An Investigation into the Markov Equivalence Class Problem in Bayesian Networks

The BTech Project of

Samkit Shah (B20CS059) and Yash Maniya (B20CS033)

Mentor: Dr Saptarshi Pyne

Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur, Rajasthan, India 342030

December 6, 2023

Objective

When predicting the underlying model of a given dataset, the 'Markov equivalence class problem in Bayesian networks' can lead to extremely **misleading models**.

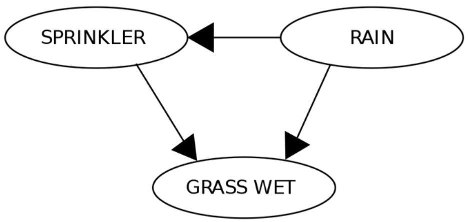
In this project, we aim to **review the literature** and **demonstrate the problem with a benchmark dataset**. This project will be our **first step** to **overcoming the problem**.

What is a Bayesian network?

A Bayesian network is a directed acyclic graph (DAG) where

- Vertices = Random variables = Components of a system
- Edge A->B represents that A may not be conditionally independent of B; in other words, A may have an influence on B

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



RAIN T	RAIN F
0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

In this figure, we see a Bayesian network with three variables and the conditional probability table of each variable.

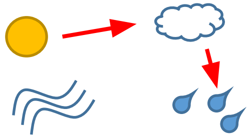
https://commons.wikimedia.org/wiki/File:SimpleBayesNet.svg

Application of Bayesian networks

The interdependencies between the components of a real-world system can be modelled as a Bayesian network.

Meteorology

Variables = Temperature, humidity, rain, wind velocity, etc.



Econometrics

Variables = Stock prices

Smart devices

Variables = Heart rate, blood sugar, etc.

Neuroscience

Variables = Brain regions

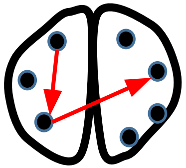
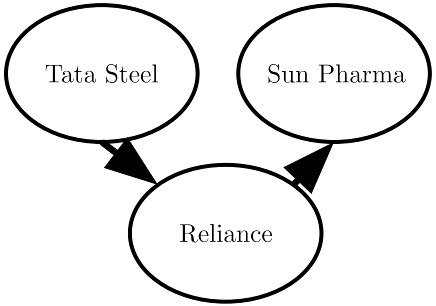


Image understanding (variables = pixels), speech recognition (variables = phonemes), natural language understanding (variables = tokens), e-commerce recommender systems (variables = products), computational genomics (variables = genes), etc.

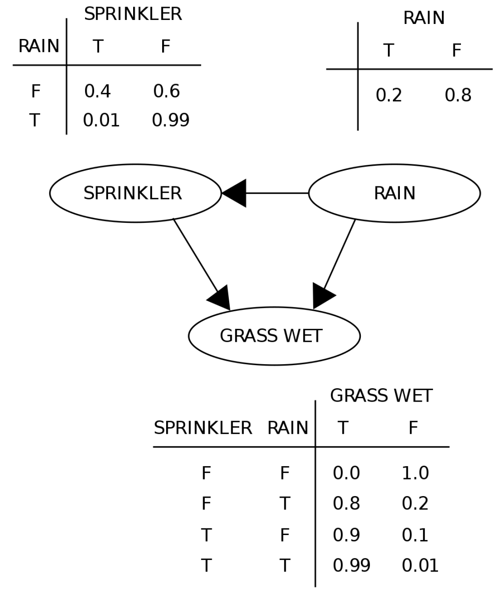


A large Bayesian network

Learning the 'structure' of a Bayesian network from data

There exists a class of algorithms for learning the edgeset/topology/structure of a Bayesian network given a cross-sectional or longitudinal measurements of the variables.

	Sample 1	Sample 2	Sample 3	
SPRINKLER	0	1	0	…
RAIN	0	0	1	…
GRASS WET	0	1	1	…

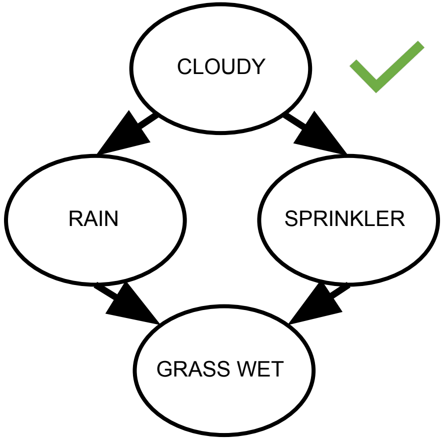


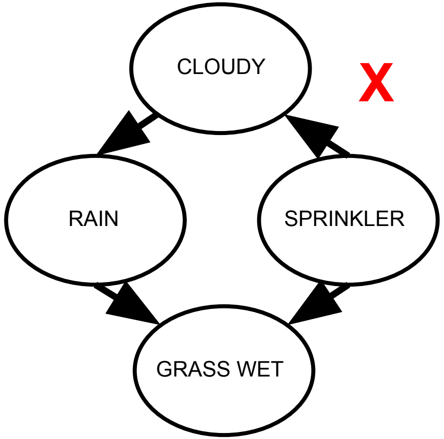
The Markov equivalence class problem in Bayesian networks

- The Bayesian network ‘structure learning’ algorithms usually find a DAG with the **maximal likelihood score** given a dataset.

- If two DAGs belong to the same Markov equivalence class, then they are **statistically indistinguishable**. Therefore, they will have the **same likelihood score**.

- If there is a tie between Markov equivalent DAGs, the existing algorithms break the ties by randomly selecting one of them. The result could be extremely **misleading**.

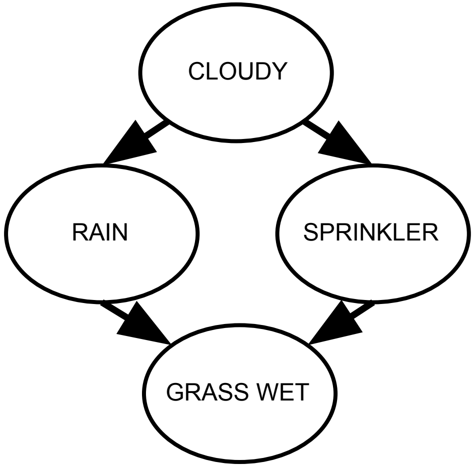


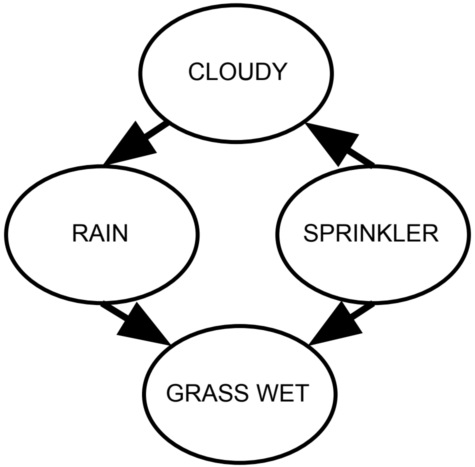


Key technical insight: Which DAGs are Markov equivalent?

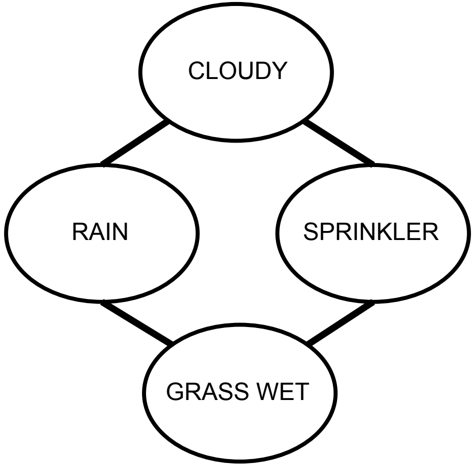
Andersson et al. proved that two DAGs are Markov equivalent if and only if

- they have the same underlying undirected structure (known as the ‘skeleton’) and
- the same ‘immorality’ sub-structures





The skeleton

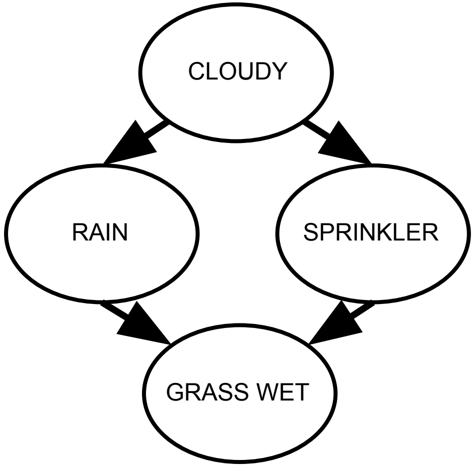


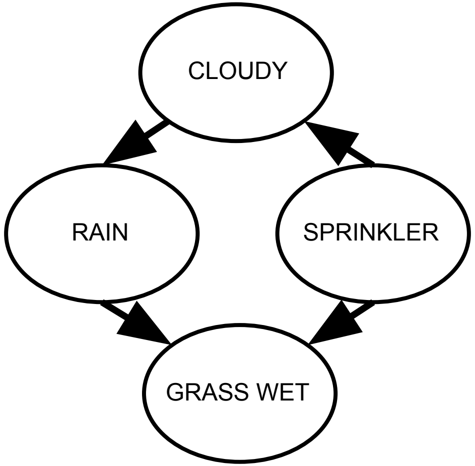
Theorem 2.1, Andersson et al., The Annals of Statistics, 1997

Key technical insight: Which DAGs are Markov equivalent?

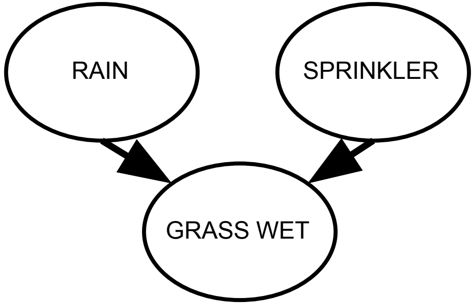
Andersson et al. proved that two DAGs are Markov equivalent if and only if

- they have the same underlying undirected structure (known as the ‘skeleton’) and
- the same ‘immorality’ sub-structures





An **immorality** is a collider
X -> Y <- Z where
X and Z are non-adjacent.



Theorem 2.1, Andersson et al., The Annals of Statistics, 1997

Research gap

The literature explains the theory of the Markov equivalence problem with **toy examples only**. To our knowledge, **there had not existed a study that demonstrated the existence of the problem in practice**.

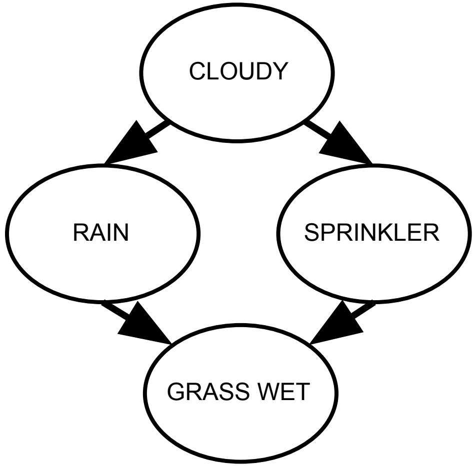
Hence, we focused on demonstrating the existence of the problem in a widely used dataset.

Experimental setup for reproducing the Markov equivalence problem

Dataset selection

We selected **the 'sprinkler' dataset**, widely used for benchmarking Bayesian network structure learning algorithms.

The ground truth network that has generated the dataset is shown on the right.



Experimental setup for reproducing the Markov equivalence problem (contd.)

Algorithm selection

1. **Exhaustive search**:
- Enumerate **all possible DAGs** on the four vertices.

- For each DAG, compute its **likelihood** score.
  We used the Bayesian information criterion (**BIC**) scoring function for calculating the likelihood score [1].
  BIC scores range from **-Inf to zero**, zero being the highest.

- Select the DAG(s) with the **global maximal likelihood** score.

[1] Schwarz, Annals of Statistics, 1978.

Experimental setup for reproducing the Markov equivalence problem (contd.)

Algorithm selection

2. **Heuristic-based search**:
- Select the DAG(s) with the **local maximal likelihood** score using the **hill climb search** algorithm.

Experimental setup for reproducing the Markov equivalence problem (contd.)

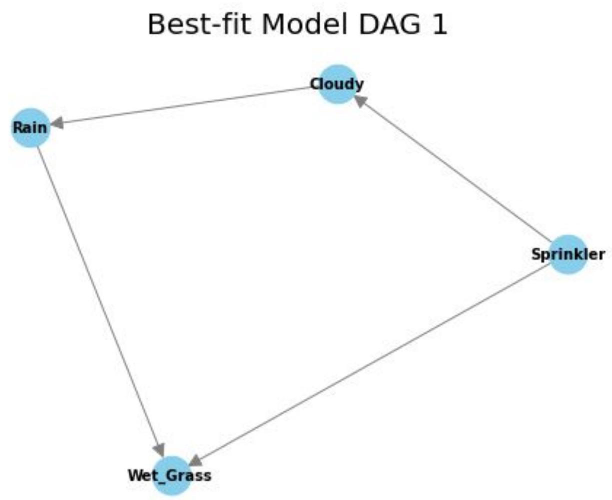
Implementation

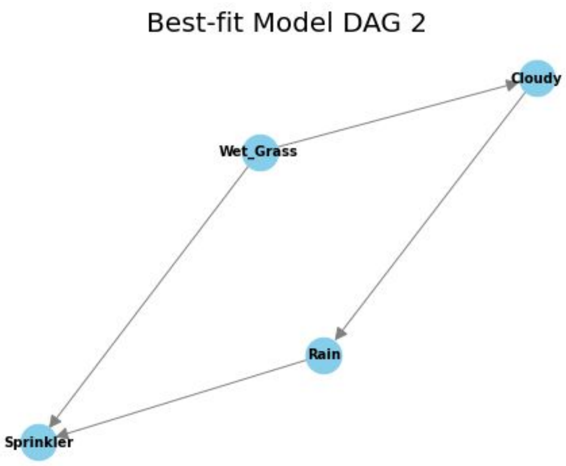
Python packages:
- pgmpy
- bnlearn

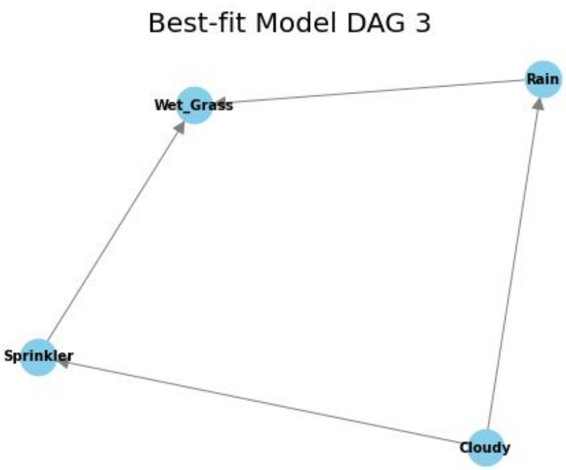
Evaluation metric

- F1-score: We calculated the F1-score(s) of the predicted DAG(s) w.r.t. the true DAG

Results: DAGs with the highest BIC score







BIC = -19379.58

F1 score = 0.875

BIC = -19379.58

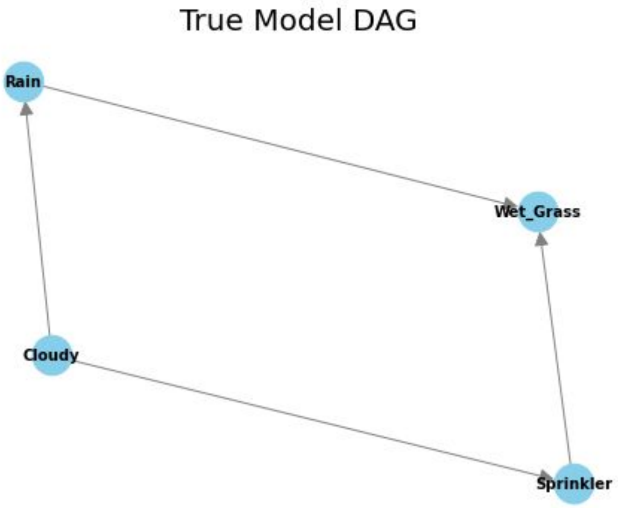
F1 score = 0.625

BIC = -19379.58

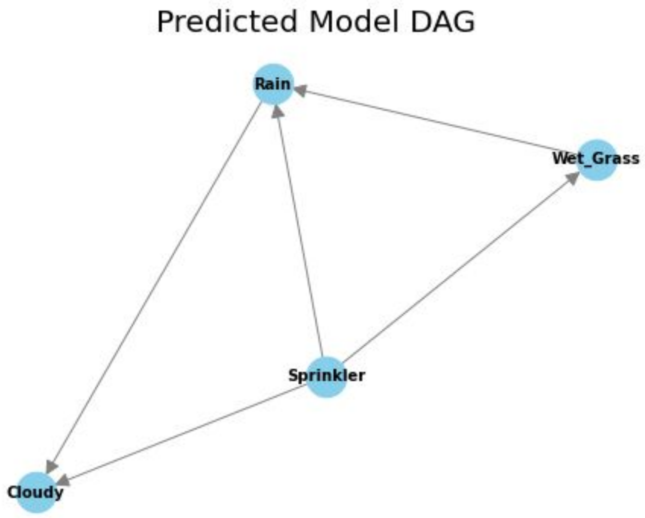
F1 score = 1

Method used : Exhaustive Search

Results: DAG with the highest BIC score (contd.)



target	Cloudy	Rain	Sprinkler	Wet_Grass
source				
Cloudy	False	True	True	False
Rain	False	False	False	True
Sprinkler	False	False	False	True
Wet_Grass	False	False	False	False



F1 Score of the predicted model : 0.5625

Method used : Hill Climb Search

Summary and Future Work

- To our knowledge, we demonstrated the Markov equivalence class problem in Bayesian networks with a benchmark dataset for the first time.

- In future, we would like to devise a heuristic-based strategy to break the ties between Markov equivalent Bayesian networks.

Thank You

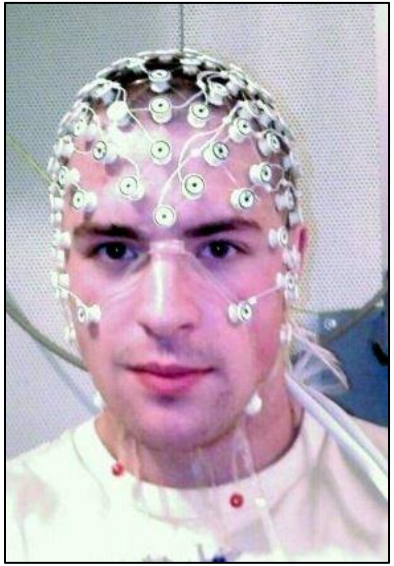
Appendix

Why do we need to **decipher the cause-effect relationships** among variables in complex real-world systems?

A Case Study in Space Research

Electroencephalography (EEG) Data

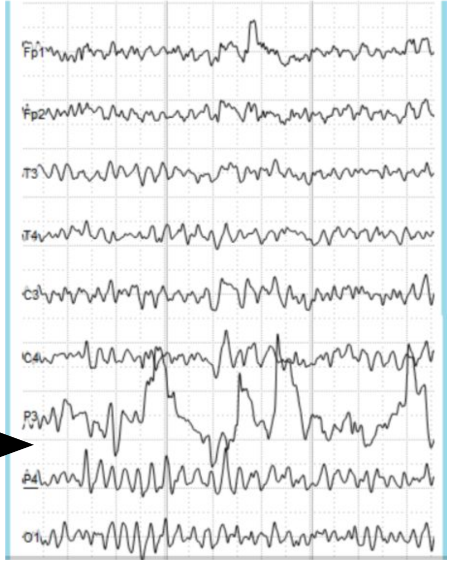
EEG cap



Region 1 (v_1)
Region 2 (v_2)
Region 3 (v_3)
•
•
•

Time

μV

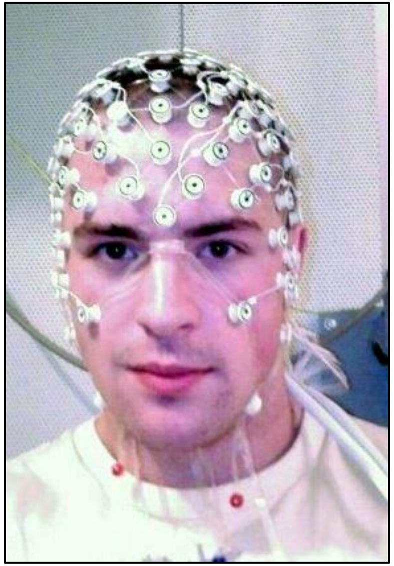


Time-series EEG data

	t_1	t_2	t_3	•	•
v_1	1.5	3.5	4.5		
v_2	2.3	3.6	2.3		
v_3	1.2	1.2	1.2		
•					
•					
•					

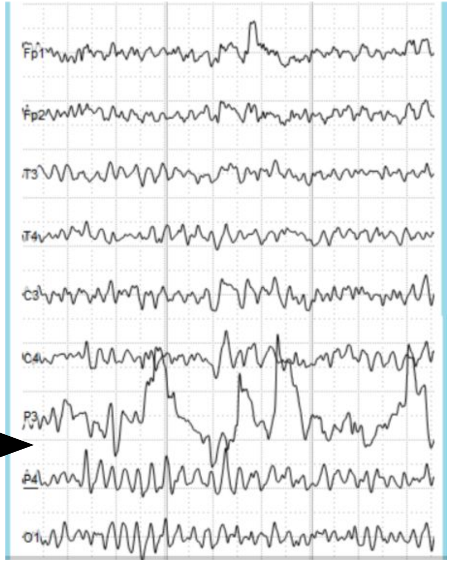
Electroencephalography (EEG) Data

EEG cap



Region 1 (v_1)
Region 2 (v_2)
Region 3 (v_3)
•
•
•

Time



μV

Time-series discretized EEG data

	t_1	t_2	t_3	•	•
v_1	0	1	1		
v_2	0	1	0		
v_3	0	0	0		
•					
•					
•					

Problem Statement

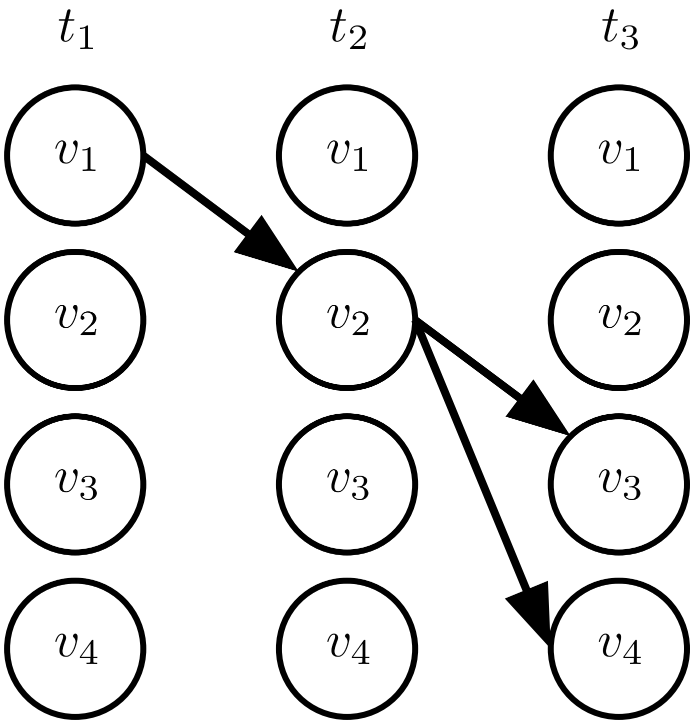
Input: Time-series brain activity data

	t_1	t_2	t_3
v_1	1	0	0
v_2	0	1	0
v_3	0	0	1
v_4	0	0	1

Assuming that
set of variables $\mathcal{V} = \{v_1, v_2, v_3, v_4\}$,
set of time points $\mathcal{T} = \{t_1, t_2, t_3\}$.

Reverse engineering

[redacted]



First-order Markovian process

Problem Statement

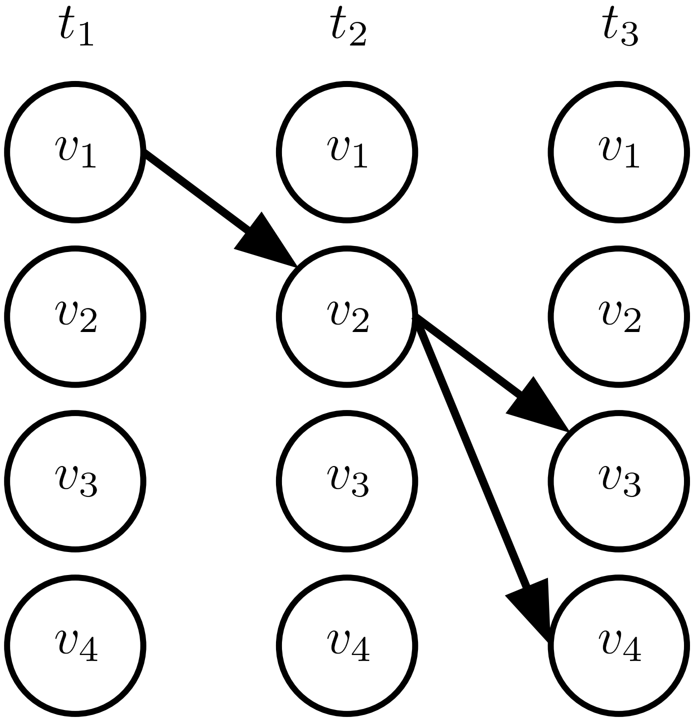
Input: Time-series brain activity data

	t_1	t_2	t_3
v_1	1	0	0
v_2	0	1	0
v_3	0	0	1
v_4	0	0	1

Multiple instances

Reverse engineering

[redacted]



Problem Statement

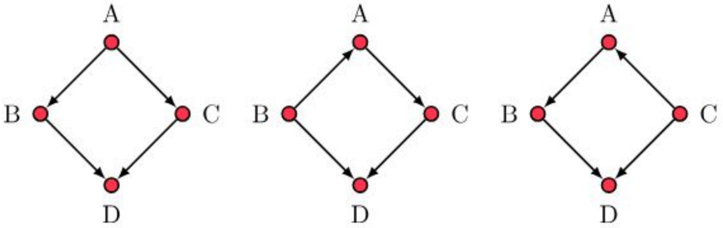


FIGURE 1: Equivalent digraphs

$$\#\mathrm{DAG}(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} \#\mathrm{DAG}(n-i)\,.$$

> Upper bound

$$\#\mathrm{EQ1}(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} (2^{n-i} - (n-i))^{i} \#\mathrm{EQ1}(n-i)\,.$$

> Lower bound

Markov Graphs

A **Markov network** is a pair (G, P), where G is an undirected graph over variables $\mathcal{V}$ and $P(\mathcal{V})$ is a joint distribution for $\mathcal{V}$ such that

$$\mathcal{X} \perp_u \mathcal{Y} \mid \mathcal{Z} \text{ only if } \mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z}\,, \qquad \text{[global Markov property]}$$

for any subsets $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \mathcal{V}$.

Scoring Functions

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

$$\mathrm{BIC} = k\ln(n) - 2\ln(\widehat{L})$$

k = No. of Parameters
L = Likelihood score
n = Total no. of vertices?

Citations

Search Algorithms

- Constraint Based Algorithms
  - Ex - Peter and Clark Algorithm
- Score based Algorithms
  - Ex – Hill Climb Search
- Hybrid Algorithms
  - Ant Colony Optimization