Factor Elimination

Brandon Malone

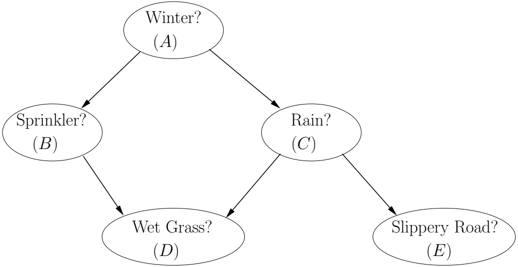
Much of this material is adapted from Chapters 6 and 7 in Darwiche's book

Many of the images were taken from the Internet

January 30, 2014

Inference in Bayesian networks

Suppose we have a general Bayesian network.



We would like to answer multiple probabilistic queries using this network. How can we efficiently answer these queries?

Factors

A **factor** f is a function which assigns a non-negative number to each instantiation of a set of variables.

B	C	D	$f_1(B,C,D)$ ($\Theta_{D\|B,C}$)
T	T	T	.95
T	T	F	.05
T	F	T	.9
T	F	F	.1
F	T	T	.8
F	T	F	.2
F	F	T	0
F	F	F	1

D	E	$f_2(D,E)$ ($\Theta_{D,E}$)
T	T	.448
T	F	.192
F	T	.112
F	F	.248

$vars(f)$ denotes the variables over which f is defined.
$f(X_1,\ldots,X_n)$ will indicate that f is defined over $X_1,\ldots,X_n$.

Summing out (a.k.a., marginalizing or projecting)

We **sum out** X from $f(\mathbf{Y} \cup X)$ by adding together rows which have common values for all other variables.

$$\left(\sum_X f\right)(\mathbf{y}) := \sum_x f(x, \mathbf{y})$$

D	E	$f_2(D,E)$ $(\Theta_{D,E})$
T	T	.448
T	F	.192
F	T	.112
F	F	.248

$\Rightarrow$

E	$\sum_D f(E)$ (Θ_E)
T	.56
F	.44

Summation is commutative.

$$\sum_Y \sum_X f = \sum_X \sum_Y f$$

So we will use the notation $\sum_{\mathbf{X}} f$ to mean "sum out the set of variables $\mathbf{X}$ from f".

Summing out (a.k.a., marginalizing or projecting)

We **sum out** X from $f(\mathbf{Y} \cup X)$ by adding together rows which have common values for all other variables.

$$\left(\sum_X f\right)(\mathbf{y}) := \sum_x f(x, \mathbf{y})$$

D	E	$f_2(D,E)$ $(\Theta_{D,E})$
T	T	.448
T	F	.192
F	T	.112
F	F	.248

$\Rightarrow$

E	$\sum_D f(E)$ (Θ_E)
T	.56
F	.44

We will sometimes **project** f onto a subset of its variables $\mathbf{Q}$.

$$\text{project}(f, \mathbf{Q}) := \sum_{\text{vars}(f) \setminus \mathbf{Q}} f$$

That is, we sum out all variables in f except for those in $\mathbf{Q}$.

Multiplication

We multiply two factors f_1 and f_2 by multiplying the rows which assign the same values to variables in common.

$$f_1(\mathbf{X}) \cdot f_2(\mathbf{Y}) := (f_1 f_2)(\mathbf{X} \cup \mathbf{Y})$$
$$(f_1 f_2)(\mathbf{z}) := f_1(\mathbf{x}) f_2(\mathbf{y}) \qquad \text{where } \mathbf{x} \sim \mathbf{z} \text{ and } \mathbf{y} \sim \mathbf{z}$$

B	C	D	$f_1(B, C, D)$
T	T	T	.95
T	T	F	.05
T	F	T	.9
T	F	F	.1
F	T	T	.8
F	T	F	.2
F	F	T	0
F	F	F	1

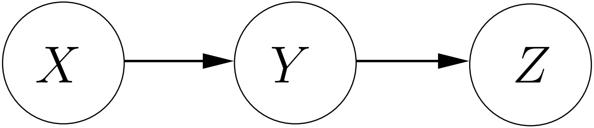
D	E	$f_2(D, E)$
T	T	.448
T	F	.192
F	T	.112
F	F	.248

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

Multiplication is commutative and associative.

Bayesian networks as factors

A Bayesian network is simply a set of factors ("factored representation of a probability distribution").



X	$f_X = \Theta_X$
T	.6
F	.4

X	Y	$f_Y = \Theta_{Y\|X}$
T	T	.9
T	F	.1
F	T	.2
F	F	.8

Y	Z	$f_Z = \Theta_{Z\|Y}$
T	T	.3
T	F	.7
F	T	.5
F	F	.5

Factor elimination

procedure FactorElimination1(Bayesian network $\mathcal{N}$, query variable Q)
  $\mathcal{S} \leftarrow$ CPTs of $\mathcal{N}$
  $f_r \leftarrow$ some factor in $\mathcal{S}$ which contains Q
  while $\mathcal{S}$ has more than one factor **do**
    remove a factor $f_i \neq f_r$ from $\mathcal{S}$
    $\mathbf{V} \leftarrow$ variables in f_i but not in $\mathcal{S}$
    $f_j \leftarrow f_j \sum_{\mathbf{V}} f_i$ for some factor f_j in $\mathcal{S}$
  end while
  return the prior marginal of Q, project(f_r, Q)
end procedure

Incorporating evidence into factors

Given some evidence **e**, we can create a **reducted factor** which incorporates **e**.

$$f^{\mathbf{e}}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \sim \mathbf{e} \\ 0 & \text{otherwise} \end{cases}$$

D	E	$f_2(D,E)$
T	T	.448
T	F	.192
F	T	.112
F	F	.248

Given evidence **e**: $E = \mathrm{T}$

D	E	$f_2^{\mathbf{e}}(D,E)$
T	T	.448
T	F	0
F	T	.112
F	F	0

We can now compute joint marginals, $P(Q, \mathbf{e})$ by using $f^{\mathbf{e}}$ instead of f.
What about computing the probability of evidence $P(e)$?

Incorporating evidence into factors

Given some evidence **e**, we can create a **reducted factor** which incorporates **e**.

$$f^{\mathbf{e}}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \sim \mathbf{e} \\ 0 & \text{otherwise} \end{cases}$$

D	E	$f_2(D,E)$
T	T	.448
T	F	.192
F	T	.112
F	F	.248

Given evidence **e**: $E = \mathrm{T}$

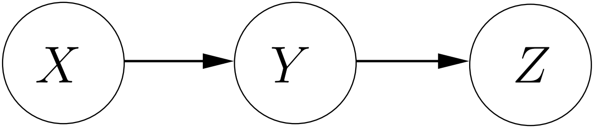
D	E	$f_2^{\mathbf{e}}(D,E)$
T	T	.448
T	F	0
F	T	.112
F	F	0

We can now compute joint marginals, $P(Q, \mathbf{e})$ by using $f^{\mathbf{e}}$ instead of f.
What about computing the probability of evidence $P(e)$?

Reduce the factors and then eliminate everything.

Factor elimination - Simple class example

Use this Bayesian network to compute the prior marginal of Z.



X	$f_X = \Theta_X$
T	.6
F	.4

X	Y	$f_Y = \Theta_{Y\|X}$
T	T	.9
T	F	.1
F	T	.2
F	F	.8

Y	Z	$f_Z = \Theta_{Z\|Y}$
T	T	.3
T	F	.7
F	T	.5
F	F	.5

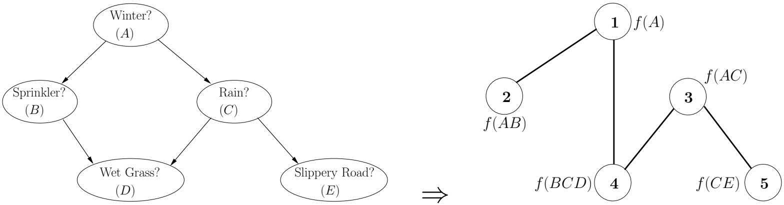
- Compute the prior marginal, $P(Z)$.
- Compute the prior marginal, $P(Y, Z)$.
- Compute the joint marginal, $P(X, Z = \mathrm{T})$.

Elimination trees

In what order should we eliminate factors?

Elimination trees

In what order should we eliminate factors?



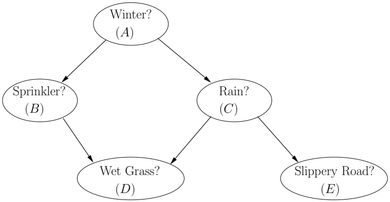
We can use an **elimination tree**, $\mathcal{T}$.

1. Each factor is assigned to exactly one node.
2. The factors for node i are multiplied to give ϕ_i.

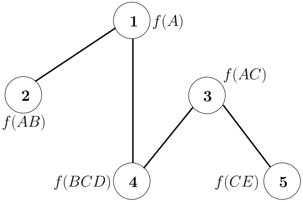
vars(i) denotes the variables appearing in factor ϕ_i.

Elimination trees

In what order should we eliminate factors?



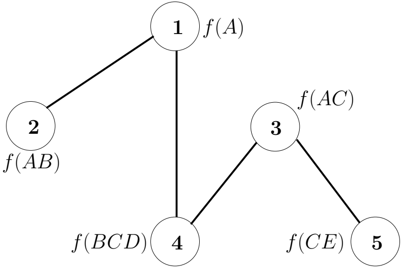
$\Rightarrow$



$r \leftarrow$ some node in $\mathcal{T}$ which contains Q
while $\mathcal{T}$ has more than one node **do**
  remove a node $i \neq r$ which has a single neighbor j from $\mathcal{T}$
  $\mathbf{V} \leftarrow$ variables in ϕ_i but not in $\mathcal{T}$
  $\phi_j \leftarrow \phi_j \sum_{\mathbf{V}} \phi_i$
end while
project(ϕ_r, Q)

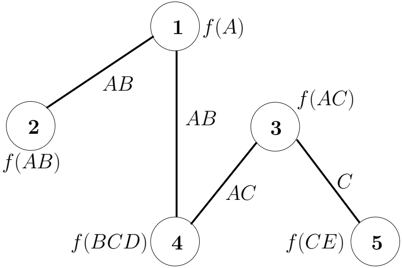
Separators

The **separator** of edge $i - j$, $\mathbf{S}_{ij}$, is the intersection of variables that occur on the i side of the edge and on the j side of the edge.



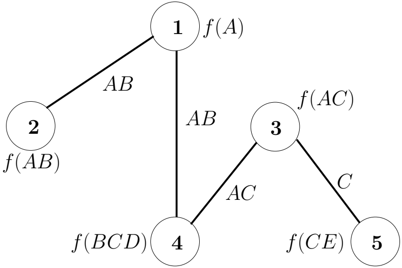
Separators

The **separator** of edge $i - j$, $\mathbf{S}_{ij}$, is the intersection of variables that occur on the i side of the edge and on the j side of the edge.



Separators

The **separator** of edge $i - j$, $\mathbf{S}_{ij}$, is the intersection of variables that occur on the i side of the edge and on the j side of the edge.



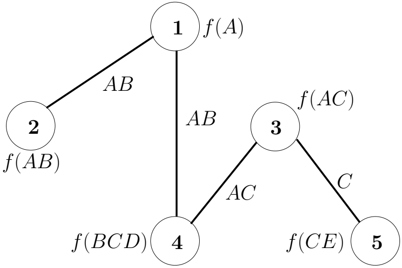
$$\sum_{\mathbf{V}} \phi_i = \text{project}(\phi_i, \mathbf{S}_{ij})$$

What is the space complexity of the resulting factor?

Clusters

The **cluster** of a node , $\mathbf{C}_i$ is *vars*(i) combined with the separators of its edges.

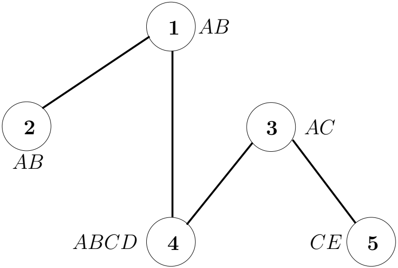
$$\mathbf{C}_i := vars(i) \cup \bigcup_j \mathbf{S}_{ij}$$



Clusters

The **cluster** of a node , $\mathbf{C}_i$ is *vars*(i) combined with the separators of its edges.

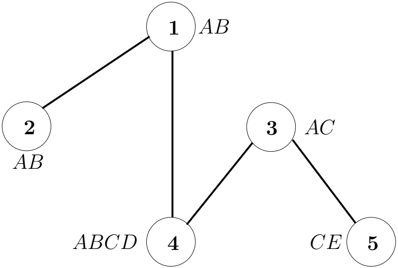
$$\mathbf{C}_i := vars(i) \cup \bigcup_j \mathbf{S}_{ij}$$



Clusters

The **cluster** of a node , $\mathbf{C}_i$ is *vars*(i) combined with the separators of its edges.

$$\mathbf{C}_i := vars(i) \cup \bigcup_j \mathbf{S}_{ij}$$

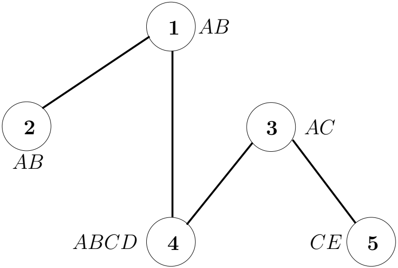


When ϕ_i is selected, it will be a factor over exactly $\mathbf{C}_i$.
What is the space complexity of this factor?

Clusters

The **cluster** of a node , $\mathbf{C}_i$ is *vars*(i) combined with the separators of its edges.

$$\mathbf{C}_i := vars(i) \cup \bigcup_j \mathbf{S}_{ij}$$

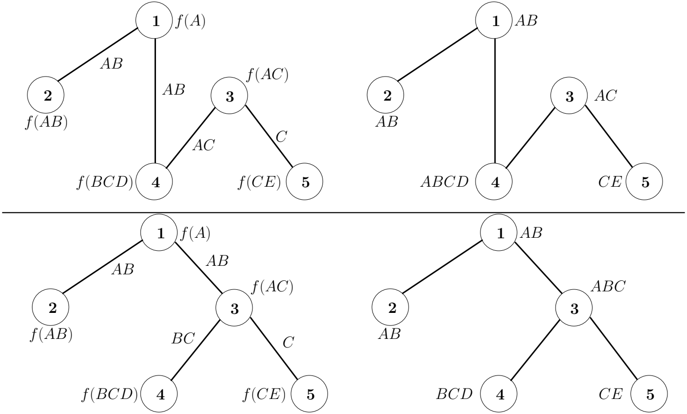


When ϕ_i is selected, it will be a factor over exactly $\mathbf{C}_i$.

What is the space complexity of this factor?

The **width** of a tree is the size of its largest cluster minus one.

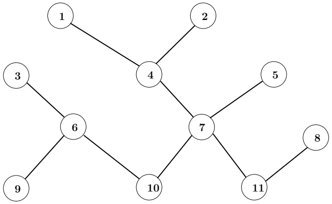
Are all elimination trees equally efficient?



Inference via message passing

For HMMs and NBCs, we could answer queries without destroying the structure.

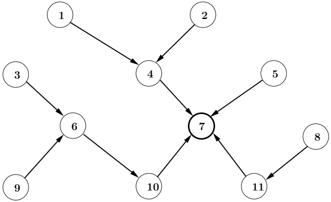
Can we do the same thing for elimination trees? (And why might we want to?)



Inference via message passing

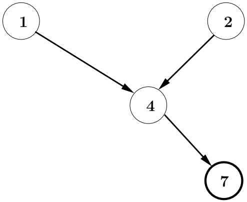
For HMMs and NBCs, we could answer queries without destroying the structure.

Can we do the same thing for elimination trees? (And why might we want to?)



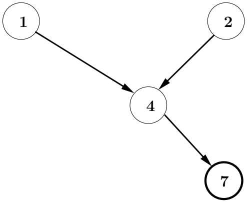
Suppose we choose **7** as the root. What choices do we really have in nodes we select to remove?

Messages between neighbors



When can we eliminate **4**?

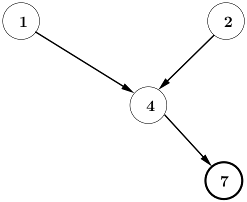
Messages between neighbors



When can we eliminate **4**?

After we eliminate **1** and **2**.

Messages between neighbors

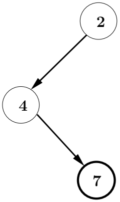


When can we eliminate **4**?

After we eliminate **1** and **2**.

What do we do to eliminate **1**?

Messages between neighbors



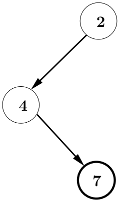
When can we eliminate **4**?

After we eliminate **1** and **2**.

What do we do to eliminate **1**

1. Calculate $\sum_{\mathbf{V}} \phi_1 = \text{project}(\phi_1, \mathcal{S}_{1,4}) := M_{1,4}$
2. Update $\phi_4 \leftarrow \phi_4 M_{1,4}$

Messages between neighbors

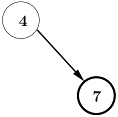


When can we eliminate **4**?

After we eliminate **1** and **2**.

What do we do to eliminate **2**?

Messages between neighbors



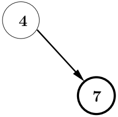
When can we eliminate **4**?

After we eliminate **1** and **2**.

What do we do to eliminate **2**?

1. Update $\phi_4 \leftarrow \phi_4 M_{2,4}$
2. So ϕ_4 is now $\phi_4 M_{1,4} M_{2,4}$

Messages between neighbors



When can we eliminate **4**?

After we eliminate **1** and **2**.

What is $M_{4,7}$?

Messages between neighbors

When can we eliminate **4**?

After we eliminate **1** and **2**.

What is $M_{4,7}$? project$(\phi_4 M_{1,4} M_{2,4}, \mathcal{S}_{4,7})$

Messages between neighbors

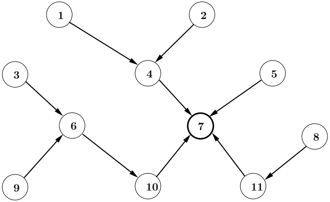
When can we eliminate **4**?

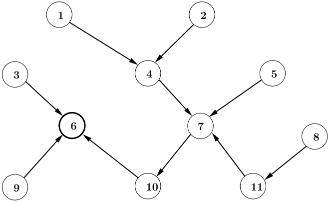
After we eliminate **1** and **2**.

In general, $M_{i,j} := \text{project}\left(\phi_i \prod_{k \neq j} M_{k,i}, \mathcal{S}_{i,j}\right)$.

Reusing messages

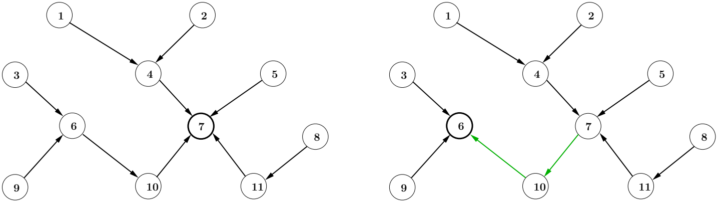
Suppose we now choose **6** as the root.





Reusing messages

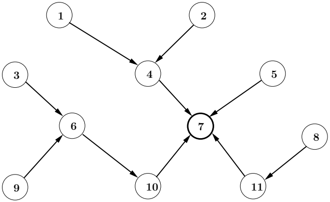
Suppose we now choose **6** as the root.



Only two of the messages actually change.

Reusing messages

Suppose we now choose **6** as the root.



In general, we can compute all messages w.r.t. a single root by

1. **pulling** messages towards the root
2. **pushing** messages away from the root

The prior marginal for $\mathbf{C}_i$ is $\phi_i \prod_k M_{k,i}$ (the product of incoming messages).

Handling evidence

Suppose we are given some evidence **e**, and we are interested in the joint marginals $P(\mathbf{C}_i, \mathbf{e})^*$. How can we handle the evidence?

* Of course, we can then use the definitions to compute conditional values.

Handling evidence

Suppose we are given some evidence $\mathbf{e}$, and we are interested in the joint marginals $P(\mathbf{C}_i, \mathbf{e})^*$. How can we handle the evidence?

We can add an indicator factor, λ_E for each $E \in \mathbf{e}$.

$$\lambda_E(e) = \begin{cases} 1 & \text{if } e \sim \mathbf{e} \\ 0 & \text{otherwise} \end{cases}$$

We then add λ_E to node i such that $E \in \mathbf{C}_i$.

We can reuse those messages until receiving additional evidence which invalidates λ_E.

* Of course, we can then use the definitions to compute conditional values.

Factor elimination algorithm

procedure FactorElimination(elimination tree $\mathcal{T}$, evidence $\mathbf{e}$)
    for each variable $E \in \mathbf{e}$ **do**
        $i \leftarrow$ node in $\mathcal{T}$ such that $E \in \mathbf{C}_i$
        $\phi_i \leftarrow \phi_i \lambda_E$ ▷ adding the evidence to node i
    end for
    Choose a root node r in $\mathcal{T}$
    Pull messages towards r
    Push messages away from r
    return $\phi_i \prod_k M_{ki}$ for each $i \in \mathcal{T}$ ▷ joint marginal $P(\mathbf{C}_i, \mathbf{e})$
end procedure

Recap

During this section, we discussed

- Factors and their operations
- Bayesian network inference as factor elimination
- Message passing to efficiently answer queries on elimination trees

Next in probabilistic models

The jointree algorithm, which gives a procedure for creating efficient elimination trees