Shortest Learning Paths in Bayesian Networks - LazyLearner

Peer Schendel | Seminar IPMoPGP

Problem Description - Real World Problem

- Student must catch up a certain competence
- Uses e-learning to achieve that
- Goal is to find the minimal number of courses to pass the target course

Problem Description - Real World Problem

- Student must catch up a certain competence
- Uses e-learning to achieve that
- Goal is to find the minimal number of courses to pass the target course

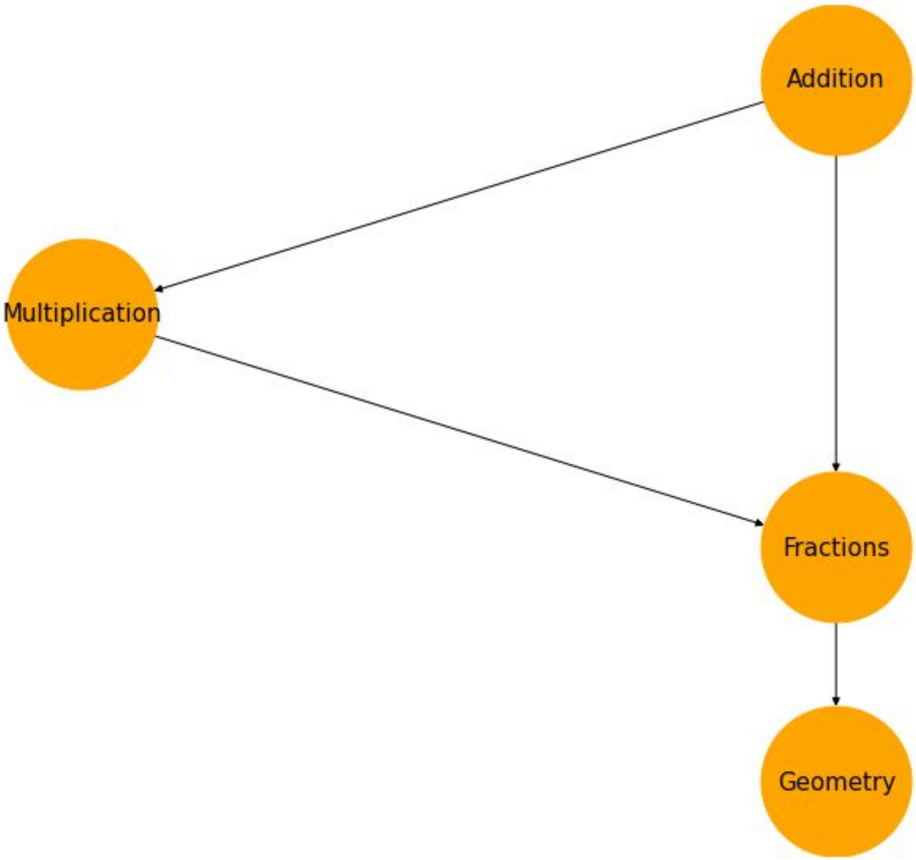


Fig 1: Different competences that are needed to learn geometry.

Bayesian Network (BN) - Data Structure

- Probabilistic graph model
- Often used for causal inference
- Directed acyclic graph (DAG)
- Edges correspond to causal relationships between nodes
- Markov Condition - the probability of one node only depends on their parents

$$P(x_1,...,x_n) = \prod_i P(x_i | pa_i)$$

Pearl, J., & Russel, S. (2000, November). BAYESIAN NETWORKS. Cambridge; MIT Press.

Minimal Cardinality Explanation

We define our decision function, whether a given instantiation yields a probability higher than the threshold as:

$$f(x) = \begin{cases} 1 \text{ if } P(\text{target node} = 1 \mid \text{x}) \geq t \\ 0 \text{ else} \end{cases}$$

- MC-explanation of a positive decision function f
  - Which positive features of instance x are responsible for the decision?
  - Find the minimal set of positive nodes that are responsible for the decision
- MC-explanations are not necessarily unique but must have identical number of 1-features

Shih, A., Choi, A., & Darwiche, A. (2018, July). A Symbolic Approach to Explaining Bayesian Network Classifiers. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18).

Conditional Probability Tables (CPT)

- BNs take advantage of conditional independence
- Reduction of parameters
- small representation of conditional probabilities only depending on parents instantiation
- We can use those CPTs per node as mapping for the IP modelling

General Problem Description

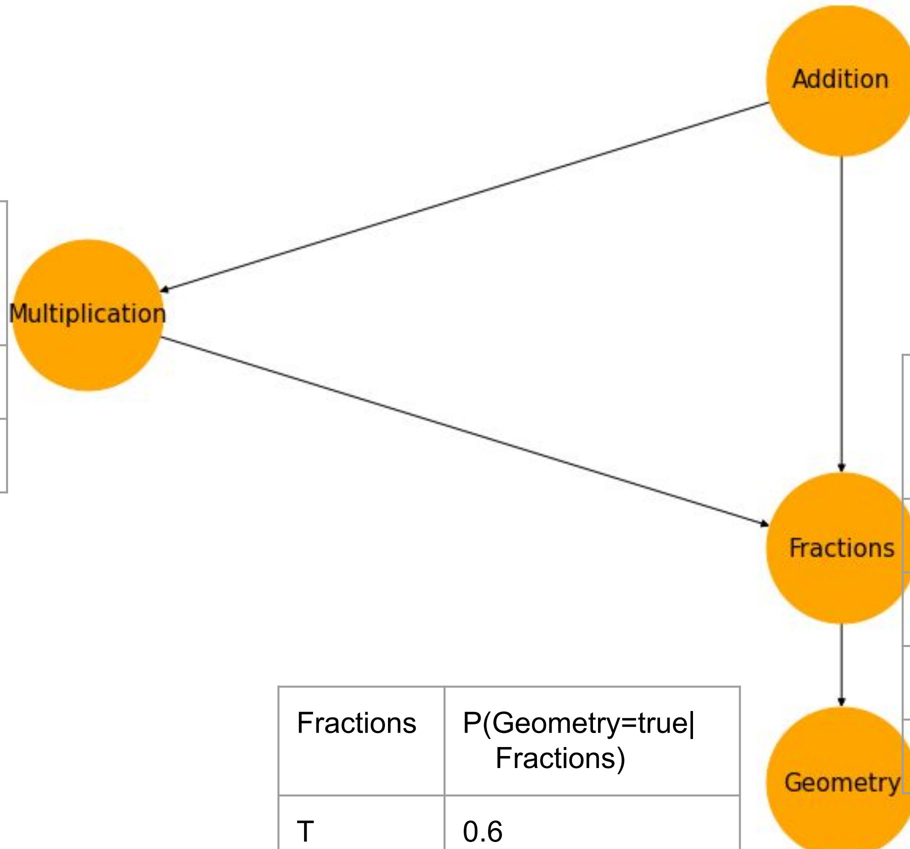
Given :

- Bayesian Network
- Target Node
- Threshold t
- Decision function f

Goal:

- Find MC-explanation of positive decision function of target node or return infeasibility
- Decision function f has to return 1 for all positive nodes
- Shortest path is the topological ordering of our MC-explanation on a complete graph

Example with Conditional Probability Tables



P(Addition=true)
0.5

Addition	P(Multipli-cation=true| Addition)
T	0.7
F	0.5

Addition	Multipli-cation	P(Fractions=true| Addition, Multiplication)
T	T	0.9
T	F	0.4
F	T	0.5
F	F	0.1

Fractions	P(Geometry=true| Fractions)
T	0.6
F	0.2

Complexity of the Problem

- We can efficiently verify if a given instantiation of parent nodes yield a “yes” instance that means P(target node = yes| instantiated parents) >= t
- We can build an edge case by having n-1 nodes as parents of 1 node:

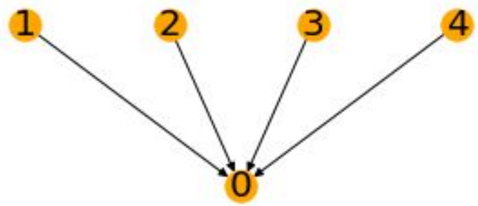


Fig 2: Edge case of n-1 parents

- The number of instances to check grow exponentially by 2^{n-1} in the worst case
- To find the minimal cardinality of such a set yields that the complexity of the problem is in NP

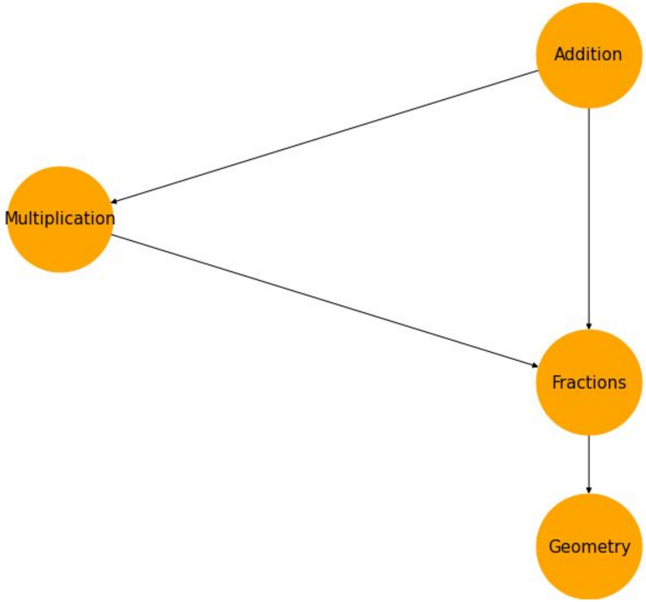
Modelling as IP

- Where t is the threshold which is for simplification 0.5 (passing an exercise)
- n is the number of nodes in the graph G
- S the target node in G
- Minimax Problem - minimize the number of nodes taken and maximize probability of target node

$$\min \quad z = -P(S=1|Y_{s-}) + \sum_{i=1}^{n} y_i$$

$$\text{subject to:} \quad \begin{aligned} P(S=1|Y_{s-}) &\geq t \\ P(Y_i=1|Y_{i-}) &\geq t \cdot y_i \quad \forall i=1,n \\ y_i &\in \{0,1\} \quad \forall i=1,n \end{aligned}$$

We define $Y_{i-} := \{Y_1 = y_1, ..., Y_{i-1} = y_{i-1}, Y_{i+1} = y_{i+1}, ..., Y_n = y_n\}$



IP Solver - GEKKO

- Open-source license
- Used python API for GEKKO [P2]
- For each node for each realization of parents add constraint:
  - check whether binary variables are equal to realization and check if binary variable is taken
- few extra variables and constraints needed for the maximization of the probability

Preprocessing of Steps

1. Get all parents recursively
2. Create CPT for every node by looping through all realizations of parents
3. Create binary variable for each node in set of all parents (1.)
4. For each node enter the recursive constraints for each realization of parents
5. Enter recursive constraints for target node
6. Add maximization of probability of target node

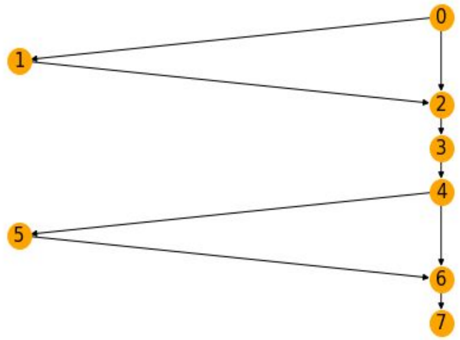
The preprocessing contains many for-loops that may be vectorized.

How to generate problem instances

- used package **pybbn** to generate small test instances [P1]
- can also generate random BN
  - no secure outcome for testing and scaling
  - bit different data format yields changes in code
- there are official data sets on the **bnlearn** homepage [P3]
  - no secure outcome for testing and scaling
  - different data format

For this presentation the scaling of problem instances was done by:

- building small test instances with known outputs
- joining them with an edge and adjusting the CPT for the successor



Computational Results

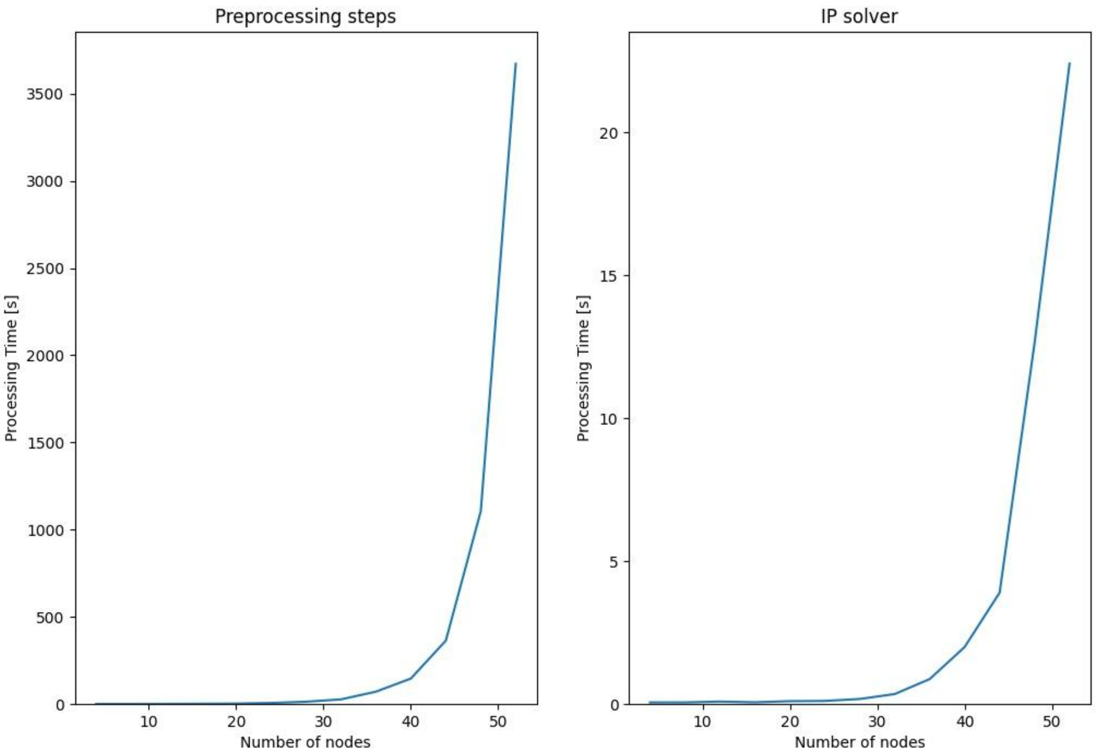


Fig 3: Processing time per number of generated test instance p.13

Possibly Variants - Strengthenings

- Pruning of nodes without interest - does not alter computational values of nodes of interest [2]
  - only using parents strengthens the IP and shortens the preprocessing

- Use another data structure to increase inference to linear time
  - compile BN to ordered binary decision diagram (OBDD) - MC-explanation runs in linear time [3]
  - has to be adjusted to fulfil the recursive threshold condition

Ordered Binary Decision Diagram - Example

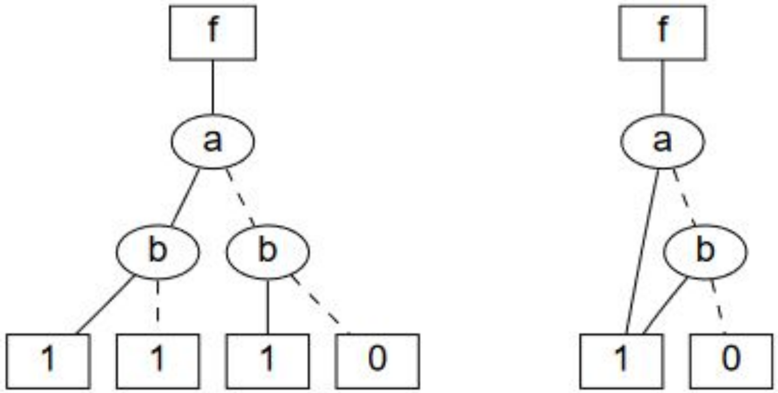


Fig 3: Decision tree and decision diagram for the disjunction from a and b [4]

- reduced decision trees
- 2 sinks - 0-sink and 1-sink
- needs compilation
- has to be modified for recursive condition

Conclusion

- Calculation of larger instances takes too long to use it in real-time
- IP without preprocessing, although to slow for real-time, solves the recursive condition by only adding one additional constraint
- Precalculated or data structures with faster calculation time are needed
    - for example OBDDs or other decision diagrams [3]

Literature Overview

- literature focuses on:
  - structure learning / parameter learning [5,8,12]
  - inference [6,7,10]
  - reasoning [6,9,11]

- No literature found for recursive explanation and not many sources on MC-Explanation
  - maybe wrong keywords
  - misunderstanding
  - no trivial solution to recursion constraint due to overlapping edges

Sources

Package sources:

- https://py-bbn.readthedocs.io/ [P1]
- https://gekko.readthedocs.io/en/latest/ [P2]
- https://www.bnlearn.com/bnrepository/ [P3]

Github repository:

- https://github.com/schpeer92/Optimal-Learning-Path-Recommender

Sources:

- Pearl, J., & Russel, S. (2000, November). BAYESIAN NETWORKS. Cambridge; MIT Press.[1]
- Boutilier, C., Friedman, N., Goldszmidt, M., & Koller, D. (1996). Context-Specific Independence in Bayesian Networks. Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI1996). [2]
- Shih, A., Choi, A., & Darwiche, A. (2018, May 9). A Symbolic Approach to Explaining Bayesian Network Classifiers. [3]
- Somenzi, F. (n.d.). Binary Decision Diagrams.[4]

Further Reading

- Trösser, F., de Givry, S., & Katsirelos, G. (2021, August). Improved Acyclicity Reasoning for Bayesian Network Structure Learning with Constraint Programming. Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21). [5]
- Nikpour, H., & Aamodt, A. (2021, March). Inference and reasoning in a Bayesian knowledge-intensive CBR system. Springer. [6]
- Darwiche, A. (2001). Recursive conditioning. Elsevier.[7]
- Friedman, N., Nachman, I., & Peér, D. (n.d.). Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm. [8]
- Chan, H., & Darwiche, A. (2003, August). Reasoning about Bayesian Network Classifiers. Acapulco; 19th Conference on Uncertainty in Artificial Intelligence.[9]
- Díez, F. J. (1996). Local conditioning in Bayesian networks. Elsevier.[10]
- Darwiche, A., & Hirth, A. (2020). On The Reasons Behind Decisions. European Conference on Artificial Intelligence (ECAI).[11]
- Cussens, J. (2012). Bayesian network learning with cutting planes.[12]