Machine Learning

4771

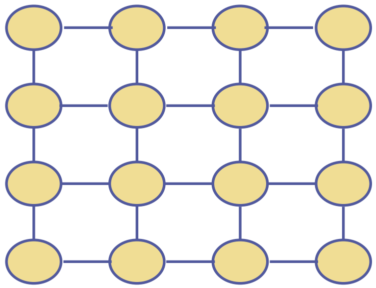
Instructor: Tony Jebara

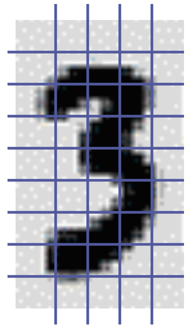
Topic 16

- Undirected Graphs
- Undirected Separation
- Inferring Marginals & Conditionals
- Moralization
- Junction Trees
- Triangulation

Undirected Graphs

- Separation is *much easier* for undirected graphs
- But, what are undirected graphs and why use them?
- Might be hard to call vars parent/child or cause/effect
- Example: Image pixels
- Each pixel is Bernoulli = {0,1}
- Where 0=dark, 1=bright

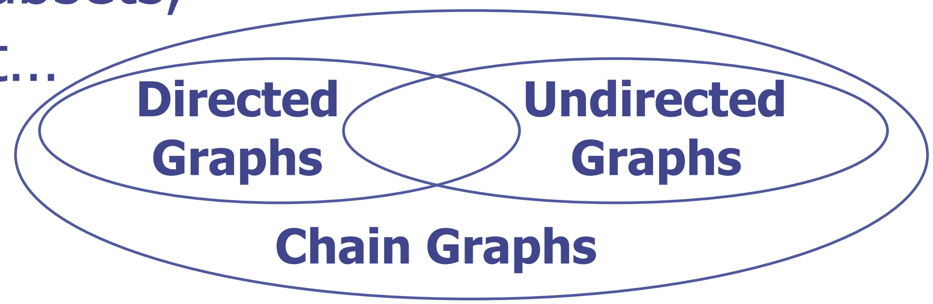




- Have probability over all pixels $p\left(x_{11},\ldots,x_{1M},\ldots,x_{M1},\ldots,x_{MM}\right)$
- Bright pixels have Bright neighbors
- Nearby pixels dependent, so connect with links
- Get a graphical model that looks like a grid
- But who is parent? No parents really, just probability
- Grid models are called Markov Random Fields
- Used in vision, physics (lattice, spin, or Ising models), etc.

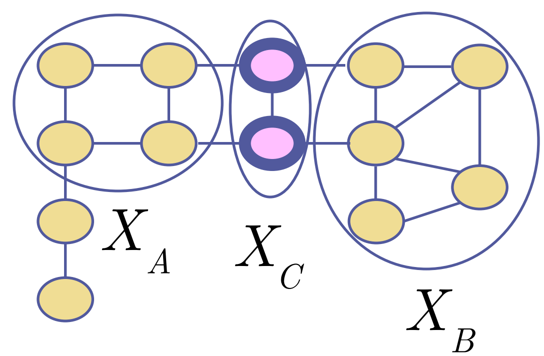
Undirected Graphs

- Undirected & directed not subsets,
- Chain Graphs are a superset…
- Some distributions behave as undirected graphs, some as directed, some as both

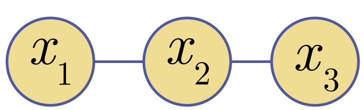


- Undirected graphs use the standard definition of separation:

  an undirected graph says that $p\left(x_1,\ldots,x_M\right)$ satisfies any statement $X_A \perp\!\!\!\perp X_B \mid X_C$ if no paths can go from X_A to X_B unless they go through X_C



- Thus, undirected graphs obey the general Markov property
- Recall the simple Markov property



$$x_1 \perp\!\!\!\perp x_3 \mid x_2 \Rightarrow p\left(x_1 \mid x_2, x_3\right) = p\left(x_1 \mid x_2\right)$$

Hammersley Clifford Theorem

Theorem[HC]: any distribution that obeys the Markov property

$$p\left(x_i \mid X_{U\setminus i}\right) = p\left(x_i \mid X_{Ne(i)}\right) \quad \forall\, i \in U$$

can be written as a product of terms over each maximal clique

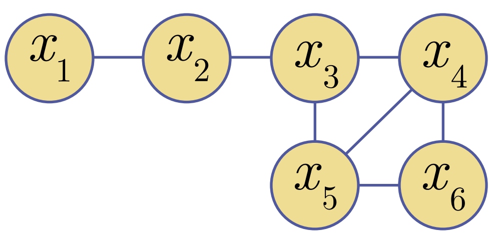
$$p\left(X_U\right) = p\left(x_1, \ldots, x_M\right) = \tfrac{1}{Z} \prod_{c \in C} \psi_c\left(X_c\right)$$

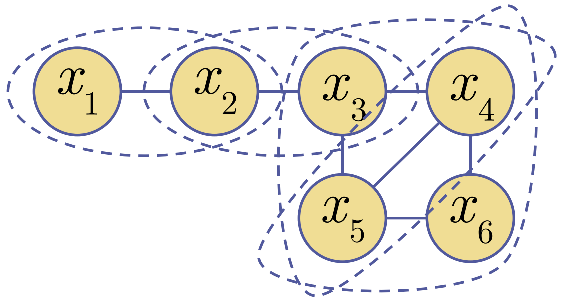
Cliques: subsets of variables that all connect to each other

Maximal: cannot add any more variables and still be a clique

Each c is a maximal clique of variables X_c in the graph

C is the set of all maximal cliques





Undirected Graph Functions

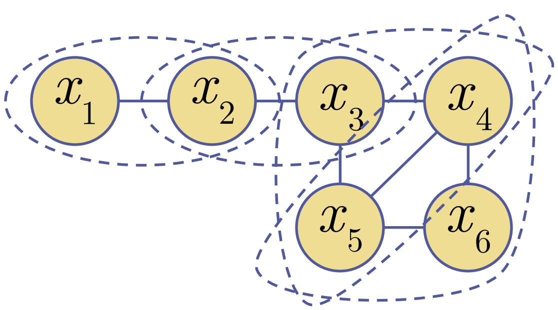
- Probability for undirected factorizes as a product of mini non-negative Potential Functions over cliques in the graph

$$\boxed{p\left(X\right) = p\left(x_1,\ldots,x_M\right) = \tfrac{1}{Z}\prod\nolimits_{c\in C}\psi_c\left(X_c\right)}$$

- Normalizing term $Z = \sum\nolimits_X \prod\nolimits_{c\in C}\psi_c\left(X_c\right)$ makes p(X) sum to 1
- Potentials ψ are non-negative un-normalized functions over cliques (subgroups of fully inter-connected variables)

- Only maximal cliques since smaller ψ absorb into larger ψ

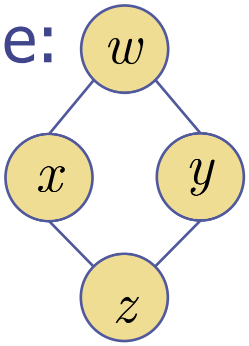
$$\psi\left(x_2,x_3\right)\psi\left(x_2\right) \rightarrow \psi\left(x_2,x_3\right) = \begin{bmatrix} 1 & 2 \\ 5 & 0 \end{bmatrix}$$



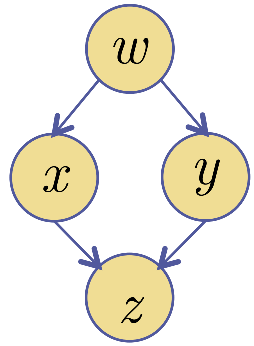
$$p\left(X\right) = \tfrac{1}{Z}\psi\left(x_1,x_2\right)\psi\left(x_2,x_3\right)\psi\left(x_3,x_4,x_5\right)\psi\left(x_4,x_5,x_6\right)$$

Undirected Separation Examples

- Example:



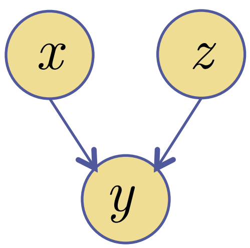
$$x \perp\!\!\!\perp y \mid \{w,z\}$$
$$w \perp\!\!\!\perp z \mid \{x,y\}$$



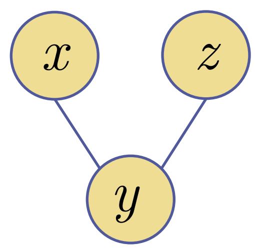
$$x \perp\!\!\!\perp y \mid \{w\}$$
$$x \cancel{\perp\!\!\!\perp} y \mid \{w,z\}$$

Directed can't do it!
Must be acyclic
Will have at least one V structure and ball goes through

- Example:

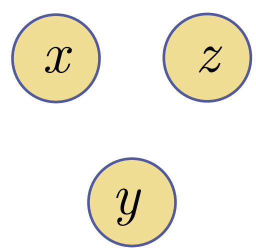


$$x \perp\!\!\!\perp z$$
$$x \cancel{\perp\!\!\!\perp} z \mid y$$



Undirected can't do it!

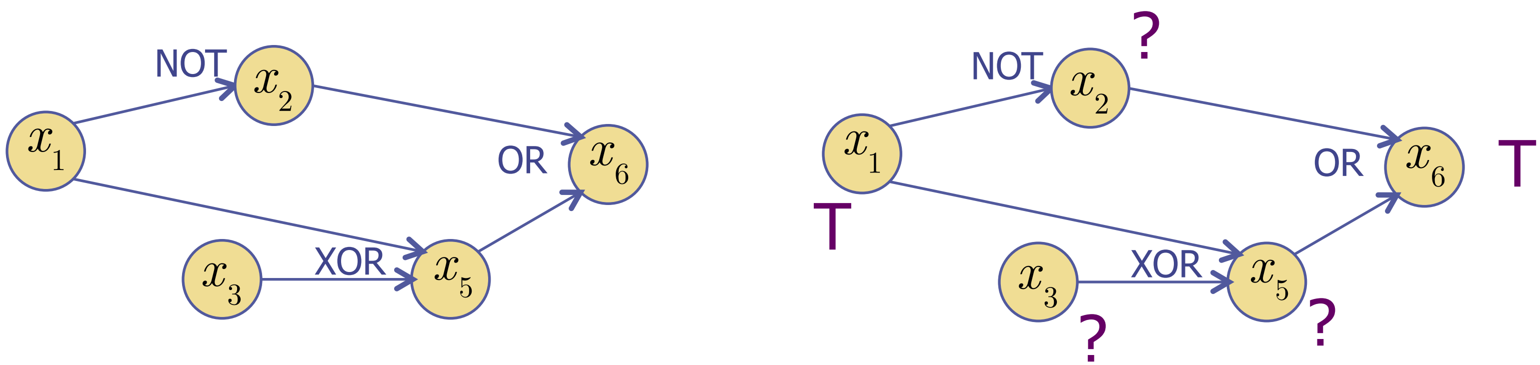
$$x \perp\!\!\!\perp z \mid y$$



$$x \perp\!\!\!\perp z$$

Logical Inference

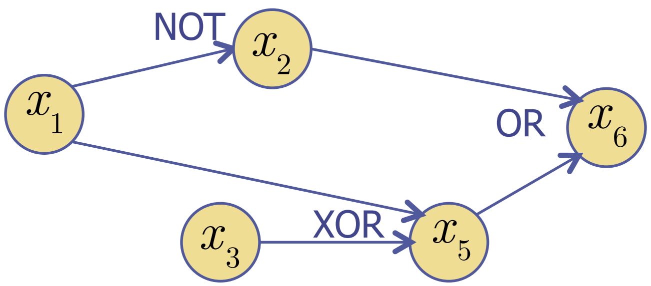
- Classic logic network: nodes are binary
- Arrows represent AND, OR, XOR, NAND, NOR, NOT etc.
- Inference: given observed binary variables, predict others



- Problems: uncertainty, conflicts and inconsistency
- Could get x_3=T and x_3=F following two different paths
- We need a way to enforce consistency and combine conflicting statements via probabilities and Bayes rule!

Probabilistic Inference

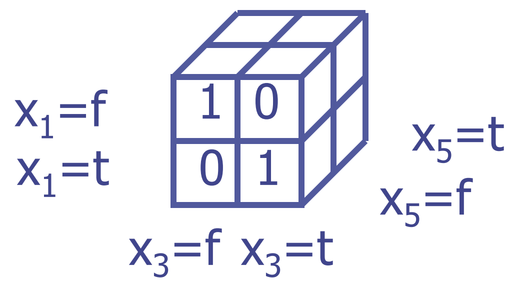
- Replace logic network with Bayesian network
- Tables represent AND, OR, XOR, NAND, NOR, NOT etc.
- Probabilistic Inference: given observed binary variables, predict marginals over others



NOT

	x_3=f	x_3=t
x_1=f	0	1
x_1=t	1	0

XOR



- Can also have soft versions of the functions

soft NOT

	x_3=f	x_3=t
x_1=f	.1	.9
x_1=t	.9	.1

Probabilistic Inference

- Two types of inference with a probability distribution:

$$p(X) = p(x_1, \ldots, x_M) \;\; \textit{with queries}\; X_F \subseteq X \;\; \textit{given evidence}\; X_E \subseteq X$$

- Marginal Inference:

$$p(X_F | X_E) = p(X_F, X_E) \big/ p(X_E) = \sum_{X \setminus X_F \cup X_E} p(X) \Big/ \sum_{X \setminus X_E} p(X)$$

or... $p(x_i | X_E) \;\; \forall \; x_i \in X_F$

- Maximum a posteriori (MAP) inference:

$$\arg\max_{X_F} \; p(X_F | X_E)$$

...for now we focus on marginal inference

Traditional Marginal Inference

- Marginal inference problem: given graph and probability function $p(X) = p(x_1,\ldots,x_M)$ for any subsets of variables find

$$p(X_F | X_E) = {p(X_F, X_E)} / {p(X_E)}$$

- So, we basically compute both marginals and divide
- But finding marginals can take exponential work!
- A problem for both directed & undirected graphs:

$$p(x_j, x_k) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} \prod_{i=1}^{M} p(x_i | \pi_i)$$

$$p(x_j, x_k) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_M} \tfrac{1}{Z} \prod_{c \in C} \psi_c(X_c)$$

- Graphs gave efficient storage, learning, Bayes Ball…
- Graphs can also be used to perform efficient inference!
- Junction Tree Algorithm: method to efficiently find marginals

Traditional Marginal Inference

- Example: brute force inference on a directed graph…
- Given a directed graph structure & *filled-in* CPTs
- We would like to efficiently compute arbitrary marginals
- Or we would like to compute arbitrary conditionals

$$p(X) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(x_6 \mid x_2, x_5)$$

$$p(x_1, x_3) = p(x_1)p(x_3 \mid x_1)$$

$$p(x_1, x_6) = \sum_{x_2, x_3, x_4, x_5} p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(x_6 \mid x_2, x_5)$$

$$p(x_1 \mid x_6) = \left. \sum\nolimits_{x_2, x_3, x_4, x_5} p(X) \middle/ \sum\nolimits_{x_1, x_2, x_3, x_4, x_5} p(X) \right.$$

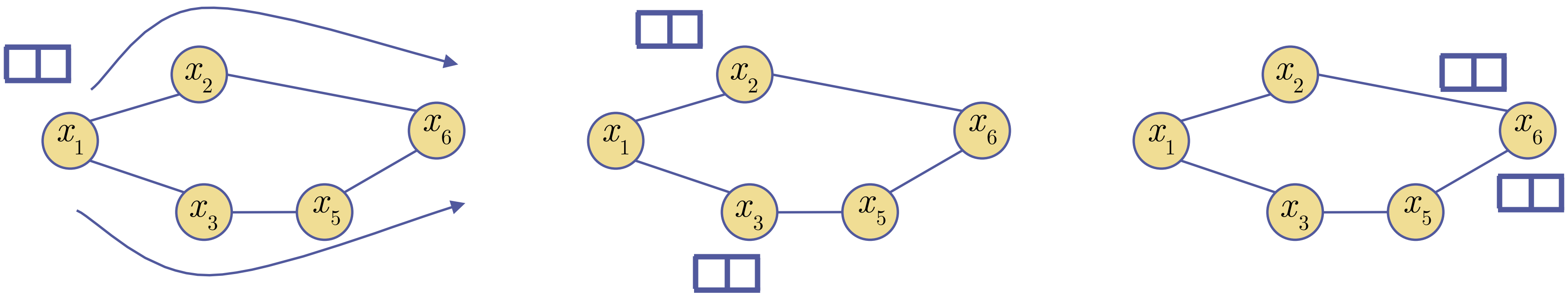
- For example, we may have some evidence, i.e. x_6=TRUE

$$p(x_1 \mid x_6 = TRUE) = \left. \sum\nolimits_{x_2, x_3, x_4, x_5} p(X_{U \setminus 6}, x_6 = TRUE) \middle/ \sum\nolimits_{x_1, x_2, x_3, x_4, x_5} p(X_{U \setminus 6}, x_6 = TRUE) \right.$$

- This is tedious & does not exploit the graph's efficiency

Efficient Marginals & Inference

- Another idea is to use some efficient graph algorithm
- Try sending messages (small tables) around the graph



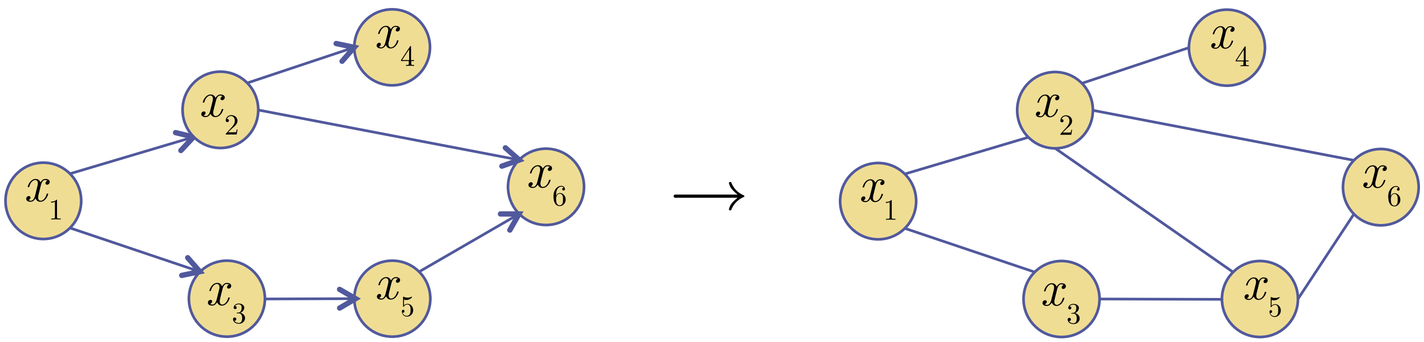
- Hopefully these somehow settle down and equal marginals

$$\hat{p}\left(x_1, x_6\right) = \sum_{x_2, x_3, x_4, x_5} p\left(X\right)$$

- AND marginals are self-consistent $\sum_{x_1} \hat{p}\left(x_1, x_6\right) = \sum_{x_2} \hat{p}\left(x_2, x_6\right)$
- Note: can't just return conditionals since they can be inconsistent $\sum_{x_1} \hat{p}\left(x_6 \mid x_1\right) \neq \sum_{x_2} \hat{p}\left(x_2 \mid x_6\right)$
- Junction Tree Algorithm must find consistent marginals

Junction Tree Algorithm

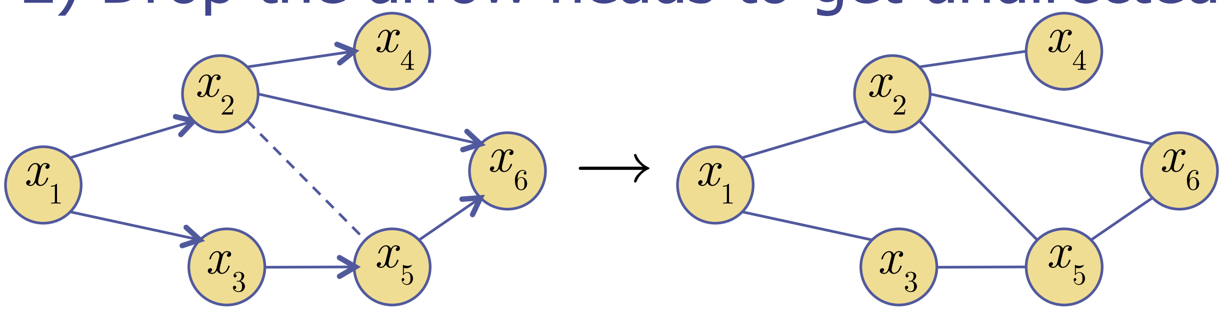
- An algorithm that achieves fast inference, by doing message passing on undirected graphs.
- We first convert a directed graph to an undirected one



- Then apply the efficient Junction Tree Algorithm:
  1) Moralization
  2) Introduce Evidence
  3) Triangulate
  4) Construct Junction Tree
  5) Propagate Probabilities (Junction Tree Algorithm)

Moralization

- Converts directed graph into undirected graph
- By moralization, marrying the parents:
  1) Connect nodes that have common children
  2) Drop the arrow heads to get undirected

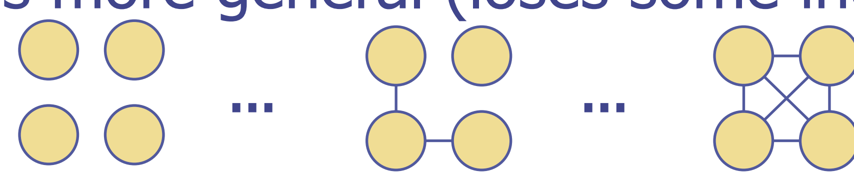


$$p(x_1)\,p(x_2 \mid x_1)\,p(x_3 \mid x_1)\,p(x_4 \mid x_2)\,p(x_5 \mid x_3)\,p(x_6 \mid x_2, x_5)$$
$$\rightarrow \quad \tfrac{1}{Z}\,\psi(x_1, x_2)\,\psi(x_1, x_3)\,\psi(x_2, x_4)\,\psi(x_3, x_5)\,\psi(x_2, x_5, x_6)$$

$$p(x_1)\,p(x_2 \mid x_1) \rightarrow \psi(x_1, x_2)$$
$$p(x_4 \mid x_2) \rightarrow \psi(x_2, x_4)$$
$$Z \rightarrow 1$$

- Note: moralization resolves *coupling* due to marginalizing
- moral graph is more general (loses some independencies)

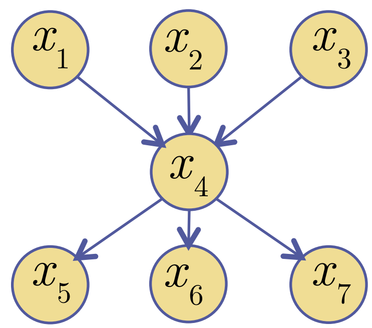
most specific



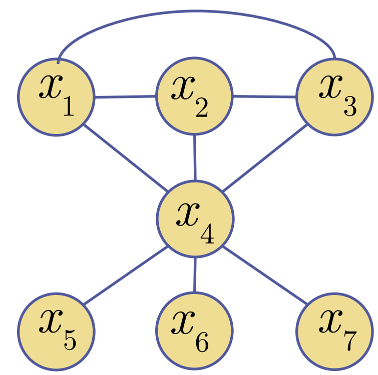
most general

Moralization

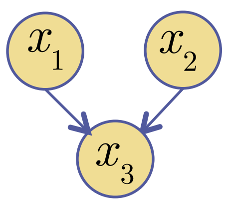
- More examples:



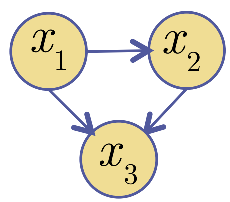
$\longrightarrow$



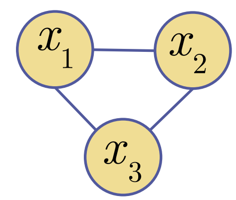
$$\rightarrow \tfrac{1}{Z}\psi\left(x_1,x_2,x_3,x_4\right)\psi\left(x_4,x_5\right)\psi\left(x_4,x_6\right)\psi\left(x_4,x_7\right)$$



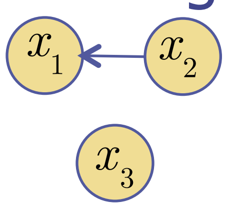
or



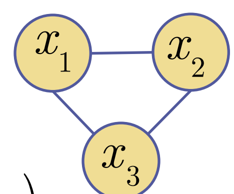
$\longrightarrow$



- More general graph less efficient but same inference:



$$\begin{aligned} p\left(x_1\right) &= \sum_{x_2,x_3} p\left(x_1,x_2,x_3\right) \\ &= \sum_{x_2,x_3} p\left(x_1 \mid x_2\right)p\left(x_2\right)p\left(x_3\right) \\ &= \sum_{x_2} p\left(x_1 \mid x_2\right)p\left(x_2\right) \end{aligned}$$



$$p\left(x_1\right) = \sum_{x_2,x_3} \tfrac{1}{Z}\psi\left(x_1,x_2,x_3\right)$$

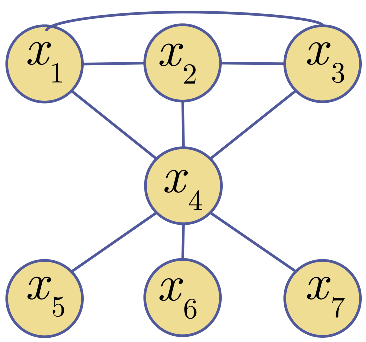
same

Introducing Evidence

- Given moral graph, note what is observed $X_E \to \bar{X}_E$

$$p\left(X_F \mid X_E = \bar{X}_E\right) \equiv p\left(X_F \mid \bar{X}_E\right)$$

- If we know this is *always* observed at $X_E \to \bar{X}_E$, simplify…
- Reduce the probability function since those X_E fixed
- Only keep probability function over remaining nodes X_F
- Only get marginals and conditionals with subsets of X_F



$$p\left(X\right) = \tfrac{1}{Z}\psi\left(x_1,x_2,x_3,x_4\right)\psi\left(x_4,x_5\right)\psi\left(x_4,x_6\right)\psi\left(x_4,x_7\right)$$

$$say\ X_E = \left\{x_3,x_4\right\} \to \bar{X}_E = \left\{\bar{x}_3,\bar{x}_4\right\}$$

Replace potential functions with slices

0.3	0.13
0.12	0.1

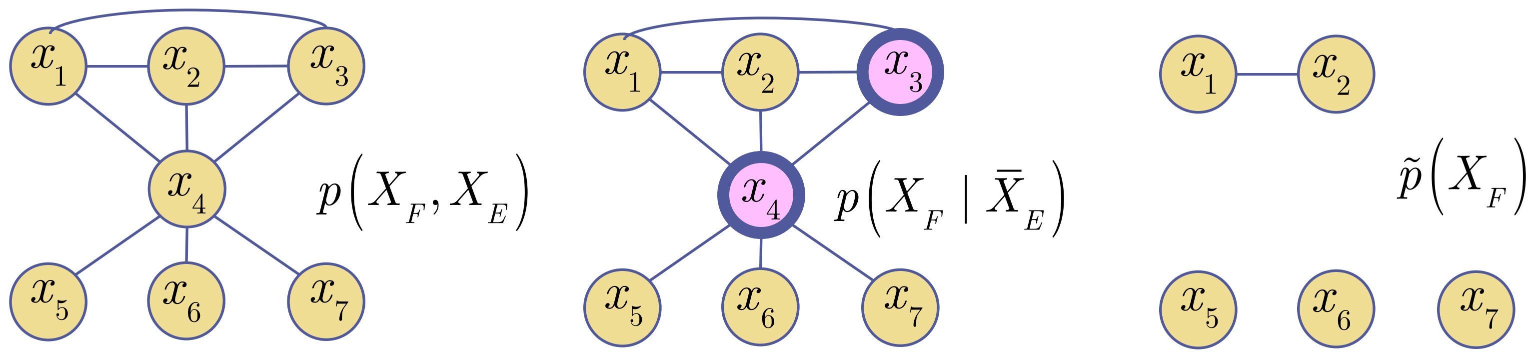
$$p\left(X_F \mid \bar{X}_E\right) \propto \tfrac{1}{Z}\psi\left(x_1,x_2,x_3 = \bar{x}_3,x_4 = \bar{x}_4\right)\psi\left(x_4 = \bar{x}_4,x_5\right)\psi\left(x_4 = \bar{x}_4,x_6\right)\psi\left(x_4 = \bar{x}_4,x_7\right)$$

$$\propto \tfrac{1}{Z}\tilde{\psi}\left(x_1,x_2\right)\tilde{\psi}\left(x_5\right)\tilde{\psi}\left(x_6\right)\tilde{\psi}\left(x_7\right)$$

But, need to recompute different normalization Z…

Introducing Evidence

- Recall undirected separation, observing X_E separates others



- But, need to recompute new normalization …

$$p(X_F \mid \bar{X}_E) \propto \frac{1}{Z} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7)$$

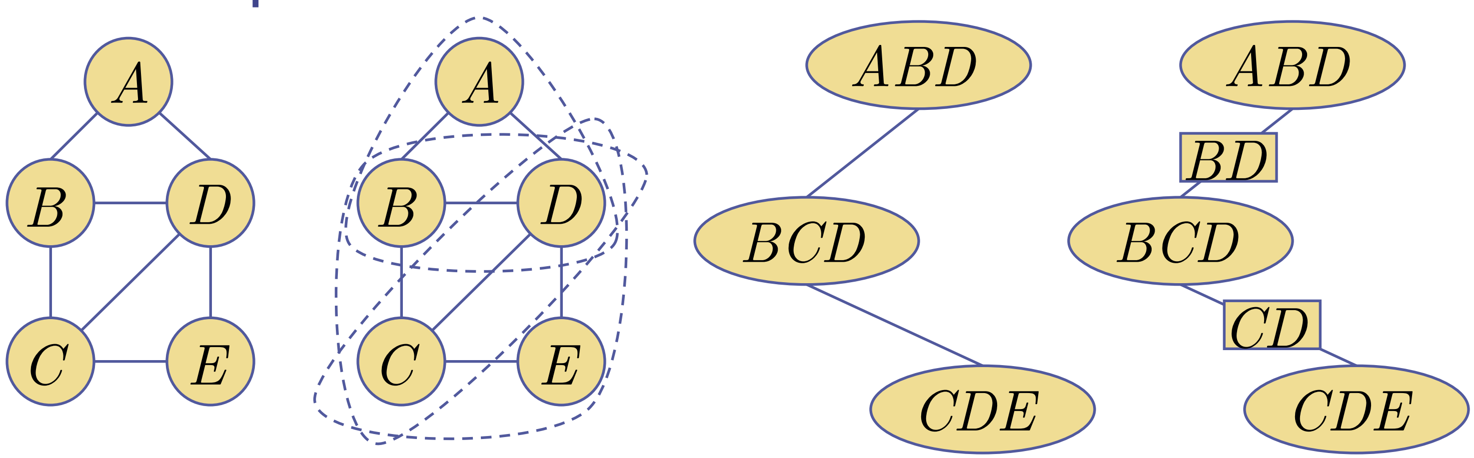
$$\tilde{p}(X_F) = \frac{1}{\tilde{Z}} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7)$$

- Just avoid Z & normalize at the end when we are querying individual marginals and conditionals as subsets of X_F

$$\tilde{p}(x_2) = \frac{\sum_{x_1, x_5, x_6, x_7} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7)}{\sum_{x_2} \sum_{x_1, x_5, x_6, x_7} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7)}$$

Junction Trees

- Given moral graph want to build Junction Tree:
  - each node is a clique (ψ) of variables in moral graph
  - edges connect cliques of the potential functions
  - unique path between nodes & root node (tree)
  - between connected clique nodes, have separators (ϕ)
  - separator nodes contain intersection of variables



$\psi(A,B,D)$

$\phi(B,D)$

$\psi(B,C,D)$

$\phi(C,D)$

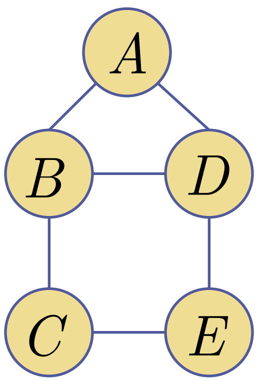
$\psi(C,D,E)$

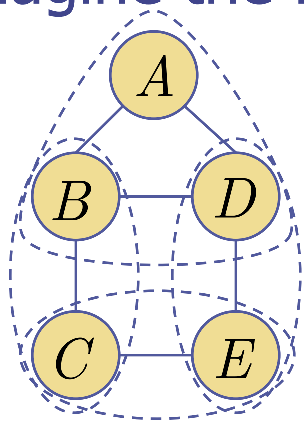
undirected **cliques** **clique tree** **junction tree**

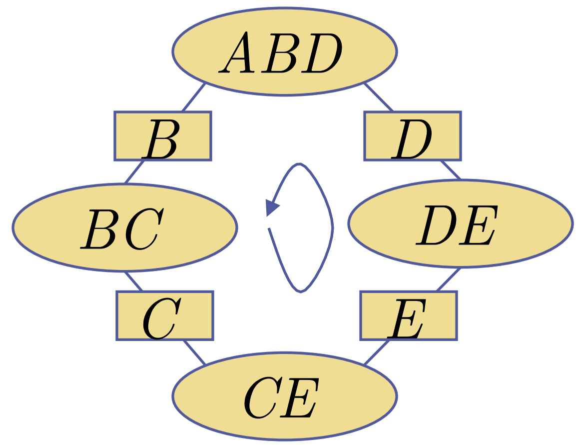
$$p(X) = \tfrac{1}{Z}\psi(A,B,D)\psi(B,C,D)\psi(C,D,E)$$

Triangulation

•Problem: imagine the following undirected graph







•Not a Tree!

•To ensure Junction Tree is a tree (no loops, etc.) before forming it must first Triangulate moral graph before finding the cliques…

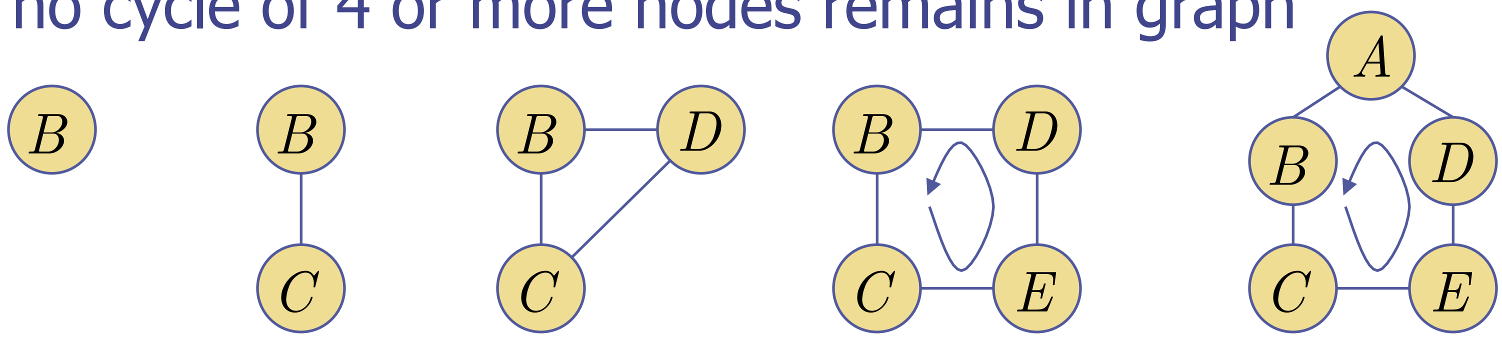
•Triangulating gives more general graph (like moralization)

•Adds links to get rid of cycles or loops

•Triangulation: Connect nodes in moral graph until no chordless cycle of 4 or more nodes remains in the graph

Triangulation

- Triangulation: Connect nodes in moral graph such that no cycle of 4 or more nodes remains in graph

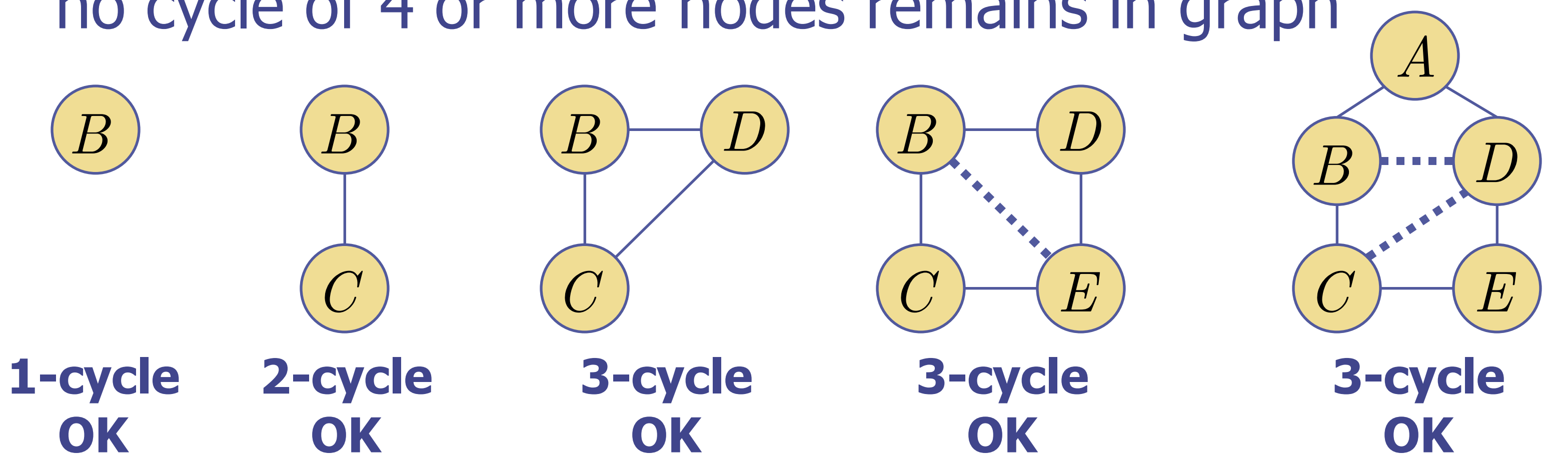


1-cycle OK **2-cycle OK** **3-cycle OK** **4-cycle BAD** **5-cycle BAD**

- So, *add links*, but many possible choices…
- HINT: Try to keep largest clique size small (makes junction tree algorithm more efficient)
- Sub-optimal triangulations of moral graph are Polynomial
- Triangulation that minimizes largest clique size is NP
- But, OK to use a suboptimal triangulation (slower JTA…)

Triangulation

- Triangulation: Connect nodes in moral graph such that no cycle of 4 or more nodes remains in graph



- So, *add links*, but many possible choices…
- HINT: Try to keep largest clique size small (makes junction tree algorithm more efficient)
- Sub-optimal triangulations of moral graph are Polynomial
- Triangulation that minimizes largest clique size is NP
- But, OK to use a suboptimal triangulation (slower JTA…)