Advanced Data Analysis and Modelling Summerschool

Inference in Bayesian Networks

José A. Gámez

jgamez@info-ab.uclm.es

Intelligent Systems and Data Mining (SIMD) research group

Computing Systems Department

University of Castilla-La Mancha

Campus Universitario s/n

Albacete, 02071, Spain

Contents

Inference

- By inference in BNs we refer to the task of computing a-posteriori probabilities.
- This task can be found under different names: probability propagation, belief updating, belief revision, ...
- Most of queries involve observations or evidence

  Evidence on a variable is a statement of the certainties of its states, i.e., (flu=yes).
- Hard evidence. An evidence function that assigns a zero probability to all but one state is said to provide hard evidence. Hard evidence **e** over a set of variables E is often referred to as instantiation.

  temperature=high, headache=no
- Soft evidence. An evidence function that assigns a probability distribution over dom(E_i) for each $E_i \in E$ is said to provide soft evidence.

- Example. If dom(temperature) = (no, ligth, high, very-high) and dom(headache)=(yes,no), the we can have the following soft evidence:
- se(temperature) = (0,0,2,1) and se(headache) = (2,1)

Type of queries

- The *simplest* query: to compute the evidence probability:

$$P(\mathbf{e}) = \sum_{X_i \notin E} P(X_1, \ldots, X_n, \mathbf{e})$$

- The most *frequent* query: compute the a-posteriori probability for a given target or interest variable.

$$P(X|\mathbf{e}) = \frac{P(X, \mathbf{e})}{P(\mathbf{e})}$$

To do this we only need to compute $P(X, \mathbf{e})$, because

$$P(\mathbf{e}) = \sum_{x \in dom(X)} P(X = x, \mathbf{e})$$

In general, we are not only interested in a single variable but in a set of them, usually all the unobserved ones.

Type of queries (II)

- Computing the a-posteriori probability of a given variable is useful in different situations:
  - *Predictive or deductive reasoning*: What is the probability of observing a sympton knowing the presence of a given desease?

    In this case the target variable usually is a descendant of the evidence.
  - *Diagnostic or abductive reasoning*: What is the probability of desease D being the correct diagnosis given some symptoms?

    In this case the target variable usually is an ancestor of the evidence.
- That is, in the BNs framework the direction of the links between the variables does not constraint the type of query to be posed.

Type of queries (III)

Queries about sets of variables:

- A-posteriori probability of a subset of variables: $P(X, Y, \ldots | \mathbf{e})$
- Searching for the most probable explanation, the configuration of maximal probabiity, belief revision or abductive inference:
  - Total abduction or MPE: If $X_1, \ldots, X_n$ are the unobserved variables, then the goal is to identify the configuration $(x_1, \ldots, x_n)$ that maximises $P(X_1, \ldots, X_n | \mathbf{e})$.
  - Partial abduction or MAP: Given a subset $\{X_1, \ldots, X_l\}$ of the unobserved variables, then the goal is to identify the configuration $(x_1, \ldots, x_l)$ that maximises $P(X_1, \ldots, X_l | \mathbf{e})$.
  - In general, the goal is to look for the K most probable explanations.

Basic operations

To get answers for the previous queries we only need a few operations:

- **Projection**. Given two sets of variables $\mathbf{X}$ and $\mathbf{Y}$, such that, $\mathbf{X} \cap \mathbf{Y} \neq \emptyset$, then

$$\mathbf{Z} = \mathbf{X}^{\downarrow \mathbf{Y}}$$

  contains the variables of $\mathbf{X}$ that also are in $\mathbf{Y}$.

  - Projection also applies to configurations, thus, if $\mathbf{x}$ and $\mathbf{y}$ are configurations of $\mathbf{X}$ and $\mathbf{Y}$, then

$$\mathbf{z} = \mathbf{x}^{\downarrow \mathbf{y}}$$

    contains the sub-configuration of $\mathbf{x}$ restricted to the variables in $\mathbf{X}$ that also are in $\mathbf{Y}$.

- **Combination.** Given two pieces of information defined over $\mathbf{X}$ and $\mathbf{Y}$, the goal of the *combination* is to obtain a new information defined over the $\mathbf{X} \cup \mathbf{Y}$.
  - In our case the piece of information are probability functions or potentials, and the result is a new probability function or potential obtained by point-wise multiplication:

Basic Operations (II)

- Combination Example: Assuming all variables are binary, then from $f_1(A,B)$ and $f_2(A,C)$ we get:

$$f(A,B,C) = f_1(A,B) \times f_2(A,C)$$

$$\begin{pmatrix} & b & \overline{b} \\ a & 0.5 & 0.8 \\ \overline{a} & 0.5 & 0.2 \end{pmatrix} \times \begin{pmatrix} & c & \overline{c} \\ a & 1.0 & 0.4 \\ \overline{a} & 0.0 & 0.6 \end{pmatrix} = \begin{pmatrix} & b,c & b,\overline{c} & \overline{b},c & \overline{b},\overline{c} \\ a & 0.50 & 0.20 & 0.80 & 0.32 \\ \overline{a} & 0.00 & 0.30 & 0.00 & 0.12 \end{pmatrix}$$

- Division. Point-wise division is used.

  However, we distinguish two cases in order to take care with division by zero, thus

$$(\phi/\psi)(\mathbf{z}) = \begin{cases} 0 & if\,\psi(x^{\downarrow Z}) = 0 \\ \phi(x^{\downarrow Z})/\psi(y^{\downarrow Z}) & if\,\psi(y^{\downarrow Z}) \neq 0 \end{cases}$$

- In fact in the operations involved in probabilistic networks, $\psi(x^{\downarrow Z}) = 0$ implies $\psi(y^{\downarrow Z}) = 0$ upon division of ϕ by ψ, and thus defining $0/0 = 0$, the division operator is always defined.

Basic Operations (III)

- Marginalization. Given an information defined over a set of variables X_I, marginalization restricts that information over a subset of $X_J \subseteq X_I$.

  In Belief revision variables not included in the interest set are marginalised out by addition, while in belief revision or abduction they are marginalised out by maximum.

  Example:

$$f_1(A) = \sum_{B,C} f(A,B,C) = \sum_{B,C} \begin{array}{c} \\ a \\ \bar{a} \end{array} \begin{pmatrix} b,c & b,\bar{c} & \bar{b},c & \bar{b},\bar{c} \\ 0.50 & 0.20 & 0.80 & 0.32 \\ 0.00 & 0.30 & 0.00 & 0.12 \end{pmatrix} = \begin{pmatrix} a & 1.82 \\ \bar{a} & 0.42 \end{pmatrix}$$

$$f_1(A) = \max_{B,C} f(A,B,C) = \max_{B,C} \begin{array}{c} \\ a \\ \bar{a} \end{array} \begin{pmatrix} b,c & b,\bar{c} & \bar{b},c & \bar{b},\bar{c} \\ 0.50 & 0.20 & 0.80 & 0.32 \\ 0.00 & 0.30 & 0.00 & 0.12 \end{pmatrix} = \begin{pmatrix} a & 0.80 \\ \bar{a} & 0.30 \end{pmatrix}$$

Computing $P(X_i|\mathbf{e})$: Brute-force approach

- We focus on the computation of $P(X|\mathbf{e})$.
- By the moment we suppose that no evidence has been entered $\rightarrow$ to compute $P(X)$.
- Given a BN with n variables $\{X_1,\ldots,X_n\}$ and their probability families $f_i, i=1,\ldots,n$, then to compute $P(X_i)$ (or $P(X_i|\mathbf{e})$ is:

Conceptually easy

Computationally complex

- Brute-force approach:

$$P(X_i) = \sum_{X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_n} \left(\prod_{j=1}^{n} f_j \right),$$

Problem: this is the same to compute the j.p.d. $\Rightarrow$ computationally very inneficient and even intractable in most of cases.

Improving brute-force approach

In order to improve the brute-force approach we will take advantage from two sources:

- The factorisation encoded by the BN
- The distributive law

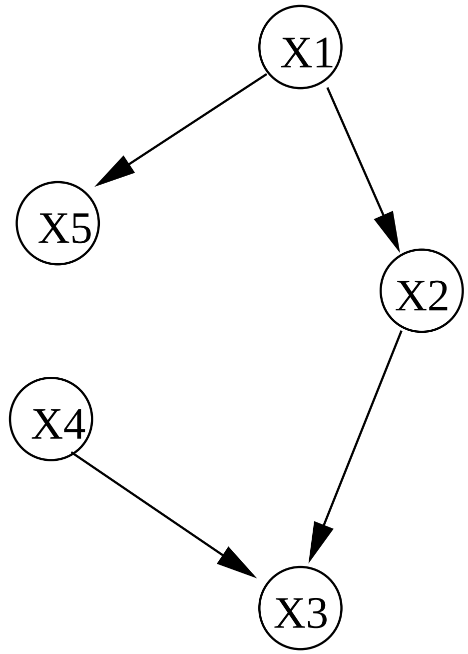
Distributive Law.

Let f and g to be potentials or probability fucntions defined over $dom(\mathbf{X}) = \{\mathbf{x_1}, \ldots, \mathbf{x_m}\}$ and $dom(\mathbf{Y}) = \{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$, where $\mathbf{X} \cap \mathbf{Y} = \emptyset$, and given $\mathbf{X}' \subseteq \mathbf{X}$ and $\mathbf{Y}' \subseteq \mathbf{Y}$, then we get

$$
\begin{aligned}
\sum_{\mathbf{X}\setminus\mathbf{X}'} \sum_{\mathbf{Y}\setminus\mathbf{Y}'} (f \times g) &= \sum_{\mathbf{x}\in dom(\mathbf{X}\setminus\mathbf{X}')} \sum_{\mathbf{y}\in dom(\mathbf{Y}\setminus\mathbf{Y}')} (f(\mathbf{x}) \times g(\mathbf{y})) \\
&= f(\mathbf{x_1})g(\mathbf{y_1}) + \cdots + f(\mathbf{x_1})g(\mathbf{y_n}) + \cdots + \\
&\quad f(\mathbf{x_m})g(\mathbf{y_1}) + \cdots + f(\mathbf{x_m})g(\mathbf{y_n}) \\
&= f(\mathbf{x_1})[g(\mathbf{y_1}) + \cdots + g(\mathbf{y_n})] + \cdots + \\
&\quad f(\mathbf{x_m})[g(\mathbf{y_1}) + \cdots + g(\mathbf{y_n})] \\
&= \sum_{\mathbf{x}\in dom(\mathbf{X}\setminus\mathbf{X}')} f(\mathbf{x}) \sum_{\mathbf{y}\in dom(\mathbf{Y}\setminus\mathbf{Y}')} g(\mathbf{y}) \\
&= \sum_{\mathbf{X}\setminus\mathbf{X}'} f \sum_{\mathbf{Y}\setminus\mathbf{Y}'} g
\end{aligned}
$$

Ordering the computations effectively

As we will see with the following example, the use of the distributive law can help a lot in terms of reducing computations:



f1 = P(X1)

f2 = P(X2 | X1)

f3 = P(X3 | X2, X4)

f4 = P(X4)

f5 = P(X5 | X1)

As commented before we suppose that any evidence has been observed. Our goal is to compute $P(X_2)$. Thus, be brute-force approach we have:

$$P(X_2) = \sum_{X_1, X_3, X_4, X_5} \left(\prod_{j=1}^{5} f_j \right) =$$

Ordering the computations effectively (II)

$$\sum_{X_1,X_3,X_4,X_5} \{P(X_1)\cdot P(X_2|X_1)\cdot P(X_3|X_2,X_4)\cdot P(X_4)\cdot P(X_5|X_1)\}$$

If $|dom(X)| = 2$ for all the variables, then this expression implies to construct a probability table with 32 entries (i.e. the j.p.d.).

However, from the factorisation and the distributive law we can simplify the process by moving in some additions:

$$P(X_2) =$$

Moving the summation over X_5.

$$= \sum_{X_1,X_3,X_4} \left\{ \left(\underbrace{\sum_{X_5} P(X_5|X_1)}_{f_1(X_1)} \right) \cdot P(X_1)\cdot P(X_2|X_1)\cdot P(X_3|X_2,X_4)\cdot P(X_4) \right\}$$

Ordering the computations effectively (III)

Moving the summation over X_3.

$$= \sum_{X_1,X_4} \left\{ \left(\underbrace{\sum_{X_5} P(X_5|X_1)}_{f_1(X_1)} \right) \cdot P(X_1) \cdot P(X_2|X_1) \cdot \left(\underbrace{\sum_{X_3} P(X_3|X_2,X_4)}_{f_2(X_2,X_4)} \right) \cdot P(X_4) \right\}$$

And now we can move the summation over X_4.

$$= \sum_{X_1} \left\{ \left(\underbrace{\sum_{X_5} P(X_5|X_1)}_{f_1(X_1)} \right) \cdot P(X_1) \cdot P(X_2|X_1) \cdot \left[\underbrace{\sum_{X_4} \left(\underbrace{\sum_{X_3} P(X_3|X_2,X_4)}_{f_2(X_2,X_4)} \right) \cdot P(X_4)}_{f_3(X_2)} \right] \right\}$$

Ordering the computations effectively (summary)

- Brute-force approach:

$$\sum_{X_1,X_3,X_4,X_5} \{P(X_1)\cdot P(X_2|X_1)\cdot P(X_3|X_2,X_4)\cdot P(X_4)\cdot P(X_5|X_1)\}$$

  This means to build a table with 5 variables and 32 entries. Then, we need:
  - 160 multiplications (in most implementations)
  - 52 multiplications (selecting the tables in the appropriate way).
  - 30 additions for the marginalization of X_1, X_3, X_4 and X_5 (16, 8, 4 and 2).
- Taking advantage from the factorisation and the distributive law:

$$= \sum_{X_1}\left\{\left(\sum_{X_5} P(X_5|X_1)\right)\cdot P(X_1)\cdot P(X_2|X_1)\cdot\left[\sum_{X_4}\left(\sum_{X_3} P(X_3|X_2,X_4)\right)\cdot P(X_4)\right]\right\}$$

  This means to deal with a table of size 8 and three of size 4. Then, we need:
  - 14 multiplications.
  - 10 additions for the marginalization.

Complexity of exact inference in BNs

- If the DAG no cycles in the underlying undirected graph (poly-trees), inference is easy because we can move the additions in such a way that we never create a table larger than those included in the BN representation.
- In the general case (the underlying undirected graph has cycles) inference is NP-Complete (Cooper, 1990)
- The complexity of the previous method is exponential in the width (number of variables minus one) of the largest factor set involved in the pocess.
- The key to efficient inference with this method lies in finding a good *summation order* (or *elimination order* or *deletion sequence* or ...)

Entering evidence

- Up to this moment we have supposed the lack of evidence. But, what happens if we have some observations, i.e, $\mathbf{e} = (E_1 = e_1, \ldots, E_n = e_n)$?.
- The answer is quite simple, before running our algorithm, for each E_i we identify the potentials or prob. fucntions in which it is included, then:

$$f(x) = \begin{cases} f(x) & \text{if x is consistent with } \mathbf{e} \\ 0 & \text{otherwise} \end{cases}$$

- Sometimes we can use evidence absorption, which implies the removal of the observed variable from the potential.
- Example: $\mathbf{e} = (B = b)$

$$\left(\begin{array}{ccccc} & b,c & b,\overline{c} & \overline{b},c & \overline{b},\overline{c} \\ a & 0.5 & 0.2 & 0.0 & 0.0 \\ \overline{a} & 0.0 & 0.3 & 0.0 & 0.0 \end{array}\right) \longleftarrow \left(\begin{array}{ccccc} & b,c & b,\overline{c} & \overline{b},c & \overline{b},\overline{c} \\ a & 0.5 & 0.2 & 0.8 & 0.32 \\ \overline{a} & 0.0 & 0.3 & 0.0 & 0.12 \end{array}\right) \longrightarrow \left(\begin{array}{ccc} & c & \overline{c} \\ a & 0.50 & 0.20 \\ \overline{a} & 0.00 & 0.30 \end{array}\right)$$

Variable Elimination Algorithm

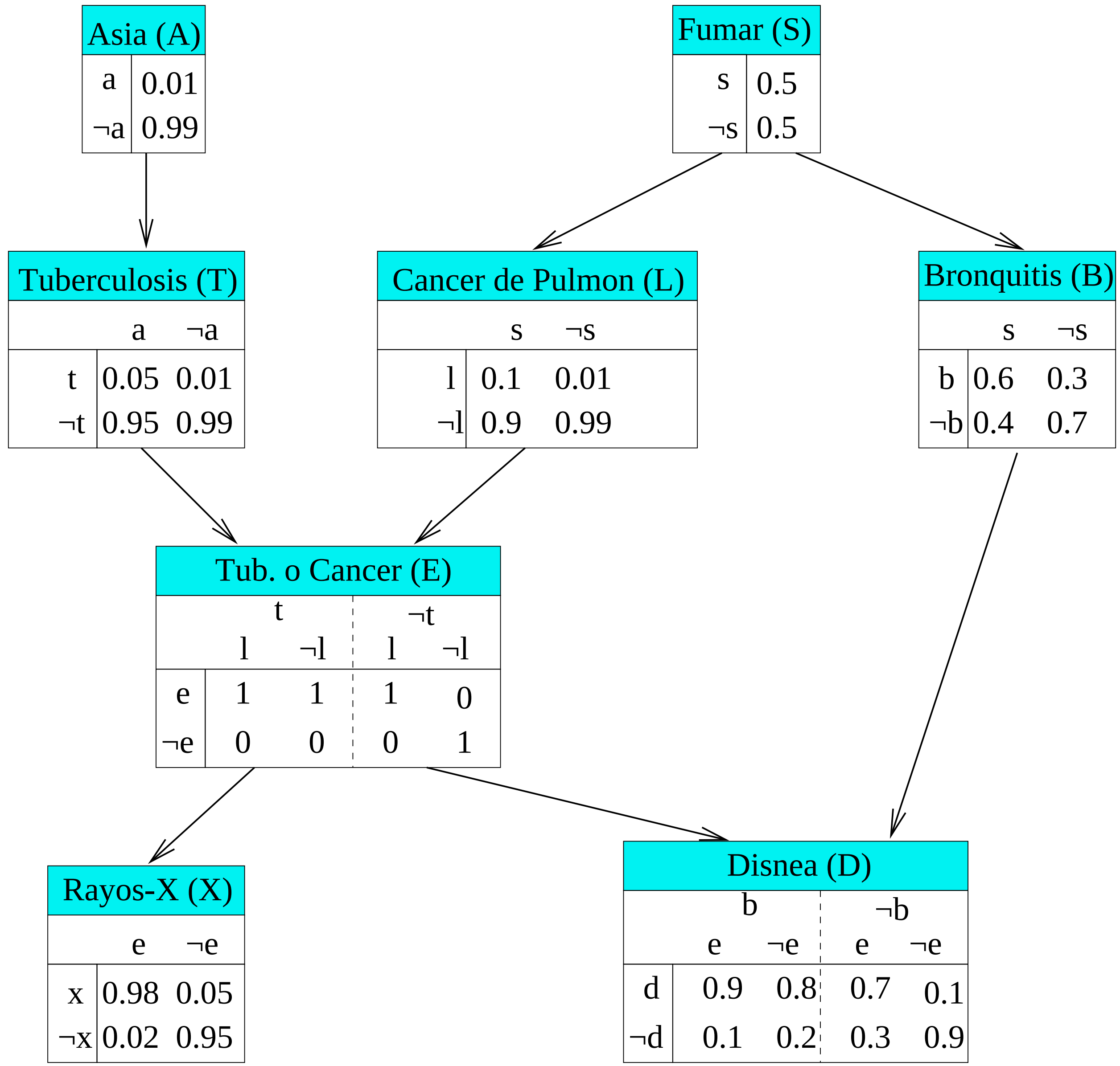
- Input: A BN over $\mathcal{U} = \{X_1, \ldots, X_n\}$, the evidence $\mathbf{e}$ and ONE target variable X_i.
- Output: $P(X_i|\mathbf{e})$.

1. Let $\mathcal{L}$ be a list containing all the probability functions $\{f_1, f_2, \ldots, f_n\}$.
2. Enter the evidence $\mathbf{e}$.
3. Select an elimination order σ containing all the variables but the target one (X_i).
4. For $k = 1$ to $n - 1$ do
   (a) $X_k \leftarrow \sigma(k)$.
   (b) Let F be the set of prob. functions in $\mathcal{L}$ that contains variable X_k.
   (c) $\mathcal{L} = \mathcal{L} - F$.
   (d) $f' = \sum_{X_k} \left(\prod_{f \in F} f \right)$
   (e) $\mathcal{L} = \mathcal{L} \cup f'$.
5. Combine in a single function f all the functions in $\mathcal{L}$. Normalize f to obtain $P(X_i)$.

This algorithm has to be repeated for each target variable.

Examples

Let us consider the following BN (ASIA or chest-clinic) for the examples:



Examples (cont.)

Example 1. Get the probability of having Dysnoea (D), using the following elimination order $\sigma_1 = T, S, E, A, L, B, X$.

1 $\mathcal{L} = \{ f_A(A), \underbrace{f_T(T,A)}, f_S(S), f_L(L,S), f_B(B,S), \underbrace{f_E(E,T,L)}, f_X(X,E),\ f_D(D,E,B)\}$. **Delete T**.

$$g_1(A,E,L) = \sum_T (f_T(A,T) \times f_E(E,T,L))$$

size = 16

2 $\mathcal{L} = \{ f_A(A), \underbrace{f_S(S), f_L(L,S), f_B(B,S)}, f_X(X,E), f_D(D,E,B), g_1(A,E,L)\}$. **Delete S**.

$$g_2(L,B) = \sum_S (f_S(S) \times f_L(L,S) \times f_B(B,S))$$

size = 8

3 $\mathcal{L} = \{ f_A(A), \underbrace{f_X(X,E), f_D(D,E,B), g_1(A,E,L)}, g_2(L,B)\}$. **Del. E**

$$g_3(X,D,B,A,L) = \sum_E (f_X(X,E) \times f_D(D,E,B) \times g_1(A,E,L))$$

size = 64

Examples (cont.)

4 $\mathcal{L} = \{\underbrace{f_A(A)}, g_2(L,B), \underbrace{g_3(X,D,B,A,L)}\}$. **Delete A** size = 32

$$g_4(X,D,B,L) = \sum_A (f_A(A) \times g_3(X,D,B,A,L))$$

5 $\mathcal{L} = \{\underbrace{g_2(L,B), g_4(X,D,B,L)}\}$. **Delete L**. size = 16

$$g_5(X,D,B) = \sum_L g_2(L,B) \times g_4(X,D,B,L)$$

6 $\mathcal{L} = \{\underbrace{g_5(X,D,B)}\}$. **Delete B**. size = 8

$$g_6(X,D) = \sum_B g_5(X,D,B)$$

7 $\mathcal{L} = \{\underbrace{g_6(X,D)}\}$. **Delete X**. size = 8

$$g_7(D) = \sum_X g_6(X,D)$$

8 return normalize($g_7(D)$)

Examples (cont.)

Ejemplo 2. Get the probability of having Dysnoea (D), by using the following elimination order $\sigma_1 = A, X, T, S, L, E, B$.

$$f_A(A), f_T(T,A), f_S(S), f_L(L,S), f_B(B,S), f_E(E,T,L), f_X(X,E), f_D(D,E,B)$$

Delete A Use: $f_A(A), f_T(A,T)$ New: $g_1(T)$ 4

$$f_S(S), f_L(L,S), f_B(B,S), f_E(E,T,L), f_X(X,E), f_D(D,E,B), g_1(T)$$

Delete X Use: $f_X(X,E)$ New: $g_2(E)$ 4

$$f_S(S), f_L(L,S), f_B(B,S), f_E(E,T,L), f_D(D,E,B), g_1(T), g_2(E)$$

Delete T Use: $f_E(E,T,L), g_1(T)$ New: $g_3(E,L)$ 8

$$f_S(S), f_L(L,S), f_B(B,S), f_D(D,E,B), g_2(E), g_3(E,L)$$

Examples (cont.)

$$f_S(S),\, f_L(L,S),\, f_B(B,S),\, f_D(D,E,B), g_2(E), g_3(E,L)$$

Delete S   Use: $f_S(S),\, f_L(L,S),\, f_B(B,S)$ New: $g_4(L,B)$ 8

$$f_D(D,E,B), g_2(E), g_3(E,L), g_4(L,B)$$

Delete L   Use: $g_3(E,L), g_4(L,B)$ New: $g_5(E,B)$ 8

$$f_D(D,E,B), g_2(E), g_5(E,B)$$

Delete E   Use: $f_D(D,E,B), g_2(E), g_5(E,B)$ New: $g_6(D,B)$ 8

$$g_5(D,B)$$

Delete B   Use: $g_6(D,B)$ New: $g_7(D)$ 4

$$g_7(D)$$

Query-based inference

- In a concrete scenary the network's variables can be divided into: *interest* ($\mathbf{I}$) *variables*, *observed variables or evidence* ($\mathbf{E} = \mathbf{e}$) and the remaining ones ($\mathbf{R}$).
- The question now is: *do we need to consider all the variables in* $\mathbf{R}$*?*
- Answer: usually no
- Then, prior to solving the query, the network can be pruned to include only the variables relevant for the query.

  To prune the network (i.e. to discard some variables in $\mathbf{R}$) we will use as tools the concepts of *barren* nodes and *d-separation*
- D-separation: We can prune all variables $\mathbf{K}$ such that $I(\mathbf{I}|\mathbf{E}|\mathbf{K})$. There exists efficient algorithms for this task (i.e. BayesBall (Schacther, 1998)