Computational Complexity of Inference

Sargur Srihari
srihari@cedar.buffalo.edu

Topics

1. What is Inference?
2. Complexity Classes
3. Exact Inference
    1. Variable Elimination
        - Sum-Product Algorithm
    2. Factor Graphs
    3. Exact Inference for Tree graphs
    4. Exact inference in general graphs

Common BN Inference Problem

- Assume set of variables χ
  - $\boldsymbol{E}$: evidence variables, whose known value is e
  - $\boldsymbol{Y}$: query variables, whose distrib. we wish to know
- Conditional probability query $P(\boldsymbol{Y}|\boldsymbol{E}=e)$

$$\boxed{P(\mathbf{Y} \mid \boldsymbol{E} = e) = \frac{P(\mathbf{Y}, e)}{P(e)}}$$

From product rule

  – Evaluation of Numerator $P(\boldsymbol{Y}, e)$
    - If $\boldsymbol{W}=\chi - \boldsymbol{Y} - \boldsymbol{E}$

$$\boxed{P(\mathbf{y}, e) = \sum_{w} P(\mathbf{y}, e, w)}$$

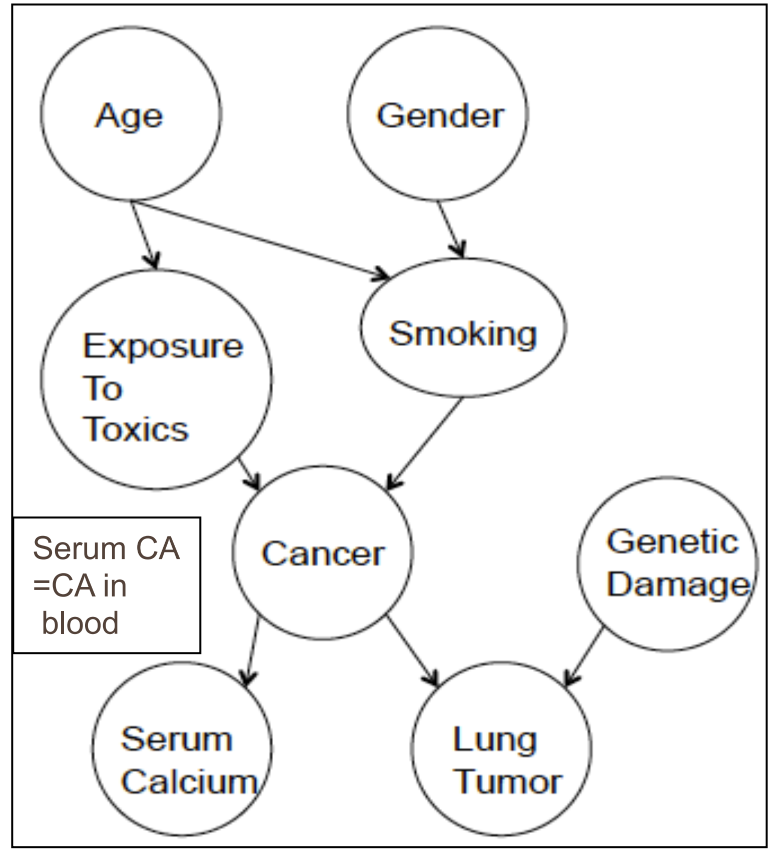
(1) Each term in summation is simply an entry in the distribution

  – Evaluation of Denominator $P(e)$

$$\boxed{P(e) = \sum_{y} P(y, e)}$$

Rather than marginalizing over $P(\boldsymbol{y},, e,, \boldsymbol{w})$ this allows reusing computation of (1)

Ex: Inference with Cancer BN



- Lung Cancer BN is given
  - If Serum Calcium is known, what is $P(C)$?
- We can evaluate it as:

$$\boxed{P(C \mid Se) = \frac{P(C,Se)}{P(Se)}}$$

  - where

$$P(C,Se) = \sum_{A,Ge,E,Sm,L,Gd} P(A,Ge,E,Sm,Se,C,L,Gd)$$

$$= \sum_{A,Ge,E,Sm,L,Gd} P(A)P(Ge)P(Gd)P(E \mid A),P(Sm \mid A,G)P(Se \mid C)P(C \mid E,Sm)P(L \mid C,Gd)$$

$$P(Se) = \sum_{C} P(C,Se)$$

Analysis of Complexity

- Approach of summing out the variables in the joint distribution is unsatisfactory

$$P(y,e) = \sum_{w} P(y,e,w)$$

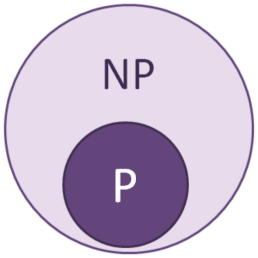
  - Returns us to exponential blow-up
    - PGM was precisely designed to avoid this!
- We now show that problem of inference in PGMs is $\mathcal{NP}$-hard
  - Requires exponential time in the worst case except if $\mathcal{P}=\mathcal{NP}$
  - Even worse, approximate inference is $\mathcal{NP}$- hard
- Discussion for BNs applies to MNs also

What is a decision problem?

- Subset sum decision problem
  - Given a set of integers, is there a subset that sums to zero?
    - No polynomial time algorithm to *decide* this
  - Given an instance, say set $\omega = \{-2, -3, 15, 14, 7, -10\}$
    - The guess $\{-2, -3, -10, 15\}$ can be *verified* in poly time
- Definition of Decision Problem Π:
  - L_Π defines a precise set of instances
    - L_Π consists of integer sets which have a subset that sums to zero
  - Decision problem Π: Is instance ω in L_Π?
    - Is ω a set of integers with a subset that sums to zero?

$\mathcal{P}$ and $\mathcal{NP}$ decision problems

- Definition of Decision Problem Π:
  - L_Π defines a precise set of instances
  - Decision problem: Is instance ω in L_Π ?



- Decision problem Π is in
  - $\mathcal{P}$ if there is an algorithm that decides in poly time
  - $\mathcal{NP}$ if a guess can be verified in poly time
    - Guess is produced non-deterministically
    - Hence the name *non-deterministic polynomial time*
  - Subset sum problem is in $\mathcal{NP}$
    - Whether a given subset sums to zero verified in poly time
    - But not in $\mathcal{P}$
      - No poly algorithm to determine whether there exists any subset that sums to zero

3-SAT (Satisfiability) decision problem

- 3-SAT formula over binary variables $q_1,..,q_n$
  - has the form $C_1 \wedge C_2 \wedge ..C_m$

    C_i is a clause of form $l_{i,1} \vee l_{i,2} \vee l_{i,3}$; $l_{i,j}$ $i=1,..,m;\ j=1,2,3$ are literals which are either q_k or $\sim q_k$

    e.g., with $n{=}3$ $m{=}2$ and clauses $(q_1 \vee \sim q_2 \vee \sim q_3) \wedge (\sim q_1 \vee q_2 \vee \sim q_3)$

    there is a satisfying assignment (assigns true to formula): $q_1{=}q_2{=}q_3{=}$ true

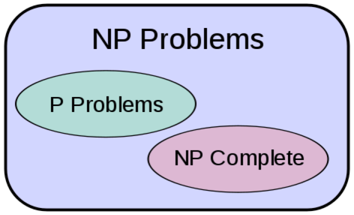
    with clauses $(\sim q_1 \vee q_2) \wedge (q_2 \vee q_3) \wedge (\sim q_1 \vee q_3)$ there is no satisfying assignment

    Each assignment verified in polynomial time

- Decision problem Π: Given a 3-SAT formula of size n, is there a satisfying assignment?
  - To answer this we need to check n binary variables with 2^n assignments
    - L_Π is the set of 3-SAT formulas that have a satisfying assignment

What is $\mathcal{P}$=$\mathcal{NP}$?

- Input is a formula of size n
  - A particular assignment γ can be verified in polynomial time, e.g., $q_1 = q_2 = q_3 = \text{true}$
  - Suppose generate guess γ and verify if it satisfies
    - Since guess verified in polynomial time, decision problem Π is in $\mathcal{NP}$
- Deterministic problems are subset of nondeterministic ones. So $P \subseteq NP$.



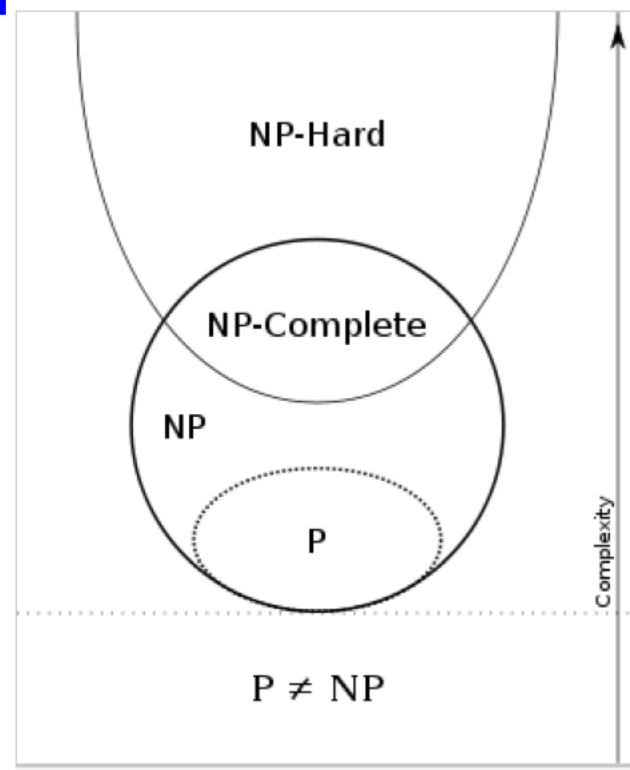
  - Converse is biggest problem in complexity
    - If you can verify in polynomial time, can you decide in polynomial time?
      - Eg., is there a prime greater than n?

$\mathcal{P}=\mathcal{NP}$ intuition

- *Creating* great art
  - Complexity is $\mathcal{NP}$
- *Appreciating* art:
  - Complexity is $\mathcal{P}$
- Is $\mathcal{P}=\mathcal{NP}$?
  - Most mathematicians don't think so

$\mathcal{NP}$-hard and $\mathcal{NP}$-complete

- Hardest problems in $\mathcal{NP}$ are called $\mathcal{NP}$-complete
  - If poly time solution exists, can solve any in $\mathcal{NP}$
    - $\mathcal{NP}$-hard problems need not have polynomial time verification
- If Π is $\mathcal{NP}$-hard it can be transformed into Π' in $\mathcal{NP}$
- 3-SAT is $\mathcal{NP}$-complete



BN for 3-SAT

- Propositional variables $q_1,..,q_n$
  - Return *true* if $C_1 \wedge C_2 \wedge ..C_m$, where C_i is a DNF of 3 binary variables q_k , has a satisfying assignment,
    - e.g., return true for 3-SAT formula $(q_1 \vee \sim q_2 \vee \sim q_3) \wedge (\sim q_1 \vee q_2 \vee \sim q_3)$ since $q_1=q_2=q_3=\text{true}$ is a satisfying assignment and return false for $(\sim q_1 \vee q_2 \vee \sim q_3) \wedge (q_2 \vee q_3) \wedge (\sim q_1 \vee q_3)$ which has no satisfying assignments

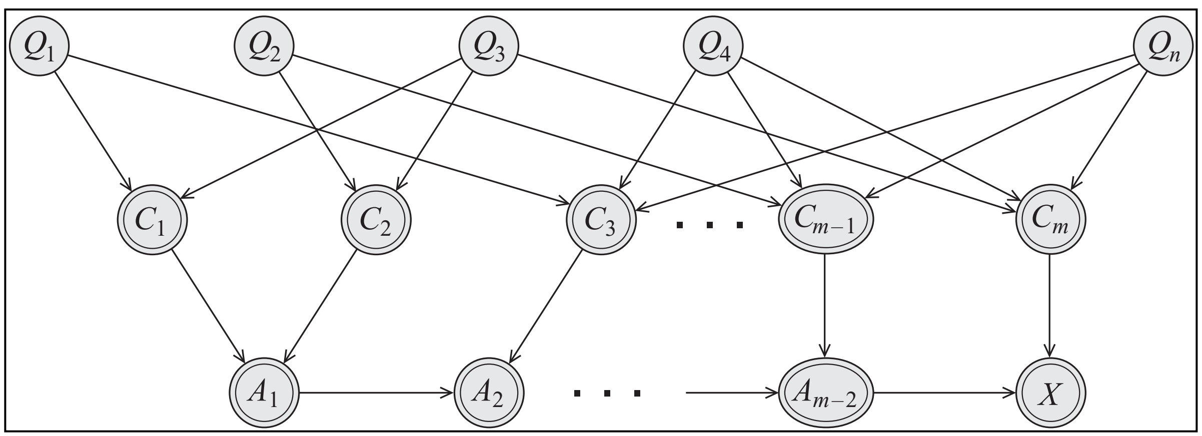
BN to infer this:

$P(q_k^1)=0.5$

C_i are deterministic OR

A_i are deterministic AND

X is output (has value 1 iff all of the C_i's are 1



$\#\mathcal{P}$-complete Problem

- Counting the no. of satisfying assignments
  - E.g., Propositional variables $q_1,..,q_n$

    Return true if $C_1 \wedge C_2 \wedge ..C_m$,

    where C_i is a DNF of 3 binary variables q_k , has a satisfying assignment,

Analysis of Exact Inference

- Worst case: CPD is a table of size

  $|Val(\{X_i\} \vee Pa_{X_i})|$

- Most analyses of complexity are stated as decision-problems
  - Consider decision problem first, then numerical one
- Natural version of conditional probability task:
  - *BN-Pr-DP*: Bayesian Network Decision Problem
    - Given a BN $\mathcal{B}$ over χ, a variable $X \in \chi$, and a value $x \in Val(X)$ decide $P_B(X=x)>0$
  - This decision problem can be shown to be $\mathcal{NP}$-complete

Proof of *BN-Pr-DP* is $\mathcal{NP}$-complete

- Whether in $\mathcal{NP}$:
  - Guess assignment ξ to network variables. Check whether $X=x$ and $P(\xi)>0$
  - One such guess succeeds iff $P(X=x)>0$.
  - Done in linear time
- Is $\mathcal{NP}$-hard:
  - Answer for instances in BN-Pr-DP can be used to answer an $\mathcal{NP}$-hard problem
  - Show a reduction from 3-SAT problem

Reduction of 3-SAT to BN inference

- Given a 3-SAT formula ϕ create BN $\mathcal{B}_\phi$ with variable X such that ϕ is satisfiable iff $P_{\mathcal{B}\phi}(X=x_1)>0$
- If BN inference is solved in poly time we can also solve 3-SAT in poly time

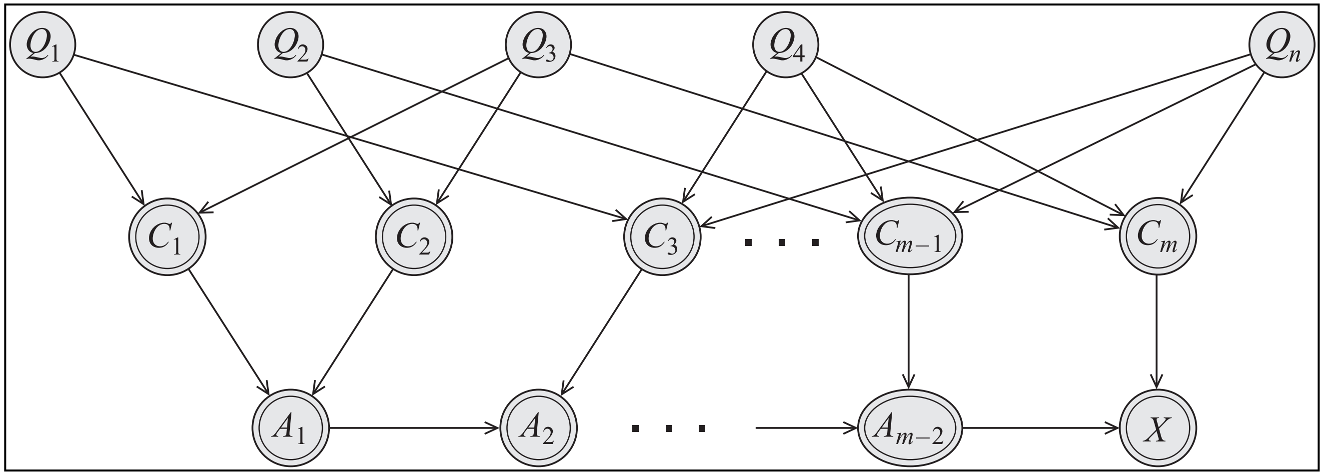
BN to infer this:

$P(q_k^1)=0.5$

C_i are deterministic OR

A_i are deterministic AND

X is output



Original Inference Problem

$$\boxed{p(y) = \sum_x p(y / x)p(x)}$$

- It is a numerical problem
  - rather than a decision problem
- *Define BN-Pr*
  - Given a BN $\mathcal{B}$ over χ, a variable $X \varepsilon \chi$, and a value $x \varepsilon Val(X)$ compute $P_{\mathcal{B}}(X=x)$
  - Task is to compute the total probability of instantiations that are consistent with $X=x$
    - Weighted count of instantiations, with weight being the probability
    - This problem is $\#\mathcal{P}$-complete

Analysis of Approximate Inference

- Metrics for quality of approximation
- Absolute Error
  - Estimate ρ has *absolute error* ε for $P(\boldsymbol{y}|\boldsymbol{e})$ if

    $$|P(\boldsymbol{y}|\boldsymbol{e})-\rho| \leq \varepsilon$$

    - A weak definition of error. If a rare disease has probability 0.0001 then error of 0.0001 is unacceptable. If the probability is 0.3 then error of 0.0001 is fine
- Relative Error
  - Estimate ρ has *relative error* ε for $P(\boldsymbol{y}|\boldsymbol{e})$ if

    $$\rho/(1+\varepsilon) \leq P(\boldsymbol{y}|\boldsymbol{e}) \leq \rho(1+\varepsilon)$$

    - ε=4 means $P(\boldsymbol{y}|\boldsymbol{e})$ is at least 20% of ρ and at most 600% of ρ. For low values much better than absolute error

Approximate Inference is NP-hard

- The following problem is $\mathcal{NP}$-hard
  - Given a BN $\mathcal{B}$ over χ, a variable $X \varepsilon \chi$ and a value $x\ \varepsilon\ Val(X)$, find a number ρ that has relative error ε for $P_{\mathcal{B}}(X=x)$
- Proof:
  - It is NP-hard to decide if $P_{\mathcal{B}}(x^1) > 0$
  - Assume algorithm returns estimate ρ to $P_{\mathcal{B}}(x^1)$ which has relative error ε for some $\varepsilon > 0$
  - $\rho > 0$ if and only if $P_{\mathcal{B}}(x^1) > 0$
  - This achieving relative error is $\mathcal{NP}$-hard

Inference Algorithms

- Worst case is exponential
- Two types of inference algorithms
  - Exact
    - Variable Elimination
    - Clique trees
  - Approximate
    - Optimization
      - Propagation with approximate messages
      - Variational (analytical approximations)
    - Particle-based (sampling)