Bayesian Network Learning with Discrete Case-Control Data

Giorgos Borboudakis
Comp. Sci. Dept., University of Crete
Institute of Computer Science, FORTH

Ioannis Tsamardinos
Comp. Sci. Dept., University of Crete
Institute of Computer Science, FORTH

Abstract

We address the problem of learning Bayesian networks from discrete, unmatched case-control data using specialized conditional independence tests. Those tests can also be used for learning other types of graphical models or for feature selection. We also propose a post-processing method that can be applied in conjunction with any Bayesian network learning algorithm. In simulations we show that our methods are able to deal with selection bias from case-control data.

1 INTRODUCTION

Most Bayesian network learning algorithms assume i.i.d. data. In many studies, such as **case-control** studies, this not the case. In case-control studies data are selected based on one or multiple variables, usually using a comparable number of samples from different values of those variables, leading to non i.i.d. data. Such data are also called **artificially balanced** in the machine learning literature. Case-control sampling is often used in epidemiological or biomedical studies [Breslow, 1996], particularly when studying a disease that is relatively rare. Their goal is usually to identify differences between patients (cases) and healthy individuals (controls), such as identifying differentially expressed genes between the two groups from gene expression data. Case-control sampling is especially important when cases are very rare, as much fewer samples have to be collected compared to cross-sectional studies, significantly reducing the time and cost for obtaining the data.

We address the problem of learning Bayesian networks from discrete case-control data. This is challenging because case-control data do not necessarily represent a sample from the general population. In general, non i.i.d. sampling may lead to **selection bias**, which could alter the (conditional) independence relations in the data; case-control sampling is a special case of such sampling. Because of that, one cannot usually use methods designed for i.i.d. data.

To the best of our knowledge, there has been only one previous approach to learn Bayesian networks from case-control data by Cooper [2000]. The idea is to use a Bayesian method to integrate over all possible values for all non-sampled cases and controls, assuming that those numbers are known a priori. This method is very impractical, mainly due to its high computational cost. There is a vast literature on methods for case-control data ([Rothman et al., 2008] contains a review of many methods, as well as relevant references) but they are mostly concerned with modeling the outcome (on which selection is based on), and thus are not applicable for modeling other measured variables or for Bayesian network learning.

Learning from case-control data is important for the following reasons. First, a lot of case-control datasets have accumulated over the years and such methods could be used to identify novel associations or possibly even causal relations. For instance, the NCBI GEO database contains thousands of biological datasets [Edgar et al., 2002, Barrett et al., 2013], many of which stem from case-control studies. In addition, this would allow co-analysis of data collected under different experimental designs. Currently, there exist several methods for learning causal networks from multiple heterogeneous datasets [Cooper and Yoo, 1999, Tillman and Spirtes, 2011, Hyttinen et al., 2013, Triantafillou and Tsamardinos, 2014]. Those methods are able to use observational and experimental i.i.d. data. Extending them for other types of data, such as case-control data, is a natural next step.

In this paper we define the problem and show the implications of case-control sampling on the observed independencies. We propose different conditional independence tests for discrete data, as well as a post-processing method that can be used with any Bayesian network learning algorithm. Finally, we investigate the behavior of our methods in various, simulated experiments and show that they are able to handle selection bias from case-control data.

2 PRELIMINARIES

We will briefly introduce the basic theory and notation used throughout the paper. Interested readers may refer to Pearl [2000] or Spirtes et al. [2000].

We use upper-case and lower-case letters to refer to random variables (e.g. X) and values of those variables (e.g. x), and bold letters to refer to sets of variables or values. Let $\mathbf{V}$ be a set of random variables. A **Bayesian Network** (BN) over $\mathbf{V}$ is a pair $\mathcal{B} = \langle \mathcal{G}, \mathcal{P} \rangle$, where $\mathcal{G}$ is a **Directed Acyclic Graph** (DAG) representing conditional independencies between variables $\mathbf{V}$, and $\mathcal{P}$ is the joint probability distribution of $\mathbf{V}$. We will use the terms variable and node interchangeably. The graph and distribution are connected through the **Markov Condition**: a variable is conditionally independent of all its non-descendants given its parents. The **skeleton** $\mathcal{G}_S$ of a BN $\mathcal{G}$ is the undirected graph which can be constructed by ignoring the orientations of $\mathcal{G}$. A triple of nodes $\langle X, Y, Z \rangle$ is called a **collider** in $\mathcal{G}$, if $X \rightarrow Y \leftarrow Z$ is in $\mathcal{G}$. Two variables X and Y are **d-separated** given a (possibly empty) set of variables $\mathbf{Z}$ if and only if for all paths between X and Y one of the following is true: (a) there is a collider $U \rightarrow V \leftarrow W$ on that path and neither V nor any of its descendants is in $\mathbf{Z}$, or (b) there is a consecutive triple $\langle U, V, W \rangle$ that is not a collider and V is in $\mathbf{Z}$. If X and Y are not d-separated given $\mathbf{Z}$ they are **d-connected**. We assume the **Faithfulness Condition** that (together with the Markov Condition) implies that *there is a d-connecting path between X and Y given $\mathbf{Z}$, if and only if X and Y are statistically dependent given $\mathbf{Z}$*. We denote conditional dependence and independence of two variables X and Y given $\mathbf{Z}$ as $Dep(X; Y|\mathbf{Z})$ and $Ind(X; Y|\mathbf{Z})$ respectively.

3 PROBLEM DEFINITION

In this work we consider discrete data from **unmatched case-control** studies. In unmatched studies samples are assumed to be sampled in an i.i.d. fashion from the respective subpopulations. We assume that the data have been selected based on a set of *measured* variables $\mathbf{T}$; we will call those variables **selection variables**. In case $\mathbf{T}$ contains only a single variable, we will refer to it as T. We denote with S a binary variable that indicates whether a sample has been selected or not. In our case, we assume that S only depends on $\mathbf{T}$; their relation can be modeled by nodes with directed edges from each $\mathbf{T}$ to S.

Assumption 1. *S depends only on $\mathbf{T}$ and all variables in $\mathbf{T}$ have been measured.*

Case-control sampling induces a type of **selection bias**. Selection bias arises if samples are less probably to be sampled based on some criteria. When analyzing such data it may happen that **spurious dependencies** are identified which do not exist in the general population, but are due to the selection process. The reason for that is that the data $\mathcal{D}$ are collected from the conditional distribution $P(\mathcal{D}|S = 1)$. Consider the example shown in Figure 1. Although X and Y are independent in the general population, a naive analysis that does not account for the sampling process would identify a spurious dependence between them. This happens because the data are selected conditional on $S = 1$, d-connecting X and Y through T, as S is a descendant of T and T is a collider on the path $X \rightarrow T \leftarrow Y$.

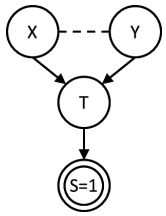

Figure 1: Conditioning on $S = 1$ introduces a spurious dependence between X and Y.

The question is if and when it is possible to estimate the joint probability distribution of a set of variables $\mathbf{X}$ in the general population, $P(\mathbf{X})$, from data collected with case-control sampling, that is following $P(\mathbf{X}|S = 1)$. Bareinboim et al. [2014] address the general problem of recoverability of conditional distributions when data are collected under selection. They show that the conditional distribution $P(Y|X)$ is not recoverable when data are collected with case-control sampling. This result can be trivially extended to the case of estimating the joint distribution of a set of variables. Fortunately, they show that it is possible to recover the population distribution if the joint distribution of $\mathbf{T}$ is known. Then, $P(\mathbf{X})$ can be estimated as follows.

$$\begin{aligned} P(\mathbf{X}) &= \sum_{\mathbf{t}} P(\mathbf{X}|\mathbf{T} = \mathbf{t}) P(\mathbf{T} = \mathbf{t}) \\ &= \sum_{\mathbf{t}} P(\mathbf{X}|\mathbf{T} = \mathbf{t}, S = 1) P(\mathbf{T} = \mathbf{t}) \end{aligned} \tag{1}$$

The second equality follows by Assumption 1. The conditional probability $P(\mathbf{X}|\mathbf{T} = \mathbf{t}, S = 1)$ can be directly estimated from $\mathcal{D}$. Thus, in order to estimate $P(\mathbf{X})$ for any set of variables we only need the joint probability distribution of $\mathbf{T}$ in the general population. This could either be provided as prior knowledge by a domain expert or from the literature, or estimated from an external data source.

Assumption 2. *The joint probability distribution of $\mathbf{T}$ in the general population is known.*

Notice that there are cases where the equality $P(\mathbf{X}) = P(\mathbf{X}|S = 1)$ holds. For example, if none of the variables in $\mathbf{X}$ is dependent with $\mathbf{T}$, $P(\mathbf{X})$ can be directly estimated from $\mathcal{D}$. We will further investigate the conditions under which this holds in the next section.

We conclude with some comments on our assumptions. Assumption 1 is reasonable and probably holds for most case-control studies. In case it is violated additional spurious dependencies may be introduced. Regarding Assumption 2: although it may be restrictive in some cases, prior information about the joint distribution is often available. In fact, many methods for analyzing case-control data require such prior information. For example, in logistic regression models for an outcome that has been selected on (e.g. disease), such knowledge is necessary in order to estimate the intercept of the model, although it is not needed in order to estimate the remaining parameters [Breslow and Day, 1980].

4 IMPLICATIONS OF CONDITIONING ON S

In the previous section we saw an example where conditioning on $S = 1$ introduces a spurious dependence. Next, we will further investigate how the dependencies and independencies are affected after conditioning on $S = 1$. Most of those results are based on previous results by Spirtes et al. [2000] (see Section 9.3). Those results are general, allowing for S do be in any position in the graph. We will show their consequences for the special case of case-control sampling.

First we investigate whether conditioning on $S = 1$ removes any dependencies that exist in the general population. Spirtes et al. [2000] have characterized all situations where this happens.

Corollary 1 (Adapted from [Spirtes et al., 2000])**.** *If $Dep(X;Y|\mathbf{Z})$, then $Ind(X;Y|\mathbf{Z},S=1)$ holds if and only if there exists a path U between X and Y such that (a) every collider on U has a descendant in $\mathbf{Z}$, (b) no non-collider in U is in $\mathbf{Z}$, and (c) S is a non-collider on every such path.*

In our case there is no such path because the third condition can never be satisfied, as S does not have any outgoing edges. Because of that, S can only be a collider on all such paths. Therefore, *conditioning on $S = 1$ does not remove any dependencies*.

The next corollary characterizes the cases where a conditional independence may turn into a dependence.

Corollary 2 (Adapted from [Spirtes et al., 2000])**.** *If $Ind(X;Y|\mathbf{Z})$, then $Dep(X;Y|\mathbf{Z},S=1)$ holds if and only if there exists a path U between X and Y such that (a) no non-collider on U is in $\mathbf{Z} \cup \{S\}$, (b) every collider on U has a descendant in $\mathbf{Z} \cup \{S\}$, and (c) some collider on U does not have a descendant in $\mathbf{Z}$.*

Based on this result, we will characterize all cases where a spurious dependence will appear in the graph after conditioning on $S = 1$ that cannot be removed by conditioning on any set of variables. We proceed by stating and proving the result.

Theorem 1. *Let X, Y be two variables. If $\exists \mathbf{Z}\ Ind(X;Y|\mathbf{Z})$, then $\forall \mathbf{Z}'\ Dep(X;Y|\mathbf{Z}', S = 1)$ holds if and only if there is a node W such that (a) $X \rightarrow W \leftarrow Y$, and (b) $W = S$ or W is an ancestor of S.*

Proof.
Sufficiency: Follows trivially, as conditioning on S d-connects X and Y through W.

Necessity: We will show this by contradiction. Assume that there is no W satisfying both conditions. From Corollary 2 we know that for some $\mathbf{Z}$ satisfying both $Ind(X;Y|\mathbf{Z})$ and $Dep(X;Y|\mathbf{Z},S=1)$, there is a path U between X and Y with at least one collider V on U that is also an ancestor of S. The only case were this holds is if there is at least one node between X and V or Y and V on U. But then at least one of those nodes has to be a noncollider and we could d-separate X and Y by conditioning on any such noncollider, contradicting our assumptions. □

In words, this theorem characterizes all cases where two variables X and Y can be d-separated in the population graph, but cannot d-separated by any set after conditioning on S. We will later use this to learn Bayesian networks from case-control data.

5 CONDITIONAL INDEPENDENCE TESTING

As we saw in the previous section, conditioning on S may introduce spurious dependencies in the data. Because of that, one cannot use independence tests designed for i.i.d. data. In this section we will describe various conditional independence tests for case-control data.

5.1 TEST STATISTIC

As a test statistic we consider the conditional mutual information (CMI) $I(X;Y|\mathbf{Z})$, which is defined as:

$$I(X;Y|\mathbf{Z}) = \sum_{x,y,\mathbf{z}} P(x,y,\mathbf{z}) \log \frac{P(x,y,\mathbf{z})P(\mathbf{z})}{P(x,\mathbf{z})P(y,\mathbf{z})} \quad (2)$$

Assuming that we know the distribution of $\mathbf{T}$, we can use Equation 1 to compute $I(X;Y|\mathbf{Z})$ for any variables X, Y and $\mathbf{Z}$. For the case of i.i.d. data the CMI is closely related to the G-statistic used by the G-test:

$$G(X;Y|\mathbf{Z}) = 2 \cdot N \cdot I(X;Y|\mathbf{Z}) \quad (3)$$

where N is the sample size. Under the null hypothesis this statistic is asymptotically χ^2 distributed with $(|X| - 1)(|Y| - 1)|\mathbf{Z}|$ degrees of freedom.

Unfortunately, the G-test cannot be trivially applied to case-control data. Intuitively, the reason is that N case-control samples are not always equivalent to N samples from an i.i.d. dataset. We will show the intuition behind this with an example [1].

Example 1. *Assume that* $\mathbf{T}$ *contains a single binary variable* T *and we sample* $N = 1000$ *samples,* 500 *for each value of* T*, and that* $P(T = 0) = 0.2$*. The probability of* $\mathbf{X}$ *given by Equation 1 is* $P(\mathbf{X}) = P(\mathbf{X}|T = 0) \cdot 0.2 + P(\mathbf{X}|T = 1) \cdot 0.8$*. If we use* N *as our sample size, we essentially assume that we have estimated* $P(\mathbf{X}|T = 0)$ *and* $P(\mathbf{X}|T = 1)$ *using 200 and 800 samples respectively even though we used 500 for each of them. As a result,* ***we overestimate and underestimate the variance in the estimation of*** $P(\mathbf{X}|T = 1)$ ***and*** $P(\mathbf{X}|T = 0)$ ***respectively, which can lead to false results****.*

Next we consider various strategies to deal with this.

5.2 UNDERSAMPLING

The trivial approach is to use a subset of the samples such that the proportion of values of $\mathbf{T}$ in the resulting dataset coincides with the distribution of $\mathbf{T}$. For Example 1 we could use 125 samples with $T = 0$ and 500 samples with $T = 1$, as $125/625 = 0.2$ and $500/625 = 0.8$, and perform a standard independence test. In general, one can use at most $N = \min_{\mathbf{t}} N(\mathbf{T} = \mathbf{t})/P(\mathbf{T} = \mathbf{t})$ samples, where $N(\mathbf{T} = \mathbf{t})$ is the number of samples with $\mathbf{T} = \mathbf{t}$ (proof omitted).

There are several downsides to this approach. First, undersampling often ignores a significant amount of samples (375 in the previous example), possibly reducing the power of the test. Second, the result may vary a lot, depending on the selected samples. One possibility to reduce this variance is to create multiple datasets by undersampling, perform a test on each such dataset and combine the results somehow (e.g. taking the median p-value). Finally, undersampling may be problematic if the marginal distribution of T contains extreme values. In the previous example, if $P(\mathbf{X}|T = 0)$ was 0.01, only 5 samples with $T = 0$ could be used to (approximately) satisfy the marginals. In practice, those values will often be even more extreme.

5.3 A PERMUTATION TEST

Permutation tests are non-parametric procedures for statistical significance testing. The basic idea is that, under the null hypothesis, one can permute the data in an appropriate way to generate another, permuted dataset. *Specifically, for a permutation test to be exact, the permutation has to be performed in a way that preserves the distribution of the observations under the null hypothesis* [Good, 2004]. Then, the test statistic computed on that dataset is a sample from its null distribution. Because the number of all permutations is usually astronomically large, making complete enumeration infeasible, one usually resorts to Monte Carlo approximations that sample a relatively small number of permutations (usually between 1000 and 10000). The p-value is computed as the proportion of permutation statistics that are at least as extreme as the statistic on the original data.

Permutation Testing for Discrete Data. For conditional independence testing the null hypothesis is that X and Y are conditionally independent given $\mathbf{Z}$. This means that conditional independence holds for any value $\mathbf{z}$ of $\mathbf{Z}$. A permuted dataset can be created by randomly permuting the columns of X and Y for each value $\mathbf{z}$ of $\mathbf{Z}$ [Tsamardinos and Borboudakis, 2010]. For example, if Z is a binary variable, a permuted dataset is created by splitting the original dataset $\mathcal{D}$ into two datasets, $\mathcal{D}_{Z=0}$ and $\mathcal{D}_{Z=1}$, randomly permuting the columns of X and Y on each of them and then combining the resulting datasets. This results in an exact test, as the conditional distribution of X and Y for each value of $\mathbf{Z}$ remains fixed.

We use the same procedure for discrete case-control data using the CMI as our test statistic. It is important to note that *this permutation approach does not always preserve the distribution of* X *and* Y *under the null* when applied to case-control data. We show this with the following example. Let X be a discrete variable, T a binary variable and $P(T = 0) = 0.2$. Now, assume that for some value x of X we have 100 samples with $T = 0$ and 50 samples with $T = 1$ in the original dataset, and 50 and 100 respectively for some permuted dataset. Then, $P(x) = 100/500 \cdot 0.2 + 50/500 \cdot 0.8 = 0.12$ for the original dataset but $P(x) = 50/500 \cdot 0.2 + 100/500 \cdot 0.8 = 0.18$ for the permuted dataset. Let Y be another discrete variable with the same marginals as X. Then, the joint distribution of X and Y under the null is $P(x, y) = 0.12 \cdot 0.12$ in the original dataset and $P(x, y) = 0.18 \cdot 0.18$ in the permuted dataset. Thus, their joint distribution under the null is not invariant for this type of permutations.

We conducted various simulations to investigate the behavior of this test and, although this approach does not result in an exact test, they suggest that it works reasonably well in practice; the results are presented in Section 7.

5.4 AN ASYMPTOTIC TEST

An interesting observation that we made is that, under the null hypothesis, the permutation distribution of the G-

[1] We have also conducted several, anecdotal simulations which confirm this problem.

Algorithm 1 Estimate Effective Sample Size

Input: $P(\mathbf{T})$, $N(\mathbf{T})$, N, K
Output: N_{ESS}

1: **for** $i \leftarrow 1 : K$ **do**
2: $X \leftarrow$ Random(Uniform, Binary, N)
3: $Y \leftarrow$ Random(Uniform, Binary, N)
4: $Stats_i \leftarrow$ MutualInformation($X, Y, N(\mathbf{T}), P(\mathbf{T})$)
5: **end for**
6: $Stats \leftarrow$ Sort($Stats$, Ascending)
7: $Stats' \leftarrow$ Inverse-χ^2-Cdf(0:1/(K-1):1, DoF = 1)
8: $N_{ESS} \leftarrow$ Median($1/2 \cdot Stats'/Stats$)
9: $N_{ESS} \leftarrow$ Min(N_{ESS}, N)

statistic computed on the case-control data, for some unknown number of samples N, seems to also follow a χ^2 distribution with $(|X|-1)(|Y|-1)|\mathbf{Z}|$ degrees of freedom. We observed this behavior for a large number of different: (i) conditional and unconditional tests, (ii) probability distributions of $\mathbf{T}$, and (iii) types of discrete variables X, Y and $\mathbf{Z}$. We conjecture that this is always the case.

Conjecture 1. *The G-statistic $G(X;Y|\mathbf{Z})$ as defined by Equation 3 and computed for case-control data using Equations 1 and 2 is asymptotically distributed as a χ^2 random variable with $(|X|-1)(|Y|-1)|\mathbf{Z}|$ degrees of freedom for some unknown number of samples N.*

We will call this unknown number of samples the **effective sample size** and denote it as N_{ESS}. Based on this conjecture, we will devise a simple procedure to estimate N_{ESS}.

Let $\mathcal{D}$ be a dataset obtained from case-control sampling, $P(\mathbf{T})$ be the joint distribution of $\mathbf{T}$, $N(\mathbf{T})$ be the number of samples in $\mathcal{D}$ for each value of $\mathbf{T}$, and N be the total number of samples in $\mathcal{D}$. Suppose that we generate a large number K of independent random variables X and Y, each of N samples, assuming that the first $N(\mathbf{T} = \mathbf{t}_0)$ samples correspond to $\mathbf{T} = \mathbf{t}_0$, the next $N(\mathbf{T} = \mathbf{t}_1)$ to $\mathbf{T} = \mathbf{t}_1$ and so on, and then compute their mutual information. Let $Stats$ contain all statistics in ascending order. If K is large enough we would expect the i-th value in $Stats$, $Stats_i$, to correspond to a p-value of $i/(K-1)$. According to Conjecture 1 the G-statistic for some N_{ESS} of any two independent random variables follows a χ^2 distribution. Thus, for each such p-value, we can use the inverse χ^2 cumulative distribution to compute its corresponding statistic $Stats'$. We know that $Stats_i = I(X_i;Y_i)$ and that $Stats'_i \simeq 2 \cdot N_{ESS} \cdot I(X_i;Y_i)$. Thus, we can estimate N_{ESS} as $N_{ESS} \simeq 1/2 \cdot Stats'_i/Stats_i$. Because the procedure is not exact, we suggest to compute this value for each pair of $Stats$ and $Stats'$ values, and use the median value as an estimate for N_{ESS}. Naturally, this value cannot be larger than N, so we use the minimum of those values. The procedure is shown in Algorithm 1.

The method only needs to be applied once before analyzing a dataset, adding only a constant computational overhead. As a result, the cost of analyzing a case-control dataset is essentially identical to analyzing any other dataset, up to a constant additive factor.

6 BAYESIAN NETWORK LEARNING

We propose two different strategies for learning Bayesian networks from case-control data.

One is to use a test suited for case-control data with any existing constraint-based method. This strategy also allows one to learn other graphs such as Maximal Ancestral Graphs [Richardson and Spirtes, 2002], or to perform feature selection using a conditional independence based method [Tsamardinos et al., 2006].

Another approach is to learn a network using an independence test suited for i.i.d. data and perform a **post-processing** step to correct the graph by identifying and removing spurious dependencies using an independence test for case-control data. Theorem 1 characterizes all cases where a spurious dependence will be identified. Of course, we cannot directly apply Theorem 1 as we do not know the real DAG. Instead, we will use the skeleton of the DAG without any orientations. The next corollary characterizes all potentially spurious edges in a skeleton.

Corollary 3. *Let G_S be the skeleton of a DAG $\mathcal{G}$. An edge between variables X and Y is **potentially spurious**, if and only if there is a node W such that (a) X, Y and W are adjacent and form a triangle, and (b) $W = S$ or there is a potentially directed path from W to S (the path cannot go through X or Y). As S will not be in $\mathcal{G}$ we have to check if $W \in \mathbf{T}$ or if W is a potential ancestor of any variable in $\mathbf{T}$.*

This result follows from Theorem 1 and is stated without proof. This directly suggests how to use it with existing learning algorithms. After identifying $\mathcal{G}$, take its skeleton $\mathcal{G}_S$, check for triples X, Y, W that satisfy those criteria, and finally try to remove potentially spurious edges by performing a series of independence tests with an appropriate method. Note that W never has to be conditioned on in those tests as it either is a collider and would d-connect X and Y or, if not, the edge between X and Y can not be spurious and should not be removed. The second condition can be checked by removing all edges at X and Y and checking whether W and $\mathbf{T}$ are connected by a path.

We have to point out that this approach may not be optimal. Instead of the skeleton, there may be a way to partially orient the graph and further narrow down the cases where Corollary 3 applies. For example, if the edge from W to X is oriented towards X, then the edge between X and Y cannot be due to a spurious dependence, but applying Corollary 3 on the skeleton will try to remove it. However, this is not trivial; a naive application of the PC rules may

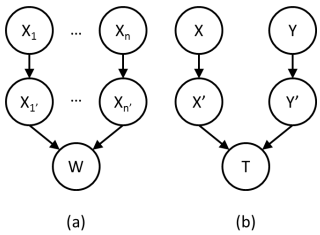

Figure 2: (a) Graphical representation of noisy model. (b) Collider model for the first set of experiments.

result in false orientations due to the presence of spurious edges. We did not further investigate this possibility.

7 EXPERIMENTAL EVALUATION

We performed simulations to investigate the behavior of the proposed independence tests in different situations. We consider only a single selection variable T. When we generate data, we select an equal number of samples for each value of T (that is, T is uniformly distributed in the case-control data). For a given Bayesian network we compute the marginal distribution of T using an exact inference algorithm implemented in the Bayes Net toolbox [Murphy, 2001]. We used $K = 100000$ to estimate the effective sample size.

First, we evaluate the tests in simple Noisy-MAX and Noisy-SUM scenarios. Then, we investigate the sensitivity of the tests with respect to the prior distribution of T. Finally, we compare the proposed Bayesian network learning strategies on the INSURANCE network [Binder et al., 1997]. We used MATLAB to conduct the simulations and create the figures (the code is available at http://www.mensxmachina.org/).

7.1 NOISY-MAX AND NOISY-SUM

We use the model proposed by [Srinivas, 1993], a generalization of the Noisy-OR model [Pearl, 1988], to generate data from a Noisy-MAX or Noisy-SUM distribution. Those models are members of the family of independence of causal influences (ICI) models [Heckerman and Breese, 1994]. A graphical representation of noisy models is shown in Figure 2. The nodes X_i' are called **inhibitor** nodes and are used to introduce noise in the model. Noisy-MAX distributions in particular are interesting as they have been shown to be very common in practice [Zagorecki and Druzdzel, 2006].

7.1.1 Setup

In all our experiments the inhibitor nodes take the same number of values as their parents and their distribution is: $P(X_i' = 0|X_i = 0) = 1$, $P(X_i' = k|X_i = k) = 1 - e$ and $P(X_i' = k|X_i \neq k) = e/(|X_i| - 1)$, where e is the noise parameter. The value of W is a deterministic function of its parent values (MAX or SUM).

We use two different cases for our evaluation. The first is a simple collider graph (see Figure 2 (b)). Here we test whether X and Y are unconditionally independent. We use this to evaluate the ability of the tests to handle the case of spurious dependencies. The second is a chain graph ($X \to X' \to T \to T' \to Y$), with two additional nodes – inhibitor node pairs, one into T and one into Y. For this case we test whether X and Y are unconditionally and conditionally independent given T. This is done to investigate the behavior of the tests in case no spurious dependencies are present.

For the collider case we generated data from the Noisy-MAX and Noisy-SUM distributions, and for the chain graph we generated data from the Noisy-MAX distribution. The parameters we used are: noise $e \in \{0, 0.3, 0.7\}$, sample size $N \in \{250, 1000\}$, range of values $r \in \{2, 4\}$ for X and Y. We used 6 independence tests: G^2 test, Permutation G^2 test, G^2_{cc} test for case-control data, Permutation G^2_{cc} test for case-control data, Undersampling G^2_u test, Bootstrapping and Undersampling G^2_u test using the median p-value. For the permutation tests we used 1000 permutations, and for the bootstrapping test we used 500 samples.

7.1.2 Results

The results for the collider and chain graphs are shown in Figures 3 and 4. In each figure we show the empirical CDF function of the p-values. In case independence holds, the p-values should be uniformly distributed and the CDF should be on the diagonal. In case of dependence, we would ideally have low p-values only. We use Test($X; Y$) and Test($X; Y|T$) to refer to the unconditional and conditional tests of X and Y. The first and last two columns of each group of figures correspond to data with $r = 2$ and $r = 4$ respectively.

G^2 and Permutation G^2. For the collider graph both tests identify a spurious dependence, as expected, unless the noise is too high or the sample size is too low. For the chain graph, where case-control sampling does not affect the independencies, both tests perform well. The asymptotic test does not always produce calibrated p-values for the test Test($X; Y|T$), agreeing with previous results [Tsamardinos and Borboudakis, 2010]. The simulations confirm that tests designed for i.i.d. data should not be applied on case-control data.

G^2_{cc} and Permutation G^2_{cc} For the collider graph, both tests produce p-values close to the ideal uniform distribution (black diagonal line), or overestimate the p-value; this can be seen especially in the noiseless Noisy-SUM case. Although this is not ideal, it is still useful, as the significance level upper bounds the actual type I error. Unfortu-

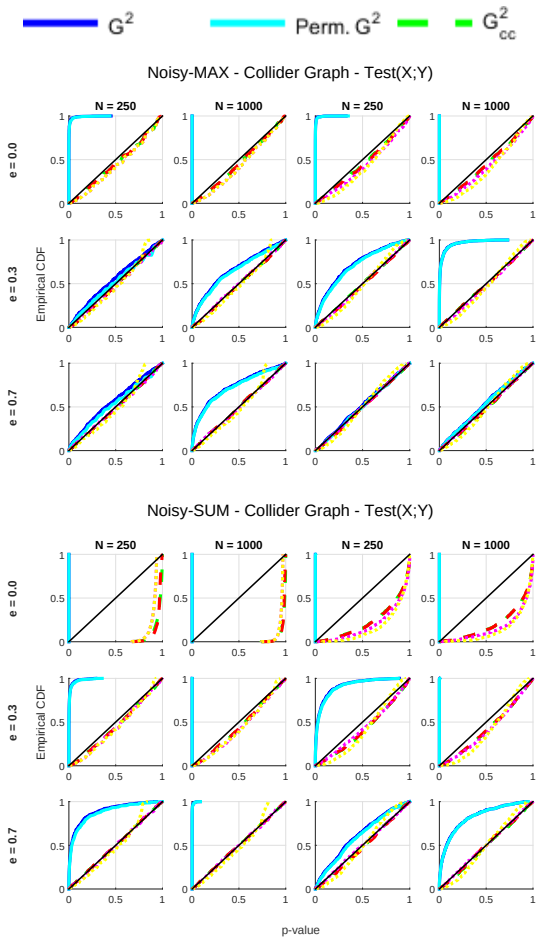

Figure 3: Results for the collider graph.

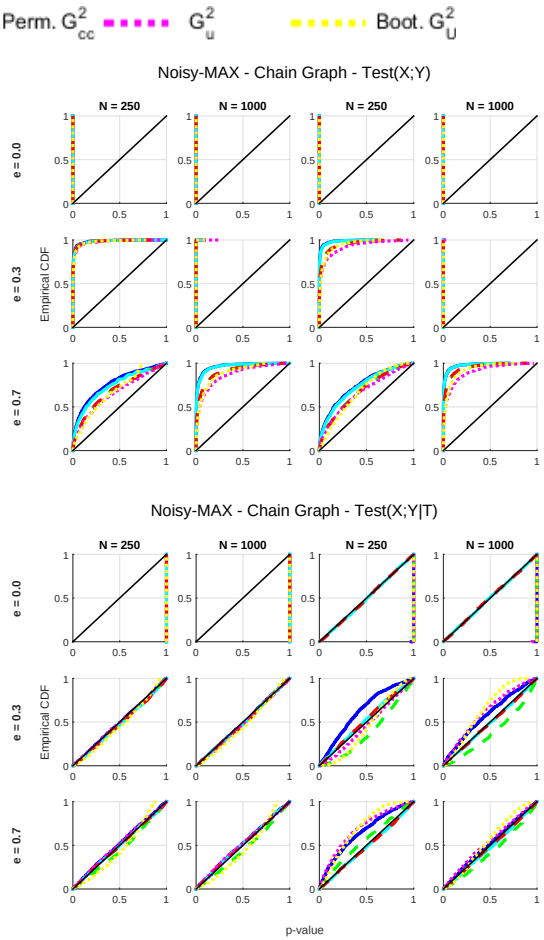

Figure 4: Results for the chain graph.

nately, we were not able to identify the circumstances under which this happens. For the chain graph, both tests perform reasonably well. In comparison to the G^2 and Permutation G^2 tests, the specialized tests have less power. Nevertheless, they will still be useful for network learning using the post-processing method, as it combines the best of both worlds. Again, the permutation test produces calibrated p-values for Test$(X; Y|T)$, whereas the asymptotic one does not.

G^2_u and Bootstrapping G^2_u. Again, both tests perform similarly to the other specialized tests. However, undersampling exhibits a large variance, as it highly depends on the selected samples. The bootstrapping version reduces this variance, but its distribution has a heavy tail close to one. This bias may be the result of taking the median p-value. We discourage the use of the bootstrapping method, but there may be other similar approaches that do not exhibit this behavior.

Comparison of G^2_{cc} and Permutation G^2_{cc} Figure 5 (a,b) shows that the G^2_{cc} and Permutation G^2_{cc} produce almost identical p-values on the unconditional tests. In Figure 5 (c,d) we compare the asymptotic and permutation tests for the conditional case with noisy data and $r = 4$ (Figure 4, bottom right). We see that in this case the G^2_{cc} does not produce the same p-values as the permutation G^2_{cc} test. However, the same behavior can be observed for the standard G^2 test and the permutation G^2 test. Also, for both cases, the p-values of the asymptotic tests are highly correlated with the ones of the permutation tests. The results suggest that the G^2_{cc} test is a reasonable approximation to the permutation version.

7.2 SENSITIVITY TO $P(\mathbf{T})$

We conducted a small experiment to investigate the sensitivity of the tests to the distribution of T. We simulated 1000 datasets from a Noisy-MAX collider graph with binary variables X and Y, with $e = 0$ and $N \in \{250, 1000\}$. The distribution of T is $P(T = 0) = 0.25$ and $P(T = 1) = 0.75$ and we selected 500 samples for each value of T. In order to measure the sensitivity of the methods to the specified marginal distribution we computed the area under the empirical CDF of the p-values. Values close to 0.5

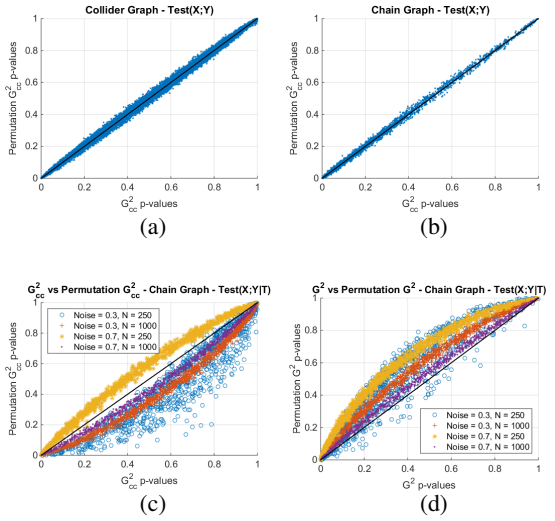

Figure 5: Comparison of the p-values from the G^2_{cc} and Permutation G^2_{cc} tests (a) on all tests for the collider graph (b) on the unconditional tests for the chain graph. Comparison on noisy data with $r = 4$ of (c) G^2_{cc} vs Permutation G^2_{cc} (d) G^2 vs Permutation G^2.

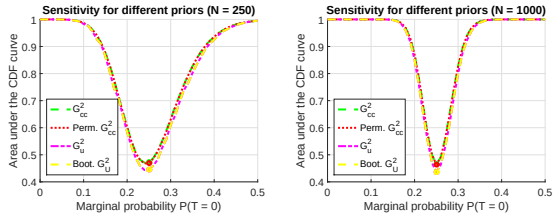

Figure 6: Sensitivity of methods to prior distribution.

indicate that the p-values are uniformly distributed (this is not always true, but should be reasonable in our case due to the convexity/concavity and monotonicity of the CDF; see Figure 3 for $e = 0$ and $N = 1000$). The results are summarized in Figure 6. We only show results from $P(T = 0) = 0$ to 0.5. We see that the methods are sensitive to the correct specification of the prior distribution, and that their sensitivity highly depends on the sample size. For $N = 250$, small deviations are acceptable, whereas for $N = 1000$ even deviations of 0.05 significantly reduce the ability of the tests to detect spurious dependencies. Of course, this can not be generalized and it may highly vary for different distributions, but it indicates that those methods have to be used with care.

7.3 INSURANCE NETWORK

We evaluated our methods on the INSURANCE network [Binder et al., 1997]. It contains 27 nodes and 52 edges. This network is appropriate for our purposes as it has many nodes for introducing spurious dependencies that also have relatively extreme distributions.

Table 1: Characteristics of the selection variables. The second row shows the number of spurious dependencies induced by selecting on those variables. The last four rows show their marginal distribution.

Node	18	19	20	21	25	26
Spurious	6	13	6	14	9	5
P(T = 0)	0.001	0.788	0.844	0.541	0.888	0.965
P(T = 1)	0.999	0.09	0.08	0.286	0.054	0.018
P(T = 2)	-	0.09	0.077	0.12	0.036	0.011
P(T = 3)	-	0.032	1.2e-05	0.052	0.023	0.007

7.3.1 Setup

We selected 6 nodes from the INSURANCE network as selection variables. The selection variables as well as their characteristics are shown in Table 1.

Algorithms. For Bayesian network learning we used the PC algorithm with Heuristic 3, as described in [Spirtes et al., 2000] (Section 5.4.2), except with an additional modification that sets an upper limit on the size of the conditioning set for each test. This is necessary especially for lower sample sizes, as conditioning on many variables tends to give very high p-values. In our simulations we set that parameter to 3 (maximum in-degree in the network). The significance level was set to 0.05 for all tests. We used 5 different methods to learn the network from case-control data: (a) G^2 test, (b) G^2_{cc} test for case-control data, (c) Undersampling G^2_u test, (d) G^2 test + post-processing with G^2_{cc} and (e) G^2 test + post-processing with G^2_u. Methods (a-c) did not apply the post-processing step. Whenever the G^2_u test was used, undersampling was performed only once for each dataset. In addition, we also ran the PC algorithm with the G^2 test on i.i.d. data to compare our methods against (Reference). The reference results should be close to the best achievable performance for a given sample size.

Data. Again, we generated data with equal proportions of each value of T. For each selection variable, as well as for the reference case, we generated 100 datasets for each of three different sample sizes $N \in \{1000, 10000, 100000\}$.

7.3.2 Results

The results are summarized in Table 2. For each method we report the extra edges ("+") and missing edges ("-"), averaged over all 100 runs.

PC with G^2 test. We observe that ignoring the sampling and using PC with the standard G^2 test performs very well for $N = 1000$ and does not identify many spurious edges. However, as expected, it identifies a significant amount of extra edges with larger sample sizes.

PC with G^2_{cc} and G^2_u tests. The case-control tests with-

Table 2: Results on the INSURANCE network. Extra edges are denoted with "+" and missing edges with "-".

	Method	T = 18		T = 19		T = 20		T = 21		T = 25		T = 26		Reference	
		+	-	+	-	+	-	+	-	+	-	+	-	+	-
N = 1K	G^2	1.00	26.83	0.10	29.10	0.22	28.81	0.15	29.54	0.49	28.59	0.10	28.91	0.14	28.73
	G^2_{cc}	0.35	32.19	0.28	33.51	0.23	34.20	0.36	30.97	0.21	34.44	0.31	35.11		
	G^2_u	0.33	32.15	0.28	34.15	0.23	34.41	0.21	32.51	0.32	34.82	0.30	34.94		
	$G^2 + G^2_{cc}$	1.00	26.84	0.09	29.10	0.22	28.88	0.15	29.54	0.49	28.65	0.10	28.94		
	$G^2 + G^2_u$	1.00	26.84	0.09	29.14	0.22	28.92	0.15	29.54	0.49	28.66	0.10	28.94		
N = 10K	G^2	6.44	11.24	3.04	14.33	1.84	14.15	0.85	14.54	3.70	14.70	1.61	13.65	0.23	14.35
	G^2_{cc}	0.00	16.96	0.00	18.89	0.00	19.81	0.00	16.38	0.00	20.82	0.01	21.90		
	G^2_u	0.00	16.93	0.00	20.20	0.02	20.66	0.00	17.19	0.10	21.18	0.00	22.08		
	$G^2 + G^2_{cc}$	0.50	13.07	0.18	16.51	0.00	16.72	0.00	16.32	1.09	16.72	0.07	16.13		
	$G^2 + G^2_u$	0.50	13.07	0.18	17.06	0.02	17.42	0.00	16.44	1.09	16.79	0.06	16.17		
N = 100K	G^2	7.00	5.00	5.21	7.09	3.97	6.17	4.84	7.66	4.67	8.10	2.74	7.75	0.01	8.65
	G^2_{cc}	0.01	10.96	0.01	11.01	0.00	11.03	0.01	10.97	0.02	11.14	0.02	11.43		
	G^2_u	0.01	10.96	0.01	11.08	0.00	11.10	0.00	10.99	0.00	11.26	0.01	11.44		
	$G^2 + G^2_{cc}$	0.00	5.00	0.16	8.89	0.15	8.78	0.00	7.66	0.44	9.09	0.49	8.76		
	$G^2 + G^2_u$	0.00	5.00	0.16	9.31	0.15	8.94	0.00	7.73	0.43	9.15	0.49	8.76		

out post-processing identify fewer extra edges at the cost of missing some edges. This happens both, because they are conservative and because of lower power than their i.i.d. counterparts, as we saw in previous experiments. Again, this improves with more samples but they do not seem to significantly outperform the G^2 test, at least in those experiments. Increasing the significance level may improve the situation.

PC with G^2 test + post-processing with G^2_{cc} and G^2_u tests. The best results, in terms of total number of errors, are achieved when post-processing is applied. For $N = 1000$ they perform similarly to the G^2 method without post-processing. This is expected, as the first step does not identify many extra edges and thus, post-processing is rarely applied. For larger sample sizes almost no spurious edges are identified, but a few more edges than G^2 without post-processing are missed. This happens because the post-processing rule is erroneously applied to edges that are not due to spurious dependencies and removes them.

Compared to the previous two methods without post-processing, slightly more edges are found but fewer edges are missed. This agrees with the simulations on the simple collider and chain graphs, which showed that the G^2 test is more powerful and therefore misses fewer edges than the G^2_u and G^2_{cc} tests.

Finally, the results are similar to the reference results, demonstrating the effectiveness of the proposed methods.

Comparison of G^2_u and G^2_{cc}. In all simulations the G^2_u test performs very similar to the G^2_{cc} test, with the latter being marginally better on average. Note however that averaging may hide the variance of the G^2_u test. In any case, undersampling is an alternative that can be generalized to other types of data, and should be further investigated.

8 CONCLUSION

We proposed methods to learn Bayesian networks from discrete, unmatched case-control data. We showed that one can first learn a network by ignoring the case-control sampling and then apply a post-processing step to remove spurious edges using a specialized test for case-control data. To do this the joint distribution of the selection variables must be available. In case it is not correctly specified the tests may fail to remove spurious edges. Finally, the trivial approach of undersampling seems to be a reasonable alternative, with the advantage that it easily generalizes to other types of data, such as continuous data with discrete selection variables. A drawback however is that it exhibits a large variance as it highly depends on the selected samples.

There is a lot of room for improvement and extensions. First, the proposed post-processing method could be improved to reduce the number of false removals of edges. Second, it is important to investigate additional case-control samplings, such as those from matched or nested studies. Finally, devising methods for other types of data, such as continuous data, would further broaden the scope. One possible approach would be to use or extend the ideas by Kuroki and Cai [2006] for recovering the population covariance matrix. Another possibility is to find a way to perform undersampling multiple times and combine the results appropriately.

Acknowledgements

We would like to thank Greg Cooper, Sofia Triantafillou and the anonymous reviewers for their comments. This work was partially funded by the ERC Consolidator Grant No 617393 CAUSALPATH and the Greek GSRT ARISTEIA II No 3446 Epilogeas.

References

E. Bareinboim, J. Tian, and J. Pearl. Recovering from Selection Bias in Causal and Statistical Inference. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-14)*, 2014.

T. Barrett, S. Wilhite, P. Ledoux, C. Evangelista, I. Kim, M. Tomashevsky, K. Marshall, K. Phillippy, P. Sherman, M. Holko, A. Yefanov, H. Lee, N. Zhang, C. Robertson, N. Serova, S. Davis, and A. Soboleva. NCBI GEO: archive for functional genomics data sets–update. *Nucleic Acids Res.*, 41(Database issue):D991–5, January 2013.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2-3):213–244, Nov. 1997.

N. Breslow. Statistics in epidemiology: The case-control study. *Journal of the American Statistical Association*, 91:14–28, March 1996.

N. Breslow and N. Day. *Statistical Methods in Cancer Research. Vol. 1 The Analysis of Case-Control Studies*. IARC, Lyon, 1980.

G. F. Cooper. A Bayesian Method for Causal Modeling and Discovery Under Selection. In *Proceedings of the 16th International Conference on Uncertainty in Artificial Intelligence (UAI 2000)*, 2000.

G. F. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. In *Proceedings of the 15th International Conference on Uncertainty in Artificial Intelligence (UAI 1999)*, 1999.

R. Edgar, M. Domrachev, and A. E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.*, 30(1):207–10, January 2002.

P. Good. *Permutation, Parametric, and Bootstrap Tests of Hypotheses*. Springer Series in Statistics. Springer, 3rd edition, 2004.

D. Heckerman and J. S. Breese. A new look at causal independence. In *Proceedings of the 10th International Conference on Uncertainty in Artificial Intelligence (UAI 1994)*, 1994.

A. Hyttinen, P. O. Hoyer, F. Eberhardt, and M. Jrvisalo. Discovering cyclic causal models with latent variables: A general sat-based procedure. In *Proceedings of the 29th International Conference on Uncertainty in Artificial Intelligence (UAI 2013)*, 2013.

M. Kuroki and Z. Cai. On recovering a population covariance matrix in the presence of selection bias. *Biometrika*, 93(3):601–611, 2006.

K. Murphy. The Bayes Net Toolbox for MATLAB. *Computing science and statistics*, 2001.

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

J. Pearl. *Causality, Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, USA, 2000.

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

K. J. Rothman, S. Greendland, and T. L. Lash. *Modern Epidemiology*. Williams & Wilkins, Philadelphia, PA: Lippincott, 3rd edition, 2008.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2000.

S. Srinivas. A generalization of the noisy-or model. In *Proceedings of the Ninth International Conference on Uncertainty in Artificial Intelligence (UAI 1993)*, 1993.

R. E. Tillman and P. Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, 2011.

S. Triantafillou and I. Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *CoRR*, abs/1403.2150, 2014.

I. Tsamardinos and G. Borboudakis. Permutation Testing Improves Bayesian Network Learning. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2010)*, 2010.

I. Tsamardinos, L. Brown, and C. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

A. Zagorecki and M. Druzdzel. Knowledge engineering for Bayesian networks: How common are noisy-max distributions in practice? In *Proceedings of 17th European Conference on Artificial Intelligence (ECAI 2006)*, 2006.