SIMULATION-BASED METHODS FOR INVESTIGATING THE IDENTIFIABILITY OF BAYESIAN NETWORKS WITH CROSS-SECTIONAL OBSERVATIONAL DATA

An Undergraduate Research Scholars Thesis

by

SAHIL PATEL

Submitted to the Undergraduate Research Scholars Program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Yang Ni

May 2020

Major: Computer Science

TABLE OF CONTENTS

ABSTRACT

Simulation-based Methods for Investigating the Identifiability of Bayesian Networks with Cross-sectional Observational Data

Sahil Patel
Department of Computer Science and Engineering
Texas A&M University

Research Advisor: Dr. Yang Ni
Department of Statistics
Texas A&M University

Bayesian networks are widely adopted to model complex systems by characterizing their information into conditional independencies of 2 or more system variables. For example, Bayesian networks have been commonly used for identifying gene regulatory networks and modeling decision networks in machine learning. While being popular, the structure of a Bayesian network is usually unknown and has to be inferred from available data in most of the cases.

To date, learning the structure of Bayesian networks is still a very challenging and nuanced task partly due to the non-identifiability issue of Bayesian networks, especially when the data are cross-sectional and observational. In this thesis, we are going to use simulation-based approaches to investigate precisely under what conditions a Bayesian network can be identifiable, and therefore recoverable, for cross-sectional observational data. We will also explore required assumptions and overall implications of our work.

ACKNOWLEDGMENTS

I want to sincerely thank Dr. Yang Ni for his incredible help throughout the project.

SECTION I

INTRODUCTION AND BACKGROUND

Bayesian Networks

Graphical models are probabilistic models for multivariate random variables whose Markov properties (i.e., conditional independencies) are characterized by an underlying graph. Graphical models provide a compact representation of joint distribution and allow for local computations via Gibbs factorization. In this thesis, we are going to focus on one type of graphical models, namely, the Bayesian networks which assume the joint distribution of the multivariate random variables factorizes with respect to a directed acyclic graph (DAG). DAG is a directed graph $G = \{V, E\}$ with a set of vertices $V = \{v_1, v_2, ...v_n\}$ and a set of directed edges $E = \{e_1, e_2, ...e_m\}$ that do not form directed cycles. That is, starting from any node, one cannot return to itself by following the direction of the edges. In Bayesian networks, vertices represent random variables and the directed edges encode the conditional independence. In addition, the directed edges can be potentially interpreted as causal relationships under the following assumptions [1, 2]: (i) *causal Markov*: the conditional independence relationships encoded in the DAG hold in the population, (ii) *faithfulness*: the conditional independence relationships encoded in the DAG are the only ones that hold in the population, and (iii) *causal sufficiency*: there is no unmeasured con-founder. Without causing confusions, we will use the terms Bayesian network and DAG interchangeably hereafter.

A probability distribution is said to factorize with respect to a DAG if the joint distribution can be written as a product of local distributions with each local distribution being the conditional distribution of each vertex given its parents (vertices pointing towards the given (child) vertex). There are two immediate implications of such factorization. First, the computation of the jointly distribution is much simplified because each vertex usually only has a handful number of parent vertices in a sparse network. Second, factorization implies all Markov properties (i.e., condition independence assertions). For instance, each vertex is conditionally independent of its non-descendants

given it parents. The Markov properties greatly improve the interpretation and extensibility of a Bayesian network.

Because Bayesian networks provide a straightforward framework to define conditional independence (lack of edges) as well as causal relationships (directed edges) of a complex multivariate system, they have become very popular in various research areas such as biomedical science, chemistry, computer science, material engineering, and artificial intelligence. In these areas, researchers can use the compact representation of Bayesian networks to model the complex systems to then use statistical analysis [3, 4, 5, 6].

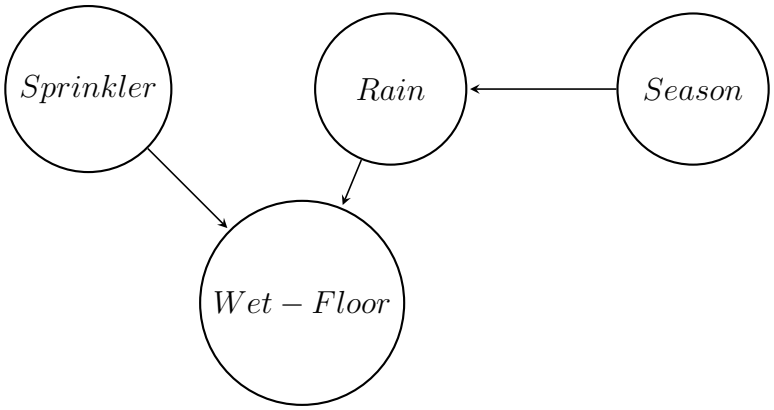


Figure 1: Example DAG with Practical Elements

An example of a practical DAG is shown in Figure 1 for the causal relationships among the following variables: Rain, Sprinkler, Season and Wet-Floor. We know that 'Rain' and 'Sprinkler' can make the floor wet. These relationships are signified by the arrows or directed edges from the vertices 'Rain' and 'Sprinkler' to the vertex 'Wet-Floor'. In addition, we also know that 'Season' (whether it is a rainy-season or not) will cause 'Rain' in expecting rain when the season permits it, which in turn causes 'Wet-Floor'. All these relationships are concisely summarized by the Bayesian network in Figure 1. Given the network structure, we can read off the conditional independence from the graph, e.g., 'Wet-Floor' is independent of 'season' given 'Sprinkler' and

'Rain'. However, in many applications, these causal relationships are unknown especially for large systems or the task is to find these relationships themselves. What we get to instead observe in practice are often just the measurements of these variables. The general structure in this problem is to track statistical analyses on the network using only variable data.

Therefore, the focus of this thesis is to learn the network structure based on observations of the random variables without assuming any prior knowledge of their causal relationships. Learning this network structure can ease computations and provide researchers comfort-ability in getting access to greater amount of information. In the example above, these variables may take Boolean/binary values: 0 or 1, asserting as either existing or not. For instance, 'Rain' = 1 or 0 indicates whether it has or has not rained; similar interpretation applies to 'Sprinkler','Season', and 'Wet-Floor'. In all, the network can provide time-dependent observational data wherein the variables potentially form a complex joint density function. With multiple realizations of these variables, one may be able to reverse engineer the network structure.

Depending on the applications, variables represented by the vertices can take more than two values and they will be modeled with distinct distributions with associated density functions. For example, we will use Bernoulli distribution for binary variables, Poisson distribution for count/integer variables, Gaussian distribution for continuous variables taking values on the entire real line, and gamma distribution for continuous variables taking values on the positive real line. In addition to discrete and continuous random variables, we will also consider mixed distributions such as zero-inflated (ZI) distributions and Censored distributions. With this point of view, we will use cross-sectional observational variable data to learn the network structure. In this, we will need to take into account possible network graph structure that the data can take, and recovering the correct one as the estimated resultant structure.

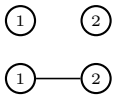
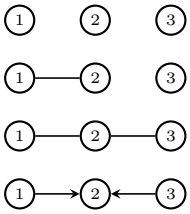
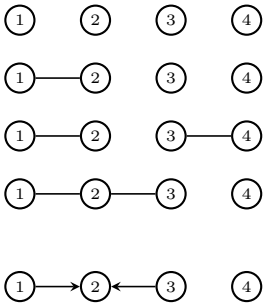
Markov Equivalence Class

As mentioned earlier, the conditional independence relationships are encoded in the structure of DAG. However, two distinct DAGs do not necessarily define different conditional independence assertions. In fact, DAGs can be grouped into Markov equivalence classes (MECs): within

each class, all DAGs represent exactly the same conditional independence structure. Markov equivalent DAGs are those with the same skeleton and v-structures [6]. These equivalence relationships among DAGs can lead to non-identifiable issue in identifying unique DAGs given an observational dataset. As we will discuss later, to classify a DAG with a certain distribution as identifiable, we must show that DAGs within the same MEC have different distribution. For example, suppose we are given data for two variables and the true data generating mechanism is given by the DAG $1 \longrightarrow 2$, i.e., vertex 1 is the cause of vertex 2. This DAG is Markov equivalent to $1 \longleftarrow 2$. To be able to identify the true data generating graph, we will have to show that (1) the true structure is different from the other incorrect structure(s) and (2) the true structure has the highest likelihood compared to the other incorrect structure(s).

MECs are provided in Table 1 [6] for n = $\{2, 3\}$ nodes and partially for n=4. The undirected edges represent the existence of an edge between the two nodes, wherein either of the nodes can be the parent or the child of the other node. Other columns in the table include Markov properties of each class ("Markov Property"), the number distinct DAGs within each MEC discounting the effect of labeling ("Number of Graphs"), and the total number of DAGs ("Possible DAGs") which represents the number of DAG structures that need be accounted for when learning from data. In our example (Figure 1), the DAG is part of MEC with n = 4 where Node '1' = 'Sprinkler', Node '2' = 'Wet-Floor', Node '3' = 'Rain' and Node '4' = 'Season'. According to the column "Markov Property", 'Sprinkler' is independent of 'Rain' and 'Season' and 'Web-Floor' is conditional independent of 'Season' given 'Sprinkler' and 'Rain'. Unfortunately, the number of DAGs grows super-exponentially with respect to the number n of nodes [6, 5, 7]. Therefore, for our experiment, we will empirically investigate the identifiability properties of DAGs with up to four nodes for various distributions; but there is no obvious reason for us to question the validity of our results for networks with more than four nodes.

Table 1: Markov Equivalence Classes and Graphs

Number of Variables	Graph Structure	Markov Property	Number of Graphs	Possible DAGs
2		$1 \perp 2$	1	
		(none)	2	
				3
3		$1 \perp 2 \perp 3$	1	
		$(1,2) \perp 3$	2	
		$1 \perp 3 \mid 2$	3	
		$1 \perp 3$	1	
		(none)	6	
				25
4		$1 \perp 2 \perp 3 \perp 4$	1	
		$(1,2) \perp 3 \perp 4$	2	
		$(1,2) \perp (3,4)$	4	
		$1 \perp 3 \mid 2$ $(1,2,3) \perp 4$	3	
		$1 \perp 3$ (1,2,3) $\perp$ 4	1	
		(1,2,3) $\perp$ 4	6	
		$1 \perp 3 \mid 2$ (1,2)$\perp$ 4|3	4	
		$1 \perp (3,4)$ $2 \perp 4 \mid 1,3$	2	
	...	...	...	
				543

DAG Learning

DAG learning approaches generally fall into two categories, 1) search-and-score methods and 2) constraint-based methods. Search-and-score approaches typically involve two parts. In the first part, the algorithm defines a metric (e.g., the likelihood) for DAGs. In the second part, the algorithm searches the DAG space to identify the optimal DAG with the highest score. On the other hand, constraint-based methods iteratively test for conditional independence between variables to remove non-significant edges and orient edges when possible. Constraint-based approaches are usually much less computationally straining since one can go through possible edges potentially in polynomial time when DAGs are sparse, which is generally the case in real-world applications [8].

PC Algorithm

One of most popular algorithms for learning DAG structure (up to the MEC) is the constraint-based method, Peter and Clark (PC) algorithm [9]. The algorithm starts with a complete undirected graph. It then iteratively deletes the non-significant edges and possibly orients the undirected edges according to conditional independence test. The pseudo-code for the PC algorithm [9, 4] can be found in Algorithm 1. The input for the algorithm is the multivariate data and the output is an optimal essential graph (the union of all DAGs within a MEC).

Algorithm 1: PC Algorithm

Input: Vertex Set V

Output: Essential Graph C for V

l = -1; C = Complete Convoluted DAG for V;

do

l = l+1;

do

Select a (new) ordered pair of nodes i, j that are adjacent in C such that $|adj(C, i) \ / \ \{j\}| \geq l$

do

Choose (new) k $\subseteq adj(C, i) \ / \ \{j\}$ with |k| = l

if (i and j are conditionally independent given k)

Delete edge i, j;

Denote this new graph by C;

Save k in S(i,j) and S(j,i);

end

while (!(edge i, j is deleted or all $k \subseteq adj(C, i) \ / \ \{j\}$ with |k| = l have been chosen))

while (!(all ordered pairs of adjacent variables i and j such that $|adj(C, i) \ / \ \{j\}| \geq l$ and $k \subseteq adj(C, i) \ / \ \{j\}$ with |k| = l have been tested for conditional independence))

while (**for each** (!(ordered pair of adjacent nodes i,j: $|adj(C, i) \ / \ \{j\}| < l$)))

G = C;

for each (pairs of nonadjacent variables i, j with common neighbour k)

if ($k \notin S(i, j)$)

Replace i–k–j in G_{skel} by $i \rightarrow k \leftarrow j$;

end

end

In the resulting DAG, try to orient as many undirected edges as possible by rules:

R1: Orient j–k into j→k whenever there is an arrow i→j such that i and k are nonadjacent.

R2: Orient i–j into i→j whenever there is a chain i→k→j.

R3: Orient i–j into i→j whenever there are two chains i–k→ j | l and k | i –l→j such that k and l are nonadjacent.

Identifiability

The PC algorithm and many other constraint-based learning algorithms only identify the optimal MEC represented by an essential graph. Moreover, this essential graph can be a shared representation for multiple network structures Table 1. In other words, these algorithms do not identify a unique Bayesian network in most of cases; which can potentially provide more rewarding information about the networks variables.

In order to uniquely identify optimal causal DAGs, we will focus on score-and-learning approaches which have the potential in differentiating DAGs within the same Markov Equivalence Class. The possibility of these approaches to actually work and recover the true structure defines identifiability of that network in question. However, as we will show later, identifiability is non-trivial and it is the underlying distribution that determines whether we can identify the correct DAG within the MEC.

First of all, when no distribution assumption is made (e.g., the PC algorithm), DAGs are non-identifiable. These findings were echoed in many of the existing learning algorithms. Imposing certain distribution assumptions can uniquely identify causal DAGs. However, not all distributions were created equal. For example, in causal DAG with two variables, X and Y, DAG X $\rightarrow$ Y and DAG X $\leftarrow$ Y will have the exactly same likelihood if the distribution is assumed to be normal and hence they are not identifiable. Interestingly, additional assumptions (e.g., those described in [7]) will allow for unique identification. In [7], they assumed a simultaneous equation model (SEM) with centered-Gaussian noises with equal variances. With this additional assumption, DAGs within the same MEC have distinct likelihoods and the correct structure can be identified through the distinction. Recently, the work by Park and Raskutti [10] found Bayesian network Models with Poisson distributions identifiable within the same Markov Equivalence Class. This result was later extended to the generalized hypergeometric distributed Bayesian networks, adding to the identifiability classification [11]. In addition, Hoyer et al. [12] and [13] showed that nonlinear DAGs modeled by nonlinear SEMs and linear SEMs with non-Gaussian errors are identifiable.

Despite these encouraging results, the identifiability properties of many other distributions

(such as beta and zero-inflated distributions) are still unknown. These distributions find many important applications in practice such as DNA methylation which takes value between 0 and 1 and scRNA-seq data which are zero-inflated counts. We aim to fill this gap in this thesis by empirically investigating the identifiability of a broad range of distribution types as to understand conditions that stimulate identifiablility in Bayesian networks.

SECTION II

METHOD AND EXPERIMENT

Main Problem Statement

We will look at identifiability of graph structures for DAGs with a wide range of distribution types. As we have mentioned above, some existing work has empirically or mathematically proved identifiability (ability to pick the correct and specific graph structure, past learning the MEC) for some distribution types (e.g. Poisson). However, investigation into identifiability of a broader range of distribution types still remains lacking. We will empirically address this facet of the identifiability problem.

We will adopt a simulation-based approach. We will simulate data from some underlying "true DAG" and then compute the likelihoods of the true DAG and its Markov equivalent DAGs. If the true DAG always has the highest likelihood, then we say the DAG is identifiable. As we will see later, when a DAG is non-identifiable (e.g., a Gaussian DAG), all Markov equivalent DAGs have exactly the same likelihoods. We will first validate this procedure for DAGs with known theoretical results and then apply it to distributions with no prior identifiability results. In summary, our focus is on finding which distribution allows for differentiation between true DAG structure from other DAGs that are part of the same MEC. We will also consider differentiating true DAG from other DAG structures with the same number of variables which are not necessarily in the same equivalence class. We will simulate data from all DAGs with $n \leq 4$ as provided in Table 1.

We will investigate identifiability properties for DAGs with various distributions including discrete (Poisson, Bernoulli, binomial), continuous (Gaussian, beta, gamma), zero-inflated model (beta, gamma, negative binomial, Poisson), hurdle model (Poisson, negative binomial) and censored (Poisson) distributions. Each of these distributions finds useful applications in a broad range of research areas. We will first simulate data based on the DAGs in Table 1. We will then check

identifiability by computing the likelihood-based score for each DAG. Using these scores, we will be able to say under which scenario DAGs are identifiable. We now describe the details of the approaches for data generation and identifiability check.

DAG:

Markov Property:

$$1 \perp 3$$

Adjacency Matrix:

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 2: Mathematical Representation of DAG structure

Data Generation

In our data generating algorithm, we will mathematically represent the graph structure by an adjacency matrix. The adjacency matrix defines which pairs of nodes are connected or disconnected in the graph. The adjacency matrix is an n by n square matrix of Boolean values (0 or 1). The ith row of the adjacency matrix defines the incoming edges of vertex i as 1's and missing incoming edges as 0's, for $i = 1, \ldots, n$. Note that the diagonal entries are always 0's as DAGs do not allow self-loops. These 1's in the row represent the variable's parents. An example of representing a DAG as an adjacency matrix is shown in Figure 2. In the example, variables 1 and 3 do not have any parents, as such row 1 and 3 are only 0's. On the other hand, variable 2 has two parents, namely 1 and 3. Therefore, the entries (2,1) and (3,1) are 1's. This process is used to define all DAG structures in the Markov Equivalence Class (MEC).

Given a DAG structure, we will simulate data from different probability distributions which factorize with respect to the DAG. Using the factorization, we can first generate m realizations of

variables that do not have parents from the marginal distributions and then recursively simulate the values of the child nodes given the values of their parents. We will provide the list of probabilistic distributions in a later section. Taking Poisson DAGs [10] as an example, we incrementally move through the variables in the for-loop and for each of them, we define a Poisson regression model that regresses the current In variable on its parents, from which we will sample for m times. When the variables do not have parents, they are generated from Poisson distribution.

In general, to find the vertices with no parents, one can rearrange the rows and columns of the adjacency matrix to be a lower triangular matrix so that the first vertex in this new ordering is guaranteed to have no parents. This ordering allows the algorithm to first generate the parents before moving to generate their child variables that require the data from those parents to define their probability distribution functions. The psuedo-code for our data-generation algorithm is shown in Algorithm 2.

Algorithm 2: DAG Generation Algorithm for Poisson

Given: Number of variables N, Number of sampling M, Adjacency matrix $A_{N\times N}$

$X_{M\times N} \leftarrow 0$: DAG Matrix with N variables as columns with M data-points each

$R_{N\times N} \leftarrow LowerTriangleRep.[A]$

for $k = 1 : N$ **do**

  $X_{:,v} \leftarrow \mathcal{P}(\lambda = (0.3 + R_{k,:} \times e^{X_{:,v}}))$ where v = variable in X for k^{th} row in R

end

Result: Data matrix X

Identifiability Check

We will now attempt to identify this original true DAG structure using only the multivariate data. Empirically, through this process, we will try to only use observational samples of the variables to find which variables are related to each-other. In essence, we will check if it is possible to identify the correct structure from all possible Bayesian network structures with the same number of variables n. Moreover, this identification of the true structure has to also differentiate between the possible structures in the same Markov Equivalence Class, going further than PC algorithm.

We know from previous research that this check of identifiability depends on the underly-

ing distribution assumption of the variables. DAGs are indistinguishable within the same MEC without additional assumption. Recent work has looked at different types of distributions and proved some of them as identifiable. For example, while we know that Gaussian distributed DAG is non-identifiable, recent look at specific types of Gaussian distributions such as those with equal residual variances by Peters and Buhlmann have shown otherwise [7]. In our work, we will repeat this process for other popular distributions as well as repeat some of the already established distributed DAG types. List and details of our implementations can be found in the next section. Moreover, We have created a table of these recently discovered as identifiable/non-identifiable as well as classifications from our work in the results section later.

In general, we start the Identifiability Check sub-part of our work with data from the Data Generation sub-part as well as the Adjacency matrix of the true structure of the variables. Next, we add this adjacency matrix into a list of other adjacency matrices that represent possible structures with the same number of variables (this can be changed to only include those within the same MEC for actual check of identifiability). Now, we will attempt to fit all the structures by incrementally going through the list of adjacency matrices. We will iterate through the variables and compute the log-likelihood (conditional on the parents). More details on this procedure for checking identifiability can be found in later section [14, 15, 16, 17, 18]. The psuedo-code for our identifiability check can be found in Algorithm 3.

Algorithm 3: Identifiablity Algorithm

Given: Multivariate data matrix X, Adjacency matrices A's

$X_{M\times N}$ $\leftarrow$ DAG Matrix with N variables as columns with M data-points each

$A_{N\times N}$ $\leftarrow$ Adjacency Matrix representation of the True structure

$T_{N\times N\times P}$ $\leftarrow$ N$\times$N Adjacency Mat. for all P graphs in a Markov Eq. Class

$T_{N\times N\times g}$ $\leftarrow$ A

$L_{P\times 1}$ $\leftarrow 0$: Log Likelihood Values for each Graph in Vector form

for *i = 1 : P* **do**

  for *j = 1 : N* **do**

    L_i += log (max_π Likelihood(π | $X_{:,i} \sim (X_{:,j} \times T_{:,j,i})$)

  end

end

if$(L_g > L_{(all\ except\ g)})$ where L_g is the log-likelihood of the true DAG

  return True

else

  return False

Result: True or False for whether the specific DAG can be identified

Discrete Distributions

Prior work in identifiability of discrete distribution types have shown that Poisson and Binomial distributed DAGs are identifiable and Bernoulli/Binary distributed DAGs are non-identifiable. We list the discrete distributions under consideration below.

Poisson Distribution

Parameters	$\lambda = \lambda_0 + e^{p_1 + \dots p_k}$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$\frac{(\lambda)_i^x e^{(\lambda)}}{x_i!}$

Binomial Distribution

Parameters	$v = v_0 + e^{p_1 + \dots p_k}/(1 + e^{p_1 + \dots p_k})$,$u = v * m$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$\binom{m}{u} v^u (1-v)^{m-u}$

Bernoulli Distribution

Parameters	$v = v_0 + e^{p_1 + \dots p_k}/(1 + e^{p_1 + \dots p_k})$,$u \in \{0, 1\}$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$v^u (1-v)^{1-u}$

Continuous Distributions

Prior work has shown that general Gaussian/Normal distributed DAGs are non-identifiable. We will explore the identifiability properties of Beta and Gamma distributed DAGs. We list the continuous distributions under consideration below.

Normal Distribution

Parameters	$\mu = \mu_0 + p_1 + ...p_k/k$,x
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$\frac{1}{\sqrt{2\pi}} e^{-(x-\mu)^2/2}$

Beta Distribution

Parameters	$\alpha = \alpha_0 + e^{p_1+...p_k}/(1+e^{p_1+...p_k})$, $\beta = \beta_0 + e^{2*(p_1+...p_k)}/(1+e^{2*(p_1+...p_k)})$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}}$

Gamma Distribution

Parameters	$\alpha = \alpha_0 + e^{p_1+...p_k}/(1+e^{p_1+...p_k})$, $\beta = \beta_0 + e^{2*(p_1+...p_k)}/(1+e^{2*(p_1+...p_k)})$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$

Zero-inflated Distributions

No prior work has been developed for identifiability in zero-inflated distributions. Zero-inflated distributions add additional probability at zero compared to standard distributions. We list

the zero-inflated distributions under consideration below.

Zero-inflated Poisson Distribution

Parameters	$\lambda = \lambda_0 + e^{p_1+...p_k}, z = \lambda/(1+\lambda)$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$P(X = x_i) = \begin{cases} z + (1-z)PoisDist(x_i = 0), & x_i = 0 \\ (1-z)e^{-\lambda}\lambda^{x_i}/x_i!, & x_i = 1, 2... > 0 \end{cases}$

Zero-inflated Negative Binomial Distribution

Parameters	$v = v_0 + e^{p_1+...p_k}/(1+e^{p_1+...p_k}), \lambda = \lambda_0 + e^{p_1+...p_k}, z = \lambda/(1+\lambda), r = v * m$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$P(X = x_i) = \begin{cases} z + (1-z)NegBinomDist(x_i = 0), & x_i = 0 \\ (1-z)\dfrac{\Gamma(x_i + 1/v)}{\Gamma(x_i)\Gamma(1/v)}\dfrac{vr_i^x}{1+vr^{x_i+1/v}}, & x_i > 0 \end{cases}$

Zero-inflated Beta Distribution

Parameters	$\alpha = \alpha_0 + e^{p_1+...p_k}/(1+e^{p_1+...p_k})$, $\beta = \beta_0 + e^{2*(p_1+...p_k)}/(1+e^{2*(p_1+...p_k)}), \lambda = \lambda_0 + e^{p_1+...p_k}, z = \lambda/(1+\lambda)$,x
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$P(X = x_i) = \begin{cases} z + (1-z)BetaDist(x_i = 0), & x_i = 0 \\ (1-z)\dfrac{x_i^{\alpha-1}(1-x_i)^{\beta-1}}{\dfrac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}}, & x_i > 0 \end{cases}$

Zero-inflated Gamma Distribution

Parameters	$\alpha = \alpha_0 + e^{p_1+...p_k}/(1+e^{p_1+...p_k})$,
	$\beta = \beta_0 + e^{2*(p_1+...p_k)}/(1+e^{2*(p_1+...p_k)}), \lambda = \lambda_0 + e^{p_1+...p_k}, z = \lambda/(1+\lambda)$,x
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$P(X = x_i) = \begin{cases} z + (1-z)GammaDist(x_i = 0), & x_i = 0 \\ (1-z)\dfrac{\beta^\alpha}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\beta x_i}, & x_i > 0 \end{cases}$

Hurdle Models

No prior work has been developed for identifiability in hurdle models. Hurdle models are somewhat similar to zero-inflated distributions in that they have a larger amount of zeros than possible in general distributions. Essentially, the probability function produces values similar to the general distribution, but, it must be greater than some value (usually zero) to qualify as being counted as part of a density. A more applicable approach to this is in using a binary response and a zero-truncated response. We list the hurdle models under consideration below.

Poisson Hurdle Model

Parameters	$\lambda = \lambda_0 + e^{p_1+...p_k}, h = h_0 + e^{e^{-(p_1+...p_k)}}$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$P(X = x_i) = \begin{cases} h, & x_i = 0 \\ (1-h)\dfrac{e^\lambda \lambda^{x_i}}{(1-e^\lambda)x_i!}, & x_i > 0 \end{cases}$

Negative Binomial Hurdle Model

Parameters	$v = v_0 + e^{p_1+...p_k}/(1+e^{p_1+...p_k}), r = p*m, h = h_0 + e^{e^{-(p_1+...p_k)}}$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$P(X = x_i) = \begin{cases} h, & x_i = 0 \\ (1-h)\frac{\Gamma(x_i+1/v)}{\Gamma(x_i)\Gamma(1/v)}\frac{vr_i^x}{1+vr^{x_i+1/v}}\frac{1}{1-NegB(x_i=0)}, & x_i > 0 \end{cases}$

Censored Models

No prior work has been developed for identifiability in censored distributions. Censored models also juggle the occurrences of zeros into the original general form of a distribution. We define a threshold value wherein all values greater than it will be censored back. One approach is to equate definition to that of zero-inflated Poisson with right-side direction distinction instead. We list the censored distributions under consideration below.

Censored Poisson Model

Parameters	$\lambda = \lambda + e^{p_1+...p_k}, c = \lambda/(1+\lambda)$
Parents(If any)	$p_1, p_2, ..., p_k$
Distribution	$P(X = x_i) = \begin{cases} 1-(e^{-\lambda}\sum_{i=0}^{c-1}\lambda^i/i!), & x_i \geq c \\ \frac{e^\lambda \lambda^{x_i}}{(1-e^\lambda)x_i!}, & x_i < c \end{cases}$

SECTION III

RESULTS

Non-identifiable Distributions

The results are summarized in Table 2. We first confirmed that as predicted by the well-known theories, Gaussian and Bernoulli DAGs are not identifiable: DAGs within the same MEC have identical likelihood value. These two distributions serve as negative controls which demonstrated that our simulation-based approach will not falsely flag non-identifiable DAGs as identifiable DAGs.

Identifiable Distributions

On the other hand, all the other models were identifiable in our simulation studies. Some of the distributions (such as Poisson and negative binomial) are supported by the existing theories. These distributions serve as positive controls which showed that our simulation-based approaches will not falsely flag identifiable DAGs as non-identifiable DAGs. Confirmed by both positive controls and negative controls, we are confident in interpreting the findings of identifiability for distributions with unknown theoretical results.

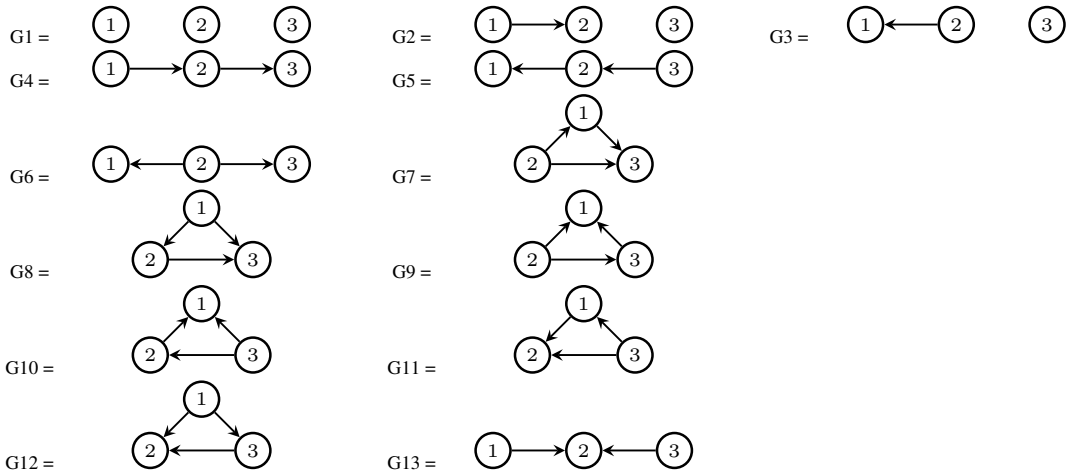
In our new results, we have found that DAG structures with beta, gamma, zero-inflated distributions, hurdle models, and censored Poisson are identifiable. Particularly we found the true data generating DAG always had higher likelihood than all the other DAGs within the same MEC (also true for DAGs outside the MEC). One key note we see in our results is that most distributions are in fact identifiable.

In Figure 3, we demonstrate our process for the general Poisson DAGs. We used our process for Bayesian networks with number of variables ≤ 4, with checking and processing all possible network structures. In the figure, we first define the graphs with number of variables = 3 with their labels going from G1 to G13 (with G13 being the special v-shape). In the two matrix

depictions, we show the negative log-likelihood values for fitting the true structure (X-Axis) on all possible graphs (Y-Axis). The identifiability can be inferred from the fact that the correct model in each of the columns (located in the diagonals) has the lowest negative log-likelihood (i.e. the highest likelihood). We also use dark color to indicate large negative log-likelihood values and hence the lowest value has no grey-scale. A similar story is shown in the heatmap representation of the matrix. Again, the diagonal and correct identification is inferred.

Again, this classification means that in the Bayesian Network with variable data distributed as such, it is possible to learn how the edges are oriented. It is possible to know how the variables are related to each other and therefore be able to use other similar analysis that depend on understanding graph structures or even understanding dependencies of variables. Below in Table 2, we summarize our results as well as show other classifications from other previous works.

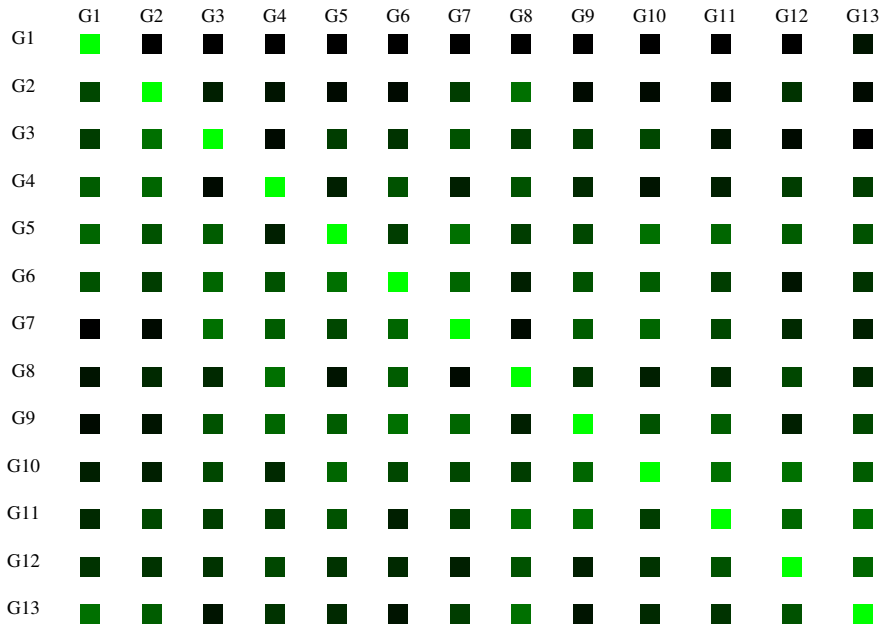
Table of Raw Values in Graph Checking as AIC Values (Which are Derived using Likelihood Function for the Model to the Data)



Fit Structure [Y-Axis] vs True Structure [X-Axis]

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13
G1	422.9												724.0
G2	424.7	503.6	589.2	$7E+5$	$1E+7$	1000.6	576.9	282.7	$9E+4$	$8E+7$	859.1	$3E+5$	725.7
G3	424.7	505.2	474.0	$7E+5$	$5E+6$	862.6	433.3	288.1	5791.6	1052.0	844.6	$9E+5$	
G4	424.6	505.3	589.6	1373.5	$1E+7$	805.2	578.9	284.7	$9E+4$	$8E+7$	818.0	$3E+5$	703.4
G5	424.5	506.8	475.5	$7E+5$	$3E+6$	825.2	430.7	288.1	5761.3	826.40	703.6	$9E+2$	553.6
G6	424.6	506.9	475.4	1636.5	$3E+6$	662.3	432.3	290.1	5655.5	1015.8	803.5	$9E+5$	703.6
G7		508.7	474.4	1636.0	$5E+6$	667.7	430.3	292.1	5653.8	1014.1	798.8	$9E+5$	705.1
G8	426.5	507.1	589.0	1375.1	$1E+7$	800.7	580.9	280.7	$9E+4$	$9E+7$	813.2	$3E+5$	704.9
G9	426.5	508.6	477.4	1601.0	$3E+6$	667.0	432.3	290.1	763.3	1017.6	745.9	$9E+5$	700.1
G10	426.5	508.5	477.4	$6E+5$	$3E+6$	820.0	434.3	288.1	869.1	824.2	645.9	869.3	550.1
G11	426.5	506.9	575.5	$6E+5$	$3E+6$	895.1	576.9	282.7	867.2	$5E+6$	640.3	875.3	546.6
G12	426.5	507.0	587.9	$5E+5$	$1E+7$	890.5	578.9	284.7	$9E+4$	$8E+7$	770.5	732.6	546.7
G13	424.5	505.4	589.6	$7E+5$	$1E+7$	991.0	576.9	282.7	$9E+4$	$8E+7$	812.6	913.1	546.1

Values are Gray-scaled: Darker the shading in the column, the larger the value. No Shading (White) = Smallest Value.



Heat Map

Figure 3: Illustration of Poisson DAGs with n=3

Table 2: Identifiability

Distribution Name	Unidentifiable	Identifiable
Gaussian	✓	
Non-Linear Gaussian		✓
Non-Parametric Non-Gaussian		✓
Gaussian with Equal Residual Variances		✓
Generalized Hypergeometric		✓
Bernoulli	✓	
Poisson		✓
Negative Binomial		✓
Binomial		✓
Beta		⋆
Gamma		⋆
Zero-Inflated Beta		⋆
Zero-Inflated Gamma		⋆
Zero-Inflated Negative Binomial		⋆
Zero-Inflated Poisson		⋆
Hurdle Poisson		⋆
Hurdle Negative Binomial		⋆
Censored Poisson		⋆

✓ Prior Work
⋆ Our Contribution

SECTION IV

CONCLUSION

Conclusion and Future Work

In this work, we have used simulation-based approaches to study the identifiability properties of various types of Bayesian networks. We mentioned some applications, difficulties and approaches of solving the problem. We then reviewed one of the approaches, namely the PC Algorithm, and then transitioned into exploring identifiability of specific network graph structures from MECs, shown in Table 1 [6]. The relatively new problem was previously intractable, however, recent approach in categorizing based on distribution types have shown otherwise [5, 10, 11, 7, 12, 19, 20]. We extend these works to other popular distributions including ones that are Gamma, Beta, Zero-inflated Beta, Zero-inflated Gamma, Zero-inflated Negative Binomial, Zero-inflated Poisson, Hurdle Poisson, Hurdle Negative Binomial, and Censored Poisson, all summarized in Table 2. As negative and positive controls, we have also shown that our results are consistent with existing theoretical results.

Our extensive look at identifiability of Bayesian networks with broad range of distributions suggest Bayesian networks are generally identifiable. With the two exceptions (confirmed by us) being unrestricted general Gaussian and Bernoulli, we were able to identify the correct graph structures - and therefore learn the Bayesian network itself - for models with all the different distribution families.

A natural step forward from our empirical approach is a theoretical investigation of identifiability in Bayesian networks for which no theoretical results are available. Furthermore, we have considered a small number of variables in our simulations. For Bayesian network with moderate to large number of variables, exhaustive enumeration of all relevant DAGs (which was done in this paper) and computing their respective likelihood become infeasible. Therefore, in our future work, we will design more efficient search-and-score algorithms to explore the DAG space.

REFERENCES

[1] P. Spirtes, R. Scheines, C. Glymour, T. Richardson, and C. Meek, "Causal inference," *Handbook of Quantitative Methodology in the Social Sciences*, pp. 447–478, 2004.

[2] J. Pearl, *Causality: Models, Reasoning and Inference*. USA: Cambridge University Press, 2nd ed., 2009.

[3] A. Mittal and A. Kassim, *Bayesian network technologies: Applications and graphical models*. 01 2007.

[4] M. Kalisch and P. Bühlmann, "Estimating high-dimensional directed acyclic graphs with the pc-algorithm," *J. Mach. Learn. Res.*, vol. 8, p. 613–636, May 2007.

[5] S. Andersson and M. Perlman, "Normal linear regression models with recursive graphical markov structure, by andersson, steen a.; perlman, michael d.," Jan 1998.

[6] S. A. Andersson, D. Madigan, and M. D. Perlman, "A characterization of markov equivalence classes for acyclic digraphs," *Ann. Statist.*, vol. 25, pp. 505–541, 04 1997.

[7] J. Peters and P. Bühlmann, "Identifiability of gaussian structural equation models with equal error variances," *Biometrika*, vol. 101, p. 219–228, Nov 2013.

[8] T. D. Le, T. Hoang, J. Li, L. Liu, H. Liu, and S. Hu, "A fast pc algorithm for high dimensional causal discovery with multi-core pcs," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 5, pp. 1483–1495, 2019.

[9] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, prediction, and search*. MIT press, 2000.

[10] G. Park and G. Raskutti, "Learning large-scale poisson DAG models based on overdispersion scoring," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 631–639, Curran Associates, Inc., 2015.

[11] G. Park and H. Park, "Identifiability of generalized hypergeometric distribution (GHD) directed acyclic graphical models," in *Proceedings of Machine Learning Research* (K. Chaudhuri and M. Sugiyama, eds.), vol. 89 of *Proceedings of Machine Learning Research*, pp. 158–166, PMLR, 16–18 Apr 2019.

[12] P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf, "Nonlinear causal discovery with additive noise models," in *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 689–696, Curran Associates, Inc., 2009.

[13] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen, "A linear non-gaussian acyclic model for causal discovery," *J. Mach. Learn. Res.*, vol. 7, p. 2003–2030, Dec. 2006.

[14] M. D. Stasinopoulos, R. A. Rigby, G. Z. Heller, V. Voudouris, and F. D. Bastiani, "Flexible regression and smoothing," 2017.

[15] A. Zeileis, C. Kleiber, and S. Jackman, "Regression models for count data in R," *Journal of Statistical Software*, vol. 27, no. 8, 2008.

[16] T. J. Hastie and D. Pregibon, "Generalized linear models," *Statistical Models in S*, p. 195–247, Jan 2017.

[17] T. W. Yee, J. Stoklosa, and R. M. Huggins, "The VGAM package for capture-recapture data using the conditional likelihood," *Journal of Statistical Software*, vol. 65, no. 5, pp. 1–33, 2015.

[18] R. Raciborski, "Right-censored poisson regression model," *The Stata Journal*, vol. 11, no. 1, pp. 95–105, 2011.

[19] J. Peters, J. Mooij, D. Janzing, and B. Schoelkopf, "Identifiability of causal graphs using functional models," Feb 2012.

[20] R. P. Merkow, T. A. Schwartz, and A. B. Nathens, "Practical guide to comparative effectiveness research using observational data," *JAMA Surgery*, 2020.