COMP9418: Advanced Topics in Statistical Machine Learning

MAP Inference

Instructor: Gustavo Batista

University of New South Wales

Introduction

- In this lecture, we study algorithm to compute queries of the form
  - MAP: maximum a posteriori hypothesis
  - MPE: maximum a posteriori explanation
- In these queries, we are interested in finding the most probable instantiations of a subset of variables
- We discuss algorithms to compute MAP and MPE based on different strategies such as
  - Variable elimination
  - Systematic search
  - Belief propagation

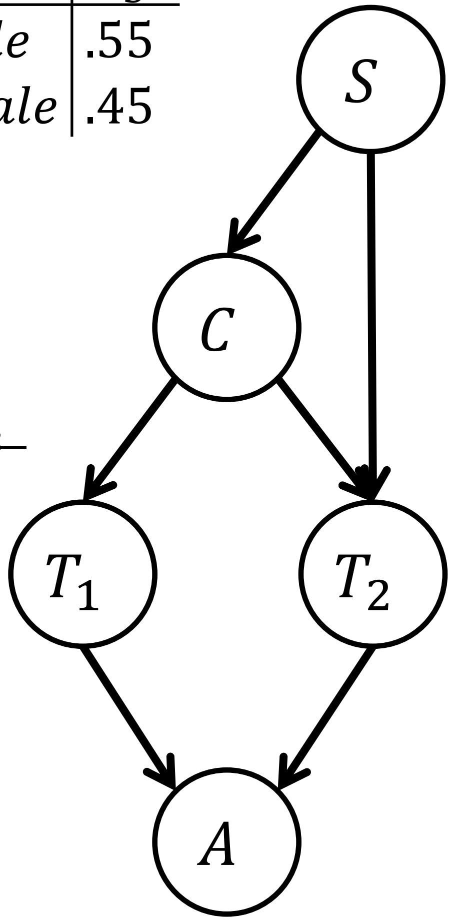
Introduction: Example

- Consider a Bayesian network on the right
  - It concerns a population of 55% males and 45% females
  - They can suffer of a medical condition C that is more likely in males
  - There are two diagnosis tests for C, T_1 and T_2
  - T_2 is more effective on females
  - Both tests are equally effective on males

S	C	$\Theta_{S\|C}$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\Theta_{T_1\|C}$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	Θ_S
$male$	.55
$female$	.45

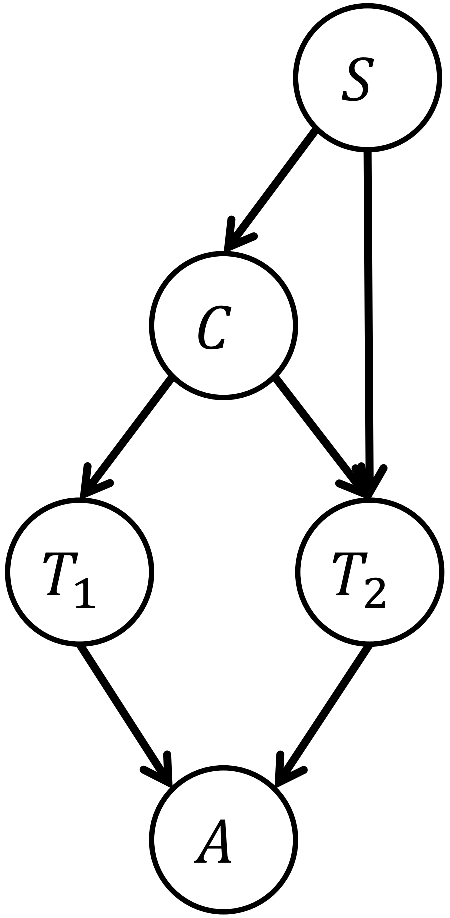


S	C	T_2	$\Theta_{T_2\|C,S}$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\Theta_{A\|T_1,T_2}$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0

Introduction: Example

- We can partition this population in four groups
  - Males and females, with or without the condition
- Suppose a person takes both tests with the same results
  - Leads to the evidence $A = yes$
- What is the most likely group this individual belongs?
  - This is an example of MAP instantiation
  - The most likely instantiation of S and C given $A = yes$
  - In this query, S and C are *MAP variables*
- The answer for this example is
  - $S = male$ and $C = no$ with posterior probability of $\sim 49.3\%$



MAP and Inference

- Variable and factor elimination algorithms can compute MAP instantiations
  - They are efficient with small number of MAP variables
  - We compute the posterior marginal over MAP variables and select the instantiation with maximal probability
- However, this approach is exponential in the number of MAP variables
- Our objective in this lecture is to present algorithms for MAP instantiations
  - Not necessarily exponential in the number of MAP variables

MAP and MPE

- MPE is a special case of MAP when MAP variables contain all unobserved network variables
  - In the previous example, it would result in 16 groups
  - Males and females, with or without the condition and the four possible outcomes for the two tests
- This is the MAP instantiation for S, C, T_1 and T_2
  - The answer is $S = female, C = no,$ $T_1 = \overline{ve}, T_2 = \overline{ve}$
  - With posterior probability $\sim 47\%$
- This case of MAP is known as *MPE instantiation*
  - MPE instantiations are much easier to compute than MAP
  - That is why they have their own name
- MPE is not the answer for MAP
  - MPE projection on variables S and C is $S = female, C = no$
  - But the MAP answer of previous slides is $S = male$ and $C = no$
- Although this technique is sometimes used as an approximation for MAP

Computing MPE

- Given a network
  - The *MPE probability* for the variables $\boldsymbol{Q}$ of a network given evidence $\boldsymbol{e}$ is defined as

$$MPE_P(\boldsymbol{e}) \stackrel{\text{def}}{=} \max_{\boldsymbol{q}} P(\boldsymbol{q}, \boldsymbol{e})$$

- There may be several instantiations $\boldsymbol{q}$ with maximal probability
  - Each of them is an *MPE instantiation*
  - The set of such instantiations is defined as

$$MPE(\boldsymbol{e}) \stackrel{\text{def}}{=} \underset{\boldsymbol{q}}{argmax}\, P(\boldsymbol{q}, \boldsymbol{e})$$

- MPE instantiations can be characterized as instantiations $\boldsymbol{q}$ that maximize the posterior distribution

$$MPE(\boldsymbol{e}) \stackrel{\text{def}}{=} \underset{\boldsymbol{q}}{argmax}\, P(\boldsymbol{q}|\boldsymbol{e})$$

  - Since $P(\boldsymbol{q}|\boldsymbol{e}) = \frac{P(\boldsymbol{q},\boldsymbol{e})}{P(\boldsymbol{e})}$
  - $P(\boldsymbol{e})$ is independent of the instantiation $\boldsymbol{q}$

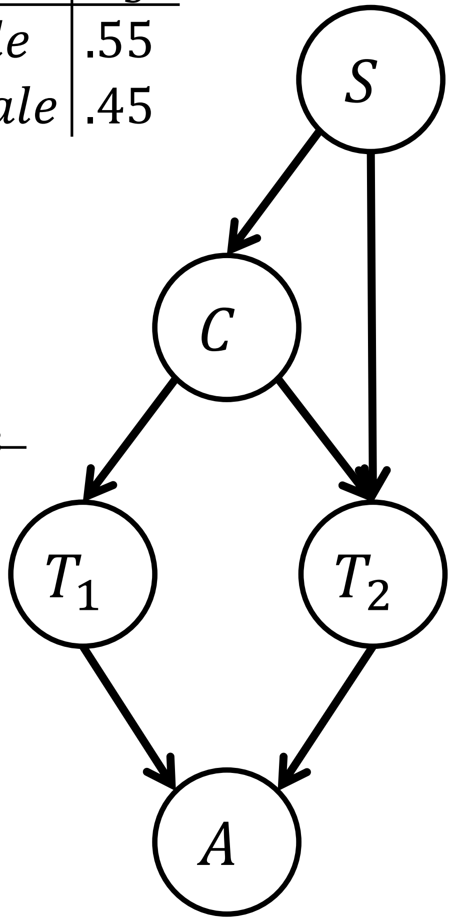
Computing MPE by Variable Elimination

- Returning to our example

S	C	$\Theta_{S\|C}$
$male$	yes	$.05$
$male$	no	$.95$
$female$	yes	$.01$
$female$	no	$.99$

C	T_1	$\Theta_{T_1\|C}$
yes	ve	$.80$
yes	$\overline{ve}$	$.20$
no	ve	$.20$
no	$\overline{ve}$	$.80$

S	Θ_S
$male$	$.55$
$female$	$.45$



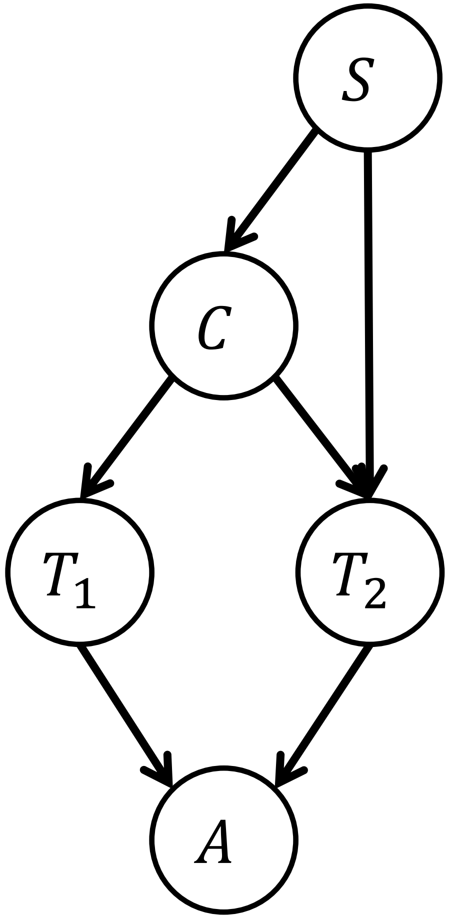
S	C	T_2	$\Theta_{T_2\|C,S}$
$male$	yes	ve	$.80$
$male$	yes	$\overline{ve}$	$.20$
$male$	no	ve	$.20$
$male$	no	$\overline{ve}$	$.80$
$female$	yes	ve	$.95$
$female$	yes	$\overline{ve}$	$.05$
$female$	no	ve	$.05$
$female$	no	$\overline{ve}$	$.95$

T_1	T_2	A	$\Theta_{A\|T_1,T_2}$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0

Computing MPE by Variable Elimination

- Returning to our example
  - We can compute the joint probability for this Bayesian network
  - (Even rows omitted since they have zero probabilities)
  - The MPE instantiation (assuming no evidence) is given in row 31
  - MPE probability (MPE_P) is .338580

	S	C	T_1	T_2	A	$P(.)$
1	$male$	yes	ve	ve	yes	.017600
3	$male$	yes	ve	$\overline{ve}$	no	.004400
5	$male$	yes	$\overline{ve}$	ve	no	.004400
7	$male$	yes	$\overline{ve}$	$\overline{ve}$	yes	.001100
9	$male$	no	ve	ve	yes	.020900
11	$male$	no	ve	$\overline{ve}$	no	.083600
13	$male$	no	$\overline{ve}$	ve	no	.083600
15	$male$	no	$\overline{ve}$	$\overline{ve}$	yes	.334400
17	$female$	yes	ve	ve	yes	.003420
19	$female$	yes	ve	$\overline{ve}$	no	.000180
21	$female$	yes	$\overline{ve}$	ve	no	.000855
23	$female$	yes	$\overline{ve}$	$\overline{ve}$	yes	.000045
25	$female$	no	ve	ve	yes	.004455
27	$female$	no	ve	$\overline{ve}$	no	.084645
29	$female$	no	$\overline{ve}$	ve	no	.017820
31	$female$	no	$\overline{ve}$	$\overline{ve}$	yes	.338580



Computing MPE by Variable Elimination

- We can compute MPE_p using Variable Elimination
  - However, when eliminating a variable, we maximize out instead of summing it out
- To maximize out a variable B from a factor $\phi(A, B, C)$, we produce another factor over remaining variables A and C
  - By merging all rows that agree on the values of these remaining variables
  - As we merge rows, we drop reference to the maximized variable and assign to the resulting row the maximum probability associated with the merged rows

A	B	C	$\phi(A, B, C)$
0	0	0	7
0	0	1	4.5
0	1	0	.2
0	1	1	2
1	0	0	3
1	0	1	.5
1	1	0	1.2
1	1	1	3

A	C	$\max_B \phi(A, C)$
0	0	7
0	1	4.5
1	0	3
1	1	3

Computing MPE by Variable Elimination

- The result of maximizing out variable B from factor ϕ is
  - Another factor, $\max_B \phi$ that does not mention B
  - The new factor agrees with the old factor on the MPE probability
- We can continue to maximize out $\max_B \phi$ until we get the trivial factor
  - The probability assigned to this factor is the MPE probability
  - This method can be extended to provide the MPE instantiation (more later)
- Maximization is commutative
  - Allow us to refer to maximizing out a set of variables without specifying the order
  - Also, $\max_X \phi_1 \phi_2 = \phi_1 \max_X \phi_2$ if variable X appears only in ϕ_2

MPE VE: Algorithm

$\boldsymbol{Q} \leftarrow$ variables in the network

$\pi \leftarrow$ elimination order of variables $\boldsymbol{Q}$

$\boldsymbol{S} \leftarrow \{\phi^{\boldsymbol{e}}: \phi$ is a factor of the network$\}$

for $i = 1$ to $|\boldsymbol{Q}|$ **do**

  $\sigma_i \leftarrow \prod_k \phi_k$, where ϕ_k belongs to $\boldsymbol{S}$ and mentions variable $\pi(i)$

  $\tau_i \leftarrow \max_{\pi(i)} \sigma_i$

  replace all factors ϕ_k in $\boldsymbol{S}$ by factor τ_i

return trivial factor $\prod_{\tau \in S} \tau$

Notes:

- All factors are eliminated leading to a trivial factor
- $\phi^{\boldsymbol{e}}$ is a factor with the rows of factor ϕ that match the evidence $\boldsymbol{e}$
- Pruning should eliminate edges only since all variables are relevant to the answer
- This algorithm has the same complexity as VE, i.e., the time and space complexity are $O(n \exp(w))$ for n variables and an elimination width w

Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - With the elimination order $\pi = S, C, A, T_1, T_2$

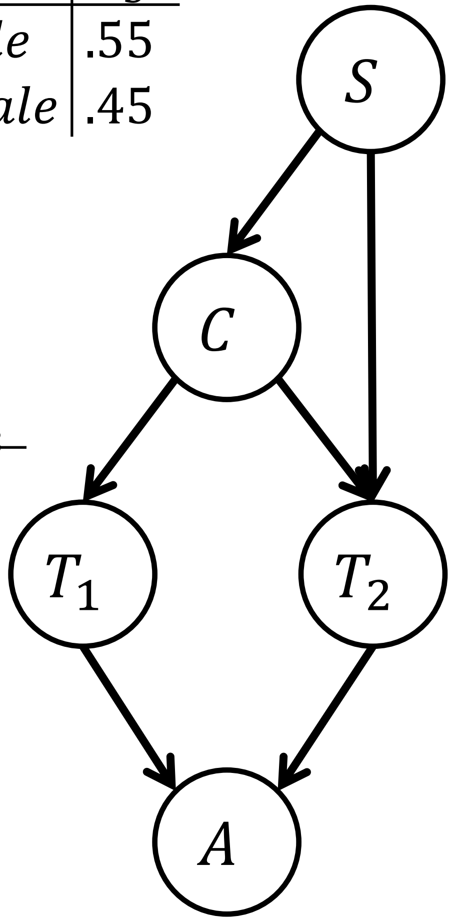
S	C	$\Theta_{S\|C}$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\Theta_{T_1\|C}$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	Θ_S
$male$	.55
$female$	.45

S	C	T_2	$\Theta_{T_2\|C,S}$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\Theta_{A\|T_1,T_2}$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - With the elimination order $\pi = S, C, A, T_1, T_2$

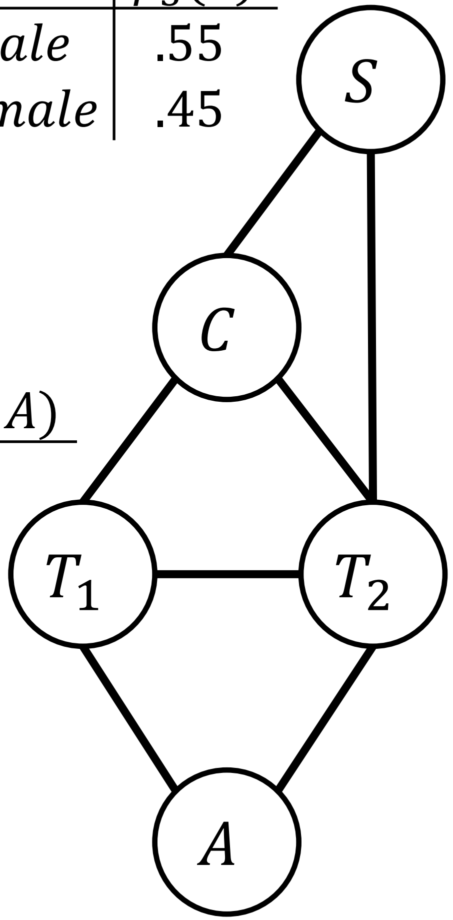
S	C	$\phi_1(S,C)$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\phi_2(T_1,C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

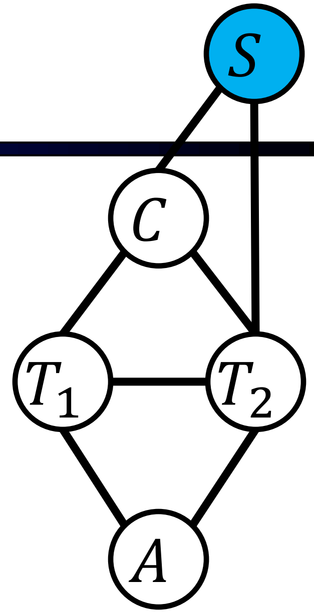
S	$\phi_3(S)$
$male$	.55
$female$	.45

S	C	T_2	$\phi_4(T_2,C,S)$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\phi_5(T_1,T_2,A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



Computing MPE: Example



- Returning to our example
  - Let us run MPE VE on this example
  - With the elimination order $\pi = S, C, A, T_1, T_2$

S	C	$\phi_1(S,C)$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\phi_2(T_1,C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	$\phi_3(S)$
$male$	.55
$female$	.45

$\times$

S	C	T_2	$\phi_4(T_2,C,S)$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

$\approx$

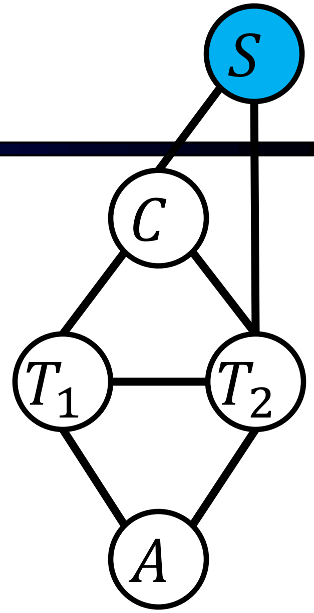
S	C	T_2	$\sigma_1(T_2,\mathrm{S},\mathrm{C})$
$male$	yes	ve	.440
$male$	yes	$\overline{ve}$	.110
$male$	no	ve	.110
$male$	no	$\overline{ve}$	.440
$female$	yes	ve	.428
$female$	yes	$\overline{ve}$	.023
$female$	no	ve	.023
$female$	no	$\overline{ve}$	.428

T_1	T_2	A	$\phi_5(T_1,T_2,A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0

Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80



S	C	$\phi_1(S, C)$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

$\times$

S	C	T_2	$\sigma_1(T_2, S, C)$
$male$	yes	ve	.440
$male$	yes	$\overline{ve}$	.110
$male$	no	ve	.110
$male$	no	$\overline{ve}$	.440
$female$	yes	ve	.428
$female$	yes	$\overline{ve}$	.023
$female$	no	ve	.023
$female$	no	$\overline{ve}$	.428

$\approx$

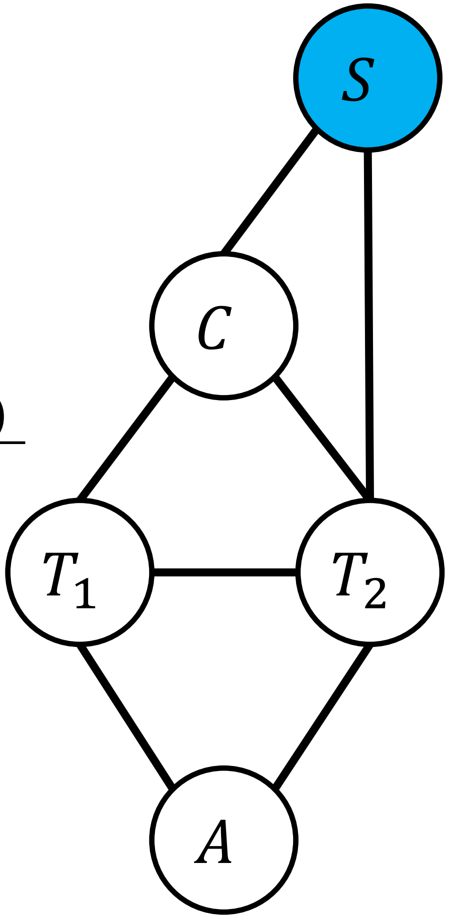
S	C	T_2	$\sigma_2(T_2, S, C)$
$male$	yes	ve	.0220
$male$	yes	$\overline{ve}$	.0055
$male$	no	ve	.1045
$male$	no	$\overline{ve}$	.4180
$female$	yes	ve	.0043
$female$	yes	$\overline{ve}$	.0002
$female$	no	ve	.0228
$female$	no	$\overline{ve}$	.4237

T_1	T_2	A	$\phi_5(T_1, T_2, A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0

Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80



S	C	T_2	$\sigma_2(T_2, S, C)$
$male$	yes	ve	.0220
$male$	yes	$\overline{ve}$	.0055
$male$	no	ve	.1045
$male$	no	$\overline{ve}$	.4180
$female$	yes	ve	.0043
$female$	yes	$\overline{ve}$	.0002
$female$	no	ve	.0228
$female$	no	$\overline{ve}$	.4237

C	T_2	$\tau_2(T_2, C)$
yes	ve	.0220
yes	$\overline{ve}$	.0055
no	ve	.1045
no	$\overline{ve}$	.4237

T_1	T_2	A	$\phi_5(T_1, T_2, A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0

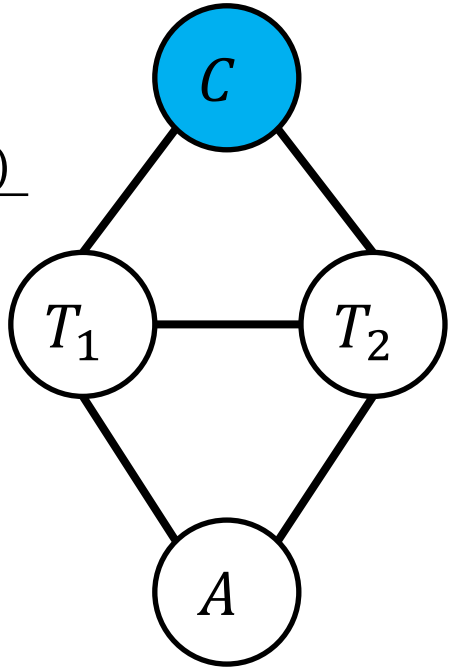
Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

C	T_2	$\tau_2(T_2, C)$
yes	ve	$.0220$
yes	$\overline{ve}$	$.0055$
no	ve	$.1045$
no	$\overline{ve}$	$.4237$

C	T_1	$\phi_2(T_1, C)$
yes	ve	$.80$
yes	$\overline{ve}$	$.20$
no	ve	$.20$
no	$\overline{ve}$	$.80$

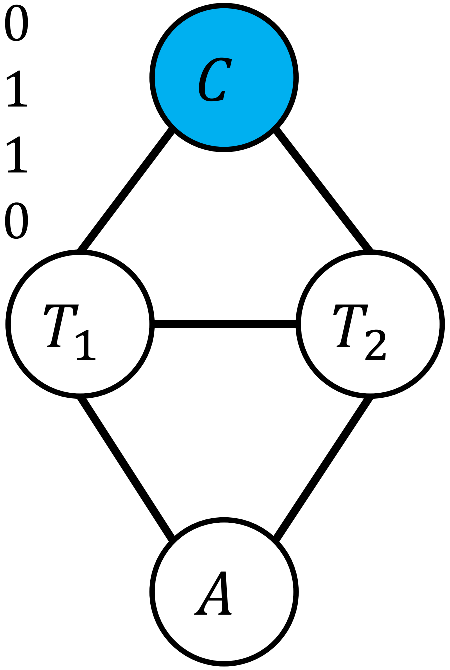
T_1	T_2	A	$\phi_5(T_1, T_2, A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

T_1	T_2	A	$\phi_5(T_1, T_2, A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



C	T_2	$\tau_2(T_2, C)$
yes	ve	.0220
yes	$\overline{ve}$	.0055
no	ve	.1045
no	$\overline{ve}$	.4237

$\times$

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

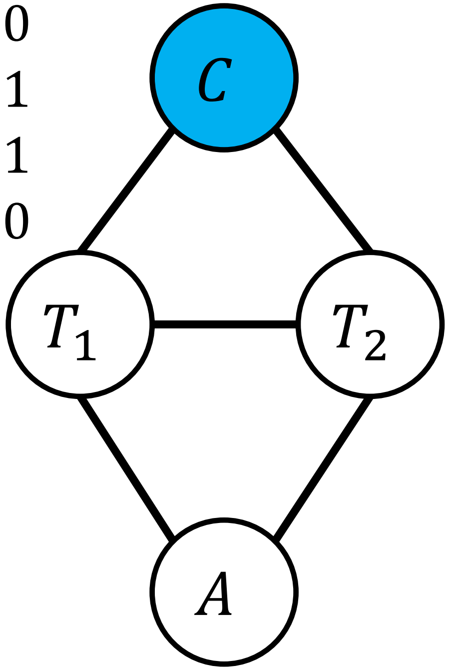
$\approx$

C	T_2	T_1	$\sigma_3(C, T_1, T_2)$
yes	ve	ve	.0176
yes	ve	$\overline{ve}$	.0044
yes	$\overline{ve}$	ve	.0044
yes	$\overline{ve}$	$\overline{ve}$	.0011
no	ve	ve	.0209
no	ve	$\overline{ve}$	.0836
no	$\overline{ve}$	ve	.0847
no	$\overline{ve}$	$\overline{ve}$	.3390

Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

T_1	T_2	A	$\phi_5(T_1, T_2, A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



C	T_2	T_1	$\sigma_3(C, T_1, T_2)$
yes	ve	ve	.0176
yes	ve	$\overline{ve}$	.0044
yes	$\overline{ve}$	ve	.0044
yes	$\overline{ve}$	$\overline{ve}$	.0011
no	ve	ve	.0209
no	ve	$\overline{ve}$	.0836
no	$\overline{ve}$	ve	.0847
no	$\overline{ve}$	$\overline{ve}$	.3390

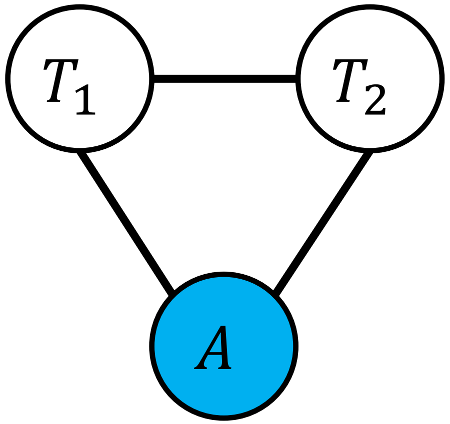
T_2	T_1	$\tau_3(T_2, T_1)$
ve	ve	.0209
ve	$\overline{ve}$	.0836
$\overline{ve}$	ve	.0847
$\overline{ve}$	$\overline{ve}$	.3390

Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

T_2	T_1	$\tau_3(T_2, T_1)$
ve	ve	.0209
ve	$\overline{ve}$	.0836
$\overline{ve}$	ve	.0847
$\overline{ve}$	$\overline{ve}$	.3390

T_1	T_2	A	$\phi_5(T_1, T_2, A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



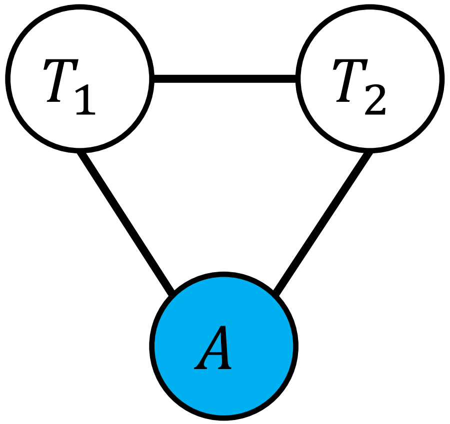
Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

T_2	T_1	$\tau_3(T_2, T_1)$
ve	ve	.0209
ve	$\overline{ve}$	.0836
$\overline{ve}$	ve	.0847
$\overline{ve}$	$\overline{ve}$	.3390

T_1	T_2	A	$\phi_5(T_1, T_2, A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0

T_1	T_2	$\tau_4(T_1, T_2)$
ve	ve	1
ve	$\overline{ve}$	1
$\overline{ve}$	ve	1
$\overline{ve}$	$\overline{ve}$	1

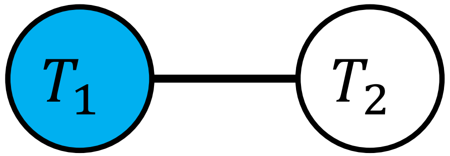


Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

T_1	T_2	$\tau_4(T_1, T_2)$
ve	ve	1
ve	$\overline{ve}$	1
$\overline{ve}$	ve	1
$\overline{ve}$	$\overline{ve}$	1

T_2	T_1	$\tau_3(T_2, T_1)$
ve	ve	.0209
ve	$\overline{ve}$	.0836
$\overline{ve}$	ve	.0847
$\overline{ve}$	$\overline{ve}$	.3390



Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

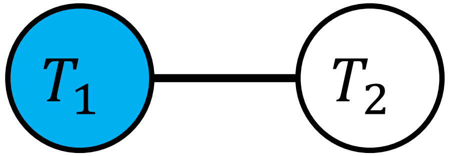
T_1	T_2	$\tau_4(T_1, T_2)$
ve	ve	1
ve	$\overline{ve}$	1
$\overline{ve}$	ve	1
$\overline{ve}$	$\overline{ve}$	1

$\times$

T_2	T_1	$\tau_3(T_2, T_1)$
ve	ve	.0209
ve	$\overline{ve}$	.0836
$\overline{ve}$	ve	.0847
$\overline{ve}$	$\overline{ve}$	.3390

$=$

T_1	T_2	$\sigma_5(T_1, T_2)$
ve	ve	.0209
ve	$\overline{ve}$	.0847
$\overline{ve}$	ve	.0836
$\overline{ve}$	$\overline{ve}$	.3390

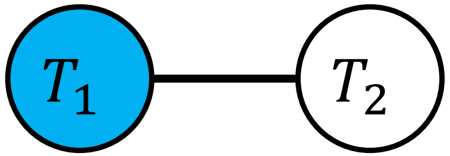


Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

T_1	T_2	$\sigma_5(T_1, T_2)$
ve	ve	.0209
ve	$\overline{ve}$	.0847
$\overline{ve}$	ve	.0836
$\overline{ve}$	$\overline{ve}$	.3390

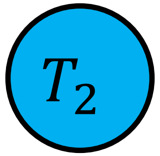
T_2	$\tau_5(T_2)$
ve	.0836
$\overline{ve}$	.3390



Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$

T_2	$\tau_5(T_2)$
ve	$.0836$
$\overline{ve}$	$.3390$



Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example
  - And use the elimination order $\pi = S, C, A, T_1, T_2$
  - $MPE_P \approx 0.3390$
- We can also be interested in the MPE instantiation
  - However, we lost this piece of information during the elimination

Recovering MPE Instantiation

- We can modify the previous algorithm to compute the MPE instantiation
  - In addition to the MPE probability
- The idea is to use *extended factors*
  - It assigns to each instantiation a number and an instantiation
- We use $\phi[x]$ to denote the instantiation
  - While continuing to use $\phi(x)$ for denoting the number
  - The instantiation $\phi[x]$ is used to record the MPE instantiation as it is being constructed

S	C	T_2	$\phi(.)$
$male$	yes	ve	$.0220$
$male$	yes	$\overline{ve}$	$.0055$
$male$	no	ve	$.1045$
$male$	no	$\overline{ve}$	$.4180$
$female$	yes	ve	$.0043$
$female$	yes	$\overline{ve}$	$.0002$
$female$	no	ve	$.0228$
$female$	no	$\overline{ve}$	$.4237$

C	T_2	$\phi(.)$	$\phi[.]$
yes	ve	$.0220$	$male$
yes	$\overline{ve}$	$.0055$	$male$
no	ve	$.1045$	$male$
no	$\overline{ve}$	$.4237$	$female$

Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

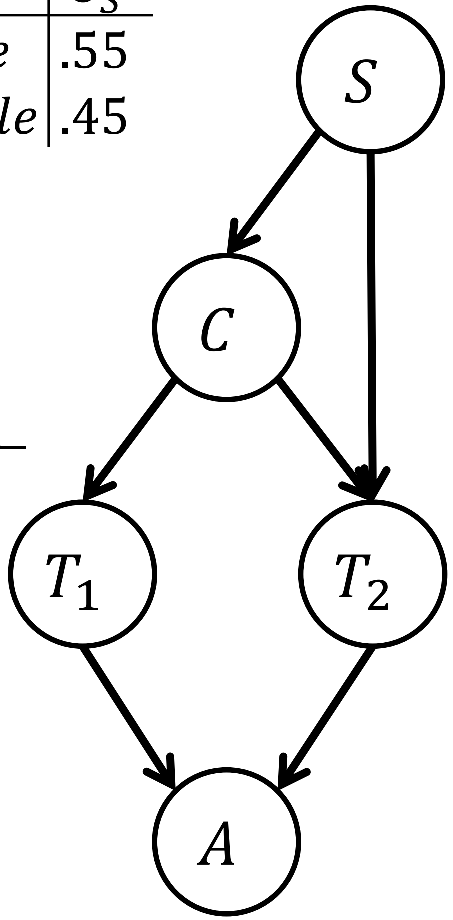
S	C	$\Theta_{C\|S}$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\Theta_{T_1\|C}$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	Θ_S
$male$	.55
$female$	.45

S	C	T_2	$\Theta_{T_2\|C,S}$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\Theta_{A\|T_1,T_2}$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



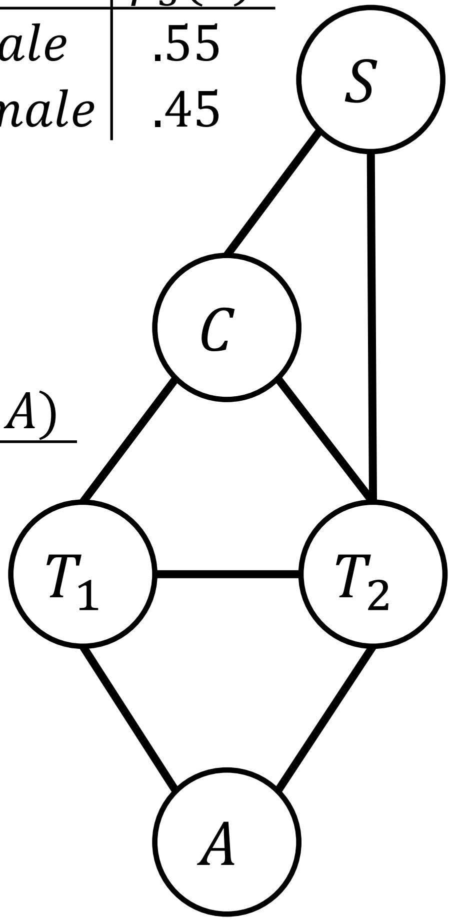
Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

S	C	$\phi_1(S, C)$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	$\phi_3(S)$
$male$	.55
$female$	.45



S	C	T_2	$\phi_4(T_2, C, S)$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\phi_5(T_1, T_2, A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0

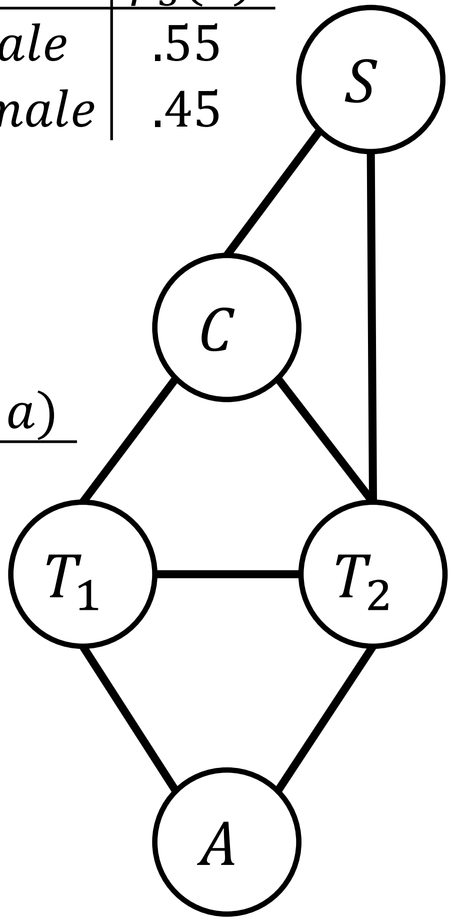
Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

S	C	$\phi_1(S,C)$
$male$	yes	$.05$
$male$	no	$.95$
$female$	yes	$.01$
$female$	no	$.99$

C	T_1	$\phi_2(T_1,C)$
yes	ve	$.80$
yes	$\overline{ve}$	$.20$
no	ve	$.20$
no	$\overline{ve}$	$.80$

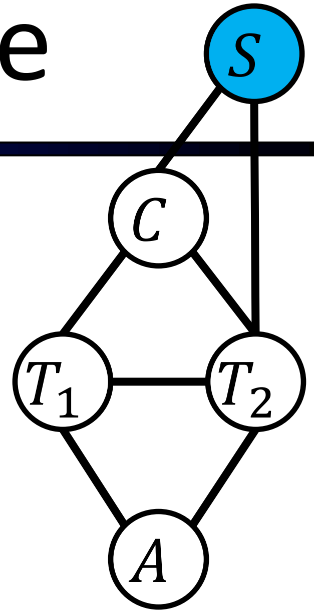
S	$\phi_3(S)$
$male$	$.55$
$female$	$.45$



S	C	T_2	$\phi_4(T_2,C,S)$
$male$	yes	ve	$.80$
$male$	yes	$\overline{ve}$	$.20$
$male$	no	ve	$.20$
$male$	no	$\overline{ve}$	$.80$
$female$	yes	ve	$.95$
$female$	yes	$\overline{ve}$	$.05$
$female$	no	ve	$.05$
$female$	no	$\overline{ve}$	$.95$

T_1	T_2	a	$\phi_5(T_1,T_2,a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1

Computing MPE Instantiation: Example



- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

S	C	$\phi_1(S,C)$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\phi_2(T_1,C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	$\phi_3(S)$
$male$	.55
$female$	.45

$\times$

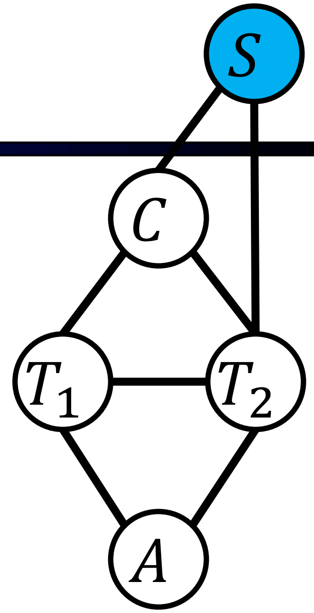
S	C	T_2	$\phi_4(T_2,C,S)$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

$\approx$

S	C	T_2	$\sigma_1(T_2,\mathrm{S},\mathrm{C})$
$male$	yes	ve	.440
$male$	yes	$\overline{ve}$	.110
$male$	no	ve	.110
$male$	no	$\overline{ve}$	.440
$female$	yes	ve	.428
$female$	yes	$\overline{ve}$	.023
$female$	no	ve	.023
$female$	no	$\overline{ve}$	.428

T_1	T_2	a	$\phi_5(T_1,T_2,a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1

Computing MPE: Example



- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	C	$\phi_1(S, C)$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

$\times$

S	C	T_2	$\sigma_1(T_2, S, C)$
$male$	yes	ve	.440
$male$	yes	$\overline{ve}$	.110
$male$	no	ve	.110
$male$	no	$\overline{ve}$	.440
$female$	yes	ve	.428
$female$	yes	$\overline{ve}$	.023
$female$	no	ve	.023
$female$	no	$\overline{ve}$	.428

$\approx$

S	C	T_2	$\sigma_2(T_2, S, C)$
$male$	yes	ve	.0220
$male$	yes	$\overline{ve}$	.0055
$male$	no	ve	.1045
$male$	no	$\overline{ve}$	.4180
$female$	yes	ve	.0043
$female$	yes	$\overline{ve}$	.0002
$female$	no	ve	.0228
$female$	no	$\overline{ve}$	.4237

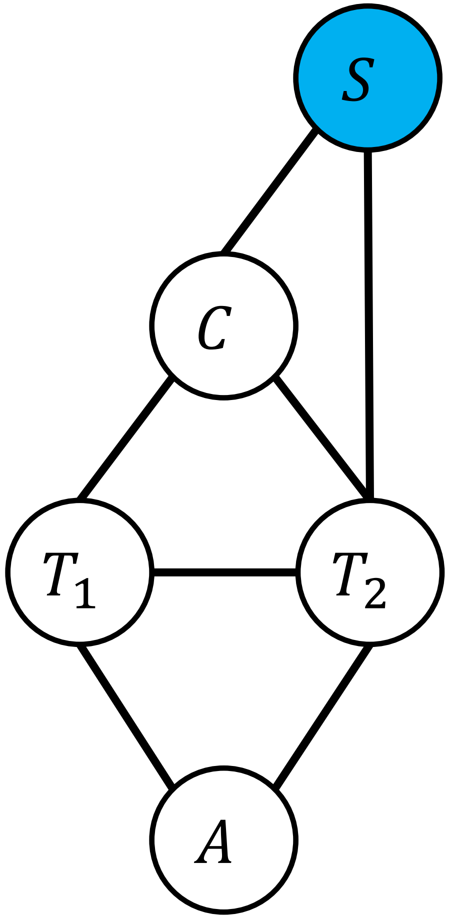
T_1	T_2	a	$\phi_5(T_1, T_2, a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1

Computing MPE: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

T_1	T_2	a	$\phi_5(T_1, T_2, a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80



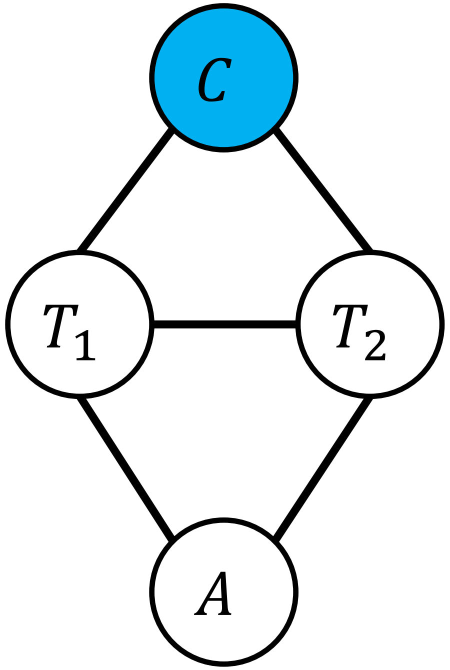
S	C	T_2	$\sigma_2(T_2, S, C)$
$male$	yes	ve	.0220
$male$	yes	$\overline{ve}$	.0055
$male$	no	ve	.1045
$male$	no	$\overline{ve}$	.4180
$female$	yes	ve	.0043
$female$	yes	$\overline{ve}$	.0002
$female$	no	ve	.0228
$female$	no	$\overline{ve}$	.4237

C	T_2	$\tau_2(T_2, C)$	
yes	ve	.0220	$male$
yes	$\overline{ve}$	.0055	$male$
no	ve	.1045	$male$
no	$\overline{ve}$	.4237	$female$

Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

T_1	T_2	a	$\phi_5(T_1, T_2, a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1



C	T_2	$\tau_2(T_2, C)$	
yes	ve	.0220	$male$
yes	$\overline{ve}$	.0055	$male$
no	ve	.1045	$male$
no	$\overline{ve}$	.4237	$female$

$\times$

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

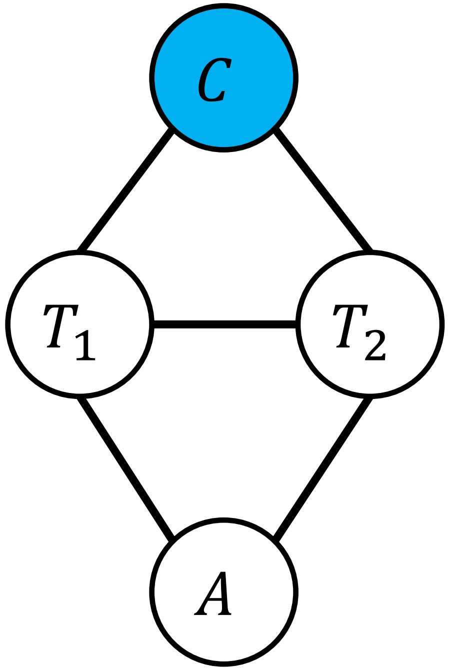
$\approx$

C	T_2	T_1	$\sigma_3(C, T_1, T_2)$	
yes	ve	ve	.0176	$male$
yes	ve	$\overline{ve}$	.0044	$male$
yes	$\overline{ve}$	ve	.0044	$male$
yes	$\overline{ve}$	$\overline{ve}$	.0011	$male$
no	ve	ve	.0209	$male$
no	ve	$\overline{ve}$	.0836	$male$
no	$\overline{ve}$	ve	.0847	$female$
no	$\overline{ve}$	$\overline{ve}$	.3390	$female$

Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

T_1	T_2	a	$\phi_5(T_1, T_2, a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1



C	T_2	T_1	$\sigma_3(C, T_1, T_2)$	
yes	ve	ve	.0176	$male$
yes	ve	$\overline{ve}$	.0044	$male$
yes	$\overline{ve}$	ve	.0044	$male$
yes	$\overline{ve}$	$\overline{ve}$	.0011	$male$
no	ve	ve	.0209	$male$
no	ve	$\overline{ve}$	.0836	$male$
no	$\overline{ve}$	ve	.0847	$female$
no	$\overline{ve}$	$\overline{ve}$	.3390	$female$

T_2	T_1	$\tau_3(T_2, T_1)$	
ve	ve	.0209	$male, no$
ve	$\overline{ve}$	.0836	$male, no$
$\overline{ve}$	ve	.0847	$female, no$
$\overline{ve}$	$\overline{ve}$	.3390	$female, no$

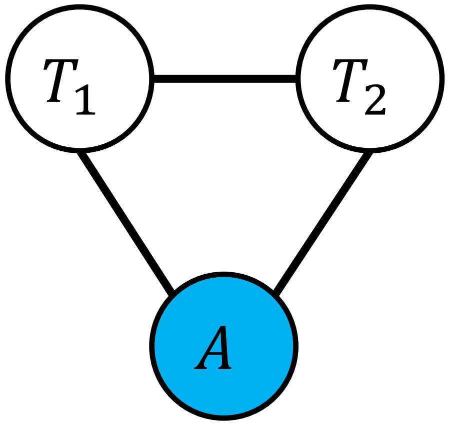
Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

T_2	T_1	$\tau_3(T_2, T_1)$	
ve	ve	$.0209$	$male, no$
ve	$\overline{ve}$	$.0836$	$male, no$
$\overline{ve}$	ve	$.0847$	$female, no$
$\overline{ve}$	$\overline{ve}$	$.3390$	$female, no$

T_1	T_2	a	$\phi_5(T_1, T_2, a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1

T_1	T_2	$\tau_4(T_1, T_2)$
ve	ve	1
ve	$\overline{ve}$	0
$\overline{ve}$	ve	0
$\overline{ve}$	$\overline{ve}$	1



Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

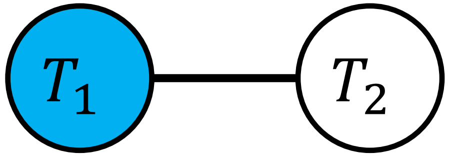
T_2	T_1	$\tau_3(T_2, T_1)$	
ve	ve	$.0209$	$male, no$
ve	$\overline{ve}$	$.0836$	$male, no$
$\overline{ve}$	ve	$.0847$	$female, no$
$\overline{ve}$	$\overline{ve}$	$.3390$	$female, no$

$\times$

T_1	T_2	$\tau_4(T_1, T_2)$
ve	ve	1
ve	$\overline{ve}$	0
$\overline{ve}$	ve	0
$\overline{ve}$	$\overline{ve}$	1

$\approx$

T_1	T_2	$\sigma_5(T_1, T_2)$	
ve	ve	$.0209$	$male, no$
ve	$\overline{ve}$	0	$female, no$
$\overline{ve}$	ve	0	$male, no$
$\overline{ve}$	$\overline{ve}$	$.3390$	$female, no$

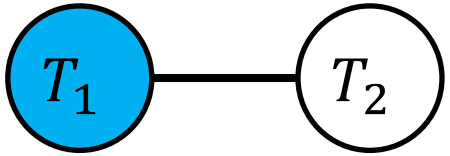


Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

T_1	T_2	$\sigma_5(T_1, T_2)$	
ve	ve	.0209	$male, no$
ve	$\overline{ve}$	0	$female, no$
$\overline{ve}$	ve	0	$male, no$
$\overline{ve}$	$\overline{ve}$	.3390	$female, no$

T_2	$\tau_5(T_2)$	
ve	.0290	$male, no, ve$
$\overline{ve}$	.3390	$female, no, \overline{ve}$



Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$

T_2	$\tau_5(T_2)$	
ve	.0290	$male, no, ve$
$\overline{ve}$	.3390	$female, no, \overline{ve}$

τ_6	
.3390	$female, no, \overline{ve}, \overline{ve}$

T_2

Computing MPE Instantiation: Example

- Returning to our example
  - Let us run MPE VE on this example, but now computing the MPE instantiation with evidence $A = yes$
- Since $P(\boldsymbol{e}) = P(A = yes) = .7205$
  - $MPE_p(\boldsymbol{Q}|\boldsymbol{e}) = 47\%$

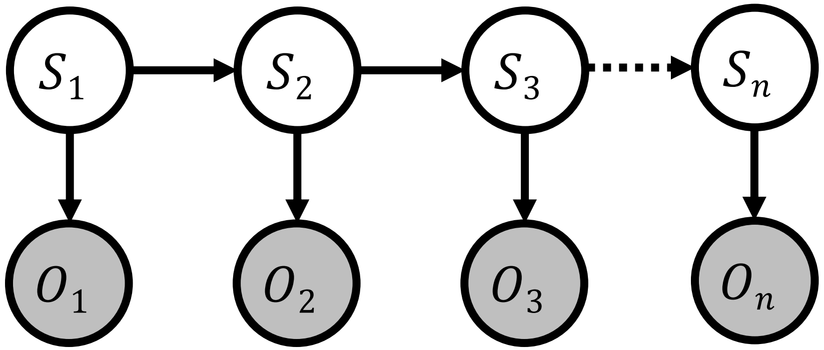
T_2	$\tau_5(T_2)$	
ve	$.0290$	$male, no, ve$
$\overline{ve}$	$.3390$	$female, no, \overline{ve}$

τ_6	
.3390	$female, no, \overline{ve}, \overline{ve}$

T_2

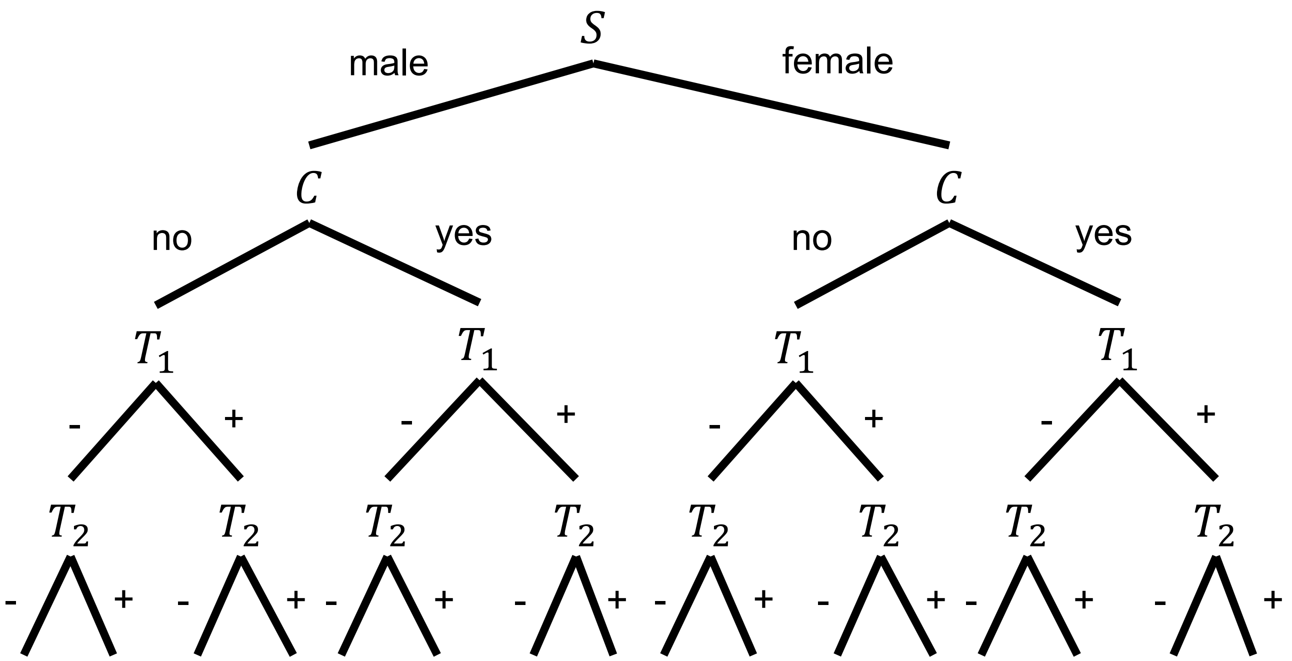
MPE and HMM

- We have seen MPE queries in the context of HMM
  - If we apply the MPE algorithm with elimination order $\pi = O_1, S_1, O_2, S_2, \dots, O_n, S_n$, we obtain the Viterbi algorithm
  - If we apply the VE algorithm with same order, we obtain the Forward algorithm
- This elimination order has width = 1 for HMMs
  - Therefore both algorithms have linear time and space complexity



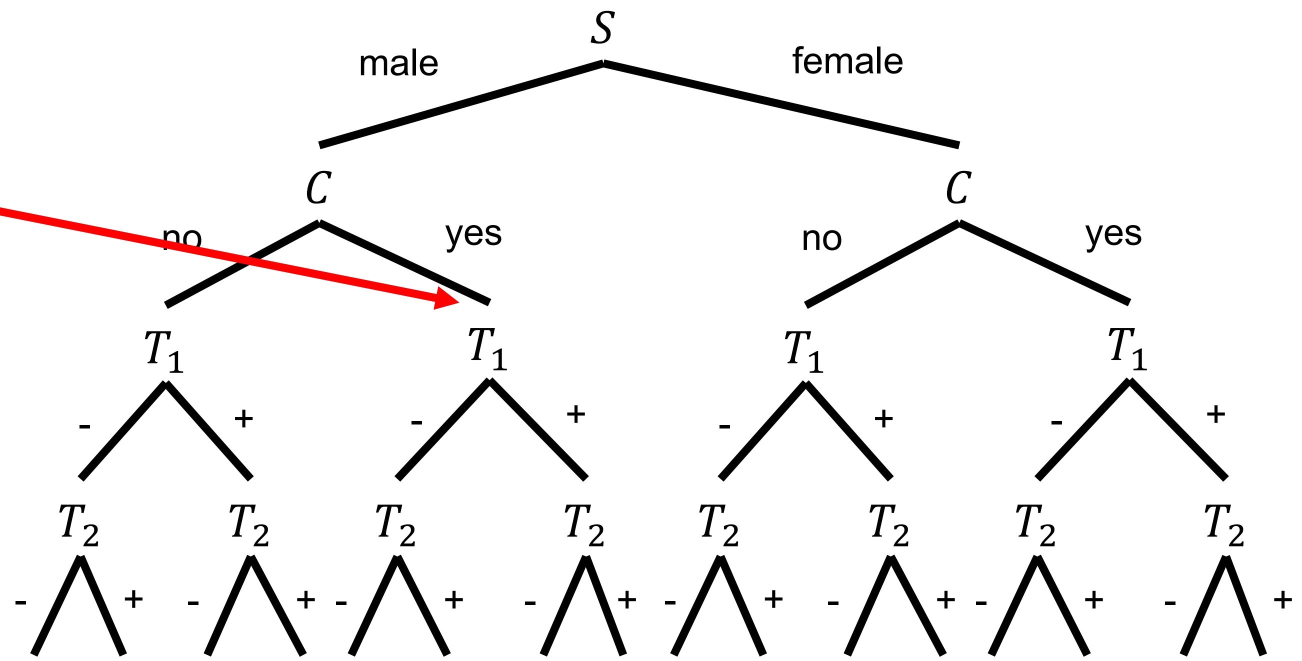
Computing MPE by Systematic Search

- We now consider a different class of algorithms for Bayesian networks
  - Based on systematic search
  - They can be more efficient than algorithms based on VE
- Suppose we want to find an MPE instantiation given $A = true$
  - We can use depth-first search on the tree on right side
  - The leaf nodes are instantiations of unobserved variables



Computing MPE by Systematic Search

- Each non-leaf is a partial instantiation
  - This node represents $S = male, C = yes$
  - The children of this node are extensions of this instantiation
- We can traverse this search tree using depth-first search
  - With n unobserved variables, the search takes $O(n)$ space and $O(n \exp(n))$ time



DFS MPE: Algorithm

main:

$\boldsymbol{Q} \leftarrow$ variables in the network distinct from variables $\boldsymbol{E}$

$\boldsymbol{s} \leftarrow$ network instantiation compatible with evidence $\boldsymbol{e}$ # global variable

$p \leftarrow$probability of instantiation $\boldsymbol{s}$ # global variable

DFS_MPE($\boldsymbol{e}$, $\boldsymbol{Q}$)

return $\boldsymbol{s}$

DFS_MPE($\boldsymbol{i}, \boldsymbol{X}$)

if $\boldsymbol{X}$ is empty **then** # $\boldsymbol{i}$ is a network instantiation

if $P(\boldsymbol{i}) > p$ **then**

$\boldsymbol{s} \leftarrow \boldsymbol{i}$

$p \leftarrow P(\boldsymbol{i})$

else

$X \leftarrow$ a variable in $\boldsymbol{X}$

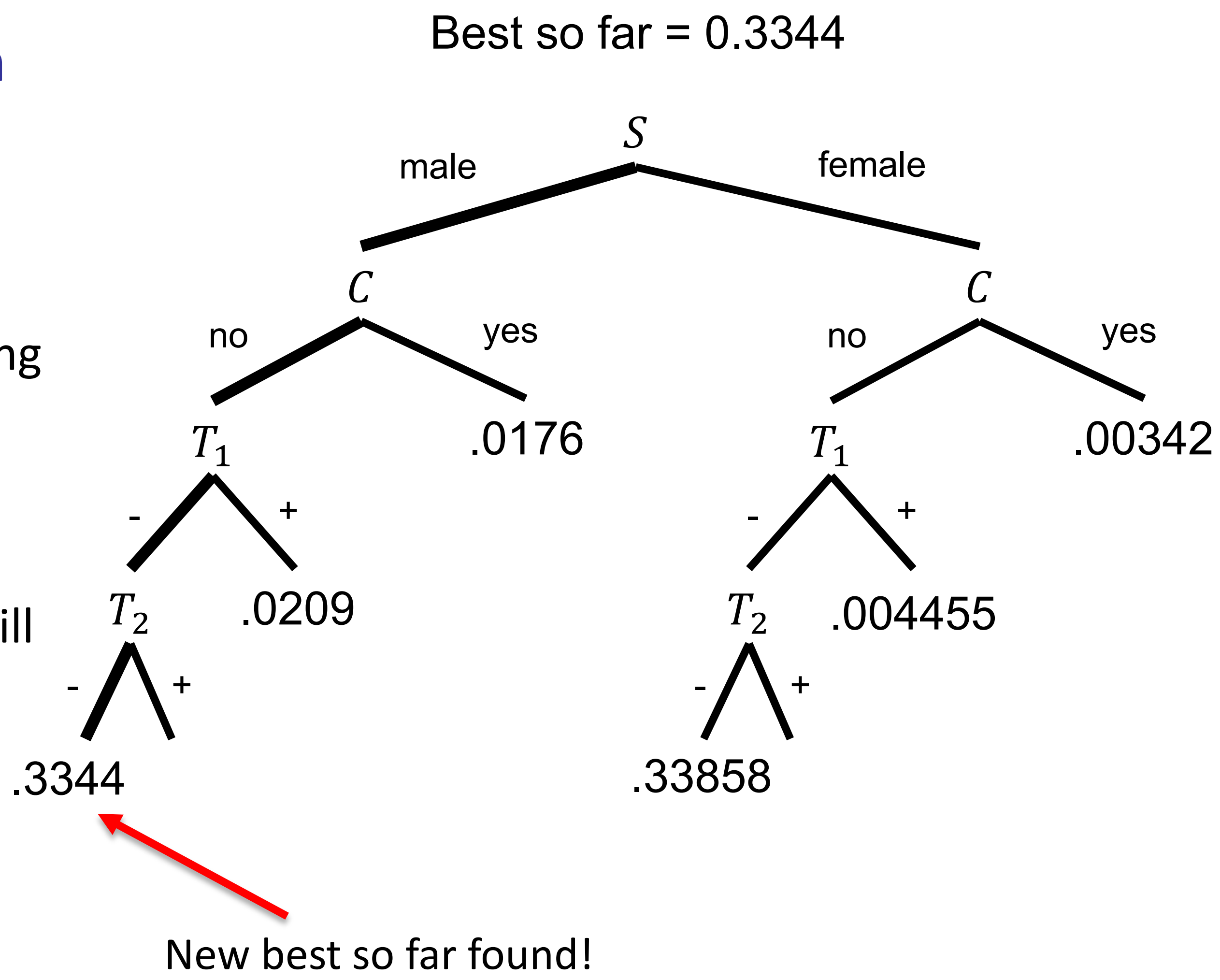
for each value x of variable X **do**

DFS_MPE($\boldsymbol{i}x, \boldsymbol{X} \backslash \{X\}$)

$P(\boldsymbol{i})$ can be computed in linear time using the chain rule for Bayesian networks

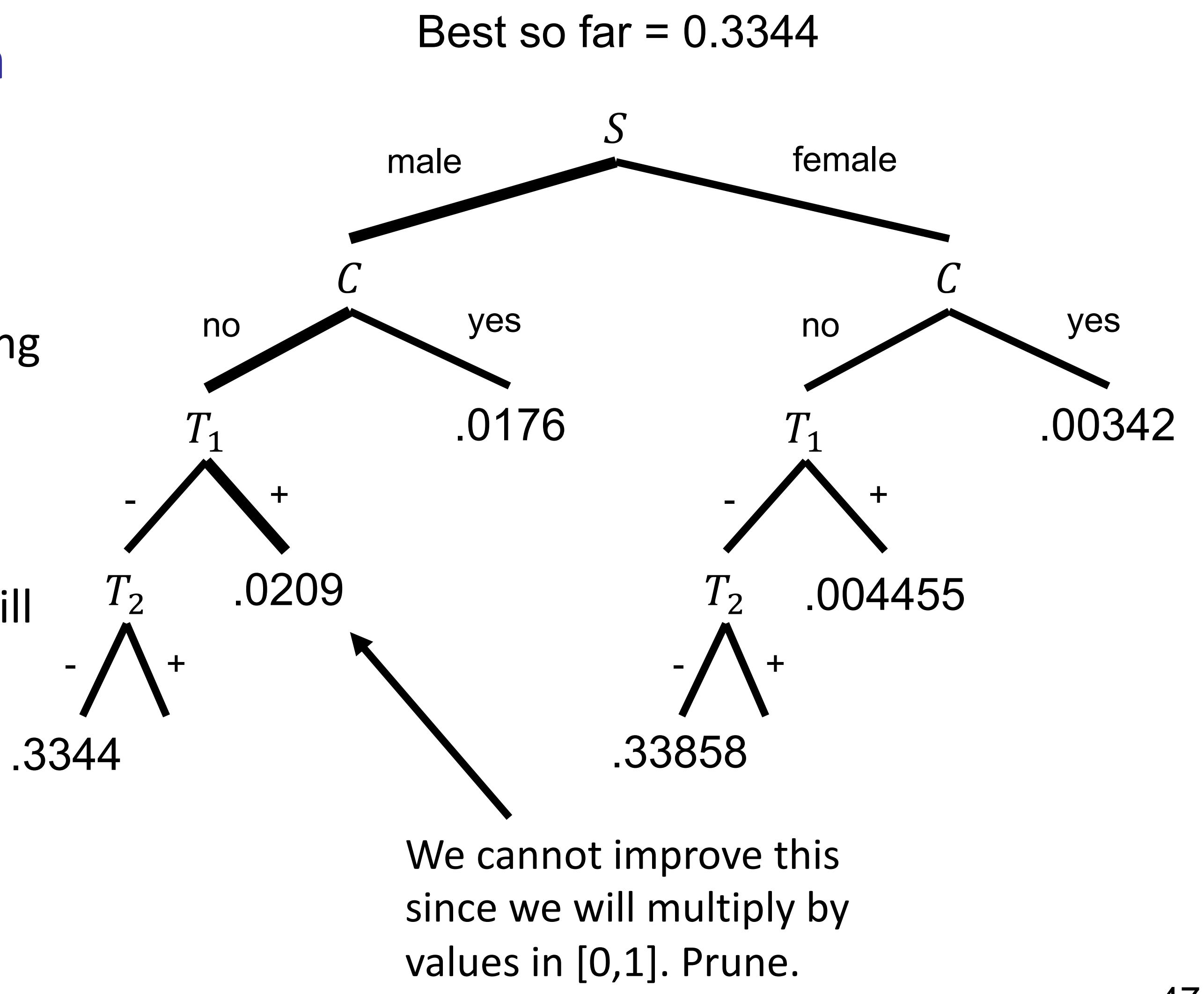
Branch-and-Bound

- We can prune the search tree using an upper bound on the MPE probability
  - Suppose we are exploring a search node corresponding to instantiation $\boldsymbol{i}$
  - If we have visited a leaf node corresponding to the instantiation $\boldsymbol{s}$ with probability p
  - And we know that every completion of $\boldsymbol{i}$ will not have higher probability
  - We can abandon the search node $\boldsymbol{i}$ as it will not lead to improvement over $\boldsymbol{s}$



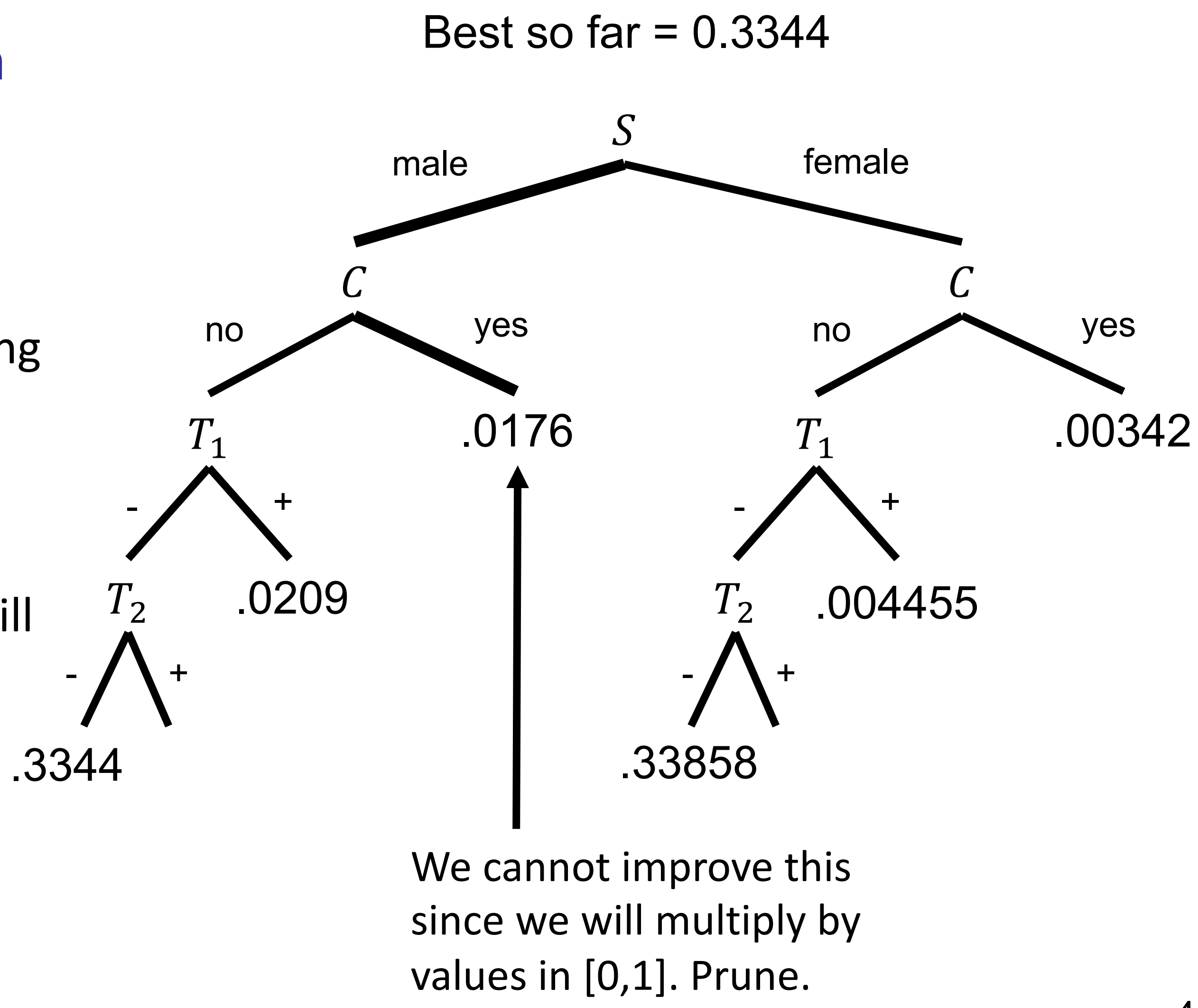
Branch-and-Bound

- We can prune the search tree using an upper bound on the MPE probability
  - Suppose we are exploring a search node corresponding to instantiation $\boldsymbol{i}$
  - If we have visited a leaf node corresponding to the instantiation $\boldsymbol{s}$ with probability p
  - And we know that every completion of $\boldsymbol{i}$ will not have higher probability
  - We can abandon the search node $\boldsymbol{i}$ as it will not lead to improvement over $\boldsymbol{s}$



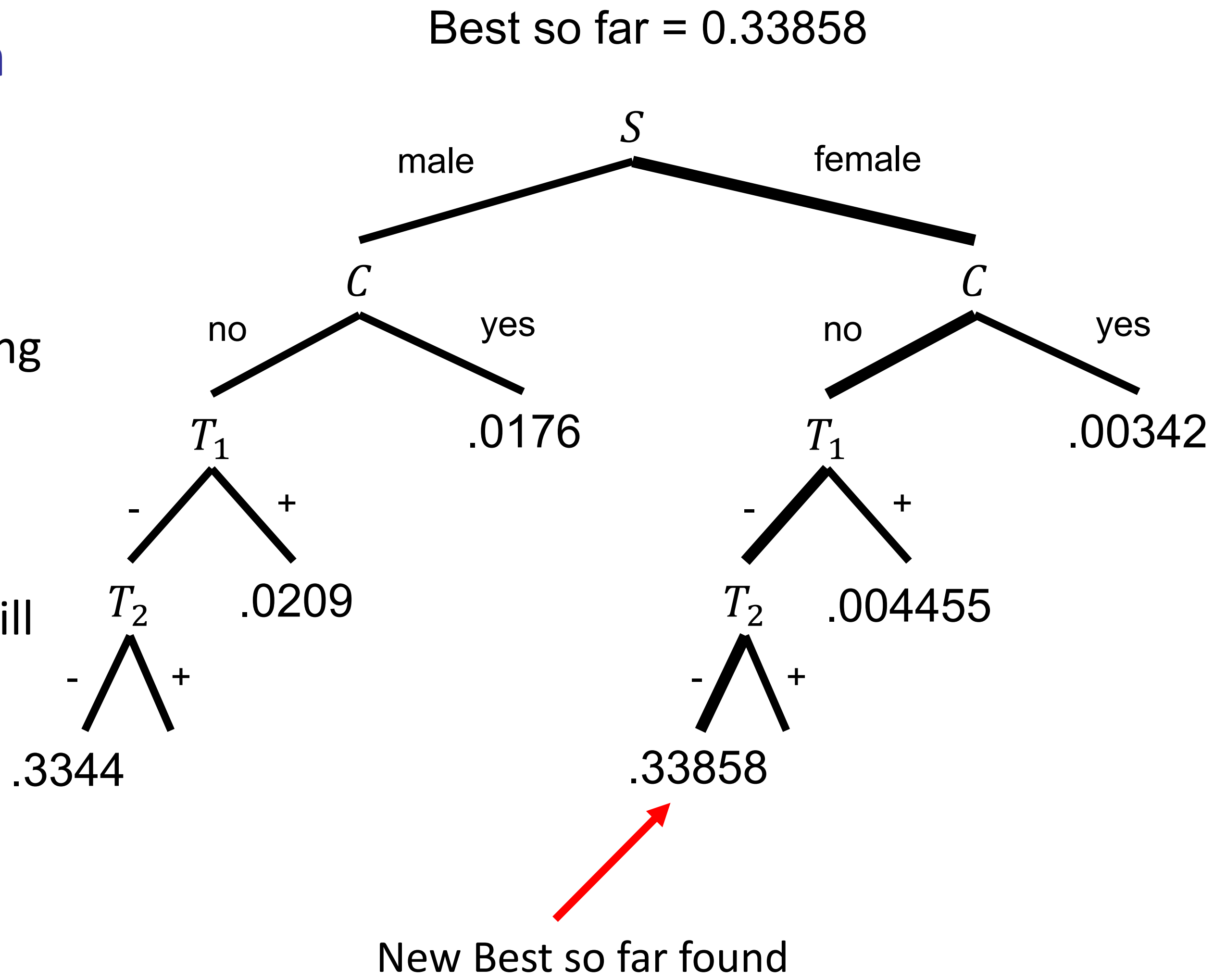
Branch-and-Bound

- We can prune the search tree using an upper bound on the MPE probability
  - Suppose we are exploring a search node corresponding to instantiation $\boldsymbol{i}$
  - If we have visited a leaf node corresponding to the instantiation $\boldsymbol{s}$ with probability p
  - And we know that every completion of $\boldsymbol{i}$ will not have higher probability
  - We can abandon the search node $\boldsymbol{i}$ as it will not lead to improvement over $\boldsymbol{s}$



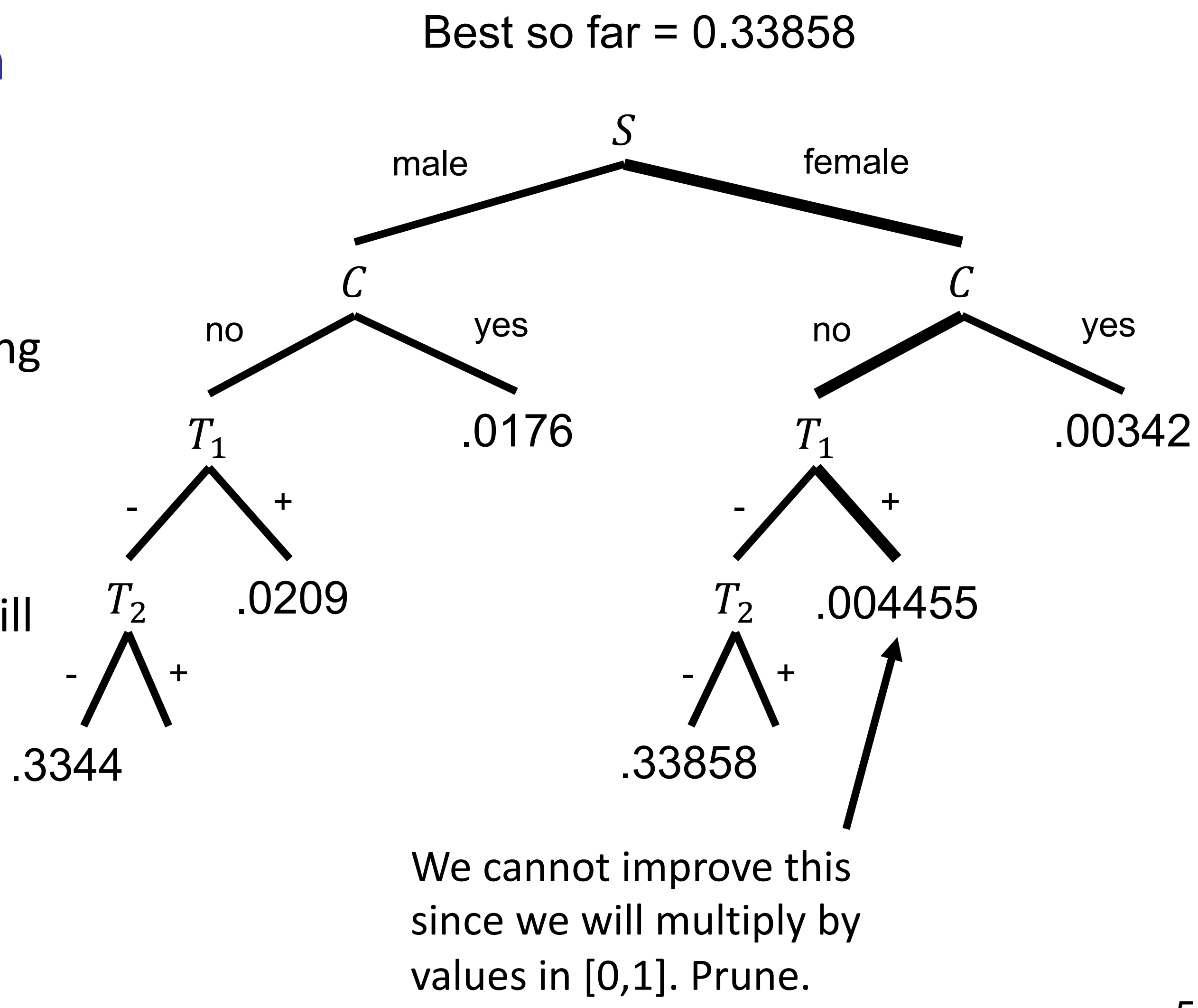
Branch-and-Bound

- We can prune the search tree using an upper bound on the MPE probability
  - Suppose we are exploring a search node corresponding to instantiation $\boldsymbol{i}$
  - If we have visited a leaf node corresponding to the instantiation $\boldsymbol{s}$ with probability p
  - And we know that every completion of $\boldsymbol{i}$ will not have higher probability
  - We can abandon the search node $\boldsymbol{i}$ as it will not lead to improvement over $\boldsymbol{s}$



Branch-and-Bound

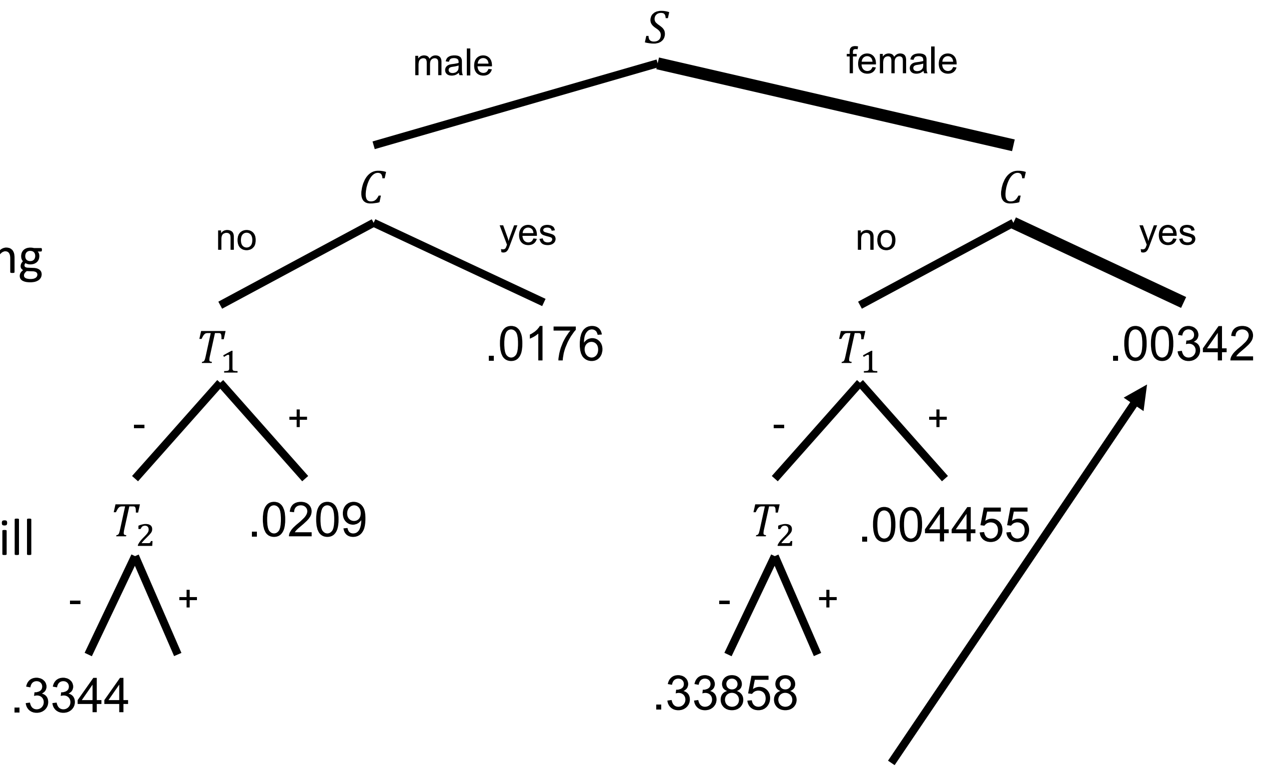
- We can prune the search tree using an upper bound on the MPE probability
  - Suppose we are exploring a search node corresponding to instantiation $\boldsymbol{i}$
  - If we have visited a leaf node corresponding to the instantiation $\boldsymbol{s}$ with probability p
  - And we know that every completion of $\boldsymbol{i}$ will not have higher probability
  - We can abandon the search node $\boldsymbol{i}$ as it will not lead to improvement over $\boldsymbol{s}$



Branch-and-Bound

- We can prune the search tree using an upper bound on the MPE probability
  - Suppose we are exploring a search node corresponding to instantiation $\boldsymbol{i}$
  - If we have visited a leaf node corresponding to the instantiation $\boldsymbol{s}$ with probability p
  - And we know that every completion of $\boldsymbol{i}$ will not have higher probability
  - We can abandon the search node $\boldsymbol{i}$ as it will not lead to improvement over $\boldsymbol{s}$
- The efficiency of this algorithm depends on the tightness of the upper bounds and the time to compute them

Best so far = 0.33858



We cannot improve this since we will multiply by values in [0,1]. Prune.

DFS Branch-and-Bound MPE: Algorithm

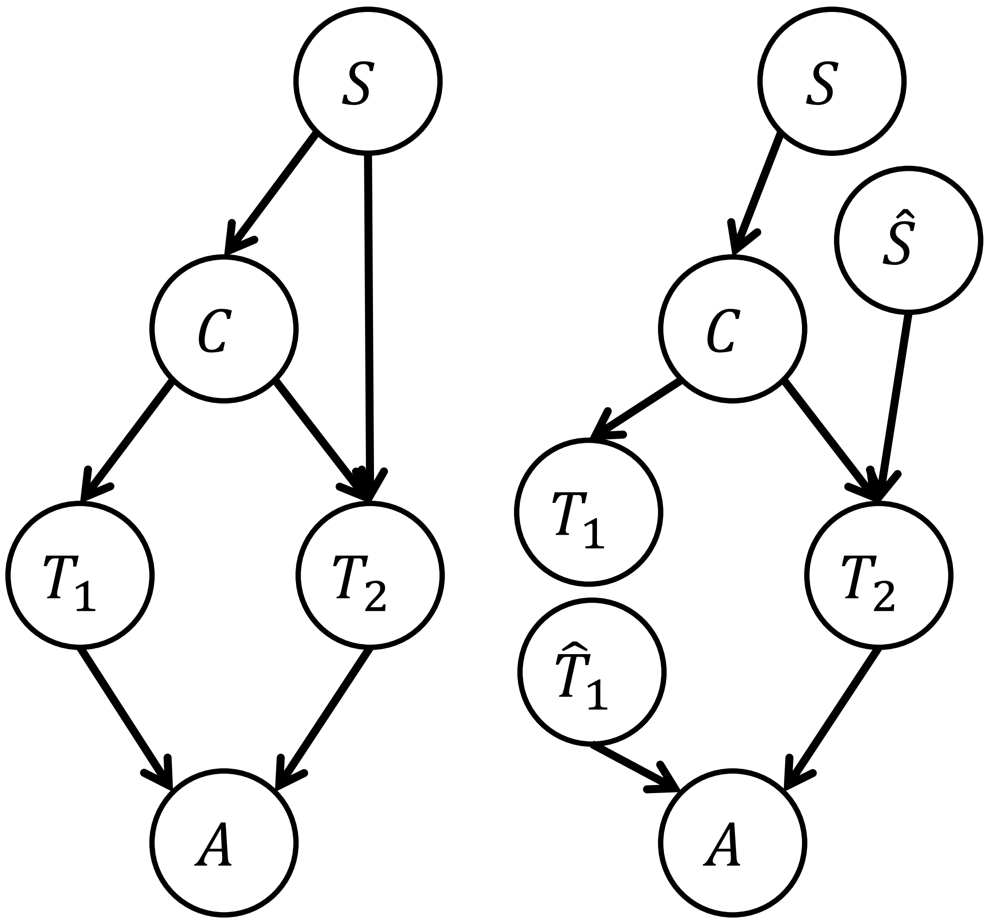
```
main:
   Q ← variables in the Bayes Net distinct from variables E
   s ← network instantiation compatible with evidence e        # global variable
   p ← probability of instantiation s                          # global variable
   DFS_MPE(e, Q)
   return s

DFS_MPE(i, X)
   if X is empty then                                          # i is a network instantiation
      if P(i) > p then
         s ← i; p ← P(i)
   else
      if MPE_u(i) ≤ p then return
      X ← a variable in X
      for each value x of variable X do
         DFS_MPE(ix, X\{X})
```

$MPE_u(i)$ stands for the computed upper bound on the MPE probability

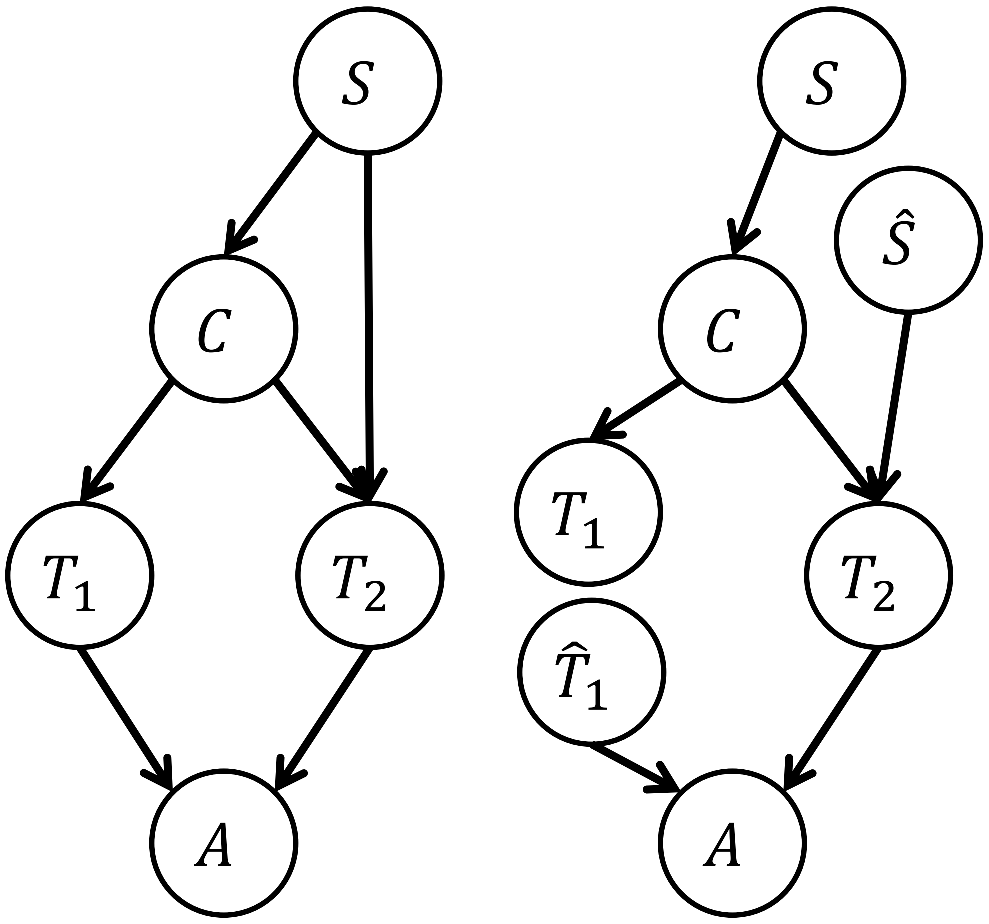
Upper Bounds by Node Splitting

- We discuss a technique that allows to generate a spectrum of upper bounds
  - In which we can trade off the bound tightness with the time it takes to compute it
- Consider the example network and a corresponding transformation
  - The tail edge $S \rightarrow T_2$ has been cloned by variable $\hat{S}$
  - Similarly, the tail of edge $T_1 \rightarrow A$ has been cloned by variable $\hat{T}_1$
  - Both clone variables are roots and have uniform CPTs
  - All other variables maintain their original CPTs (except that we need to replace S by its clone in the CPT of T_2. Similarly for the CPT of A)



Upper Bounds by Node Splitting

- These transformations can be used to reduce the treewidth of a network
  - To a point where applying algorithms such as VE becomes feasible
  - However, the transformed network does not induce the same distribution of the original network
  - But it does produce upper bounds on MPE probabilities

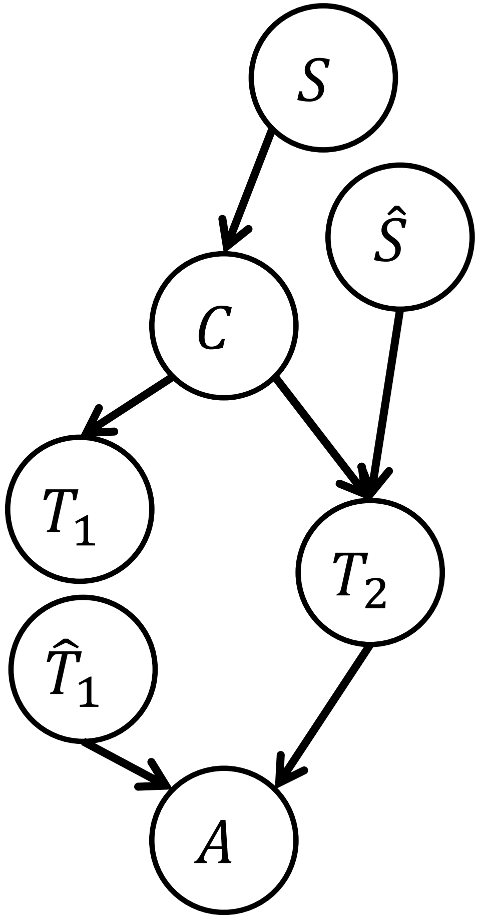


Upper Bounds by Node Splitting

- Suppose that $\boldsymbol{e}$ is the given evidence and $\hat{\boldsymbol{e}}$ is the instantiation of clone variables implied by $\boldsymbol{e}$
  - For every variable X instantiated to x in $\boldsymbol{e}$, its clone $\hat{X}$ will also be instantiated to x in $\boldsymbol{e}$.
- The following is guaranteed to hold

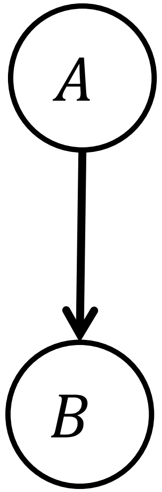
$$MPE_p(\boldsymbol{e}) \leq \beta \; MPE'_p(\boldsymbol{e}, \hat{\boldsymbol{e}})$$

  - $MPE_p(\boldsymbol{e})$ and $MPE'_p(\boldsymbol{e}, \hat{\boldsymbol{e}})$ are MPE probabilities for the original and transformed networks
  - β is the total number of instantiations for the cloned variables
- Therefore, we can compute an upper bound by performing inference on the transformed network
  - Which usually has low treewidth by design



The Effect of Node Splitting on Networks

- Let us analyse a simple case of node splitting
  - With only two variables
  - We computed the join $P(A, B)$



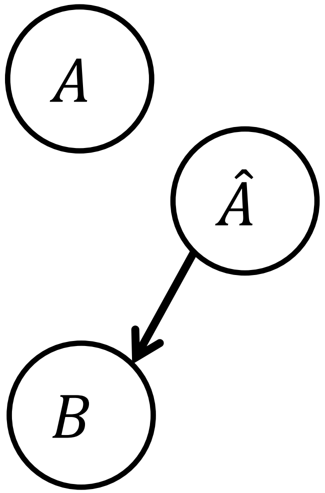
A	$P(A)$
a	.2
$\bar{a}$	.8

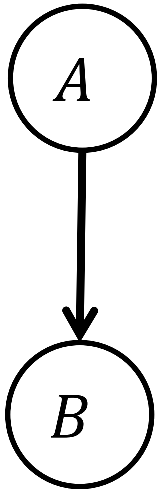
A	B	$P(B \mid A)$
a	b	.1
a	$\bar{b}$	.9
$\bar{a}$	b	.7
$\bar{a}$	$\bar{b}$	.3

A	B	$P(A, B)$
a	b	.02
a	$\bar{b}$	.18
$\bar{a}$	b	.56
$\bar{a}$	$\bar{b}$	.24

The Effect of Node Splitting on Networks

- Let us analyse a simple case of node splitting
  - With only two variables
  - We computed the join $P(A, B)$
- Now, we split node A according to B
  - $\hat{A}$ has a uniform distribution, i.e., $P(\hat{A}_i) = \frac{1}{|\hat{A}|}$
  - Notice $P'(A, \hat{A}, B)|\hat{A}| = P(A, B)$ for the matching instantiations of A and $\hat{A}$





A	$P(A)$
a	.2
$\bar{a}$	.8

A	B	$P(B\|A)$
a	b	.1
a	$\bar{b}$	.9
$\bar{a}$	b	.7
$\bar{a}$	$\bar{b}$	.3

A	B	$P(A,B)$
a	b	.02
a	$\bar{b}$	.18
$\bar{a}$	b	.56
$\bar{a}$	$\bar{b}$	.24

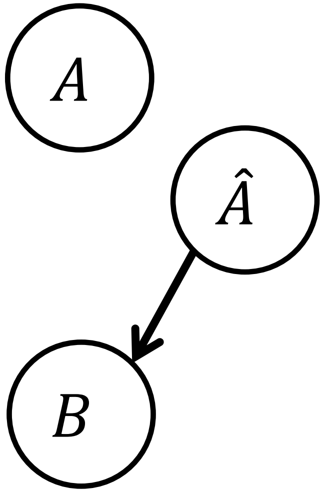
$\hat{A}$	$P'(\hat{A})$
$\hat{a}$	.5
$\bar{\hat{a}}$	.5

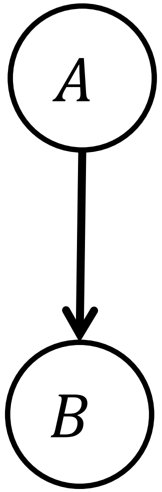
$\hat{A}$	B	$P'(B\|\hat{A})$
$\hat{a}$	b	.1
$\hat{a}$	$\bar{b}$	.9
$\bar{\hat{a}}$	b	.7
$\bar{\hat{a}}$	$\bar{b}$	.3

A	$\hat{A}$	B	$P'(A,\hat{A},B)$
a	$\hat{a}$	b	.01
a	$\hat{a}$	$\bar{b}$	.09
a	$\bar{\hat{a}}$	b	.07
a	$\bar{\hat{a}}$	$\bar{b}$	.03
$\bar{a}$	$\hat{a}$	b	.04
$\bar{a}$	$\hat{a}$	$\bar{b}$	.36
$\bar{a}$	$\bar{\hat{a}}$	b	.28
$\bar{a}$	$\bar{\hat{a}}$	$\bar{b}$	.12

The Effect of Node Splitting on Networks

- Considering the split cardinality as
  - $\beta = \prod_{C \in \boldsymbol{C}} |C|$, where $\boldsymbol{C}$ is the set of clone nodes
  - If we compute $P'(A, \hat{A}, B) \times \beta$ we can recover $P(A, B)$ including its MPE probability
  - However, the MPE probability in $P'(A, \hat{A}, B) \times \beta$ is an upper bound for the MPE probability in $P(A, B)$





A	$\hat{A}$	B	$P'(A, \hat{A}, B)$	$P'(A, \hat{A}, B)\beta$
a	$\hat{a}$	b	.01	.02
a	$\hat{a}$	$\bar{b}$	.09	.18
a	$\bar{\hat{a}}$	b	.07	.14
a	$\bar{\hat{a}}$	$\bar{b}$	.03	.06
$\bar{a}$	$\hat{a}$	b	.04	.08
$\bar{a}$	$\hat{a}$	$\bar{b}$	.36	.72
$\bar{a}$	$\bar{\hat{a}}$	b	.28	.56
$\bar{a}$	$\bar{\hat{a}}$	$\bar{b}$	.12	.24

A	B	$P(A, B)$
a	b	.02
a	$\bar{b}$	.18
$\bar{a}$	b	.56
$\bar{a}$	$\bar{b}$	.24

Upper Bounds by Node Splitting: Example

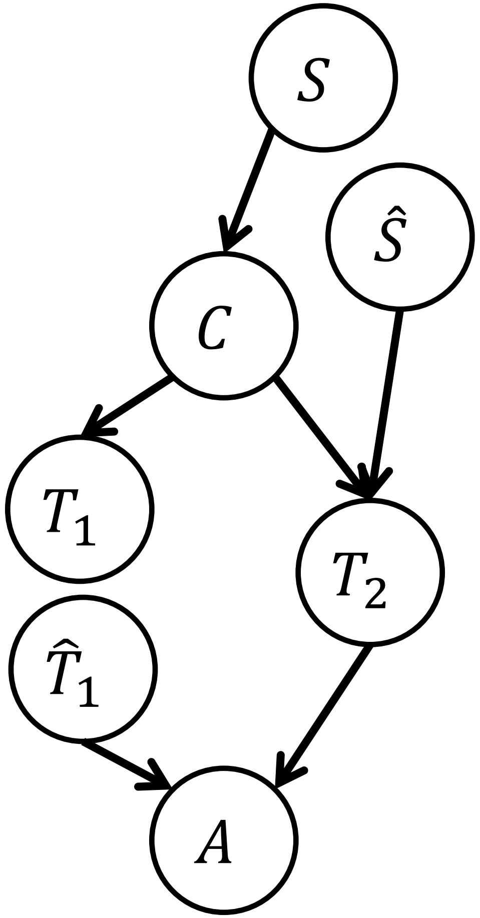
- If evidence $\boldsymbol{e}$ is $\{S = male, A = yes\}$, then $\hat{\boldsymbol{e}}$ is $\{\hat{S} = male\}$

$$MPE_p(\boldsymbol{e}) = .3344$$
$$MPE'_p(\boldsymbol{e}, \hat{\boldsymbol{e}}) = .0836$$
$$\beta = 4$$

- Leading to

$$.3344 \leq (4)(.0836) = .3344$$

- Therefore the upper bound is exact in this case



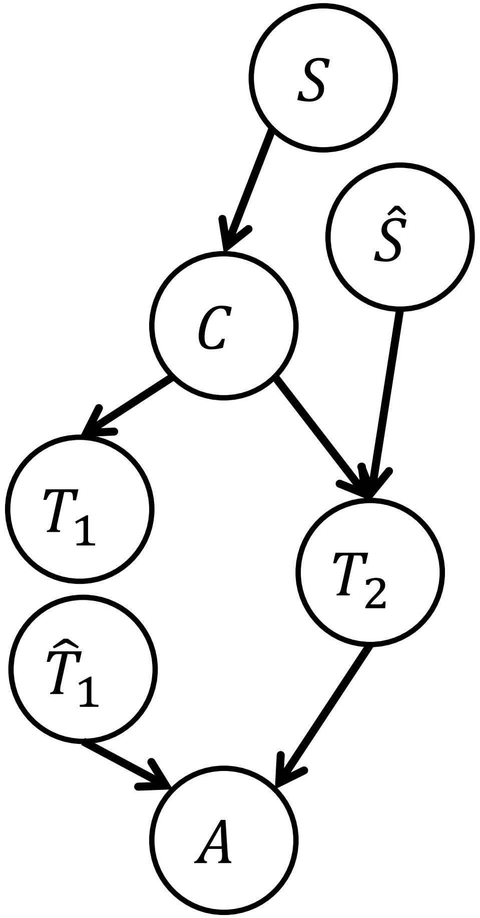
Upper Bounds by Node Splitting: Example

- If evidence $\boldsymbol{e}$ is only $\{A = yes\}$, then $\hat{\boldsymbol{e}}$ is empty

$$MPE_p(\boldsymbol{e}) = .33858$$
$$MPE'_p(\boldsymbol{e}, \hat{\boldsymbol{e}}) = .099275$$
$$\beta = 4$$

- Leading to

$$.33858 \leq (4)(.099275) = .3971$$



MPE Complexity Analysis

- Using VE, we can solve MPE problems
  - In time and space $O(n \exp(w))$ for n variables
- For branch-and-bound
  - Where we split m variables leading to a model with $n + m$ variables
  - An elimination order of width w' and use MPE VE for inference
  - The complexity of inference on the split network is $O\big((n + m) \exp(w')\big)$
  - As inference is performed in each node of the search tree and the search tree has $O(\exp(n))$ nodes
  - Total time complexity of $O\big((n + m)\exp(n + w')\big)$
- It does not look favourable to branch-and-bound, but we should note
  - Branch-and-bound reduces the exponential component from w to w' by splitting loops
  - The time complexity of VE is its worst and best-case. For branch-and-bound it is only the worst-case since pruning can improve the average-case

Computing MAP

- Given a network
  - The *MAP probability* for the variables $\boldsymbol{M}$ given evidence $\boldsymbol{e}$ is defined as
- There may be several instantiations $\boldsymbol{m}$ with maximal probability
  - Each of them is a MAP instantiation
  - The set of such instantiations is defined as
- MPE instantiations can be characterized as instantiations m that maximize the posterior distribution $P(\boldsymbol{m}|\boldsymbol{e})$
  - Since $P(\boldsymbol{m}|\boldsymbol{e}) = \frac{P(\boldsymbol{m},\boldsymbol{e})}{P(\boldsymbol{e})}$
  - $P(\boldsymbol{e})$ is independent of the instantiation $\boldsymbol{m}$

$$MAP_P(\boldsymbol{M},\boldsymbol{e}) \stackrel{\text{def}}{=} \max_{m} P(\boldsymbol{m},\boldsymbol{e})$$

$$MAP(\boldsymbol{M},\boldsymbol{e}) \stackrel{\text{def}}{=} \underset{m}{argmax}\, P(\boldsymbol{m},\boldsymbol{e})$$

$$MAP(\boldsymbol{M},\boldsymbol{e}) \stackrel{\text{def}}{=} \underset{q}{argmax}\, P(\boldsymbol{q}|\boldsymbol{e})$$

Computing MAP by Variable Elimination

- We can compute the MAP probability $MAP_P(\boldsymbol{M}, \boldsymbol{e})$ using the VE algorithm
  - First, summing out all non-MAP variables: computes the marginal $P(\boldsymbol{M}, \boldsymbol{e})$ in factored form
  - Second, maximizing out MAP variables $\boldsymbol{M}$: solve MPE problem over the resulting marginal

- The resulting algorithm can be thought of a combination of MPE and VE algorithms

- We can use extended factors just as when computing an MPE instantiation

MAP VE: Algorithm

$\boldsymbol{Q} \leftarrow$ variables in the network

$\pi \leftarrow$ elimination order of variables $\boldsymbol{Q}$ in which variables $\boldsymbol{M}$ appear last

$\boldsymbol{S} \leftarrow \{\phi^{\boldsymbol{e}} : \phi$ is a factor of the network$\}$

for $i = 1$ **to** length of order π **do**

  $\sigma_i \leftarrow \prod_k \phi_k$, where ϕ_k belongs to $\boldsymbol{S}$ and mentions variable $\pi(i)$

  if $\pi(i) \in \boldsymbol{M}$ **then**

    $\tau_i \leftarrow \max_{\pi(i)} \sigma_i$

  else

    $\tau_i \leftarrow \sum_{\pi(i)} \sigma_i$

  replace all factors ϕ_k in $\boldsymbol{S}$ by factor τ_i

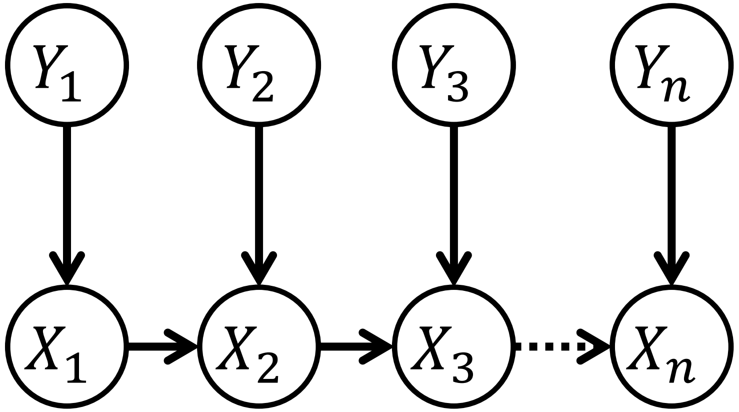
return trivial factor $\prod_{\tau \in S} \tau$

Notes:

- If the network is a Bayesian network, you can prune nodes and edges
- The elimination is special in the sense the MAP variables appear last
- The algorithm perform both types of elimination: maximizing-out MAP variables and summing-out non-MAP variables

MAP VE Complexity

- Given n variables and an elimination order of width w
  - The time and space complexity of MAP is $O(n \exp(w))$
  - Like MPE VE algorithm
- However, MAP variable order is constrained
  - It requires MAP variables to be last in the order
  - This means that we may not be able to use a good ordering because low-width orders do not satisfy this constraint
- For example, the polytree structure on the right
  - It has treewidth of 2 since it has at most two parents per node
  - If we want to compute MAP for variables $\boldsymbol{M} = \{Y_1, \dots, Y_n\}$, any order that $\boldsymbol{M}$ comes last has $width \geq n$
  - Therefore, MPE VE is linear, but MAP VE is exponential in this case



MAP VE Complexity

- In general, we cannot use arbitrary elimination orders
  - We cannot interleave variables that are summing out with those maximizing out
  - Maximization does not commute with summation

$$\left[\sum_X \max_Y f\right](\boldsymbol{z}) \geq \left[\max_Y \sum_X f\right](\boldsymbol{z})$$

for all instantiations $\boldsymbol{z}$

- The complexity of MAP VE is at best exponential in the constrained treewidth
  - A variable order π is **M**-constrained iff variables **M** appear last in the order π.
  - The ***M**-constrained treewidth* of a graph is the width of its best **M**-constrained variable order

- Computing MAP is therefore more difficult than computing MPE in the context of VE

Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$

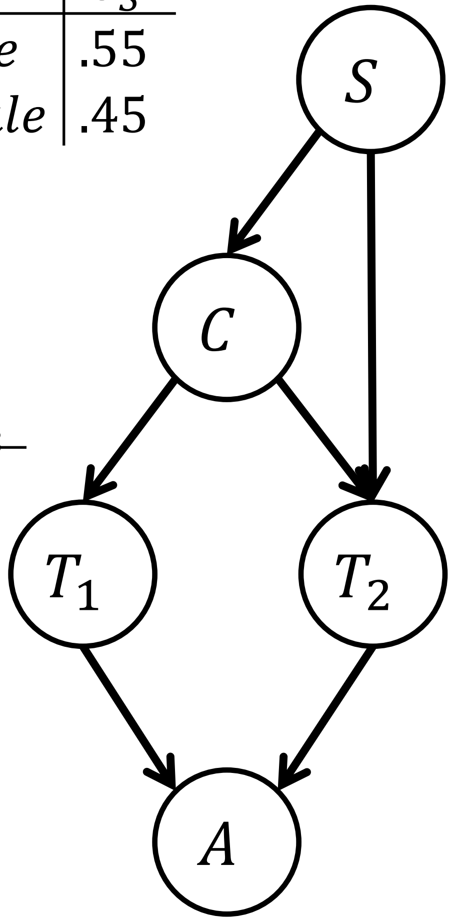
S	C	$\Theta_{C\|S}$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\Theta_{T_1\|C}$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	Θ_S
$male$	.55
$female$	.45

S	C	T_2	$\Theta_{T_2\|C,S}$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\Theta_{A\|T_1,T_2}$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



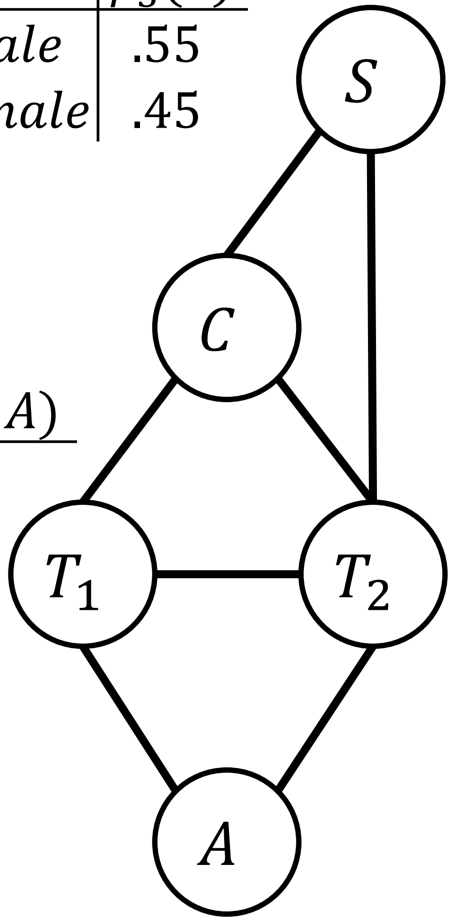
Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$

S	C	$\phi_1(S,C)$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\phi_2(T_1,C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	$\phi_3(S)$
$male$	.55
$female$	.45



S	C	T_2	$\phi_4(T_2,C,S)$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\phi_5(T_1,T_2,A)$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0

Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$

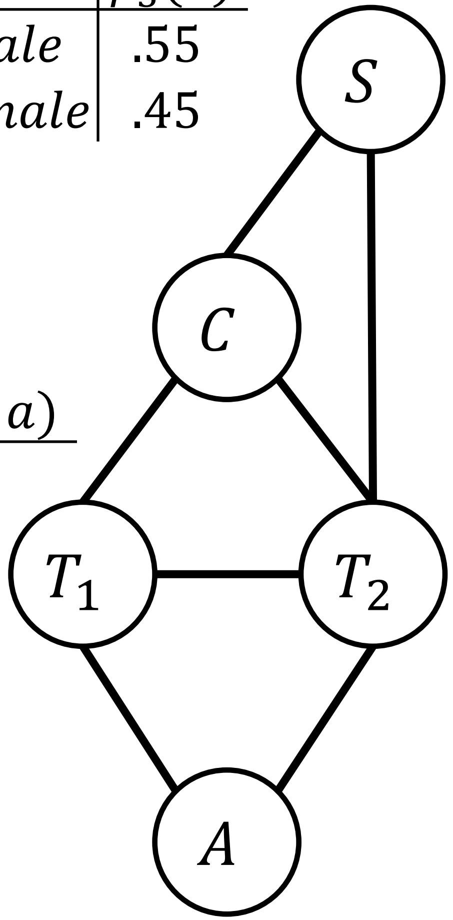
S	C	$\phi_1(S, C)$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	$\phi_3(S)$
$male$	.55
$female$	.45

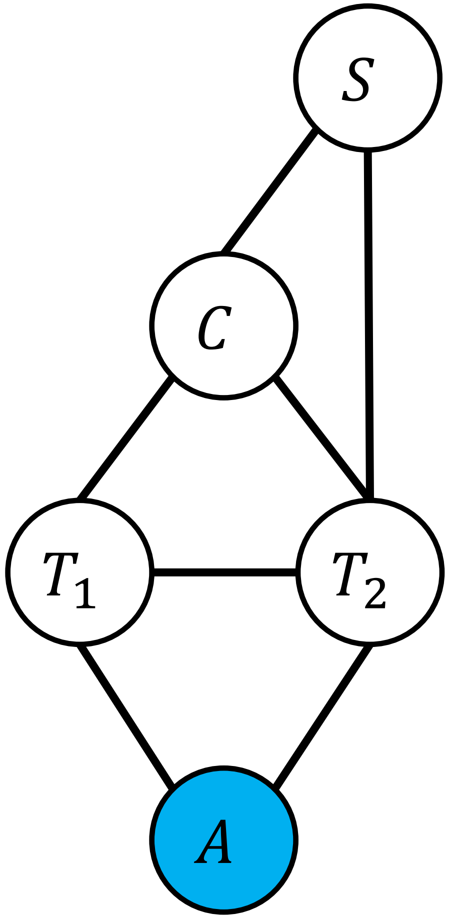
S	C	T_2	$\phi_4(T_2, C, S)$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	a	$\phi_5(T_1, T_2, a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1



Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$



T_1	T_2	a	$\phi_5(T_1, T_2, a)$
ve	ve	yes	1
ve	$\overline{ve}$	yes	0
$\overline{ve}$	ve	yes	0
$\overline{ve}$	$\overline{ve}$	yes	1

T_1	T_2	$\tau_1(T_1, T_2)$
ve	ve	1
ve	$\overline{ve}$	0
$\overline{ve}$	ve	0
$\overline{ve}$	$\overline{ve}$	1

Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$

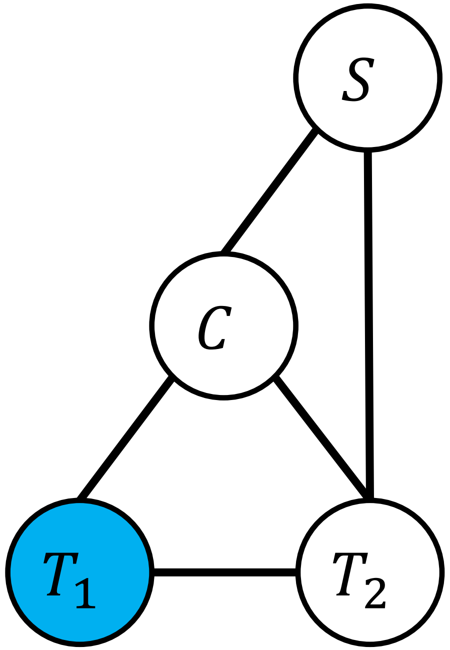
T_1	T_2	$\tau_1(T_1, T_2)$
ve	ve	1
ve	$\overline{ve}$	0
$\overline{ve}$	ve	0
$\overline{ve}$	$\overline{ve}$	1

$\times$

C	T_1	$\phi_2(T_1, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

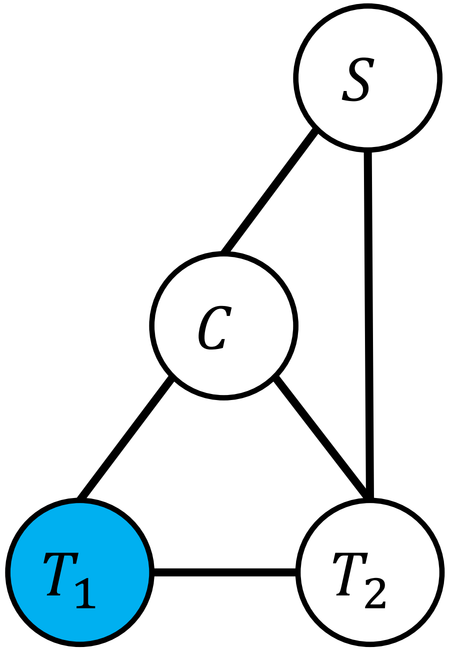
$=$

C	T_1	T_2	$\sigma_2(T_1, T_2, C)$
yes	ve	ve	.80
yes	ve	$\overline{ve}$	0
yes	$\overline{ve}$	ve	0
yes	$\overline{ve}$	$\overline{ve}$	.20
no	ve	ve	.20
no	ve	$\overline{ve}$	0
no	$\overline{ve}$	ve	0
no	$\overline{ve}$	$\overline{ve}$	.80



Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$



C	T_1	T_2	$\sigma_2(T_1, T_2, C)$
yes	ve	ve	.80
yes	ve	$\overline{ve}$	0
yes	$\overline{ve}$	ve	0
yes	$\overline{ve}$	$\overline{ve}$	.20
no	ve	ve	.20
no	ve	$\overline{ve}$	0
no	$\overline{ve}$	ve	0
no	$\overline{ve}$	$\overline{ve}$	.80

C	T_2	$\tau_2(T_2, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$

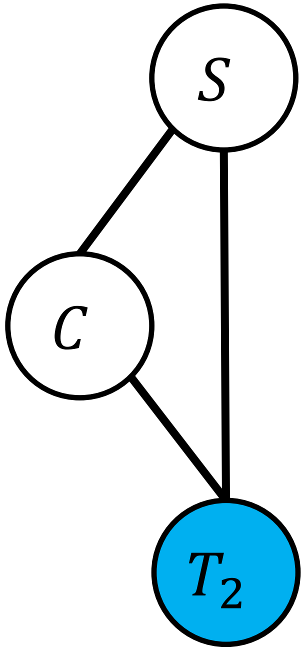
C	T_2	$\tau_2(T_2, C)$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

$\times$

S	C	T_2	$\phi_4(T_2, C, S)$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

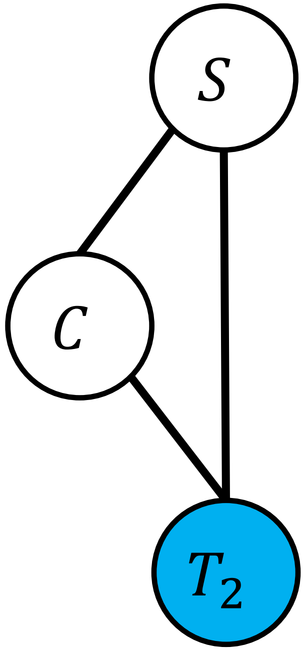
$=$

S	C	T_2	$\sigma_3(T_2, S, C)$
$male$	yes	ve	.64
$male$	yes	$\overline{ve}$	.04
$male$	no	ve	.04
$male$	no	$\overline{ve}$	.64
$female$	yes	ve	.76
$female$	yes	$\overline{ve}$	.01
$female$	no	ve	.01
$female$	no	$\overline{ve}$	.76



Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$

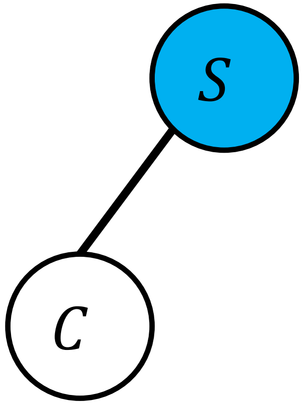


S	C	T_2	$\sigma_3(T_2, S, C)$
$male$	yes	ve	.64
$male$	yes	$\overline{ve}$	.04
$male$	no	ve	.04
$male$	no	$\overline{ve}$	.64
$female$	yes	ve	.76
$female$	yes	$\overline{ve}$	.01
$female$	no	ve	.01
$female$	no	$\overline{ve}$	.76

S	C	$\tau_3(C, S)$
$male$	yes	.68
$male$	no	.68
$female$	yes	.77
$female$	no	.77

Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$



S	C	$\tau_3(C,S)$
$male$	yes	.68
$male$	no	.68
$female$	yes	.77
$female$	no	.77

$\times$

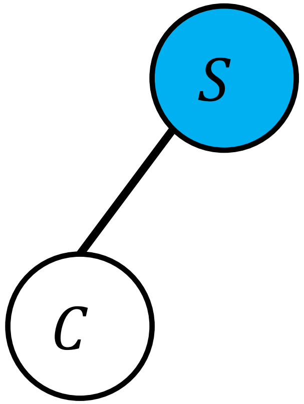
S	$\phi_3(S)$
$male$	.55
$female$	.45

$\approx$

S	C	$\sigma_4(C,S)$
$male$	yes	.37
$male$	no	.37
$female$	yes	.35
$female$	no	.35

Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$



S	C	$\sigma_4(C,S)$
$male$	yes	$.37$
$male$	no	$.37$
$female$	yes	$.35$
$female$	no	$.35$

$\times$

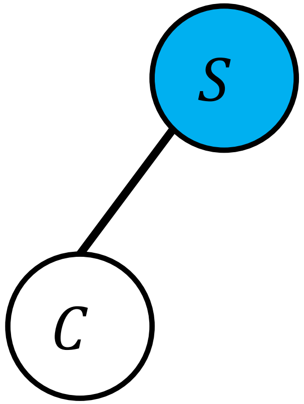
S	C	$\phi_1(S,C)$
$male$	yes	$.05$
$male$	no	$.95$
$female$	yes	$.01$
$female$	no	$.99$

$\approx$

S	C	$\sigma_5(C,S)$
$male$	yes	$.018$
$male$	no	$.351$
$female$	yes	$.004$
$female$	no	$.347$

Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$

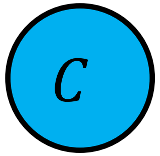


S	C	$\sigma_5(C,S)$
$male$	yes	.018
$male$	no	.351
$female$	yes	.004
$female$	no	.347

C	$\tau_5(C)$	
yes	.018	$male$
no	.351	$male$

Computing MAP Instantiation: Example

- Let us run MAP VE with evidence $A = yes$ and MAP variables S and C, and elimination order $\pi = A, T_1, T_2, S, C$
- Since $P(\boldsymbol{e}) = P(A = yes) = .7205$
  - $MAP_p(S, C|\boldsymbol{e}) \approx 49\%$



C	$\tau_5(C)$	
yes	$.018$	$male$
no	$.351$	$male$

τ_6	
$.351$	$male, no$

Computing MAP by Systematic Search

- MAP can be solved with depth-first branch-and-bound search
  - Similar to MPE
- The upper bound can be computed based on the split network
  - However, the quality for this upper bound is somewhat loose in practice
  - Compared to the MPE case
- We later introduce a different upper bound for MAP

DFS MAP: Algorithm

main:

$\boldsymbol{m} \leftarrow$ some instantiation of variables $\boldsymbol{M}$ # global variable

$p \leftarrow$probability of instantiation $\boldsymbol{m}, \boldsymbol{e}$ # global variable

DFS_MAP($\boldsymbol{e}$, $\boldsymbol{M}$)

return $\boldsymbol{m}$

DFS_MAP($\boldsymbol{i}, \boldsymbol{X}$)

if $\boldsymbol{X}$ is empty **then** # leaf node

if $P(\boldsymbol{i}) > p$ **then**

$\boldsymbol{m} \leftarrow \boldsymbol{i}$

$p \leftarrow P(\boldsymbol{i})$

else if $MAP_u(\boldsymbol{X}, \boldsymbol{i}) > p$ **then**

$X \leftarrow$ a variable in $\boldsymbol{X}$

for each value x of variable X **do**

DFS_MAP($\boldsymbol{i}x, \boldsymbol{X} \backslash \{X\}$)

$P(\boldsymbol{i})$ does not involve all variables anymore.
We will need to do inference to compute this

The previous bound based on network splitting is shown to be less tight than for the MPE case.
We will explore a new bound

Computing $P(i)$ for MPE and MAP

- In MPE, all variables but the evidence are in the query
  - Therefore, computing $P(i)$ is linear time for any instantiation i
  - For instance, $P(male, no, \overline{ve}, \overline{ve}, yes) =$ $.95 \times .80 \times .55 \times .80 \times 1 = .3344$

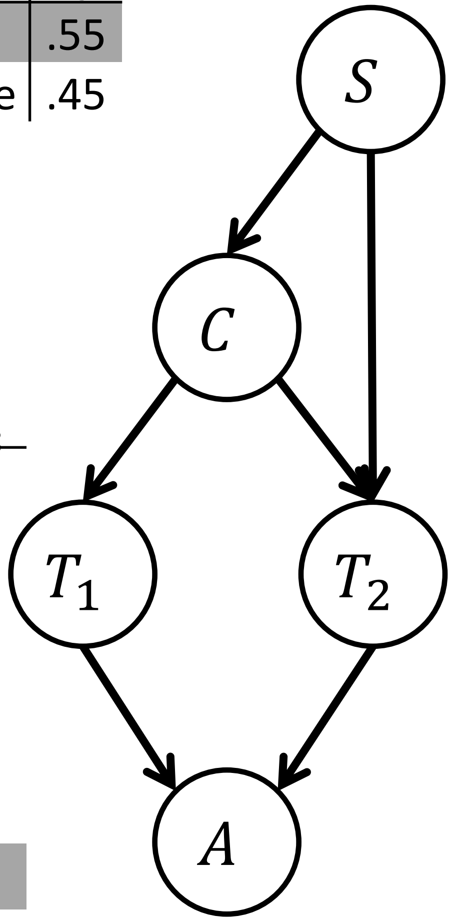
S	C	$\Theta_{C\|S}$
$male$	yes	.05
$male$	no	.95
$female$	yes	.01
$female$	no	.99

C	T_1	$\Theta_{T_1\|C}$
yes	ve	.80
yes	$\overline{ve}$	.20
no	ve	.20
no	$\overline{ve}$	.80

S	Θ_S
male	.55
female	.45

S	C	T_2	$\Theta_{T_2\|C,S}$
$male$	yes	ve	.80
$male$	yes	$\overline{ve}$	.20
$male$	no	ve	.20
$male$	no	$\overline{ve}$	.80
$female$	yes	ve	.95
$female$	yes	$\overline{ve}$	.05
$female$	no	ve	.05
$female$	no	$\overline{ve}$	.95

T_1	T_2	A	$\Theta_{A\|T_1,T_2}$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



Computing $P(i)$ for MPE and MAP

- In MPE, all variables but the evidence are in the query
  - Therefore, computing $P(i)$ is linear time for any instantiation i
  - For instance, $P(male, no, \overline{ve}, \overline{ve}, yes) =$ $.95 \times .80 \times .55 \times .80 \times 1 = .3344$
- In MAP, MAP variables are a subset of all variables
  - Computing $P(i)$ such as $P(S = male, C = yes | A = yes)$ involves variable elimination
  - $P(S = male, C = yes | A = yes) \approx .0260$

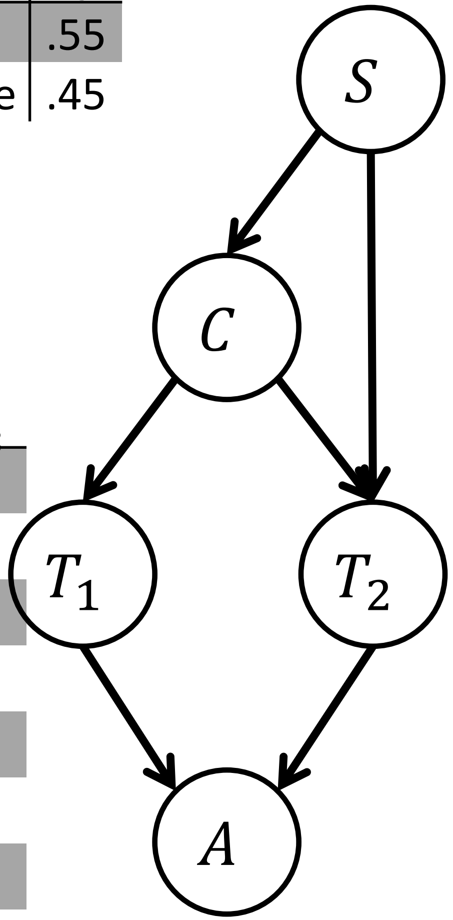
S	C	$\Theta_{C\|S}$
$male$	yes	$.05$
$male$	no	$.95$
$female$	yes	$.01$
$female$	no	$.99$

C	T_1	$\Theta_{T_1\|C}$
yes	ve	$.80$
yes	$\overline{ve}$	$.20$
no	ve	$.20$
no	$\overline{ve}$	$.80$

S	Θ_S
male	.55
female	.45

S	C	T_2	$\Theta_{T_2\|C,S}$
$male$	yes	ve	$.80$
$male$	yes	$\overline{ve}$	$.20$
$male$	no	ve	$.20$
$male$	no	$\overline{ve}$	$.80$
$female$	yes	ve	$.95$
$female$	yes	$\overline{ve}$	$.05$
$female$	no	ve	$.05$
$female$	no	$\overline{ve}$	$.95$

T_1	T_2	A	$\Theta_{A\|T_1,T_2}$
ve	ve	yes	1
ve	ve	no	0
ve	$\overline{ve}$	yes	0
ve	$\overline{ve}$	no	1
$\overline{ve}$	ve	yes	0
$\overline{ve}$	ve	no	1
$\overline{ve}$	$\overline{ve}$	yes	1
$\overline{ve}$	$\overline{ve}$	no	0



Upper Bound for MAP

- We propose a different upper bound for MAP queries. It comes from the following observation
  - If we use an arbitrary elimination order on Line 2 of algorithm MAP VE, the number q it returns satisfies

$$MAP_p(\boldsymbol{M}, \boldsymbol{e}) \leq q \leq P(\boldsymbol{e})$$

- Remember that maximization does not commute with summation
  - If we precede some maximizations of summations, we get an upper bound

$$\left[\sum_X \max_Y f\right](\boldsymbol{z}) \geq \left[\max_Y \sum_X f\right](\boldsymbol{z})$$

- We cannot use arbitrary elimination orders for computing MAP
  - But we can use such an order to compute an upper bound
  - Some orders produce better upper bounds than others
  - The goal is to find an order that produces one of the better bounds

MAP VE: Algorithm

$\boldsymbol{Q} \leftarrow$ variables in the network

$\pi \leftarrow$ elimination order of variables $\boldsymbol{Q}$ in which variables $\boldsymbol{M}$ appear last

$S \leftarrow \{\phi^{\boldsymbol{e}} : \phi$ is a factor of the PGM$\}$

for $i = 1$ **to** length of order π **do**

$\sigma_i \leftarrow \prod_k \phi_k$, where ϕ_k belongs to S and mentions variable $\pi(i)$

if $\pi(i) \in \boldsymbol{M}$ **then**

$$\tau_i \leftarrow \max_{\pi(i)} \sigma_i$$

else

$$\tau_i \leftarrow \sum_{\pi(i)} \sigma_i$$

replace all factors ϕ_k in S by factor τ_i

return trivial factor $\prod_{\tau \in S} \tau$

π is now an arbitrary order, not $\boldsymbol{M}$-constrained

We are calling this quantity q

Upper Bound for MAP

- We propose a different upper bound for MAP queries. It comes from the following observation
  - If we use an arbitrary elimination order on Line 2 of algorithm MAP VE, the number q it returns satisfies

$$MAP_p(\boldsymbol{M}, \boldsymbol{e}) \leq q \leq P(\boldsymbol{e})$$

- Remember that maximization does not commute with summation
  - If we precede some maximizations of summations, we get an upper bound

$$\left[\sum_X \max_Y f\right](\boldsymbol{z}) \geq \left[\max_Y \sum_X f\right](\boldsymbol{z})$$

- We cannot use arbitrary elimination orders for computing MAP
  - But we can use such an order to compute an upper bound
  - Some orders produce better upper bounds than others
  - The goal is to find an order that produces one of the better bounds

MAP Complexity Analysis

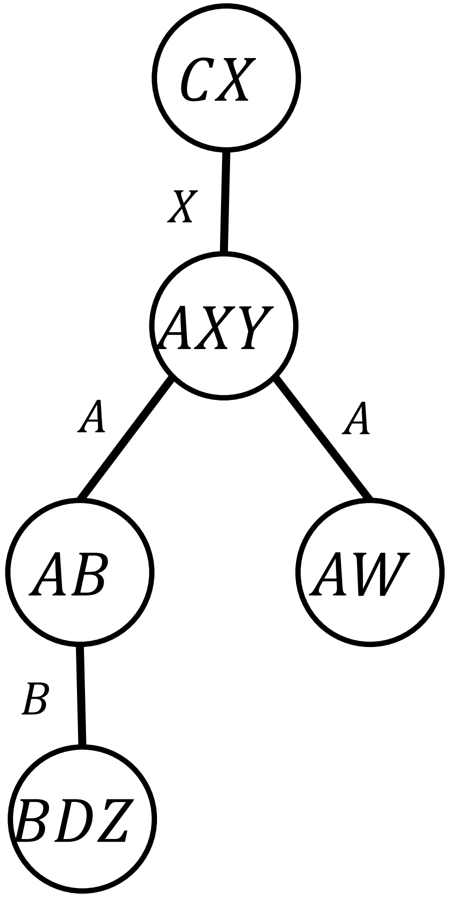
- Suppose we have
  - n variables
  - an elimination order of width w
  - a constrained elimination order of $w + c$
- MAP variable elimination takes $O(n \exp(w + c))$
- DFS MAP
  - The search tree has $O(\exp(m))$, where m is the number of MAP variables
  - The computation of $P(i)$ for each leaf node takes $O(n \exp(w))$ time and space with VE
  - The computation of the upper bound, MAP_u, has a similar complexity
  - Therefore, the algorithm has
    - Space complexity of $O(n \exp(w))$
    - Time complexity of $O(n \exp(w + m))$

MAP Complexity Analysis

- Comparing DFS MAP and MAP VE
  - DFS MAP has better space complexity
  - The time complexity depends on the constrained width $w + c$ and number of MAP variables m
- Even if $m > c$, DFS MAP may perform better
  - The complexity of variable elimination is the best and worst case at the same time
  - The complexity for DFS MAP is the worst case
  - The pruning may lead to better average complexity in practice

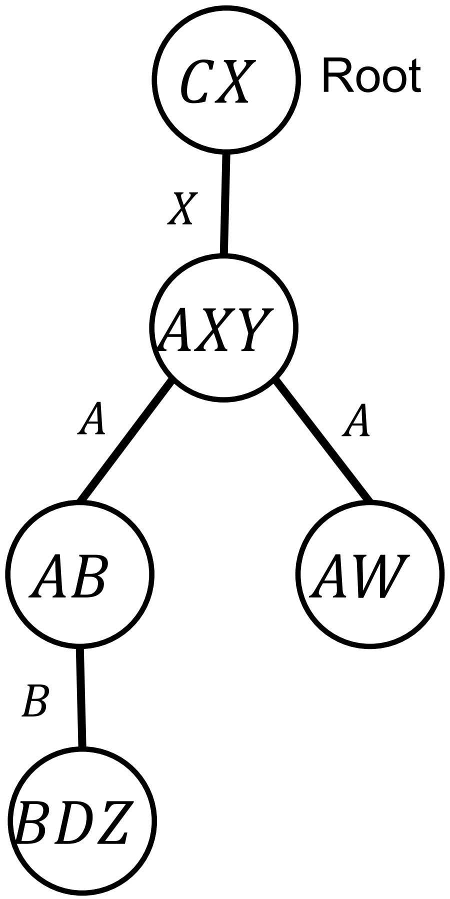
Improving the Upper Bound

- As mentioned before, any elimination order can be used to produce an upper bound
  - But some orders will produce better upper bounds than others
  - The closer is the order to the constrained order, the tighter the bound tends to be
- A technique to select one of the better unconstrained orders
  - Remember, we can convert jointrees to elimination orders
  - For instance, considering the jointree on the right, we can generate several elimination orders, including
    - $\pi_1 = D, Z, B, W, A, Y, C, X$
    - $\pi_2 = Z, D, W, B, Y, A, X, C$



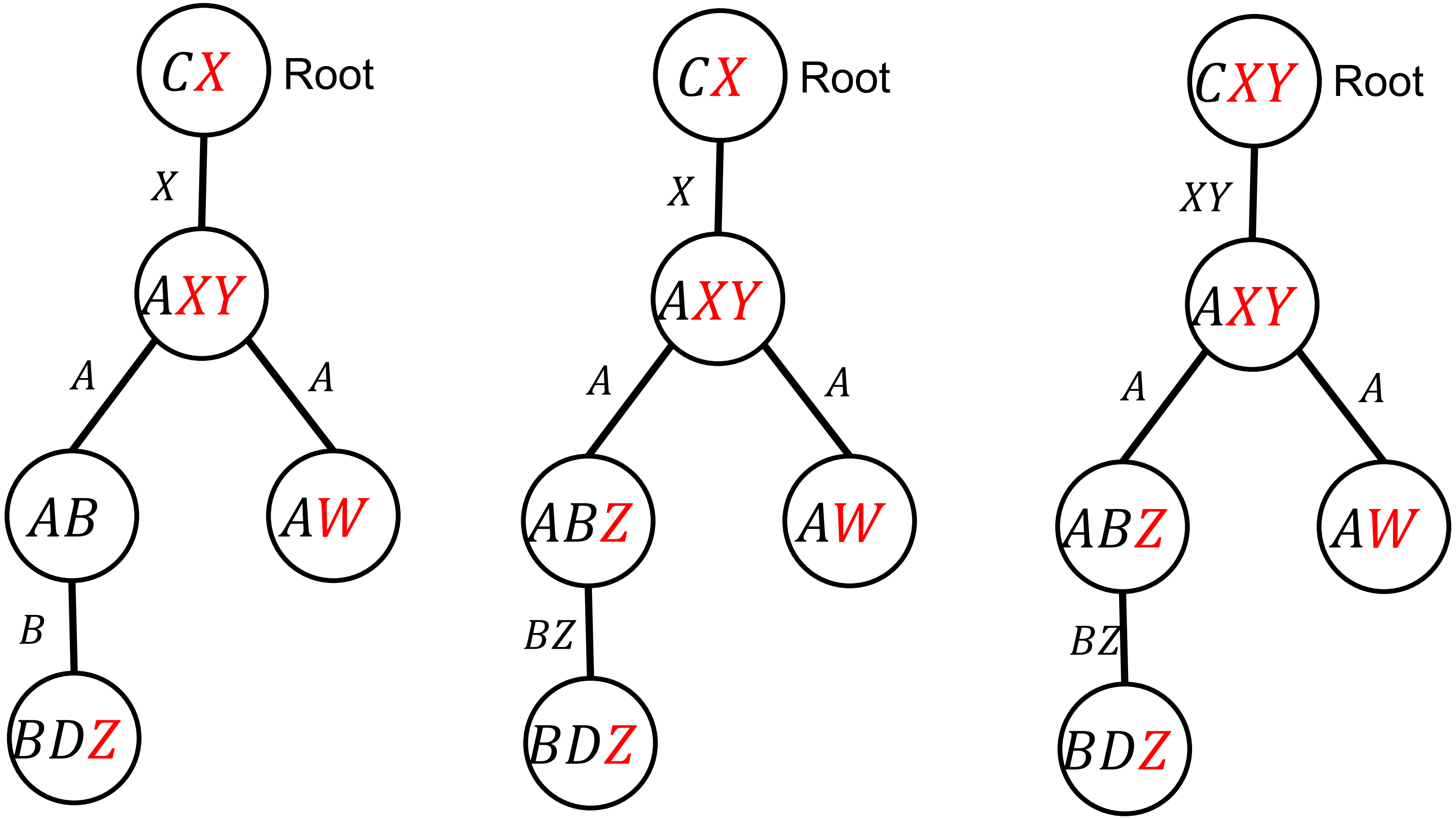
Improving the Upper Bound

- Suppose that MAP variables are W, X, Y, Z
  - Neither π_1 nor π_2 are m-constrained
  - But π_1 will produce better bound as MAP variables appear later
- We can transform a jointree of width w into another jointree
  - With the same width w
  - That induces elimination orders closer to a constrained order
- The idea is to promote MAP variables toward the root without increasing the jointree width w
  - Let $\boldsymbol{C}_k$ and $\boldsymbol{C}_i$ be two adjacent clusters where $\boldsymbol{C}_i$ is closer to the root
  - Any MAP variable that appears in $\boldsymbol{C}_k$ is added to $\boldsymbol{C}_i$ as long the new cluster size does not exceed $w + 1$
  - This is the "add variable" operation over jointrees

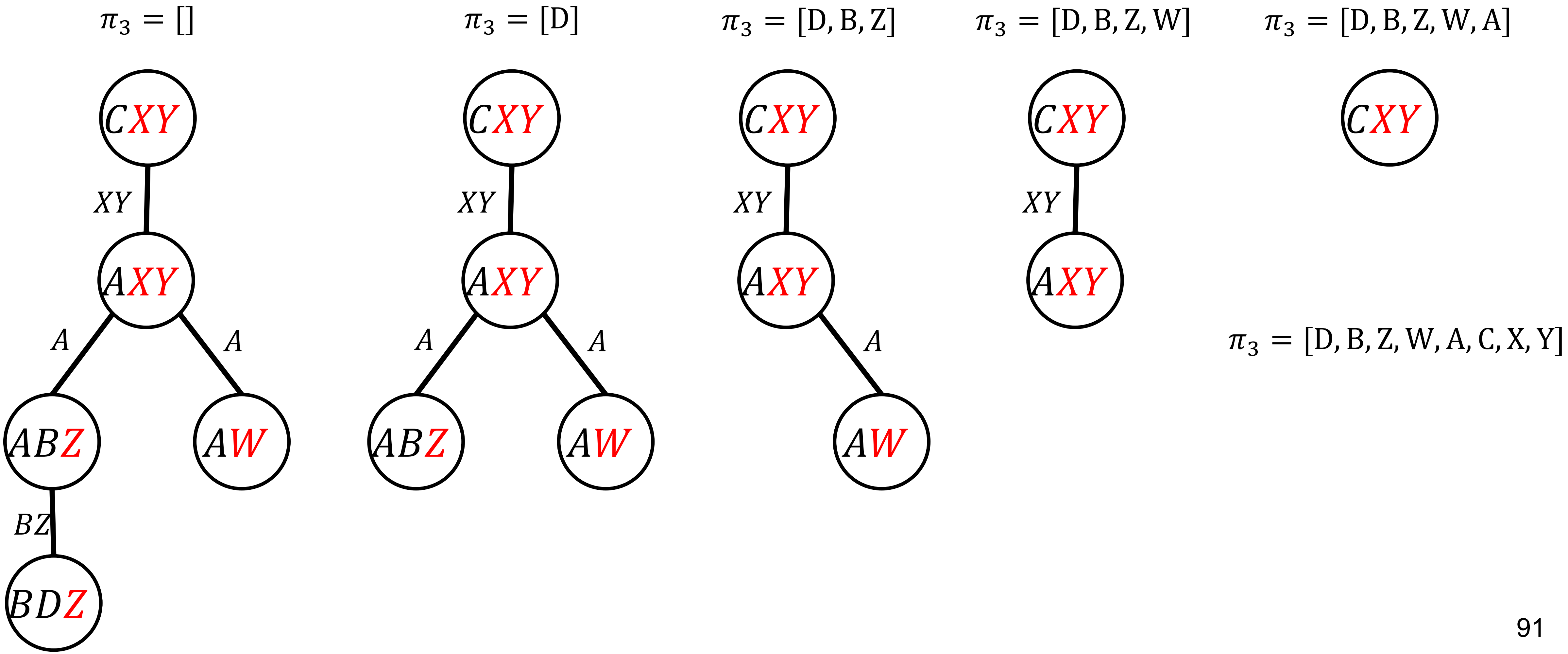


$\pi_1 = D, Z, B, W, A, Y, C, X$
$\pi_2 = Z, D, W, B, Y, A, X, C$

Improving the Upper Bound

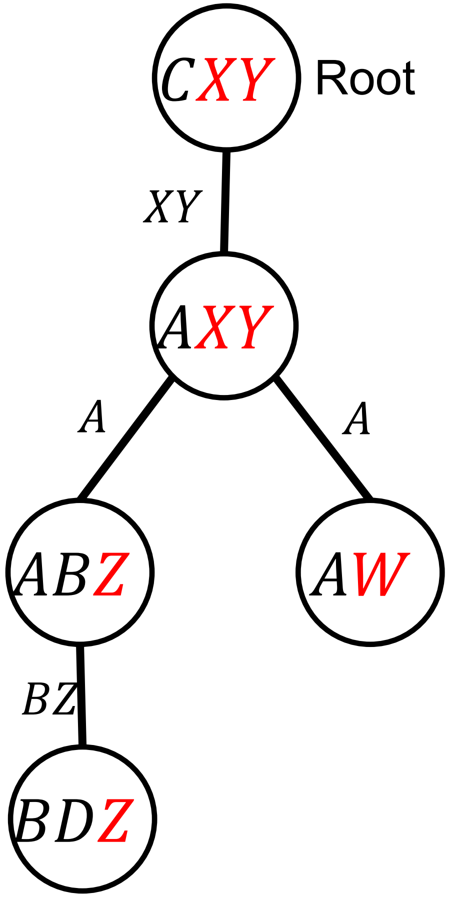


Improving the Upper Bound



Improving the Upper Bound

- π_3 is closer to the constrained order than π_1
  - Because MAP variables Y and Z appear later in the order
- Each time a MAP variable is promoted to a cluster closer to the root
  - The elimination of that variable is postponed
  - Pushing it past all non-MAP variables that are eliminated in original cluster
- This has a monotonic effect in the upper bound
  - Although in rare cases, the upper bound remains the same
- This technique improves the bounds computed by elimination orders that are induced by the jointree and a selected root
  - The quality induced by some other root may worsen



$\pi_1 = D, Z, B, W, A, Y, C, X$
$\pi_2 = Z, D, W, B, Y, A, X, C$
$\pi_3 = D, B, Z, W, A, C, X, Y$

MPE and Belief Propagation

- We can easily adapt the jointree algorithm to compute MPE assignments
  - We messages will be computed using maximization and multiplication operations
  - We frequently call this approach as the *max-product algorithm*
  - While the original algorithm for computing marginal probabilities is known as *sum-product algorithm*

X	Y	P
0	0	8
0	1	1
1	0	0.5
1	1	2

↓

X	Y	$\log_2 P$
0	0	3
0	1	0
1	0	-1
1	1	1

- However, it also very common to convert the probabilities to log-probabilities
  - The max-product algorithm becomes *max-sum algorithm*
  - Usually numerically more stable than max-product

Factor Summation

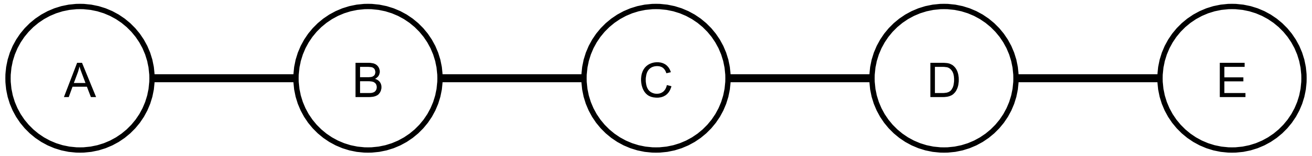
- We need to define a new operation: factor summation
  - Similar to factor multiplication
  - We match rows based on common variables (if any) creating a new factor
  - Add add the rows according to the value of the variables

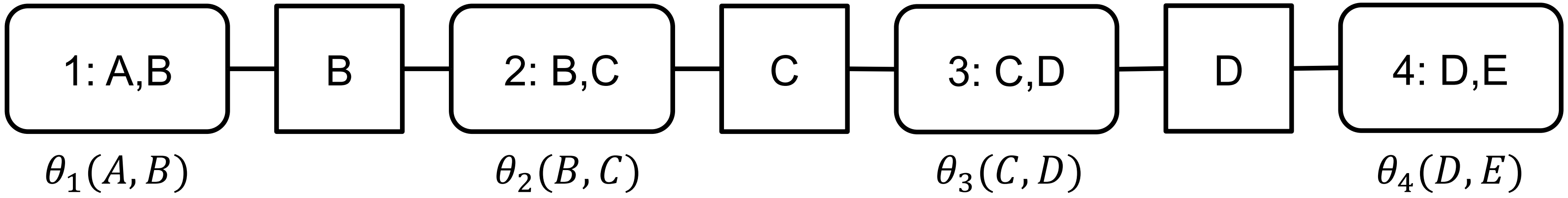
A	B	
0	0	3
0	1	0
1	0	-1
1	1	1

B	C	
0	0	4
0	1	1.5
1	0	0.2
1	1	2

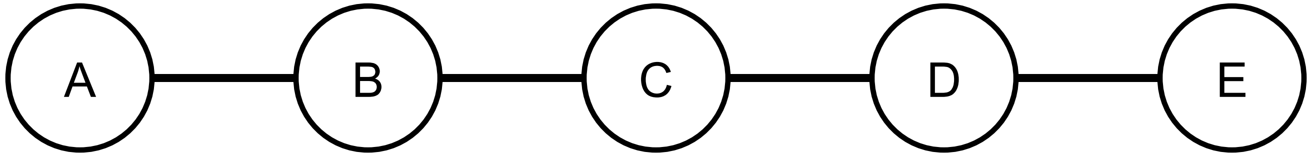
A	B	C	
0	0	0	$3 + 4 = 7$
0	0	1	$3 + 1.5 = 4.5$
0	1	0	$0.0 + 02 = 0.2$
0	1	1	$0 + 2 = 2$
1	0	0	$-1 + 4 = 3$
1	0	1	$-1 + 1.5 = 0.5$
1	1	0	$1 + 0.2 = 1.2$
1	1	1	$1 + 2 = 3$

Max-Sum in Jointrees

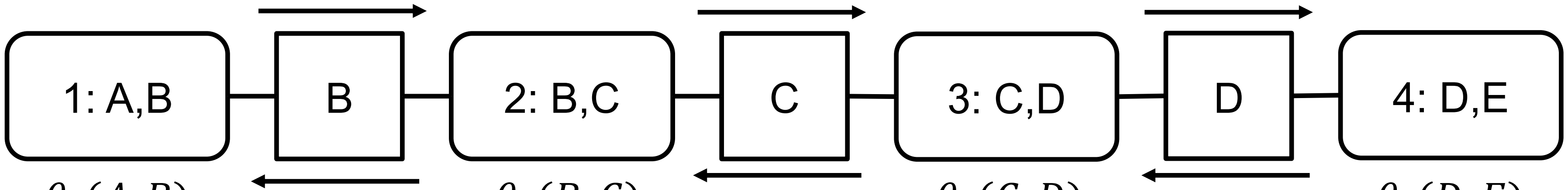




Max-Sum in Jointrees



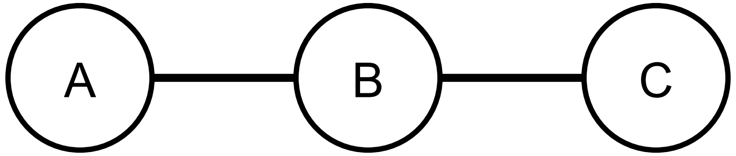
$M_{1,2}(B) = max_A \theta_1$ $M_{2,3}(C) = max_B(\theta_2 + M_{1,2})$ $M_{3,4}(D) = max_C(\theta_3 + M_{2,3})$



$\theta_1(A,B)$ $\theta_2(B,C)$ $\theta_3(C,D)$ $\theta_4(D,E)$

$M_{2,1}(B) = max_C(\theta_2 + M_{3,2})$ $M_{3,2}(C) = max_D(\theta_3 + M_{4,3})$ $M_{4,3}(D) = max_E \theta_4$

Max-Sum: Simple Example



A	B	$\theta_1(A,B)$
a	b	3
a	$\bar{b}$	0
$\bar{a}$	b	-1
$\bar{a}$	$\bar{b}$	1

B	C	$\theta_2(B,C)$
b	c	4
b	$\bar{c}$	1.5
$\bar{b}$	c	0.2
$\bar{b}$	$\bar{c}$	2

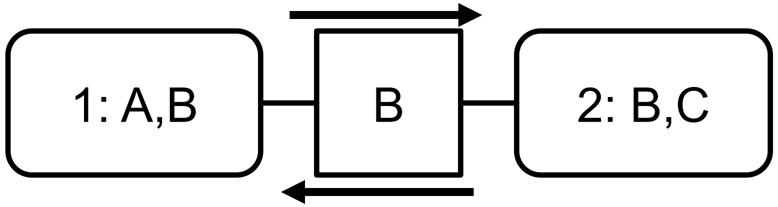
A	B	C	$\theta = \theta_1 + \theta_2$
a	b	c	$3 + 4 = 7$
a	b	$\bar{c}$	$3 + 1.5 = 4.5$
a	$\bar{b}$	c	$0.0 + 02 = 0.2$
a	$\bar{b}$	$\bar{c}$	$0 + 2 = 2$
$\bar{a}$	b	c	$-1 + 4 = 3$
$\bar{a}$	b	$\bar{c}$	$-1 + 1.5 = 0.5$
$\bar{a}$	$\bar{b}$	c	$1 + 0.2 = 1.2$
$\bar{a}$	$\bar{b}$	$\bar{c}$	$1 + 2 = 3$

Max-Sum: Simple Example

A	B	$\theta_1(A,B)$
a	b	3
a	$\bar{b}$	0
$\bar{a}$	b	-1
$\bar{a}$	$\bar{b}$	1

B	C	$\theta_2(B,C)$
b	c	4
b	$\bar{c}$	1.5
$\bar{b}$	c	0.2
$\bar{b}$	$\bar{c}$	2

B	$M_{1,2}(B)$
b	3
$\bar{b}$	1



B	$M_{2,1}(B)$
b	4
$\bar{b}$	2

A	B	$\beta_1(A,B)$
a	b	7
a	$\bar{b}$	2
$\bar{a}$	b	3
$\bar{a}$	$\bar{b}$	3

B	C	$\beta_2(B,C)$
b	c	7
b	$\bar{c}$	4.5
$\bar{b}$	c	1.2
$\bar{b}$	$\bar{c}$	3

Conclusion

- MPE and MAP queries can be answered with simple adaptations of the VE and FE algorithms
  - We introduced an elimination operation with max
- In MPE, we operate only with multiplication and maximization operations
  - We can easily replace probabilities with log-probabilities
  - This creates numerically stable algorithms, since multiplying several small numbers can lead to underflows
- We also studied algorithms based on systematic search
  - Although the worst-case analysis of these algorithms tend to show a large time complexity
  - They can perform better than VE on the average case due to pruning
- We discussed two techniques for creating upper-bounds
  - Node splitting is used with MPE
  - Jointrees are used to create low width elimination orders similar to constrained ones for MAP
- Task
  - Read chapter 10 of the textbook