MLRG: The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks

Presented by Kui Tang

20 June 2012

Bayesian belief networks are essential modelling tools

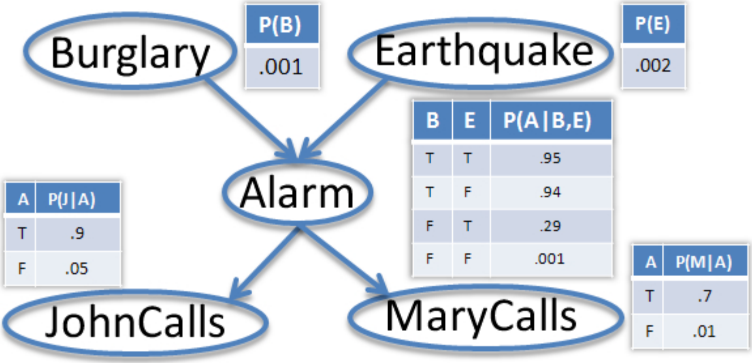
- Express densities functions of the form

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\pi_i)$$

where π_i denotes the parents of x_i in the graph.

- Also called *graphical models*, *causal networks*
- Concisely express probabilistic models
  - Visualize, design, and motivate new models.
  - Simplify analysis and insight.
- Examples: HMMs, Kalman filters, medical diagnoses.
- Sum-product algorithm provides exact inference for discrete and Gaussian trees.
- Other distributions can be computed by MCMC.

A Bayesian belief network



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

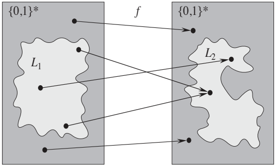
A	P(J|A)
T	.9
F	.05

A	P(M|A)
T	.7
F	.01

Inference in general discrete graphs is NP-complete.

- Definition: *inference* in a belief network means calculating $p(S_1|S_2)$ where S_1 and S_2 are two sets of variables.
  - Example: $p(u_1 = T|Y = F, u_2 = T)$.
- The *most restricted form* of inference is computing $p(Y = T)$.
- If we show this form of inference is NP-hard, then all other forms of inference are also NP-hard. (Why?)

Reduce from 3SAT



- To show decision problem L_2 is NP-complete, find an NP-complete problem L_1.
- Let x be a binary encoding for some input to problem L_1.
- Give a polynomial-time algorithm f to convert all instances of L_1 into an instance of L_2 such that for all strings x, $x \in L_1 \Leftrightarrow f(x) \in L_2$.

The definition of 3SAT

- Let $C = c_1, \dots, c_m$ be a set of clauses on a finite set of U boolean variables
- A *clause* is a disjunction of three literals.
  - Given a variable u, a *literal* is either u or $\neg u$. If $u = T$, then $\neg u = F$.
  - Example clause: $(\neg u_2 \vee u_3 \vee \neg u_4)$.
- C is *satisfiable* $\Leftrightarrow$ there exists some truth assignment for U that satisfies each clause in C.
- 3SAT returns YES if C is satisfiable and NO otherwise.

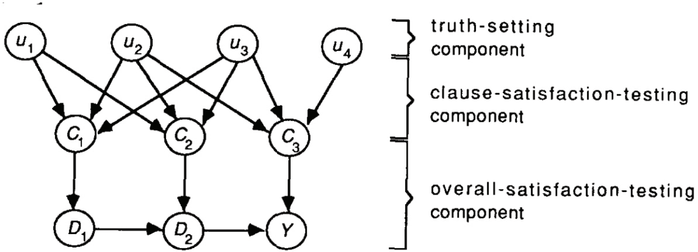
The definition of a Bayesian network

A *Bayesian network* is a tuple (V, A, P) where V is a set of variables (vertices), A is a set of arcs (edges), and P is a set of conditional probability tables, one for each vertex.

Bayesian network inference as decision problem

- Inference (IBN; inference in Bayesian network) is a computation problem. We need a decision problem to do the reduction.
- Define IBND (IBN decision) to
  - Return YES if $P(Y = T) > 0$
  - Return NO otherwise.
- Clearly IBND $\leq$ IBN: if we had a solver for IBN, we simply compare the output to 0 as above to get a solver for IBND.
- *Conversely, if we show IBND is NP-complete, then IBN must also be NP-complete.*

Constructing a Bayesian network from 3SAT, overview



Constructing a Bayesian network from 3SAT

- A *truth setting component* adds one node u_i for each boolean variable in U and sets $p(u_i = T) = \frac{1}{2}$.
- A *clause-satisfaction-testing subcomponent* adds one node C_j for each clause and drawns an arc from edge variable to its clause. $p(C_j = T | \pi_{C_j}) = 1$ if and only if C_j is satisfied.
  - Consider (again) $C_j = (\neg u_2 \vee u_3 \vee \neg u_4)$.
  - $p(C_j = T | u_2 = T, u_3 = F, u_4 = F) = 1$
  - $p(C_j = T | u_2 = T, u_3 = F, u_4 = T) = 0$.
- An *overall-satisfaction-testing component* adds one node D_j for each clause and adds the arcs (C_j, D_j) and (D_{j-1}, D_j). $p(D_j = T | \pi_{D_j}) = 1$ if and only if all of its parents are T.
- Define $D_n = Y$. Then equivalently, $P(Y = T | C_1, \ldots, C_m) = 1$ if and only if each $C_j = T$. *For the rest of the proof, we ignore the* D_j.

C is satisfiable $\Rightarrow p(Y = T) > 0$, part 1

- We want $P(Y)$, but the graph defines $P(Y, C_1, \ldots, C_m, u_1, \ldots, u_n)$
- So define α and β to be binary strings representing particular instantiations of $u_1, \ldots, u_n$ and $C_1, \ldots, C_m$.
  - E.g. $\alpha = 5 = 0101 \Rightarrow u_1 = F, u_2 = T, u_3 = F, u_4 = T$.
  - Enumerating α and β enumerates each assignment of the variables.
- For notation, define $U_\alpha = \{u_i = \alpha_i | 1 \leq i \leq n\}$ and C_β similarly.
- We can then marginalize the C_j and the u_i:

$$p(Y = T) = \sum_{\alpha=0}^{2^n-1} \sum_{\beta=0}^{2^m-1} p(Y = T | C_\beta) p(C_\beta | U_\alpha) p(U_\alpha)$$

C is satisfiable $\Rightarrow p(Y = T) > 0$, part 2

- If C is satisfiable, then there exists a satisfying assignment U_s.
- This results in clause assignment C_1, i.e. $c_i = T$ for $1 \leq i \leq m$, and appears a term in the sum, so

$$p(Y = T) \geq p(Y = T | C_1) p(C_1 | U_s) p(U_s).$$

- Show each term is > 0:
  - Since C_1 satisfies each clause, $p(Y = T | C_1) = 1$.
  - By construction, $p(U_s) = \left(\frac{1}{2}\right)^n$.
  - Since U_s satisfies each clause, $p(C_s | U_s) = 1$ by construction.
- Thus $p(Y = T) > 0$. $\square$

$p(Y = T) > 0 \Rightarrow C$ is satisfiable

- Prove the contrapositive: suppose C is not satisfiable.
- Then for any truth assignment U_α, some clause c_j is not satisfied. Thus $p(C_1 = T|U_\alpha) = 0$, so p
- So any nonzero term in the sum must have $\beta \neq 1$.
- But if $\beta \neq 1$, we have $p(Y = T|C_\beta) = 0$.
- So each term is 0, so $p(Y = T) = 0$.

The form of our constructed network provides strong corollaries

- Each node has indegree no larger than 3 (conditioned on more than 3 variables) $\Rightarrow$ even graphs of simple topology are intractable.
- Planar 3SAT is NP-hard, so even planar bipartite belief networks are NP-hard.
- *Arguably, our construction relies on probabilities of 0 or 1 in the C_j and D_j, so is a pathological case. Real-world cases are unlikely to resemble this reduction.*