Lecture 5: Exact inference

- Queries
- Inference in chains
- Variable elimination
  - Without evidence
  - With evidence
- Complexity of variable elimination

Queries

Bayesian networks can answer questions about the underlying probability distribution:

- Likelihood: what is the probability of a given value assignment for a subset of variables Y?
- Conditional probability query: what is the probability of different value assignments for query variables Y given evidence about variables Z? I.e. compute $P(Y|Z=z)$
- Most probable evidence (MPE): given evidence $Z=z$, find an instantiation of all other variables in the network, $W=X-Z$, which has the highest probability:

$$MPE(W|Z=z) = \arg\max_w P(W=w|Z=z)$$

Queries (continued)

Maximum a posteriori (MAP) query: given evidence $Z = z$, and given a subset of variables Y, find the most likely assignment of values to the variables in Y given that $Z = z$:

$$MAP(Y|Z = z) = \arg\max_{y} P(Y = y|Z = z)$$

Examples of MAP queries:

- In speech recognition, given a speech signal, one can attempt to reconstruct the most likely sequence of words that could have generated the signal.
- In classification, given the training data and a new example, we want to determine the most probable class label of the new example.

Complexity of inference

Given a Bayesian network and a random variable X, deciding whether $P(X = x) > 0$ is NP-hard (see Friedman and Koller's notes for details).

- This implies that there is no general inference procedure that will work efficiently for all network configurations
- But for particular families of networks, inference can be done efficiently.

Likelihood inference in simple chains

Consider a simple chain of nodes:

$$A \to B \to C \to D$$

How do we compute $P(B)$?

$$P(B) = \sum_a P(A=a)P(B|A=a)$$

All the numbers required are in the CPTs. If A has k possible values and B has m possible values, this requires $O(km)$ operations: k multiplications and $k-1$ additions for each of the m values of B.

Inference in simple chains (2)

$$A \rightarrow B \rightarrow C \rightarrow D$$

Now how do we compute $P(C)$?

$$\begin{aligned} P(C) &= \sum_b P(B=b)(C|B=b) \\ &= \sum_b \left(\sum_a P(A=a)P(B=b|A=a) \right) P(C|B=b) \end{aligned}$$

We use $P(B)$, which is already computed, and the local CPT of node C.

Inference in simple chains (3)

$$X_1 \to X_2 \to \ldots \to X_n$$

How do we compute $P(X_n)$?

We compute $P(X_2), \ldots P(X_n)$ iteratively. Each step only takes $O(|X_i| \cdot |X_{i+1}|)$ operations (where $|X|$ is the number of possible values of variable X), and the algorithm is linear in the number of variables.

If we would have generated the whole joint distribution and summed out, we would have needed $O((\max_i |X_i|)^n)$ operations!

Elimination of variables in chains

Let us examine the chain example again: $A \rightarrow B \rightarrow C \rightarrow D$.
Suppose we want to compute $P(D)$:

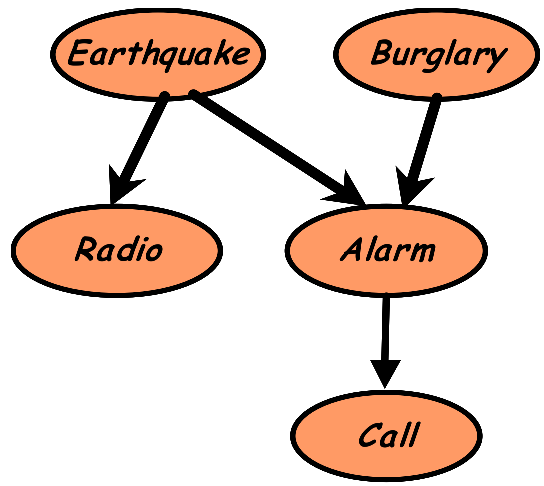
$$\begin{aligned}
P(D) &= \sum_{A,B,C} P(A,B,C,D) \\
&= \sum_{A,B,C} P(A)P(B|A)P(C|B)P(D|C) \\
&= \sum_{C} P(D|C) \sum_{B} P(C|B) \sum_{A} P(B|A)P(A)
\end{aligned}$$

The innermost summation depends only on the value of B. So we can compute a *factor* $f_1(B)$, with one entry for each value of B. Then we can use this to compute a factor $f_2(C)$ etc.

This is a form of dynamic programming

Pooling

Consider the case when a node has more than one parent, e.g.:

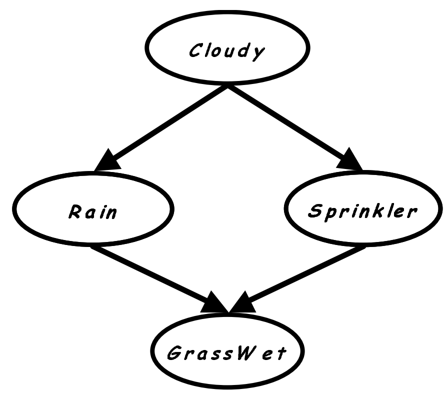


How do we compute $P(C)$?

$$P(C) = \sum_{A} P(C|A)P(A) = \sum_{A} P(C|A) \sum_{E,B} P(A|B,E)P(E)P(B)$$

A Bayes network is called a **polytree** if the underlying undirected graph is a tree.

What if the network is not a polytree?



Suppose we want to compute $P(W)$.

$$\begin{aligned} P(W) &= \sum_{R,S,C} P(W,R,S,C) = \sum_{R,S,C} P(W|R,S)P(R|C)P(S|C)P(C) \\ &= \sum_{R,S} P(W|R,S) \sum_{C} P(R|C)P(S|C)P(C) \end{aligned}$$

Note that in this case we have a more complex factor, which depends on two variables.

Variable elimination without evidence

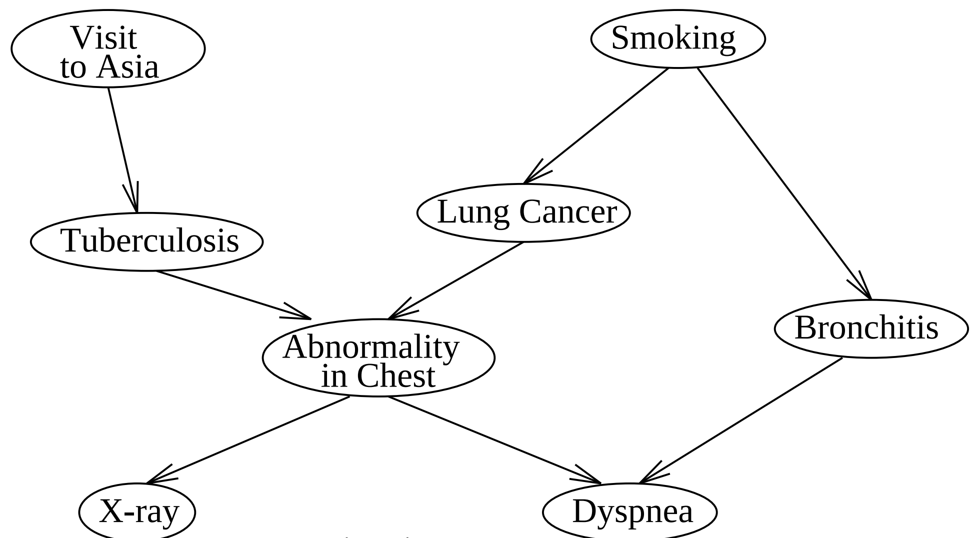
Given: A Bayes network and a set of query variables $Y_1, \dots Y_k$

1. Initialize the set of factors: $F = \{P(X_i | Parents(X_i))\}, \forall i$.
2. Let $\{Z_1 \dots Z_m\} = \{X_1, \dots X_m\} - \{Y_1 \dots Y_k\}$
3. For $i = 1 \dots m$ do:
   (a) Extract from F all factors $f_1, \dots f_r$ mentioning Z_i
   (b) Let $f' = \prod_{j=1}^{r} f_j$
   (c) Let $f'' = \sum_{Z_i} f'$.
   (d) Insert f'' in F
4. Return $\prod_{f \in F} f$

Steps (a) and (b) *eliminate* variable Z_i; this is where the computations actually take place.

Example: Asia network

This example is taken from Koller and Friedman's notes:



Suppose we want to compute $P(D)$. So $Z = \{V, X, S, T, L, B, A\}$.

Note that we can use any ordering of the variables during elimination

Predictive inference with evidence in chains

Suppose we know that $A = a$. How do we compute $P(C|A = a)$?

$$\begin{aligned} P(C|A=a) &= \frac{P(C, A=a)}{P(A=a)} = \frac{\sum_B P(C, B, A=a)}{P(A=a)} \\ &= \frac{\sum_B P(C|B)P(B|A=a)P(A=a)}{P(A=a)} \\ &= \sum_B P(C|B)P(B|A=a) \end{aligned}$$

Without knowing A, computing $P(C, A)$ required another factor:

$$P(C, A) = \sum_B P(C, B, A) = \sum_B P(C|B) \sum_A P(B|A)P(A)$$

Computing $P(C, A = a)$ requires using $P(B|A = a)P(A = a)$ instead of $\sum_A P(B|A)P(A)$. We eliminated the factor *inconsistent with the evidence*.

Causal inference with evidence in chains

Again the chain example: $A \to B \to C \to D$. Suppose we know that $B = b$. How do we compute $P(A|B = b)$?

We apply Bayes rule:

$$P(A|B = b) = \frac{P(A, B = b)}{P(B = b)}$$

We do not need to compute $P(B = b)$, that comes out of summing the numerators for all values of A.

$P(A, B = b)$ can be computed using Bayes rule:

$P(A, B = b) = P(B = b|A)P(A)$.

This can be viewed as a message passing from B to A.

Causal inference with evidence in chains (2)

$$A \rightarrow B \rightarrow C \rightarrow D$$

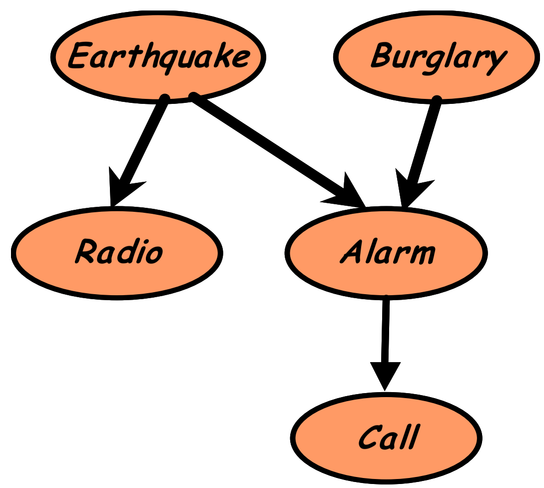
Suppose we know that $C = c$. How do we compute $P(B|C = c)$?

$$P(B|C = c) = \frac{P(B, C = c)}{P(C = c)} = \frac{P(C = c|B)P(B)}{P(C = c)}$$

$P(C = c|B)$ is known from the CPT of node C. $P(B)$ can be computed using forward inference, just like before.

B receives some information from C (backward pass) and some from A (forward pass), and performs the computation.

Inference with evidence in polytrees

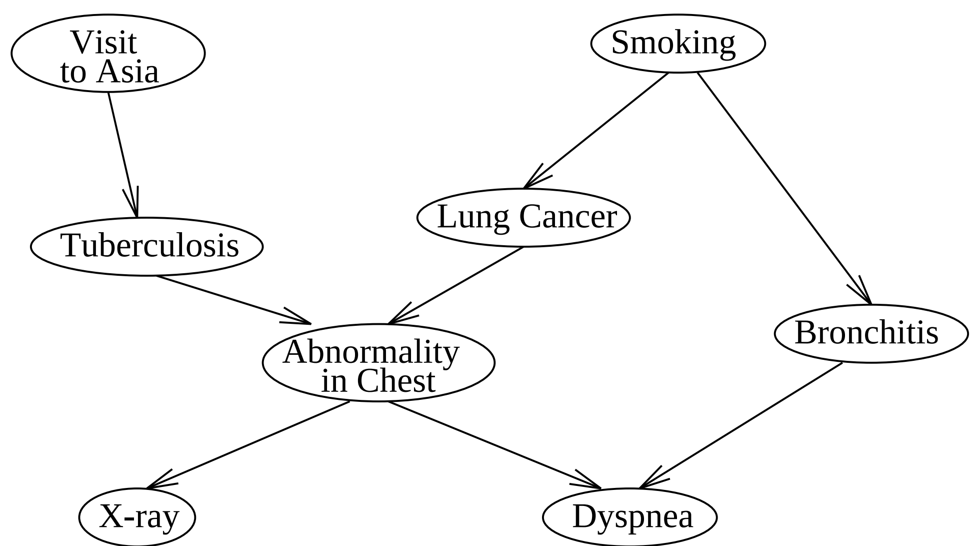


How do we compute $P(E|C=t)$? We need $P(E, C=t)$.

$$\begin{aligned} P(E, C=t) &= \sum_{A,B} P(E, A, C=t, B) \\ &= \sum_{A} P(C=t|A) \sum_{B} P(E)P(B)P(A|E,B) \end{aligned}$$

Example: Asia network

Suppose we observe that the person smokes and that the X-ray comes out negative. How does this change the variable elimination we did before?



First we reduce all factors to eliminate the parts inconsistent with the evidence

Variable elimination with evidence

Given: A Bayes network, a set of query variables $Y_1, \dots Y_k$, and evidence $u_1, \dots u_l$.

1. Initialize the set of factors: $F = \{P(X_i|Parents(X_i))\}, \forall i$.
2. For each factor, if it contains u_i, retain only the appropriate portion (to be consistent with the evidence)
3. Let $\{Z_1 \dots Z_m\} = \{X_1, \dots X_n\} - \{Y_1 \dots Y_k\} - \{U_1 \dots U_l\}$
4. For $i = 1 \dots m$ do:
   (a) Extract form F all factors $f_1, \dots f_r$ mentioning Z_i
   (b) Let $f' = \prod_{j=1}^{r} f_j$
   (c) Let $f'' = \sum_{Z_i} f'$.
   (d) Insert f'' in F
5. Return $\prod_{f \in F} f$

Complexity of variable elimination

- We need at most $O(n)$ multiplications to create one entry in a factor f
- The size of a factor f containing m variables is k^m, where k is the maximum arity of a variable
- We need $O(n)$ additions

So to be efficient, it is important to have **small factors**.

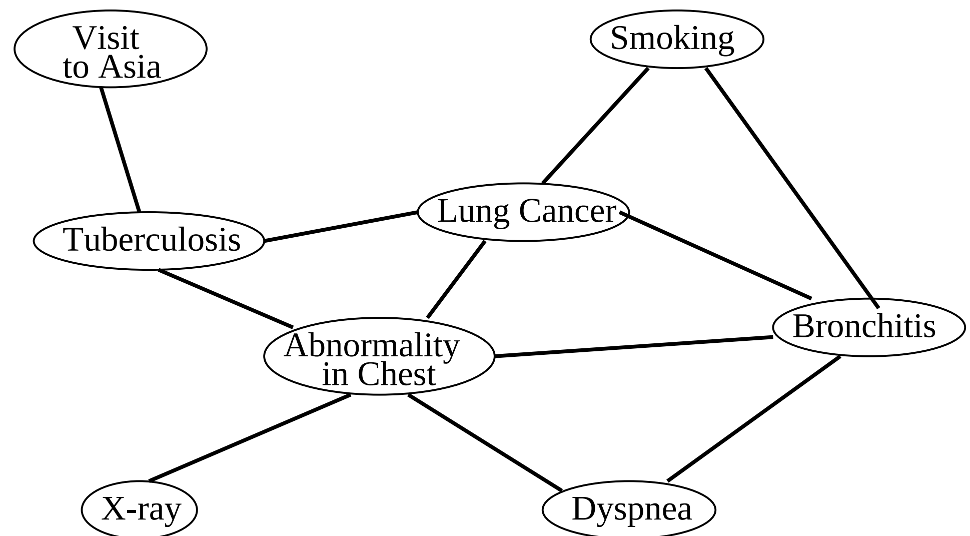
Induced graph

We will try to understand the size of the factors in terms of the graph structure.

Given a Bayes net structure G and an elimination ordering $Y_1, \ldots Y_k$, the **induced graph** H is an undirected graph over $X_1 \ldots X_n$ where X_i and X_j are connected by an edge if they both appear in an intermediate factor f generated by variable elimination.

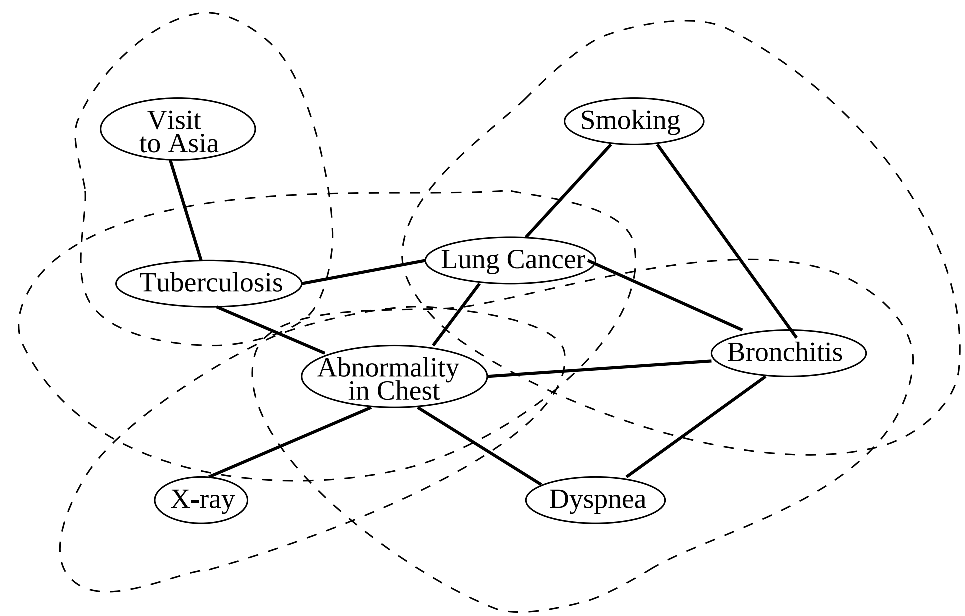
Example: Asia network

For our previous example, let us construct the induced graph:



Note that the moralized graph is always a subgraph of the induced graph.

Cliques



- A complete subgraph of H is a subset of vertices such that each vertex is connected to every other vertex
- A clique is a maximal complete subgraph (one to which no vertices can be added)

Complexity of variable elimination

Theorem:

1. Every clique in the induced graph corresponds to an intermediate factor in the computation
2. Every factor generated during variable elimination is a subset of some maximal clique.

See Koller and Friedman notes for the proof details.

Therefore, *complexity is exponential in the size of the largest clique*

Consequence: Polytree inference

For the class of polytree networks, the problem of computing $P(X, y)$ for any X can be solved in time linear in the size of the network (which includes all the CPTs).

Proof: We can order the nodes from the leaves inward. The induced graph is exactly the moral graph of the tree. So the largest clique is the largest family in the graph. The conclusion follows.

Heuristics for node ordering

- *Maximum cardinality:* Number the nodes from 1 to n, always assigning the next number to the vertex having the largest set of previously numbered neighbors. Then eliminate nodes from n to 1.
- *Minimum discrepancy:* Always eliminate the node that causes the fewest edges to be added to the induced graph
- *Minimum size/weight:* Eliminate the node that causes the smallest clique to be created (either in terms of number of nodes, or in terms of number of entries).

Summary

- General exact inference in Bayesian networks is NP-hard
- Variable elimination is a general algorithm for exact inference
- By analyzing variable elimination we can see the “easy” cases for inference:
  - when the net is a polytree
  - when the maximum clique of the induced graph is small
- Heuristics for ordering work pretty well in practice