Graphical Models

Lecture 5:

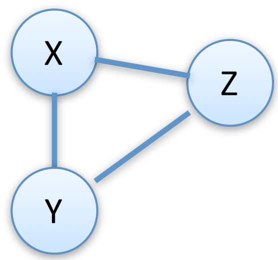
Undirected Graphical Models, continued

Andrew McCallum
mccallum@cs.umass.edu

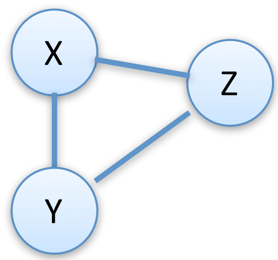
Thanks to Noah Smith and Carlos Guestrin for some slide materials.

What are *factor graphs*?

What are the Factors?



What are the Factors?

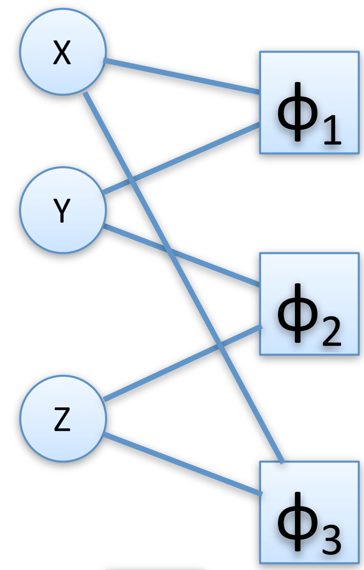


(You can't tell from the graph.)

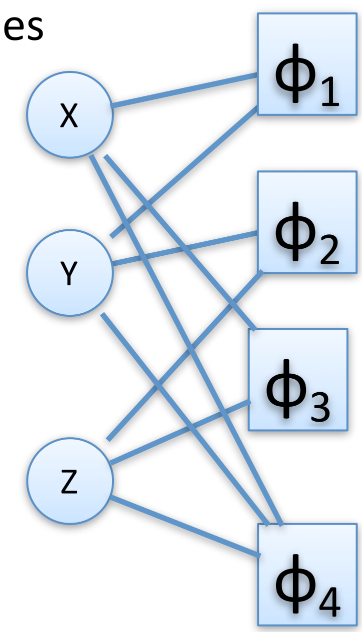
Factor Graphs

- Bipartite graph
  - Variable nodes (circles)
  - Factor nodes (squares)
  - Edge between variable and factor if the factor depends on that variable.
- Makes the factors more obvious.
- Other advantages later, in approximate inference.

pairwise Markov network

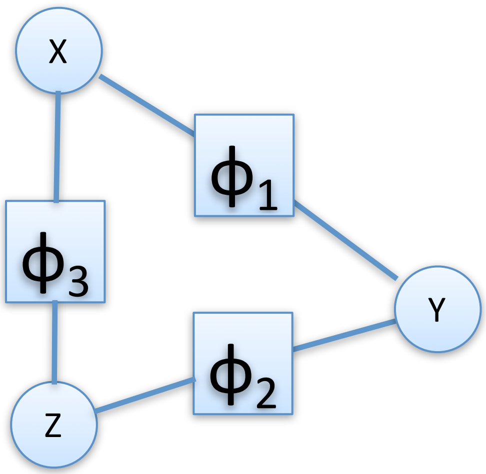


all cliques

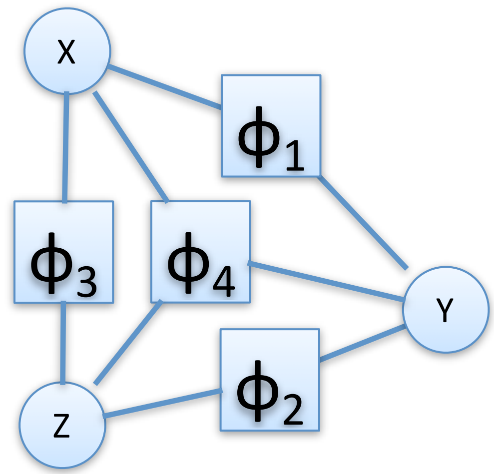


Factor Graphs

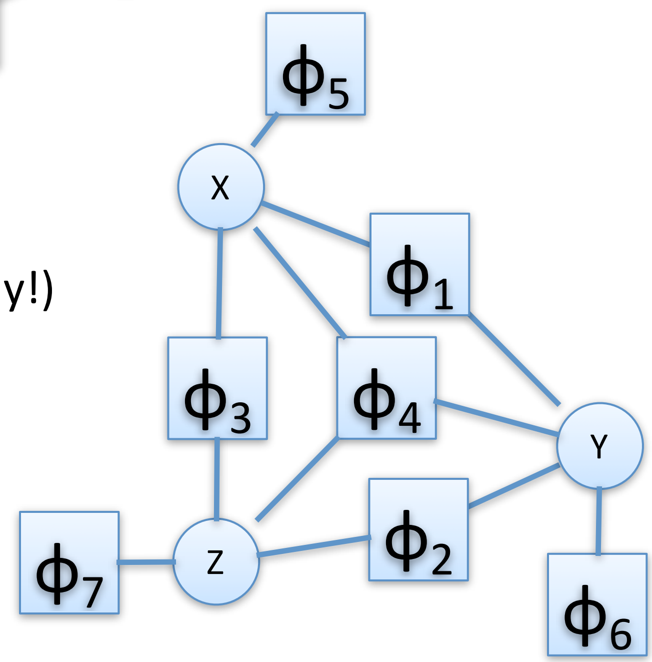
pairwise Markov network



all cliques



all cliques (really!)



How are undirected models typically parameterized?

Markov Networks (General Form)

- Let $\mathbf{D}_i$ denote the set of variables (subset of **X**) in the ith clique.
- Probability distribution is a **Gibbs** distribution:

$$
\begin{aligned}
P(\boldsymbol{X}) &= \frac{U(\boldsymbol{X})}{Z} \\
U(\boldsymbol{X}) &= \prod_{i=1}^{m} \phi_i(\boldsymbol{D}_i) \\
Z &= \sum_{\boldsymbol{x} \in \mathrm{Val}(\boldsymbol{X})} U(\boldsymbol{x})
\end{aligned}
$$

Logarithmic Representation

- Markov network:

$$\begin{aligned} P(\boldsymbol{X}) &= \frac{U(\boldsymbol{X})}{Z} \\ U(\boldsymbol{X}) &= \prod_{i=1}^{m} \phi_i(\boldsymbol{D}_i) \\ Z &= \sum_{\boldsymbol{x} \in \mathrm{Val}(\boldsymbol{X})} U(\boldsymbol{x}) \end{aligned}$$

Draw exp function.
Discuss meaning of
+/- ψ

- Logarithmic:

$$\phi(D_i) = e^{-\psi_i(D_i)}$$

$$\begin{aligned} \phi_i(\boldsymbol{D}_i) &= e^{\log \phi_i(\boldsymbol{D}_i)} \\ P(\boldsymbol{X}) &= \frac{1}{Z} e^{\sum_i \log \phi_i(\boldsymbol{D}_i)} \\ &= \frac{1}{Z} e^{-\sum_i \psi_i(\boldsymbol{D}_i)} \end{aligned}$$

Energy (lower energy = higher probability) = $-\sum_i \psi_i(\boldsymbol{D}_i)$

Log-Linear Markov Networks with *features*

Example:
Feature testing equality.
Compare # params

- A **feature** is a function f : Val($\mathbf{D}_i$) → $\mathbb{R}$.
- Log-linear model:

$$P(\boldsymbol{X}) = \frac{1}{Z}e^{\sum_i \log \phi_i(\boldsymbol{D}_i)}$$
$$= \frac{1}{Z}e^{-\sum_i \psi_i(\boldsymbol{D}_i)}$$
$$= \frac{1}{Z}e^{\sum_i \sum_j f_j(\boldsymbol{D}_i) w_j}$$

- Features and weights can be *reused* for different factors.
  - Typical: features designed by expert, weights learned from data.
  - (Note that *reusing* breaks parameter independence.)

More about *reusing* when we get to *Template* models.

- Log of the probability is *linear* in the weights **w**.
  - Ignoring Z, which is a constant for a given **w**.

Generalized Linear Model

- Score is defined as a *linear* function of **X**:

$$f(\boldsymbol{X}) = w_0 + \underbrace{\sum_i w_i X_i}_{Z}$$

Z = f(X) is a random variable

- Probability distribution over binary value Y is defined by:

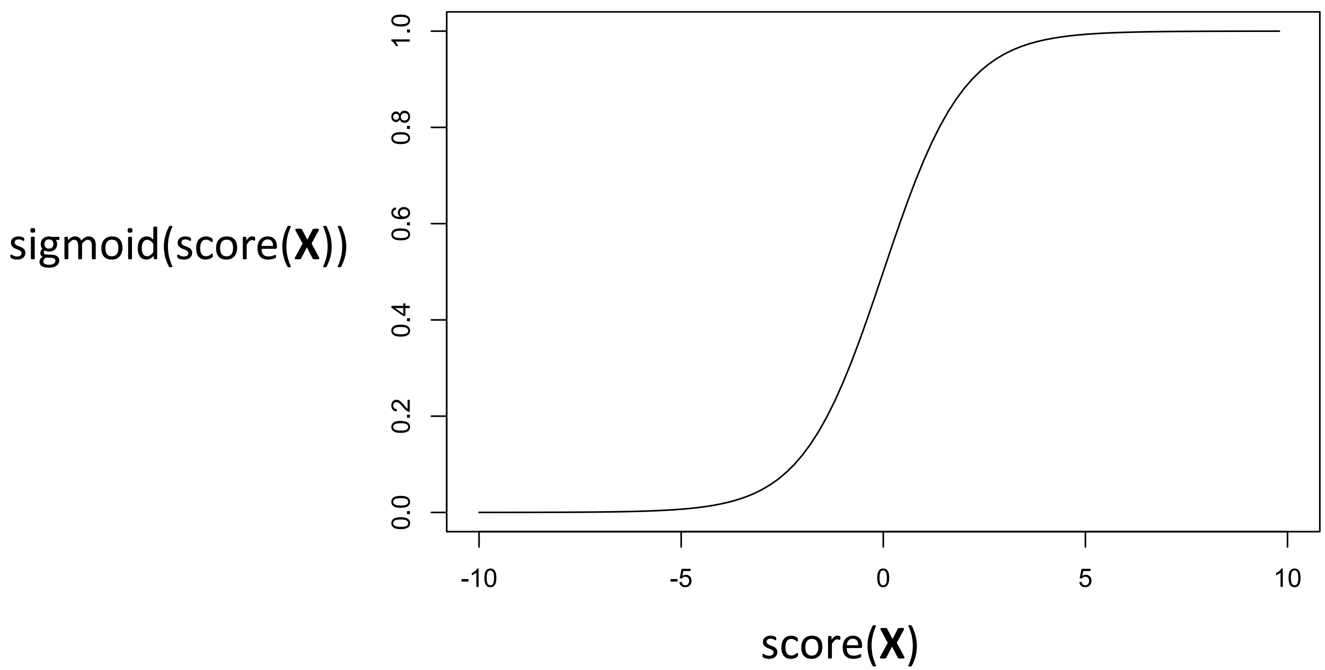
$$P(Y = 1) = \text{sigmoid}(f(\boldsymbol{X}))$$

- Sample Y.

From lecture 3!

Independent Causes

- Many "additive" effects combine to score **X**
- P(Y = 1) is defined as a function of **X**



From lecture 3!

$$\text{sigmoid}(z) = \frac{e^z}{1+e^z}$$

Markov Networks as a Generalized Linear Model

- Sigmoid equates to *binary* output log-linear model.
- More generally, *multinomial* logit:
  take a linear score (Z in lecture 3),
  exponentiate, and normalize (Z in Gibbs dist.)
  - Don't confuse the Zs.

- The generalized linear model we used for CPDs is a log-linear distribution.

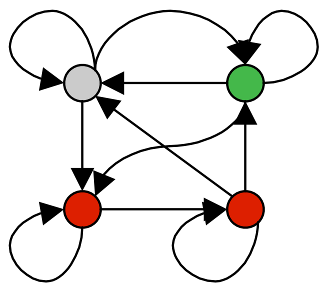
What is a
Conditional Random Field?

How are they motivated?

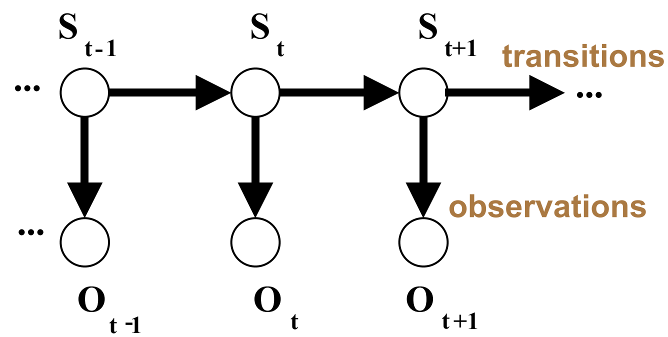
Hidden Markov Models

HMMs are the standard sequence modeling tool in genomics, music, speech, NLP, …

Finite state model

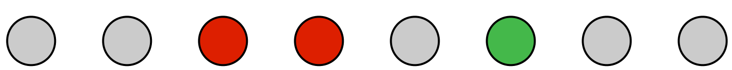


Graphical model



Generates:

State sequence



Observation sequence

o_1 o_2 o_3 o_4 o_5 o_6 o_7 o_8

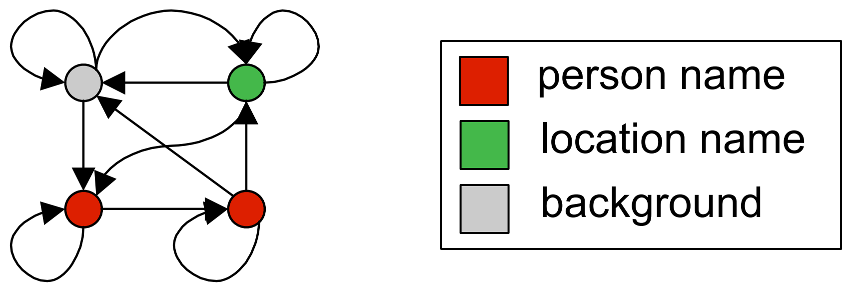
$$P(\vec{s},\vec{o}) \propto \prod_{t=1}^{|\vec{o}|} P(s_t \mid s_{t-1}) P(o_t \mid s_t)$$

IE with Hidden Markov Models

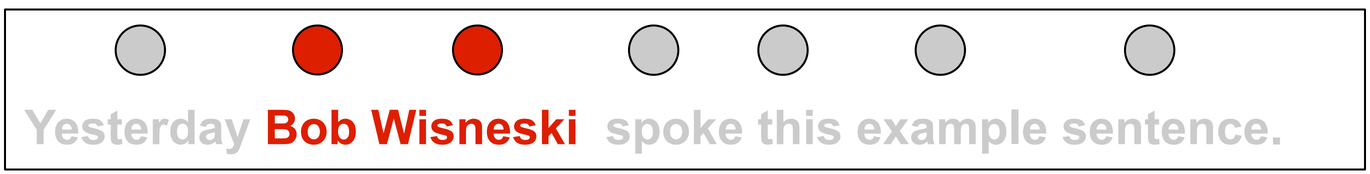
Given a sequence of observations:

Yesterday Yoav Freund spoke this example sentence.

and a trained HMM:



Find the most likely state sequence: (Viterbi)



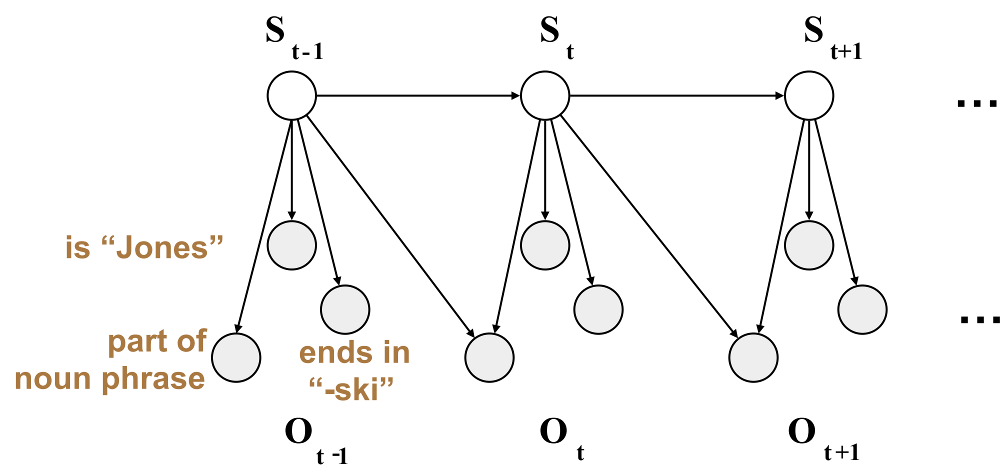
Any words said to be generated by the designated "person name" state extract as a person name:

Person name: Bob Wisneski

We want More than an Atomic View of Words

Would like richer representation of text: many arbitrary, overlapping features of the words.

- identity of word
- ends in "-ski"
- is capitalized
- is part of a noun phrase
- is in a list of city names
- is under node X in WordNet
- is in bold font
- is indented
- is in hyperlink anchor
- last person name was female
- next two words are "and Associates"



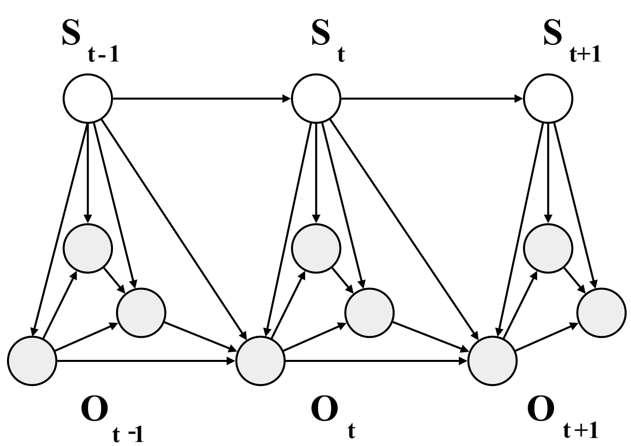
Problems with Richer Representation and a Joint Model

These arbitrary features are not independent.

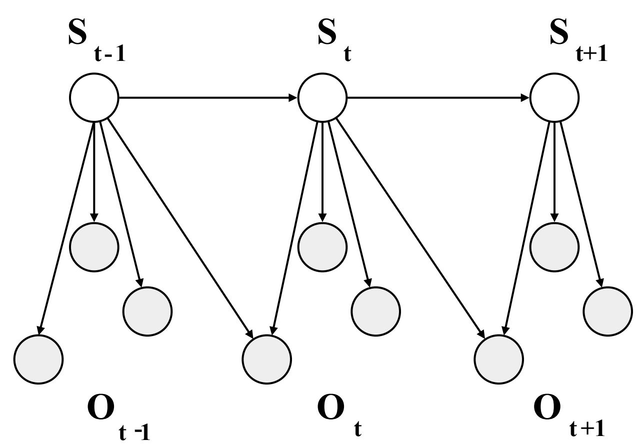
- Multiple levels of granularity (chars, words, phrases)
- Multiple dependent modalities (words, formatting, layout)
- Past & future

Two choices:

Model the dependencies.
Each state would have its own Bayes Net. *But we are already starved for training data!*



Ignore the dependencies.
This causes "over-counting" of evidence (ala naïve Bayes). *Big problem when combining evidence, as in Viterbi!*



Conditional Sequence Models

- We prefer a model that is trained to maximize a *conditional* probability rather than *joint* probability: **$P(\bar{s}|\bar{o})$ instead of $P(\bar{s},\bar{o})$:**
  - Can examine features, but not responsible for generating them.
  - Don't have to explicitly model their dependencies.
  - Don't "waste modeling effort" trying to generate what we are given at test time anyway.

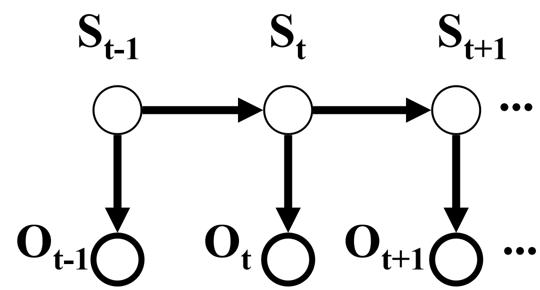
From HMMs to Conditional Random Fields

[Lafferty, McCallum, Pereira 2001]

$$\vec{s} = s_1, s_2, \ldots s_n \qquad \vec{o} = o_1, o_2, \ldots o_n$$

Joint

$$P(\vec{s}, \vec{o}) = \prod_{t=1}^{|\vec{o}|} P(s_t \mid s_{t-1}) P(o_t \mid s_t)$$

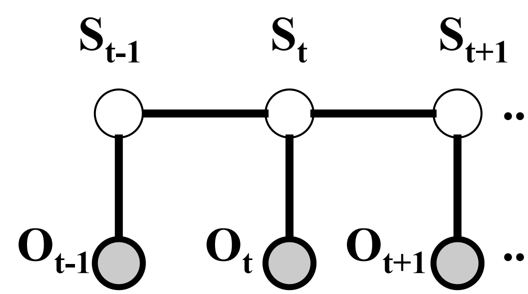


Conditional

$$P(\vec{s} \mid \vec{o}) = \frac{1}{P(\vec{o})} \prod_{t=1}^{|\vec{o}|} P(s_t \mid s_{t-1}) P(o_t \mid s_t)$$

$$= \frac{1}{Z(\vec{o})} \prod_{t=1}^{|\vec{o}|} \Phi_s(s_t, s_{t-1}) \Phi_o(o_t, s_t)$$

where $\Phi_o(t) = \exp\left(\sum_k \lambda_k f_k(s_t, o_t) \right)$



(A super-special case of Conditional Random Fields.)

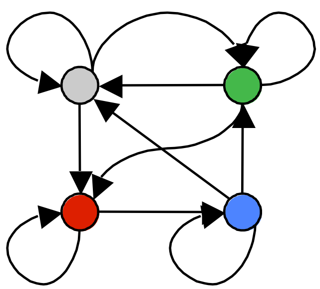
Set parameters by maximum likelihood, using optimization method on δL.

(Linear Chain) Conditional Random Fields

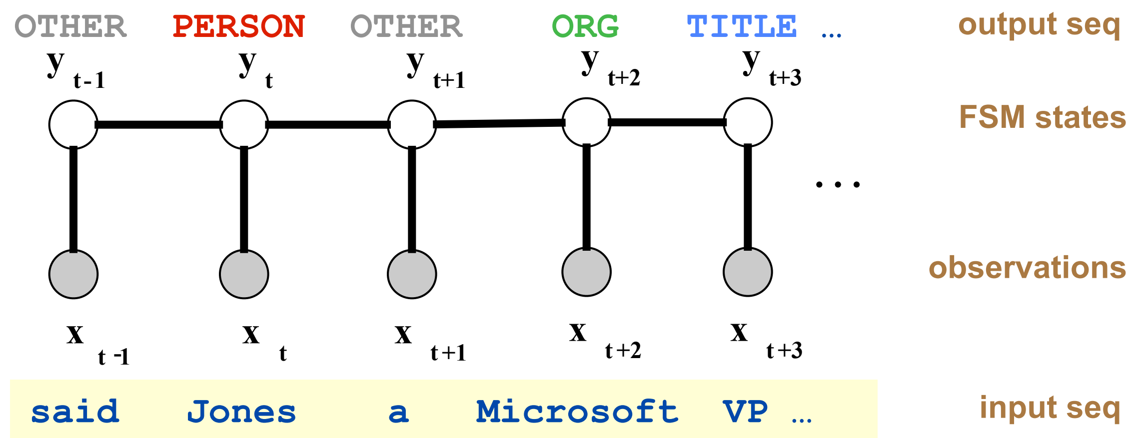
[Lafferty, McCallum, Pereira 2001]

Undirected graphical model, trained to maximize *conditional* probability of output (sequence) given input (sequence)

Finite state model



Graphical model



$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \prod_t \Phi(y_t, y_{t-1}, \mathbf{x}, t) \quad \textbf{where} \quad \Phi(y_t, y_{t-1}, \mathbf{x}, t) = \exp\left(\sum_k \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}, t) \right)$$

Wide-spread interest, positive experimental results in many applications.

Noun phrase, Named entity [HLT'03], [CoNLL'03]
Protein structure prediction [ICML'04]
IE from Bioinformatics text [Bioinformatics '04],…

Asian word segmentation [COLING'04], [ACL'04]
IE from Research papers [HTL'04]
Object classification in images [CVPR '04]

Table Extraction from Government Reports

Cash receipts from marketings of milk during 1995 at $19.9 billion dollars, was slightly below 1994. Producer returns averaged $12.93 per hundredweight, $0.19 per hundredweight below 1994. Marketings totaled 154 billion pounds, 1 percent above 1994. Marketings include whole milk sold to plants and dealers as well as milk sold directly to consumers.

An estimated 1.56 billion pounds of milk were used on farms where produced, 8 percent less than 1994. Calves were fed 78 percent of this milk with the remainder consumed in producer households.

Milk Cows and Production of Milk and Milkfat:
United States, 1993-95

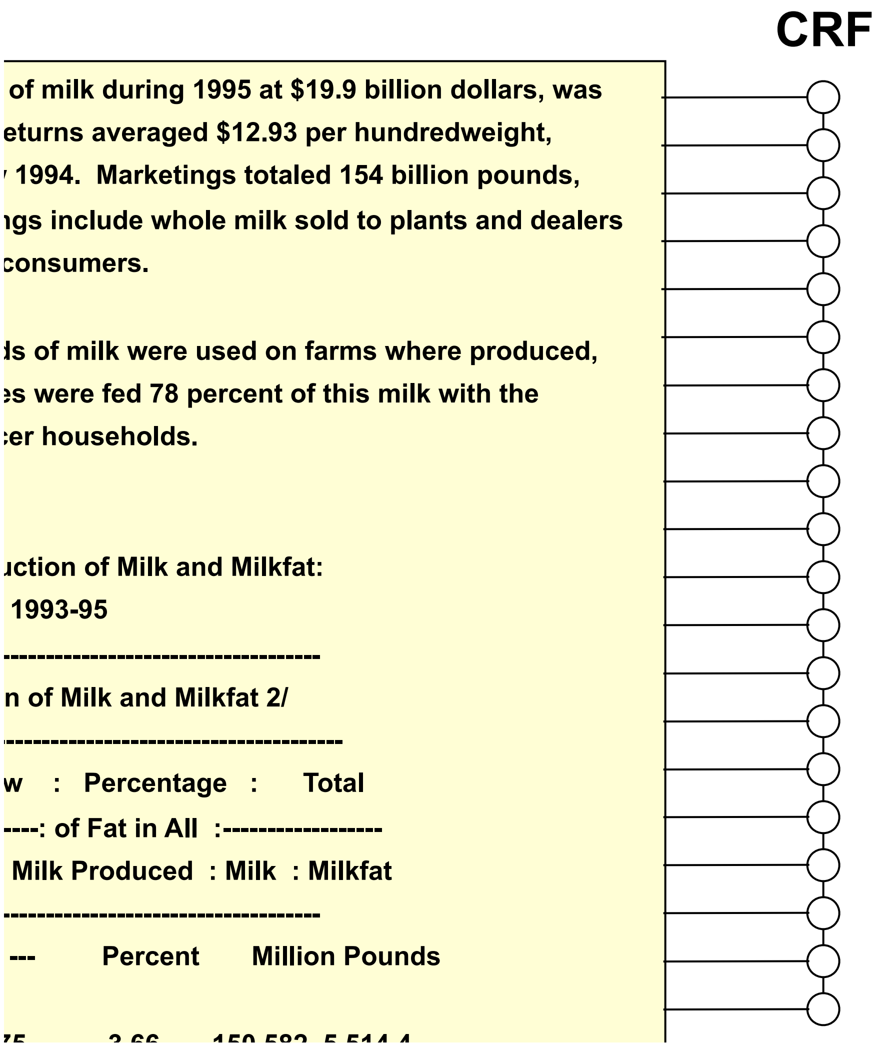
Year	Number of Milk Cows 1/	Production of Milk and Milkfat 2/: Per Milk Cow: Milk	Per Milk Cow: Milkfat	Percentage of Fat in All Milk Produced	Total: Milk	Total: Milkfat
	1,000 Head	Pounds	Pounds	Percent	Million Pounds	Million Pounds
1993	9,589	15,704	575	3.66	150,582	5,514.4
1994	9,500	16,175	592	3.66	153,664	5,623.7
1995	9,461	16,451	602	3.66	155,644	5,694.3

1/ Average number during year, excluding heifers not yet fresh.
2/ Excludes milk sucked by calves.

Table Extraction from Government Reports

[Pinto, McCallum, Wei, Croft, 2003 SIGIR]

100+ documents from www.fedstats.gov



Labels:

- Non-Table
- Table Title
- Table Header
- Table Data Row
- Table Section Data Row
- Table Footnote
- ... *(12 in all)*

Features:

- Percentage of digit chars
- Percentage of alpha chars
- Indented
- Contains 5+ consecutive spaces
- Whitespace in this line aligns with prev.
- ...
- Conjunctions of all previous features, time offset: {0,0}, {-1,0}, {0,1}, {1,2}.

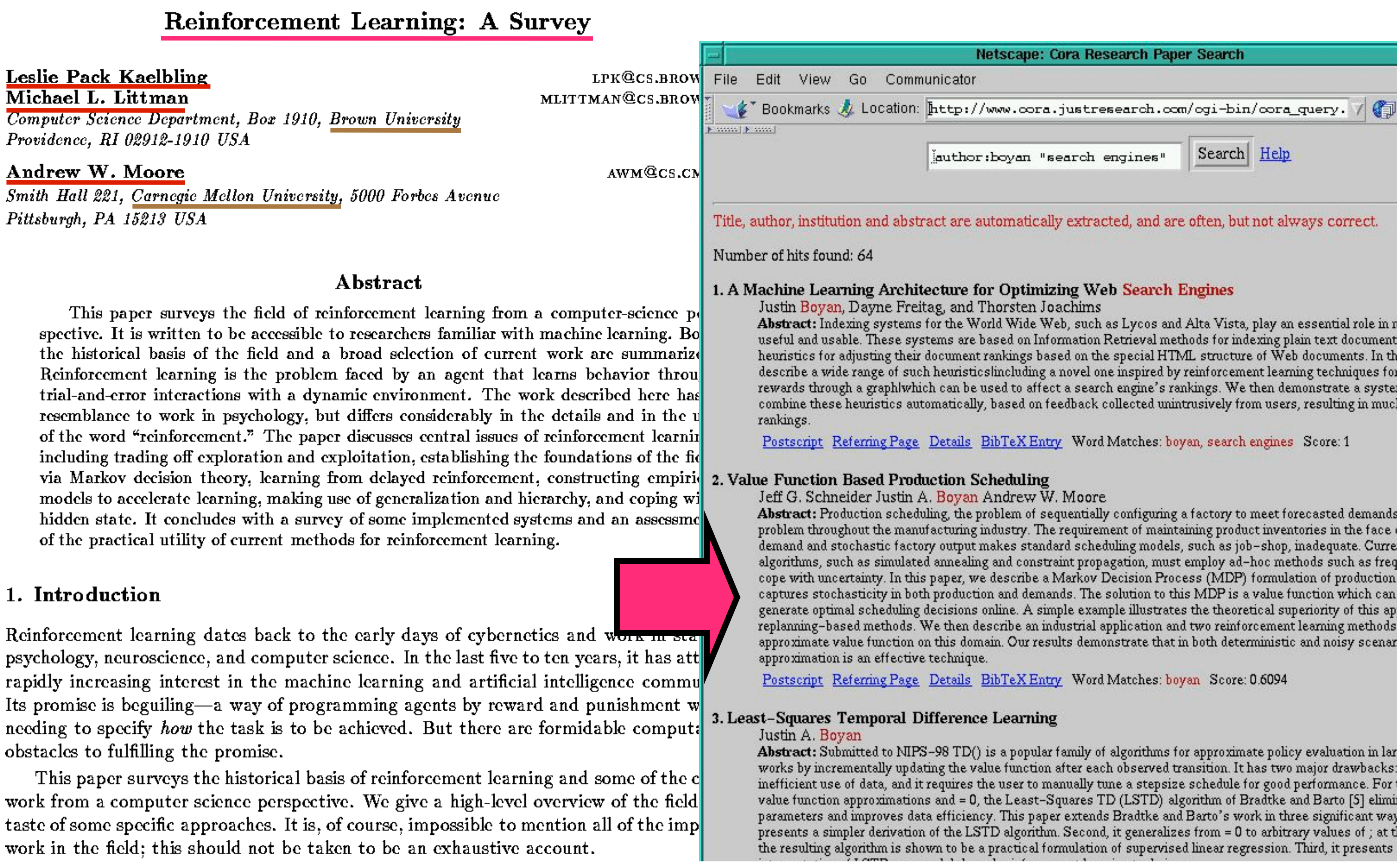
Table Extraction Experimental Results

[Pinto, McCallum, Wei, Croft, 2003 SIGIR]

	Line labels, percent correct	Table segments, F1
HMM	65 %	64 %
Stateless MaxEnt	85 %	-
CRF	95 %	92 %

IE from Research Papers

[McCallum et al '99]



Reinforcement Learning: A Survey

Leslie Pack Kaelbling LPK@CS.BROW
Michael L. Littman MLITTMAN@CS.BROW
Computer Science Department, Box 1910, Brown University
Providence, RI 02912-1910 USA

Andrew W. Moore AWM@CS.CM
Smith Hall 221, Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213 USA

Abstract

This paper surveys the field of reinforcement learning from a computer-science pe spective. It is written to be accessible to researchers familiar with machine learning. Bo the historical basis of the field and a broad selection of current work are summariz Reinforcement learning is the problem faced by an agent that learns behavior throu trial-and-error interactions with a dynamic environment. The work described here has resemblance to work in psychology, but differs considerably in the details and in the u of the word "reinforcement." The paper discusses central issues of reinforcement learnin including trading off exploration and exploitation, establishing the foundations of the fi via Markov decision theory, learning from delayed reinforcement, constructing empiri models to accelerate learning, making use of generalization and hierarchy, and coping wi hidden state. It concludes with a survey of some implemented systems and an assessme of the practical utility of current methods for reinforcement learning.

1. Introduction

Reinforcement learning dates back to the early days of cybernetics and work in sta psychology, neuroscience, and computer science. In the last five to ten years, it has att rapidly increasing interest in the machine learning and artificial intelligence commu Its promise is beguiling—a way of programming agents by reward and punishment w needing to specify *how* the task is to be achieved. But there are formidable computa obstacles to fulfilling the promise.

This paper surveys the historical basis of reinforcement learning and some of the c work from a computer science perspective. We give a high-level overview of the field taste of some specific approaches. It is, of course, impossible to mention all of the imp work in the field; this should not be taken to be an exhaustive account.

Netscape: Cora Research Paper Search

File Edit View Go Communicator

Bookmarks Location: http://www.cora.justresearch.com/cgi-bin/cora_query.

author:boyan "search engines" Search Help

Title, author, institution and abstract are automatically extracted, and are often, but not always correct.

Number of hits found: 64

1. A Machine Learning Architecture for Optimizing Web Search Engines
Justin Boyan, Dayne Freitag, and Thorsten Joachims
Abstract: Indexing systems for the World Wide Web, such as Lycos and Alta Vista, play an essential role in m useful and usable. These systems are based on Information Retrieval methods for indexing plain text document heuristics for adjusting their document rankings based on the special HTML structure of Web documents. In th describe a wide range of such heuristicslincluding a novel one inspired by reinforcement learning techniques fo rewards through a graphlwhich can be used to affect a search engine's rankings. We then demonstrate a syste combine these heuristics automatically, based on feedback collected unintrusively from users, resulting in muc rankings.
Postscript Referring Page Details BibTeX Entry Word Matches: boyan, search engines Score: 1

2. Value Function Based Production Scheduling
Jeff G. Schneider Justin A. Boyan Andrew W. Moore
Abstract: Production scheduling, the problem of sequentially configuring a factory to meet forecasted demands problem throughout the manufacturing industry. The requirement of maintaining product inventories in the face demand and stochastic factory output makes standard scheduling models, such as job-shop, inadequate. Curre algorithms, such as simulated annealing and constraint propagation, must employ ad-hoc methods such as freq cope with uncertainty. In this paper, we describe a Markov Decision Process (MDP) formulation of production captures stochasticity in both production and demands. The solution to this MDP is a value function which can generate optimal scheduling decisions online. A simple example illustrates the theoretical superiority of this ap replanning-based methods. We then describe an industrial application and two reinforcement learning methods approximate value function on this domain. Our results demonstrate that in both deterministic and noisy scenar approximation is an effective technique.
Postscript Referring Page Details BibTeX Entry Word Matches: boyan Score: 0.6094

3. Least-Squares Temporal Difference Learning
Justin A. Boyan
Abstract: Submitted to NIPS-98 TD() is a popular family of algorithms for approximate policy evaluation in lar works by incrementally updating the value function after each observed transition. It has two major drawbacks: inefficient use of data, and it requires the user to manually tune a stepsize schedule for good performance. For value function approximations and = 0, the Least-Squares TD (LSTD) algorithm of Bradtke and Barto [5] elimi parameters and improves data efficiency. This paper extends Bradtke and Barto's work in three significant way presents a simpler derivation of the LSTD algorithm. Second, it generalizes from = 0 to arbitrary values of ; at t the resulting algorithm is shown to be a practical formulation of supervised linear regression. Third, it presents

IE from Research Papers

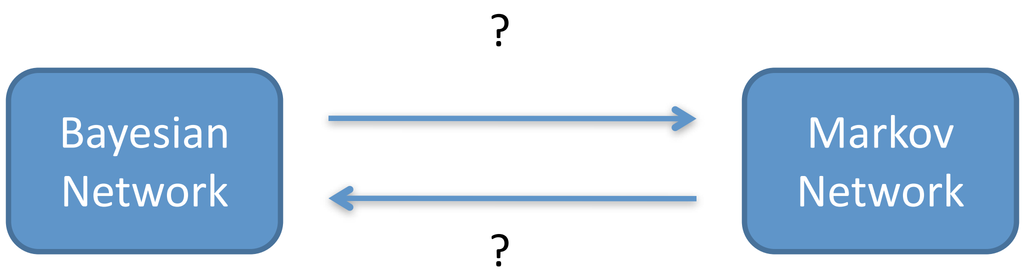
	Field-level F1
Hidden Markov Models (HMMs) *[Seymore, McCallum, Rosenfeld, 1999]*	**75.6**
Support Vector Machines (SVMs) *[Han, Giles, et al, 2003]*	**89.7**
Conditional Random Fields (CRFs) *[Peng, McCallum, 2004]*	**93.9**

Δ error 40%

When to use a *directed* or *undirected* model?

Directed	Undirected
	Increasingly popular in NLP and Vision
• Captures inter-causal reasoning, *eg* explaining away	• Captures "affinity" Symmetrical. Cyclical graphs
• Parameters interpretable, can be set by hand.	• Param's not so interpretable, usually learned from data
• Usually easier parameter estimation	• Trickier parameter estimation, but not too bad
• Can easily generate data from the model	• Can easily add factors & overlapping features to the model
• Rich existing work in latent-variable models	• Less work in latent-variable models, but there is some

Transforming Between Directed and Undirected Models


?
Bayesian Network
Markov Network
?

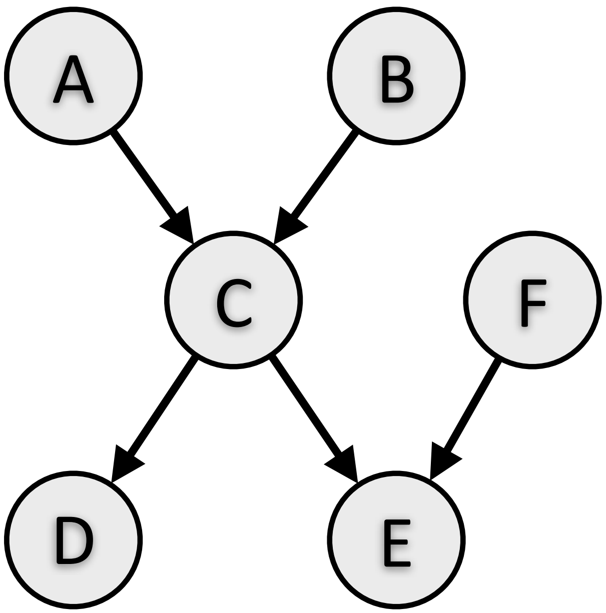
Bayesian Network to Gibbs Distribution

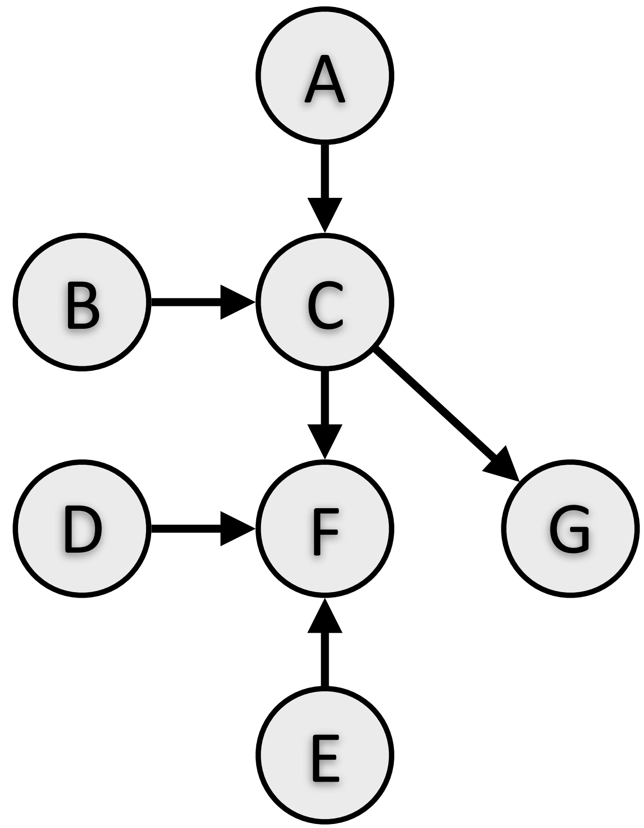
- Each conditional probability distribution is a factor. Trivial !
- Also works when conditioning on some evidence.

- Can we go from a Bayesian Network to an undirected graph that's an I-map?

Ask about example on the board

Example





Intuition

- In the Markov network, each factor must correspond to a subset of a clique.
- The "factors" in Bayesian networks are the CPDs.
  - Node + parents
- **Moralize** the graph: add an edge between any two nodes that share a child
- Moralizing ensures that a node and its parents form a clique.
  - But some independencies in the Bayesian network graph may be lost in the Markov network graph.

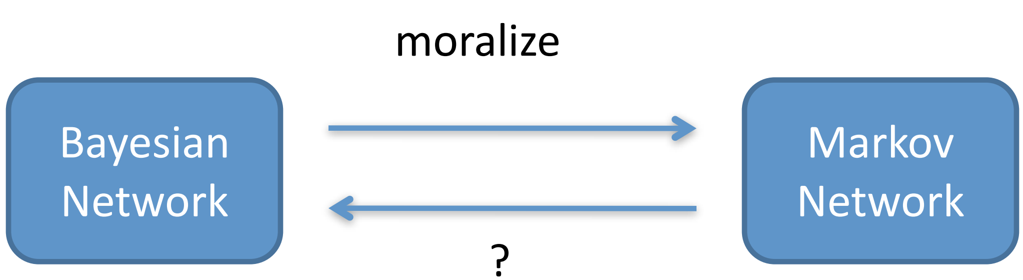
Bayesian Network Structure to Markov Network Structure

Recipe

- Start with the Bayesian network skeleton of $\mathcal{G}$.
- **Moralize** the graph: add an edge between any two nodes that share a child.

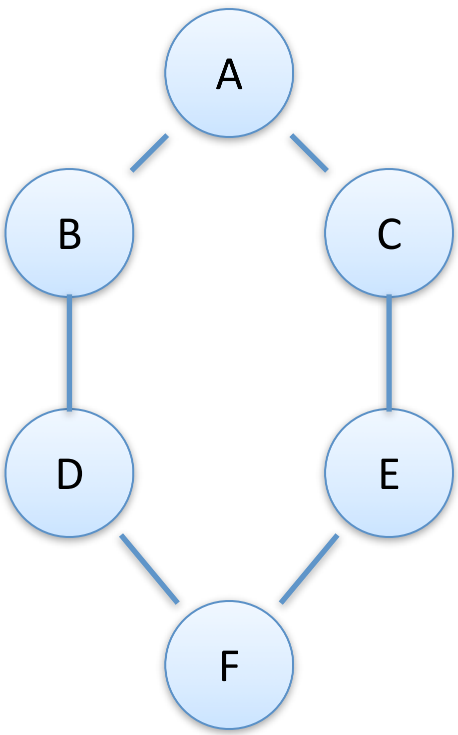
- Result: moralized (undirected) graph is a minimal I-map for $\mathcal{G}$.
  - If $\mathcal{G}$ was moral already, P-map.

You should know how to perform this conversion directed -> undirected.


moralize
Bayesian Network
Markov Network
?

Markov Network to Bayesian Network

- Example: P given by a Markov network.



How do we build BN I-maps in general?

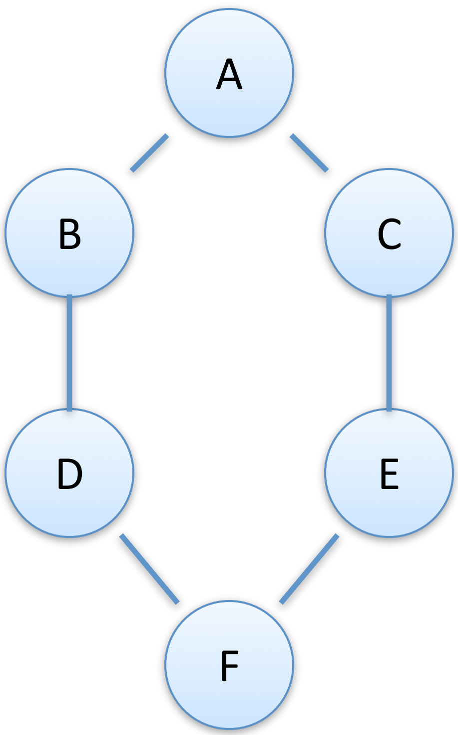
Building a Minimal I-Map

- Order variables arbitrarily,
  so that X_i precedes all its descendants.
- For i from 1 to n:
  - Add X_i to the network
  - Let **Parents**(X_i) be the minimal subset **S** of $\{X_1, \ldots, X_{i-1}\}$ such that
    $X_i \perp (\{X_1, \ldots, X_{i-1}\} \setminus \mathbf{S}) \mid \mathbf{S}$

Lecture 2!

Markov Network to Bayesian Network

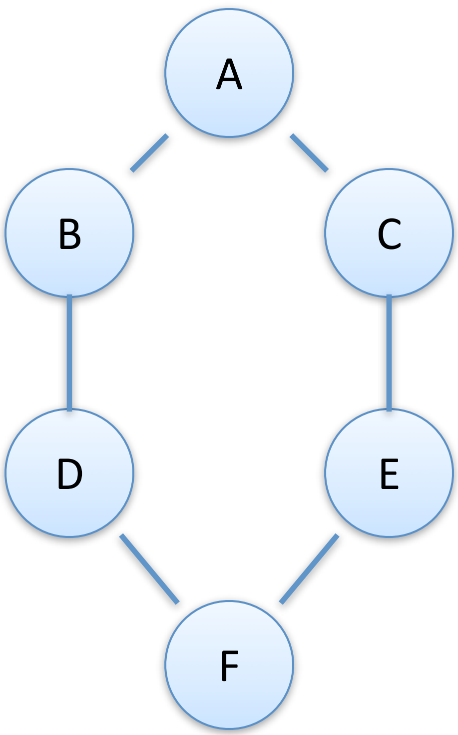
- Example: P given by a Markov network.



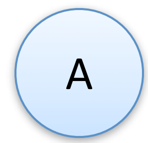
{A, B, C, D, E, F}

Markov Network to Bayesian Network

- Example: P given by a Markov network.

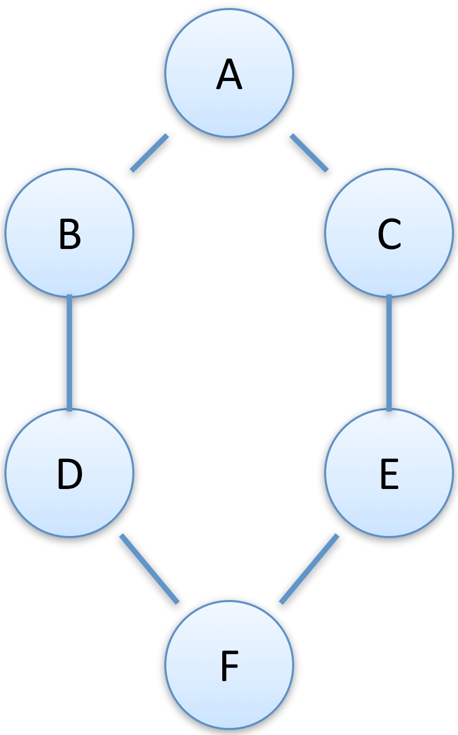


{A, B, C, D, E, F}

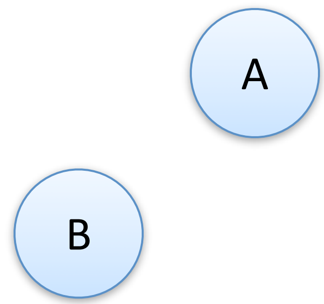


Markov Network to Bayesian Network

- Example: P given by a Markov network.

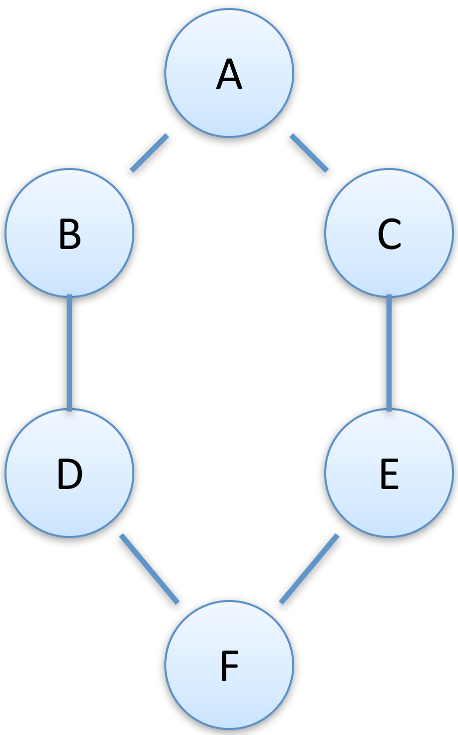


{A, B, C, D, E, F}

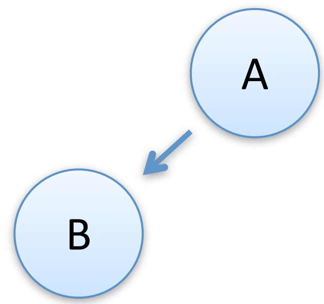


Markov Network to Bayesian Network

- Example: P given by a Markov network.

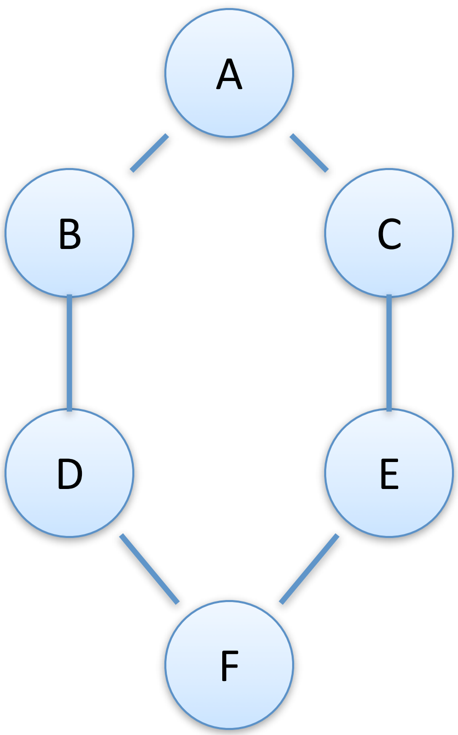


{A, B, C, D, E, F}

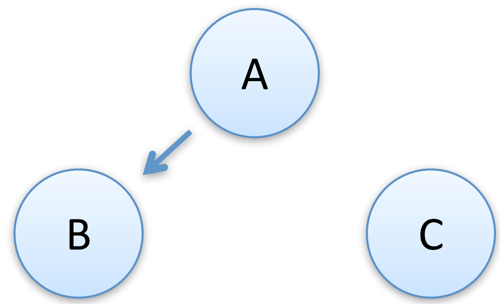


Markov Network to Bayesian Network

- Example: P given by a Markov network.

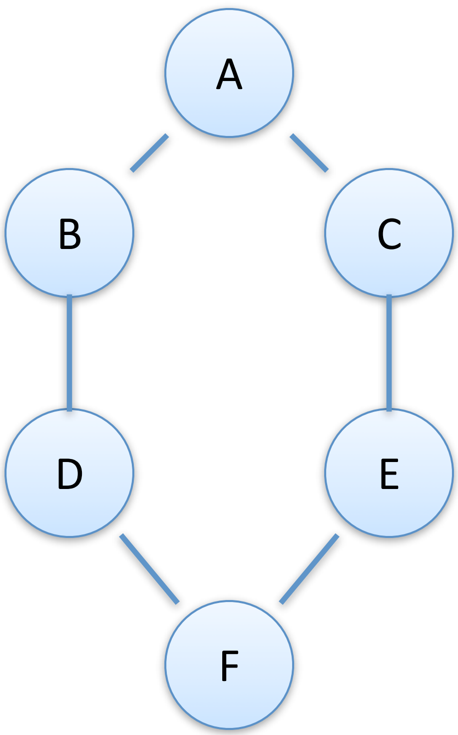


{A, B, C, D, E, F}

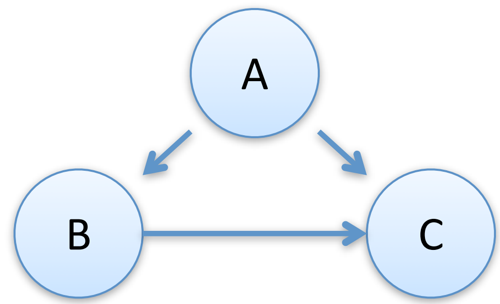


Markov Network to Bayesian Network

- Example: P given by a Markov network.

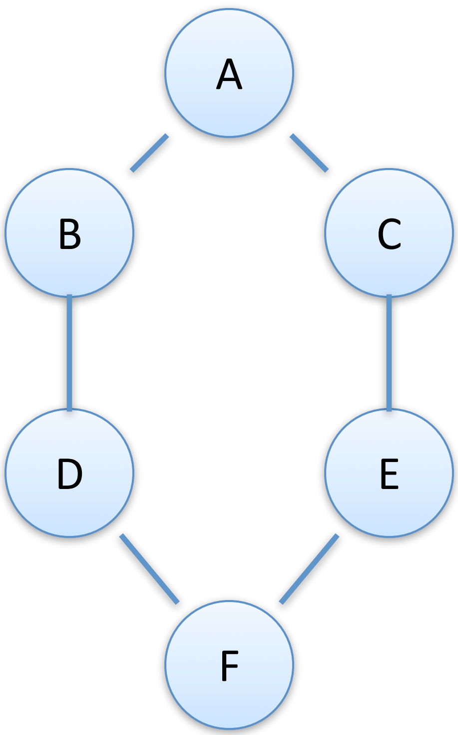


{A, B, C, D, E, F}

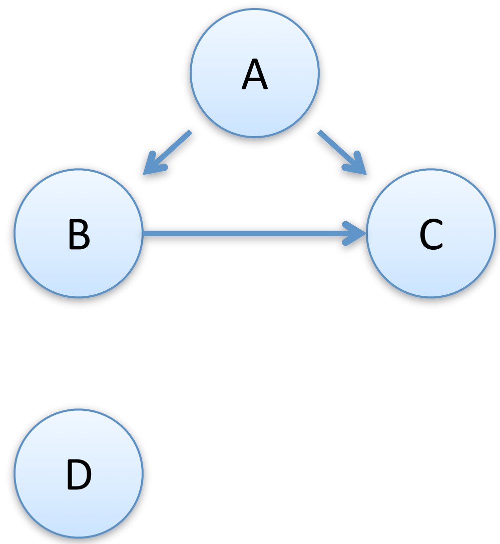


Markov Network to Bayesian Network

- Example: P given by a Markov network.

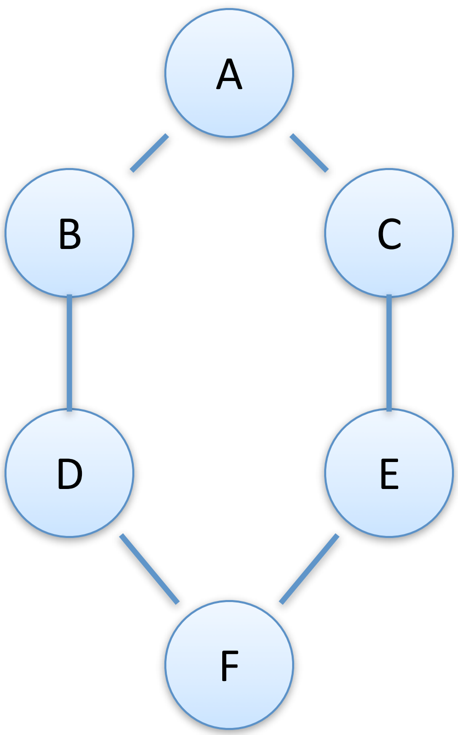


{A, B, C, D, E, F}

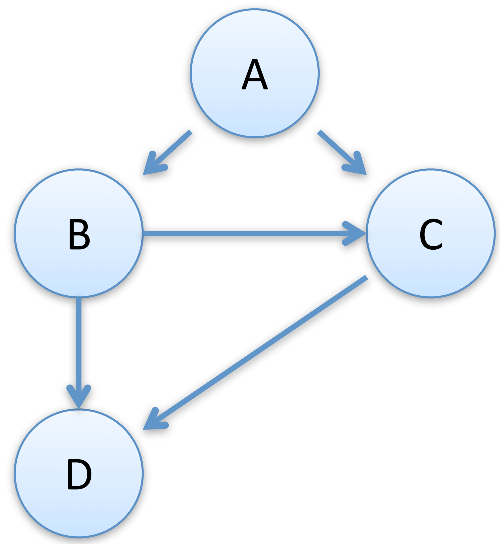


Markov Network to Bayesian Network

- Example: P given by a Markov network.

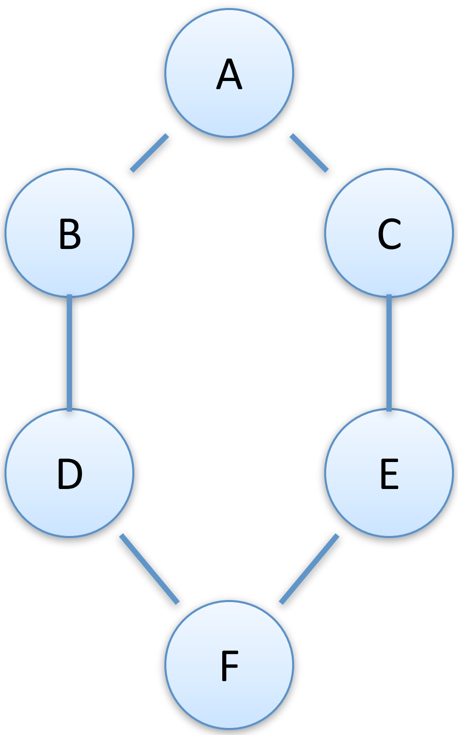


{A, B, C, D, E, F}

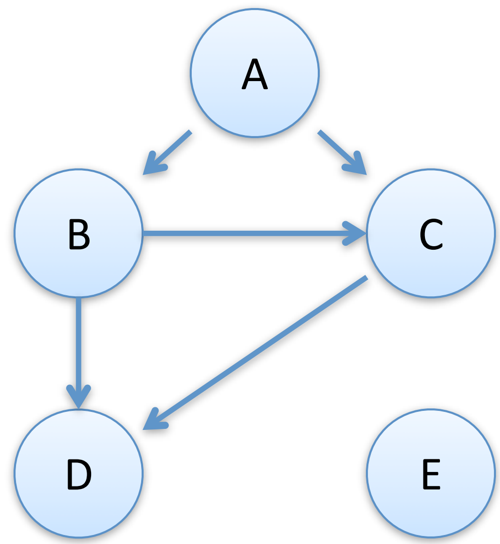


Markov Network to Bayesian Network

- Example: P given by a Markov network.

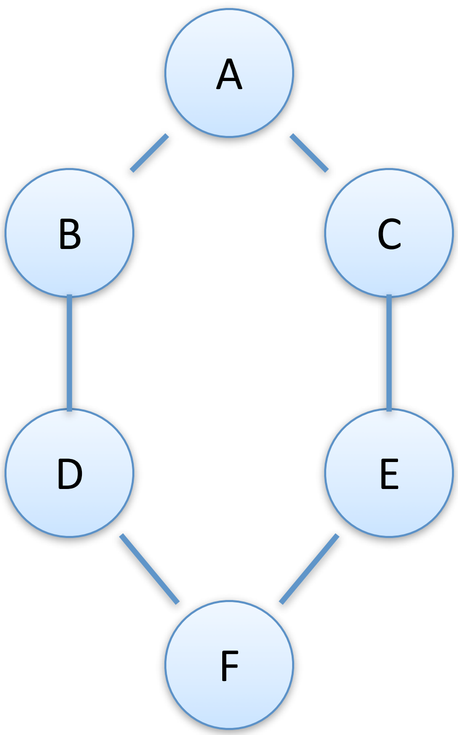


{A, B, C, D, E, F}

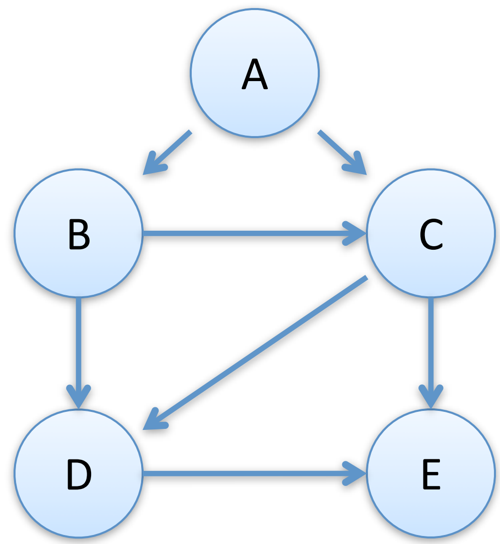


Markov Network to Bayesian Network

- Example: P given by a Markov network.

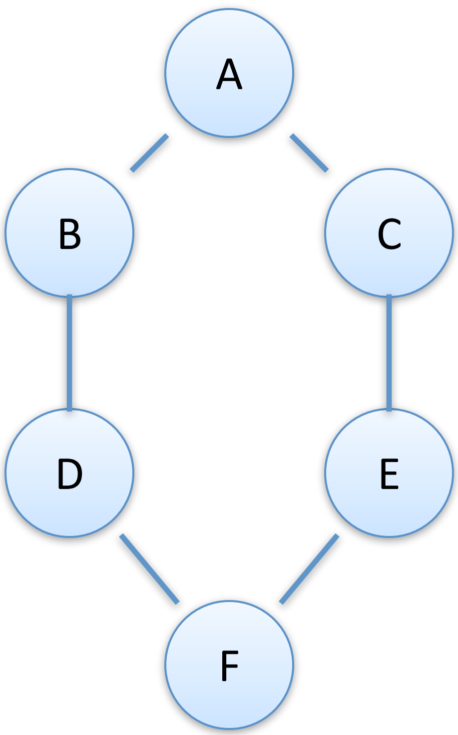


{A, B, C, D, E, F}

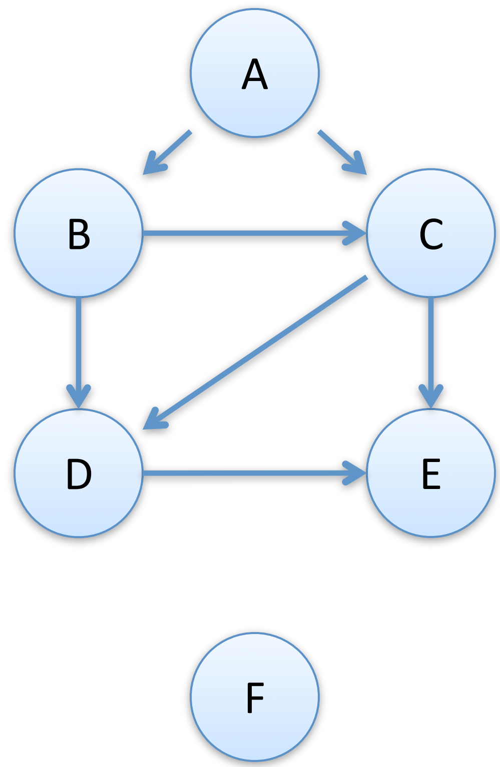


Markov Network to Bayesian Network

- Example: P given by a Markov network.

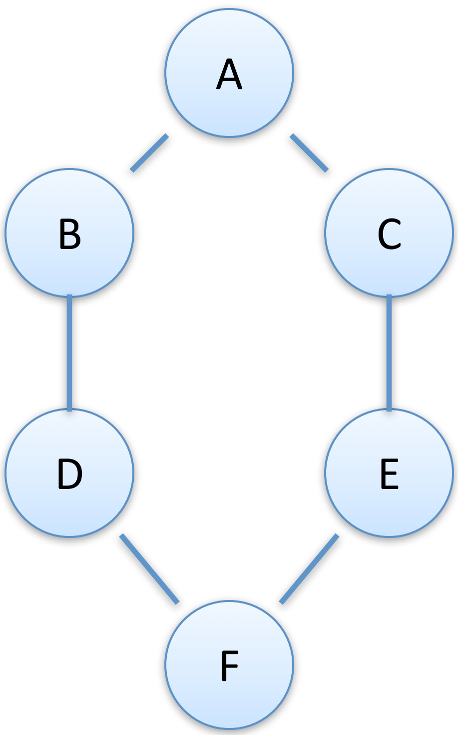


{A, B, C, D, E, F}

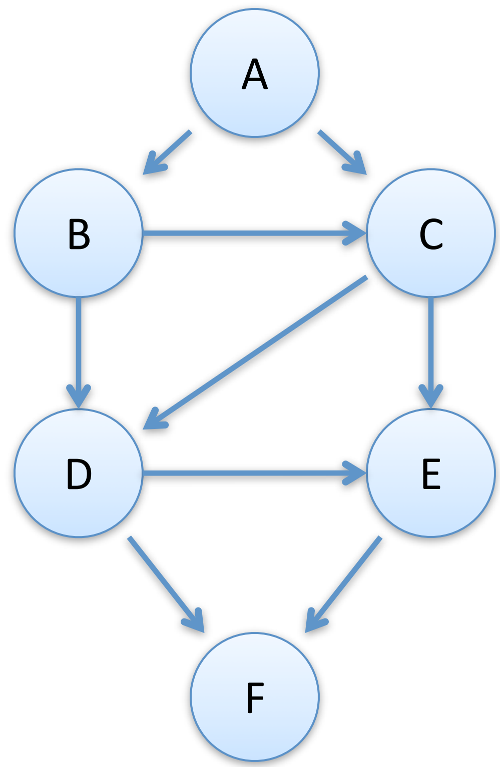


Markov Network to Bayesian Network

- Example: P given by a Markov network.



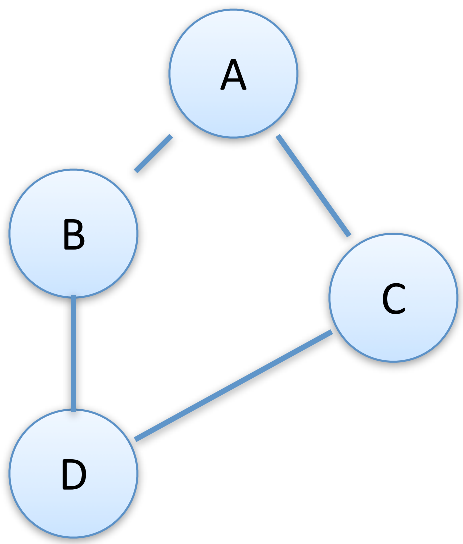
{A, B, C, D, E, F}

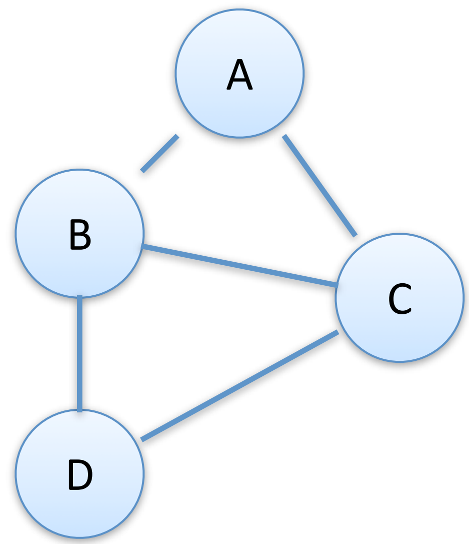


You should know how to perform this conversion undirected -> directed.

Chordal Graphs

- Undirected graph whose minimal cycles are not longer than 3.





Markov Network to Bayesian Network

- If $\mathcal{G}$ is a minimal I-map Bayesian network for Markov network $\mathcal{H}$, then $\mathcal{G}$ has no immoralities.
  - And is therefore **chordal**, since any loop of length ≥ 4 in a Bayesian network graph must have immoralities.

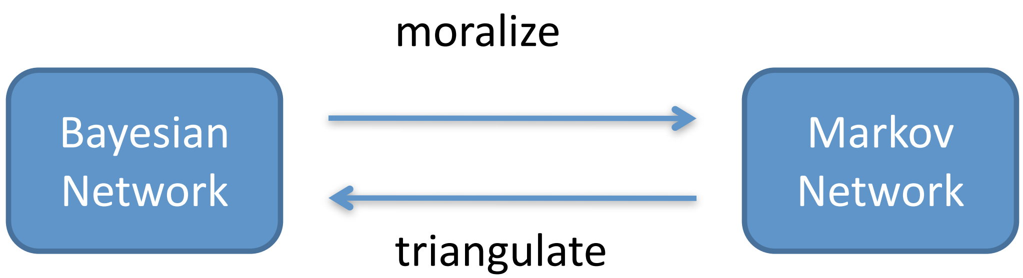
Proof in the book, but think about matching skeleton and v-structures.

- The Bayesian network we create cannot have any immoralities!

Markov Network to Bayesian Network

- Conversion from MN to BN requires *triangulation*.
  - May lose some independence information.
  - May involve a lot of additional edges.
  - Different orderings of chain rule may yield different numbers of additional edges.

Do a few more examples on the board


moralize
Bayesian Network
Markov Network
triangulate

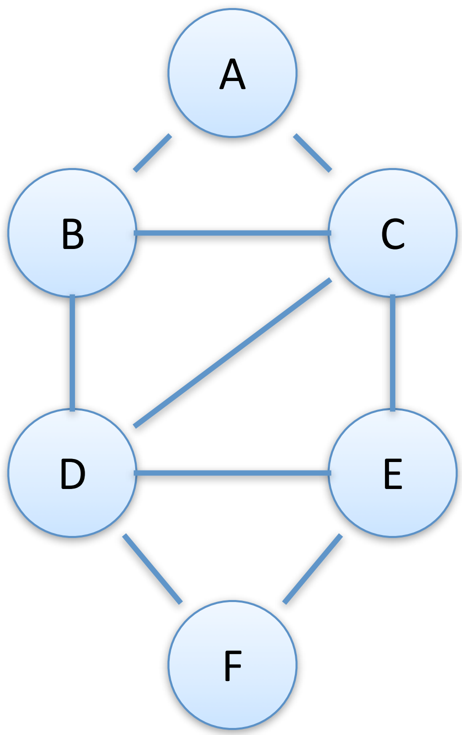
One More Formalism

- Bayesian network/Markov network conversion can lead to addition of edges and loss of independence information.
- Is there a subset of distributions that can be captured *perfectly* in both models?
  - Yes! Undirected chordal graphs.

Theorem

- If $\mathcal{H}$ (a Markov network) is non-chordal, then there is no Bayesian network $\mathcal{G}$ such that I($\mathcal{G}$) = I($\mathcal{H}$), i.e., no P-map.
- Why? Minimal I-map for $\mathcal{G}$ must be chordal. If $\mathcal{G}$ is an I-map for $\mathcal{H}$, it must include some additional edges not in $\mathcal{H}$, but that eliminates independence assumptions. So I($\mathcal{H}$) can't be perfectly encoded.

Clique Tree

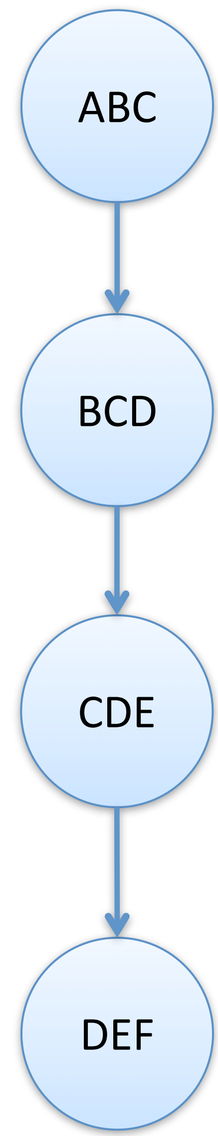


Every maximal clique becomes a vertex.

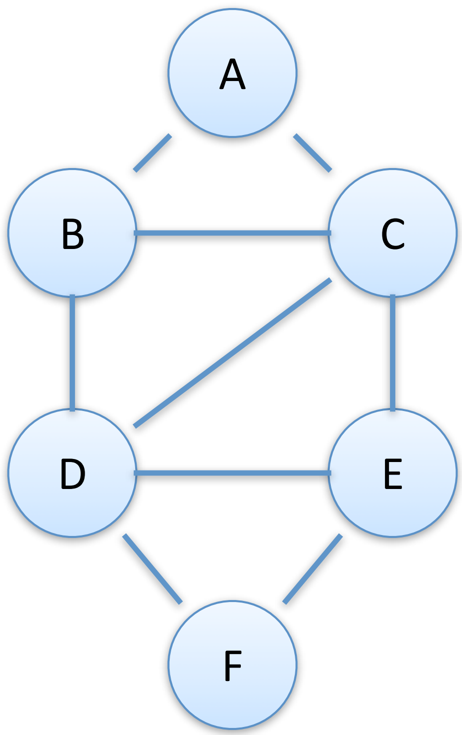
Connect vertices with overlapping variables

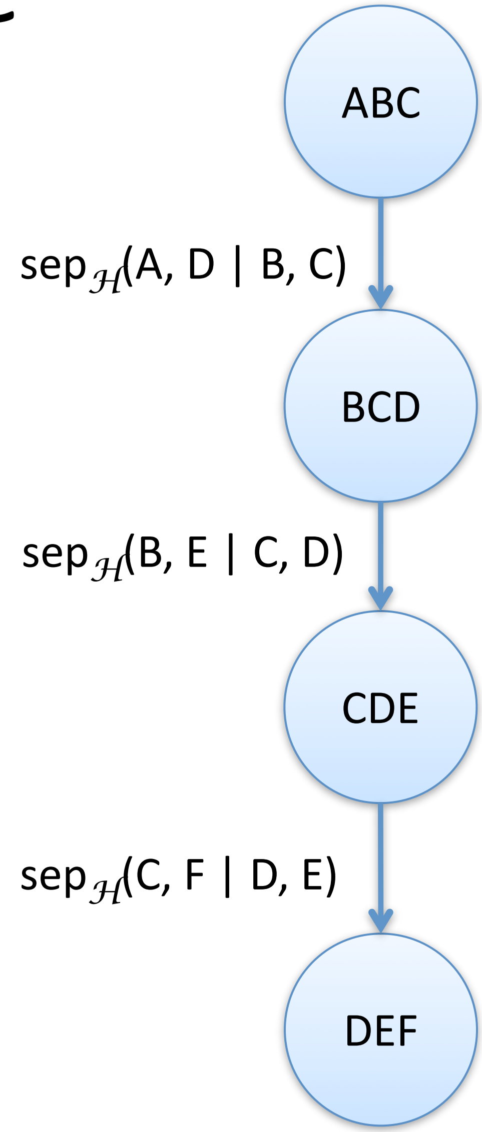
Tree structure?

then "Clique Tree"



Clique Tree





For each edge, intersection of r.v.s *separates* the rest in $\mathcal{H}$.

Clique Tree

- Does a clique tree exist?
  - Yes, if the undirected graph $\mathcal{H}$ is chordal!
  - Construction: inductive proof (K&F 4.5.3)
  - We will return to this later.

Work out example of
non-chordal graph that doesn't provide a clique tree

Clique Tree

- Does a clique tree exist?
  - Yes, if the undirected graph $\mathcal{H}$ is chordal!

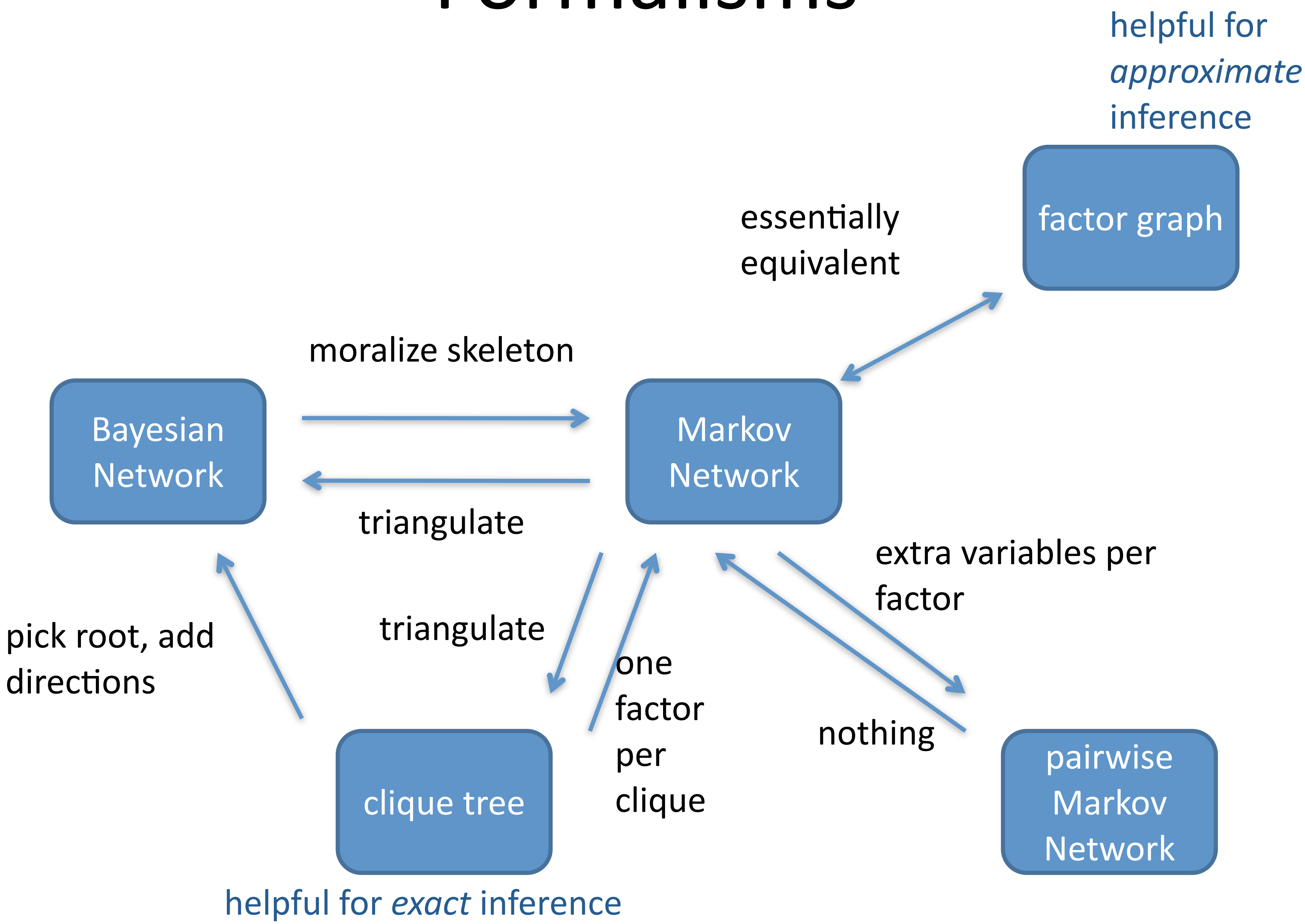
- Result: If undirected graph $\mathcal{H}$ is chordal, then there is a Bayesian network structure $\mathcal{G}$ that is a P-map for $\mathcal{H}$.
  - Need: Markov network to clique tree (above), clique tree to Bayesian network.

Chordal Markov Network to Bayesian Network

- Transform chordal graph into clique tree.
- Arbitrarily pick root node, and topologically order cliques from there.
- Build minimal I-map (lecture 4).
  - Clique tree makes independence tests easy.

- Can then show that $\mathcal{G}$ and $\mathcal{H}$ have the same set of edges.
- $\mathcal{G}$ is moral, so they are P-maps for each other.

Formalisms


helpful for *approximate* inference
essentially equivalent
factor graph
moralize skeleton
Bayesian Network
Markov Network
triangulate
extra variables per factor
pick root, add directions
triangulate
one factor per clique
nothing
clique tree
pairwise Markov Network
helpful for *exact* inference