Estimating Directed Bayesian Networks from Data

Chase Mathis, Ethan Ouellette

December 11, 2024

Conditional Independence

If X, Y are random variables. We say they are independent if

$$\mathrm{P}(X \leq x, Y \leq y) = \mathrm{P}(X \leq x) * \mathrm{P}(Y \leq y)$$

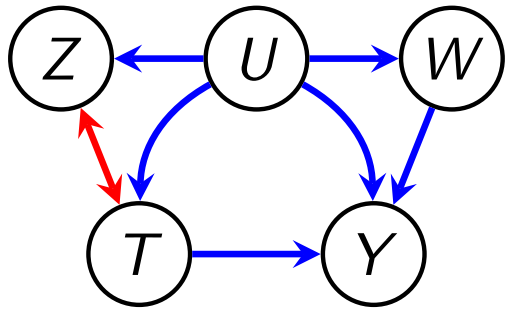
Same X, Y, Z random variables. We say that $X \perp\!\!\!\perp Y \mid Z$ if

$$\mathrm{P}(X \leq x, Y \leq y \mid Z = z) = \mathrm{P}(X \leq x \mid Z = z) * \mathrm{P}(Y \leq y \mid Z = z)$$

How can we draw a picture of our assumptions?

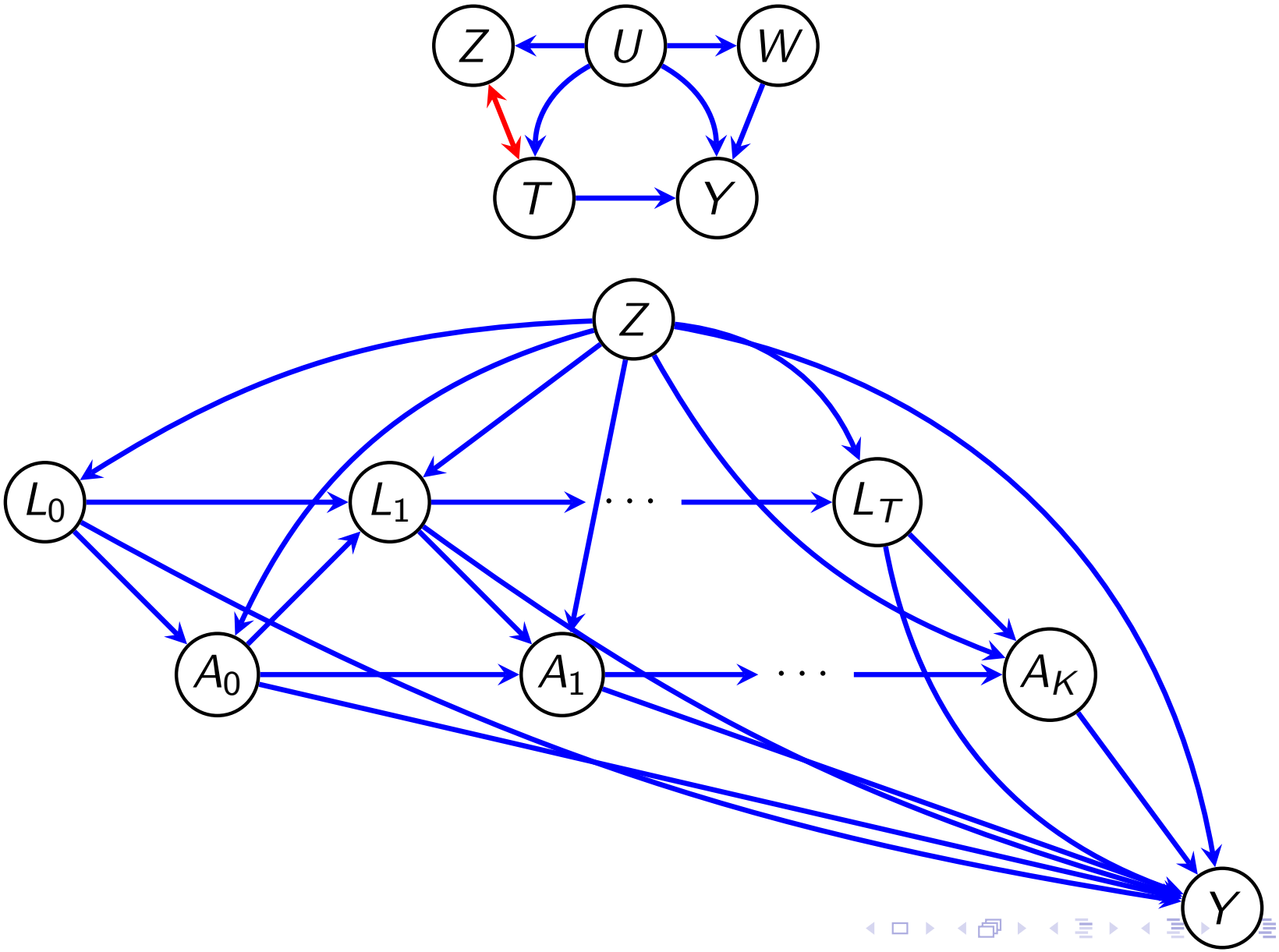
How can we draw a picture of our assumptions?

Graphs.



How can we draw a picture of our assumptions?

Graphs.



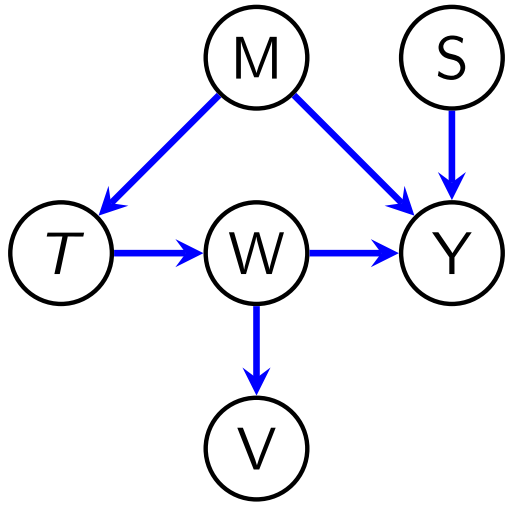


Figure: Example Graph

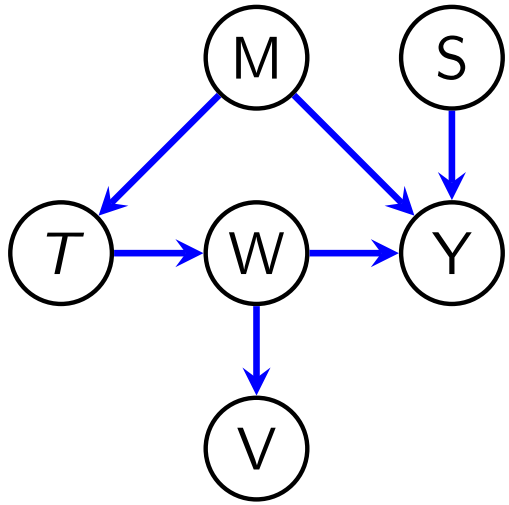


Figure: Example Graph

The best way to write

$$p(T, M, W, V, Y, S) = p(M) * p(S) \\ * p(T \mid M) * p(W \mid T) * p(V \mid W) \\ * p(Y \mid W, M, S)$$

How to Sample Data from Graph

Graph $\rightarrow$ Dataset

How to Sample Data from Graph

Graph $\rightarrow$ Dataset

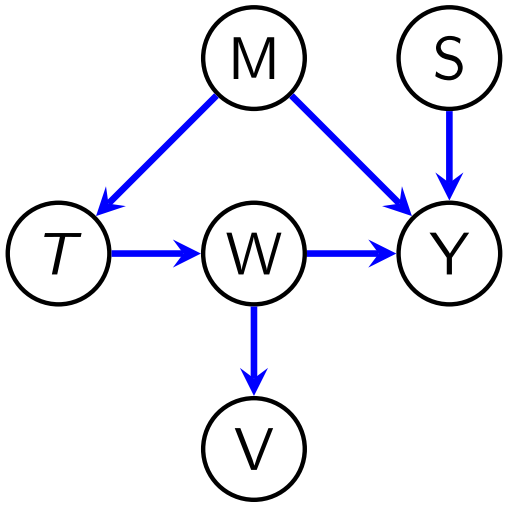


Figure: Example Graph

How to Sample Data from Graph

Graph $\rightarrow$ Dataset

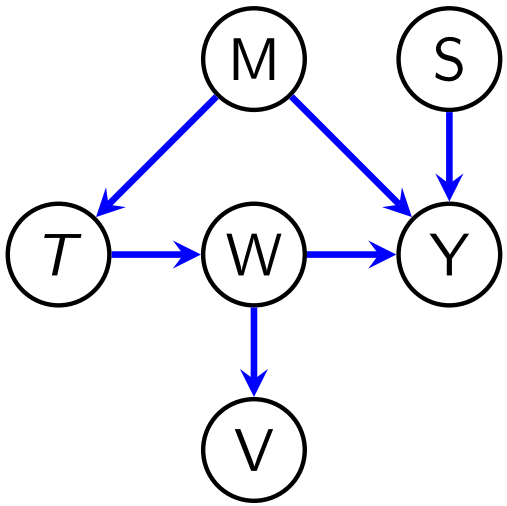
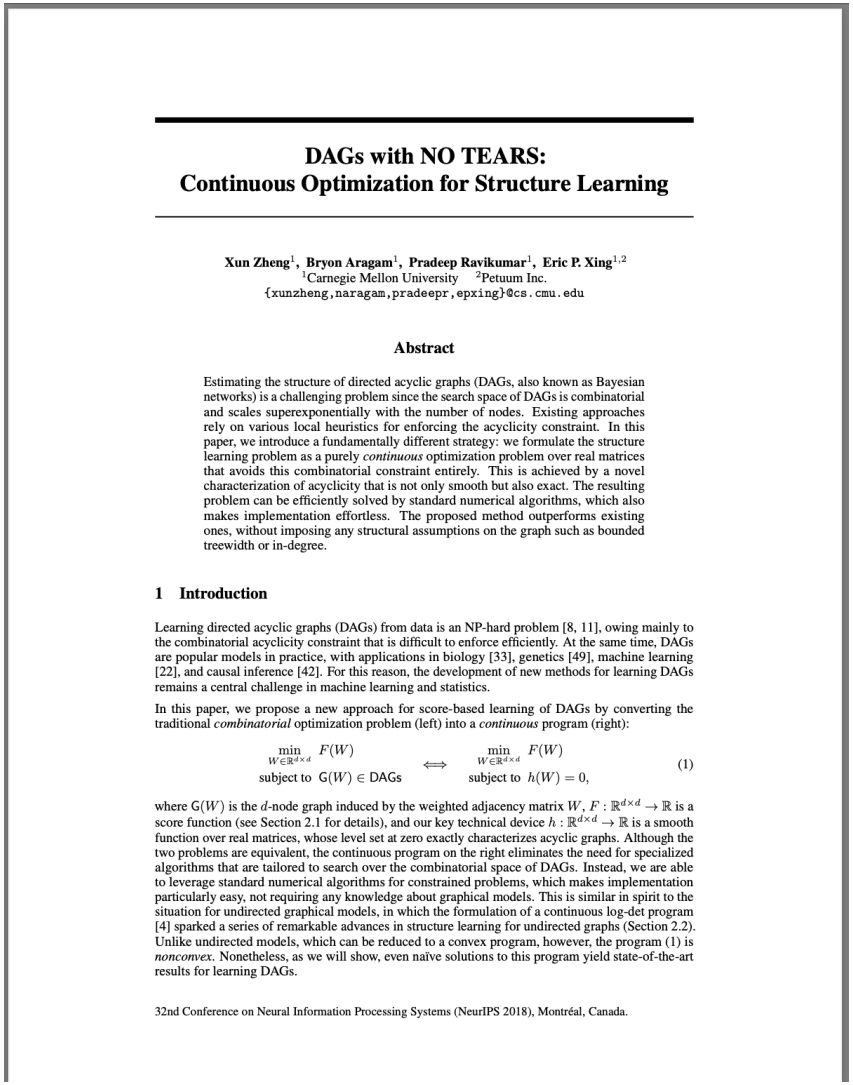


Figure: Example Graph

Algorithm 3 Generating a Dataset N rows

1: **for** $i = 1$ to N **do**
2: $M_i \sim P(M)$
3: $S_i \sim P(S)$
4: $T_i \sim P(T \mid M = M_i)$
5: $W_i \sim P(W \mid T = T_i)$
6: $V_i \sim P(V \mid W = W_i)$
7: $Y_i \sim P(Y \mid W = W_i, M = M_i, S = S_i)$
8: **end for**

How can we Generate a Graph From a Random Dataset?



DAGs with NO TEARS:
Continuous Optimization for Structure Learning

Xun Zheng[1], **Bryon Aragam**[1], **Pradeep Ravikumar**[1], **Eric P. Xing**[1,2]
[1]Carnegie Mellon University [2]Petuum Inc.
{xunzheng,naragam,pradeepr,epxing}@cs.cmu.edu

Abstract

Estimating the structure of directed acyclic graphs (DAGs, also known as Bayesian networks) is a challenging problem since the search space of DAGs is combinatorial and scales superexponentially with the number of nodes. Existing approaches rely on various local heuristics for enforcing the acyclicity constraint. In this paper, we introduce a fundamentally different strategy: we formulate the structure learning problem as a purely *continuous* optimization problem over real matrices that avoids this combinatorial constraint entirely. This is achieved by a novel characterization of acyclicity that is not only smooth but also exact. The resulting problem can be efficiently solved by standard numerical algorithms, which also makes implementation effortless. The proposed method outperforms existing ones, without imposing any structural assumptions on the graph such as bounded treewidth or in-degree.

1 Introduction

Learning directed acyclic graphs (DAGs) from data is an NP-hard problem [8, 11], owing mainly to the combinatorial acyclicity constraint that is difficult to enforce efficiently. At the same time, DAGs are popular models in practice, with applications in biology [33], genetics [49], machine learning [22], and causal inference [42]. For this reason, the development of new methods for learning DAGs remains a central challenge in machine learning and statistics.

In this paper, we propose a new approach for score-based learning of DAGs by converting the traditional *combinatorial* optimization problem (left) into a *continuous* program (right):

$$\min_{W \in \mathbb{R}^{d\times d}} F(W) \quad \text{subject to } \mathsf{G}(W) \in \mathsf{DAGs} \iff \min_{W \in \mathbb{R}^{d\times d}} F(W) \quad \text{subject to } h(W) = 0, \tag{1}$$

where $\mathsf{G}(W)$ is the d-node graph induced by the weighted adjacency matrix W, $F : \mathbb{R}^{d\times d} \to \mathbb{R}$ is a score function (see Section 2.1 for details), and our key technical device $h : \mathbb{R}^{d\times d} \to \mathbb{R}$ is a smooth function over real matrices, whose level set at zero exactly characterizes acyclic graphs. Although the two problems are equivalent, the continuous program on the right eliminates the need for specialized algorithms that are tailored to search over the combinatorial space of DAGs. Instead, we are able to leverage standard numerical algorithms for constrained problems, which makes implementation particularly easy, not requiring any knowledge about graphical models. This is similar in spirit to the situation for undirected graphical models, in which the formulation of a continuous log-det program [4] sparked a series of remarkable advances in structure learning for undirected graphs (Section 2.2). Unlike undirected models, which can be reduced to a convex program, however, the program (1) is *nonconvex*. Nonetheless, as we will show, even naïve solutions to this program yield state-of-the-art results for learning DAGs.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Figure: No Tears Neurips 2018 [3]

Major Theorem

Theorem 1. *A matrix $W \in \mathbb{R}^{d \times d}$ is a DAG if and only if*

$$h(W) = \operatorname{tr}\left(e^{W \circ W}\right) - d = 0, \tag{5}$$

where $\circ$ is the Hadamard product and e^A is the matrix exponential of A. Moreover, $h(W)$ has a simple gradient

$$\nabla h(W) = \left(e^{W \circ W}\right)^T \circ 2W, \tag{6}$$

and satisfies all of the desiderata (a)-(d).

Figure: Major Theorem

Guess which variables cause which?

X_1	X_2	X_3	X_4	X_5
-3.09	0.68	0.75	0.59	-0.84
1.97	0.02	-0.73	0.17	0.52
1.69	0.86	0.11	1.05	-0.55
-1.76	0.18	1.25	-1.25	-1.90
4.92	-0.42	-2.05	0.61	3.23
-3.35	1.25	1.53	0.77	-1.48
-6.43	1.25	3.63	-0.58	-3.66
2.14	-0.76	-1.50	1.20	1.99
6.79	0.59	-3.00	2.79	1.76
0.88	0.35	-0.83	-0.15	0.29

Table: How could you guess?

NoTEARS can Guess!

	X_1	X_2	X_3	X_4	X_5
X_1	0.00	0.00	0.00	0.00	0.00
X_2	0.00	0.00	0.00	0.33	-1.80
X_3	-1.82	0.00	0.00	0.00	0.00
X_4	0.00	0.00	0.00	0.00	1.02
X_5	0.00	0.00	-0.93	0.00	0.00

	X_1	X_2	X_3	X_4	X_5
X_1	0.00	0.00	0.00	0.00	0.00
X_2	0.00	0.00	0.00	0.53	-1.87
X_3	-1.94	0.00	0.00	0.00	0.00
X_4	0.00	0.00	0.00	0.00	1.12
X_5	0.00	0.00	-0.87	0.00	0.00

NoTEARS must be Perfect Then... Right?

$$X_1 \sim U(0,1); X_2 \sim U(0,1); Y = X_1 + X_2$$

A cool counter example where

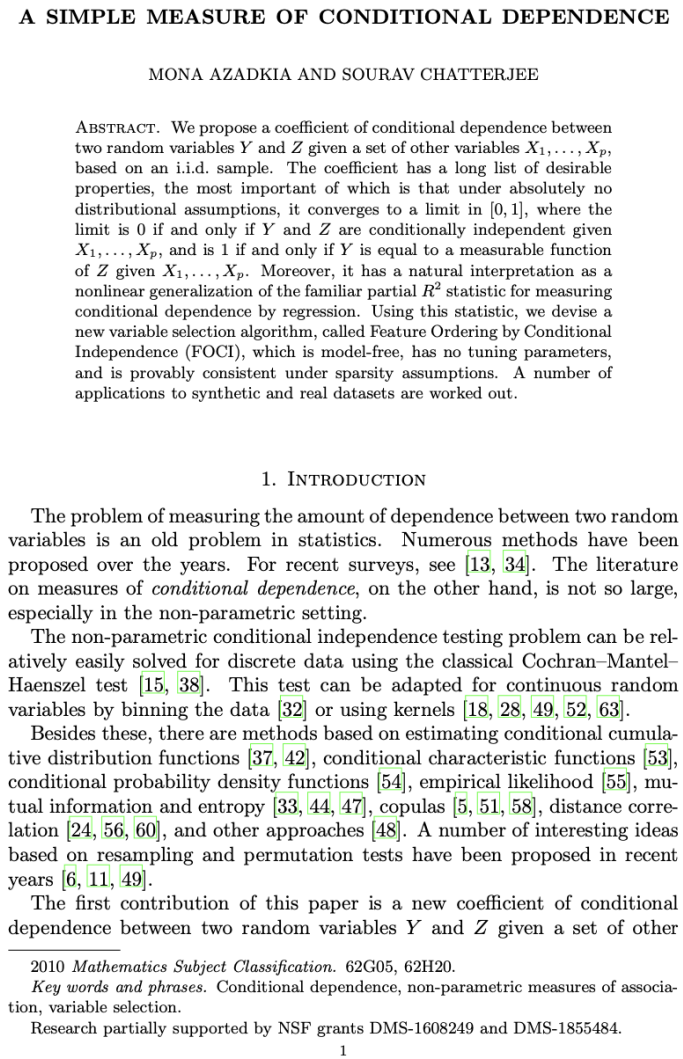
$$Y \perp\!\!\!\perp X_1, Y \perp\!\!\!\perp X_2, Y \not\perp\!\!\!\perp (X_1, X_2)$$

Let's put No Tears to the test.

Its Guess...

0.00	0.00	0.00
0.00	0.00	0.00
0.00	0.00	0.00

Can anyone get this right?



A SIMPLE MEASURE OF CONDITIONAL DEPENDENCE

MONA AZADKIA AND SOURAV CHATTERJEE

ABSTRACT. We propose a coefficient of conditional dependence between two random variables Y and Z given a set of other variables $X_1, \ldots, X_p$, based on an i.i.d. sample. The coefficient has a long list of desirable properties, the most important of which is that under absolutely no distributional assumptions, it converges to a limit in $[0, 1]$, where the limit is 0 if and only if Y and Z are conditionally independent given $X_1, \ldots, X_p$, and is 1 if and only if Y is equal to a measurable function of Z given $X_1, \ldots, X_p$. Moreover, it has a natural interpretation as a nonlinear generalization of the familiar partial R^2 statistic for measuring conditional dependence by regression. Using this statistic, we devise a new variable selection algorithm, called Feature Ordering by Conditional Independence (FOCI), which is model-free, has no tuning parameters, and is provably consistent under sparsity assumptions. A number of applications to synthetic and real datasets are worked out.

1. INTRODUCTION

The problem of measuring the amount of dependence between two random variables is an old problem in statistics. Numerous methods have been proposed over the years. For recent surveys, see [13, 34]. The literature on measures of *conditional dependence*, on the other hand, is not so large, especially in the non-parametric setting.

The non-parametric conditional independence testing problem can be relatively easily solved for discrete data using the classical Cochran–Mantel–Haenszel test [15, 38]. This test can be adapted for continuous random variables by binning the data [32] or using kernels [18, 28, 49, 52, 63].

Besides these, there are methods based on estimating conditional cumulative distribution functions [37, 42], conditional characteristic functions [53], conditional probability density functions [54], empirical likelihood [55], mutual information and entropy [33, 44, 47], copulas [5, 51, 58], distance correlation [24, 56, 60], and other approaches [48]. A number of interesting ideas based on resampling and permutation tests have been proposed in recent years [6, 11, 49].

The first contribution of this paper is a new coefficient of conditional dependence between two random variables Y and Z given a set of other

2010 *Mathematics Subject Classification.* 62G05, 62H20.

Key words and phrases. Conditional dependence, non-parametric measures of association, variable selection.

Research partially supported by NSF grants DMS-1608249 and DMS-1855484.

Figure: Azadkia and Chatterjee [1]

The Chatterjee correlation coefficient

$$\xi_n(X, Y) := 1 - \frac{3 \sum_{i=1}^{n-1} |r_{i+1} - r_i|}{n^2 - 1}$$

and can be extended to

$$\xi_n(X, Y \mid Z)$$

- $\xi_n \in [0, 1]$
- $X \perp\!\!\!\perp Y \mid Z, \xi_n \to 0$ with probability one.
- $X = f(Y \mid Z), \xi_n \to 1$ with probability one.

Our Counter example

```
> x1 <- runif(n)
> x2 <- runif(n)
> y <- (x1 + x2) %% 1
> bnlearn::ci.test(x = x1, y = y, z = x2)

        Pearson's Correlation

data:  x1 ~ y | x2
cor = 0.00049908, df = 99997, p-value = 0.8746
alternative hypothesis: true value is not equal to 0
```

Our Counter example

```
> x1 <- runif(n)
> x2 <- runif(n)
> y <- (x1 + x2) %% 1
> bnlearn::ci.test(x = x1, y = y, z = x2)

        Pearson's Correlation

data:  x1 ~ y | x2
cor = 0.00049908, df = 99997, p-value = 0.8746
alternative hypothesis: true value is not equal to 0
```

Chatterjee

```
> FOCI::codec(y, x1, x2)
[1] 0.9925471
```

Spectral Theory?

Spectral Bayesian Network Theory

Luke Duttweiler, Sally W. Thurston, Anthony Almudevar

October 17, 2022

Abstract

A Bayesian Network (BN) is a probabilistic model that represents a set of variables using a directed acyclic graph (DAG). Current algorithms for learning BN structures from data focus on estimating the edges of a specific DAG, and often lead to many 'likely' network structures. In this paper, we lay the groundwork for an approach that focuses on learning global properties of the DAG rather than exact edges. This is done by defining the *structural hypergraph* of a BN, which is shown to be related to the inverse-covariance matrix of the network. Spectral bounds are derived for the normalized inverse-covariance matrix, which are shown to be closely related to the maximum indegree of the associated BN.

Keywords: Weighted hypergraph, Bayesian Network, Hypergraph Laplacian, Eigenvalue bound, Directed acyclic graph, Linear structural equation model

MSC: 05C50, 62H22

Figure: [2]

Hypergraph Matrices

- **Magnitude Matrix**: $M(G) = \text{diag}(m_1, \ldots, m_n)$
- **Incidence Matrix**: $H(G) \in \mathbb{R}^{n \times m}$, where $H(G)_{ij} = \omega(v_i, e_j)$
- **Adjacency Matrix**:

$$A(G)_{ij} = \sum_{e \in E} \zeta_e(v_i, v_j)$$

- **Normalized Adjacency Matrix**:

$$C(G) = M(G)^{-1/2} A(G) M(G)^{-1/2}$$

- **Kirchhoff Laplacian**:

$$K(G) = M(G) - A(G) = H(G)H(G)^T$$

- **Normalized Laplacian**:

$$L(G) = I_n - C(G)$$

From the DAG to a Hypergraph

The **Structural Hypergraph** of a Bayesian Network:

- $V_{ST} = V_G$
- $E_{ST} = e_1, \ldots, e_n$ with $e_k = \{v_k\} \cup pa_G(v_k)$
- $\mathcal{I}_{ST} \subseteq (v_i, e_k) \in \mathcal{I} \iff v_i \in e_k$
- $\omega_{ST}(v_i, e_k) = \begin{cases} -\frac{\beta_{ik}}{\sigma_k} & \text{if } i \neq k \\ \frac{1}{\sigma_k} & \text{if } i = k \end{cases}$

First Theorem: Relating BN to Structural Hypergraph

Theorem 1

Let X be a linear Bayesian Network following a DAG G with weighted adjacency matrix A, and let $\mathcal{G}_{ST}$ be the structural hypergraph of X. Then, if Σ is the covariance matrix of X,

$$\Sigma^{-1} = K(\mathcal{G}_{ST}).$$

Additionally, if the normalized inverse covariance matrix of X is Ω, then

$$\Omega = L(\mathcal{G}_{ST}).$$

- Implications?

Quick Interlude- A Moral Graph

One way to convert a Directed Graph to an Undirected graph is to *Moralize it*. How to get the moral graph G_m?

Quick Interlude- A Moral Graph

One way to convert a Directed Graph to an Undirected graph is to *Moralize it*. How to get the moral graph G_m?

- If $X \to Y$, $Z \to Y$ make $X \to Z$.
- Drop arrows.

It is called moral because if two parents share a child node, they should be married (connected).
Why is it important?

Quick Interlude- A Moral Graph

One way to convert a Directed Graph to an Undirected graph is to *Moralize it*. How to get the moral graph G_m?

- If $X \to Y$, $Z \to Y$ make $X \to Z$.
- Drop arrows.

It is called moral because if two parents share a child node, they should be married (connected).
Why is it important? The conditional independencies in a moral graph are the same as in the original graph.

Second Theorem: Ruling out Trees

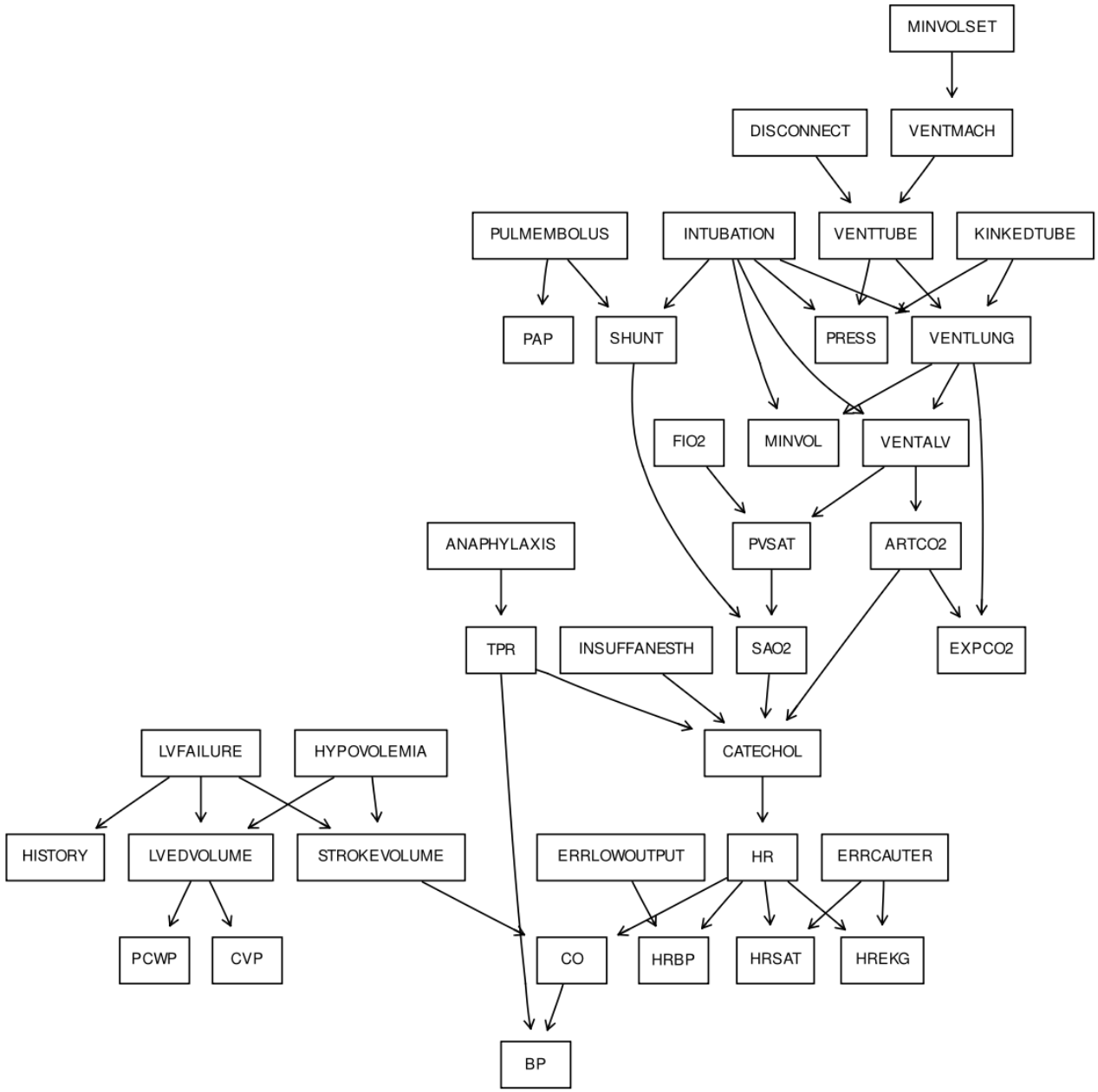
Theorem 2

Let X be a linear Bayesian Network with moral graph G_M. Then, if G_M is a tree (or a subgraph of a tree), and Ω is the normalized precision matrix of X, we must have

$$\lambda_1(\Omega) \leq 2.$$

- Allows us to "rule out" tree possibilities.

Alarm Dataset Experiment



Theorem 2 Trial

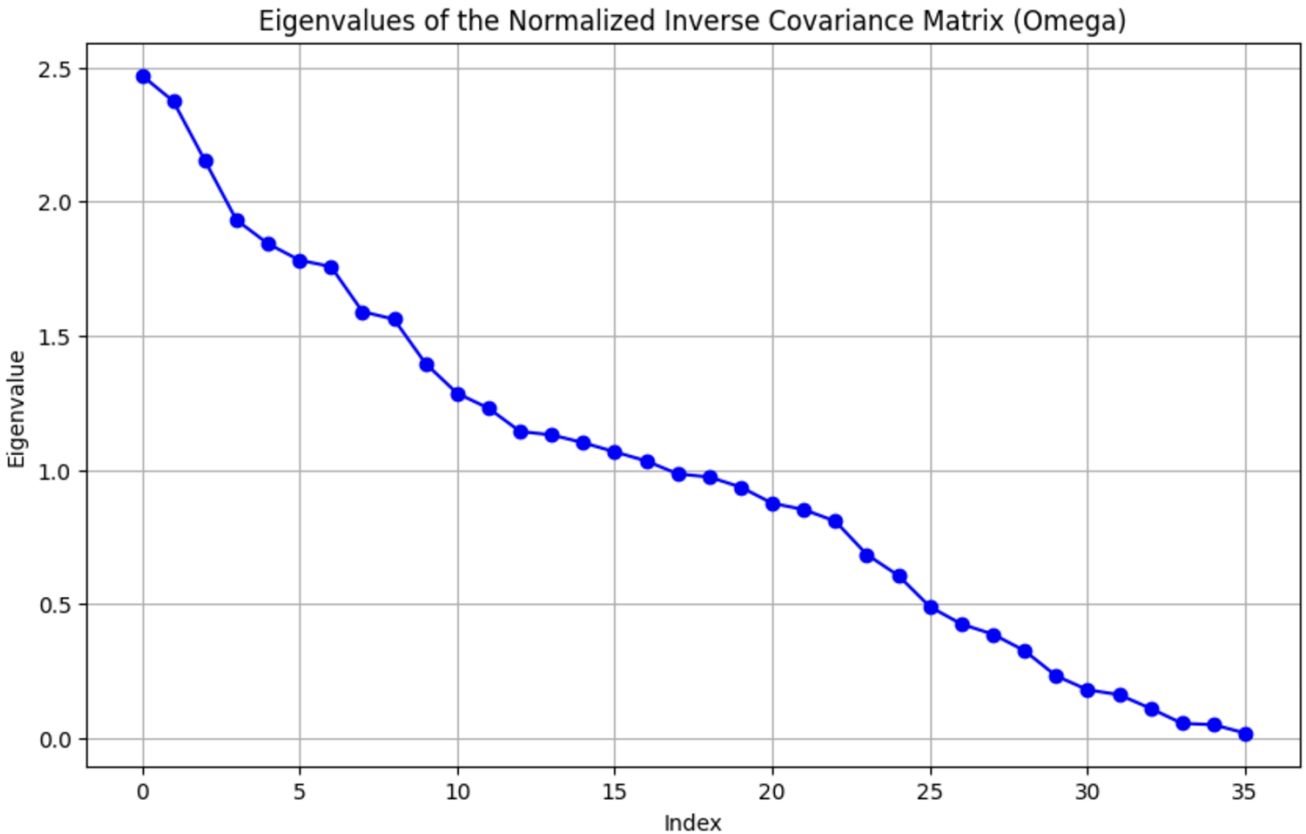


Figure: Eigenvalues of sampled Ω, `n_samples=1000`

Third Theorem: Confirming a tree structure

Theorem 3

Let X be a linear Bayesian Network following a DAG G with moral graph G_M and structural hypergraph $\mathcal{G}_{ST}$. Let Ω be the normalized inverse covariance matrix of X. Then the following statements are true:

(a) If G_M is a tree, the eigenvalues of Ω are additively symmetric about 1.

(b) Under Assumptions 1 and 2, if the eigenvalues of Ω are additively symmetric about 1, then G_M is a tree.

Theorem 3 Trial

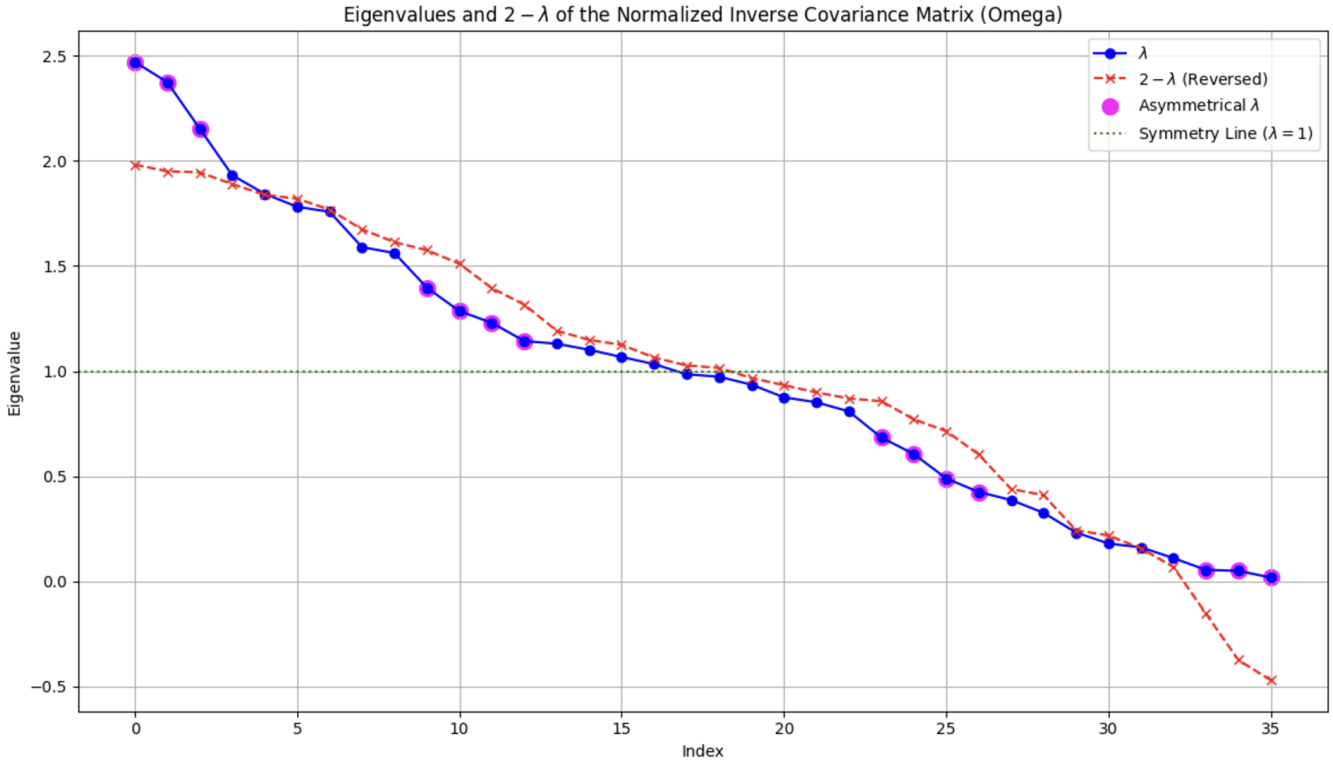


Figure: Sorted eigenvalues and the reflection of the reversed list about 1

Theorem 3 Trial

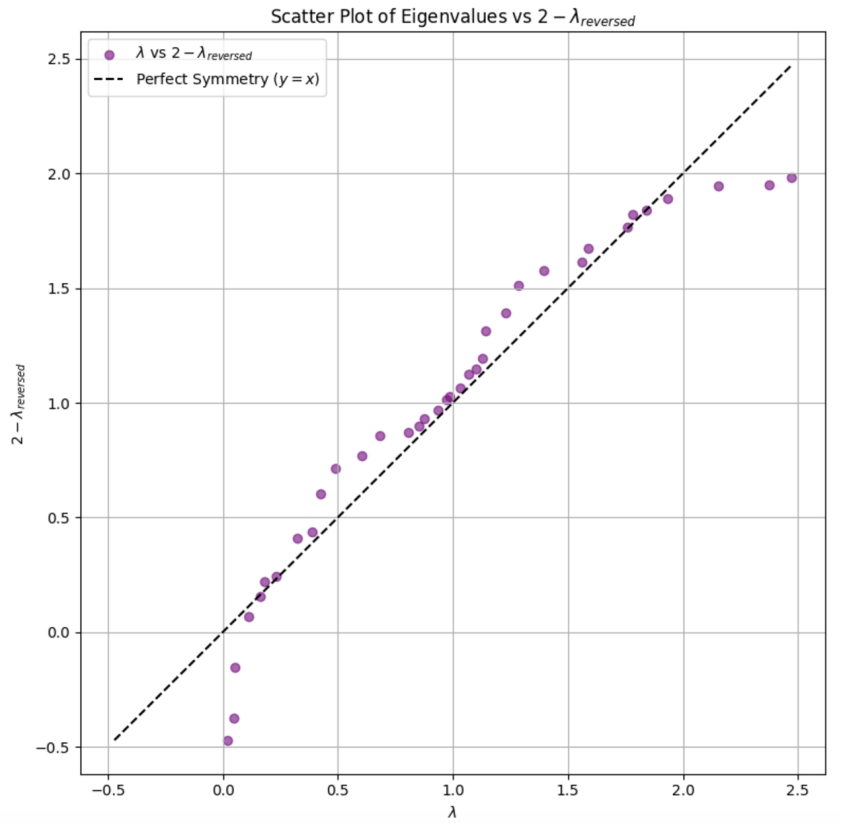


Figure: λ vs. $(2-\lambda_{reversed})$

Begs the question...

How can we be certain?

Begs the question...

How can we be certain?

- Requires statistical tests for eigenspaces (Silin and Fan, 2020).
- Searching for other structural properties that can be gleaned from sampling. A 2022 survey paper on structure learning is a good place to start (Kitson, et al., 2022).
- The appeal of this method is that there are n eigenvalues of this matrix, while there are $n(n-1)/2$ lower triangular elements of A.

Contributions

- Exposed the problem of estimating directed acyclic graphs from sampled data.
- Illustrated a counterexample where NoTEARS fails and the Azadkia correlation coefficient succeeds.
- Discussed necessary and sufficient conditions for DAGs using Spectral Theory
- Applied this technique to a real world dataset.
- Combine all three techniques to make a robust unified Bayesian Network estimator.

References


Mona Azadkia and Sourav Chatterjee.
A simple measure of conditional dependence.
arXiv preprint arXiv:1910.12327, v6, March 2019.
Final version. To appear in Ann. Statist.


Luke Duttweiler, Sally W. Thurston, and Anthony Almudevar.
Spectral bayesian network theory.
arXiv preprint arXiv:2210.07962v1, October 17 2022.


Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing.
Dags with no tears: Continuous optimization for structure learning.
arXiv preprint arXiv:1803.01422, v2, November 2018.
Accepted to NIPS 2018.