Technische Universiteit **Eindhoven** University of Technology

The Junction Tree Algorithm

A Visual Guide

Martin Roa Villescas

December 12, 2021

Eindhoven University of Technology

Table of Contents

Problem Statement

What do these applications have in common?

INSURANCE

Uncover fraud, identify attempts at money laundering and assess a range of risks that can impact costs and competitive position with predictive analytics. Using predictive analytics insurers can build and deploy intelligent detection solutions that provide a proactive heads-up to risk and abuse, and protect profits for a better bottom line.

BayesFraud

BANKING

Become more proactive in addressing risk with predictive analytics. Utilize your data to the fullest to create automated solutions for default prediction, risk compliance, operational risk calculation, anti-money laundering and more.

Nykredit

FORENSICS

Make sense of complex data and resolve questions of identity in a range of forensic contexts. Predictive analytics enable real-time probability calculations of complex evidence, including mixed traces of DNA from multiple sources.

Forensic Identification

TELECOM

Identify risk scenarios that have a significant impact on telecoms. Real-time assessment solutions based on analytics reduce the risk of non-payment and promote stable customer relationships to keep telecom businesses competitive.

The Cure

MANUFACTURING

Troubleshoot breakdowns before they happen and keep crucial systems up and running. Predictive analytics help manufacturers automate component replacement and maintenance to protect businesses from costly operational downtime.

ABB

MEDICAL

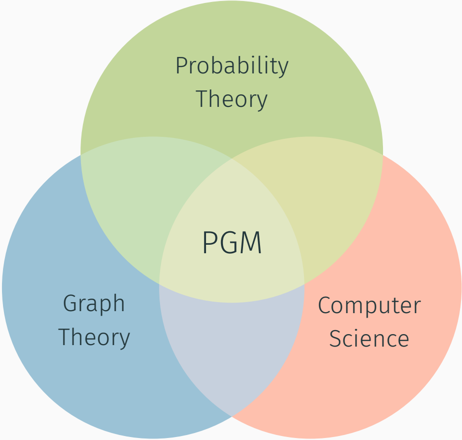
An automated decision support solutions can assist physicians in making potentially life-saving diagnoses and managing healthcare treatment. Using predictive analytics medical staff can diagnose illness and select the correct treatment with great speed and accuracy, improving care and cost management.

Treat Steward

Complexity and Uncertainty

Probabilistic Graphical Models

Probabilistic Graphical Models



Examples of PGMs

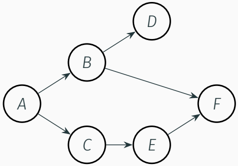


Figure 1: A Bayesian network

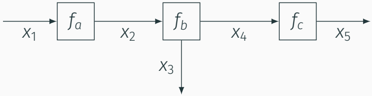


Figure 2: A factor graph

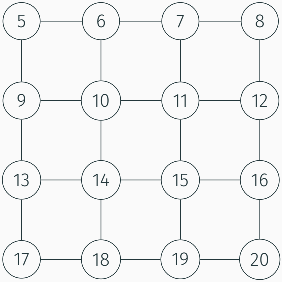


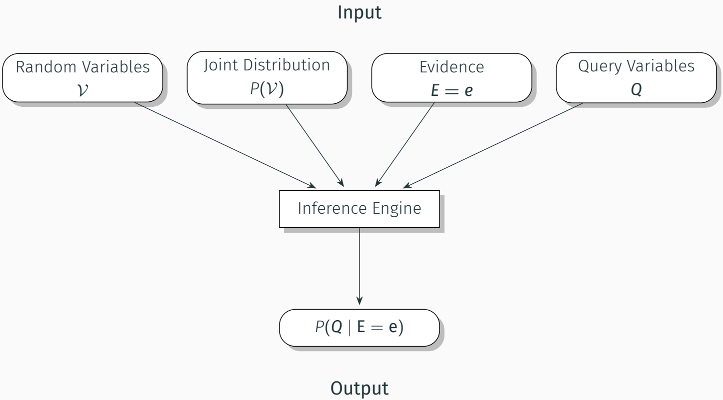
Figure 3: A Markov random field

Bayesian Inference

The Inference Problem

Given a set of ***random variables*** $\mathcal{V}$ *and their* ***joint distribution*** $P(\mathcal{V})$*, compute one or more conditional distributions given observations.*

The Inference Problem



The Junction Tree Algorithm

The Junction Tree Algorithm

*The **junction tree algorithm** is an efficient method to perform Bayesian inference in general graphs.*

The Junction Tree Algorithm in Practice

HUGINEXPERT

INSURANCE

Uncover fraud, identify attempts at money laundering and assess a range of risks that can impact costs and competitive position with predictive analytics. Using predictive analytics insurers can build and deploy intelligent detection solutions that provide a proactive heads-up to risk and abuse, and protect profits for a better bottom line.

BayesFraud

BANKING

Become more proactive in addressing risk with predictive analytics. Utilize your data to the fullest to create automated solutions for default prediction, risk compliance, operational risk calculation, anti-money laundering and more.

Nykredit

FORENSICS

Make sense of complex data and resolve questions of identity in a range of forensic contexts. Predictive analytics enable real-time probability calculations of complex evidence, including mixed traces of DNA from multiple sources.

Forensic Identification

TELECOM

Identify risk scenarios that have a significant impact on telecoms. Real-time assessment solutions based on analytics reduce the risk of non-payment and promote stable customer relationships to keep telecom businesses competitive.

The Cure

MANUFACTURING

Troubleshoot breakdowns before they happen and keep crucial systems up and running. Predictive analytics help manufacturers automate component replacement and maintenance to protect businesses from costly operational downtime.

ABB

MEDICAL

An automated decision support solutions can assist physicians in making potentially life-saving diagnoses and managing healthcare treatment. Using predictive analytics medical staff can diagnose illness and select the correct treatment with great speed and accuracy, improving care and cost management.

Treat Steward

The Junction Tree Algorithm in Practice

HUGINEXPERT

Steffen Lauritzen
Chairman

INSURANCE

Uncover fraud, identify attempts at money laundering and assess a range of risks that can impact costs and competitive position with predictive analytics. Using predictive analytics insurers can build and deploy intelligent detection solutions that provide a proactive heads-up to risk and abuse, and protect profits for a better bottom line.

BayesFraud

BANKING

Become more proactive in addressing risk with predictive analytics. Utilize your data to the fullest to create automated solutions for default prediction, risk compliance, operational risk calculation, anti-money laundering and more.

Nykredit

FORENSICS

Make sense of complex data and resolve questions of identity in a range of forensic contexts. Predictive analytics enable real-time probability calculations of complex evidence, including mixed traces of DNA from multiple sources.

Forensic Identification

TELECOM

Identify risk scenarios that have a significant impact on telecoms. Real-time assessment solutions based on analytics reduce the risk of non-payment and promote stable customer relationships to keep telecom businesses competitive.

The Cure

MANUFACTURING

Troubleshoot breakdowns before they happen and keep crucial systems up and running. Predictive analytics help manufacturers automate component replacement and maintenance to protect businesses from costly operational downtime.

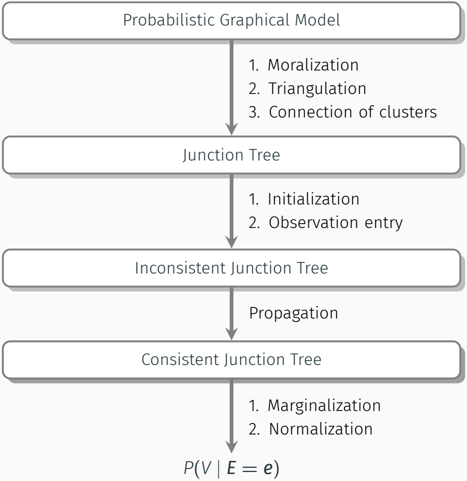
ABB

MEDICAL

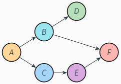
An automated decision support solutions can assist physicians in making potentially life-saving diagnoses and managing healthcare treatment. Using predictive analytics medical staff can diagnose illness and select the correct treatment with great speed and accuracy, improving care and cost management.

Treat Steward

Overview of the Junction Tree Algorithm



Overview of the Junction Tree Algorithm



Probabilistic Graphical Model

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

1. Initialization
2. Observation entry

Inconsistent Junction Tree

Propagation

Consistent Junction Tree

1. Marginalization
2. Normalization

$P(V \mid \boldsymbol{E} = \boldsymbol{e})$

Overview of the Junction Tree Algorithm

Probabilistic Graphical Model

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

1. Initialization
2. Observation entry

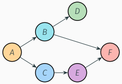
Inconsistent Junction Tree

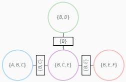
Propagation

Consistent Junction Tree

1. Marginalization
2. Normalization

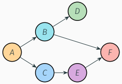
$P(V \mid E = e)$





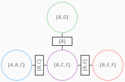
Overview of the Junction Tree Algorithm

Probabilistic Graphical Model



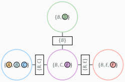
1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree



1. Initialization
2. Observation entry

Inconsistent Junction Tree



Propagation

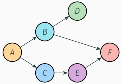
Consistent Junction Tree

1. Marginalization
2. Normalization

$$P(V \mid E = e)$$

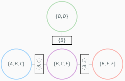
Overview of the Junction Tree Algorithm

Probabilistic Graphical Model



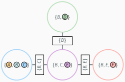
1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree



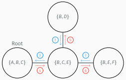
1. Initialization
2. Observation entry

Inconsistent Junction Tree



Propagation

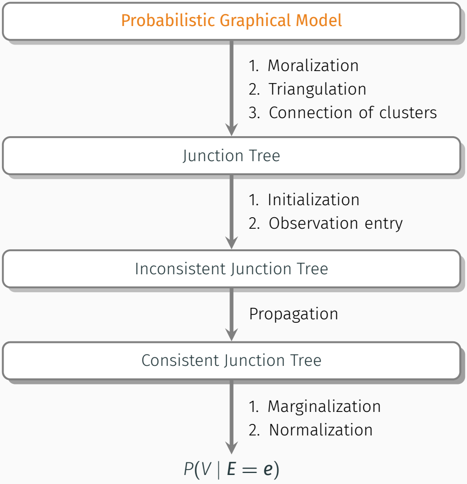
Consistent Junction Tree



1. Marginalization
2. Normalization

$P(V \mid E = e)$

Overview


Probabilistic Graphical Model
1. Moralization
2. Triangulation
3. Connection of clusters
Junction Tree
1. Initialization
2. Observation entry
Inconsistent Junction Tree
Propagation
Consistent Junction Tree
1. Marginalization
2. Normalization
$P(V \mid E = e)$

Probabilistic Graphical Model

Joint probability distribution

$$P(\mathcal{V}) = \prod_{V \in \mathcal{V}} P(V \mid pa(V))$$

Conditional probability distribution

$$P(B \mid A) = \begin{array}{c|cc} & \multicolumn{2}{c}{P(b \mid a)} \\ a & \text{yes} & \text{no} \\ \hline \text{yes} & 0.1 & 0.5 \\ \text{no} & 0.4 & 0.3 \end{array}$$

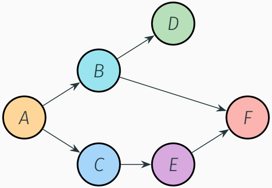


Figure 4: A Bayesian network[1]

[1] Example borrowed from Mark A. Paskin - A Short Course on Graphical Models

Starting Point

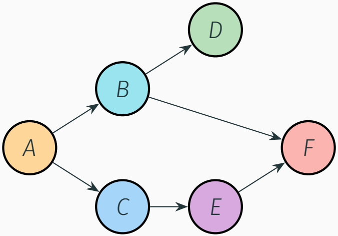


Figure 5: A Bayesian network

Overview

Probabilistic Graphical Model

↓

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

↓

1. Initialization
2. Observation entry

Inconsistent Junction Tree

↓

Propagation

Consistent Junction Tree

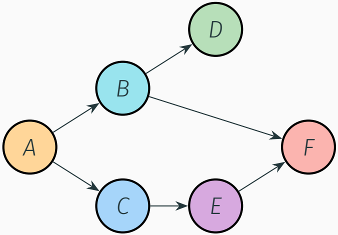
↓

1. Marginalization
2. Normalization

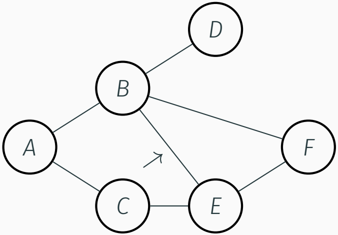
$P(V \mid E = e)$

Moralization

Marry the parents of each variable and drop the directions of the edges.



(a) Bayesian network



(b) Moral graph

Figure 6: Moralization

Moralization

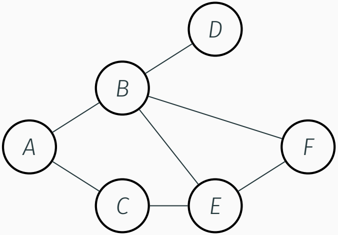
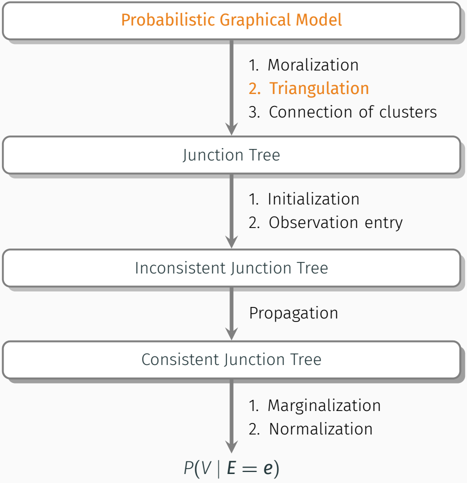


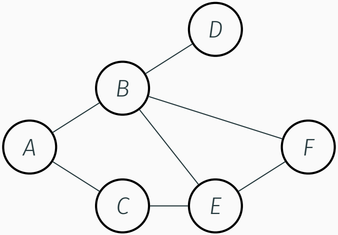
Figure 7: Moral Graph

Overview

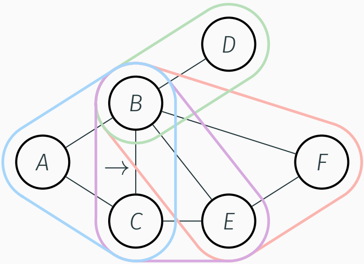

Probabilistic Graphical Model
1. Moralization
2. Triangulation
3. Connection of clusters
Junction Tree
1. Initialization
2. Observation entry
Inconsistent Junction Tree
Propagation
Consistent Junction Tree
1. Marginalization
2. Normalization
$P(V \mid E = e)$

Triangulation

Consists of removing every cycle of length greater than three in a graph.



(a) Moral graph

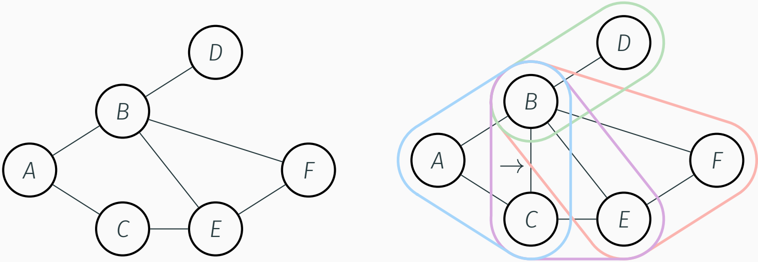


(b) Triangulated graph

Figure 8: Triangulation

Triangulation

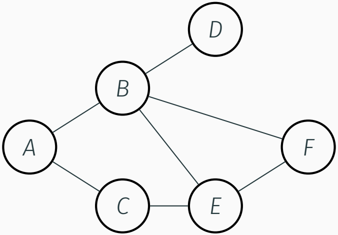
We do so by connecting two nonadjacent nodes in every cycle of length > three.



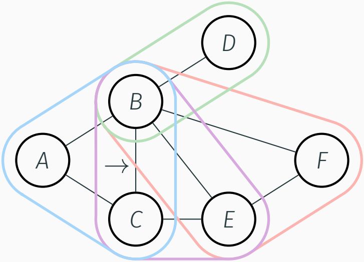
(a) Moral graph **(b)** Triangulated graph

Figure 8: Triangulation

An optimal triangulation minimizes the sum of the state space sizes of the cliques.



(a) Moral graph



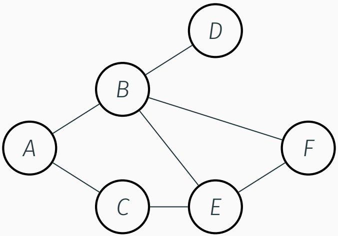
(b) Triangulated graph

Figure 8: Triangulation

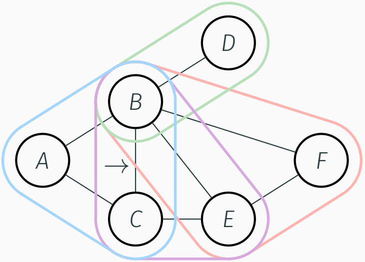
Triangulation

An optimal triangulation minimizes the sum of the state space sizes of the cliques.

This is equivalent to minimizing the size of the largest clique.



(a) Moral graph

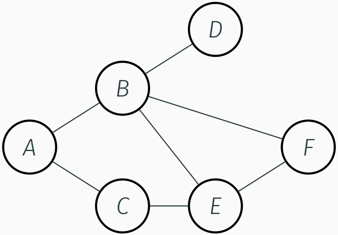


(b) Triangulated graph

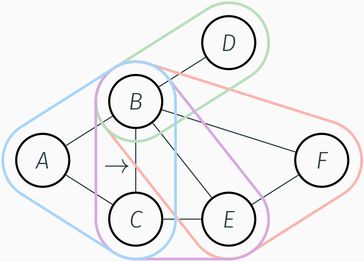
Figure 8: Triangulation

Triangulation

This problem is *NP-complete*.



(a) Moral graph

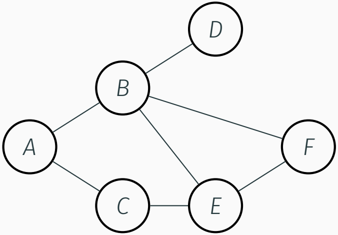


(b) Triangulated graph

Figure 8: Triangulation

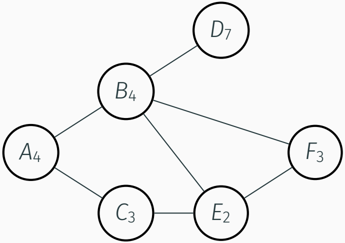
Triangulation: Min-fill Algorithm

We will now demonstrate the **min-fill algorithm** [3]: A greedy, polynomial-time *heuristic* that produces high-quality triangulations in real-world settings.



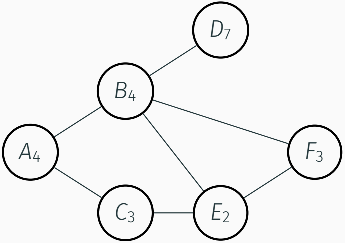
Triangulation: Min-fill Algorithm

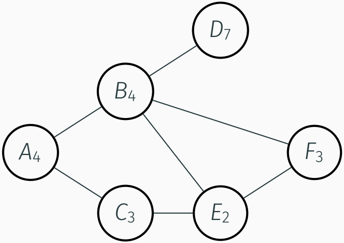
Subscripts denote the variable's *cardinality*.



Triangulation: Min-fill Algorithm

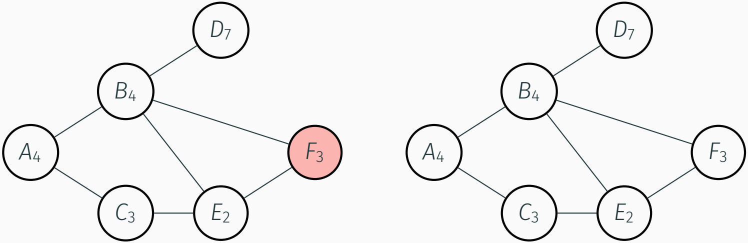
Make a copy of the graph.





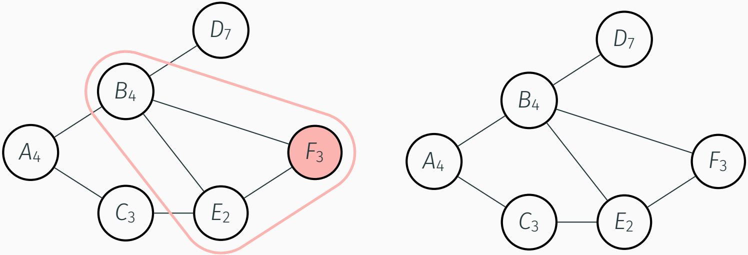
Triangulation: Min-fill Algorithm

Select a node in the left graph according to the criterion described below.

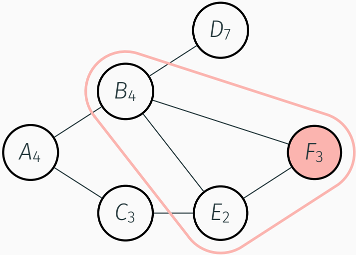


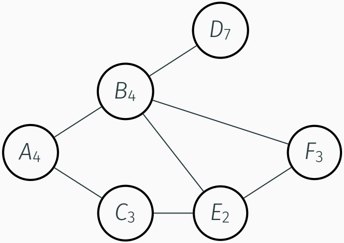
The selected variable and its neighbors form a *cluster*.

Connect all the nodes in the cluster (in this case they are already connected).



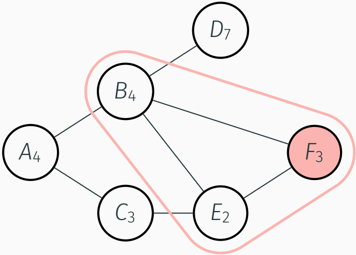
Copy the cluster with the added edges to the right graph
only if it is not contained inside an already copied cluster.

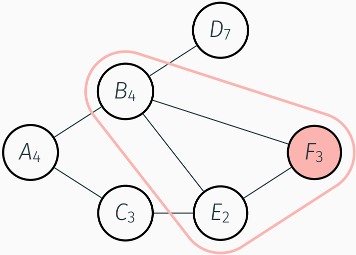




Triangulation: Min-fill Algorithm

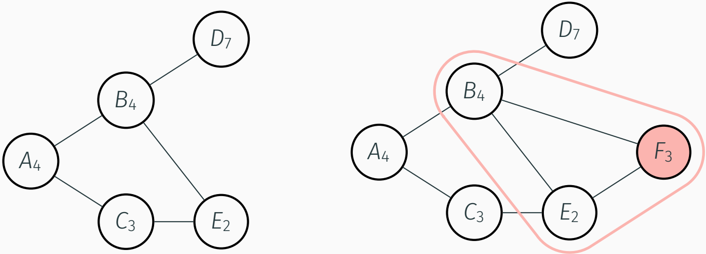
Copy the cluster with the added edges to the right graph *only if* it is not contained inside an already copied cluster.





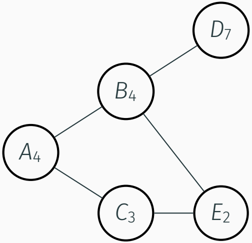
Triangulation: Min-fill Algorithm

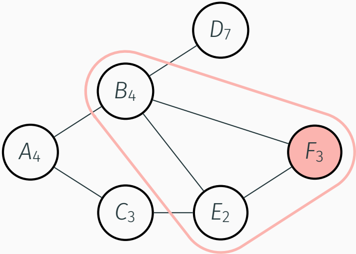
Remove F from the left graph.



Triangulation: Min-fill Algorithm

Repeat until there are no nodes left in the left graph.

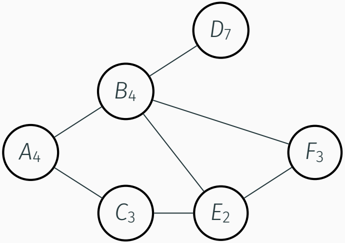


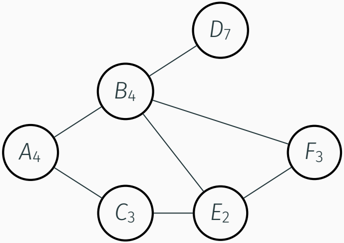


Triangulation: Min-fill Algorithm

Node selection criterion:

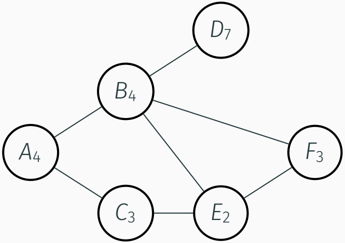
- Select the node that causes the *least* number of edges to be added in the cluster.
- Break ties by choosing the node that induces the cluster with the *smallest* state space size.

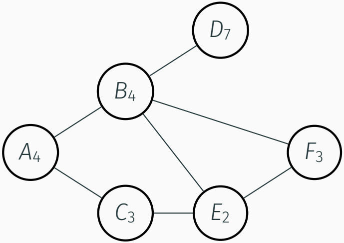




Triangulation: Min-fill Algorithm

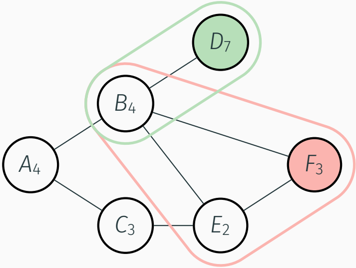
Selecting which node causes the least # of edges to be added in the induced cluster?

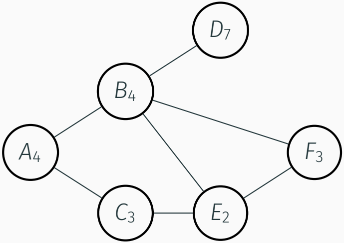




Triangulation: Min-fill Algorithm

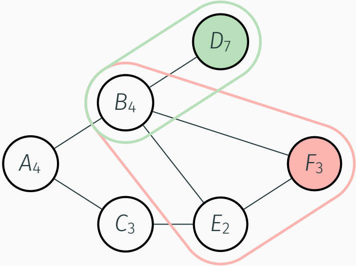
Tie.

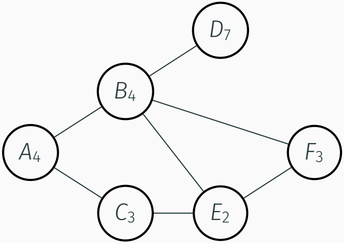




Triangulation: Min-fill Algorithm

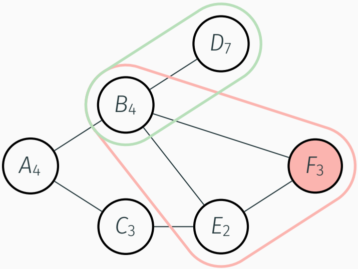
Then which of the two clusters has a smaller state space size?

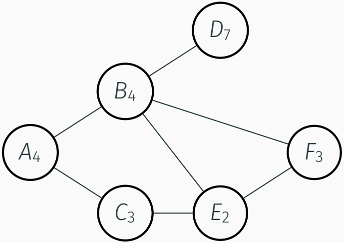




Triangulation: Min-fill Algorithm

$|BEF| < |BD|$, therefore F wins.

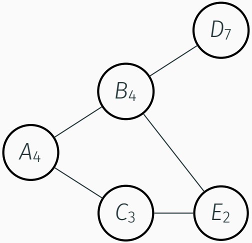


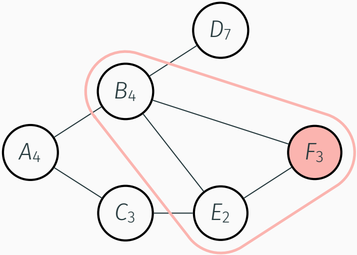


Triangulation: Min-fill Algorithm

Back to where we were.

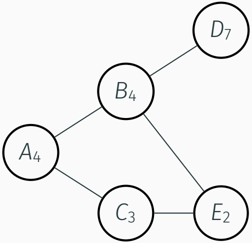
Repeat until there are no nodes left in the left graph.

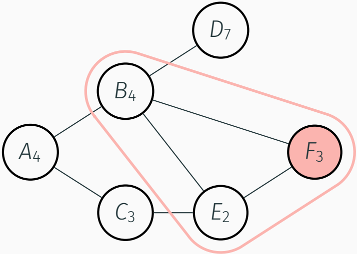




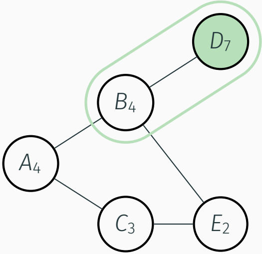
Triangulation: Min-fill Algorithm

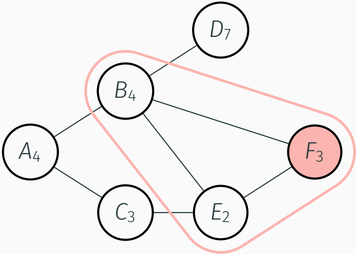
Selecting which node causes the least # of edges to be added in the induced cluster?



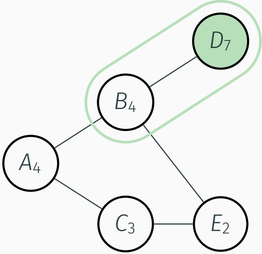


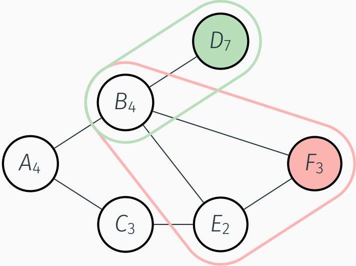
Triangulation: Min-fill Algorithm



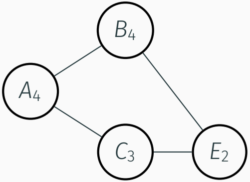


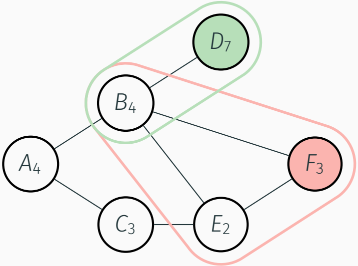
Triangulation: Min-fill Algorithm



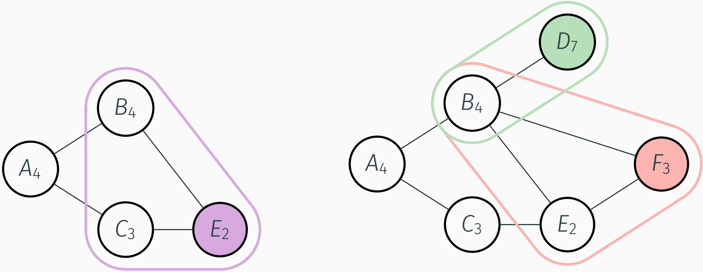


Triangulation: Min-fill Algorithm

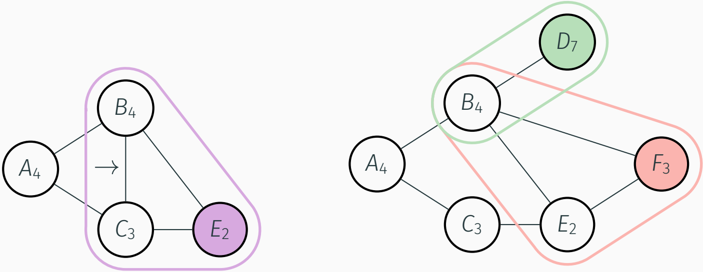




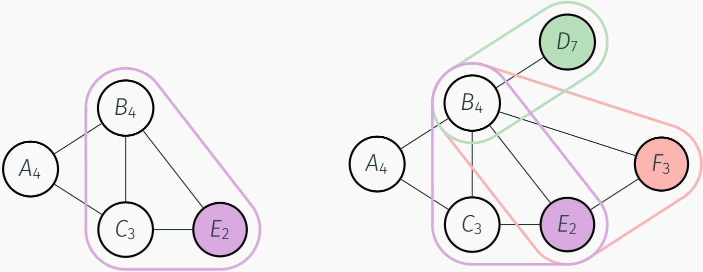
Triangulation: Min-fill Algorithm



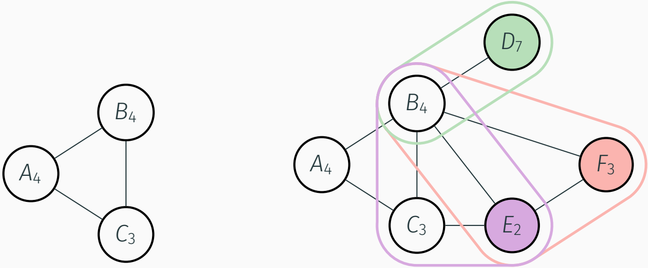
Triangulation: Min-fill Algorithm



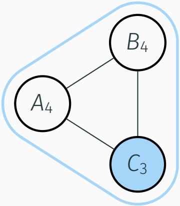
Triangulation: Min-fill Algorithm

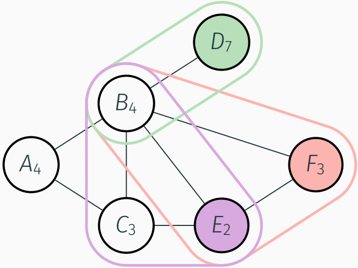


Triangulation: Min-fill Algorithm

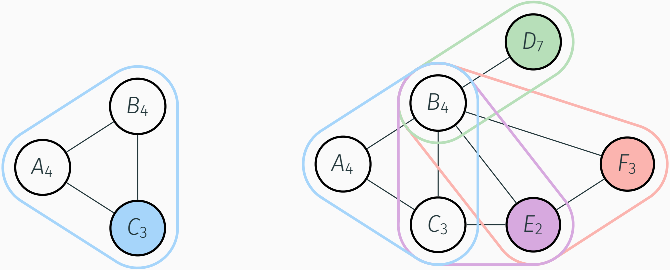


Triangulation: Min-fill Algorithm

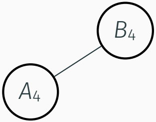


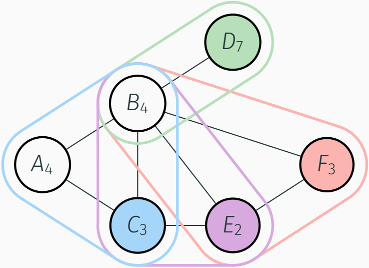


Triangulation: Min-fill Algorithm

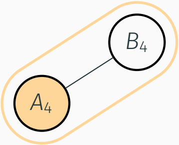


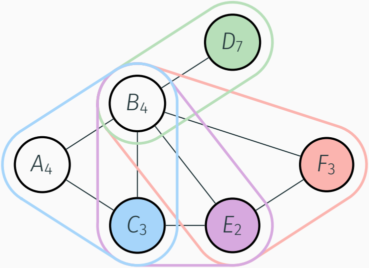
Triangulation: Min-fill Algorithm



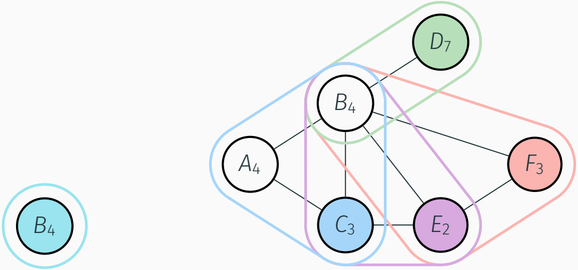


Triangulation: Min-fill Algorithm

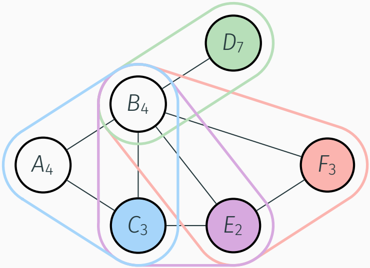




Triangulation: Min-fill Algorithm



Triangulation: Min-fill Algorithm



Triangulated Graph

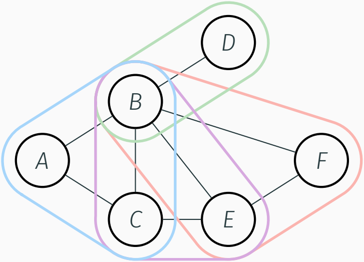


Figure 9: Triangulated graph with its set of maximal cliques

Overview

Probabilistic Graphical Model

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

1. Initialization
2. Observation entry

Inconsistent Junction Tree

Propagation

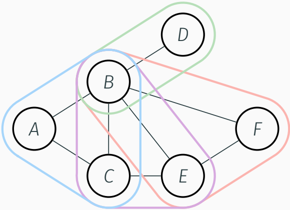
Consistent Junction Tree

1. Marginalization
2. Normalization

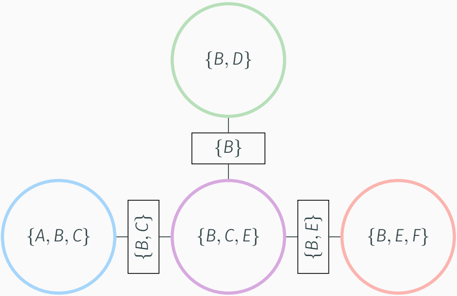
$P(V \mid \mathbf{E} = \mathbf{e})$

Connection of Clusters

Consists of transforming the triangulated graph into a *junction tree*.



(a) Triangulated graph



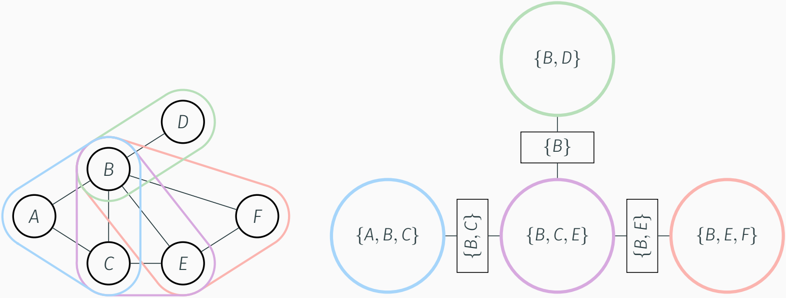
(b) Junction tree

Figure 10: Junction tree construction from a triangulated graph

Connection of Clusters

A junction tree is a tree that satisfies the *running intersection property*:

All clusters on the path between two given clusters contain their common variables.



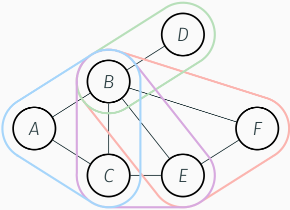
(a) Triangulated graph

(b) Junction tree

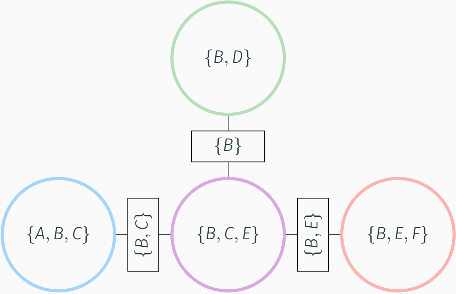
Figure 10: Junction tree construction from a triangulated graph

Connection of Clusters

We now present an *optimal* algorithm to perform this transformation [2].



(a) Triangulated graph

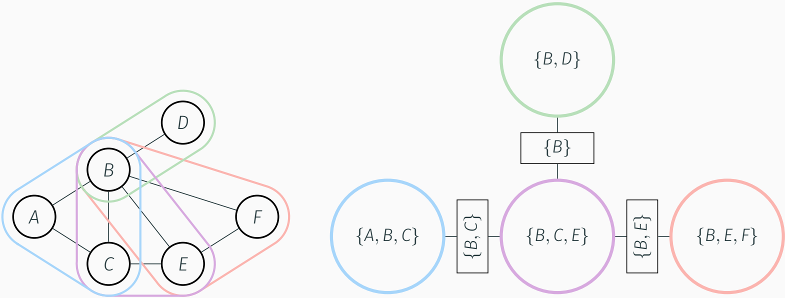


(b) Junction tree

Figure 10: Junction tree construction from a triangulated graph

Connection of Clusters

The clusters of the triangulated graphs correspond to the nodes of the junction tree.

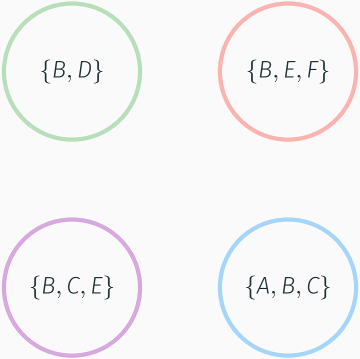


(a) Triangulated graph (b) Junction tree

Figure 10: Junction tree construction from a triangulated graph

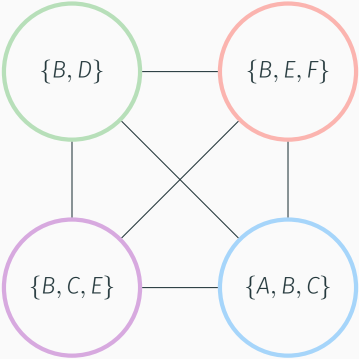
Connection of Clusters

The clusters of the triangulated graphs correspond to the nodes of the junction tree.



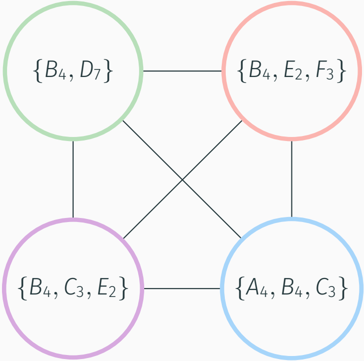
Connection of Clusters

Form a *complete graph*, i.e. connect each node with every other node.



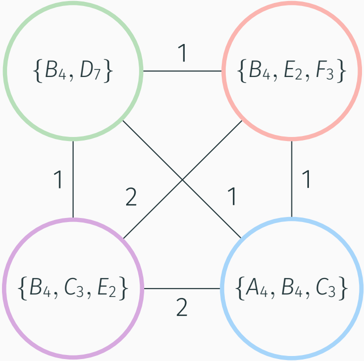
Connection of Clusters

Like before, subscripts denote the variable's *cardinality*.



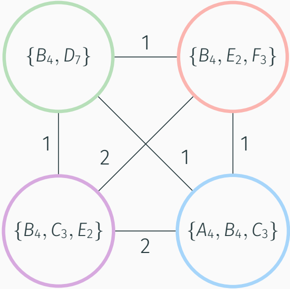
Connection of Clusters

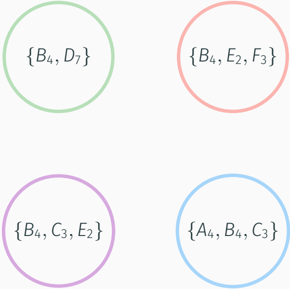
Count the common variables between each pair of clusters.



Connection of Clusters

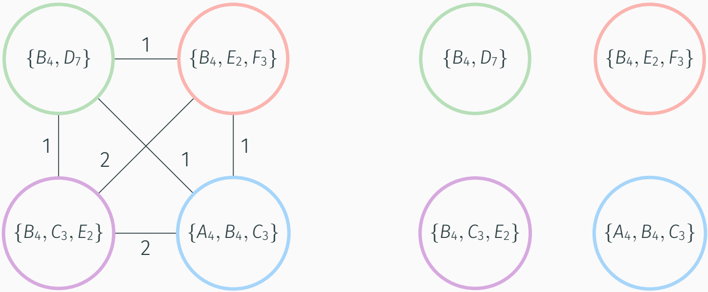
Create a new graph with only the nodes of the complete cluster graph.





Connection of Clusters

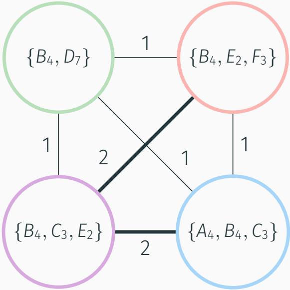
Select the edge that connects the two clusters with the most common variables *and* that would not create a loop in the right graph if moved.

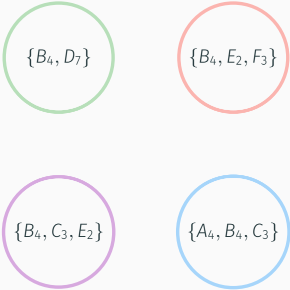


Connection of Clusters

Tie.

Then select the edge connecting the clusters that have the smallest *state space* size.

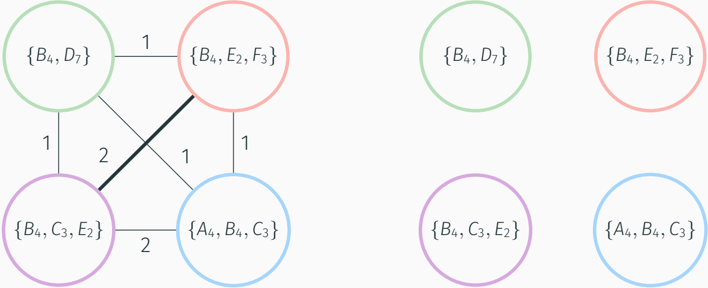




Connection of Clusters

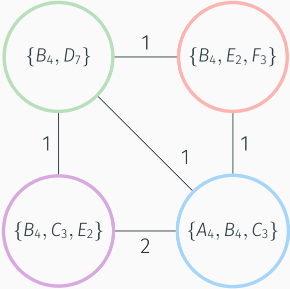
$$|\{B_4, C_3, E_2\}| + |\{B_4, E_2, F_3\}| < |\{B_4, C_3, E_2\}| + |\{A_4, B_4, C_3\}|,$$

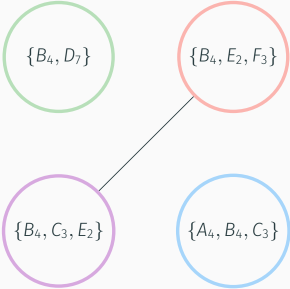
therefore the edge connecting $\{B_4, C_3, E_2\}$ and $\{B_4, E_2, F_3\}$ wins.



Connection of Clusters

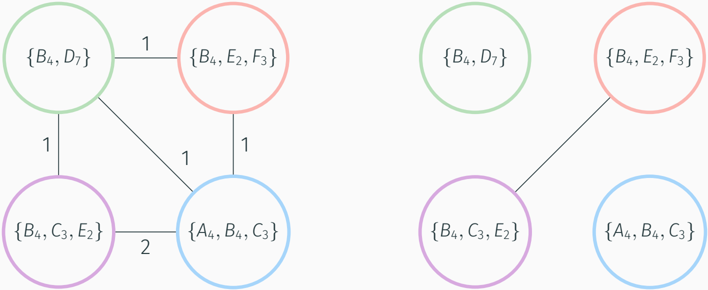
Move the selected edge to the right graph.



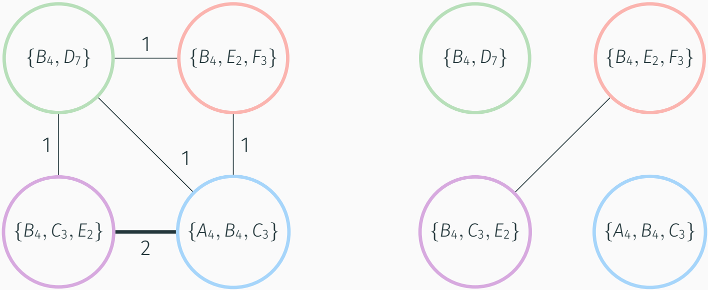


Connection of Clusters

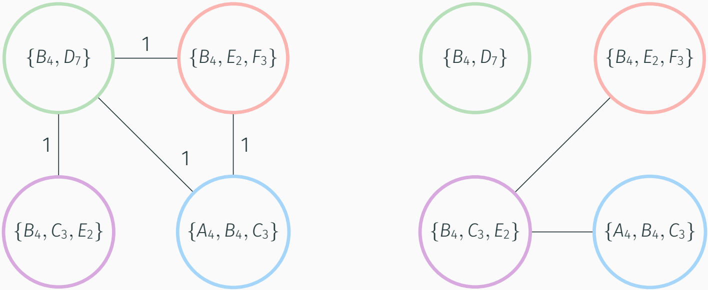
Repeat this procedure until the right graph has $N-1$ edges, where N is the number of clusters.



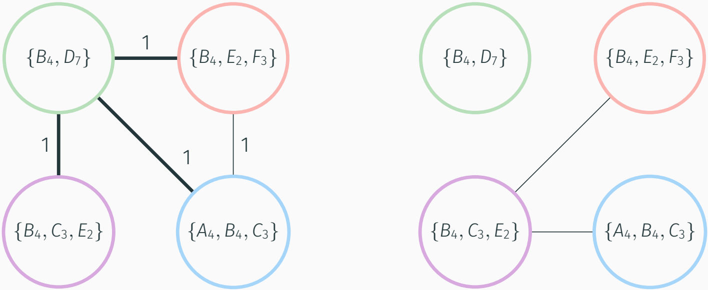
Connection of Clusters



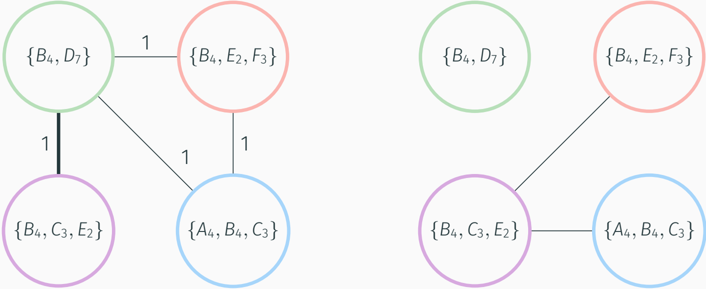
Connection of Clusters



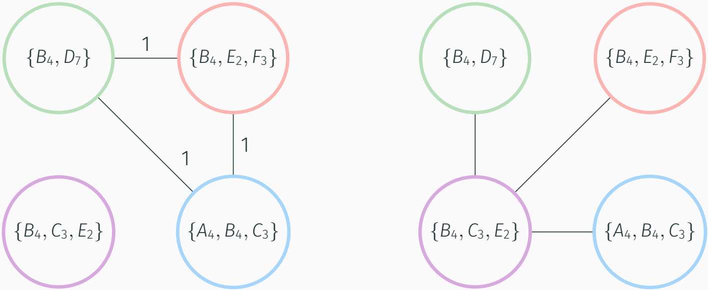
Connection of Clusters



Connection of Clusters



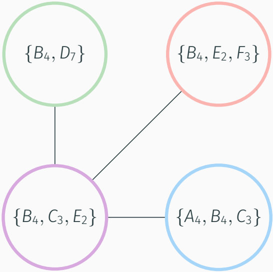
Connection of Clusters

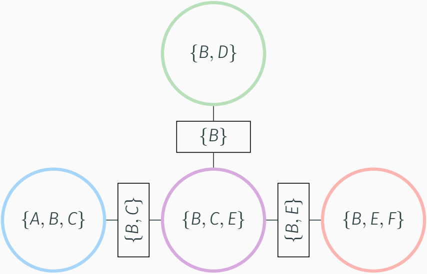


Connection of Clusters

Finally, we label each edge with a *sepset*,

i.e. the intersection of variables between adjacent clusters.





Junction Tree

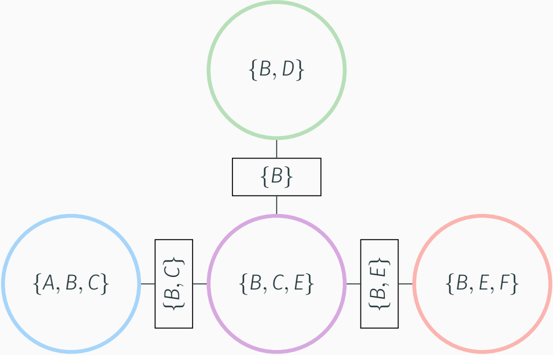


Figure 23: Junction tree

Overview

Probabilistic Graphical Model

$\downarrow$

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

$\downarrow$

1. Initialization
2. Observation entry

Inconsistent Junction Tree

$\downarrow$

Propagation

Consistent Junction Tree

$\downarrow$

1. Marginalization
2. Normalization

$P(V \mid E = e)$

Initialization

Multiply each conditional probability distribution $P(V \mid pa(V))$
into a cluster potential that contains its variables.

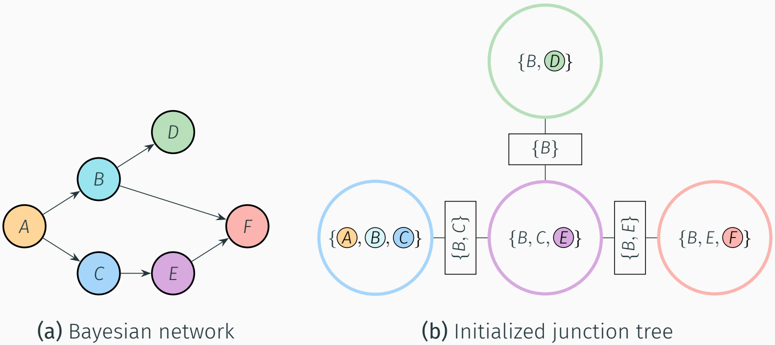


Figure 24: Initialization

Overview

Probabilistic Graphical Model

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

1. Initialization
2. Observation entry

Inconsistent Junction Tree

Propagation

Consistent Junction Tree

1. Marginalization
2. Normalization

$P(V \mid E = e)$

Observation Entry

Suppose that E is an observed variable and that
the table below is the *factor* associated to cluster BEF.

F	B	E	Element
0	0	0	0.25
0	0	1	0.35
0	1	0	0.08
0	1	1	0.16
1	0	0	0.05
1	0	1	0.07
1	1	0	0.00
1	1	1	0.00
2	0	0	0.15
2	0	1	0.21
2	1	0	0.09
2	1	1	0.18

Observation Entry

Now suppose that we observe that $E = 0$.

We enter this observation by zeroing all entries that do not agree with the evidence.

F	B	E	$\phi(F, B, E)$
0	0	0	0.25
0	0	1	0.00
0	1	0	0.08
0	1	1	0.00
1	0	0	0.05
1	0	1	0.00
1	1	0	0.00
1	1	1	0.00
2	0	0	0.15
2	0	1	0.00
2	1	0	0.09
2	1	1	0.00

Overview

Probabilistic Graphical Model

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

1. Initialization
2. Observation entry

Inconsistent Junction Tree

Propagation

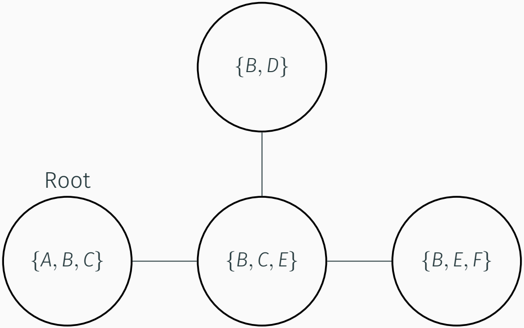
Consistent Junction Tree

1. Marginalization
2. Normalization

$P(V \mid E = e)$

Message Passing

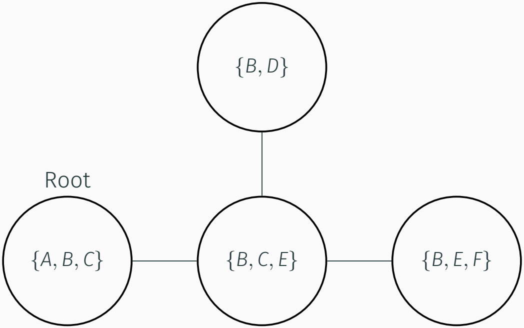
Represents the local computations that are necessary to spread each cluster's information with every other cluster in the graph.



Message Passing

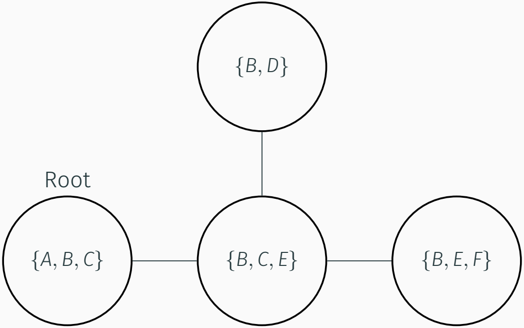
Designate an arbitrary cluster as the *root*.

This gives "direction" to the edges.



Message Passing

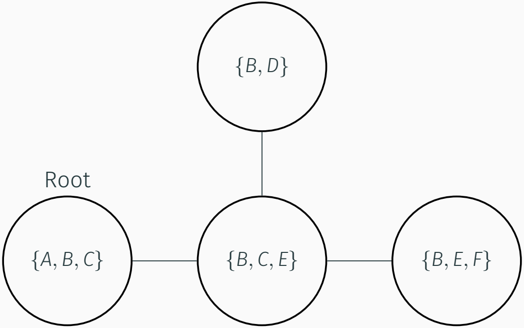
Two passes: *inward* and *outward*.



Message Passing

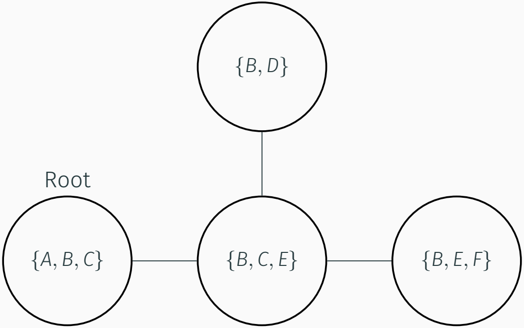
Inward pass: each cluster passes a message to its *parent*.

Backward pass: each cluster passes a message to each of its *children*.

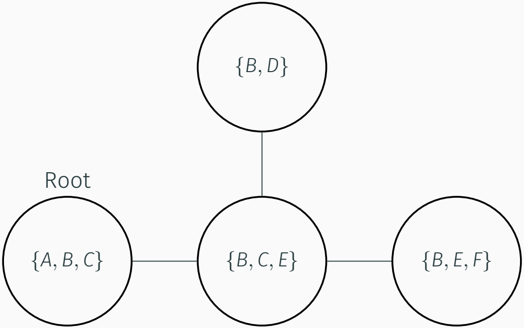


Message Passing

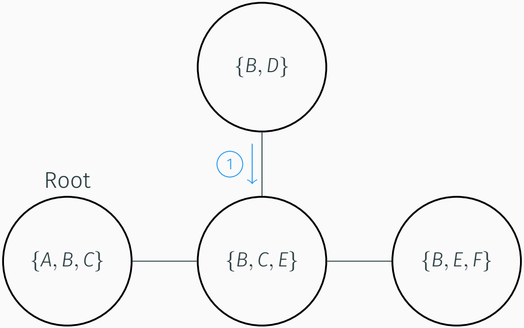
A cluster can only pass a message to a neighbor after it has received messages from all *other* neighbors.



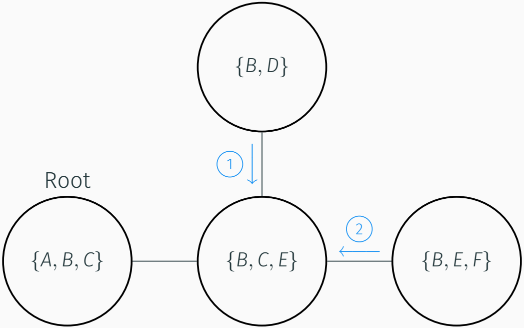
Message Passing



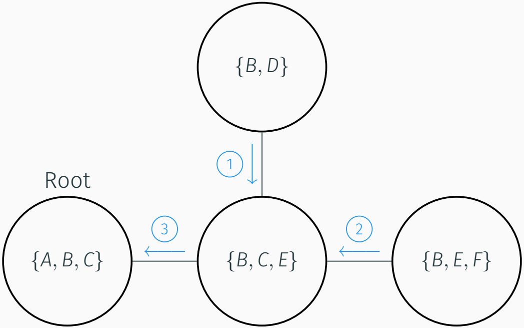
Message Passing



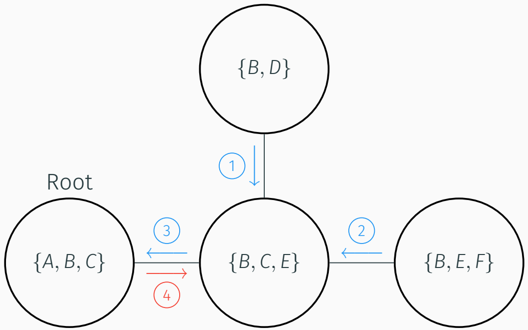
Message Passing



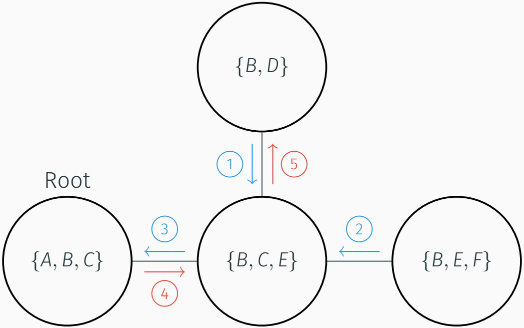
Message Passing



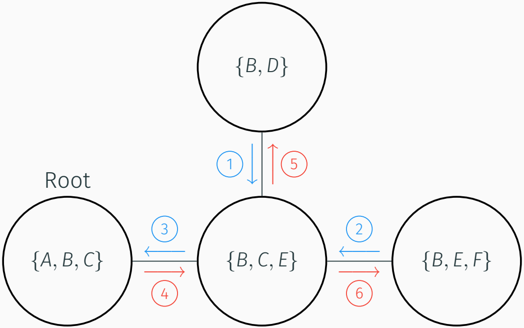
Message Passing



Message Passing



Message Passing



Overview

Probabilistic Graphical Model

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

1. Initialization
2. Observation entry

Inconsistent Junction Tree

Propagation

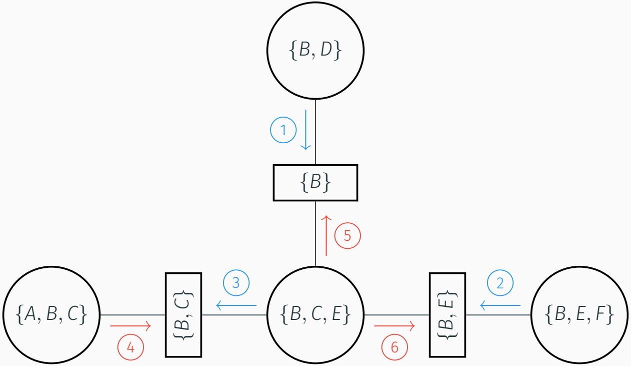
Consistent Junction Tree

1. Marginalization
2. Normalization

$P(V \mid \mathbf{E} = \mathbf{e})$

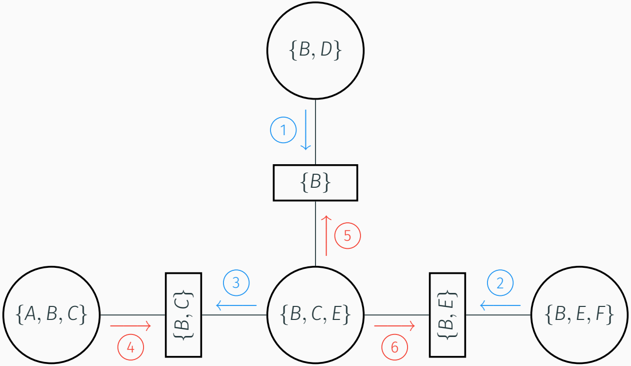
Marginalization

We marginalize each variable of interest from a sepset or cluster that contains it.



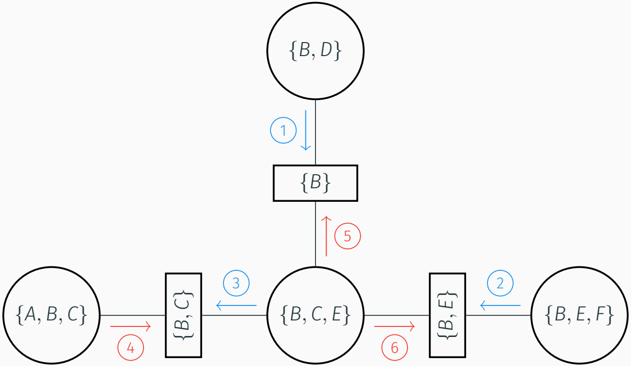
Marginalization

Suppose we are interested in variables A, B, and C.



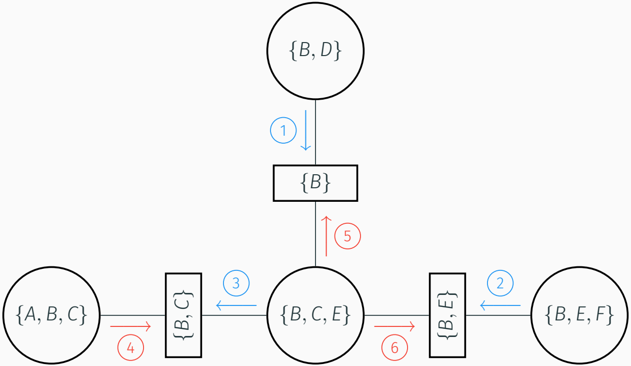
Marginalization

To marginalize a variable from a *cluster* we perform a product between the incoming messages and the cluster potential, and sum out all other variables.



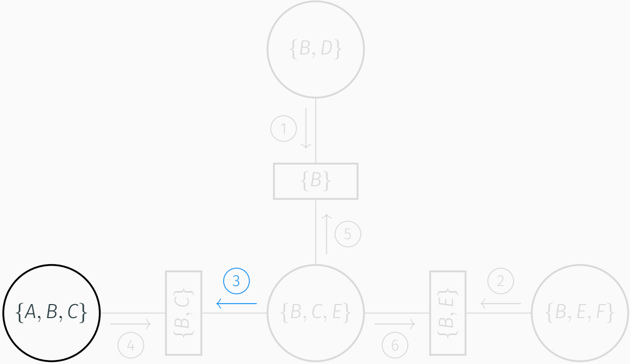
Marginalization

To marginalize a variable from a *sepset* we perform a product between the two messages incoming messages and sum out all other variables.



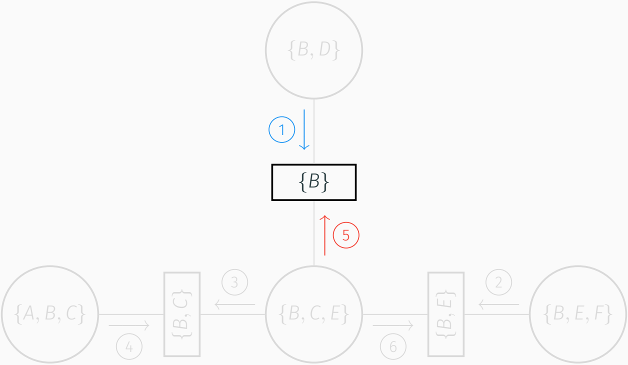
Marginalization

$$p(A, \boldsymbol{E} = \boldsymbol{e}) = \sum_{B,C} \psi_{\{A,B,C\}} \times \underset{\longleftarrow}{③}$$



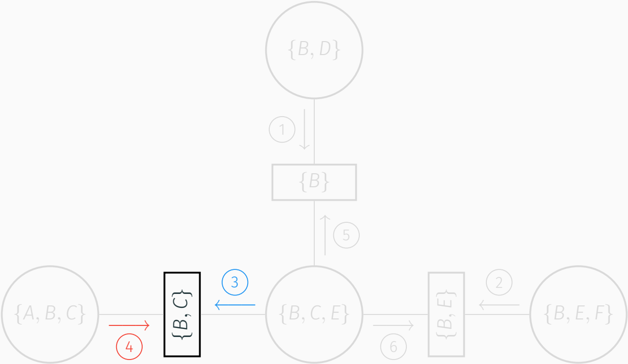
Marginalization

$$p(B, \boldsymbol{E} = \boldsymbol{e}) = \quad 1 \downarrow \times \uparrow 5$$



Marginalization

$$p(C, \boldsymbol{E} = \boldsymbol{e}) = \sum_{B} \overrightarrow{4} \times \overleftarrow{3}$$



Overview

Probabilistic Graphical Model

1. Moralization
2. Triangulation
3. Connection of clusters

Junction Tree

1. Initialization
2. Observation entry

Inconsistent Junction Tree

Propagation

Consistent Junction Tree

1. Marginalization
2. Normalization

$P(V \mid E = e)$

Normalization

The last step is to compute $P(V \mid \mathbf{E} = \mathbf{e})$ for each variable of interest V.

We do so by normalizing $P(V, \mathbf{E} = \mathbf{e})$, e.i.

$$P(V \mid \mathrm{E} = \mathrm{e}) = \frac{P(V, \mathrm{E} = \mathrm{e})}{P(\mathrm{E} = \mathrm{e})} = \frac{P(V, \mathrm{E} = \mathrm{e})}{\sum_V P(V, \mathrm{E} = \mathrm{e})}.$$

For example,

V	$\phi(V, \mathbf{E} = \mathbf{e})$
0	0.25
1	0.05
2	0.15

$\rightarrow$

V	$P(V \mid \mathbf{E} = \mathbf{e})$
0	0.55
1	0.11
2	0.33

The End

References i

C. Huang and A. Darwiche.
Inference in belief networks: A procedural guide.
International Journal of Approximate Reasoning, 15(3):225–263, 1996.

F. V. Jensen and F. Jensen.
Optimal junction trees, 2013.

U. Kjærulff.
Triangulation of Graphs - Algorithms giving small total state space.
Technical Report R 90-09, Department of Mathematics and Computer Science, Strandvejen, DK 9000 Aalborg, Denmark, 1990.

M. A. Paskin.
A short course on graphical models, 2003.