Scale Up Bayesian Network Learning

Xiannian Fan

City University of New York
Overview

Structure Learning of Bayesian Networks
• Graph Search Formulation
• Proposal 1: Tightening Bounds
• Proposal 2: Prune using Constraints
• Summary
Learning optimal Bayesian networks

- Very often we have data sets.
- We can extract knowledge from these data.

<table>
<thead>
<tr>
<th>VisitToAsia</th>
<th>Smoking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuberculos...</td>
<td>LungCanc...</td>
</tr>
<tr>
<td>TuberOrCa...</td>
<td>Bronchitis</td>
</tr>
<tr>
<td>XRay</td>
<td>Dyspnea</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Success</th>
<th>Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Structure Learning
Graph Search
Tighten the Bounds
Prune by Constraints
Summary

<table>
<thead>
<tr>
<th>Success</th>
<th>Success</th>
<th>Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>Moderate</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>Poor</td>
<td>0.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Data
Structure
Numerical parameters
Score-based learning

- Find a Bayesian network that **optimizes** a scoring function, e.g.,

- Decomposability

\[
s(N) = \sum_{i=1}^{n} s_i(PA_i),
\]

where \( PA_i \) is the parent set of \( X_i \) in \( N \).
Graph Search Formulation

- Formulate the learning task as a **shortest path finding problem**
  - The shortest path solution to a graph search problem corresponds to an optimal Bayesian network

[Yuan, Malone, Wu, IJCAI-11]
Search graph (Order graph)

Formulation:
Search space: Variable subsets
Start node: Empty set
Goal node: Complete set
Edges: Select parents
Edge cost: BestScore\((X,U)\) for edge \(U \rightarrow U \cup \{X\}\)

Task: find the shortest path between start and goal nodes

[Yuan, Malone, Wu, IJCAI-11]
Search Space

- **Size:** $O(2^n)$
Sparse Parent Graph

- Potential Optimal Parent Set for $X_1$

(a) $s_1(PA_1)$

(b) $s_1(PA_1)$

Propagate the best scores.
Spare Parent Graph

- Potential Optimal Parent Set for $X_1$

(b) $s_1(PA_1)$

(c) $s_1(PA_1)$

Prune useless local scores.
Potentially Optimal Parent Sets (POPS)

• Store the potentially optimal parent sets in a **sorted** array based on their scores

<table>
<thead>
<tr>
<th>$X_2, X_3$</th>
<th>$X_3$</th>
<th>$X_2$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>8</td>
<td>10</td>
</tr>
</tbody>
</table>

• For any candidate set $U$, find optimal parents by scanning from the beginning until finding a subset of $U$. 
Heuristic search: Expands the nodes in the order of promisingness: $f = g + h$

- $g(U) = MDL(U)$
- $h_{\text{simple}}(U) = \sum_{X \in V \setminus U} \text{BestMDL}(X, V \setminus \{X\})$

$h_{\text{simple}}(\{2,3\})$:
Pruning Based on Bounds

- **Improved search heuristic** [Yuan and Malone, UAI-12]
- **Tightening Bounds** [Fan, Yuan and Malone, AAA-14]
Proposal I: Tightening Bounds

- Tightening the Upper Bound:

  - Anytime window A* (AWA*) was shown to find high quality, often optimal, solutions very quickly, thus provided a tight upper bound.
Proposal I: Tightening Bounds

• Tightening the Lower Bound: By Heuristics
  – Simple Heuristic [Yuan IJCAI 2011]
  – K-Cycle Heuristics [Yuan and Malone, UAI-12]
  – Proposed New Static K-Cycle Heuristics [Fan, Yuan and Malone, AAAI14]
Simple Heuristic

• Let each variable to choose optimal parents from all the other variables
• Completely relaxes the *acyclic* constraint

---

Bayesian network

Heuristic estimation

![Diagram showing a Bayesian network and its heuristic estimation with relaxation](image)
Static k-cycle conflict heuristic

- Also called static pattern database
- Calculate joint costs for all subsets of non-overlapping static groups by enforcing acyclicity within a group:

$$\{1,2,3,4,5,6\} \Rightarrow \{1,2,3\}, \{4,5,6\}$$

$$h(\{1\}) = c(\{2,3\}) = g^r(\{1\})$$

[Yuan, Malone, UAI-12]
Computing heuristic value using static PD

• Sum costs of pattern databases according to static grouping

\[ h(\{1, 5, 6\}) = c(\{2, 3\}) + c(\{4\}) = g^r(\{1\}) + g^r(\{5, 6\}) \]

[Yuan, Malone, UAI-12]
Tightening the Lower Bound

- Tightness of the heuristic highly depends on the grouping

- Characteristics of a good grouping
  - Reduce directed cycles between groups
  - Enforce as much acyclicity as possible

[Fan, Yuan, AAAI-15]
More Informed Grouping Strategies,
Rather than use SG (1\textsuperscript{st} half VS 2\textsuperscript{nd} half grouping), we developed more informed grouping strategies.

– Maximizing the correlation between the variables within each group, and Minimize the correlation between groups.

– Using Topological Ordering Information
Correlation Based Grouping

- Create an undirected graph as skeleton
  - Parent grouping: connecting each variable to potentials parents in the best POPS
  - Family grouping: use Min-Max Parent Child (MMPC) [Tsarmardinos et al. 06]

- Use independence tests in MMPC to estimate edge weights

- Partition the skeleton into balanced subgraphs
  - by minimizing the total weights of the edges between the subgraphs

[Fan, Yuan, AAAI-15]
Result of Upper Bounds by AWA*

(a) Parkinsons

(b) Steel Plates
Results of Different Grouping

(a) Parkinsons

(b) Steel Plates
First observation

• Potentially Optimal Parent Sets (POPS) Table
  – Contain all parent-child relations

Another example with 6 variables:

<table>
<thead>
<tr>
<th>variable</th>
<th>POPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_1)</td>
<td>{X_2}</td>
</tr>
<tr>
<td>(X_2)</td>
<td>{X_1}</td>
</tr>
<tr>
<td>(X_3)</td>
<td>{X_1, X_2}</td>
</tr>
<tr>
<td>(X_4)</td>
<td>{X_1, X_3}</td>
</tr>
<tr>
<td>(X_5)</td>
<td>{X_4}</td>
</tr>
<tr>
<td>(X_6)</td>
<td>{X_2, X_5}</td>
</tr>
</tbody>
</table>

• Observation: Not all variables can possibly be ancestors of the others.
  – E.g., any variables in \{X_3, X_4, X_5, X_6\} can not be ancestor of \(X_1\) or \(X_2\)
POPS Constraints

- **Parent Child Graph**
  - Formed by Aggregating the POPS

- **Component Graph**
  - Formed by Strongly Connected Components (SCCs)
  - Give the Ancestor Constraints

---

Structure Learning
Graph Search
Tighten the Bounds
Prune by Constraints
Summary
POPS Constraints

• Decompose the Problem
  – Each SCC corresponds to a smaller subproblem
  – Each subproblem can be solved independently.
POPS Constraints

- Recursive POPS Constraints
  - Selecting the parents for one of the variables has the effect of removing that variable from the parent relation graph.
Experiment: POPS and Recursive POPS Constraints

Alarm, 37: # Expanded Nodes (million)

- No Constraint
- POPS
- Recursive POPS

Alarm, 37: Running Time (seconds)

- No Constraint
- POPS
- Recursive POPS
Experiment: POPS and Recursive POPS Constraints

Barley, 48: # Expanded Nodes (million)

- No Constraint
- POPS
- Recursive POPS

Barley, 48: # Running Time (seconds)

- No Constraint
- POPS
- Recursive POPS
Experiment: POPS and Recursive POPS Constraints

Soybean, 36: # Expanded Nodes (seconds)

No Constraint  | POPS  | Recursive POPS
--- | --- | ---

Soybean, 36: # Running Time (seconds)

No Constraint  | POPS  | Recursive POPS
--- | --- | ---
Prune using POPS Constraints

- This Work: Reduce Search Space Using POPS Constraints

The (worst-case) size of the original order graph for $n$ variables is as follows.

$$O(2^{scc_1} + \ldots + 2^{scc_m}) = O(2^n)$$  \hspace{1cm} (3)

The worst-case size of the search space after splitting into subproblems using the SCCs is as follows.

$$O(2^{scc_1} + \ldots + 2^{scc_m}) = O(m \cdot \max_{i} 2^{\left|scc_i\right|})$$  \hspace{1cm} (4)
Bridge Constraints

• Example

• Using Strong Bridges Finding Algorithm to find the bridges that better partition the groups (Currently we just consider the strong bridge, so only 2 partition groups).
Prune Using Bridge Constraints

Structure Learning
Graph Search
Tighten the Bounds
Prune by Constraints
Summary
Summary: Scale Up Bayesian Networks Learning

• Potential Ways to Scale Up
  – Tightening bounds
  – Extract more information to prune

• Future Work
  – Extract more information from data
  – Using domain knowledge