VU VRIJE UNIVERSITEIT AMSTERDAM

Bayesian Networks

Knowledge Representation

- Andreas Sauter
- Nov. 2022
- (Content adapted from Erman Acar)

Overview

1. Foundations

    Degrees of Belief, Belief Dynamics, Independence, Bayes Theorem, Marginalization

2. Bayesian Networks

    Graphs and their Independencies, Bayesian Networks, d-Separation

3. Tools for Inference

    Factors, Variable Elimination, Elimination Order, Interaction Graphs, Graph pruning

4. Exact Inference in Bayesian Networks

    Posterior Marginal, Maximum – A-posteriori, Most Probable Explanation

Lecture 3: Tools for Inference

Lecture Overview

Factors

Definition, Marginalization, Multiplication

Variable Elimination

Motivation, Process of Elimination, Prior Marginals

Elimination Order

Interaction Graphs, Order Width, Min Degree Order, Min Fill Order

Network Pruning

Edge Pruning, Node Pruning

Factors

Definition

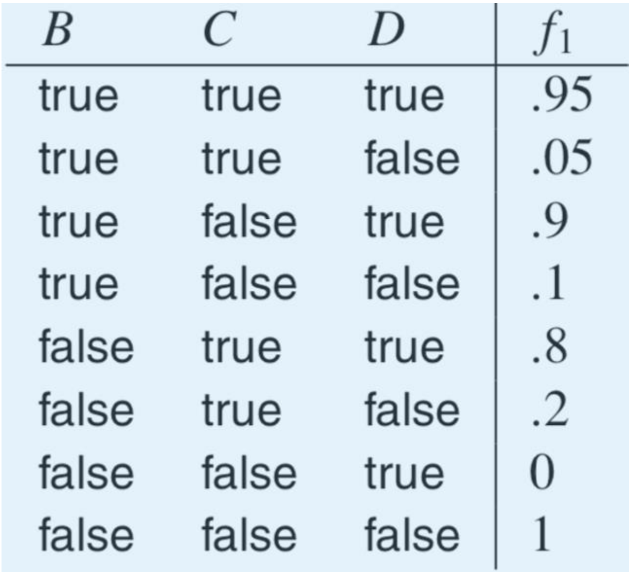
In BN inference we deal with probabilities, conditional probabilities, marginalizations, ect. Factors can describe them all.

Given a set of variables X, a factor f is a function mapping each instantiation to a non-negative number.

This number is not constraint to be <1, hence a factor does not necessarily represent a probability distribution.

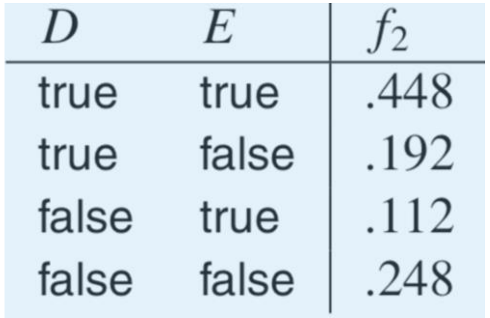
We allow factors to be defined over an empty set of variables and will call such a factor a trivial factor.

Examples



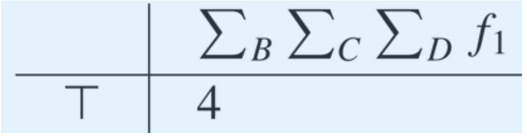
B	C	D	f_1
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

Represents $f_1(B,C,D) = \Pr(D|B,C)$



D	E	f_2
true	true	.448
true	false	.192
false	true	.112
false	false	.248

Represents $f_1(D,E) = \Pr(D,E)$



	$\sum_B \sum_C \sum_D f_1$
$\top$	4

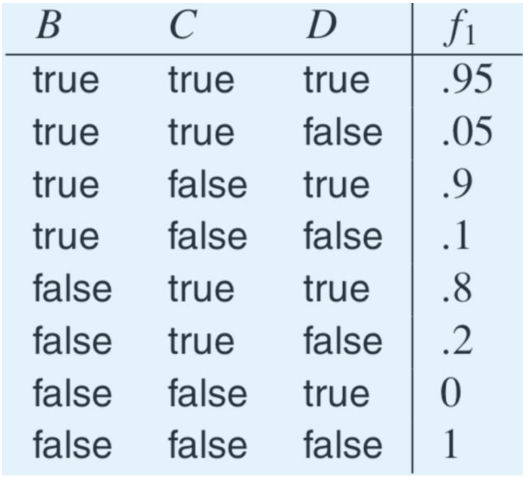
Represents the trivial factor

Summing-Out/Marginalization of Factors

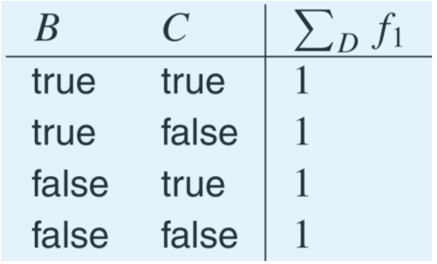
We want to determine the value for a subset of variables by doing a case analysis.

Given a factor f over variables $\boldsymbol{X}$, summing-out variable $X \in \boldsymbol{X}$ results in another factor $(\Sigma_X f)(\boldsymbol{Y}) \overset{\text{def}}{=} \Sigma_X f(X, \boldsymbol{Y})$, where $\boldsymbol{Y} = \boldsymbol{X} \backslash \mathrm{X}$.

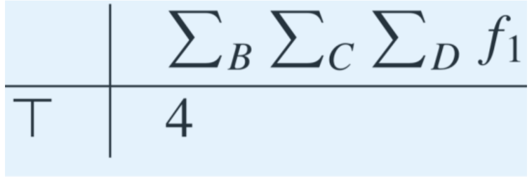
Example: Marginalizing D from f_1 resulting in $\Sigma_D f_1$ and marginalizing all variables resulting in $\Sigma_B \Sigma_C \Sigma_D f_1$.



B	C	D	f_1
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1



B	C	$\sum_D f_1$
true	true	1
true	false	1
false	true	1
false	false	1



	$\sum_B \sum_C \sum_D f_1$
$\top$	4

Remarks

Marginalization is commutative:

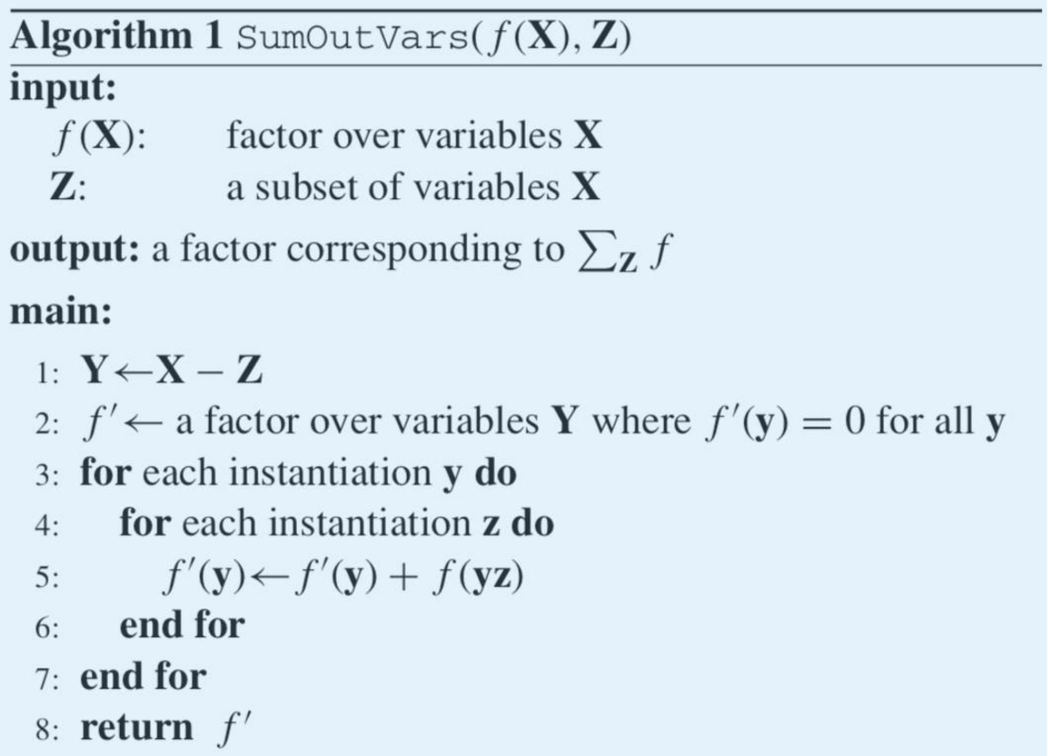
$$\Sigma_Y \Sigma_X f = \Sigma_X \Sigma_Y f$$

For a set of variables $\boldsymbol{Z} \in \boldsymbol{X}$ we can simply write $\Sigma_{\boldsymbol{Z}} f$.

If the variable X appears only in f_2:

$$\Sigma_X f_1 f_2 = f_1 \Sigma_X f_2$$

The time and space complexity of marginalization is exponential in $|\boldsymbol{X}|$.



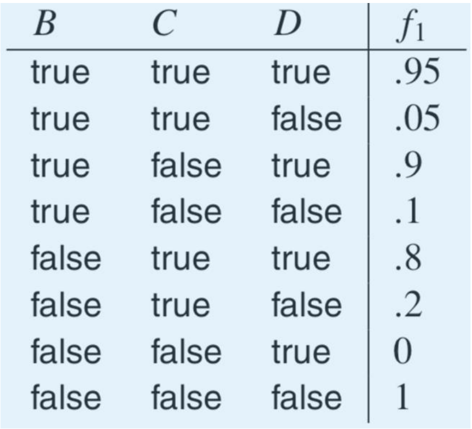
```
Algorithm 1 SumOutVars($f(\mathbf{X}), \mathbf{Z}$)
input:
    $f(\mathbf{X})$:    factor over variables $\mathbf{X}$
    $\mathbf{Z}$:       a subset of variables $\mathbf{X}$
output: a factor corresponding to $\sum_{\mathbf{Z}} f$
main:
    $\mathbf{Y} \leftarrow \mathbf{X} - \mathbf{Z}$
    $f' \leftarrow$ a factor over variables $\mathbf{Y}$ where $f'(\mathbf{y}) = 0$ for all $\mathbf{y}$
    for each instantiation $\mathbf{y}$ do
        for each instantiation $\mathbf{z}$ do
            $f'(\mathbf{y}) \leftarrow f'(\mathbf{y}) + f(\mathbf{yz})$
        end for
    end for
    return $f'$
```

Multiplication of Factors

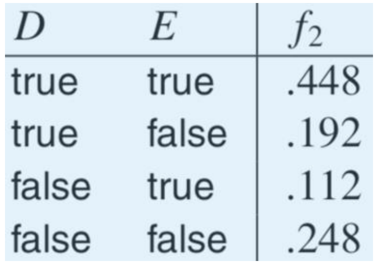
We can also multiply factors which will come in handy when we need the chain rule.

Given two factors f_1, f_2 over variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively, multiplying f_1, f_2 results in a new factor over variables $\boldsymbol{Z} = \boldsymbol{X} \cup \boldsymbol{Y}$ s.t. $(f_1 f_2)(\boldsymbol{z}) \overset{\text{def}}{=} f_1(\boldsymbol{x}) f_2(\boldsymbol{y})$, where $\boldsymbol{x}, \boldsymbol{y} \sim \boldsymbol{z}$.

Example:

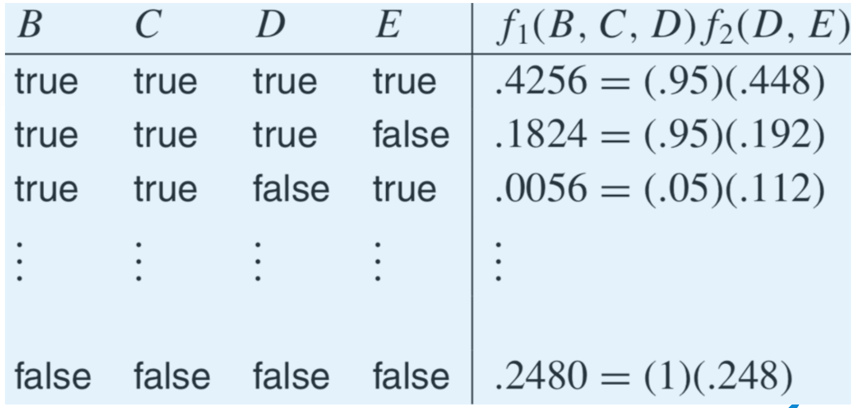


B	C	D	f_1
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1



D	E	f_2
true	true	.448
true	false	.192
false	true	.112
false	false	.248

$(f_1 f_2)(\boldsymbol{Z}) =$



B	C	D	E	$f_1(B, C, D) f_2(D, E)$
true	true	true	true	$.4256 = (.95)(.448)$
true	true	true	false	$.1824 = (.95)(.192)$
true	true	false	true	$.0056 = (.05)(.112)$
⋮	⋮	⋮	⋮	⋮
false	false	false	false	$.2480 = (1)(.248)$

Remarks

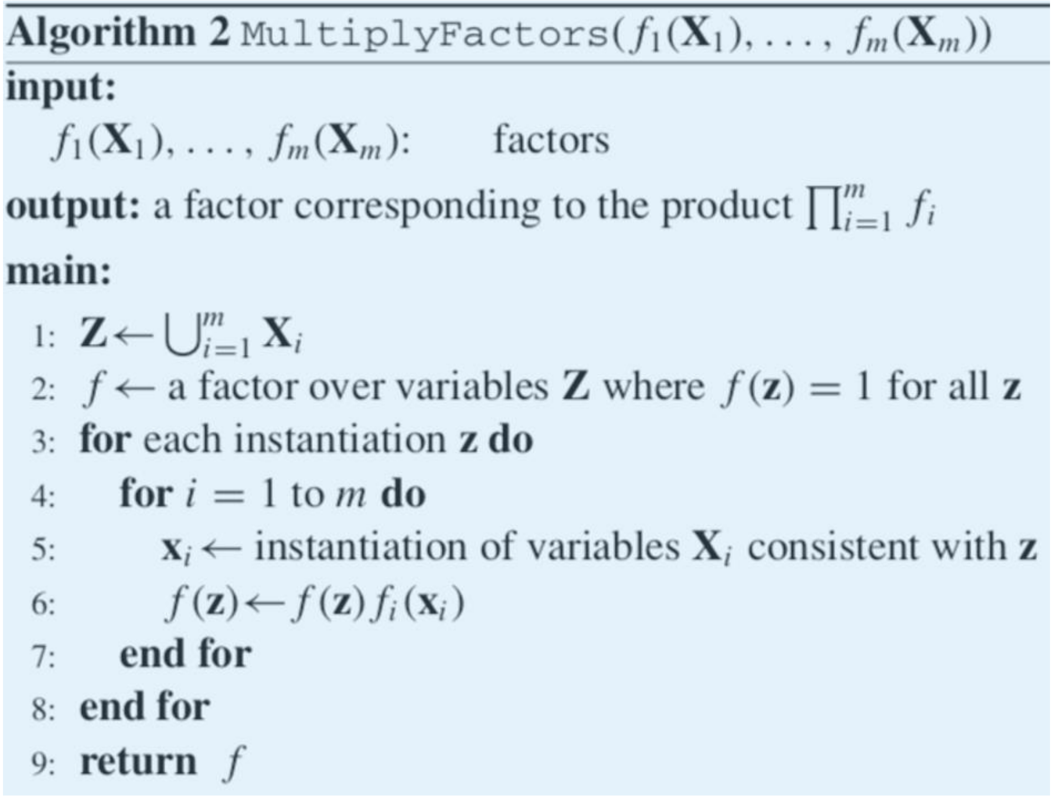
Factor multiplication is commutative:

$$f_1 f_2 = f_2 f_1$$

Factor multiplication is associative:

$$f_1(f_2 f_3) = (f_1 f_2) f_3$$

Factor multiplication is exponential in time and space w.r.t. $|\boldsymbol{Z}|$.



Algorithm 2 `MultiplyFactors`$(f_1(\mathbf{X}_1), \ldots, f_m(\mathbf{X}_m))$

```
input:
    $f_1(\mathbf{X}_1), \ldots, f_m(\mathbf{X}_m)$:    factors
output: a factor corresponding to the product $\prod_{i=1}^{m} f_i$
main:
  $\mathbf{Z} \leftarrow \bigcup_{i=1}^{m} \mathbf{X}_i$
  $f \leftarrow$ a factor over variables $\mathbf{Z}$ where $f(\mathbf{z}) = 1$ for all $\mathbf{z}$
  for each instantiation $\mathbf{z}$ do
      for $i = 1$ to $m$ do
          $\mathbf{x}_i \leftarrow$ instantiation of variables $\mathbf{X}_i$ consistent with $\mathbf{z}$
          $f(\mathbf{z}) \leftarrow f(\mathbf{z}) f_i(\mathbf{x}_i)$
      end for
  end for
  return $f$
```

Variable Elimination

Motivation

A BN encodes the joint probability over a set of variables.

For some computations/inferences we are only interested in the distribution over a subset of those variables.

We can do this through marginalization which will have a exponential runtime in the number of variables.

Variable elimination is an algorithm that lets us consider only the relevant variables of a BN while sometimes avoiding exponential runtime.

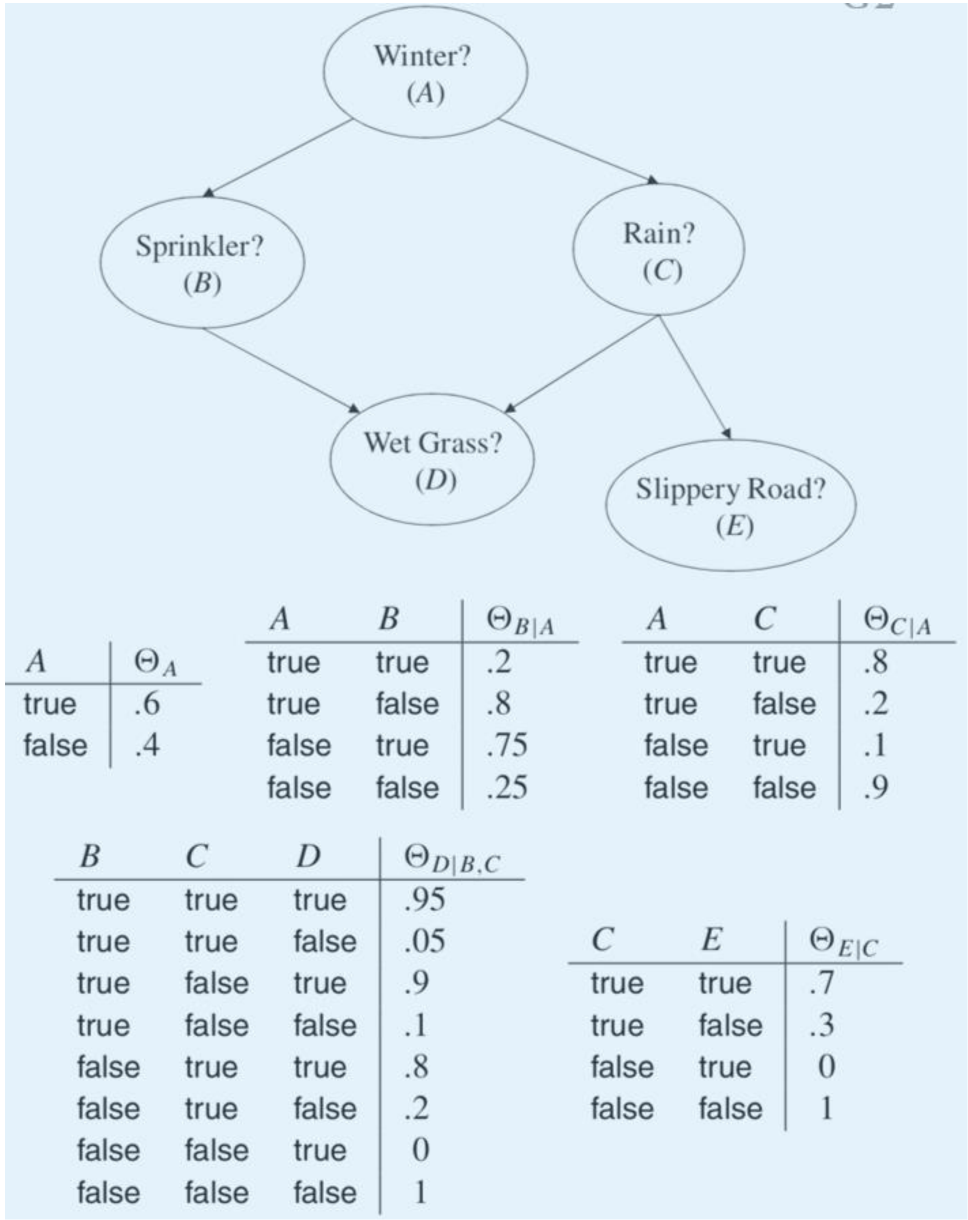
Approach so far

We want the prior marginal $P(D, E)$.

First, we have to compute the joint distribution $P(A, B, C, D, E)$ via the chain rule.

Then we can marginalize $A, B, and\ C$.

Yes! But, …



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|B,C}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

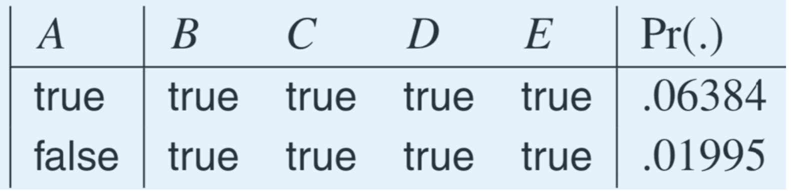
C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

Approach so far - Continued

The joint distribution $\Pr(A, B, C, D, E) =$

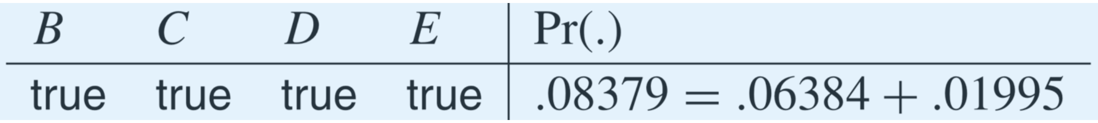
Now, marginalize $A, B, and\ C$ one by one.

E.g. summing out A from the following rows:

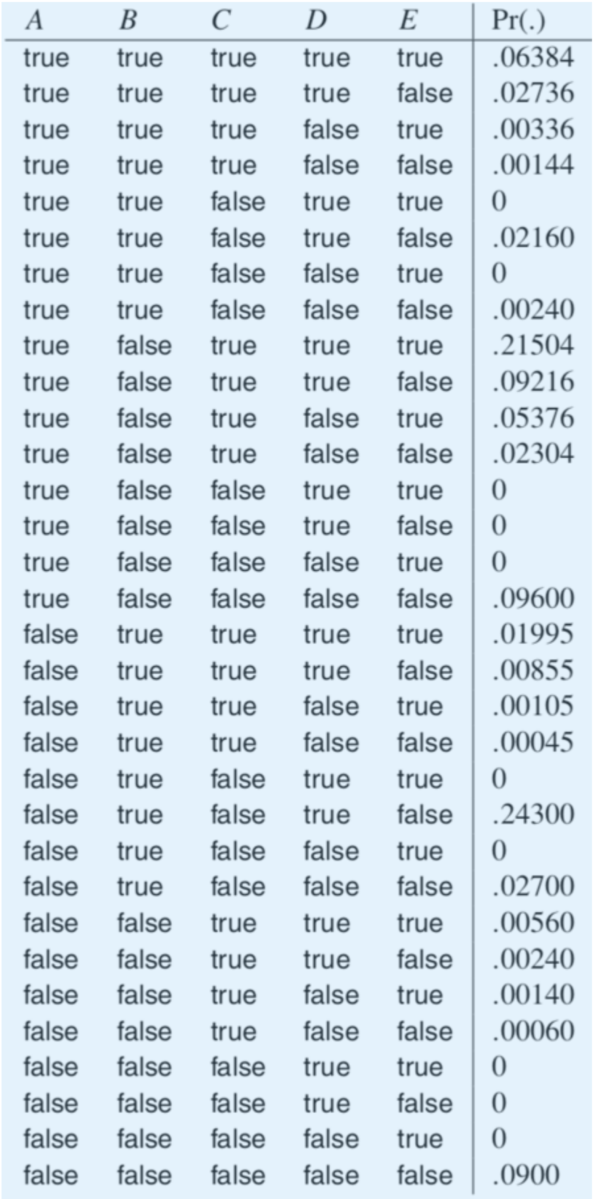


A	B	C	D	E	Pr(.)
true	true	true	true	true	.06384
false	true	true	true	true	.01995

Merges into:



B	C	D	E	Pr(.)
true	true	true	true	$.08379 = .06384 + .01995$



A	B	C	D	E	Pr(.)
true	true	true	true	true	.06384
true	true	true	true	false	.02736
true	true	true	false	true	.00336
true	true	true	false	false	.00144
true	true	false	true	true	0
true	true	false	true	false	.02160
true	true	false	false	true	0
true	true	false	false	false	.00240
true	false	true	true	true	.21504
true	false	true	true	false	.09216
true	false	true	false	true	.05376
true	false	true	false	false	.02304
true	false	false	true	true	0
true	false	false	true	false	0
true	false	false	false	true	0
true	false	false	false	false	.09600
false	true	true	true	true	.01995
false	true	true	true	false	.00855
false	true	true	false	true	.00105
false	true	true	false	false	.00045
false	true	false	true	true	0
false	true	false	true	false	.24300
false	true	false	false	true	0
false	true	false	false	false	.02700
false	false	true	true	true	.00560
false	false	true	true	false	.00240
false	false	true	false	true	.00140
false	false	true	false	false	.00060
false	false	false	true	true	0
false	false	false	true	false	0
false	false	false	false	true	0
false	false	false	false	false	.0900

Main Idea of Variable Elimination

We want to leverage the structure of the BN for marginalization.

Considering the probability tables as factors.

Recall: we can compute the joint distribution as a factor multiplication via the chain rule.

We can restrict summing-out of a variables on just those factors in which this variable appears.

This allows us to sum out over smaller probability tables.

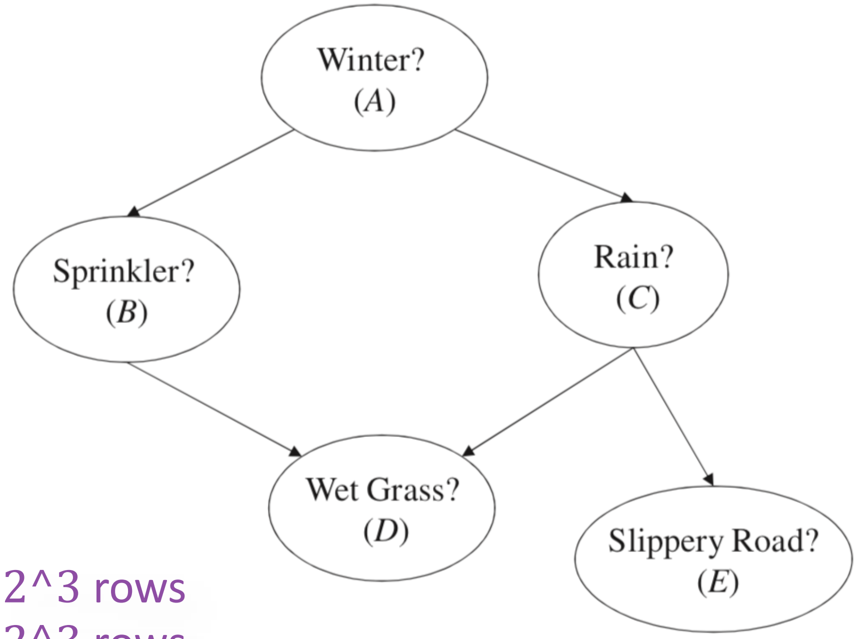
Example

As before, the prior marginal is:

$$\Pr(D,E) = \boldsymbol{\Sigma}_{\boldsymbol{A,B,C}} \underbrace{\Theta_{E|C}\Theta_{D|C,B}\Theta_{C|A}\Theta_{B|A}\Theta_{A}}_{\text{2\^{}5 rows}}$$

Because the tables are factors, we can express this as:

$$\Pr(D,E) = \boldsymbol{\Sigma}_{\boldsymbol{C}} \underbrace{\Theta_{E|C} \boldsymbol{\Sigma}_{\boldsymbol{B}} \underbrace{\Theta_{D|C,B} \boldsymbol{\Sigma}_{\boldsymbol{A}} \underbrace{\Theta_{C|A}\Theta_{B|A}\Theta_{A}}_{\text{2\^{}3 rows}}}_{\text{2\^{}3 rows}}}_{\text{2\^{}3 rows}}$$

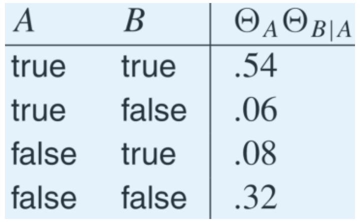


Which keeps the tables resulting from the factor multiplications smaller.

Example

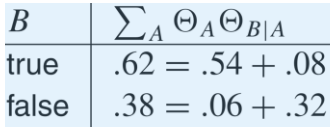
We want to compute $\Pr(C) = \Sigma_B \Theta_{C|B} \Sigma_A \Theta_A \Theta_{B|A}$

First, compute $\Theta_A \Theta_{B|A}$:



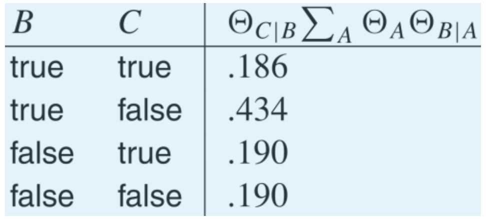
A	B	$\Theta_A \Theta_{B\|A}$
true	true	.54
true	false	.06
false	true	.08
false	false	.32

Then sum-out A to obtain $\Sigma_A \Theta_A \Theta_{B|A}$:



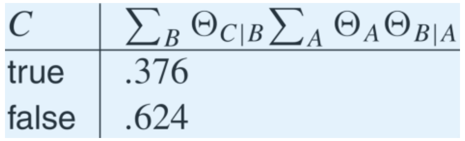
B	$\sum_A \Theta_A \Theta_{B\|A}$
true	$.62 = .54 + .08$
false	$.38 = .06 + .32$

Multiplied with $\Theta_{C|B}$:

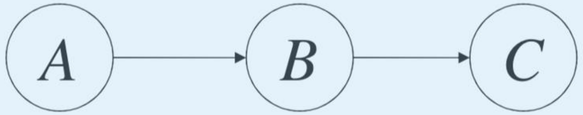


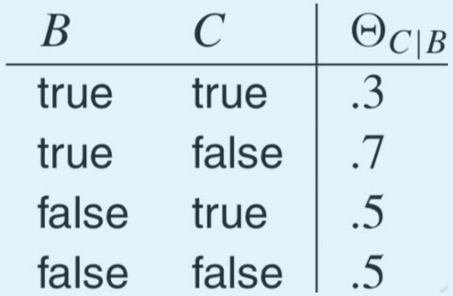
B	C	$\Theta_{C\|B} \sum_A \Theta_A \Theta_{B\|A}$
true	true	.186
true	false	.434
false	true	.190
false	false	.190

And, finally, sum-out B to get $\Pr(C)$:

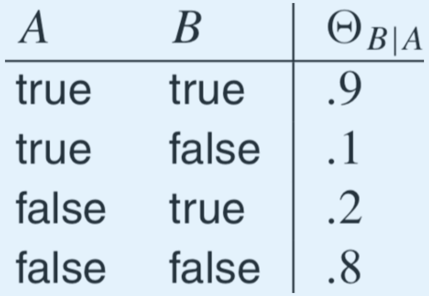


C	$\sum_B \Theta_{C\|B} \sum_A \Theta_A \Theta_{B\|A}$
true	.376
false	.624

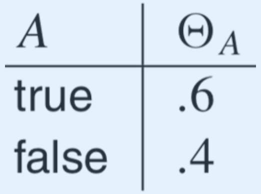




B	C	$\Theta_{C\|B}$
true	true	.3
true	false	.7
false	true	.5
false	false	.5



A	B	$\Theta_{B\|A}$
true	true	.9
true	false	.1
false	true	.2
false	false	.8



A	Θ_A
true	.6
false	.4

Elimination Order

Motivation

In the previous examples we implicitly decided which variable we sum-out first.

In principle, every order gives us the same result.

e.g: $\Sigma_B \Theta_{C|B} \Sigma_A \Theta_A \Theta_{B|A} = \Sigma_A \Theta_A \Sigma_B \Theta_{B|A} \Theta_{C|B}$

But the amount of variables appearing in each factor influences the runtime.

In this example, first summing-out A results in smaller factors.

We want to choose the elimination order which most efficient.

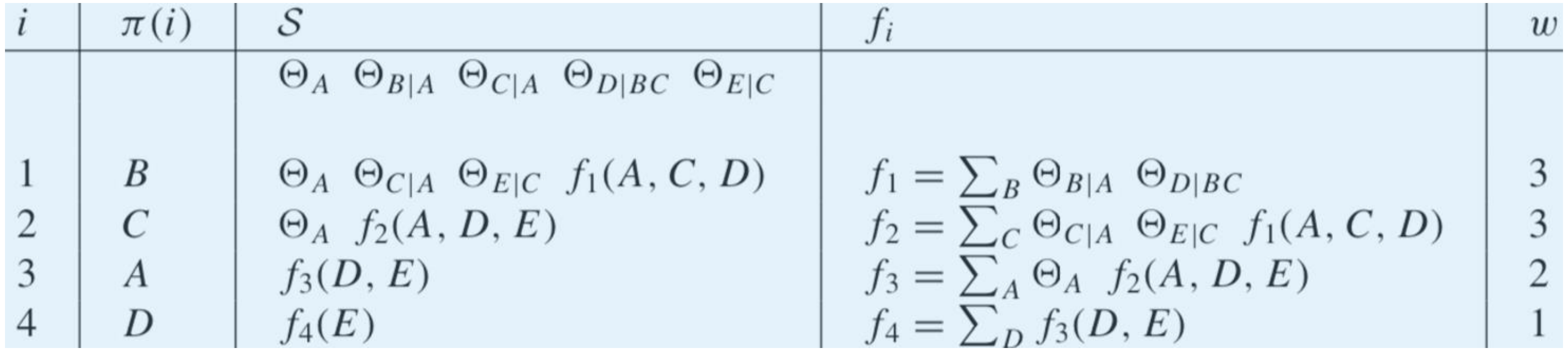
Order Width

To choose a good elimination order often the order width w is used as a metric.

The width of a factor is the number of variables that appear in that factor.

The order width is the maximum width of the factors needed for elimination.

Example: Computing $\Pr(E)$ given ordering $\pi = (B, C, A, D)$:



i	$\pi(i)$	$\mathcal{S}$	f_i	w
		$\Theta_A \;\; \Theta_{B\|A} \;\; \Theta_{C\|A} \;\; \Theta_{D\|BC} \;\; \Theta_{E\|C}$		
1	B	$\Theta_A \;\; \Theta_{C\|A} \;\; \Theta_{E\|C} \;\; f_1(A, C, D)$	$f_1 = \sum_B \Theta_{B\|A} \;\; \Theta_{D\|BC}$	3
2	C	$\Theta_A \;\; f_2(A, D, E)$	$f_2 = \sum_C \Theta_{C\|A} \;\; \Theta_{E\|C} \;\; f_1(A, C, D)$	3
3	A	$f_3(D, E)$	$f_3 = \sum_A \Theta_A \;\; f_2(A, D, E)$	2
4	D	$f_4(E)$	$f_4 = \sum_D f_3(D, E)$	1

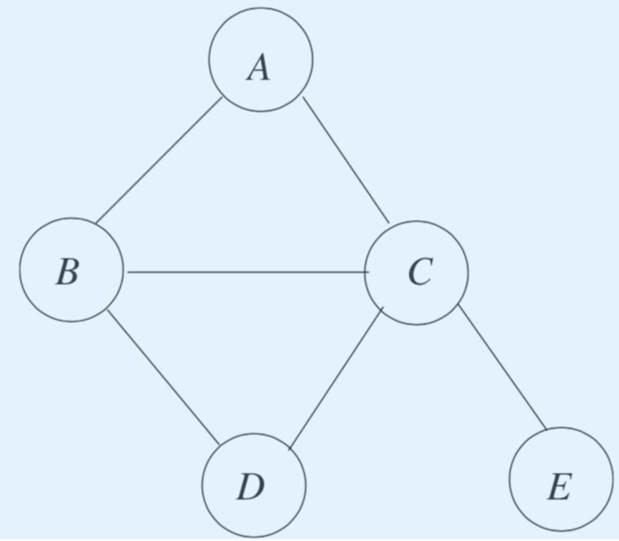
$width(\pi) = 3$

Interaction Graphs

An easier way of visualizing the width is via interaction graphs.

The interaction graph G over factors $f_1, \dots, f_n$ is an undirected graph constructed as follows:

- The nodes of G are the variables that appear in factors $f_1, \dots, f_n$.
- The edges connect the variables that appear in the same factor.



The width of a factor that sums-out $X \in V$ is then the degree of X in the interaction graph.

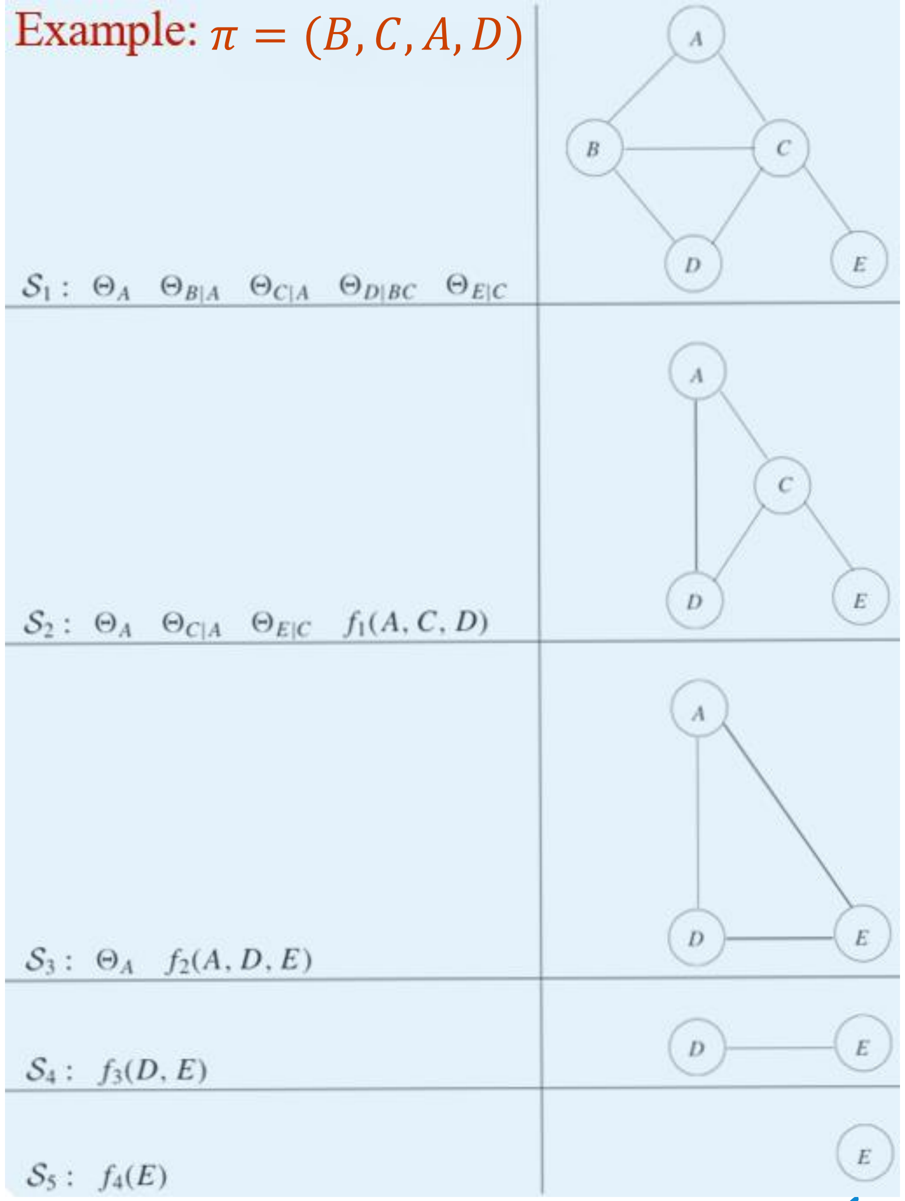
Example: $\Theta_A \Theta_{B|A} \Theta_{C|A} \Theta_{D|B,C} \Theta_{E|C}$ has the interaction graph

Interaction Graphs - Continued

Summing-out X, leads to a new interaction graph G'. It can be constructed as follows:

- Add an edge between every neighbour of X that is not already connected by an edge. ← factor multiplication
- Delete the node X. ← summing-out X

The order width of $\pi = (B, C, A, D)$ is then the maximum degree of any node that was summed-out.


Example: $\pi = (B, C, A, D)$
$S_1: \Theta_A \quad \Theta_{B|A} \quad \Theta_{C|A} \quad \Theta_{D|BC} \quad \Theta_{E|C}$
$S_2: \Theta_A \quad \Theta_{C|A} \quad \Theta_{E|C} \quad f_1(A, C, D)$
$S_3: \Theta_A \quad f_2(A, D, E)$
$S_4: f_3(D, E)$
$S_5: f_4(E)$
A
B
C
D
E

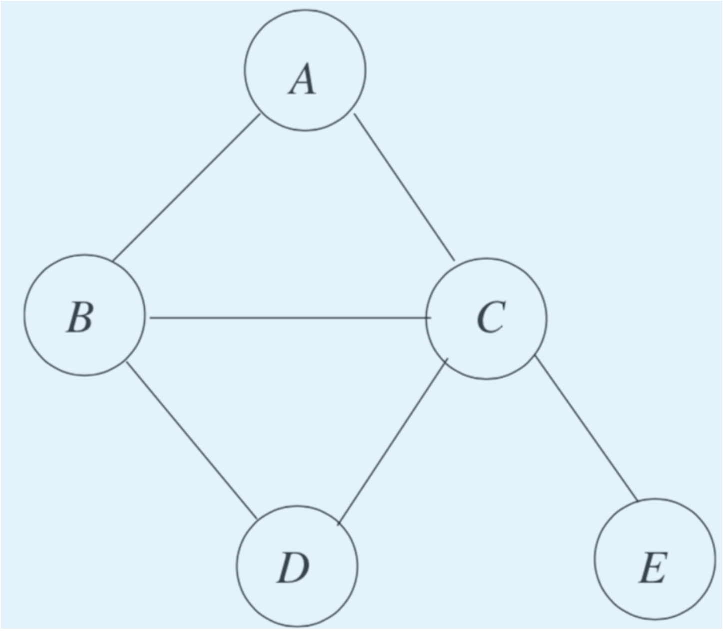
Heuristics for Choosing an Elimination Order

Minimum degree ordering of $\boldsymbol{X} \subseteq V$:

- Queue variable $X \in \boldsymbol{X} \subseteq V$ with the minimum degree in the interaction graph to the ordering.
- Sum-out X from the interaction graph.
- Repeat until all variables $\boldsymbol{X}$ are summed-out.

Minimum fill ordering of $\boldsymbol{X} \subseteq V$:

- Queue variable $X \in \boldsymbol{X}$ whose deletion would add the fewest new interactions to the ordering.
- Sum-out X from the interaction graph.
- Repeat until all variables $\boldsymbol{X}$ are summed-out.



Network Pruning

Motivation

The structure of the graph can have a significant influence on the performance of answering queries.

The inference complexity can be highly influenced by the number and location of query variables Q and evidence e.

Therefore, we want to simplify the network structure as much as possible without influencing the result of the query.

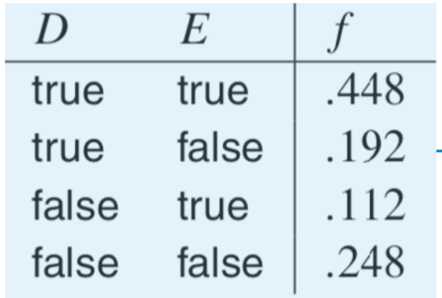
Factor Reduction

Given a factor f and an evidence e, the reduced factor f^e can be obtained by zeroing the rows which are incompatible with the evidence.

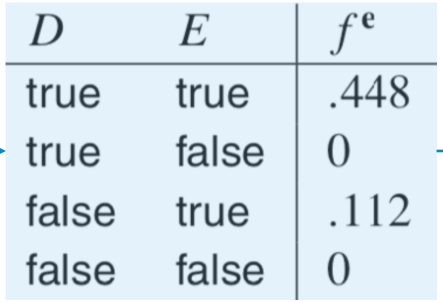
$$f^e(x) \overset{\text{def}}{=} \left\{ \begin{matrix} f(x), & if\ x \sim e \\ 0, & otherwise \end{matrix} \right\}$$

Note that reduction is distributive: $(f_1 f_2 \dots f_n)^e = f_1^e f_2^e \dots f_n^e$

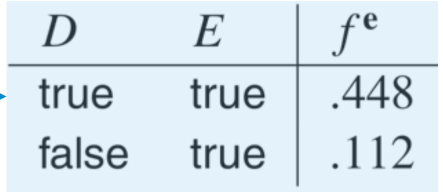
Example: f given $E = true$:



D	E	f
true	true	.448
true	false	.192
false	true	.112
false	false	.248



D	E	f^e
true	true	.448
true	false	0
false	true	.112
false	false	0



D	E	f^e
true	true	.448
false	true	.112

Edge Pruning

We can delete the outgoing edges of every node that is in the evidence e without affecting the result of a query Q.

More precisely, for each edge $U \rightarrow X$ that originates from a node U in e:

- Remove the edge $U \rightarrow X$ from the network.
- Replace the factor $\Theta_{X|Pa_G(X)}$ by a reduced factor $\Theta^{e}_{X|Pa_G(X)}$
- Remark: The resulting factor corresponds to $\Sigma_U \Theta_{X|u,Pa_G(X)\setminus \mathrm{U}}$

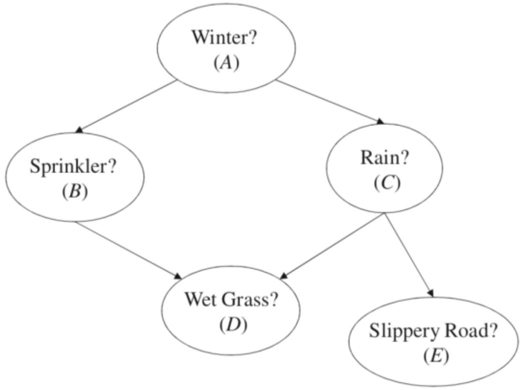
Given an BN N, an evidence e, and a pruned network $N' = pruneEdges(N, e)$, then $\Pr(Q|e) = Pr'(Q|e)$ where Pr and Pr' are the probabilities induced by networks N and N', respectively.

Example

Given evicence $C = false$:

The corresponding edges were deleted and the tables $\Theta_{D|B,C}, \Theta_{E|C}$ updated.

The pruned network is only good for answering queries of the form $\Pr(Q|e) = \Pr(A, B, D, E|\neg c)$



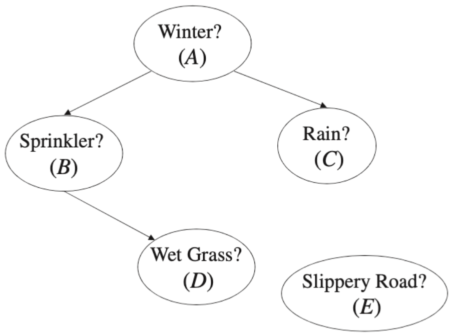
A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|B,C}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	D	$\sum_C \Theta^{C=\text{false}}_{D\|BC}$
true	true	.9
true	false	.1
false	true	0
false	false	1

E	$\sum_C \Theta^{C=\text{false}}_{E\|C}$
true	0
false	1

Node Pruning

We can delete any leaf node that doesn’t appear in Q or e.

We can do this iteratively until no node can be deleted.

This does not affect the ability of the network to answer the query thanks to the following result:

Let N be a Bayesian network and (Q, e) a corresponding query.
If $N' = pruneNodes(N, (Q, e))$, then $\Pr(Q|e) = \Pr'(Q|e)$ where Pr and Pr' are the distributions induced by N and N' , respectively

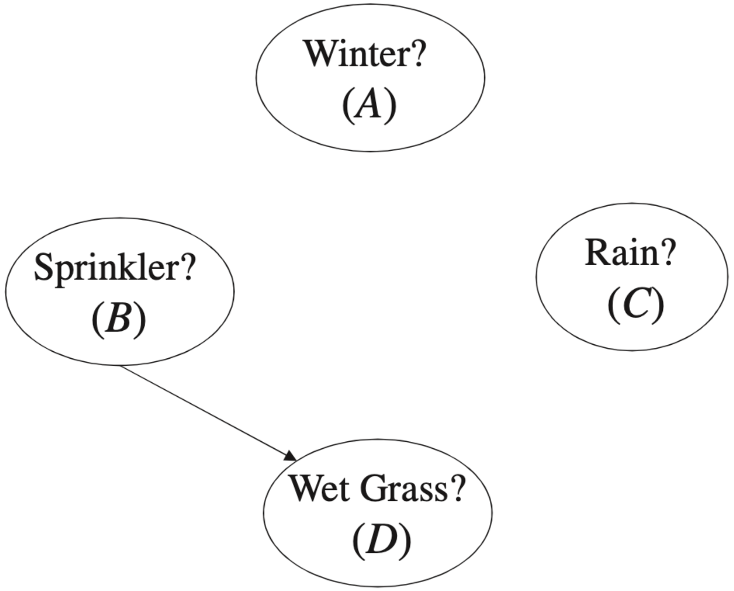
Example – Edge+Node pruning

Example: $Q = D, e = (A = true, C = false)$

First edge pruning w.r.t. e.

Then delete leaf nodes iteratively.

We significantly reduced the size of the network and, therefore, inference complexity.



B	$\Theta'_B = \sum_A \Theta_{B\|A}^{A=\text{true}}$
true	.2
false	.8

C	$\Theta'_C = \sum_A \Theta_{C\|A}^{A=\text{true}}$
true	.8
false	.2

B	D	$\Theta'_{D\|B} = \sum_C \Theta_{D\|BC}^{C=\text{false}}$
true	true	.9
true	false	.1
false	true	0
false	false	1

A	Θ_A
true	.6
false	.4

Lecture 3 – Summary

- We introduced factors, a useful object for inference in BN.
- We showed how we can compute marginals more efficiently with variable elimination.
- We considered different evaluation orderings to further increase performance.
- We discovered how we can prune a BN to reduce complexity while maintaining its ability to answer certain queries.