Weighted Acyclicity Constraint for the Bayesian Network Structure Learning Problem

Simon de Givry \(^1\), George Katsirelos \(^1\), Fulya Ural \(^2\)

\(^1\)INRA MIAT

\(^2\)Grenoble INP - ENSIMAG

1 June 2018

\(^1\)This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissement d’avenir.
Bayesian Network Structure Learning

Bayesian network
is a probabilistic graphical model that captures the conditional dependencies between a set of random variables via a directed acyclic graph (DAG).

Application areas
include gene regulatory networks [Allouche et al., 2013], risk analysis [Trucco et al., 2008] and image processing [Luo et al., 2005].
Learning a Bayesian network from discrete data is known to be an NP-Hard problem [Chickering et al., 2004] with an exponential search space of DAGs.

Many solution approaches

- dynamic programming [Silander and Myllymäki, 2006, Fan and Yuan, 2015]
- constraint programming [Van Beek and Hoffman, 2015]
- propositional calculus [Cussens, 2008]
- breadth-first branch-and-bound search [Campos and Ji, 2011, Fan et al., 2014]
- integer linear programming [Bartlett and Cussens, 2017].
Our Interpretation of the BNSL

**CAST THE BNSL AS A COMBINATORIAL OPTIMIZATION PROBLEM**

**FORMULATE IT AS A WEIGHTED CONSTRAINT SATISFACTION PROBLEM**

**FIND AND IMPLEMENT IDEAS TO SOLVE IT FASTER**
Score-and-Search Method

Cost = 12
Acyclic? NO

Cost = 17
Acyclic? YES

Cost = 13
Acyclic? YES

Variable | Parent Set | Score
---|---|---
0 | {1, 2} | 2
| {1} | 4
| {2} | 3
| {} | 10

1 | {0, 2} | 1
| {0} | 2
| {2} | 6
| {} | 8

2 | {0, 1} | 1
| {0} | 4
| {1} | 7
| {} | 9
A Constraint Satisfaction Problem [Cooper and Schiex, 2004] is a triple \( \langle X, D, C \rangle \).

- \( X \): set of \( n \) variables \( X = \{1, \ldots, n\} \).
- \( D \): set of domains \( D = \{D_i : i \in X\} \).
- \( C \): set of constraints.

Each constraint \( c_S \in C \) is defined over a set of variables \( S \subseteq X \) (its scope) by a subset of the Cartesian product \( \prod_{i \in S} D_i = \ell(S) \). The cardinality \( |S| \) is the arity of the constraint \( c_S \).

A tuple \( t \in \ell(X) \) is a solution iff it satisfies all the constraints in \( C \).
A cost function is defined over the scope $S$ of the constraint $c_S$ to which it corresponds. It associates a cost to each tuple $t \in \ell(S)$.

- $c_\emptyset$: the nullary cost function = constant cost.
- $c_i$: the unary cost function on variable $i$.
- $c_{ij}$: the binary cost function on variables $i$ and $j$. 

<table>
<thead>
<tr>
<th>$x$</th>
<th>$c_x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$y$</th>
<th>$c_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$c_{xy}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>b</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>b</td>
<td>3</td>
</tr>
</tbody>
</table>

\[ \begin{align*}
\text{Variable } & \quad \text{Ordering} \\
X & \quad a \\
Y & \quad b
\end{align*} \]
Weighted Constraint Satisfaction Problem

Weighted Constraint Satisfaction Problem (WCSP) is a quadruple $\langle X, D, C, m \rangle$ where $C$ is a set of cost functions and $m$ is the upper bound [Cooper et al., 2010].

Find a solution such that the sum

$$c_\emptyset + \sum_{i \in X} c_i + \sum_{ij \in X^2} c_{ij}$$

- is minimized,
- is less than the upper bound $m$. 

Levels of Local Consistency

Node Consistency

A WCSP is *node consistent* (NC) [Cooper et al., 2010] if for any variable \( i \in \{1, \ldots, n\} \),

1. \( \forall a \in D_i, c_i(a) \oplus c_{\emptyset} < m \)
2. \( \exists a \in D_i \) such that \( c_i(a) = 0 \)

(Soft) Arc Consistency

A binary WCSP is arc consistent if for all \( c_{xy} \in C \) we have:

\[
\forall a \in D_x, \exists b \in D_y \text{ such that } c_{xy}(a, b) = 0.
\]
Levels of Local Consistency

**Bool(P)**

If \( P = \langle X, D, C, m \rangle \) is a WCSP, then \( \text{Bool}(P) = \langle X, D, \overline{C} \rangle \) is the classical CSP where, for all scopes \( S \neq \emptyset \), \( \langle S, R_S \rangle \in \overline{C} \) iff \( \exists \langle S, c_S \rangle \in C \), where \( R_S \) is the relation defined by \( \forall x \in \ell(S) (t \in R_S \iff c_S(t) = 0) \).

**Virtual Arc Consistency**

A WCSP \( P \) is virtual arc consistent (VAC) if \( \text{Bool}(P) \) is arc consistent.
Bayesian Network Structure Learning as a WCSP

<table>
<thead>
<tr>
<th>Set</th>
<th>Variable</th>
<th>Domain</th>
<th>Unary Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$P_i$</td>
<td>Some subsets of $X \setminus i$.</td>
<td>Scores.</td>
</tr>
<tr>
<td>$E$</td>
<td>$E_{ij}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$B$</td>
<td>$B_{ij}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>$m$ if $i = j$, 0 otherwise.</td>
</tr>
</tbody>
</table>

**Table:** Decision variables.
Constraints

- **\( P_i \) and \( E_{ij} \):** For all \( i, j \in X \), we have \( E_{ji} \Leftrightarrow (j \in P_i?) \).

- **\( E_{ij} \) and \( B_{ij} \):** For all \( i, j \in X \), we have \( E_{ij} \Rightarrow B_{ij} \).

- **\( B_{ij} \)'s:** For all \( i, k, j \in X \), we have \((B_{ik} \land B_{kj}) \Rightarrow B_{ij}\), which is equivalent to \( B_{ik} \lor B_{kj} \lor B_{ij} \).

- **Enforcing acyclicity:** For all \( i, j \in X \), we have \( B_{ii} = 0 \).

<table>
<thead>
<tr>
<th>( j \in P_i? )</th>
<th>( E_{ji} )</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>( m )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>( m )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( E_{ij} ) vs ( B_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_{ij} )</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

**Table:** Table costs.
A variable $x \in X$ is strictly arc consistent if there exist values $a \in D_x$ and $b \in D_y$ for all $y \in nb(x)$ such that:

- $c_x(a) < c_x(i) \ \forall i \in D_x \setminus a$
- $c_{xy}(a, b) < c_{xy}(i, j) \ \forall (i, j) \in \ell(\{x, y\}) \setminus (a, b)$.

$x = a$ and $y = b$ is obviously the optimal assignment.
Strict Arc Consistency

A variable $x \in X$ is *strictly arc consistent* if there exist values $a \in D_x$ and $b \in D_y$ for all $y \in nb(x)$ such that:

- $c_x(a) < c_x(i) \ \forall i \in D_x \setminus a$
- $c_{xy}(a, b) < c_{xy}(i, j) \ \forall (i, j) \in \ell({x, y})\setminus(a, b)$.
[Savchynskyy et al., 2013] suggests a method for energy minimization for Markov random fields:

- Divide the problem into two: easy and difficult, to be treated by convex and combinatorial techniques, respectively.

- Easy part is strictly arc consistent, while the difficult part is the rest.

- Use a similar idea for the $Bool(P)$ to improve the choice of variables during Branch-and-Bound.
Experiments

- ToulBar2: an open-source exact solver for cost function networks that solves various combinatorial optimization problems.

- 60 instances from [Haller et al., 2018]

- Time limit: 3600 seconds
Numerical Results

![Graph showing numerical results before and after some optimization. The x-axis represents the number of instances, and the y-axis represents time in seconds. The graph compares two scenarios: "BEFORE" and "AFTER." The "BEFORE" scenario shows a steady increase in time as the number of instances increases, while the "AFTER" scenario shows a significant improvement, especially after instance 10.]
Future Work

- **Heuristics:**
  - Stronger detection of tractable part of Bool(P)

- **BNSL:**
  - Improve complexity of each iteration of VAC
  - Dynamic computation of parent sets (exponentially large domains)
References

Allouche, David and Cierco-Ayrolles, Christine and De Givry, Simon
and Guillermín, Gérald and Mangin, Brigitte and Schiex, Thomas and
Vandel, Jimmy and Vignes, Matthieu (2013)
A panel of learning methods for the reconstruction of gene regulatory
networks in a systems genetics context

Bartlett, Mark and Cussens, James (2017)
Integer linear programming for the Bayesian network structure
learning problem
*Artificial Intelligence* 244, 45 – 678.

Campos, Cassio P de and Ji, Qiang (2011)
Efficient structure learning of Bayesian networks using constraints

Chickering, David Maxwell and Heckerman, David and Meek, Christopher (2004)
Large-sample learning of Bayesian networks is NP-hard
References

Cooper, Martin and Schiex, Thomas (2004)
Arc consistency for soft constraints
*Artificial Intelligence* 154(1-2), 199 – 227.

Cooper, Martin C and De Givry, Simon and Sánchez, Martí and Schiex, Thomas and Zytnicki, Matthias and Werner, Tomáš (2010)
Soft arc consistency revisited
*Artificial Intelligence* 174(7-8), 449 – 478.

Cussens, James (2008)
Bayesian network learning by compiling to weighted MAX-SAT

Fan, Xiannian and Yuan, Changhe and Malone, Brandon M (2014)
Tightening Bounds for Bayesian Network Structure Learning
*AAAI* 4, 2439 – 2445.
References


References

Silander, Tomi and Myllymäki, Petri (2006)
A simple approach for finding the globally optimal Bayesian network structure
Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence 445 – 452.

Trucco, Paolo and Cagno, Enrico and Ruggeri, Fabrizio and Grande, Ottavio (2008)
A Bayesian Belief Network modelling of organisational factors in risk analysis: A case study in maritime transportation
Reliability Engineering & System Safety 93(6), 845 – 856.

Van Beek, Peter and Hoffmann, Hella-Franziska (2015)
Machine learning of Bayesian networks using constraint programming
The End