Bayesian Network Structure Learning with Side Constraints

Andrew Li and Peter van Beek
University of Waterloo, Canada

Introduction

- Two main approaches for constructing Bayesian networks (BNs) in practice:
  - Fully specified by domain expert
  - Learned directly from observed data
- Issues: experts rarely have complete domain knowledge; data can be limited or expensive.
- Hybrid BN structure learning methods that incorporate both **data** and **expert knowledge** have produced superior results in many fields [1,2].
- Local search methods for BN structure learning are orders of magnitude faster than exact methods, and consistently find near-optimal solutions [3].

Our Proposal

We build on **MINOBS**, a local search based memetic algorithm for BN structure learning that uses **ordering based search**. [3]

Expert Knowledge Constraints

- Express expert knowledge as a set of the following ***hard constraints***:
  - $x \rightarrow y$ (directed arc existence)
  - $x - y$ (undirected arc existence)
  - $x \nrightarrow y$ (directed arc absence)
  - $x < y$ (topological ordering)
  - $x \rightsquigarrow y$ (ancestral constraint)
- These constraints can indirectly specify causal tiers, root/leaf nodes.
- ***Ancestral constraints*** are notoriously difficult to incorporate [4] but are well motivated as they:
  - are more general than directed arc existence constraints.
  - assert (possibly indirect) causality, which is an abundant form of knowledge in many domains. [5]
  - distinguish between I-equivalent BNs

Main Goal: Find the DAG G satisfying all hard constraints such that the decomposable score $\sigma(G)$ is minimized.

Parent Set Identification

- Efficient local search requires pruning infeasible parent sets of variables:
  - If $x \rightarrow y$, prune parent sets of y not containing x.
  - If $x \nrightarrow y$, prune parent sets of y containing x.
  - If $x < y$, prune parent sets of x containing y.
  - If $x \rightsquigarrow y$, Π is a parent set of y, and for all $p \in \Pi$, $p < x$, then Π can be pruned.

Our Proposal

- The following heuristic rule prunes parent sets unlikely to appear in optimal BNs:

Let x be a node, Π, Π' potential parent sets for x, $\lambda \geq 1$ a constant such that $\Pi \subset \Pi'$ and $\lambda\sigma(x, \Pi) < \sigma(x, \Pi')$. Then Π' is pruned from the set of candidate parent sets.

Constraint-based Hill Climbing

- For a fixed topological ordering, DAGs G_1 and G_2 are ***parent assignment neighbours*** if only a single variable's parent set differs between G_1 and G_2.
- To handle ancestral constraints, we perform hill-climbing with the *parent assignment neighbourhood.*
  - In each iteration, selects the **first improving neighbour** (neighbours visited by increasing score).
  - **Optimization criteria**: most number of ancestral constraints satisfied (ties broken by lowest score)
  - Initial DAG does not necessarily satisfy ancestral constraints.
  - To overcome low quality *local minima* introduce **random walks** and a **tabu list**.

Ordering-based Search

- Search over the space of topological orderings using the **swap-adjacent** and **insert** neighbourhoods. [3]
- **Optimization criteria:** minimize score of the best DAG found under a fixed ordering (using the method from the previous section).

Experimental Evaluation

Instance	*N*	*%*	MINOBSx (time in *seconds*)	CaMML (time in *seconds*)
asia 8 vars 18 params	250	10 / 5	1.1 / 0.5	5.8 / 5.4
		100 / 20	0.5 / 0.2	8.3 / 5.7
	1000	10 / 5	0.9 / 0.4	5.5 / 5.0
		100 / 20	0.4 / 0.2	7.3 / 5.6
insurance 27 vars 984 params	500	10 / 5	180.5 / 104.9	439.5 / 325.3
		100 / 20	292.5 / 37.4	2052.3 / 571.7
	2000	10 / 5	124.0 / 88.3	438.5 / 309.8
		100 / 20	233.1 / 33.4	1956.8 / 571.1
barley 48 vars 114,005 params	2000	10 / 5	2321.4 / 5866.8	19824.8 / 11666.1
		100 / 20	7246.6 / 1806.3	114036.7 / 22366.9
	8000	10 / 5	4761.1 / 6032.7	18092.6 / 10759.6
		100 / 20	5675.6 / 1638.8	111338.4 / 21184.4

Table 1: The time required for our method (**MINOBSx**) and **CaMML** to complete on various benchmarks. *N* is the number of data points, *%* is the percentage of constraints sampled. Each entry containing a pair corresponds to a run using only ancestral constraints (first number in each pair) and using various constraints (second number in each pair).

All networks produced by MINOBSx were feasible (satisfied all hard constraints). For CaMML, on the *asia* instance nearly all networks produced were feasible, on the *insurance* instance the majority of networks were infeasible, and on the *barley* instance, all networks were infeasible.

Experimental Evaluation

Instance	*N*	*%*	Missing arcs	Extra arcs	Reversed arcs	SID	Score (BDeu)
asia 8 vars 18 params	250	0*	1.5	1.7	1.0	12.2	0%
		100 / 20	0.5 / 0.7	1.7 / 1.1	0.0 / 0.2	1.8 / 3.9	0.3% / 0.3%
	1000	0*	0.8	0.3	1.0	9.0	0%
		100 / 20	0.0 / 0.2	0.3 / 0.3	0.0 / 0.4	0 / 3.1	0% / 0%
child 20 vars 230 params	500	0*	5.3	1.0	3.0	115.7	0%
		100 / 20	2.2 / 3.6	2.2 / 0.8	0.0 / 0.3	35.8 / 57.6	0.9% / 0.4%
	2000	0*	1.7	0.2	3.5	79.2	0%
		100 / 20	0.2 / 1.0	0.3 / 0.1	0.0 / 0.0	2.5 / 16.6	0.1% / 0%
barley 48 vars 114,005 params	2000	0*	32.3	8.2	9.7	949.5	0%
		100 / 20	30.2 / 26.4	19.5/11.8	0.0 / 1.5	619.3 / 628.1	2.4% / 3.9%
	8000	0*	25.5	3.7	9.7	794.7	0%
		100 / 20	20.8 / 19.3	12.0 / 7.0	0.0 / 1.6	457.2 / 507.2	0.9% / 1.8%

Table 2: Various accuracy metrics of networks produced by **MINOBSx**. Instances with no constraints (marked by 0*) were optimally solved using GOBNILP [6]. SID measures the difference in causal statements between two networks [7]. The score column gives the % difference in score between the learned network and the optimal network produced by GOBNILP (with no constraints).

- We compared our method, **MINOBSx** against the widely-used **CaMML** [8], another stochastic score-and-search approach handling similar constraints.
- **MINOBSx** was able to quickly find feasible solutions to all problems, while **CaMML** performed consistently only on small problems.
- Networks produced by **MINOBSx** were closer to the ground truth when constraints were imposed; the score was also within 4% of optimal on all cases.
- A pitfall of **MINOBSx** is that a small parent limit (e.g. 3) must be set on large instances.

Conclusion

- We present a novel approximate method for incorporating prior knowledge constraints into BN structure learning, including non-decomposable *ancestral constraints*.
- Previous exact methods for incorporating ancestral constraints could handle up to *twenty* random variables [4]; we show our method scales to nearly *fifty* random variables while producing high quality networks.

References

1. M. J. Flores, A. E. Nicholson, A. Brunskill, K. B. Korb, and S. Mascaro. Incorporating expert knowledge when learning Bayesian network structure: a medical case study. Artificial intelligence in medicine, 53(3):181–204, 2011.
2. P. Antal, G. Fannes, D. Timmerman, Y. Moreau, and B. De Moor. Using literature and data to learn Bayesian networks as clinical models of ovarian tumors. Artificial Intelligence in Medicine, 30 (3):257–281, 2004.
3. C. Lee and P. van Beek. Metaheuristics for score-and-search Bayesian network structure learning. In Proceedings of the 30th Canadian Conference on Artificial Intelligence, pages 129–141, 2017. Available as: LNCS 10233.
4. E. Y.-J. Chen, Y. Shen, A. Choi, and A. Darwiche. Learning Bayesian networks with ancestral constraints. In Advances in Neural Information Processing Systems, pages 2325–2333, 2016.
5. T.-Y. Ma, J. Y. Chow, and J. Xu. Causal structure learning for travel mode choice using structural restrictions and model averaging algorithm. Transportmetrica A: Transport Science, 13(4):299–325, 2017.
6. M. Bartlett and J. Cussens. Integer linear programming for the Bayesian network structure learning problem. Artificial Intelligence, 244:258–271, 2017.
7. J. Peters and P. Buhlmann. Structural intervention distance for evaluating causal graphs. Neural computation, 27(3):771–799, 2015.
8. K. B. Korb and A. E. Nicholson. Bayesian Artificial Intelligence. CRC press, 2010.