CS 440/ECE448 Lecture 20: Bayes Net Inference & Learning

Slides by Svetlana Lazebnik, 11/2016

Modified by Mark Hasegawa-Johnson, 11/2017

Bayes Network Inference & Learning

Bayes net is a **memory-efficient model** of dependencies among:

- Query *variables:* **X**
- *Evidence* (*observed*) variables and their values: **E** = **e**
- *Unobserved* variables: **Y**

Inference problem: answer questions about the query variables given the evidence variables

- This can be done using the posterior distribution P(**X** | **E** = **e**)
- The posterior can be derived from the full joint P(**X**, **E**, **Y**)
- How do we make this **computationally efficient?**

Learning problem: given some training examples, how do we learn the parameters of the model?

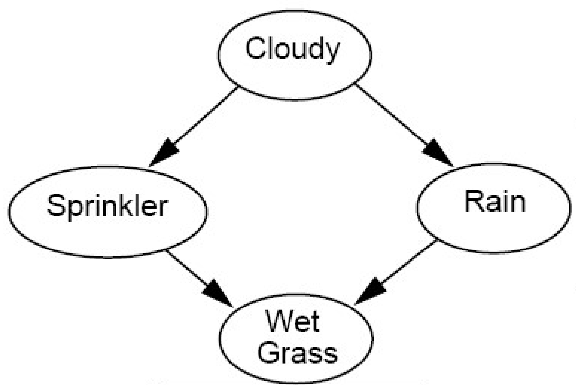
- Parameters = p(variable|parents), for each variable in the net

Outline

- Inference Examples
- Inference Algorithms
  - Trees: Sum-product algorithm
  - Graphs: No polynomial-time algorithm
- Parameter Learning
  - Expectation Maximization
- Structure Learning
  - Regularized Maximum Likelihood

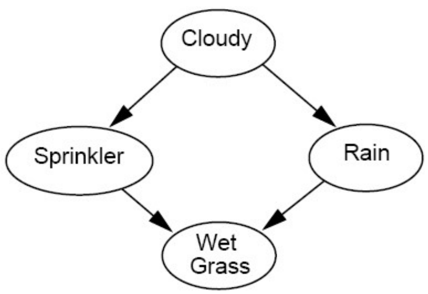
Practice example 1

- Variables: *Cloudy, Sprinkler, Rain, Wet Grass*



Practice example 1

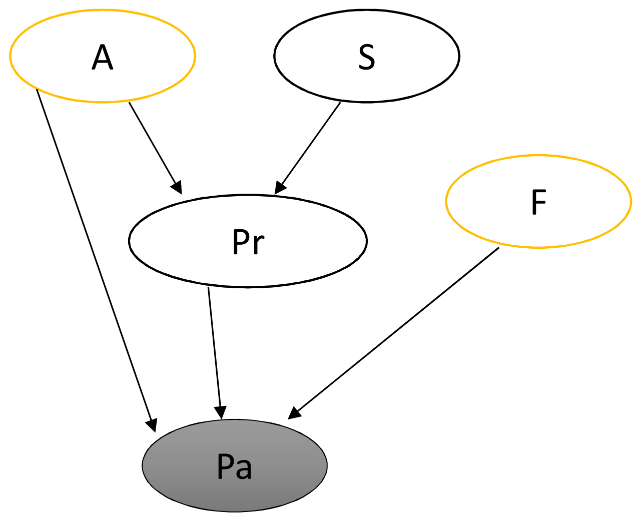
- Given that the grass is wet, what is the probability that it has rained?



$$P(r \mid w)$$

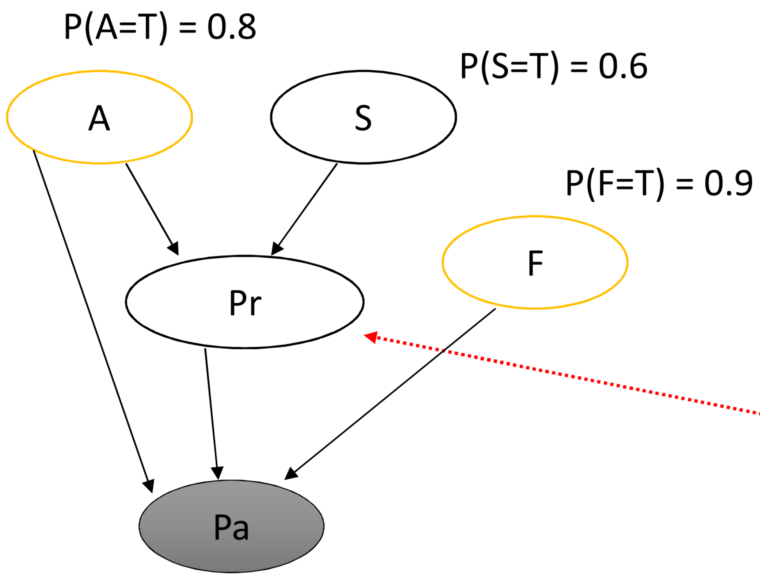
Practice example 2

- What determines whether you will pass the exam?
  - **A**: Do you attend class?
  - **S**: Do you study?
  - **Pr**: Are you prepared for the exam?
  - **F**: Is the grading fair?
  - **Pa**: Do you get a passing grade on the exam?



Source: UMBC CMSC 671, Tamara Berg

Practice example 2

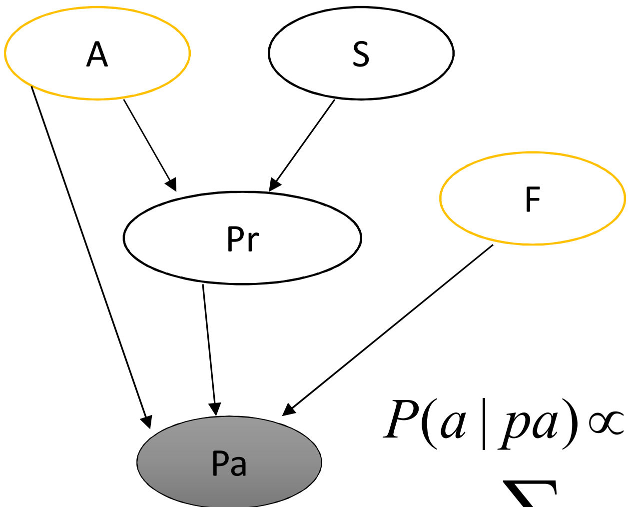


A	S	P(Pr|A,S)
T	T	0.9
T	F	0.5
F	T	0.7
F	F	0.1

Pr	A	F	P(Pa|A,Pr,F)
T	T	T	0.9
T	T	F	0.6
T	F	T	0.2
T	F	F	0.1
F	T	T	0.4
F	T	F	0.2
F	F	T	0.1
F	F	F	0.2

Source: UMBC CMSC 671, Tamara Berg

Practice example 2



Query: What is the probability that a student attended class, given that they passed the exam?

$$P(a \mid pa) \propto P(a, pa)$$

$$= \sum_{S=s, F=f, Pr=pr} P(a, s, f, pr, pa)$$

$$= \sum_{S=s, F=f, Pr=pr} P(a)P(s)P(f)P(pr \mid a, s)P(pa \mid a, pr, f)$$

Source: UMBC CMSC 671, Tamara Berg

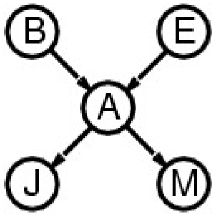
Outline

- Inference Examples
- Inference Algorithms
  - Trees: Sum-product algorithm
  - Graphs: No polynomial-time algorithm
- Parameter Learning
  - Expectation Maximization
- Structure Learning
  - Regularized Maximum Likelihood

Efficient exact inference

- Key idea: compute the results of sub-expressions in a bottom-up way and cache them for later use
  - Form of dynamic programming
  - Polynomial time and space complexity for *polytrees*: networks with at most one undirected path between any two nodes

Sum-Product Algorithm for Trees



Topologically sort the variables

- E.g., {B,E,A,J,M}

then…

for each variable:

for every possible setting of its parents:

(1) **<u>sum</u>** over every possible setting of the parent's parents:

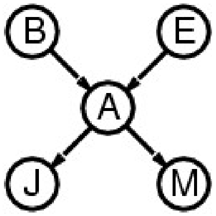
$$P(parents) = \sum P(parents, grandparents)$$

(2) the family is the **<u>product</u>** of the parents and the child

P(variable,parents)=P(variable|parents)P(parents)

Practice example 3

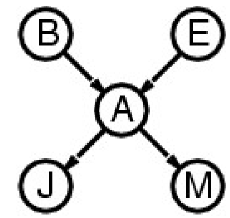
- Query: P(B=True | J=True, M=True)



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

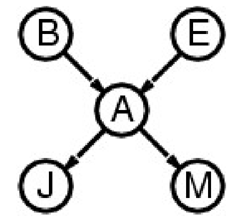
- Can we compute this sum efficiently?

Practice example 3: Sum-product algorithm



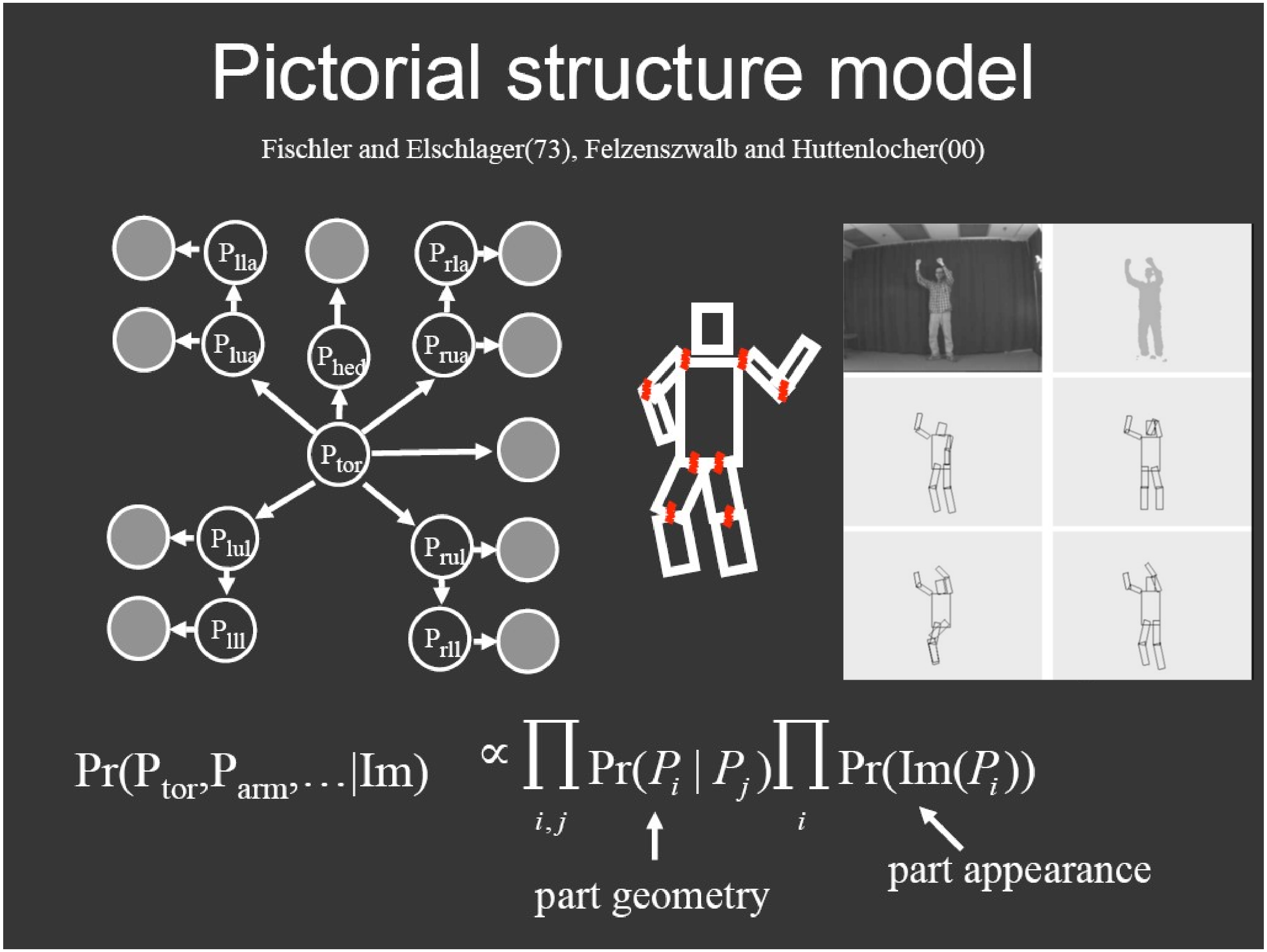
- To compute p(query variable|evidence variables) you need to find p(query,evidence) for every value of the query, but only for the given values of the evidence.
  - In this case the query variable is B (two possible settings: T or F)
  - The evidence variable is the pair (J,M) (we only need to consider one setting: both T)
- Find a path through the polytree, including all variables
  - In this case, the path (B,E,A,J,M) will work.
  - We'll consider all possible settings of (B,E,A), but only the given settings of (J=T,M=T)
- We will need to use the sum-product algorithm to propagate probabilities along that path, in order to finally compute p(B=b,J=T,M=T).

Practice example 3: Sum-product algorithm



1. A-Product: $P(A = a, B = b, E = e) = P(a|b, e)P(b)P(e)$
-- compute that for each of the 8 possible combinations of (a, b, e).
2. A-Sum: $P(A = a, B = b) = \sum_e P(a, b, e)$
-- 2-way summation for each of the 4 possible settings of (a, b).
3. (J,M)-Product: $P(J = T, M = T, A = a, B = b) =$
$$P(M = T|a)P(J = T|a)P(a, b)$$
-- 4 possible settings of (a, b)
4. (J,M)-Sum: $P(J = T, M = T, B = b) = \sum_a P(J = T, M = T, a, b)$
-- 2-way summation, 2 possible settings of (b).
5. Divide: $P(B = T|J = T, M = T) = \frac{P(B=T, J=T, M=T)}{\sum_b P(B=b, J=T, M=T)}$

Re


Pictorial structure model
Fischler and Elschlager(73), Felzenszwalb and Huttenlocher(00)
P_{lla}
P_{rla}
P_{lua}
P_{hed}
P_{rua}
P_{tor}
P_{lul}
P_{rul}
P_{lll}
P_{rll}
$\Pr(P_{tor}, P_{arm}, \ldots | \mathrm{Im}) \propto \prod_{i,j} \Pr(P_i \mid P_j) \prod_i \Pr(\mathrm{Im}(P_i))$
part geometry
part appearance

Outline

- Inference Examples
- Inference Algorithms
  - Trees: Sum-product algorithm
  - Graphs: No polynomial-time algorithm
- Parameter Learning
  - Expectation Maximization
- Structure Learning
  - Regularized Maximum Likelihood

Bayesian network inference

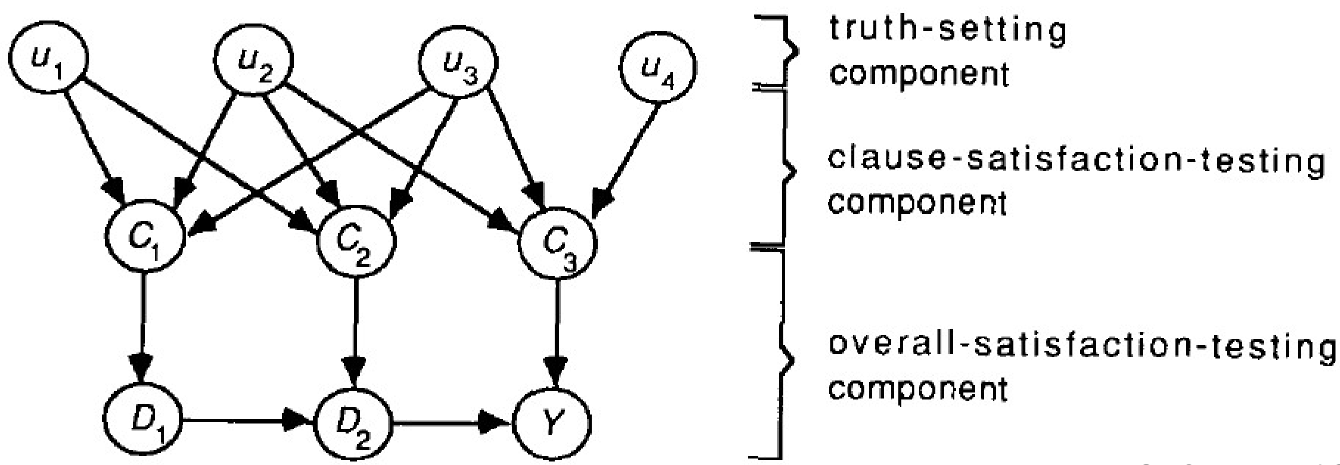
- In full generality, NP-hard
  - More precisely, #P-hard: equivalent to counting satisfying assignments
- We can reduce satisfiability to Bayesian network inference
  - Decision problem: is P(Y) > 0?

$$Y = (U_1 \vee U_2 \vee U_3) \wedge (\neg U_1 \vee \neg U_2 \vee U_3) \wedge (U_2 \vee \neg U_3 \vee U_4)$$

Bayesian network inference

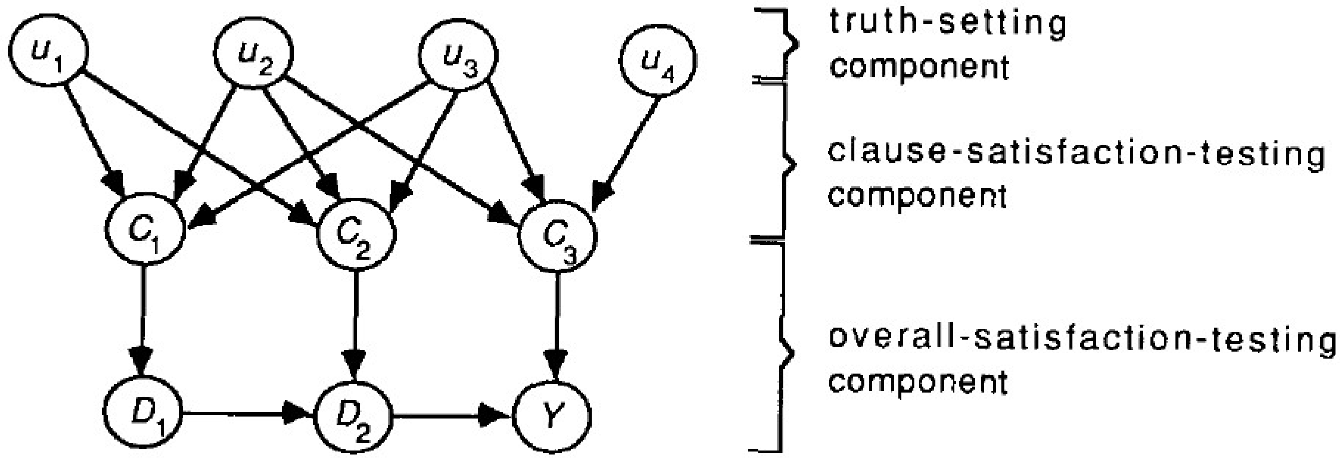
- In full generality, NP-hard
  - More precisely, #P-hard: equivalent to counting satisfying assignments
- We can reduce satisfiability to Bayesian network inference
  - Decision problem: is P(Y) > 0?

$$Y = \underbrace{(U_1 \vee U_2 \vee U_3)}_{C_1} \wedge \underbrace{(\neg U_1 \vee \neg U_2 \vee U_3)}_{C_2} \wedge \underbrace{(U_2 \vee \neg U_3 \vee U_4)}_{C_3}$$



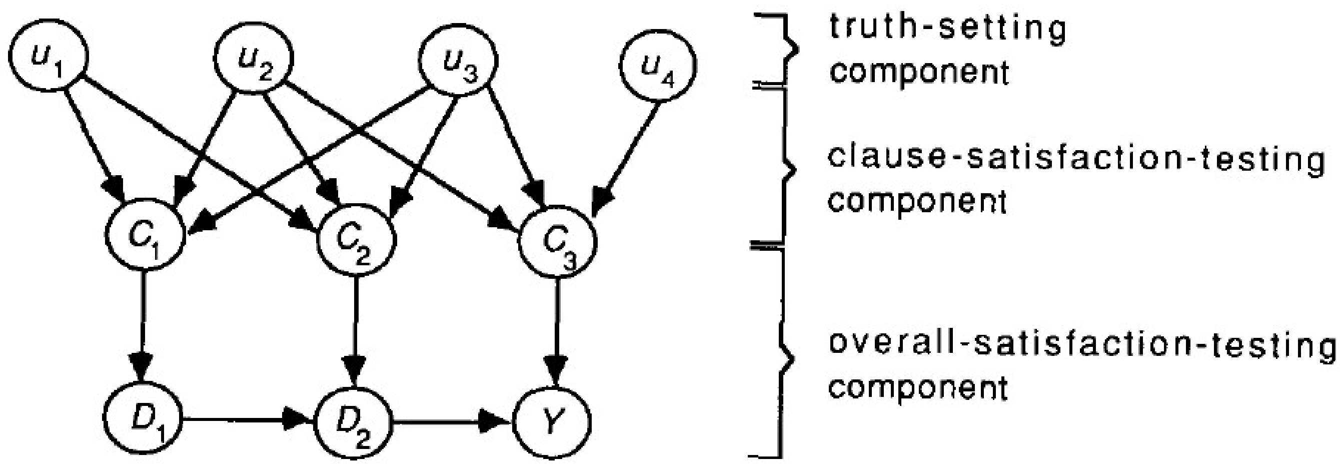
G. Cooper, 1990

Bayesian network inference



$$P(U_1, U_2, U_3, U_4, C_1, C_2, C_3, D_1, D_2, Y) =$$
$$P(U_1)P(U_2)P(U_3)P(U_4)$$
$$P(C_1 \mid U_1, U_2, U_3)P(C_2 \mid U_1, U_2, U_3)P(C_3 \mid U_2, U_3, U_4)$$
$$P(D_1 \mid C_1)P(D_2 \mid D_1, C_2)P(Y \mid D_2, C_3)$$

Bayesian network inference



Why can't we use the sum-product algorithm to efficiently compute Pr(Y)?

Bayesian network inference: Big picture

- Polytrees: exact inference is O{number of variables × (number of settings of each variable)^k}
  - Order of the polynomial (k) depends on the number of variables in each clique, e.g., in each parent-child group
- Non-polytree graphs: Exact inference is intractable
- Approximate inference (not covered)
  - Sampling, variational methods, message passing / belief propagation…

Outline

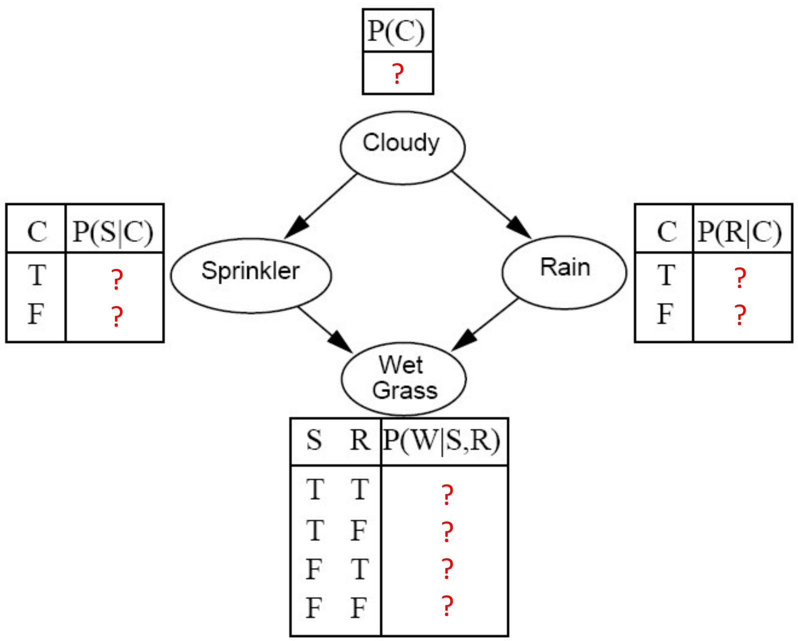
- Inference Examples
- Inference Algorithms
  - Trees: Sum-product algorithm
  - Graphs: No polynomial-time algorithm
- Parameter Learning
  - Expectation Maximization
- Structure Learning
  - Regularized Maximum Likelihood

Parameter learning

- **Inference problem**: given values of evidence variables $\mathbf{E} = \mathbf{e}$, answer questions about query *variables* $\mathbf{X}$ using the posterior $P(\mathbf{X} \mid \mathbf{E} = \mathbf{e})$
- **Learning problem:** estimate the parameters of the probabilistic model $P(\mathbf{X} \mid \mathbf{E})$ given a *training sample* $\{(\mathbf{x}_1,\mathbf{e}_1), \ldots, (\mathbf{x}_n,\mathbf{e}_n)\}$

Parameter learning

- Suppose we know the network structure (but not the parameters), and have a training set of *complete* observations



Training set

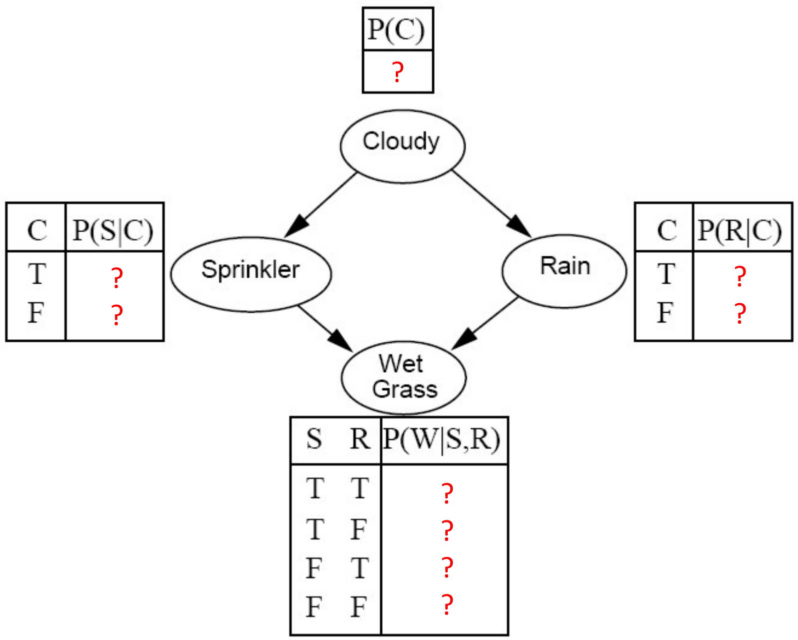
Sample	C	S	R	W
1	T	F	T	T
2	F	T	F	T
3	T	F	F	F
4	T	T	T	T
5	F	T	F	T
6	T	F	T	F
…	…	…	….	…

Parameter learning

- Suppose we know the network structure (but not the parameters), and have a training set of *complete* observations
  - P(X | Parents(X)) is given by the observed frequencies of the different values of X for each combination of parent values

Parameter learning

- Incomplete observations



Training set

Sample	C	S	R	W
1	?	F	T	T
2	?	T	F	T
3	?	F	F	F
4	?	T	T	T
5	?	T	F	T
6	?	F	T	F
…	…	…	….	…

with missing data

Parameter learning: EM

Sample	C	S	R	W
1	?	F	T	T
2	?	T	F	T
3	?	F	F	F
4	?	T	T	T
5	?	T	F	T
6	?	F	T	F
…	…	…	….	…

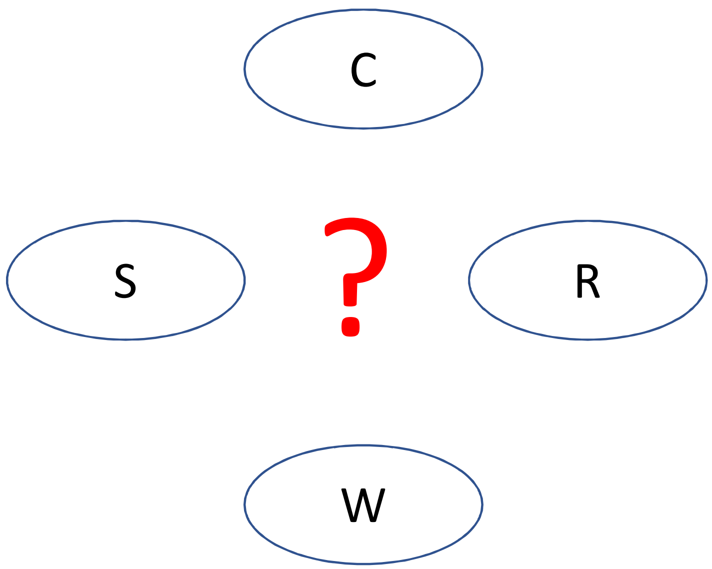
- E-STEP:
  - Find $P(C_n|S_n, R_n, W_n)$ for 1<=n<=6
  - Find $\mathrm{E}[\#\text{ times } C_n = T, S_n = T]$ by adding up the values of $P(C_n|S_n, R_n, W_n)$
- M-STEP:
  - Re-estimate the parameters as $P(C_n = T|S_n = T) \leftarrow E[\#\ times\ C_n = T, S_n = T]/E[\#times\ S_n = T]$

Outline

- Inference Examples
- Inference Algorithms
  - Trees: Sum-product algorithm
  - Graphs: No polynomial-time algorithm
- Parameter Learning
  - Expectation Maximization
- Structure Learning
  - Regularized Maximum Likelihood

Structure learning

- What if the network structure is unknown?

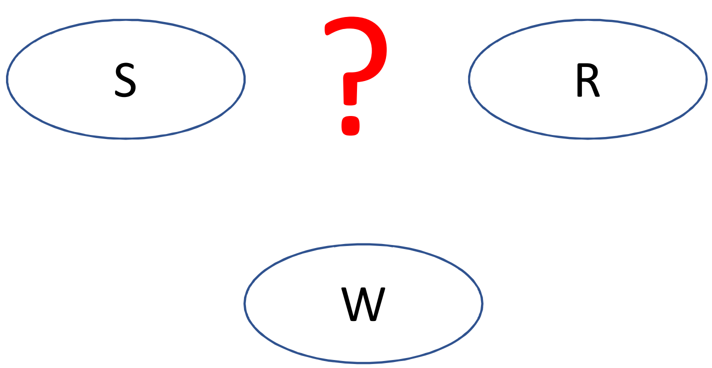


Training set

Sample	C	S	R	W
1	T	F	T	T
2	F	T	F	T
3	T	F	F	F
4	T	T	T	T
5	F	T	F	T
6	T	F	T	F
…	…	…	….	…

Structure learning

1. For every possible set of edges:
   a) Compute the data likelihood given this structure
   b) Penalize complexity (e.g., subtract # edges)
2. Find the structure with highest penalized likelihood



Sample	C	S	R	W
1	T	F	T	T
2	F	T	F	T
3	T	F	F	F
4	T	T	T	T
5	F	T	F	T
6	T	F	T	F
...	...	...	....	...

Summary: Bayesian networks

- Structure
- Parameters
- Inference
- Learning