On Sampling and Counting Directed Acyclic Graphs

Juha Harviainen

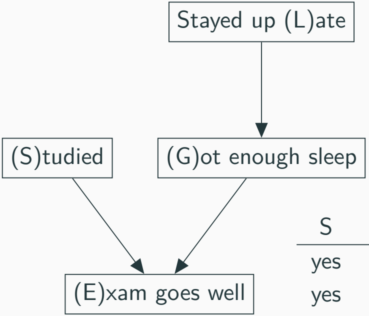
Overview

- Bayesian Networks
- Exact
- MCMC
- Constrained
- Concluding Remarks

Bayesian Networks

Bayesian Networks

- Graphical representation of multivariate probability distribution
- Structure given as a DAG
- Univariate probability tables conditionally to parents
- $\Pr(L, S, G, E) = \Pr(L)\Pr(S)\Pr(G \mid L)\Pr(E \mid S, G)$



L = yes
0.5

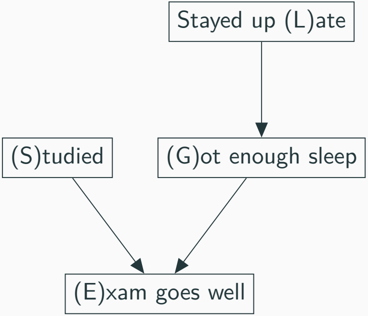
S = yes
0.9

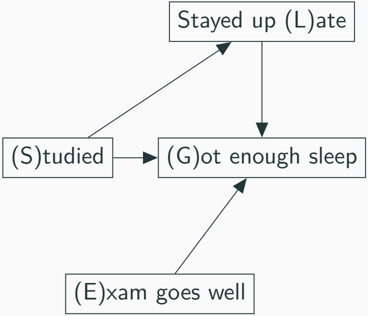
L	G = yes
yes	0.1
no	0.8

S	G	E = yes
yes	yes	0.9
yes	no	0.7
no	yes	0.4
no	no	0.1

Network Structures

- Some structures represent reality better than the others
- How to quantify the quality of a structure?
- Edges represent direct conditional dependencies, not causality





Posterior

- For data D, we obtain posterior of DAGs G as

$$f(G) := p(G \mid D) = \frac{p(G)p(D \mid G)}{p(D)} \propto p(G)p(D \mid G)$$

- Prior distribution $p(G)$
- Likelihood $p(D \mid G)$

Score-based Structure Learning

- Under certain assumptions, likelihood reduces to a product
- Each parent set G_i of i has local score $f_i(G_i)$
- Posterior is proportional to the product of local scores and prior:

$$f(G) \propto p(G) \cdot \prod_{i \in N} f_i(G_i)$$

- Modular prior Also reduces to a product
- Order-modular prior Larger by a factor of No. linear extensions of G

Sampling and Counting?

- Using only a highest-scoring DAG ignores uncertainty

DAG Counting

Objective: Compute $\sum_G f(G)$

DAG Sampling

Objective: Sample G with $\Pr(G) \propto f(G)$

- Model averaging, prevalence of features, ...
- Optimization is NP-hard[1], counting #P-hard[2]

1 David M. Chickering. Learning Bayesian networks is NP-complete. *AISTATS'95*.

2 Juha Harviainen and Mikko Koivisto. Revisiting Bayesian Network Learning with Small Vertex Cover. *UAI'23*.

Approaches

Exact

- Accurate
- Slow

MCMC

- Practical
- Convergence?

Constrained

- Tractable?
- Expressiveness?

Exact

Exact Counting and Sampling

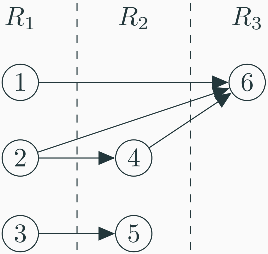
Issue Super-exponentially many DAGs

Solution Manipulate groups of DAGs simultaneously

Approach Dynamic programming, transforms, inclusion—exclusion...

Modular Counting and Sampling

- With order-modular prior, modifying optimization algorithm suffices[3]
- For modular priors, partitions of nodes are utilized[4]
- A root-layering is obtained by repeatedly removing source nodes



[3] Ru He, Jin Tian, and Huaiqing Wu. Structure Learning in Bayesian Networks of a Moderate Size by Efficient Sampling. JMLR. 2016.

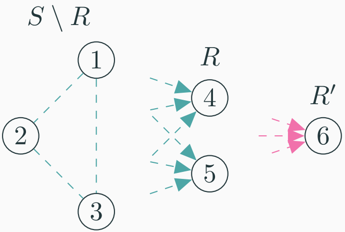
[4] Jack Kuipers and Giusi Moffa. Uniform random generation of large acyclic digraphs. Stat. Comput. 2015.

Dynamic Programming

- Weighted counting with dynamic programming over layers[5]
- Let $\mathrm{count}(S, R)$ be the total score of DAGs on S whose last layer is R

$$\mathrm{count}(S \cup R', R') = \sum_{R \subseteq S} \mathrm{count}(S, R) \cdot w(S, R, R')$$

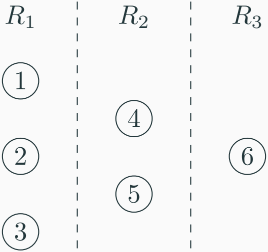
- Requires $O(3^n n)$ time to compute ($R \subseteq S \subseteq N$)



[5] Topi Talvitie, Aleksis Vuoksenmaa, and Mikko Koivisto. Exact Sampling of Directed Acyclic Graphs from Modular Distributions. *UAI'19*.

Sampling

- Stochastic backtracking over the dynamic programming table
- Construct the DAG layer by layer
- Sample parents independently of other nodes



Faster Counting

- Rectangular matrix multiplication yields running time $O(2.985^n)$[6]
- Tighter analysis might improve the result to $O(2.930^n)$
- If $\omega(2) = 3$, then improvable to $O(2^{3n/2}) = O(2.829^n)$
  - Multiplying matrices of shapes $N \times N^2$ and $N^2 \times N$ in $O(N^{3+\epsilon})$?

[6] Mikko Koivisto and Antti Röyskö. Fast Multi-Subset Transform and Weighted Sums over Acyclic Digraphs. SWAT'20.

Faster Sampling

- Complexity $O(2.829^n)$ achieved with rejection sampling[7]
- Allow some duplicate counting, reject some samples to fix distribution
- Infeasible for larger networks, e.g., $n \geq 25$
- Under SETH, an $O((2-\epsilon)^n)$-time algorithm seems unlikely

[7] Juha Harviainen and Mikko Koivisto. Faster Perfect Sampling of Bayesian Network Structures. UAI'24.

MCMC

Markov chain Monte Carlo

- Generates a sequence of DAGs $G_1, G_2, \ldots$
- Distribution of the next DAG depends only on the current state

$$p(G_{t+1} \mid G_1, G_2, \ldots, G_t) = p(G_{t+1} \mid G_t)$$

Metropolis–Hastings As t increases, makes $p(G_t)$ approach posterior

- Propose a DAG G_{t+1} given G_t
- Keep G_t with certain probability, otherwise replace it by G_{t+1}

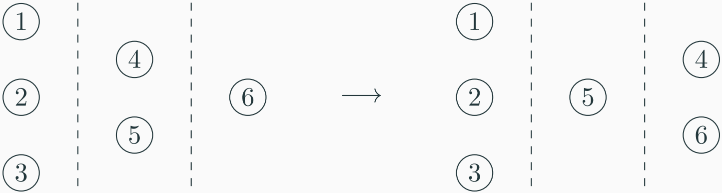
Basic Moves

- Three “basic” moves[8]
  - Add an edge
  - Remove an edge
  - Reverse an edge
- Every DAG can be reached
- Rather slow convergence

8 David Madigan and Jeremy York. Bayesian graphical models for discrete data. Int. Stat. Rev. 1995.

Directions

- New moves
  - "New edge reversal move"[9]
- Different state space
  - Orderings of nodes[10]
  - Root-layerings[11]



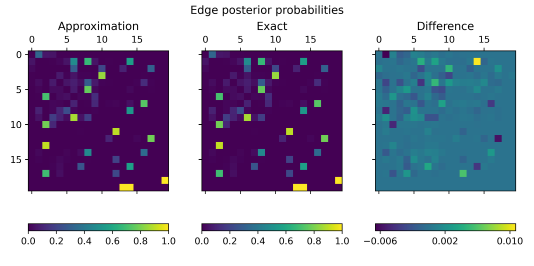
[9] Marco Grzegorczyk and Dirk Husmeier. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. Mach. Learn. 2008.

[10] Nir Friedman and Daphne Koller. Being Bayesian about network structure. Mach. Learn. 2003.

[11] Jack Kuipers and Giusi Moffa. Partition MCMC for inference on acyclic digraphs. J. Am. Stat. Assoc. 2017.

Performance

- How to measure the quality of MCMC?
- Hamming distance to ground truth
- Trace plot of scores or approximation errors
- Is there a more convincing method?



Constrained

Constrained Search Space

- Too many DAGs
- Consider only their subclass, e.g., rooted trees
- Are the problems tractable within that class?

Parameterized Learning

- Study the complexity under some parameterization
- For example, allow only DAGs with bounded vertex cover number[2]
- For optimization, wide variety of results are known[12]
- Ideally, complexities of the form $f(k) \cdot n^{O(1)}$
- More commonly W$[1]$-hard or NP-hard

---

[2] Juha Harviainen and Mikko Koivisto. Revisiting Bayesian Network Learning with Small Vertex Cover. UAI'23.

[12] Niels Grüttemeier and Christian Komusiewicz. Learning Bayesian networks under sparsity constraints: A parameterized complexity analysis. JAIR. 2022.

Superstructures

- DAGs must be subgraphs of a given superstructure
- Even optimization is NP-hard for undirected superstructures with bounded in-degree[13]
- Sampling and counting trivial for directed superstructures
- Parameterizing by a property of the superstructure

[13] Sebastian Ordyniak and Stefan Szeider. Parameterized Complexity Results for Exact Bayesian Network Structure Learning. JAIR. 2013.

Expressiveness

- DAGs with, e.g., only a few edges are rather inexpressive
- Algorithms of complexity $n^{\Omega(k)}$ impractical
- How expressive can we make the structures until the problem becomes intractable?

Concluding Remarks

Concluding Remarks

- Trade-offs
- Approach dependent on the application
- Many active research directions
- Best of all worlds?

Thank you!

Concluding Remarks

- Trade-offs
- Approach dependent on the application
- Many active research directions
- Best of all worlds?

Thank you!