Bayesian network learning by compiling to weighted MAX-SAT

James Cussens, University of York

SML, 2008-04-25
Work to be presented at UAI ’08
Introduction

Learning Bayesian networks

Weighted MAX-SAT

Compiling to weighted MAX-SAT

Results
Motivation

- A similar motivation to Siegfried . . .
- Why write (yet another) BN learning algorithm when weighted SAT search can do it?
- It’s hard to get MCMC to converge for Bayesian model averaging so use search to do it.
Outline

Introduction

Learning Bayesian networks

Weighted MAX-SAT

Compiling to weighted MAX-SAT

Results
### Bayesian network learning

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$\Rightarrow$

![Bayesian network diagram](image)

James Cussens, University of York

BN learning via weighted MAX-SAT
BN structure learning by maximising the (marginal) likelihood

- Marginal likelihood: \( P(D|G) = \int_\theta P(D|G, \theta)P(\theta)d\theta \)
- Score(\(G\)) \( \overset{\text{def}}{=} \log P(D|G) \).
- Search for \( \text{arg max}_G \text{Score}(G) \)
- Looking for DAGs \( G \) such that \( \text{Score}(G) \) is a small negative number.
Scoring Bayesian networks

\[
\text{Score} \left( \begin{array}{c}
X_2 \\
X_4
\end{array} \right) \\
\begin{array}{c}
X_1 \\
X_3
\end{array} = \text{Score}_1 \left( \begin{array}{c}
X_1
\end{array} \right) + \ldots
\]
Scoring Bayesian networks

\[
\text{Score} \left( \begin{array}{c}
X_2 \\
X_1 \\
X_3 \\
X_4
\end{array} \right) + \cdots + \text{Score}_2 \left( \begin{array}{c}
X_1 \\
X_2
\end{array} \right) + \cdots
\]
Scoring Bayesian networks

Score \left( \begin{array}{cccc}
X_1 \\
X_2 \\
X_3 \\
X_4
\end{array} \right) + \cdots + \text{Score}_3 \left( \begin{array}{c}
X_1 \\
X_3
\end{array} \right) + \cdots
Scoring Bayesian networks

\[
\text{Score} \left( \begin{array}{c}
X_1 \\
X_2 \\
X_3 \\
X_4
\end{array} \right) + \text{Score}_4 \left( \begin{array}{c}
X_2 \\
X_3 \\
X_4
\end{array} \right)
\]
Pre-computing ‘family’ scores (e.g. for $X_4$)

Add $\text{Score}_4$ to the score store for $X_4$. 

$$
\begin{pmatrix}
\end{pmatrix}
$$
Pre-computing ‘family’ scores (e.g. for $X_4$)

Add $\text{Score}_4$ to the score store for $X_4$. 

$$
\begin{pmatrix}
X_1 \\
\downarrow \\
X_4
\end{pmatrix}
$$
Pre-computing ‘family’ scores (e.g. for $X_4$)

Add $\text{Score}_4$ (\begin{pmatrix} X_2 \\ X_4 \end{pmatrix}) to the score store for $X_4$. 
Pre-computing ‘family’ scores (e.g. for $X_4$)

Add $\text{Score}_4$ to the score store for $X_4$. 

$$
\begin{pmatrix}
X_4 \\
X_3
\end{pmatrix}
$$
Pre-computing ‘family’ scores (e.g. for $X_4$)

$$\text{Add Score}_4 \begin{pmatrix} X_1 \\ X_2 \\ X_4 \end{pmatrix}$$

to the score store for $X_4$. 
Pre-computing ‘family’ scores (e.g. for $X_4$)

Add $\text{Score}_4$ to the score store for $X_4$. 
Pre-computing ‘family’ scores (e.g. for $X_4$)

Add $\text{Score}_4$ to the score store for $X_4$. 

\[ \begin{pmatrix} X_2 \\ X_3 \\ X_4 \end{pmatrix} \]
Pre-computing ‘family’ scores (e.g. for $X_4$)

Add $\text{Score}_4$ to the score store for $X_4$. 

$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$
Family scores for some datasets

With a limit of at most 3 parents for each variable.

| Name     | $n$ | $|\text{scores}|$ | Time taken for $N = \ldots$ |
|----------|-----|----------------|-----------------------------|
|          |     |                | $10^2$ | $10^3$ | $10^4$ |
| Mildew   | 35  | 230,300        | 1,678 | 1,942 | 4,502 |
| Water    | 32  | 159,744        | 22    | 123   | 1,093 |
| alarm    | 37  | 288,859        | 38    | 208   | 1,501 |
| asia     | 8   | 512            | 0     | 0     | 0     |
| carpo    | 60  | 2,056,800      | 544   | 1,811 | 13,892|
| hailfinder | 56  | 1,555,456      | 852   | 2,974 | 22,666|
| insurance| 27  | 79,704         | 29    | 97    | 749   |
Filtering ‘family’ scores

If

\[
\begin{pmatrix}
X_1 \\
X_4
\end{pmatrix}
\]

then throw RHS score away.
## Results of filtering

<table>
<thead>
<tr>
<th>Dat</th>
<th>$10^2$ $N = \ldots$</th>
<th>$10^3$</th>
<th>$10^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>max</td>
<td>$n$</td>
<td>max</td>
</tr>
<tr>
<td>Mi</td>
<td>977</td>
<td>3,515</td>
<td>17</td>
</tr>
<tr>
<td>Wa</td>
<td>44</td>
<td>482</td>
<td>44</td>
</tr>
<tr>
<td>al</td>
<td>75</td>
<td>907</td>
<td>184</td>
</tr>
<tr>
<td>as</td>
<td>10</td>
<td>41</td>
<td>24</td>
</tr>
<tr>
<td>ca</td>
<td>350</td>
<td>5,068</td>
<td>352</td>
</tr>
<tr>
<td>ha</td>
<td>22</td>
<td>244</td>
<td>77</td>
</tr>
<tr>
<td>in</td>
<td>28</td>
<td>279</td>
<td>95</td>
</tr>
</tbody>
</table>
Why not just pick the highest scoring parents for each variable?

Because (probably) that would be produce a *cyclic* graph.

BN learning is a finite domain constraint problem.

- Map each BN variable to a constrained variable; its candidate parent sets are its values.
- Infeasible assignments are those which cause cycles.
Outline

Introduction

Learning Bayesian networks

Weighted MAX-SAT

Compiling to weighted MAX-SAT

Results
Introduction
Learning Bayesian networks
Weighted MAX-SAT
Compiling to weighted MAX-SAT
Results

Motivation

- It’s quite a big problem.
- So try out an approach that has been shown to work on big problems.
SAT, MAX-SAT and weighted MAX-SAT

**SAT** Given propositional clauses, set the truth values of the atoms to satisfy all clauses; or show that no satisfying assignment exists.

**MAX-SAT** Given propositional clauses, satisfy as many clauses as possible.

**Weighted MAX-SAT** Given weighted propositional clauses, maximise the sum of the weights of satisfied clauses.

**Weighted MAX-SAT** (Equivalently) Minimise the sum of the weights of unsatisfied clauses.
The MaxWalkSAT algorithm

while stillTrying:
    somehowAssignTruthValuesToAllAtoms
while cost <= target:
    c = random_choice(unsat_clauses)
    lits = lits_of(c)
    if random_flip:
        lit = random_choice(lits)
    else:
        lit = lowest_cost_flip(lits)
    flip_truth_value(lit)
    update_cost
Outline

Introduction

Learning Bayesian networks

Weighted MAX-SAT

Compiling to weighted MAX-SAT

Results
Encoding: the atoms

- For each pair of atoms $a, b$, whether $a$ is an ancestor of $b$ is represented by a propositional atom.
- Each choice of parents for each variable is represented by a propositional atom.

- Atom 1 means Bronchitis $\rightarrow \rightarrow$ Cancer
- Atom 2 means Bronchitis $\rightarrow \rightarrow$ Dyspnea

- Atom 118 means that Cancer has exactly these parents: Dyspnea, Tuberculosis, Bronchitis.
Encoding: the hard clauses

- For all $a, b, c$: $(a \rightarrow b) \land (b \rightarrow c) \Rightarrow (a \rightarrow c)$
- Every variable must have some choice of parents.

$c$ Clause -1 -9 2: Bronchitis $\rightarrow\rightarrow$ Cancer
& Cancer $\rightarrow\rightarrow$ Dyspnea
=> Bronchitis $\rightarrow\rightarrow$ Dyspnea

$c$ Clause 58 59 ... 121:
since exactly one family for Cancer
Encoding: the hard clauses (ctd.)

- Choosing parents implies ancestor relations: e.g. if $a$ is chosen as one of the parents of $b$, then $a$ is an ancestor of $b$.

- **No cycles** $\forall a, b : \neg(a \rightarrow b) \lor \neg(b \rightarrow a)$

- c Clause -118 1: Cancer has parents
  Dyspnea, Tuberculosis, Bronchitis
  $\Rightarrow$ Bronchitis $\text{->>}$ Cancer

- c Clause -8 -9: not(Cancer $\text{->>}$ Bronchitis) or not (Bronchitis $\text{->>}$ Cancer)
Encoding: the soft clauses

- For each choice of parents for each variable assert that the choice is false (a single negative literal clause).
- But *weight* each such clause with $-1$ times the score for this choice.

 Soft clause 30 $-118$: since Cancer with parents Dyspnea, Tuberculosis, Bronchitis has score $-30.275543$ and scaling is 1
## Size of CNF problems

<table>
<thead>
<tr>
<th>Data</th>
<th>atoms</th>
<th>clauses</th>
<th>lits</th>
</tr>
</thead>
<tbody>
<tr>
<td>as_2</td>
<td>98</td>
<td>488</td>
<td>1,351</td>
</tr>
<tr>
<td>as_3</td>
<td>164</td>
<td>693</td>
<td>1,761</td>
</tr>
<tr>
<td>as_4</td>
<td>218</td>
<td>873</td>
<td>2,121</td>
</tr>
<tr>
<td>ca_2</td>
<td>8,609</td>
<td>226,406</td>
<td>661,551</td>
</tr>
<tr>
<td>ca_3</td>
<td>7,368</td>
<td>221,365</td>
<td>651,469</td>
</tr>
<tr>
<td>ca_4</td>
<td>19,932</td>
<td>269,367</td>
<td>747,473</td>
</tr>
<tr>
<td>in_2</td>
<td>982</td>
<td>18,926</td>
<td>56,049</td>
</tr>
<tr>
<td>in_3</td>
<td>1,477</td>
<td>20,346</td>
<td>58,889</td>
</tr>
<tr>
<td>in_4</td>
<td>4,355</td>
<td>30,344</td>
<td>78,885</td>
</tr>
</tbody>
</table>
Outline

Introduction

Learning Bayesian networks

Weighted MAX-SAT

Compiling to weighted MAX-SAT

Results
Synthetic data

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
It’s fast

newmaxwalksat version 20 (Huge)
numatom = 6848, numclause = 181544, numliterals = 529296

<table>
<thead>
<tr>
<th>lowest cost</th>
<th>worst clause</th>
<th>#unsat</th>
<th>#flips</th>
</tr>
</thead>
<tbody>
<tr>
<td>this try</td>
<td>506076</td>
<td>16968</td>
<td>56</td>
</tr>
<tr>
<td>this try</td>
<td>501973</td>
<td>23318</td>
<td>56</td>
</tr>
</tbody>
</table>

total elapsed seconds = 75.428415
average flips per second = 171206
number of solutions found = 1

ASSIGNMENT ACHIEVING TARGET 503040 FOUND
## Searching for high scoring BNs

<table>
<thead>
<tr>
<th>Data</th>
<th>True</th>
<th>Target</th>
<th>no cycle_atom__tries=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mi_2</td>
<td>-7,786</td>
<td>-6,611</td>
<td>-5,711</td>
</tr>
<tr>
<td>Mi_3</td>
<td>-63,837</td>
<td>-52,866</td>
<td>-47,229</td>
</tr>
<tr>
<td>Mi_4</td>
<td>-470,215</td>
<td>-459,874</td>
<td>-409,641</td>
</tr>
<tr>
<td>Wa_2</td>
<td>-1,801</td>
<td>-1,523</td>
<td>-1,488</td>
</tr>
<tr>
<td>Wa_3</td>
<td>-13,843</td>
<td>-13,304</td>
<td>-13,293</td>
</tr>
<tr>
<td>Wa_4</td>
<td>-129,655</td>
<td>-128,745</td>
<td>-129,274</td>
</tr>
<tr>
<td>al_2</td>
<td>-1,410</td>
<td>-1,383</td>
<td>-1,368</td>
</tr>
<tr>
<td>al_3</td>
<td>-11,305</td>
<td>-11,255</td>
<td>-11,599</td>
</tr>
<tr>
<td>al_4</td>
<td>-105,303</td>
<td>-105,226</td>
<td>-107,205</td>
</tr>
</tbody>
</table>
### Searching for high scoring BNs

<table>
<thead>
<tr>
<th>Data</th>
<th>True</th>
<th>Target</th>
<th>no cycle_atom tries=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>as_2</td>
<td>-247</td>
<td>-245</td>
<td>-241</td>
</tr>
<tr>
<td>as_3</td>
<td>-2,318</td>
<td>-2,318</td>
<td>-2,312</td>
</tr>
<tr>
<td>as_4</td>
<td>-22,466</td>
<td>-22,466</td>
<td>-22,462</td>
</tr>
<tr>
<td>ca_2</td>
<td>-1,969</td>
<td>-1,952</td>
<td>-1,849</td>
</tr>
<tr>
<td>ca_3</td>
<td>-17,739</td>
<td>-17,821</td>
<td>-17,938</td>
</tr>
<tr>
<td>ca_4</td>
<td>-173,682</td>
<td>-175,349</td>
<td>-175,832</td>
</tr>
</tbody>
</table>
## Searching for high scoring BNs

<table>
<thead>
<tr>
<th>Data</th>
<th>True</th>
<th>Target</th>
<th>no cycle_atom tries=100</th>
</tr>
</thead>
<tbody>
<tr>
<td>ha_2</td>
<td>-6,856</td>
<td>-6,058</td>
<td>-5,998</td>
</tr>
<tr>
<td>ha_3</td>
<td>-55,366</td>
<td>-52,747</td>
<td>-53,059</td>
</tr>
<tr>
<td>ha_4</td>
<td>-503,040</td>
<td>-498,163</td>
<td>-506,219</td>
</tr>
<tr>
<td>in_2</td>
<td>-1,951</td>
<td>-1,715</td>
<td>-1,690</td>
</tr>
<tr>
<td>in_3</td>
<td>-14,411</td>
<td>-13,919</td>
<td>-14,105</td>
</tr>
<tr>
<td>in_4</td>
<td>-133,489</td>
<td>-132,968</td>
<td>-134,378</td>
</tr>
</tbody>
</table>
Bayesian model averaging

- Just collect BNs (truth assignments) generated in the search …
- …together with their scores (likelihoods).
- (And pipe through `sort -u`.)
- Just give zero probability to all other networks and normalise to get a posterior distribution over networks.
- Excellent results for asia, generally bad in other cases.