and D_z with $n_{symbs} = 2$

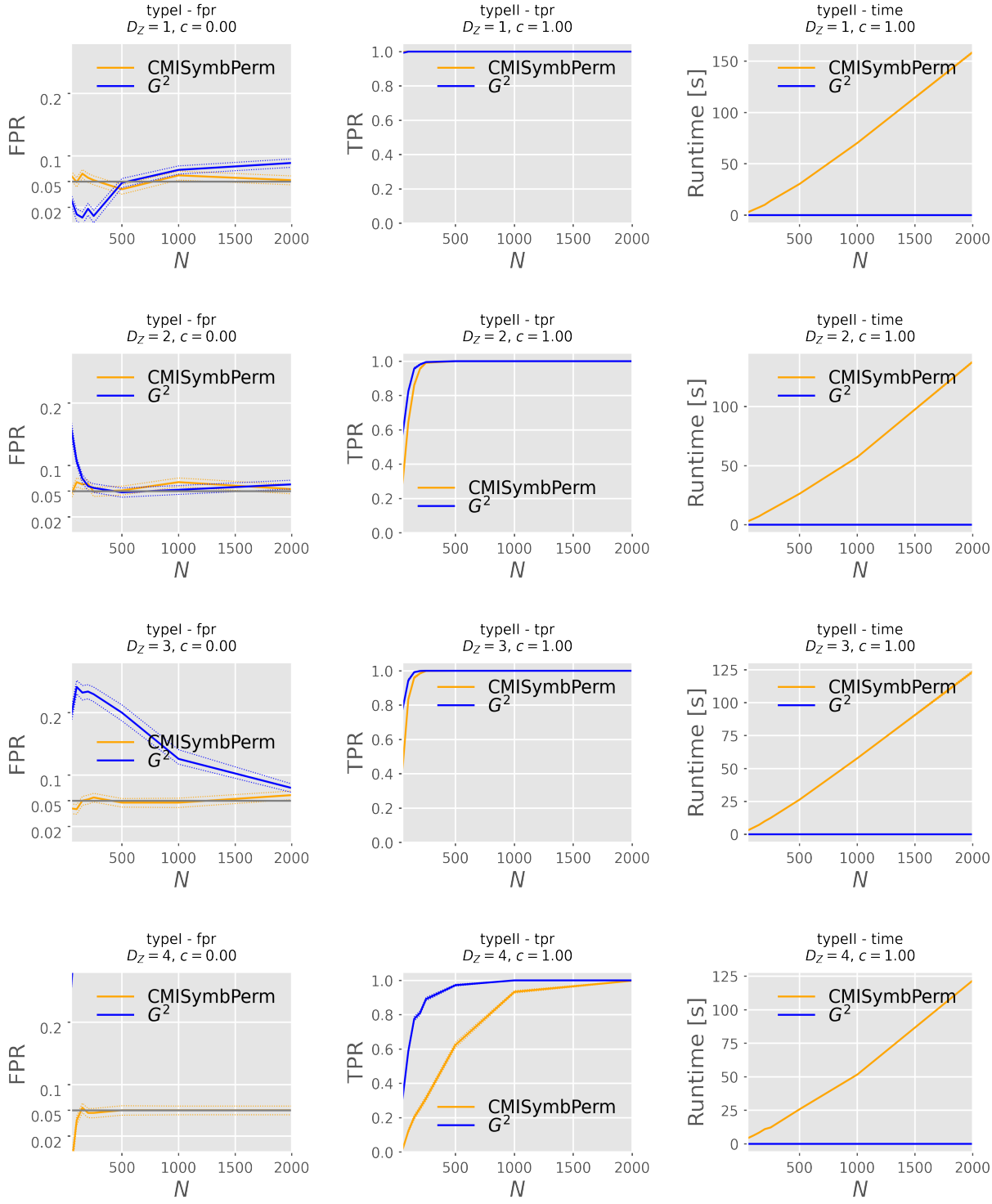

Figure A.3: Over N and D_z with $n_{symbs} = 4$

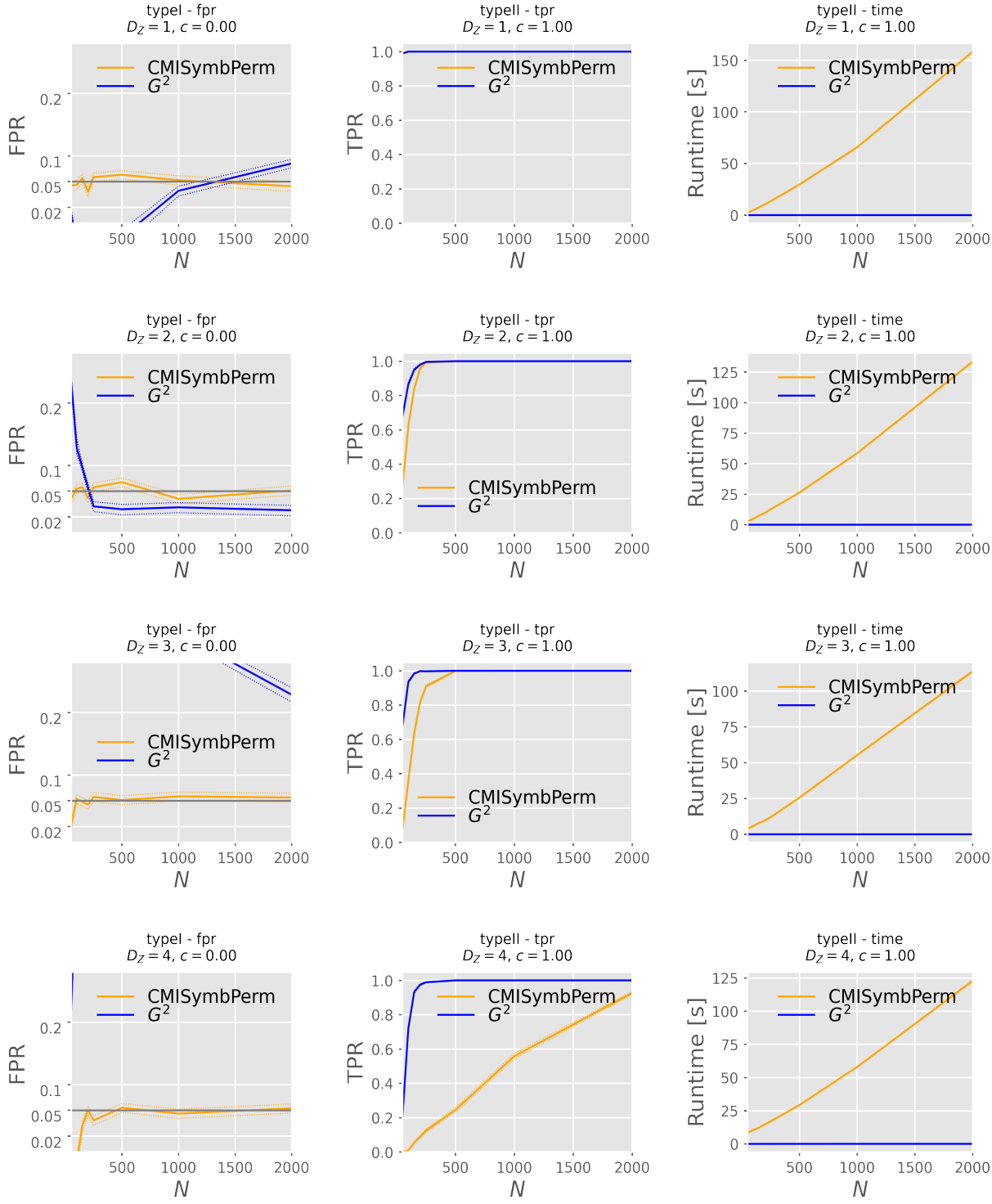

Figure A.4: Over N and D_z with $n_{symbs} = 5$

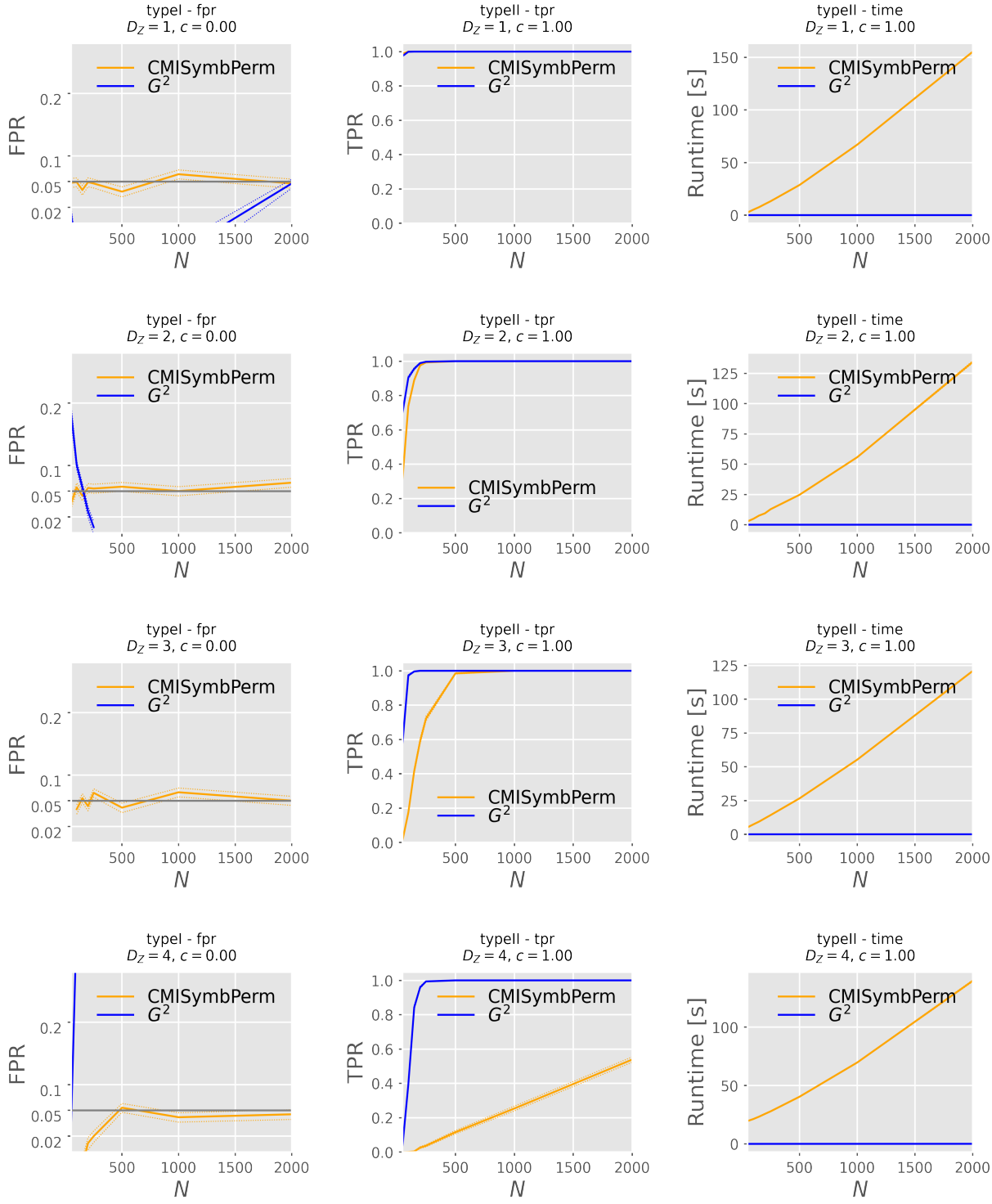

Figure A.5: Over N and D_z with $n_{symbs} = 6$

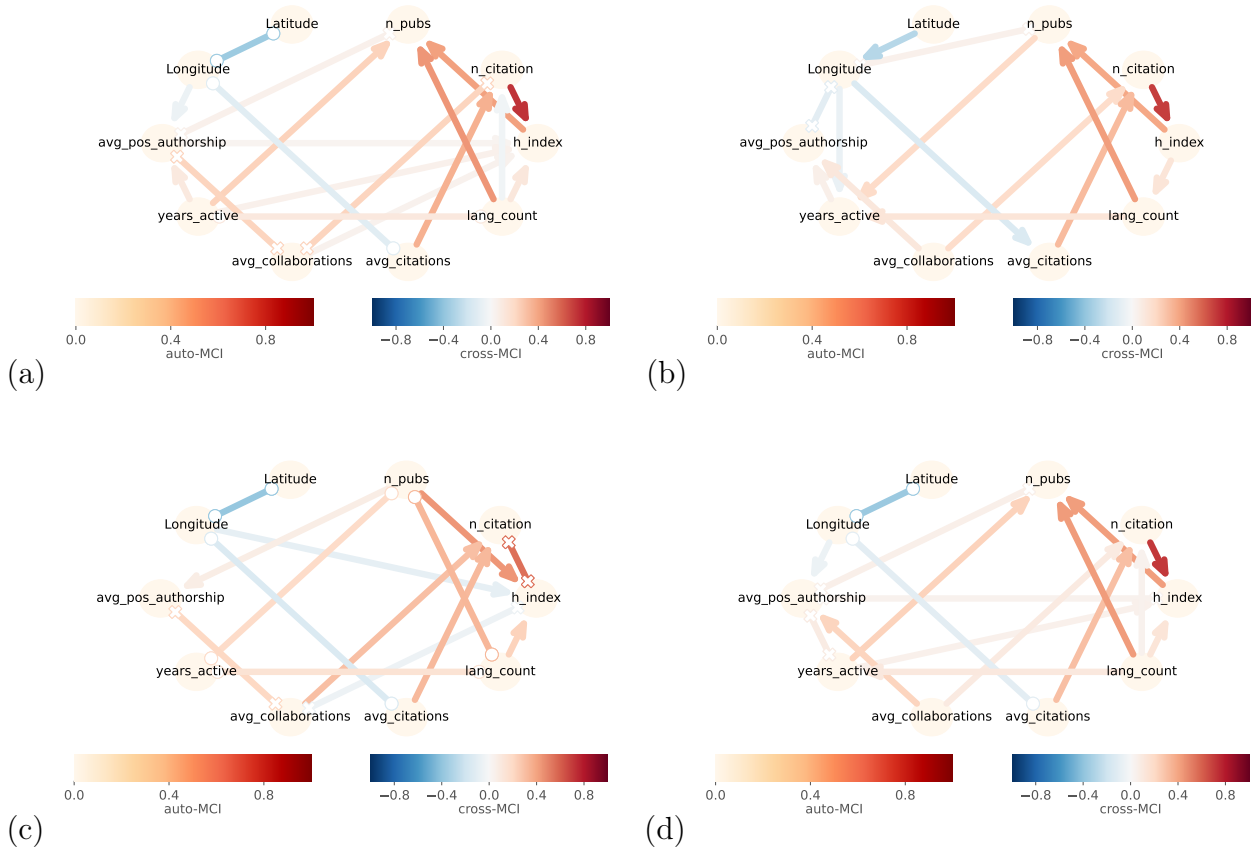

Figure A.6: pc_alpha=0.05 (a) All authors (b) Professors (c) Associate professors (d) Assistant professors

Bibliography

Open academic graph 2.1. https://www.aminer.cn/oag-2-1, 2020. Accessed: 2022-09-21.

Data generating processes. https://humboldt-wi.github.io/blog/research/applied_predictive_modeling_19/data_generating_process_blogpost/, 2022. Accessed: 2022-09-21.

R.J. Baker, M.R.B. Clarke, and P.W. Lane. Zero entries in contingency tables. *Computational Statistics Data Analysis*, 3:33–45, 1985. ISSN 0167-9473. doi: https://doi.org/10.1016/0167-9473(85)90056-8. URL https://www.sciencedirect.com/science/article/pii/0167947385900568.

Yvonne M Bishop, Stephen E Fienberg, and Paul W Holland. *Discrete multivariate analysis: theory and practice*. Springer Science & Business Media, 2007.

Lutz Bornmann and Hans-Dieter Daniel. What do we know about the h index? *J. Am. Soc. Inf. Sci. Technol.*, 58(9):1381–1385, jul 2007. ISSN 1532-2882. doi: 10.1002/asi.20609. URL https://doi.org/10.1002/asi.20609.

Justyna Brzezińska. The problem of zero cells in the analysis of contingency tables. *Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie*, pages 49–61, 01 2015. doi: 10.15678/ZNUEK.2015.0941.0504.

Elasticsearch. elasticsearch/elasticsearch, 2015. URL https://github.com/elasticsearch/elasticsearch.

Audrey Finkler. Goodness of fit statistics for sparse contingency tables, 2010. URL https://arxiv.org/abs/1006.2287.

Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12615–12625. Curran Associates,

Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/94e70705efae423efda1088614128d0b-Paper.pdf.

Dr. Andreas Gerhardus. *Causal Inference with Applications to Time Series. Lecture notes, private communication.* 2021.

Clark Glymour, Kun Zhang, and Peter L. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10, 2019.

Johannes Huegle, Christopher Hagedorn, Lukas Böhme, Mats Pörschke, Jonas Umland, and Rainer Schlosser. Manm-cs: Data generation for benchmarking causal structure learning from mixed discrete-continuous and nonlinear data. 01 2022.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the r package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. doi: 10.18637/jss.v047.i11. URL https://www.jstatsoft.org/index.php/jss/article/view/v047i11.

Teemu Kokkonen, Harri Koivusalo, Hanne Laine, Ari Jolma, and Olli Varis. A method for defining conditional probability tables with link strength parameters for a bayesian network. 2005.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* Adaptive computation and machine learning. MIT Press, 2009. ISBN 9780262013192. URL https://books.google.co.in/books?id=7dzpHCHzNQ4C.

H. H. Ku. A note on contingency tables involving zero frequencies and the 2 test. *Technometrics*, 5(3):398–400, 1963. ISSN 00401706. URL http://www.jstor.org/stable/1266344.

Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

Danielle Navarro. *Learning statistics with R: A tutorial for psychology students and other beginners. (Version 0.6).* University of New South Wales, Sydney, Australia, 2015. URL https://learningstatisticswithr.com. R package version 0.5.1.

Brady Neal. Lecture notes. introduction to causal inference. 2020. URL https://www.bradyneal.com/Introduction_to_Causal_Inference-Dec17_2020-Neal.pdf.

Richard Neapolitan. *Learning Bayesian Networks*. 01 2003. ISBN 9780123704771. doi: 10.1145/1327942.1327961.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

Judea Pearl. *Causality*. Cambridge University Press, Cambridge, UK, 2 edition, 2009. ISBN 978-0-521-89560-6. doi: 10.1017/CBO9780511803161.

Christopher Meek Thomas Richardson Clark Glymour Herbert Hoijtink Peter Spirtes, Richard Scheines and Anne Boomsma. Tetrad-iii, 1996.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2017. ISBN 978-0-262-03731-0. URL `https://mitpress.mit.edu/books/elements-causal-inference`.

Jakob Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 938–947. PMLR, 09–11 Apr 2018. URL `https://proceedings.mlr.press/v84/runge18a.html`.

Jakob Runge. *Causal Inference with Applications to Time Series. Lecture notes, private communication*. 2020a.

Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 1388–1397. PMLR, 03–06 Aug 2020b. URL `https://proceedings.mlr.press/v124/runge20a.html`.

Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11):eaau4996, 2019. doi: 10.1126/sciadv.aau4996. URL `https://www.science.org/doi/abs/10.1126/sciadv.aau4996`.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *International World Wide Web Conferences*. Microsoft, May 2015. URL

https://www.microsoft.com/en-us/research/publication/overview-microsoft-academic-service-mas-applications/.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

Sampath Srinivas. A generalization of the noisy-or model. In David Heckerman and Abe Mamdani, editors, *Uncertainty in Artificial Intelligence*, pages 208–215. Morgan Kaufmann, 1993. ISBN 978-1-4832-1451-1. doi: https://doi.org/10.1016/B978-1-4832-1451-1.50030-5. URL https://www.sciencedirect.com/science/article/pii/B9781483214511500305.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 990–998, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1402008. URL https://doi.org/10.1145/1401890.1402008.

Michail Tsagris. Conditional independence test for categorical data using poisson log-linear model. *Journal of Data Science*, 15, 06 2017a. doi: 10.6339/JDS.201704_15(2).0010.

Michail Tsagris. Conditional independence test for categorical data using poisson log-linear model. *Journal of Data Science*, 15, 06 2017b. doi: 10.6339/JDS.201704_15(2).0010.

I. Tsamardinos and Giorgos Borboudakis. Permutation testing improves bayesian network learning. In *ECML/PKDD*, 2010a.

Ioannis Tsamardinos and Giorgos Borboudakis. Permutation testing improves bayesian network learning. In José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 322–337, Berlin, Heidelberg, 2010b. Springer Berlin Heidelberg. ISBN 978-3-642-15939-8.

Olli Varis and Sakari Kuikka. Bene-eia: A bayesian approach to expert judgment elicitation with case studies on climate change impacts on surface waters. *Climatic Change*, 37:539–563, 1997.

Larry Wasserman. *All of statistics : a concise course in statistical inference*. Springer, New York, 2010. ISBN 9781441923226 1441923225. URL http://www.amazon.de/All-Statistics-Statistical-Inference-Springer/dp/1441923225/ref=sr_1_2?ie=UTF8&qid=1356099149&sr=8-2.

Wikipedia. Exact test — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Exact%20test&oldid=1062808687`, 2022a. [Online; accessed 12-October-2022].

Wikipedia. G-test — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=G-test&oldid=1086592517`, 2022b. [Online; accessed 12-October-2022].

Wikipedia. Type I and type II errors — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Type%20I%20and%20type%20II%20errors&oldid=1111052570`, 2022c. [Online; accessed 12-October-2022].

Sewall Wright. Correlation and causation. *Journal of agricultural research*, 20 (7):557–585, 1921.

F. Yates. Contingency tables involving small numbers and the 2 test. *Supplement to the Journal of the Royal Statistical Society*, 1(2):217–235, 1934. ISSN 14666162. URL `http://www.jstor.org/stable/2983604`.

Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, and Kuansan Wang. Oag: Toward linking large-scale heterogeneous entity graphs. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, August 2019. URL `https://www.microsoft.com/en-us/research/publication/oag-toward-linking-large-scale-heterogeneous-entity-graphs/`.