Web-Mining Agents

Prof. Dr. Ralf Möller

Dr. Özgür L. Özçep

Universität zu Lübeck

Institut für Informationssysteme

Tanya Braun (Lab)

Junction Trees

Agenda

Following lectures a glimpse on Logic & ML

``&" in {in, containing, for, with, using, augmented, ... }

1. Logic in ML: Constraining statistical models by background knowledge/ontology (lecture 9)
   - J. Deng et al.: Large-Scale Object Classification using Label Relation Graphs, LNCS, vol 8689, pp. 48-64, 2014.

2. ML in Logic: Computational Learning Theory in a logical framework (lecture 11)
   - M. Grohe and M. Ritzert. Learning first-order definable concepts over structures of small degree. ArXiv e-prints, Jan. 2017.

Agenda

- Lecture 8 (today): Junction trees
  - Preparation for Lecture 9
  - Recap of belief propagation
- Lecture 10: PAC Learning
  - Preparation for Lecture 11

Acknowledgements

- Slides based on slides of
  - Chris Williams: The Junction Tree Algorithm, October 2009

The Junction Tree Algorithm

Chris Williams[1]

School of Informatics, University of Edinburgh

October 2009

[1] Based on slides by David Barber

Why the Junction Tree Algorithm?

Different special vesions:
Shafer/Shenoy vs. Hugin vs Lauritzen-Spiegelhalter

- The JTA is a general-purpose algorithm for computing (conditional) marginals on graphs. We consider Hugin
- It does this by creating a tree of cliques, and carrying out a message-passing procedure on this tree belief propagation for arbitrary graphs (see lecture 2)
- The best thing about a general-purpose algorithm is that there is no longer any need to publish a separate paper explaining how to deal with each new model – the JTA generalises nearly all the popular previous special case algorithms.
- Reading: Jordan chapter 17 (Chapter of a of non-published book on probabilistic models)

Overview

- Clique Potential Representation
- Constructing a Junction Tree
  - Moralization
  - Triangulation
  - Assembling cliques into a junction tree
- Message Passing
- Introducing Evidence
- Propagation on a Junction Tree

Clique Potential Representation

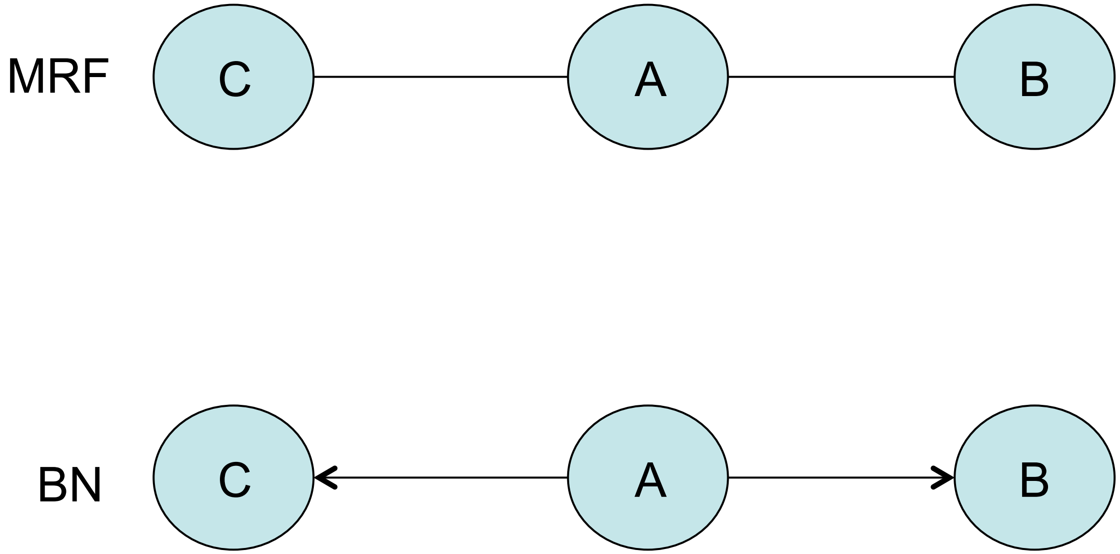
- Observe that for both directed and undirected graphs, the joint probability is in a product form.
- We can interpret the CPTs in *directed* graphs as potential functions.
- Basic idea is to represent probability distribution corresponding to any graph as a product of clique potentials:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C} \Psi_C(\mathbf{x}_C)$$

  where $\mathbf{x}_C$ is the set of variables corresponding to clique C.
- A *clique* is a fully-connected subset of nodes in a graph

Want a uniform treatment of directed and undirected models

The curse of normalization



Marginal P(C) = ?

In MRF need to calculate Z (incorporate B)
In BN not.

ÖÖ: The need for undirected models

Example

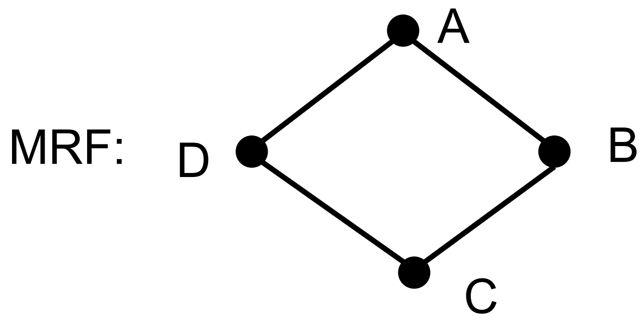
- 4 students a,b,c,d meet for homework in constellations: {a,d}, {a,b}, {d,c}, {b,c}
- Professor misspoke during lecture and gives rise to possible misconception among students
- A = student a has missconception
- Similarly boolean RVs B,C,D

- Aim: graphical model w/ independencies

$$(A \perp\!\!\!\perp C \mid \{B,D\}) \text{ and } (B \perp\!\!\!\perp D \mid \{A,C\})$$

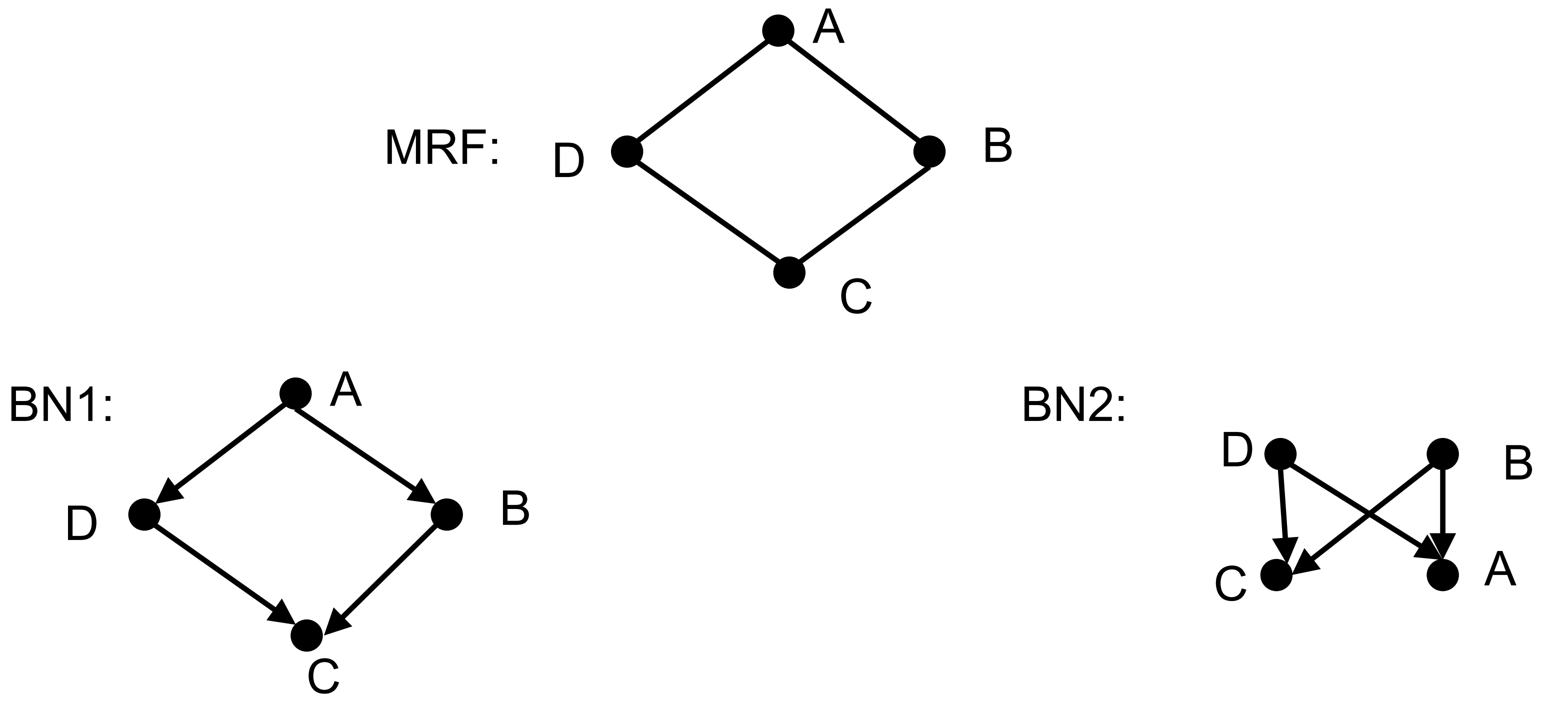
ÖÖ: The need for undirected models

- Aim: graphical model w/ independencies
  ($A \perp\!\!\!\perp C \mid \{B,D\}$) and ($B \perp\!\!\!\perp D \mid \{A,C\}$)



ÖÖ: The need for undirected models

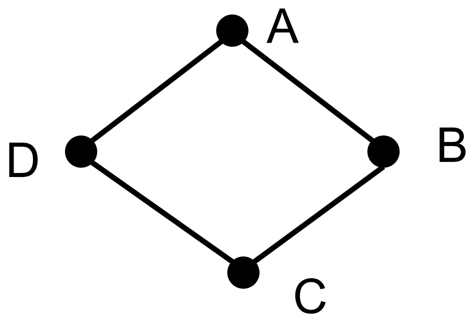
- Aim: graphical model w/ independencies
  ($A \perp\!\!\!\perp C \mid \{B,D\}$) and ($B \perp\!\!\!\perp D \mid \{A,C\}$)



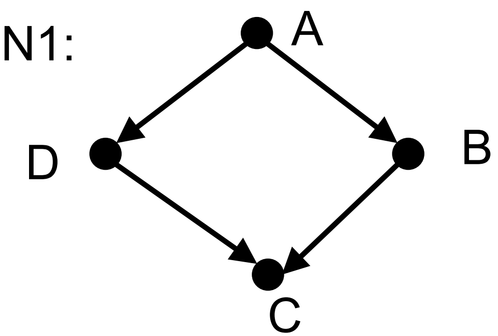
ÖÖ: The need for undirected models

- Aim: graphical model w/ independencies
  ($A \perp\!\!\!\perp C \mid \{B,D\}$) and ($B \perp\!\!\!\perp D \mid \{A,C\}$)

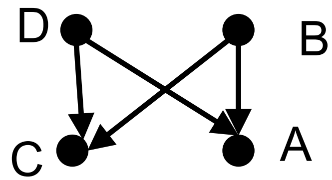
MRF:



BN1:



BN2:

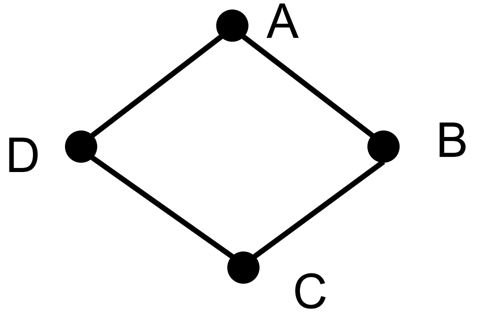


($A \perp\!\!\!\perp C \mid \{B,D\}$) captured by BN1 but also
($B \perp\!\!\!\perp D \mid \{A\}$) and not ($B \perp\!\!\!\perp D \mid \{A,C\}$)

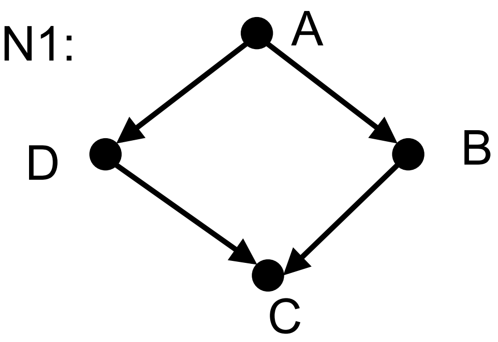
ÖÖ: The need for undirected models

- Aim: graphical model w/ independencies
  ($A \perp\!\!\!\perp C \mid \{B,D\}$) and ($B \perp\!\!\!\perp D \mid \{A,C\}$)

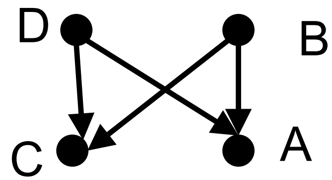
MRF:



BN1:



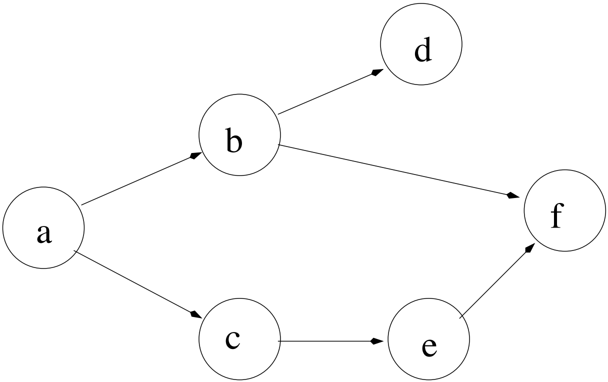
BN2:



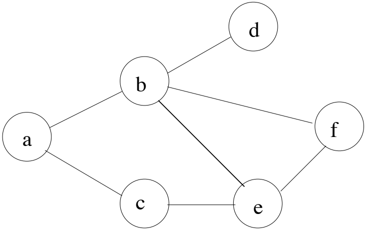
($A \perp\!\!\!\perp C \mid \{B,D\}$) captured by BN1 but also ($B \perp\!\!\!\perp D \mid \{A\}$) and not ($B \perp\!\!\!\perp D \mid \{A,C\}$)

($A \perp\!\!\!\perp C \mid \{B,D\}$) captured by BN2 but even ($B \perp\!\!\!\perp D$)

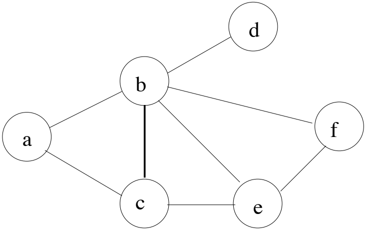
An example



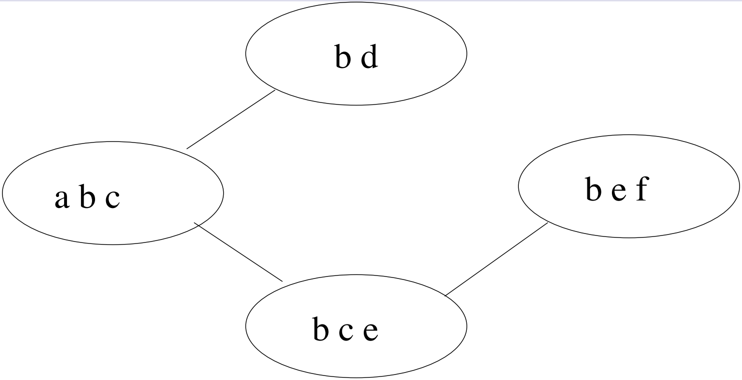
$$p(a,b,c,d,e,f) = p(a)p(b|a)p(c|a)p(d|b)p(e|c)p(f|b,e)$$



Moralization



Triangulation



The **clique potential** representation is

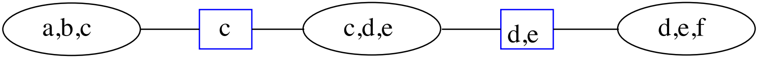
$$p(a, b, c, d, e, f) = \Psi(a, b, c)\Psi(b, d)\Psi(b, c, e)\Psi(b, e, f)$$

A valid assignment of cluster potentials is
$\Psi(a, b, c) = p(a)p(b|a)p(c|a)$, $\Psi(b, d) = p(d|b)$,
$\Psi(b, c, e) = p(e|c)$, $\Psi(b, e, f) = p(f|b, e)$ and $Z = 1$

Clique Trees and Separators

This is an example of a factor graph: factors (functions from sets of variables to real numbers, say) are presented as special nodes.

A clique tree is an (undirected) tree of cliques



Variables shared by neighbouring cliques are drawn in the separator sets in blue.

The potential representation of a clique tree is the product of the clique potentials, divided by the product of the separator potentials.

$$p(\mathbf{x}) = \frac{\prod_C \Psi_C(\mathbf{x}_C)}{\prod_S \Phi_S(\mathbf{x}_S)}$$

This is a very convenient definition. ((Normalization is handled by PhiS for empty S)

Initially, all separator potentials are set to 1.
After running the JTA, we will have

$$\Psi(\mathbf{x}_C) = p(\mathbf{x}_{\tilde{C}}, \bar{\mathbf{x}}_E)$$
$$\Phi(\mathbf{x}_S) = p(\mathbf{x}_{\tilde{S}}, \bar{\mathbf{x}}_E)$$

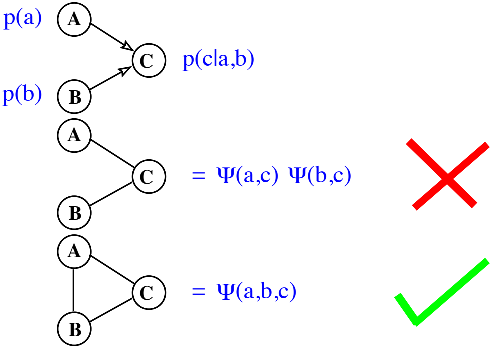
where $\tilde{C}$ denotes those variables in C that are not in E, and similarly for $\tilde{S}$.

Constructing a Junction Tree from a DAG

1. Moralize the graph
2. Triangulate the graph
3. Construct a junction tree

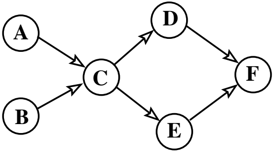
Moral Graphs

Let's represent the following DAG as a product of clique potentials:



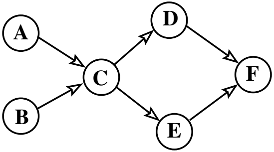
To ensure that a node and its parents are in the same clique, we have to *marry* the parents – *moralisation*.

A Moral Example to us all

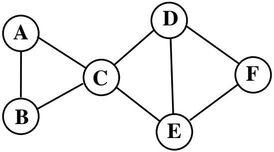


After moralisation, we get the following undirected graph

A Moral Example to us all



After moralisation, we get the following undirected graph



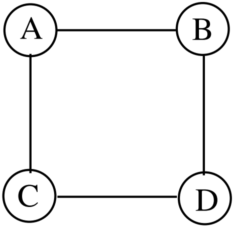
The product of clique potentials is

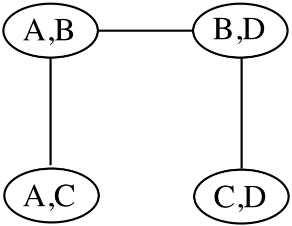
$$p(a, b, c, d, e, f) = \Psi(a, b, c)\Psi(c, d, e)\Psi(d, e, f)$$

where $\Psi(a, b, c) = p(a)p(b)p(c|a, b)$, $\Psi(c, d, e) = p(d|c)p(e|c)$, $\Psi(d, e, f) = p(f|d, e)$

The need for triangulation

Consider the following graph and a corresponding clique tree





C appears in two non-neighbouring cliques.

There is no guarantee that marginal on C in these two cliques should be equal, i.e $\sum_A \Psi(A, C) = \sum_D \Psi(C, D)$

That is, *local* consistency does not necessarily imply *global* consistency.

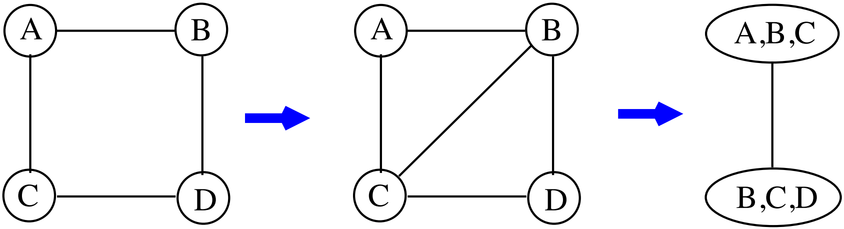
Triangulation provides a solution.

* Note in particular that in general potentials not marginal probabilities

* Remember soldier counting:
every soldier should (in the end) know
total number

Triangulation

In a triangulated graph, all loops containing 4 or more nodes contain a chord:



One way to create a triangulated graph is via the *elimination algorithm* (see Jordan §3.2)

```
UndirectedGraphEliminate(G, I)
    for each node X_i in I
        connect all of the remaining neighbors of X_i
        remove X_i from the graph
    end
```

Figure 3.5: A simple greedy algorithm for eliminating nodes in an undirected graph $\mathcal{G}$.

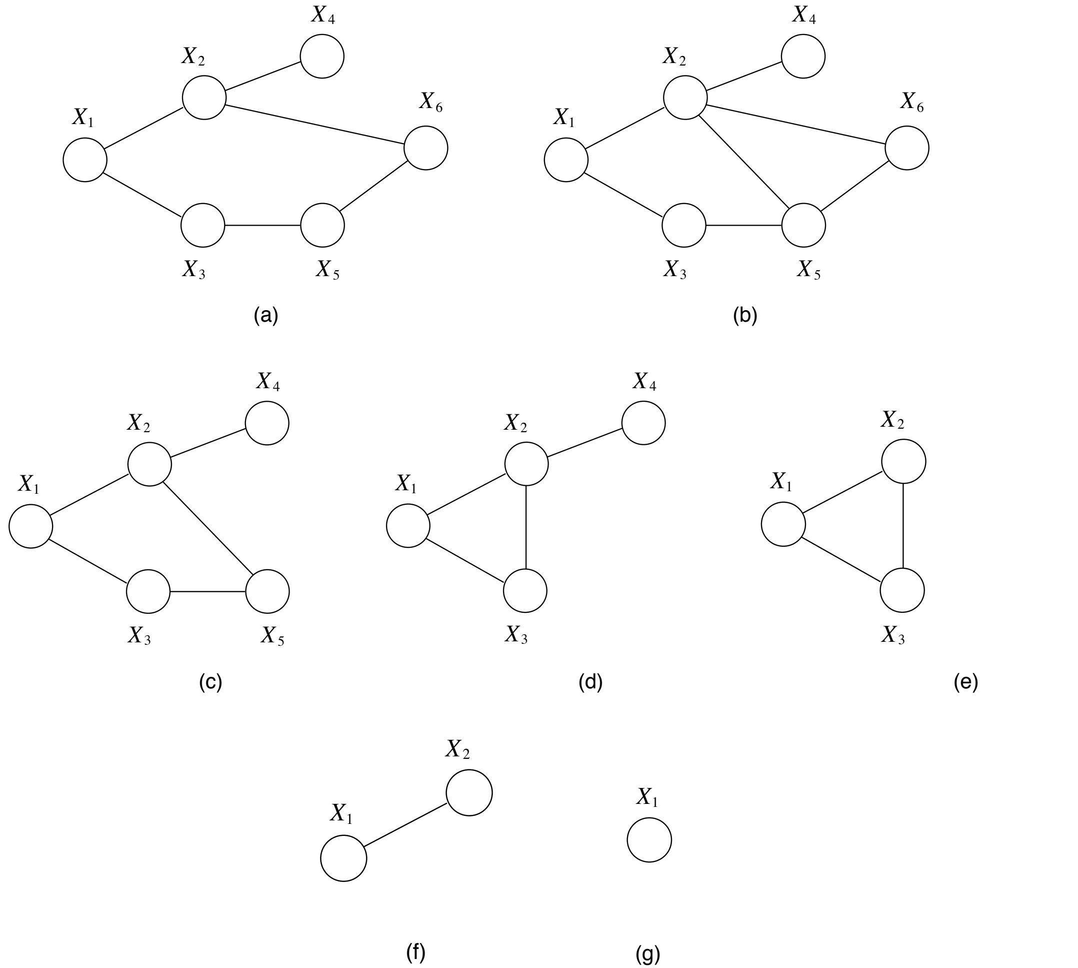


Figure 3.6: A run of the elimination algorithm under the elimination ordering $(6, 5, 4, 3, 2, 1)$. The original graph is shown in (a).

This is a triangulated graph

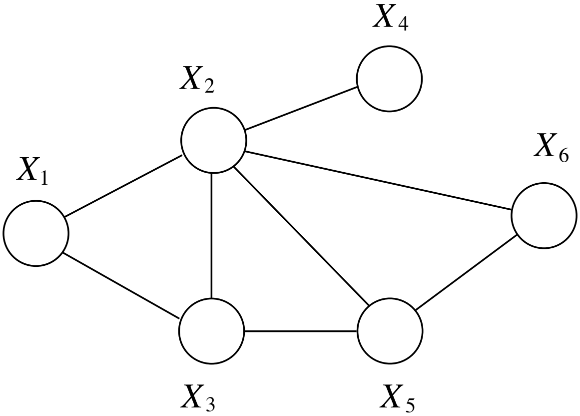
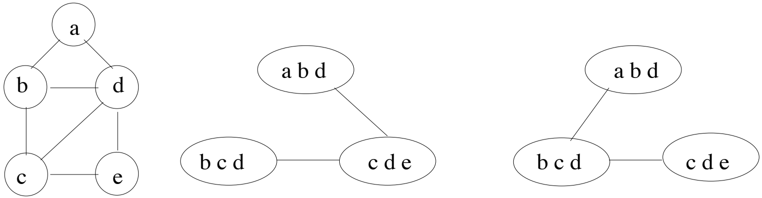


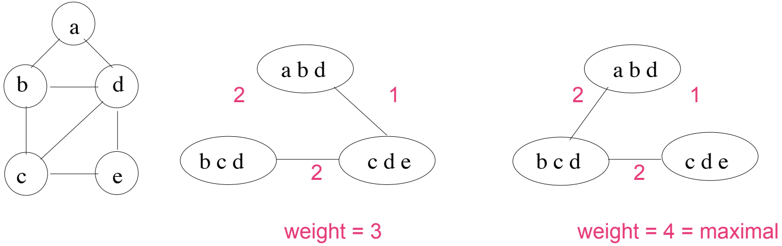
Figure 3.7: The reconstituted graph, showing the edges that were added during the elimination process.

Constructing a Junction Tree

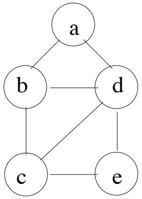
- A clique tree is a junction tree if it has the following junction tree property: if a node appears in two cliques, it appears everywhere on the path between the cliques.
- For every triangulated graph there exists a clique tree which obeys the junction tree property
- Thus local consistency implies global consistency

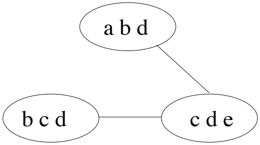


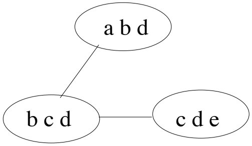
- Not all clique trees are junction trees
- **Theorem** A clique tree is a junction tree iff it is a maximal spanning tree, where the weight is given by the sum of the cardinalities of the separator sets



- Not all clique trees are junction trees
- **Theorem** A clique tree is a junction tree iff it is a maximal spanning tree, where the weight is given by the sum of the cardinalities of the separator sets







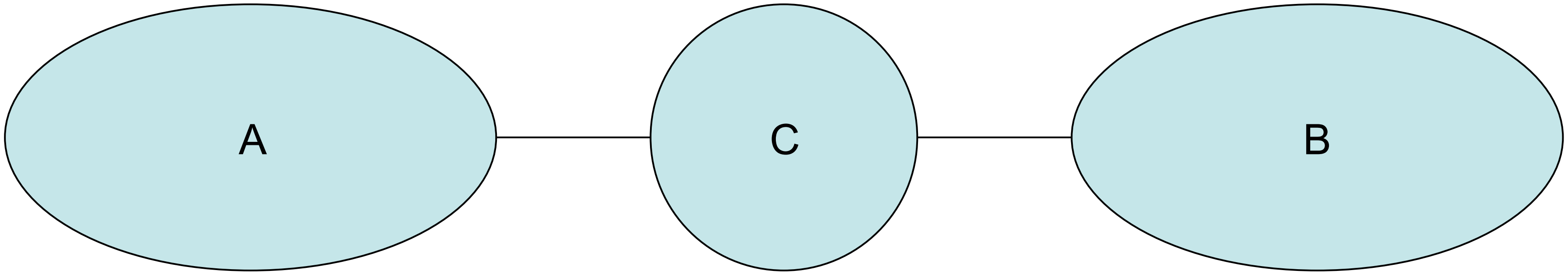
- Not all clique trees are junction trees
- **Theorem** A clique tree is a junction tree iff it is a maximal spanning tree, where the weight is given by the sum of the cardinalities of the separator sets

An alternative similar data structure are D-trees
(for decomposition tree)
(Perhaps in one of the next lectures)

Main observation: Graph decomposable iff triangulated

Decomposable Graphs

Decomposition (A,B,C)

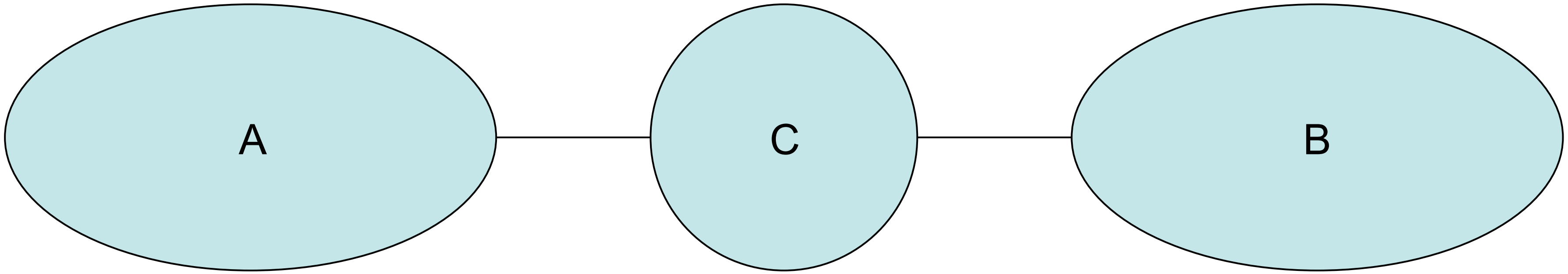


Undirected graph G = (V,E)

- $V = A \cup B \cup C$
- All paths between A and B go through C
- C is a complete subset of V

Decomposable Graphs

Decomposition (A,B,C)



Undirected graph G = (V,E)

- $V = A \cup B \cup C$
- All paths between A and B go through C
- C is a complete subset of V

Decomposable Graphs

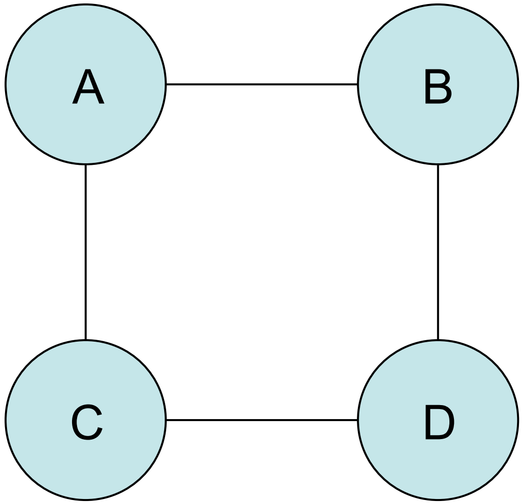
- A, B and/or C can be empty

- A, B are non-empty in a proper decomposition

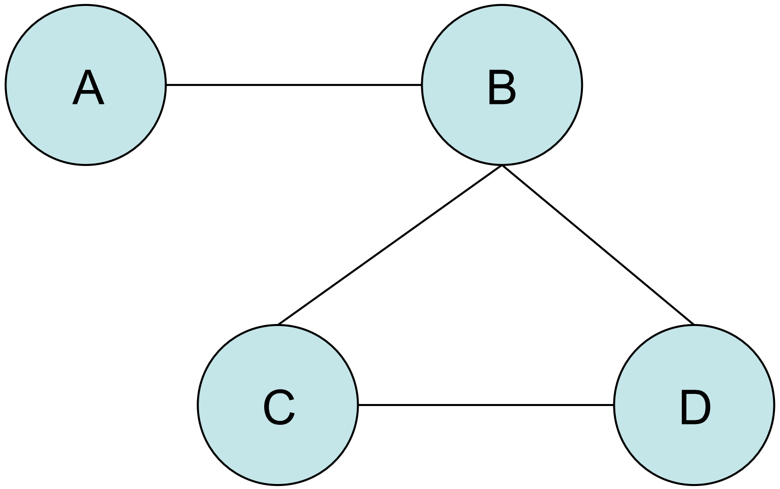
Decomposable Graphs

- G is decomposable if and only if
  - G is complete OR
  - It possesses a proper decomposition (A,B,C) such that
    - $G_{A \cup C}$ is decomposable
    - $G_{B \cup C}$ is decomposable

Decomposable Graphs

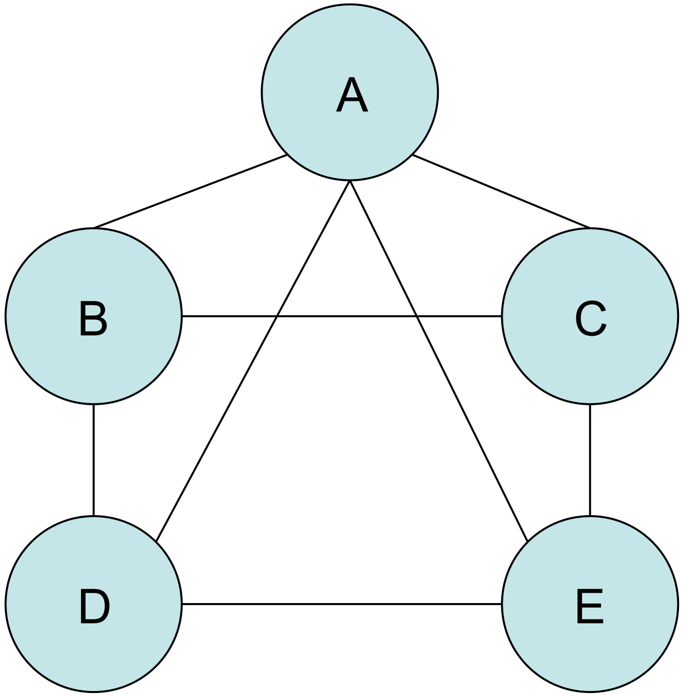


Not Decomposable

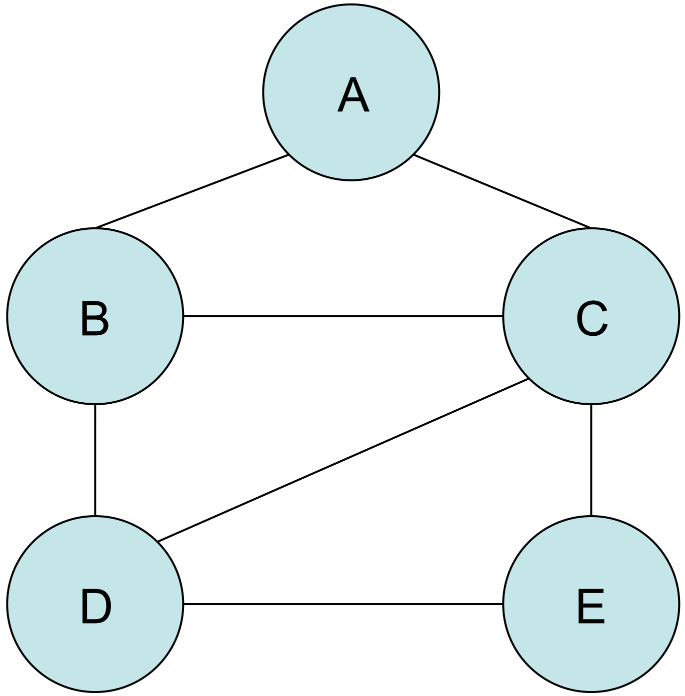


Decomposable

Decomposable Graphs



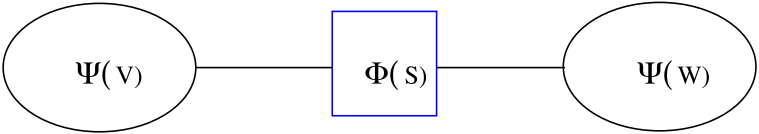
Not Decomposable



Decomposable

Message Passing

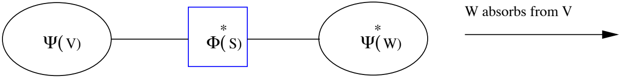
In order that the cliques contain all information required for marginals of the variables in the clique, we need to enforce *consistency*. That is, if clique V (containing a set of variables) and clique W share variables S, the marginals on their separators must be equal.



We need $\sum_{V \setminus S} \Psi(V) = \Phi(S) = \sum_{W \setminus S} \Psi(W)$.

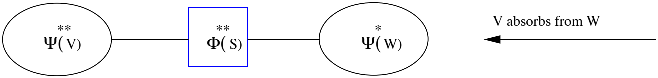
Absorption

Absorption passes a "message" from one node to another:



$\Psi^*(W) = \Psi(W)\frac{\Phi^*(S)}{\Phi(S)}$, where $\Phi^*(S) = \sum_{V\setminus S}\Psi(V)$

Similarly, after passing a message one way, we pass it the other:



$\Psi^{**}(V) = \Psi^*(V)\frac{\Phi^{**}(S)}{\Phi^*(S)}$, where $\Psi^*(V) = \Psi(V)$ and $\Phi^{**}(S) = \sum_{W\setminus S}\Psi^*(W)$

This ensures *consistency*:
$\sum_{V \setminus S} \Psi^{**}(V) = \Phi^{**}(S) = \sum_{W \setminus S} \Psi^{*}(W)$.
Also

$$\frac{\Psi(V)\Psi(W)}{\Phi(S)} = \frac{\Psi^{*}(V)\Psi^{*}(W)}{\Phi^{*}(S)} = \frac{\Psi^{**}(V)\Psi^{**}(W)}{\Phi^{**}(S)}$$

where $\Psi^{**}(W) = \Psi^{*}(W)$, thus maintaining the clique tree representation of the graph.

Show that $\Psi^{**}(V)$ and $\Psi^{**}(W)$ have the same marginals on S

Introducing Evidence

$$p(\mathbf{x}) = \prod_C \Psi_C(\mathbf{x}_C)$$

* Remember: Tilde(C) = all RVs in C not in E

Split nodes into H (hidden) and E (evidence)

* dash(x) = a concrete assignment to x

$$p(\mathbf{x}_H, \bar{\mathbf{x}}_E) = \prod_C \Psi_C(\mathbf{x}_{\tilde{C}}, \bar{\mathbf{x}}_{C\cap E}) \triangleq \prod_C \tilde{\Psi}_{\tilde{C}}(\mathbf{x}_{\tilde{C}})$$

This is a product of “slices” of potential functions.
Thus to introduce evidence, we modify the potentials in the original graph, setting any nodes to their evidential values.
One can also use the “evidence potential” approach by setting

$$\tilde{\Psi}_C(\mathbf{x}_C) = \Psi_C(\mathbf{x}_C)\delta(\mathbf{x}_{C\cap E}, \bar{\mathbf{x}}_{C\cap E})$$

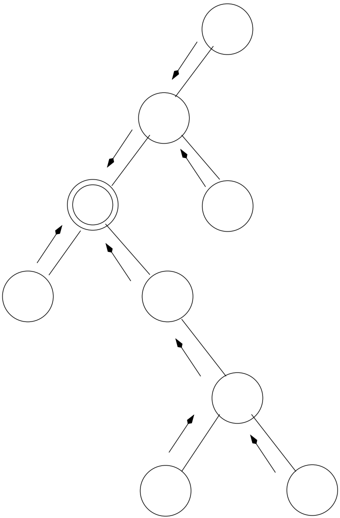
delta works like kronecker symbol

but this fills the clique potentials with lots of zeros thus and wastes storage and computation

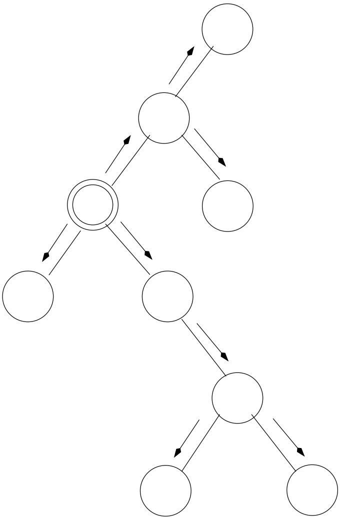
Propagation on a Junction Tree

- Node V can send exactly one message to a neighbour W, and it may only be sent when V has received a message from all of its other neighbours
- Choose one clique (arbitrarily) as a root of the tree; collect messages to this node and then distribute messages away from it
- After collection and distribution phases, we have in each clique that

$$\Psi(\mathbf{x}_C) = p(\mathbf{x}_{\tilde{C}}, \bar{\mathbf{x}}_E)$$



CollectEvidence



DistributeEvidence

Summary of JTA

- Convert belief network into JT
- Initialize potentials and separators
- Incorporate evidence (JT is inconsistent)
- CollectEvidence and DistributeEvidence (to give a consistent JT)
- Obtain clique marginals by marginalization/normalization

Proof of Correctness of JTA

Theorem

Let the probability $p(\mathbf{x}_H, \bar{\mathbf{x}}_E)$ be represented by the clique potentials of a junction tree. When the junction tree algorithm terminates, the clique potentials and separator potentials are proportional to the local marginal probabilities. In particular:

$$\Psi_C = p(\mathbf{x}_{\tilde{C}}, \bar{\mathbf{x}}_E), \qquad \Phi_S = p(\mathbf{x}_{\tilde{S}}, \bar{\mathbf{x}}_E)$$

Proof

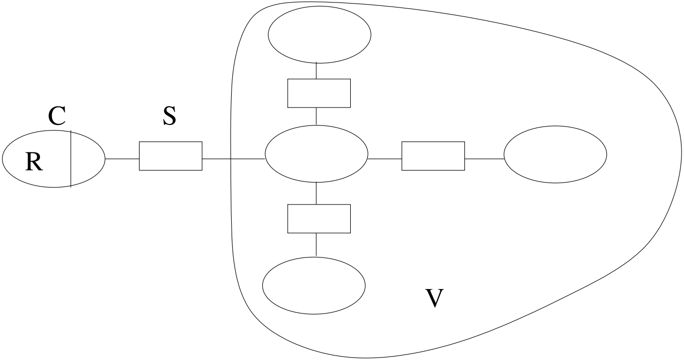
Observe that the separators are subsets of the cliques which are consistent with the cliques. Thus we only need to prove the result for the cliques.

Throughout the propagation process we have maintained the representation

$$p(\mathbf{x}_H, \bar{\mathbf{x}}_E) = \frac{\prod_C \Psi_C(\mathbf{x_C})}{\prod_S \Phi_S(\mathbf{x}_S)}$$

After the collect- and distribute-evidence stages the junction tree is consistent (i.e. the marginalization of the potentials of the cliques at either end of a separator give the same separator potential).

We now show that marginalization of the joint $p(\mathbf{x}_H, \bar{\mathbf{x}}_E)$ gives the desired result.



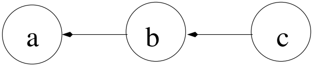
Choose a clique C that is a leaf of the JT with separator S. Let $\tilde{C} = C \backslash E$ and $\tilde{S} = S \backslash E$. Let $\tilde{R} = \tilde{C} \backslash \tilde{S}$, and the remaining non-evidence nodes be denoted $\tilde{T}$.

We now remove clique C by summing out $\tilde{R}$ from $p(\mathbf{x}_H, \bar{\mathbf{x}}_E) = p(\mathbf{x}_{\tilde{R}}, \mathbf{x}_{\tilde{S}}, \mathbf{x}_{\tilde{T}}, \bar{\mathbf{x}}_E)$

$$
\begin{aligned}
p(\mathbf{x}_{\tilde{T}}, \mathbf{x}_{\tilde{S}}, \bar{\mathbf{x}}_E) &= \sum_{\tilde{R}} p(\mathbf{x}_H, \bar{\mathbf{x}}_E) \\
&= \sum_{\tilde{R}} \frac{\prod_{\tilde{C}} \Psi_{\tilde{C}}(\mathbf{x}_{\tilde{C}})}{\prod_{\tilde{S}} \Phi_{\tilde{S}}(\mathbf{x}_{\tilde{S}})} \\
&= \sum_{\tilde{R}} \frac{\Psi_{\tilde{C}}(\mathbf{x}_{\tilde{C}})}{\Phi_{\tilde{S}}(\mathbf{x}_{\tilde{S}})} \frac{\prod_{\tilde{C}' \neq C} \Psi_{\tilde{C}'}(\mathbf{x}_{\tilde{C}'})}{\prod_{\tilde{S}' \neq S} \Phi_{\tilde{S}'}(\mathbf{x}_{\tilde{S}'})} \\
&= \frac{\sum_{\tilde{R}} \Psi_{\tilde{C}}(\mathbf{x}_{\tilde{C}})}{\Phi_{\tilde{S}}(\mathbf{x}_{\tilde{S}})} \frac{\prod_{\tilde{C}' \neq C} \Psi_{\tilde{C}'}(\mathbf{x}_{\tilde{C}'})}{\prod_{\tilde{S}' \neq S} \Phi_{\tilde{S}'}(\mathbf{x}_{\tilde{S}'})} \\
&= \frac{\prod_{\tilde{C}' \neq C} \Psi_{\tilde{C}'}(\mathbf{x}_{\tilde{C}'})}{\prod_{\tilde{S}' \neq S} \Phi_{\tilde{S}'}(\mathbf{x}_{\tilde{S}'})}
\end{aligned}
$$

Applying this process repeatedly we obtain $p(\mathbf{x}_{\tilde{C}}, \bar{\mathbf{x}}_E) = \Psi_{\tilde{C}}(\mathbf{x}_{\tilde{C}}, \bar{\mathbf{x}}_E)$

JTA example



Compute

- $p(b)$
- $p(b|a=0, c=1)$
- $p(c|b=1)$