Exact Inference: Complexity

Sargur Srihari
srihari@cedar.buffalo.edu

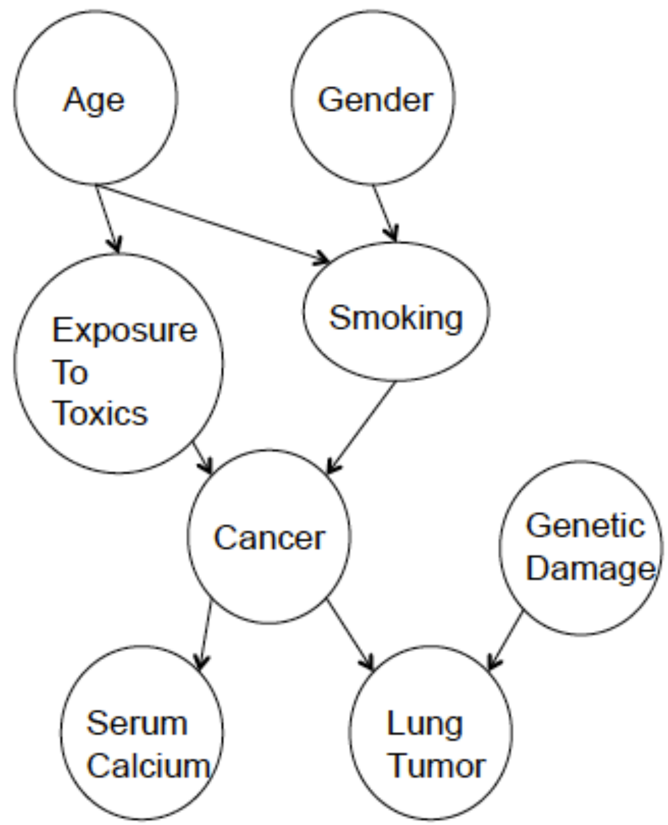
Topics

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

An example of inference

- Serum Calcium definition: calcium in blood
- If Serum Calcium is known, what is the probability of cancer?

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



Types of inference

- Graphical models represent a joint probability distribution $\rightarrow$ $\boxed{p(x,y)}$
- Types of inference tasks:
  1. Compute marginal probabilities $\rightarrow$ $\boxed{p(y) = \sum_x p(y/x)p(x)}$
     1. Conditional probabilities can be easily computed from joint and marginals $\rightarrow$ $\boxed{p(x/y) = \frac{p(x,y)}{p(y)}}$
  2. Evaluate posterior distributions
     1. Some of the nodes in a graph are clamped to observed values (X)
     2. Compute posterior distributions of one or more subsets of nodes (latent variables Z), *i.e.*, $p(Z|X)$
  3. Compute maximum a posteriori probabilities $\rightarrow$ $\max_y p(y/x)$

Common BN Inference Problem

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

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

From product rule

– Evaluation of Numerator $P(\boldsymbol{Y},e)$

- If $\boldsymbol{W}=\chi\text{-}\boldsymbol{Y}\text{-}\boldsymbol{E}$

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

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

– Evaluation of Denominator $P(e)$

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

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

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

P and NP decision problems

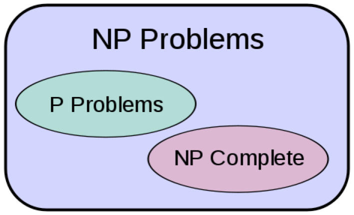
- Definition of Decision Problem Π:
  - L_{Π} defines a precise set of instances
  - Decision problem: Is instance ω in L_{Π}?
- Decision problem Π is in
  - $\mathcal{P}$ if there is algorithm decides in poly time
  - $\mathcal{NP}$ if a guess can be verified in poly time
    - Guess is produced non-deterministically
- E.g., subset sum problem in P but not NP
  - Does a subset of integers sum to zero?
  - Subset sum of {−2, −3, 15, 14, 7, −10} add up to 0?
    - Yes {−2, −3, −10, 15}, but not in P

3-SAT (Satisfiability) decision problem

- 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 $q_1=q_2=q_3=true$ 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
- SAT problem: whether there exists a satisfying assignment
  - To answer this we need to check n binary variables there are 2^n assignments

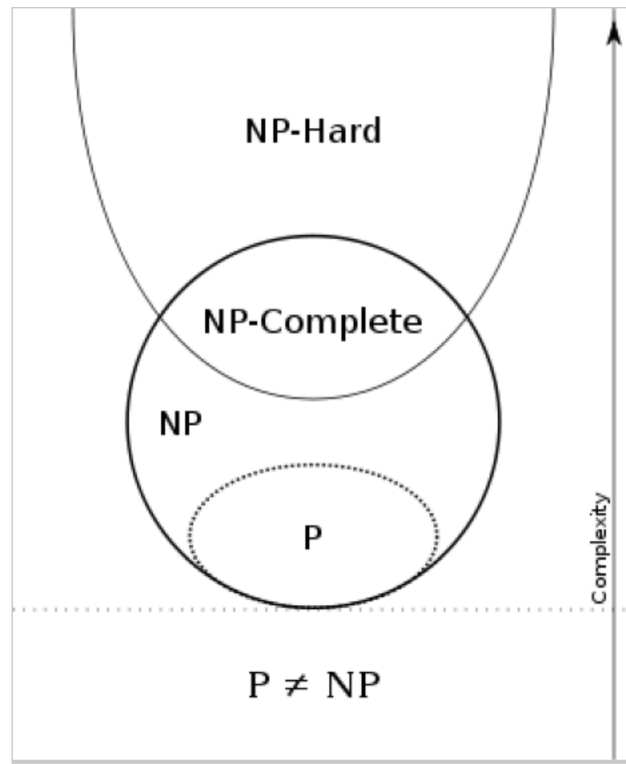
What is P=NP?

- Decision problem of whether ω in $L_{3\text{-}SAT}$
  - Can be verified in polynomial time P
  - Another solution is to generate guesses γ that satisfy $L_{3\text{-}SAT}$ and verify if one is ω
  - If guess verified in polynomial time, in 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?



NP-Hard and NP-complete

- Hardest problems in NP are called NP-complete
  - If poly time solution exists, can solve any in NP
- NP-hard problems need not have polynomial time verification
- If Π is NP-hard it can be transformed into Π' in $\mathcal{NP}$
- 3-SAT is 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 $q_1=q_2=q_3=true$ 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

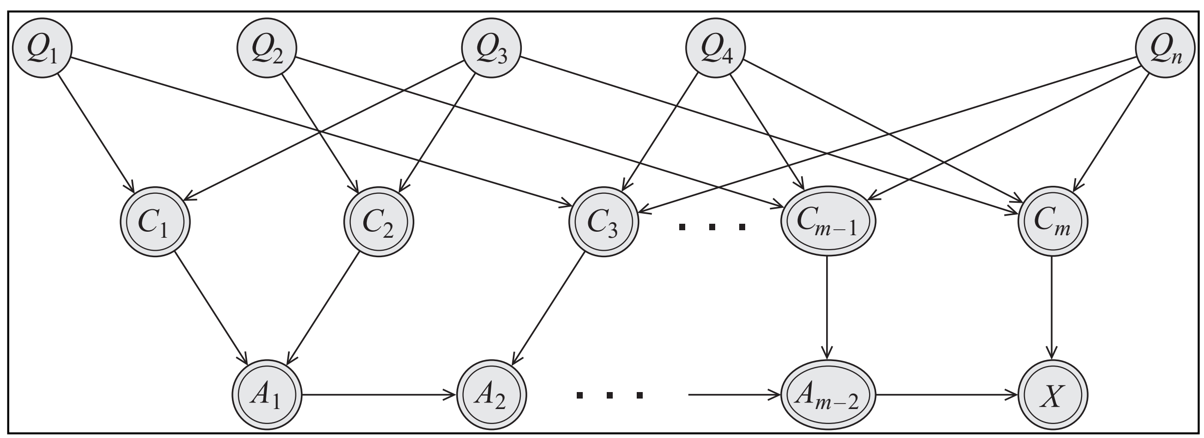
BN to infer this:

$P(q_k^1)=0.5$

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



#P-complete Problem

- Counting the number 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 $|Val\{X_i\} \cup Pa_{X_i}\}|$
- 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 $X \varepsilon \chi$, and a value $x \varepsilon Val(X)$ decide $P_{\mathcal{B}}(X=x)>0$
  - This decision problem can be shown to be NP-complete

Proof of *BN-Pr-DP* is NP-complete

- Whether in 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 NP-hard:
  - Answer for instances in BN-Pr-DP can be used to answer an 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 B_ϕ with variable X such that ϕ is satisfiable iff $P_{B\phi}(X=x_1)>0$
- If BN inference is solved in poly time we can also solve 3-SAT in poly time

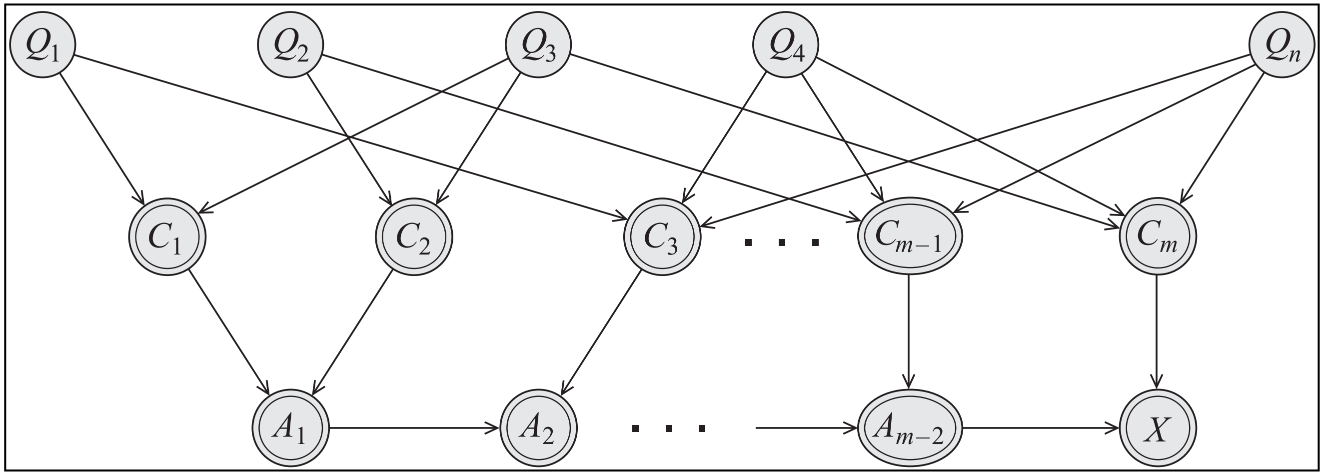
BN to infer this:

$P(q_k^1)=0.5$

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 #P-complete

Analysis of Approximate Inference

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

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

    - 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 error ε for $P(\mathbf{y}|\mathbf{e})$ if

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

    - $\varepsilon=4$ means $P(y|e)$ 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 NP-hard
- Given a BN B over χ, a variable $X \epsilon \chi$ and a value $x \epsilon Val(X)$, find a number ρ that has relative error ε for $P_B(X=x)$
- Proof:
  - It is NP-hard to decide if $P_B(x^1) > 0$
  - Assume algorithm returns estimate ρ to $P_B(x^1)$ which has relative error ε for some $\varepsilon > 0$
  - $\rho > 0$ if and only if $P_B(x^1) > 0$
  - This achieving relative error is 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)