COMP538: Introduction to Bayesian Networks

Lecture 4: Inference in Bayesian Networks: The VE Algorithm

Nevin L. Zhang
lzhang@cse.ust.hk

Department of Computer Science and Engineering
Hong Kong University of Science and Technology

Fall 2008

Objective

- Discuss the variable elimination (VE) algorithm for inference in Bayesian networks
- Reading: Zhang and Guo, Chapter 4
- Reference: Zhang and Poole (1994, 1996 (first few sections)); Dechter (1996)

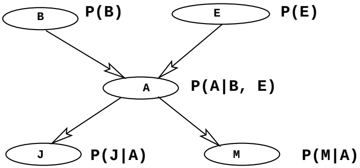
Outline

Queries about posterior probability

- Posterior queries:
  - Given: The values of some variables.
  - Task: Compute the posterior probability distributions of other variables?

  MAP and MPE queries to be discussed later.
- Example:
  - Both John and Mary called to report alarm.
  - What is the probability of burglary?
  - Formally, what is the posterior probability distribution $P(B|J{=}y, M{=}y)$?
- General form of query: $P(\mathbf{Q}|\mathbf{E}{=}\mathbf{e})$?
  - $\mathbf{Q}$ is a list of query variables, usually one.
  - $\mathbf{E}$ is a list of evidence variables,and $\mathbf{e}$ is the corresponding list observed values.
  - Note: Bold capital letters denote sets of variables.
- **Inference** refers to the process of computing the answer to a query.

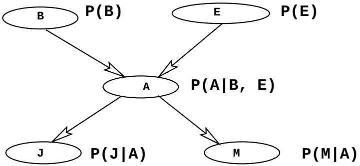
Diagnostic and Predictive Inference



Semantically, four types of queries:

- Diagnostic inference: From effects to causes.
  - $P(B|M{=}y)$
  - Machine malfunctions. What is wrong?
- Predictive/Causal inference: From causes to effects.
  - $P(M|B{=}y)$

Inter-causal inference



- Inter-causal inference:
  - Between causes of a common effect.
  - Example: $P(B|A{=}y, E{=}y)$
  - **Explaining away**:

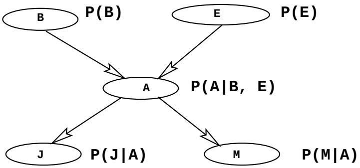
$$P(B{=}y|A{=}y) > P(B{=}y|A{=}y, E{=}y)$$

Earthquake explains away $A = y$.

$$P(B{=}y|A{=}y) < P(B{=}y|A{=}y, E{=}n)$$

  - Exercise: Verify the inequalities.
  - Note: Difficult with logic rules rules:
    - $A = y \rightarrow B = y(0.8)$.
    - $E = y \rightarrow A = y(0.9)$.
    - Fact: $E = y$
    - Conclusion: $B = y(0.72)$. Wrong!

Mixed Inference



- Mixed inference:
  - Combining two or more of the above.
  - $P(A|J{=}y, E{=}Y)$ (Simultaneous use of diagnostic and causal inferences)
  - $P(B|J{=}y, E{=}n)$ (Simultaneous use of diagnostic and inter-causal inferences)

All those types can be handled in the same way.

In logic inference, different query types are handled differently:

- Predictive inference: deduction.
- Diagnostic inference: abduction.

Outline

A naive inference algorithm

- Naive algorithm for computing $P(\mathbf{Q}|\mathbf{E}=\mathbf{e})$ in a Bayesian network:
  - Get joint probability distribution $P(\mathbf{X})$ over the set $\mathbf{X}$ of all variables by multiplying conditional probabilities.
  - Marginalize

$$P(\mathbf{Q},\mathbf{E}) = \sum_{\mathbf{X}-\mathbf{Q}\cup\mathbf{E}} P(\mathbf{X}), P(\mathbf{E}) = \sum_{\mathbf{Q}} P(\mathbf{Q},\mathbf{E})$$

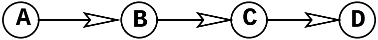
  - Condition:

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

- Example
  - $P(B,J,M) = \sum_{E,A} P(B,E,A,J,M)$, $P(J,M) = \sum_{B} P(B,J,M)$.
  - $P(B|J{=}y, M{=}y) = \frac{P(B,J=y,M=y)}{P(J=y,M=y))}$
- Not making use of the factorization,exponential complexity.
- Key issue: How to exploit the factorization to avoid exponential complexity?

Principle Through Example

- Network: $P(A)$, $P(B|A)$, $P(C|B)$, $P(D|C)$.



- Query: P(D)?
- Computation:

$$\begin{aligned}
P(D) &= \sum_{A,B,C} P(A,B,C,D) \\
&= \sum_{C}\sum_{B}\sum_{A} P(A)P(B|A)P(C|B)P(D|C) && (1) \\
&= \sum_{C}\sum_{B} P(C|B)P(D|C)\sum_{A} P(A)P(B|A) \\
&= \sum_{C} P(D|C) \;\; \sum_{B} P(C|B) \;\; \sum_{A} P(A)P(B|A) && (2)
\end{aligned}$$

Principle Through Example

- Complexity — Number of numerical summations:
  - Use (1): $2^3 + 2^2 + 2$.
  - Use (2): $2 + 2 + 2$.
- Exercise: How about numerical multiplications?

Principle Through Example

Rewrite expression (2) into an algorithm:

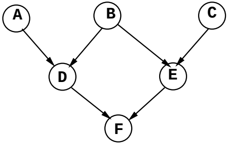
- Let $\mathcal{F} = \{P(A), P(B|A), P(C|B), P(D|C)\}$
- Remove from $\mathcal{F}$ all the functions that involve A, create a new function by
$$\psi_1(B) = \sum_A P(A)P(B|A).$$
put the new function onto $\mathcal{F} : \mathcal{F} = \{\psi_1(B), P(C|B), p(D|C)\}$.
- Remove from $\mathcal{F}$ all the functions that involve B, create a new function by
$$\psi_2(C) = \sum_B P(C|B)\psi_1(B).$$
put the new function onto $\mathcal{F} : \mathcal{F} = \{\psi_2(C), p(D|C)\}$.
- Remove from $\mathcal{F}$ all the function that involve C, create a new function by
$$\psi_3(D) = \sum_C P(D|C)\psi_2(C).$$
- Return $\psi_3(D)$ (which is exactly $P(D)$).

Factorization

- A **factorization** of a joint distribution is a list of functions whose product is the joint distribution.
  - Functions on the list are called factors.
- A BN gives a factorization of a joint probability:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|pa(X_i)).$$

- Example:



This BN factorizes $P(A, B, C, D, E, F)$ into the following list of factors:

$$P(A), P(B), P(C), P(D|A, B), P(E|B, C), P(F|D, E).$$

Eliminating a variable

- Consider a joint distribution

$$P(Z_1, Z_2, \ldots, Z_m)$$

- Eliminating Z_1 from P means to compute

$$P(Z_2, \ldots, Z_m) = \sum_{Z_1} P(Z_1, Z_2, \ldots, Z_m).$$

- The complexity is exponential in m.

Eliminating a variable

- Now suppose we have factorization: $P(Z_1, Z_2, \ldots, Z_m) = f_1 \times f_2 \times \ldots \times f_n$
- Obtaining a factorization of $P(Z_2, \ldots, Z_m)$ could be done with much less computation:

  Procedure `eliminate`$(\mathcal{F}, Z)$:

  - Inputs: $\mathcal{F}$ — A list of functions; Z — A variable.
  - Output: Another list of functions.

  1. Remove from the $\mathcal{F}$ all the functions, say f_1, ..., f_k, that involve Z,
  2. Compute new function $g = \prod_{i=1}^{k} f_i$.
  3. Compute new function $h = \sum_Z g$.
  4. Add the new function h to $\mathcal{F}$.
  5. Return $\mathcal{F}$.

- $\sum_Z \prod_{i=1}^{k} f_i$ can be much cheaper than $\sum_Z P(Z_1, Z_2, \ldots, Z_m)$.

Eliminating a variable

Theorem (4.1)

Suppose $\mathcal{F}$ is a factorization of a joint probability distribution $P(Z_1, Z_2, \ldots, Z_m)$. Then `eliminate`$(\mathcal{F}, Z_1)$ *is a factorization of the marginal probability distribution $P(Z_2, \ldots, Z_m)$.*

Proof:

- Suppose $\mathcal{F}$ consists of factors f_1, f_2, ..., f_n.
- Suppose Z_1 appears in and only in factors f_1, f_2, ..., f_k.

$$\begin{aligned} P(Z_2, \ldots, Z_m) &= \sum_{Z_1} P(Z_1, Z_2, \ldots, Z_m) \\ &= \sum_{Z_1} \prod_{i=1}^{n} f_i = \sum_{Z_1} \prod_{i=1}^{k} f_i \prod_{i=k+1}^{n} f_i \\ &= [\prod_{i=k+1}^{n} f_i][\sum_{Z_1} \prod_{i=1}^{k} f_i] = [\prod_{i=k+1}^{n} f_i]h. \text{ Q.E.D} \end{aligned}$$

Observed variable instantiation

- Function $h(X, Y)$.

X \ Y	0	1
0	.3	.8
1	.6	0

- Suppose X is observed and $X{=}0$.
- Instantiating X in h (to its observed value) resulting a function $g(Y) = h(X = 0, Y)$ of Y only:

Y	0	1
	.3	.8

The Variable Elimination Algorithm

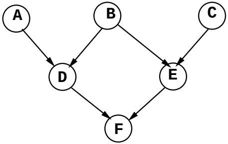
Procedure $\texttt{VE}(\mathcal{F}, \mathbf{Q}, \mathbf{E}, \mathbf{e}, \rho)$ *//for computing* $P(\mathbf{Q}|\mathbf{E}{=}\mathbf{e})$*:*

- *Inputs*: $\mathcal{F}$ — *The list of CPTs in a BN;*
  $\mathbf{Q}$ — *A list of query variables;*
  $\mathbf{E}$ — *A list of observed variables;* $\mathbf{e}$ — *Observed values;*
  ρ — *Ordering of variables* $\notin \mathbf{Q} \cup \mathbf{E}$ *(**Elimination ordering**).*
- *Output*: $P(\mathbf{Q}|\mathbf{E}{=}\mathbf{e})$.

1. **While** ρ *is not empty,*
   1. *Remove the first variable* Z *from* ρ,
   2. *Call* $\texttt{eliminate}(\mathcal{F}, Z)$. **Endwhile**
2. *Set* h = *product of all the factors in* $\mathcal{F}$.
3. *Instantiate observed variables in* h *to their observed values.*
4. *Return* $h(\mathbf{Q}) / \sum_{\mathbf{Q}} h(\mathbf{Q})$. *// Re-normalization*

Example

- Query: $P(A|F=0)$?



- Elimination ordering: ρ — C, E, B, D
- Initial factorization:
  $\mathcal{F} = \{P(A), P(B), P(C), P(D|A,B), P(E|B,C), P(F|D,E)\}$
- Inference process:
  - Step 1, eliminate C:

    $$\mathcal{F} = \{P(A), P(B), P(D|A,B), P(F|D,E), \psi_1(B,E)\}$$

    where $\psi_1(B,E) = \sum_C P(C)P(E|B,C)$.
  - Step 1, eliminate E:

    $$\mathcal{F} = \{P(A), P(B), P(D|A,B), \psi_2(B,D,F)\}$$

    where $\psi_2(B,D,F) = \sum_E P(F|D,E)\psi_1(B,E)$.

Example (cont'd)

- Continued from previous slide
  - Step 1, eliminate B:

$$\mathcal{F} = \{P(A), \psi_3(A, D, F)\}$$

    where $\psi_3(A, D, F) = \sum_B P(B)P(D|A, B)\psi_2(B, D, F)$
  - Step 1, eliminate D:

$$\mathcal{F} = \{P(A), \psi_4(A, F)\}$$

    where $\psi_4(A) = \sum_D \psi_3(A, D, F)$
  - Step 2: $h(A, F) = P(A)\psi_4(A, F)$.
  - Step 3: $h(A) = h(A, F = 0)$.
  - Step 4: $P(A|F{=}0) = \frac{h(A)}{\sum_A h(A)}$.

The Variable Elimination Algorithm

Theorem (4.2)

The output of $\mathtt{VE}(\mathcal{F}, \mathbf{Q}, \mathbf{E}, \mathbf{e}, \rho)$ *is* $P(\mathbf{Q}|\mathbf{E}{=}\mathbf{e})$.

Proof:

- By repeatedly applying Theorem 4.1, we conclude that,after the while-loop, $\mathcal{F}$ is a factorization of $P(\mathbf{Q}, \mathbf{E})$.
- Hence, after step 2, h is:
$$h(\mathbf{Q}, \mathbf{E}) = P(\mathbf{Q}, \mathbf{E}).$$
- After step 3, h is:
$$h(\mathbf{Q}) = P(\mathbf{Q}, \mathbf{E}{=}\mathbf{e}).$$
- Consequently,
$$\frac{h(\mathbf{Q})}{\sum_{\mathbf{Q}} h(\mathbf{Q})} = \frac{P(\mathbf{Q}, \mathbf{E}{=}\mathbf{e})}{\sum_{\mathbf{Q}} P(\mathbf{Q}, \mathbf{E}{=}\mathbf{e}))} = \frac{P(\mathbf{Q}, \mathbf{E}{=}\mathbf{e})}{P(\mathbf{E}{=}\mathbf{e}))} = P(\mathbf{Q}|\mathbf{E}{=}\mathbf{e}). \text{ Q.E.D}$$

A Modification

Procedure $\texttt{VE}(\mathcal{F}, \mathbf{Q}, \mathbf{E}, \mathbf{e}, \rho)$

1. Instantiate observed variables in all functions.
2. **While** ρ is not empty,
   1. Remove the first variable Z from ρ,
   2. Call $\texttt{eliminate}(\mathcal{F}, Z)$. **Endwhile**
3. Set $h =$ the multiplication of all the factors on $\mathcal{F}$.
4. Return $h(\mathbf{Q})/\sum_{\mathbf{Q}} h(\mathbf{Q})$.

Exercises:

- Formally show the correctness of this version of VE.
- Explain why it is more efficient that the version given earlier.

Note: This algorithm was first described in Zhang and Poole (1994).

Outline

Measuring the Complexity of One Step

- For any variable, let $w(X)$ be the number of possible values of X.
- Complexity of `eliminate`:
  - At step 2, a new function g is constructed.
  - The size of $g=$ $\prod\{w(X) : X$ appears in one of the functions that involve $Z.\}$.
  - The size is a good and nature measurement of the complexity of eliminating Z.
    (Accurate operation counts are difficult.)
  - We call the size of g the **cost** of eliminating z from $\mathcal{F}$ and denote it by $c(Z)$.
- In the previous example, assume all variables are binary.
  - The cost of eliminating C is: 8
  - The cost of eliminating E is: 16
  - The cost of eliminating B is: 16
  - The cost of eliminating D is: 8

Measuring the Complexity of the VE Algorithm

- Complexity of VE:
  - Suppose the elimination ordering is: $Z_1, Z_2, \ldots, Z_m$.
  - The **cost of VE** is defined to be:

$$\sum_{i=1}^{m} c(Z_i)$$

- Complexity in the previous example:
  - Cost of VE is: $8+8+8+4=36$.
- Often, one term dominates all others. The term usually referred to as **maximum clique size**. We will see the reason behind this terminology later.

Determining Complexity of Inference

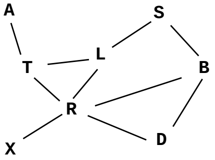
- It is often desirable to know the complexity of inference beforehand.
- In the next few slides, we show how the complexity of VE can easily be determined from network structure.

Structural Graph of Factorization

- Given a list $\mathcal{F}$ of function, the **structural graph** of $\mathcal{F}$ is an undirected graph obtained as follows:

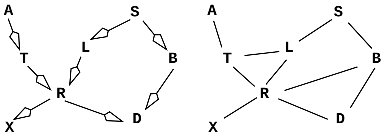
  For any two variables X and Y, connect them iff they appear in the same factor.

- Example:
  - $\mathcal{F} = \{P(A), P(T|A), P(S), P(L|S), P(B|S), P(R|T,L), P(X|R), P(D|R,B)\}$
  - The structural graph of $\mathcal{F}$ is:



Moral Graph of DAG

- The **moral** graph $m(G)$ of a DAG is the undirected graph obtained from G by
  - Marrying the parents of each node (i.e adding an edge between each pair of parents), and
  - Dropping all directions.



- Note: If $\mathcal{F}$ is the list of CPTs of a BN, then the structural graph of $\mathcal{F}$ is simply the moral graph of the BN.

$$\mathcal{F} = \{P(A), P(T|A), P(S), P(L|S), P(B|S), P(R|T, L), P(X|R), P(D|R, B)\}$$

Cost of Eliminating One Variable

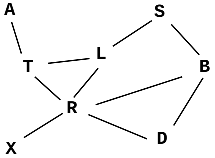
- For any vertex Z in an undirected graph, let $adj(Z)$ be the set of all neighbors of Z.
- Fact 1: If G is the structural graph of $\mathcal{F}$, the cost of eliminating Z from $\mathcal{F}$ is given by

$$c(Z) = w(Z) \prod_{X \in adj(Z)} w(X).$$

- Why?Recall
  1. Remove from the $\mathcal{F}$ all the functions, say $f_1, \ldots, f_k$, that involve Z,
  2. Compute new function $g = \prod_{i=1}^{k} f_i$.
  3. ...

Cost of Eliminating One Variable

- $\mathcal{F} = \{P(A), P(T|A), P(S), P(L|S), P(B|S), P(R|T,L), P(X|R), P(D|R,B)\}$
- The structural graph of $\mathcal{F}$ is:



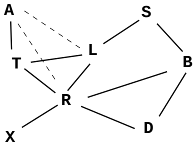
- Eliminating T:
  - Needs to compute: $P(T|A)P(R|T,L)$
  - Cost: $c(T) = w(T)w(A)w(R)w(L)$
- $adj(T) = \{A, R, L\}$.
- So,
$$c(T) = w(T) \prod_{X \in adj(T)} w(X).$$

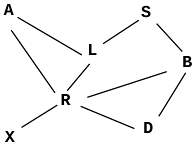
Eliminating Vertex from Graph

- **Eliminating** a vertex Z from an undirected graph G means:
  - Adding edges so that all nodes in $adj(Z)$ are pairwise adjacent,and
  - Removing Z and its incident edges.

  Denote the result graph by $\texttt{eliminate}(G, Z)$.

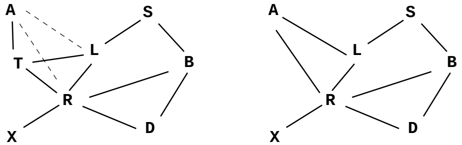
- Example: $\texttt{eliminate}(G, T)$ is





Elimination in Factorization and Elimination in Graph

- Fact 2: If G is the structural graph of $\mathcal{F}$, then $\texttt{eliminate}(G, Z)$ is the structural graph of $\texttt{eliminate}(\mathcal{F}, Z)$.
- Example:
  - $\texttt{eliminate}(\mathcal{F}, T) =$ $\{P(A), P(S), P(L|S), P(B|S), P(X|R), P(D|R, B), \psi(A, L, R)\}$, where $\psi(A, L, R) = \sum_T P(T|A)P(R|T, L)$.
  - $\texttt{eliminate}(G, T)$ is



  - We see that $\texttt{eliminate}(G, T)$ is the graph for $\texttt{eliminate}(\mathcal{F}, T)$.

Determining the Complexity of VE

- Fact 1 and Fact 2 allow us to determine the complexity of VE by manipulating graphs.

$$
\begin{array}{ccccccc}
 & Z_1 & & Z_2 & & Z_3 & \\
\mathcal{F}_1 & \rightarrow & \mathcal{F}_2 & \rightarrow & \mathcal{F}_3 & \rightarrow & \ldots \\
||| & & ||| & & ||| & & \\
G_1 & \rightarrow & G_2 & \rightarrow & G_3 & \rightarrow & \ldots \\
\hline
 & c(Z_1) & & c(Z_2) & & c(Z_3) &
\end{array}
$$

- There are no numerical calculations in the process. It is fast.

Determining the Complexity of VE

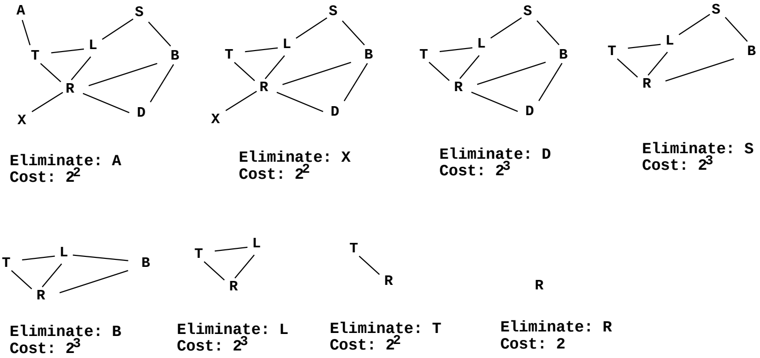
Procedure costVE($\mathcal{N}, \mathbf{E}, \rho$)

- Inputs: $\mathcal{N}$ — A Bayesian network structure.
  E — Set of observed variables.
  ρ — An elimination ordering.
- Output: complexity of VE.

1. Compute moral graph $\mathcal{G}$ of $\mathcal{N}$.
2. Remove from $\mathcal{G}$ all nodes in **E**.
   // Structural graph of $\mathcal{F}$ after step 1 of VE
3. $C = 0$.
4. **While** ρ is not empty,
   1. Remove the first variable Z from ρ,
   2. $C \mathrel{+}= w(Z) \prod_{X \in adj(Z)} w(X)$.
   3. eliminate($\mathcal{G}, Z$).
5. Return C

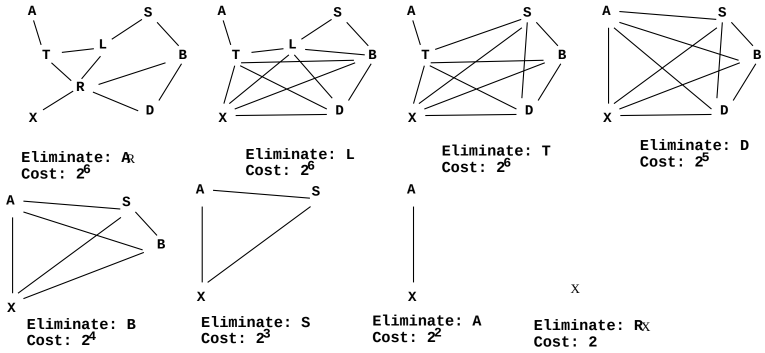
Example of costVE

Example 1: A, X, D, B, S, L, T, R



Example of costVE

Example 2: R, L, T, D, B, S, A, X



Optimal Elimination Ordering

- Different elimination orderings lead to different costs.
- The **optimal elimination ordering**: the one with minimum cost.
- It is NP-hard to find an optimal elimination ordering (Arnborg *et al*, 1987).
- The best we can hope for are some heuristics.
- Will give some heuristics in Lecture 4.1.