DNNF-based Belief State Estimation

Paul Elliott
Brian Williams

Motivation

- Complex embedded systems

Belief State Estimation

Belief State Estimation

Behavior Model (PCCA)
Observations
Commands
Plant S
Belief State S

Approximate Belief State Estimation

Observations
Commands
Plant S
Belief State Estimation k-best States S

Compiled Approx Belief State Estimation

Observations
Commands
Plant S
Belief State Estimation k-best States S

Compiled Approx Belief State Estimation

Observations
Commands
Plant S
Belief State Estimation k-best States S
Prior Work

- MEXEC (Barrett '05)
  - Trajectories
  - Max-Prod
  - sd-DNNF
- BFBSU (Martin '05)
  - Approximate Belief State
  - Optimal constraint-solver
- Dynamic Bayesian Networks
  - Max-Prod
  - Sum-Prod
  - Distribution on variables

Contributions

- New encoding of estimation for compilation
- New use of sd-DNNF
- New k-best algorithm for sd-DNNF
- New bounded believe state estimation algorithm
Estimation Function

PCCA as a Relation

PCCA as a Probabilistic Relation

PCCA as a Probabilistic Relation

sd-DNNF: Relation Encoding

sd-DNNF: Model Counting
**sd-DNNF: Probabilities**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Prev State</th>
<th>Prev Obs</th>
<th>Next State</th>
<th>Next Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TurnOn</td>
<td>Switch</td>
<td>Inverter</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

\[
\Pi_x \cdot B(S) \cdot B(S,O) / \#M(S)
\]

<table>
<thead>
<tr>
<th>States</th>
<th>#M(S,O) / #M(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.960</td>
</tr>
<tr>
<td></td>
<td>0.989</td>
</tr>
</tbody>
</table>

**sd-DNNF: States**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Prev State</th>
<th>Prev Obs</th>
<th>Next State</th>
<th>Next Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TurnOn</td>
<td>Switch</td>
<td>Inverter</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

\[
\Pi_x \cdot B(S) \cdot B(S,O) / \#M(S)
\]

<table>
<thead>
<tr>
<th>States</th>
<th>#M(S,O) / #M(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.960</td>
</tr>
<tr>
<td></td>
<td>0.989</td>
</tr>
</tbody>
</table>

**Estimation**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Prev State</th>
<th>Prev Obs</th>
<th>Next State</th>
<th>Next Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TurnOn</td>
<td>Switch</td>
<td>Inverter</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

\[
B^{**}(S) \cdot \arg \max \ B^{**}(S) / \#M(S)
\]

<table>
<thead>
<tr>
<th>States</th>
<th>#M(S,O) / #M(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.960</td>
</tr>
<tr>
<td></td>
<td>0.989</td>
</tr>
</tbody>
</table>

**Estimation: k-best Approx**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Prev State</th>
<th>Prev Obs</th>
<th>Next State</th>
<th>Next Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TurnOn</td>
<td>Switch</td>
<td>Inverter</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

\[
B^{**}(S) \cdot \arg \max \ B^{**}(S) / \#M(S)
\]

<table>
<thead>
<tr>
<th>States</th>
<th>#M(S,O) / #M(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.960</td>
</tr>
<tr>
<td></td>
<td>0.989</td>
</tr>
</tbody>
</table>

**Estimation: k-best Approx (Division)**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Prev State</th>
<th>Prev Obs</th>
<th>Next State</th>
<th>Next Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TurnOn</td>
<td>Switch</td>
<td>Inverter</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

\[
B^{**}(S) / \#M(S) \cdot \arg \max B^{**}(S) / \#M(S)
\]

<table>
<thead>
<tr>
<th>States</th>
<th>#M(S,O) / #M(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.960</td>
</tr>
<tr>
<td></td>
<td>0.989</td>
</tr>
</tbody>
</table>

**Estimation: k-best Approx**

<table>
<thead>
<tr>
<th>Commands</th>
<th>Prev State</th>
<th>Prev Obs</th>
<th>Next State</th>
<th>Next Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TurnOn</td>
<td>Switch</td>
<td>Inverter</td>
<td>Output</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{Max-Sum-Product-Sum}
\]

<table>
<thead>
<tr>
<th>States</th>
<th>#M(S,O) / #M(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.960</td>
</tr>
<tr>
<td></td>
<td>0.989</td>
</tr>
</tbody>
</table>
Estimation: k-best Approx (Division)

C2D 2005 v2.20: Compilation

Labeled sd-DNNF

Estimation Algorithm
1. Assign known values and probabilities
2. First Pass: Compute all summations
3. Merge summation results
4. Second Pass: Extract the k-best solutions
   1. Upward part: Compute the probabilities and paths of the k-best estimates
   2. Downward part: Follow the paths to extract the k-best estimates

Algorithm: Initial Assignment

Given: $B_t(S, O) = 1$

1

Given: $B_t(S, O) = 1$

Low

1

Algorithm: k-Best Probabilities

Given: OnBt( ) = 1

Algorithm: k-Best Probabilities

Given: OnBt( ) = 1

Algorithm: k-Best Probabilities

Given: OnBt( ) = 1

Algorithm: k-Best Probabilities

Given: OnBt( ) = 1

Algorithm: k-Best Solutions

Given: OnBt( ) = 1

Algorithm: k-Best Solutions

Given: OnBt( ) = 1

Algorithm: k-Best Solutions

Given: OnBt( ) = 1

Algorithm: k-Best Solutions

Given: OnBt( ) = 1
Switch
Inverter
Output
Switch
Inverter
Output
Low

Given:
On( ) = 1
P(1) = 0.999
P(2) = 0.0005

Belief
State
Estimation
Behavior
Model (PCCA)

k-best States
S
Compiler
Observations
Commands
Plant

Results

• EDL Model
  – 42 Variables
  – 10 State Variables
  – 4.4 Values/Variable

• Model Size (k > 3)
  – $(71 \pm 2.5) \times 10^3 \times k$ nodes
  – $(244 \pm 4.7) \times 10^3 \times k$ edges

• Model Size (k < 3)
  – $(4.4 \pm 0.37) \times 10^3 \times k$ nodes
  – $(14.8 \pm 0.34) \times 10^3 \times k$ edges

• Algorithm Complexity
  – $O(k^2 \epsilon_n)$ space
  – $O(k^3 \epsilon_n)$ time

Conclusion

• New Estimation Algorithm
  – Linear in the size of the sd-DNNF, cubic time with $k$
  – Estimates $k$ belief states

• New Algorithm for computing Max-Sum-Product on an sd-DNNF
  – Operates in two phases
    1. Computes Sum-Product
    2. Computes Max-Product

• An encoding of PCCA estimation as a probabilistic relation

Questions?