Lecture 19: 26 March, 2024

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
January–April 2024

Conditional probabilities

- Boolean variables $x_1, x_2, \ldots, x_n$

Conditional probabilities

- Boolean variables $x_1, x_2, \ldots, x_n$
- Joint probabilities $P(v_1, v_2, \ldots, v_n)$
  - 2^n combinations of $x_1, x_2, \ldots, x_n$
  - $2^n - 1$ parameters

Conditional probabilities

- Boolean variables $x_1, x_2, \ldots, x_n$
- Joint probabilities $P(v_1, v_2, \ldots, v_n)$
  - 2^n combinations of $x_1, x_2, \ldots, x_n$
  - $2^n - 1$ parameters
- Naïve Bayes assumption — complete independence
  - $P(x_i = 1)$ for each x_i
  - n parameters

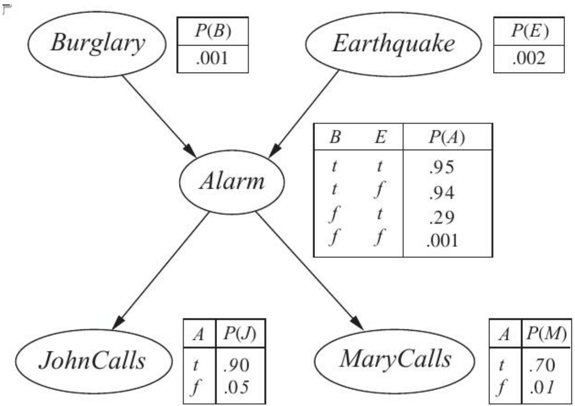
Conditional probabilities

- Boolean variables $x_1, x_2, \ldots, x_n$
- Joint probabilities $P(v_1, v_2, \ldots, v_n)$
  - 2^n combinations of $x_1, x_2, \ldots, x_n$
  - $2^n - 1$ parameters
- Naïve Bayes assumption — complete independence
  - $P(x_i = 1)$ for each x_i
  - n parameters
- Can we strive for something in between?
  - "Local" dependencies between some variables

Probabilistic graphical models — Judea Pearl, Turing Award 2011

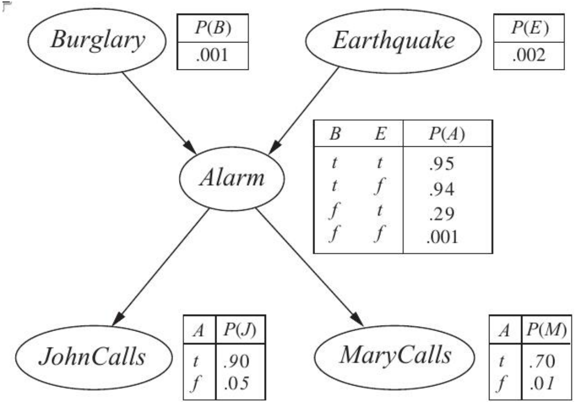
- Represent local dependencies using directed graph
- Each node has a local (conditional) probability table

- Represent local dependencies using directed graph
- Each node has a local (conditional) probability table
- Example: Burglar alarm
  - Pearl's house has a burglar alarm
  - Neighbours John and Mary call if they hear the alarm



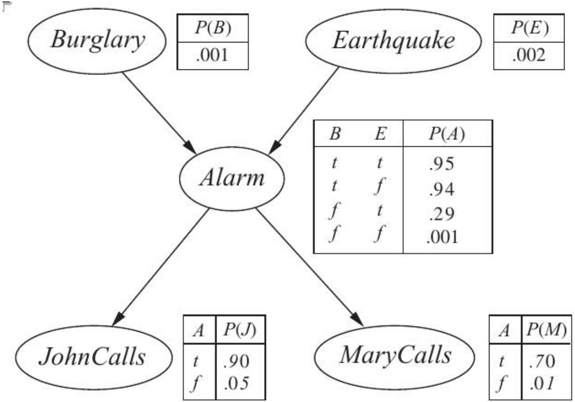
Probabilistic graphical models — Judea Pearl, Turing Award 2011

- Represent local dependencies using directed graph
- Each node has a local (conditional) probability table
- Example: Burglar alarm
  - Pearl's house has a burglar alarm
  - Neighbours John and Mary call if they hear the alarm
  - John is prone to mistaking ambulances etc for the alarm



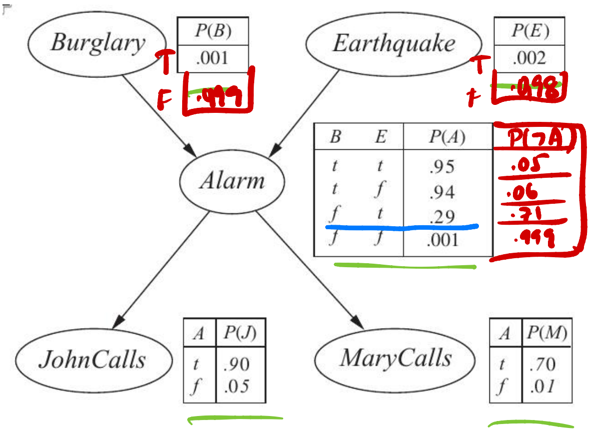
Probabilistic graphical models — Judea Pearl, Turing Award 2011

- Represent local dependencies using directed graph
- Each node has a local (conditional) probability table
- Example: Burglar alarm
  - Pearl's house has a burglar alarm
  - Neighbours John and Mary call if they hear the alarm
  - John is prone to mistaking ambulances etc for the alarm
  - Mary listens to loud music and sometimes fails to hear the alarm



Probabilistic graphical models — Judea Pearl, Turing Award 2011

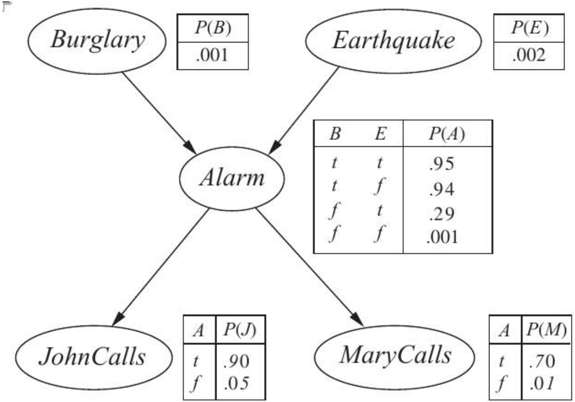
- Represent local dependencies using directed graph
- Each node has a local (conditional) probability table
- Example: Burglar alarm
  - Pearl's house has a burglar alarm
  - Neighbours John and Mary call if they hear the alarm
  - John is prone to mistaking ambulances etc for the alarm
  - Mary listens to loud music and sometimes fails to hear the alarm
  - The alarm may also be triggered by an earthquake (California!)



Probabilistic graphical models

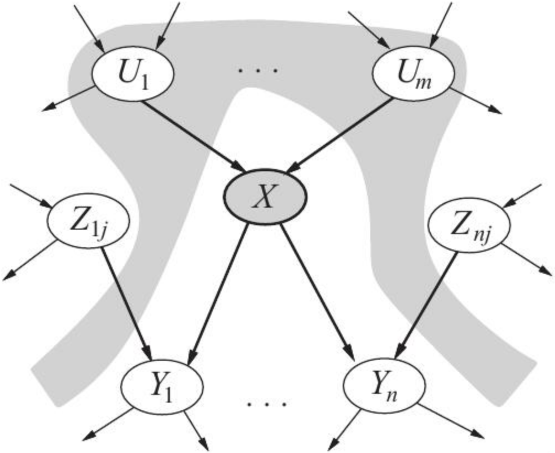
- Graph is a DAG, no cyclic dependencies

"Causality" assumption



Probabilistic graphical models

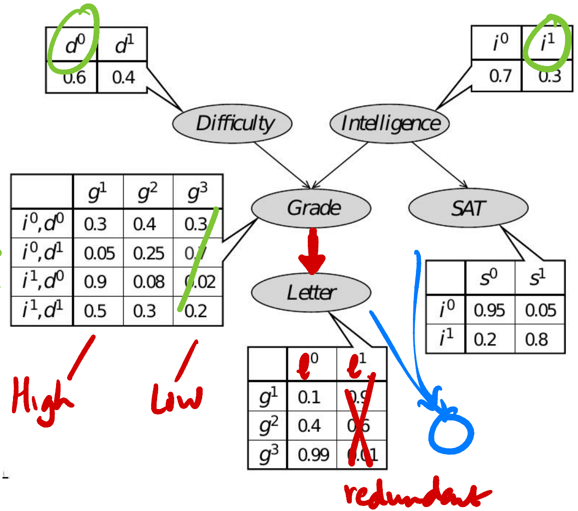
- Graph is a DAG, no cyclic dependencies
- Fundamental assumption:
  A node is conditionally independent of non-descendants, given its parents



Student example

- Example due to Nir Friedman and Daphne Koller
- Student asks teacher for a reference letter
- Teacher has forgotten the student, so letter is entirely based on student's grade in the course

Student gets admission



Evaluating a network

- John and Mary call Pearl. What is the probability that there has been a burglary?

Evaluating a network

- John and Mary call Pearl. What is the probability that there has been a burglary?
- $P(b, m, j)$, where b: burglary, j: John calls, m: Mary calls

Evaluating a network

- John and Mary call Pearl. What is the probability that there has been a burglary?
- $P(b, m, j)$, where b: burglary, j: John calls, m: Mary calls
- $P(b, m, j) = \sum_{a=0}^{1} \sum_{e=0}^{1} P(b, j, m, a, e)$, where a: alarm rings, e: earthquake

Evaluating a network

- John and Mary call Pearl. What is the probability that there has been a burglary?
- $P(b, m, j)$, where b: burglary, j: John calls, m: Mary calls
- $P(b, m, j) = \sum_{a=0}^{1} \sum_{e=0}^{1} P(b, j, m, a, e)$, where a: alarm rings, e: earthquake
- Bayes Rule: $P(A, B) = P(A \mid B)P(B)$

Evaluating a network

- John and Mary call Pearl. What is the probability that there has been a burglary?
- $P(b, m, j)$, where b: burglary, j: John calls, m: Mary calls
- $P(b, m, j) = \sum_{a=0}^{1} \sum_{e=0}^{1} P(b, j, m, a, e)$, where a: alarm rings, e: earthquake
- Bayes Rule: $P(A, B) = P(A \mid B)P(B)$
- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, x_3, \ldots, x_n)\underbrace{P(x_2, x_3, \ldots, x_n)}_{\text{Apply again}}$

$P(b, m, j)$

$P(\neg b, m, j)$

$P(b, m, j)$

$P(b) + P(\neg b)$

Evaluating a network

- John and Mary call Pearl. What is the probability that there has been a burglary?
- $P(b, m, j)$, where b: burglary, j: John calls, m: Mary calls
- $P(b, m, j) = \sum_{a=0}^{1} \sum_{e=0}^{1} P(b, j, m, a, e)$, where a: alarm rings, e: earthquake
- Bayes Rule: $P(A, B) = P(A \mid B)P(B)$
- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, x_3, \ldots, x_n)P(x_2, x_3, \ldots, x_n)$
- Applied recursively, this gives us the chain rule
  $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n)P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n)P(x_n)$

Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$

Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$

Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n)P(x_2 \mid \underline{x_3, \ldots, x_n}) \cdots P(x_{n-1} \mid x_n)P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network

Information linking

$x_3 \ldots x_n$ to x_2

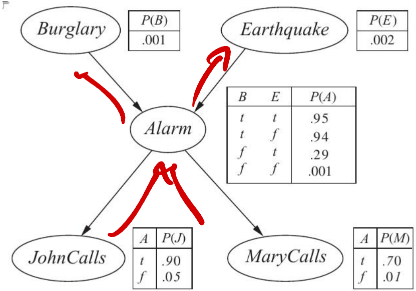
Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network
- $P(m, j, a, b, e) =$
  $P(m \mid j, a, b, e) P(j \mid a, b, e) P(a \mid b, e) P(b \mid e) P(e)$

j, m, a, b, e

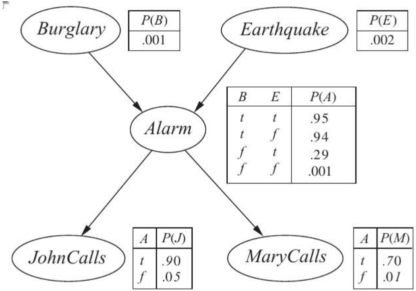
j, m, a, e, b

m, j, a, e, b



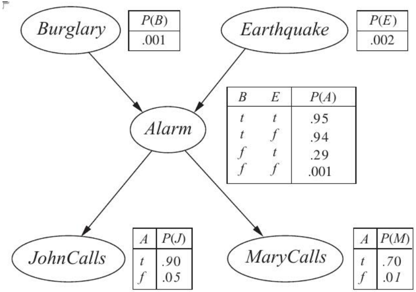
Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network
- $P(m, j, a, b, e) =$
  $P(m \mid \cancel{j}, a, \cancel{b}, \cancel{e}) P(j \mid a, \cancel{b}, \cancel{e}) P(a \mid b, e) P(b \mid \cancel{e}) P(e)$
  $= P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$



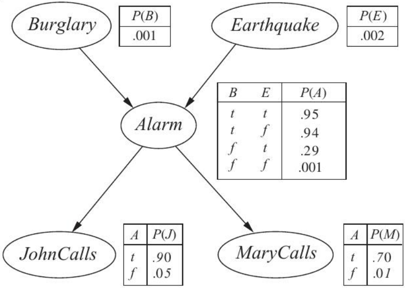
Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network
- $P(m, j, a, b, e) =$
  $P(m \mid j, a, b, e) P(j \mid a, b, e) P(a \mid b, e) P(b \mid e) P(e)$
  $= P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$
- $P(m, j, b) =$
  $$\sum_{a=0}^{1} \sum_{e=0}^{1} P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$$



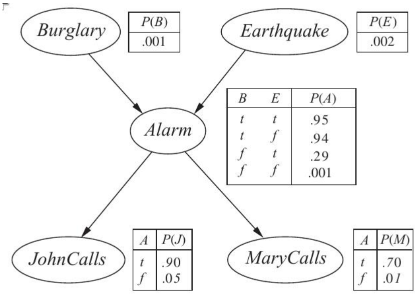
Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network
- $P(m, j, a, b, e) =$
  $P(m \mid j, a, b, e) P(j \mid a, b, e) P(a \mid b, e) P(b \mid e) P(e)$
  $= P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$
- $P(m, j, b) =$
  $\sum_{e=0}^{1} \sum_{a=0}^{1} P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$



Evaluating a network

- $P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, \ldots, x_n) P(x_2 \mid x_3, \ldots, x_n) \cdots P(x_{n-1} \mid x_n) P(x_n)$
- Can choose any ordering of $x_1, x_2, \ldots, x_n$
- Use topological ordering in a Bayesian network
- $P(m, j, a, b, e) =$
  $P(m \mid j, a, b, e) P(j \mid a, b, e) P(a \mid b, e) P(b \mid e) P(e)$
  $= P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$
- $P(m, j, b) =$
  $\displaystyle\sum_{e=0}^{1} \sum_{a=0}^{1} P(m \mid a) P(j \mid a) P(a \mid b, e) P(b) P(e)$
- $P(m, j, b) = P(b) \displaystyle\sum_{e=0}^{1} P(e) \sum_{a=0}^{1} P(m \mid a) P(j \mid a) P(a \mid b, e)$



P(B)
.001

P(E)
.002

B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

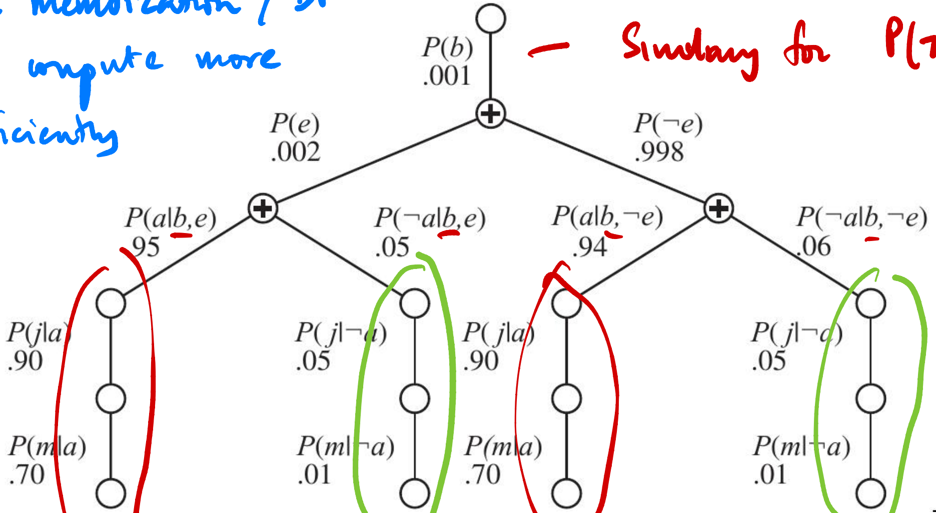
A	P(J)
t	.90
f	.05

A	P(M)
t	.70
f	.01

Evaluation tree

Use memoization / DP to compute more efficiently

— Similarly for $P(\neg b)$

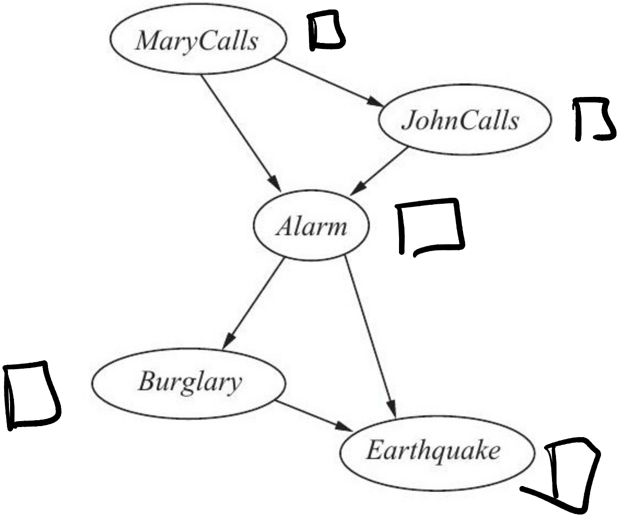


Designing the Bayesian network

- Need to choose node ordering wisely to get a compact Bayesian network

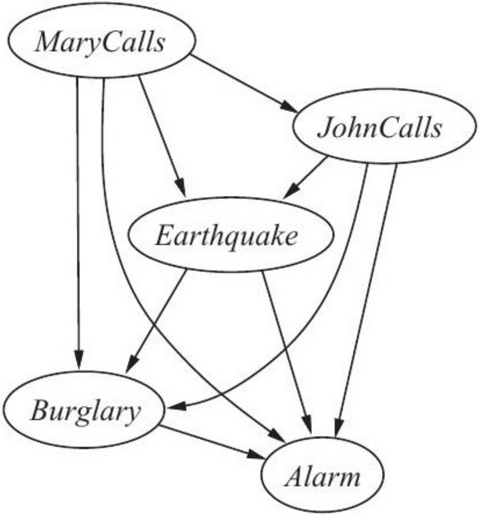
Designing the Bayesian network

- Need to choose node ordering wisely to get a compact Bayesian network
- Ordering *MaryCalls*, *JohnCalls*, *Alarm*, *Burglary*, *Earthquake* produces this network



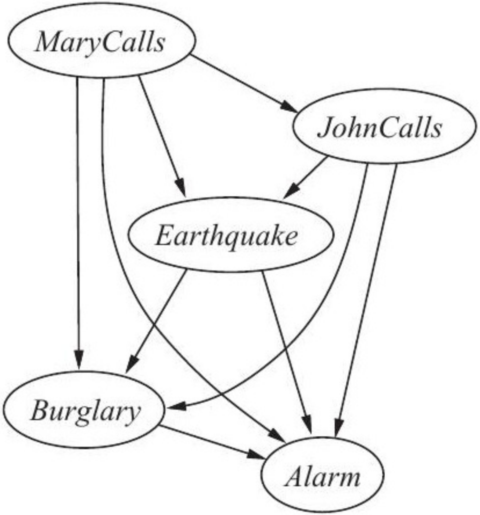
Designing the Bayesian network

- Need to choose node ordering wisely to get a compact Bayesian network
- Ordering *MaryCalls*, *JohnCalls*, *Alarm*, *Burglary*, *Earthquake* produces this network
- Ordering *MaryCalls*, *JohnCalls*, *Earthquake*, *Burglary*, *Alarm* is even worse



Designing the Bayesian network

- Need to choose node ordering wisely to get a compact Bayesian network
- Ordering *MaryCalls*, *JohnCalls*, *Alarm*, *Burglary*, *Earthquake* produces this network
- Ordering *MaryCalls*, *JohnCalls*, *Earthquake*, *Burglary*, *Alarm* is even worse
- Causal model (causes to effects) works better than diagnostic model (effects to causes)



Complexity of exact inference

- Exact inference of Bayesian networks is NP-complete

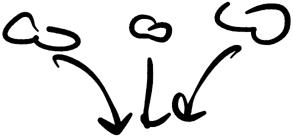
Complexity of exact inference

- Exact inference of Bayesian networks is NP-complete
- Boolean formula in Conjunctive Normal Form (CNF)
  - Boolean variables $\{u_1, u_2, \ldots, u_n\}$
  - A literal ℓ_i is either u_i or $\neg u_i$

Guess & Verify
efficiently

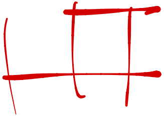
$a_1\ a_2$
$\times\ b_1\ b_2$

Bin packing



An NP Complete problem

My Problem



p, q, r - -   Boolean

$p \land q$ and

$q \lor r$ or

$\neg(p \land q)$ not (p and q)

$\neg p \land \neg q$, $\neg p \lor q$

Complexity of exact inference

- Exact inference of Bayesian networks is NP-complete
- Boolean formula in Conjunctive Normal Form (CNF)
  - Boolean variables $\{u_1, u_2, \ldots, u_n\}$
  - A literal ℓ_i is either u_i or $\neg u_i$
  - A clause is a disjunction of literals $\ell_{j_1} \vee \ell_{j_2} \vee \cdots \vee \ell_{j_k}$

Complexity of exact inference

- Exact inference of Bayesian networks is NP-complete
- Boolean formula in Conjunctive Normal Form (CNF)
  - Boolean variables $\{u_1, u_2, \ldots, u_n\}$
  - A literal ℓ_i is either u_i or $\neg u_i$
  - A clause is a disjunction of literals $\ell_{j_1} \vee \ell_{j_2} \vee \cdots \vee \ell_{j_k}$
  - A CNF formula is a conjunction of clauses $C_1 \wedge C_2 \wedge \cdots \wedge C_m$

$\ell_1 \vee \ell_2 \vee - \ell_k$ $\qquad$ $\ell_1' \vee \ell_2' \vee - \ell_m'$

Complexity of exact inference

- Exact inference of Bayesian networks is NP-complete
- Boolean formula in Conjunctive Normal Form (CNF)
  - Boolean variables $\{u_1, u_2, \ldots, u_n\}$
  - A literal ℓ_i is either u_i or $\neg u_i$
  - A clause is a disjunction of literals $\ell_{j_1} \vee \ell_{j_2} \vee \cdots \vee \ell_{j_k}$
  - A CNF formula is a conjunction of clauses $C_1 \wedge C_2 \wedge \cdots \wedge C_m$
- SAT — given a formula in CNF, is there an assignment to variables that makes the formula true?
- 3-SAT — SAT where each clause has exactly 3 literals

Complexity of exact inference

- Exact inference of Bayesian networks is NP-complete
- Boolean formula in Conjunctive Normal Form (CNF)
  - Boolean variables $\{u_1, u_2, \ldots, u_n\}$
  - A literal ℓ_i is either u_i or $\neg u_i$
  - A clause is a disjunction of literals $\ell_{j_1} \vee \ell_{j_2} \vee \cdots \vee \ell_{j_k}$
  - A CNF formula is a conjunction of clauses $C_1 \wedge C_2 \wedge \cdots \wedge C_m$
- SAT — given a formula in CNF, is there an assignment to variables that makes the formula true?
- 3-SAT — SAT where each clause has exactly 3 literals
- Both SAT and 3-SAT are NP-complete
  - No known efficient algorithm — try all possible valuations

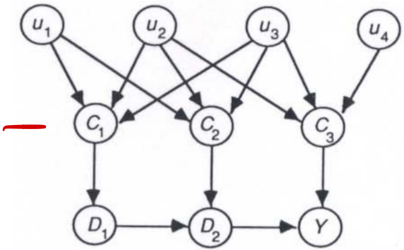
$(x \vee y \vee z)$
$\wedge\ (\neg x \vee w \vee p)$

Reducing 3-SAT to exact inference

- Convert a 3-CNF formula into a Bayesian network

Variables —

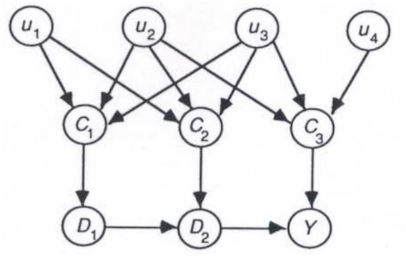
Clauses —



$(C_1 \wedge C_2) \wedge C_3$

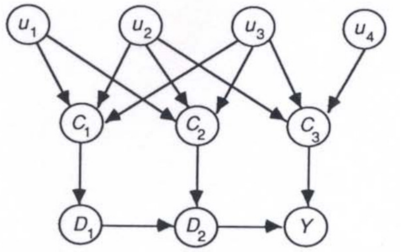
Reducing 3-SAT to exact inference

- Convert a 3-CNF formula into a Bayesian network
- Top layer: one node for each variable u_i

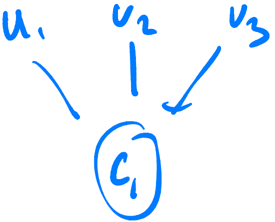


Reducing 3-SAT to exact inference

- Convert a 3-CNF formula into a Bayesian network
- Top layer: one node for each variable u_i
- Middle layer: one node for each clause C_j
  - Parents are three variables whose literals are in C_j
  - Conditional probability table for C_j has 8 rows, for all possible valuations of 3 variables
  - $P(C_j = 1) = 0$ for row where each input literal is false, $P(C_j = 1) = 1$ for remaining 7 rows



D_1	C_2	D_2
0	0	0
0	1	0
1	0	0
1	1	1



$$\begin{array}{ccc} 0 & 1 & 1 \\ \end{array}$$

$$(u_1 \lor \neg u_2 \lor \neg u_3)$$

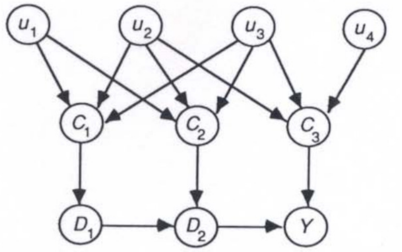
✓ ✓

C_1

u_1	u_2	u_2	$P(C_1=T)$
0	0	0	1
			⋮
0	1	1	0
			⋮

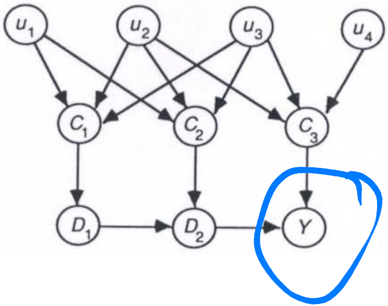
Reducing 3-SAT to exact inference

- Convert a 3-CNF formula into a Bayesian network
- Top layer: one node for each variable u_i
- Middle layer: one node for each clause C_j
  - Parents are three variables whose literals are in C_j
  - Conditional probability table for C_j has 8 rows, for all possible valuations of 3 variables
  - $P(C_j = 1) = 0$ for row where each input literal is false, $P(C_j = 1) = 1$ for remaining 7 rows
- Bottom row builds up $C_1 \wedge \cdots \wedge C_m$ one clause at a time



Reducing 3-SAT to exact inference

- Convert a 3-CNF formula into a Bayesian network
- Top layer: one node for each variable u_i
- Middle layer: one node for each clause C_j
  - Parents are three variables whose literals are in C_j
  - Conditional probability table for C_j has 8 rows, for all possible valuations of 3 variables
  - $P(C_j = 1) = 0$ for row where each input literal is false, $P(C_j = 1) = 1$ for remaining 7 rows
- Bottom row builds up $C_1 \wedge \cdots \wedge C_m$ one clause at a time
- $P(Y = 1) > 0$ iff original 3-CNF formula is satisfiable



Reducing 3-SAT to exact inference

- Convert a 3-CNF formula into a Bayesian network
- Top layer: one node for each variable u_i
- Middle layer: one node for each clause C_j
  - Parents are three variables whose literals are in C_j
  - Conditional probability table for C_j has 8 rows, for all possible valuations of 3 variables
  - $P(C_j = 1) = 0$ for row where each input literal is false, $P(C_j = 1) = 1$ for remaining 7 rows
- Bottom row builds up $C_1 \wedge \cdots \wedge C_m$ one clause at a time
- $P(Y = 1) > 0$ iff original 3-CNF formula is satisfiable