Structured Probabilistic Models
(chapter 16)

Structured Probabilistic Models

Tasks for structured probabilistic models (graphical models)

- Density estimation
- Denoising
- Sample generation
- Missing value imputation
  - Conditional sample generation
  - Conditional density estimation

Joint Probability and Inference

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix}$$

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$$

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_3) \overset{marginalization}{=} \sum_{v_i} \sum_{v_j} \boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2 = v_i, \boldsymbol{x}_3, \boldsymbol{x}_4 = v_j)$$

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \overset{product\ rule}{=} \boldsymbol{P}(\boldsymbol{x}_1 | \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \boldsymbol{P}(\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$$

$$\boldsymbol{P}(\boldsymbol{x}_1 | \boldsymbol{x}_2 = 5, \boldsymbol{x}_3 = 9) \overset{def.}{=} \frac{\sum_{v_j} \boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2 = 5, \boldsymbol{x}_3 = 9, \boldsymbol{x}_4 = v_j)}{\sum_{v_i} \sum_{v_j} \boldsymbol{P}(\boldsymbol{x}_1 = v_i, \boldsymbol{x}_2 = 5, \boldsymbol{x}_3 = 9, \boldsymbol{x}_4 = v_j)}$$

Representing Joint Probability

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix}, \quad \boldsymbol{x}_i \in \{0, 1\}$$

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$$

$\#\ parameters: \ 2^4$

$\boldsymbol{x}_3\boldsymbol{x}_4$ \ $\boldsymbol{x}_1\boldsymbol{x}_2$	00	01	10	11
00	p_1	p_2	p_3	p_4
01				
10				
11				

actually $2^4 - 1$
since $\sum_i p_i = 1$

Representing Joint Probability

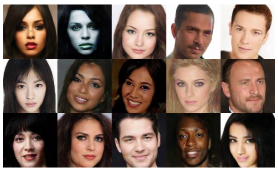
$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix}, \quad \boldsymbol{x}_i \in \{0,1\}$$

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$$

$\#\ parameters:\ 2^4$

$\boldsymbol{x}_3\boldsymbol{x}_4$ \ $\boldsymbol{x}_1\boldsymbol{x}_2$	00	01	10	11
00	p_1	p_2	p_3	p_4
01				
10				
11				

$actually\ 2^4 - 1$
$since\ \sum_i p_i = 1$



Images 128 pixels wide, 128 pixels tall, 256 levels R, G, and B pixel at each location → 128 x 128 x 3 variables in {0,…,255} → $(256)^{128 \times 128 \times 3}$ → $\boxed{256^{49152}}$

Representing Joint Probability

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix}, \quad \boldsymbol{x}_i \in \{0,1\}$$

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$$

$\boldsymbol{x}_3\boldsymbol{x}_4$ \ $\boldsymbol{x}_1\boldsymbol{x}_2$	00	01	10	11
00	p_1	p_2	p_3	p_4
01				
10				

Tabular Approach is Infeasible

- Memory: cannot store that many parameters
- Runtime: inference and sampling are both slow
- Statistical efficiency:
  extremely high number of parameters requires
  extremely high number of training examples

Representing Joint Probability

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix}, \quad \boldsymbol{x}_i \in \{0,1\}$$

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$$

$\boldsymbol{x}_3\boldsymbol{x}_4$ \ $\boldsymbol{x}_1\boldsymbol{x}_2$	00	01	10	11
00	p_1	p_2	p_3	p_4
01				
10				

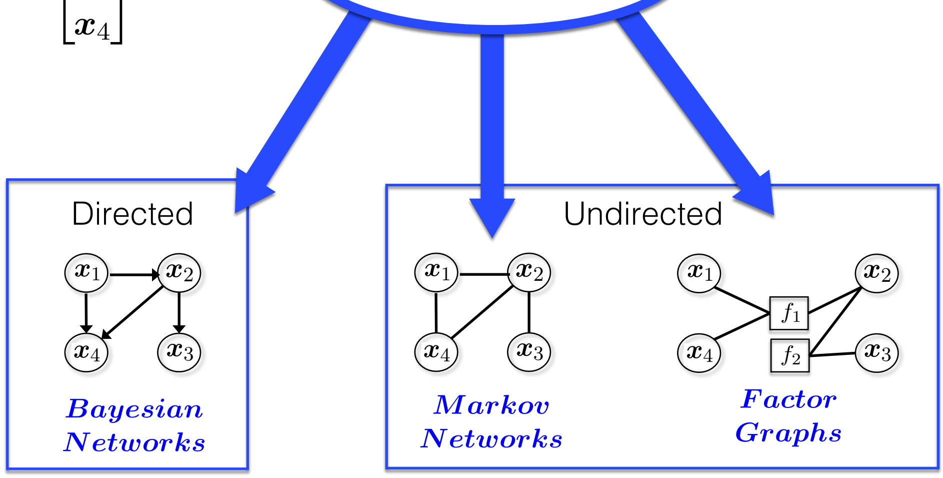
How to *save* space (and computation)

- Most variables influence each other
- Most variables do not influence each other *directly*
- Describe influence with a graph
    - Edges represent *direct* influence
    - Paths represent *indirect* influence
- Computational and statistical savings come from *omissions of edges*

Joint Probability and Graphical Models

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix}$$

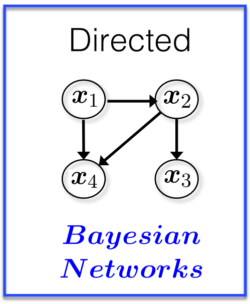
$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$$



Different ways to represent the joint probability

Directed Models

Directed models work best when influence clearly flows in one direction $\Rightarrow$ causal relation



Joint probability factorization

- Product rule applied many times (chain rule)

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = \boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_1)$$

- Use conditional independence ($\boldsymbol{x}_3$ is cond. ind. from $\boldsymbol{x}_1$ given $\boldsymbol{x}_2$)

$$\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_2)$$

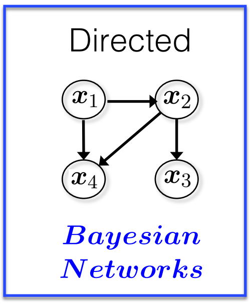
to simplify factors

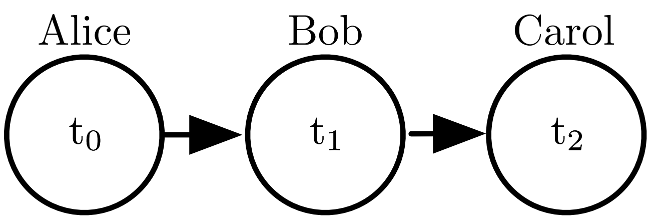
$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = \boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_1)$$

In general:

$$\boxed{\boldsymbol{P}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \prod_{i=1}^{n} \boldsymbol{P}(\boldsymbol{x}_i | \underbrace{pa_G(\boldsymbol{x}_i)}_{\substack{parents \\ in\ G}})}$$

Example: relay race



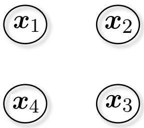


$$P(t_0, t_1, t_2) = P(t_0)P(t_1|t_0)\underbrace{P(t_2|t_0, t_1)}_{=P(t_2|t_1)}$$

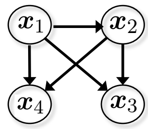
t_2 is conditionally independent from t_0 given t_1

Degrees of Dependence

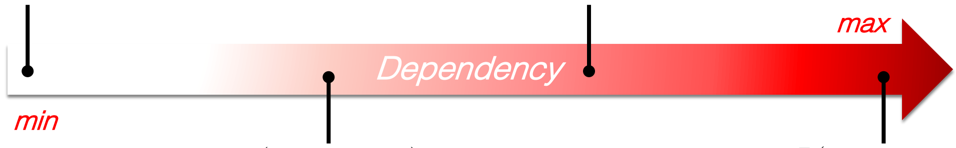
$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix}$$



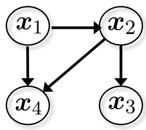
$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \stackrel{ind.}{=} \boldsymbol{P}(\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_3)\boldsymbol{P}(\boldsymbol{x}_4)$$



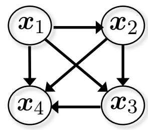
$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \stackrel{chain\ rule\ +\ cond.\ ind.}{=} \boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_1)$$



$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \stackrel{chain\ rule\ +\ cond.\ ind.}{=} \boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_1)$$



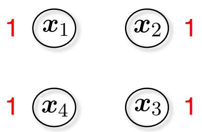
$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \stackrel{chain\ rule}{=} \boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_1)$$



Degrees of Dependence

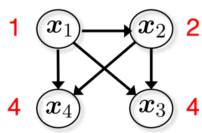
$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \boldsymbol{x}_4 \end{bmatrix}, \quad \boldsymbol{x}_i \in \{0, 1\}$$

\# param.
1+1+1+1 = 4



$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \stackrel{ind.}{=} \boldsymbol{P}(\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_3)\boldsymbol{P}(\boldsymbol{x}_4)$$

\# param.
1+2+4+4 = 11

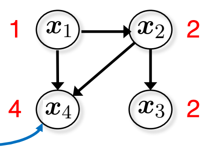


$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \stackrel{chain\ rule\ +\ cond.\ ind.}{=} \boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_1)$$

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \stackrel{chain\ rule\ +\ cond.\ ind.}{=} \boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_1)$$

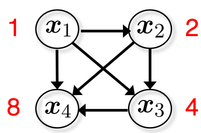
$\boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2)$

$\boldsymbol{x}_4$ \ $\boldsymbol{x}_1\boldsymbol{x}_2$	00	01	10	11
0	p_1	p_2	p_3	p_4
1	$1-p_1$	$1-p_2$	$1-p_3$	$1-p_4$



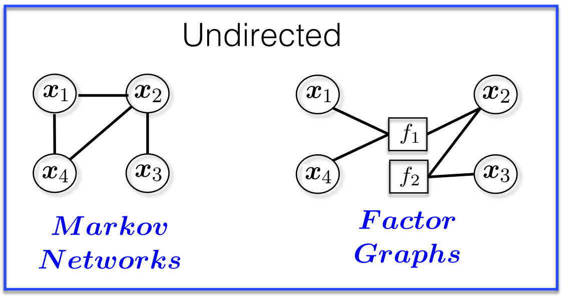
1+2+2+4 = 9
\# param.

$$\boldsymbol{P}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \stackrel{chain\ rule}{=} \boldsymbol{P}(\boldsymbol{x}_4|\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)\boldsymbol{P}(\boldsymbol{x}_3|\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)\boldsymbol{P}(\boldsymbol{x}_1)$$

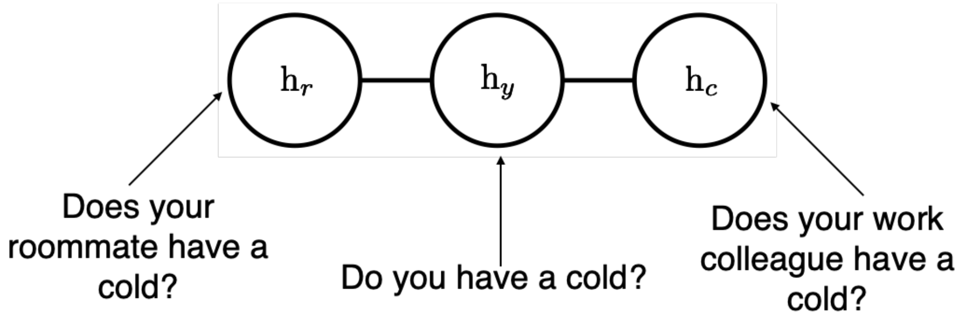


1+2+4+8 = 15
\# param.

Undirected Models



Undirected models work best when influence has no clear direction or is best modeled as flowing in both directions

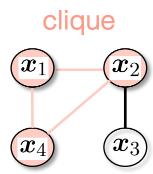


Undirected Models

Joint probability factorization

- Introduce for each clique C in G a factor $\phi(C)$ (clique potential)
  - $\phi(C) \geq 0$ (nonnegative functions)
- Define the **unnormalized probability distribution** obtained as

$$\tilde{\boldsymbol{P}}(\boldsymbol{X}) = \prod_{C \in G} \phi(C)$$



- Introduce the partition function Z

Problem: computationally intractable ⟶ $Z = \int \tilde{\boldsymbol{P}}(\boldsymbol{X}) d\boldsymbol{X}$

- to get the **normalized probability distribution**

$$\boldsymbol{P}(\boldsymbol{X}) = \frac{1}{Z} \tilde{\boldsymbol{P}}(\boldsymbol{X})$$

Undirected Models

Special case: Energy Based Models

- Enforce $\phi(C) > 0$ by defining the **unnormalized probability distribution** as
$$\tilde{\boldsymbol{P}}(\boldsymbol{X}) = \exp(-E(\boldsymbol{X}))$$
- $E(\boldsymbol{X})$ is the energy function
- different cliques will correspond to different terms of the energy function

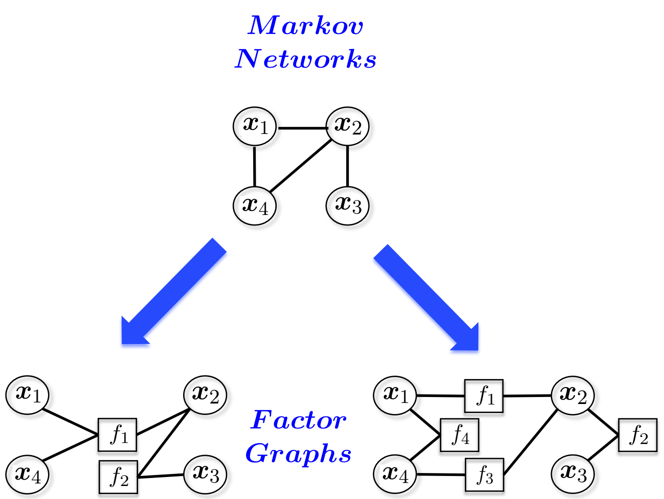
$$\tilde{\boldsymbol{P}}(\boldsymbol{X}) = \exp(-E(\boldsymbol{X})) = \exp(-\sum_{C \in G} E_C(\boldsymbol{X}_C)) = \prod_{C \in G} \exp(-E_C(\boldsymbol{X}_C))$$

where $E_C(\cdot)$ and $\boldsymbol{X}_C$ are the energy term and the subset of variables associated to clique C, respectivelly

- Example of **Boltzmann distribution**
- **Free Energy** for models with latent variables $\boldsymbol{h}$

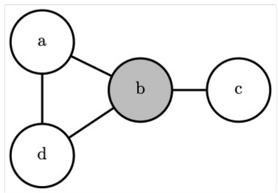
$$\mathcal{F}(\boldsymbol{x}) = -\log \sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{x}, \boldsymbol{h}))$$

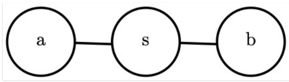
Undirected Models



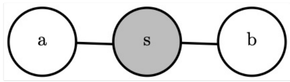
Factor Graphs remove ambiguity about graph clique factorization

Separation





When s is not observed, influence can flow from a to b and vice versa through s.



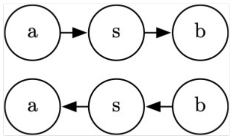
When s is observed, it blocks the flow of influence between a and b: they are *separated*

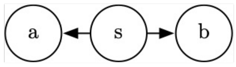
The nodes a and c are separated

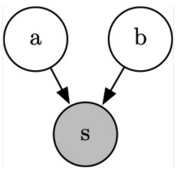
One path between a and d is still active, though the other path is blocked, so these two nodes are not separated.

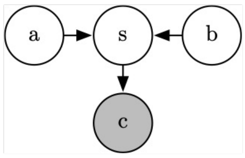
d-separation

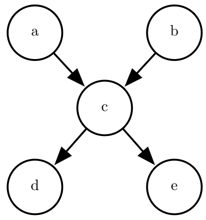
The flow of influence is more complicated for directed models
The path between a and b is active for all of these graphs:











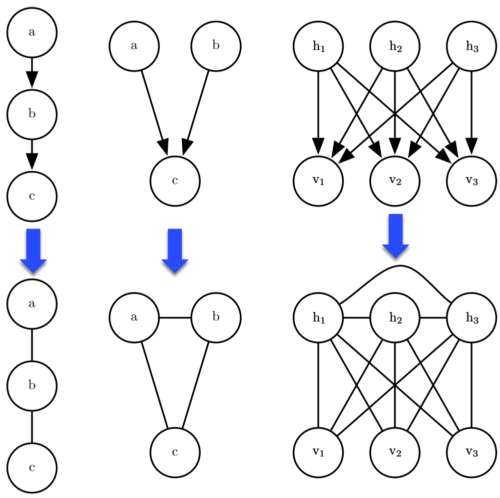
a and b are d-separated given the empty set
a and e are d-separated given c
d and e are d-separated given c

a and b are NOT d-separated given c
a and b are NOT d-separated given d

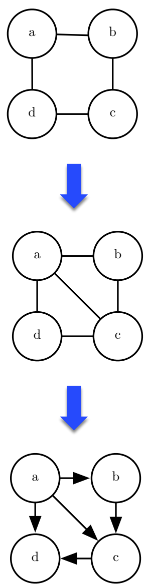
Converting between graphs

From directed to undirected

Must add an edge between unconnected coparents



From undirected to directed



No loops of length greater than three allowed!

Add edges to triangulate long loops

Assign directions to edges. No directed cycles allowed

Sampling

Directed Models

Easy and fast to draw fair samples from the whole model

Ancestral sampling :
- pass through the graph in topological order
- sample each node given its parents

Harder to sample some nodes given other nodes, unless the observed nodes are at the start of the topology

Undirected Models

Usually requires Markov chains (Gibbs sampling)

Usually cannot be done exactly

Usually requires multiple iterations even to approximate