Lecture 2

Independence Modelling

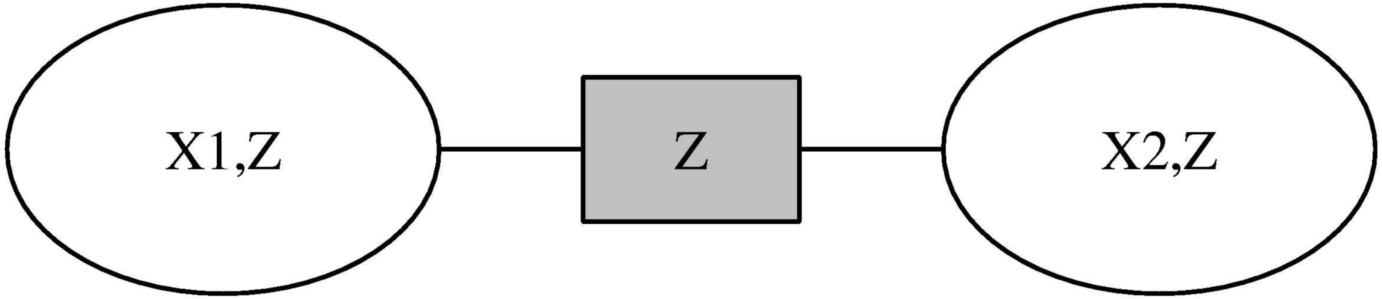
- Independence is a way of reducing/simplifying complexity/effort/cost in inference/learning/elicitation/optimization.

  NB. effective size of a search space for domain X is $2^{I(X)}$ and independence reduces entropy $I(X)$
- Independence arises naturally with causal and generative models.
- We use Lauritzen-style definitions for independence tests.

Overview

- **Independence and Problem Decomposition**
- Undirected Graphs
- A Tree of Cliques
- Directed Graphs
- A Catalogue of Graphical Forms

Definitions of Independence



- definition symmetric in X_1 and X_2
- Z is the *conditioning* or *separating* set
- Z appears on both sides of the decomposition
- for consistency $X_1 \cap X_2 \subseteq Z$, *e.g.* $\{a\} \perp\!\!\!\perp \{a,b\} | \{c,d\}$ is inconsistent because a is on both sides

Following definitions equivalent for $X_1 \perp\!\!\!\perp X_2 | Z$:

$$p(X_1, X_2 | Z) = p(X_1 | Z) p(X_2 | Z) \text{ whenever } p(Z) > 0$$
$$p(X_1 | X_2, Z) = p(X_1 | Z) \text{ whenever } p(X_2, Z) > 0$$
$$p(X_2 | X_1, Z) = p(X_2 | Z) \text{ whenever } p(X_1, Z) > 0$$
$$p(X_1, X_2, Z) = f(X_1, Z) g(X_2, Z) \text{ for some functions } f(\cdot), g(\cdot)$$

Decomposition: Maximization

Suppose we wish to maximize a function on finite discrete variable sets X_1, X_2, Z, all mutually disjoint, of the form $lf(X_1, Z) + lg(X_2, Z)$. Simplifies to:

$$\max_{Z} \left(\max_{X_1} lf(X_1, Z) + \max_{X_2} lg(X_2, Z) \right)$$

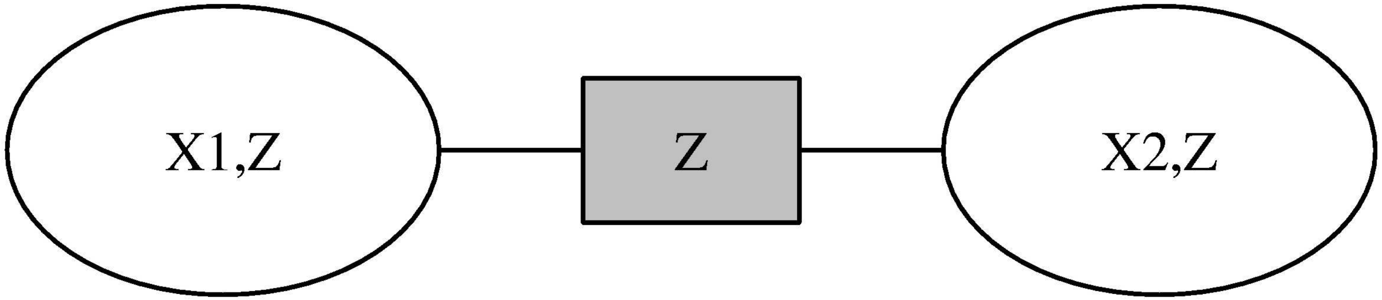
This algorithm returns $(\hat{X}_1, \hat{X}_2, \hat{Z})$ at a maximum:

1. Build a table on Z given by $lf_{X_1}(Z) = \text{max}_{X_1} lf(X_1, Z)$.
2. Build a table on Z given by $lg_{X_2}(Z) = \text{max}_{X_2} lg(X_2, Z)$.
3. From these two tables, compute

$$\hat{Z} = \text{argmax}_Z lf_{X_1}(Z) + lg_{X_2}(Z)$$

4. Compute $\hat{X}_1 = \text{argmax}_{X_1} lf(X_1, \hat{Z})$.
5. Compute $\hat{X}_2 = \text{argmax}_{X_1} lg(X_2, \hat{Z})$.

Decomposition Summary



- Reduces computation to local effort on X_1, Z and X_2, Z separately.
- Need to transfer summaries statistics of Z in both directions to make local tasks consistent with the global task.
- When computation is super-linear in number of variables, savings are made for large enough sets; significant savings made when $|Z| \ll |X_1 \cup X_2|$
- Applies to most constraint satisfaction, optimization and probability problems.

Decomposition: Summation

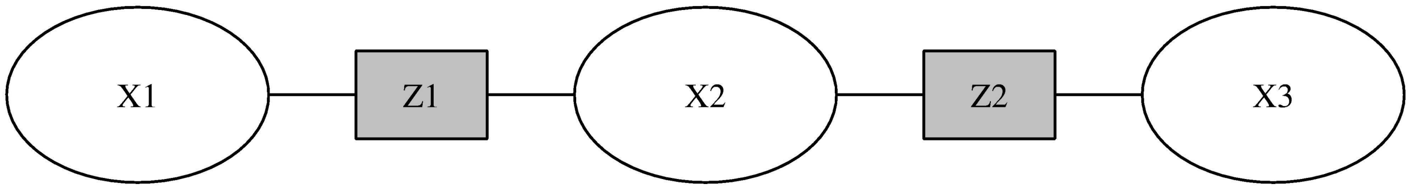
Suppose we wish to sum a function on finite discrete variable sets X_1, X_2, Z which takes the form $p(X_1, X_2, Z) = f(X_1, Z)g(X_2, Z)$. We wish to compute all marginals for $x \in X_1 \cup X_2 \cup Z$:

$$p(x) = \sum_{X_1 \cup X_2 \cup Z - \{x\}} f(X_1, Z)g(X_2, Z)$$

The following algorithm finds these:

1. Build a table on Z given by $f_{X_1}(Z) = \sum_{X_1} f(X_1, Z)$.
2. Build a table on Z given by $g_{X_2}(Z) = \sum_{X_2} g(X_2, Z)$.
3. Compute $p(X_1) = \sum_Z f(X_1, Z) g_{X_2}(Z)$.
4. Compute $p(X_2) = \sum_Z g(X_2, Z) f_{X_1}(Z)$.
5. Compute $p(Z) = g_{X_2}(Z) f_{X_1}(Z)$.
6. Compute the marginals from these.

3-way Decompositions



- Slightly different formulation, now $Z_1 = X_1 \cap X_2$ and $Z_2 = X_2 \cap X_3$
- For all possible pair-wise decompositions to be consistent independent statements,

  $$X_1 \perp\!\!\!\perp X_2 \,|\, Z_1;\; X_2 \perp\!\!\!\perp X_3 \,|\, Z_2;\; X_1 \cup X_2 \perp\!\!\!\perp X_3 \,|\, Z_2;\; X_1 \perp\!\!\!\perp X_2 \cup X_3 \,|\, Z_1$$

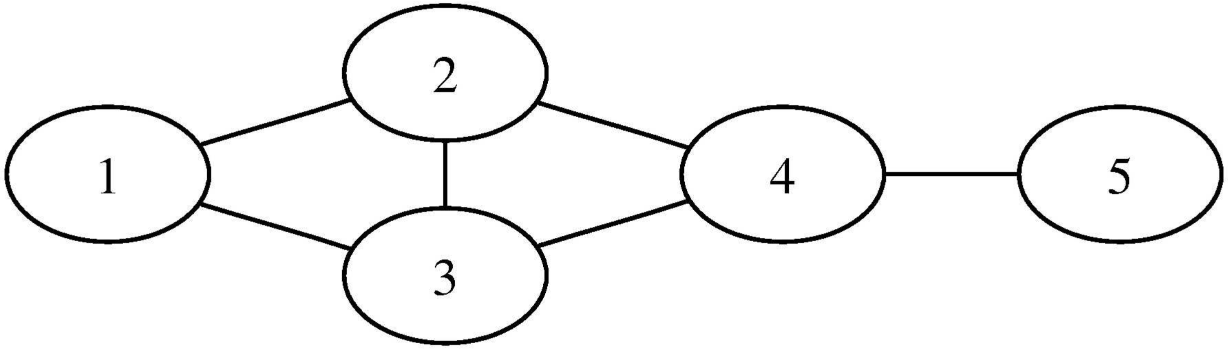
  it is necessary and sufficient that $X_1 \cap X_3 \subseteq X_2$.

  Partial Proof: for $X_1 \perp\!\!\!\perp X_2 \cup X_3 | Z_1$ case to be consistent, $X_1 \cap (X_2 \cup X_3) \subseteq Z_1 = X_1 \cap X_2$, which reduces to $X_1 \cap X_3 \subseteq X_1 \cap X_2$, likewise $X_1 \cap X_3 \subseteq X_2 \cap X_3$; intersecting these two yields $X_1 \cap X_3 \subseteq X_2 \cap (X_1 \cap X_3)$, hence $X_1 \cap X_3 \subseteq X_2$.

Overview

- Independence and Problem Decomposition
- **Undirected Graphs**
- A Tree of Cliques
- Directed Graphs
- A Catalogue of Graphical Forms

Undirected Graph, example



Local Markov Property: (first set X_1 is a singleton) $1 \perp\!\!\!\perp 4,5|2,3$; $2 \perp\!\!\!\perp 5|1,3,4$; $5 \perp\!\!\!\perp 1,2,3|4$; etc.

Global Markov Property: (indepence on general sets) $1,2 \perp\!\!\!\perp 5|4$; $1,2,3 \perp\!\!\!\perp 5|4$; etc.

Functional form:

$$f(1,2,3)g(2,3,4)h(4,5)$$

Undirected Graph

For an undirected graph on variables X, the following are equivalent when $p(X) > 0$ for all values of X:

Local Markov Property: for all $x \in X$,

$$\{x\} \perp\!\!\!\perp (X - \text{nbrs}(x) - \{x\}) \,|\, \text{nbrs}(x)$$

Global Markov Property: for all $X_1, X_2, Z \subseteq X$, $X_1 \perp\!\!\!\perp X_2 | Z$ iff X_1 is separated from X_2 in the graph by Z.

Functional Form: for $\mathscr{C}$ the set of cliques in the graph, X_C the restriction of X to the set C, functions $f_C(\cdot)$ exist so that

$$p(X) \;=\; \prod_{C \in \mathscr{C}} f_C(X_C) \;.$$

Undirected Graph, cont.

- Equivalence between functional form and the local Markov property for finite discrete variables is called the Hammersley-Clifford Theorem. It generalizes the corresponding definitions of independence.
- Exercise: find a simple half page proof of this.
- Alternative functional form with parameters α_C:

$$\log p(X) \; = \; \sum_{C \in \mathscr{C}} \alpha_C l f_C(X_C) - \log Z \ .$$

Note physicists, statisticians and others often like their log-probability functions to be nice simple additive forms like this!

Undirected Graph, Independence

- The Global Markov Property defines how to test for independence:
  $X_1 \perp\!\!\!\perp X_2 \,|\, Z$ iff X_1 is separated from X_2 in the graph by Z
- The same independence test applies for doing problem decomposition in contraint graphs (i.e., constraint satisfaction and optimization).
- Finding a good separating set Z is like a Mincut problem, but in the dual space (swapping roles of nodes and edges).

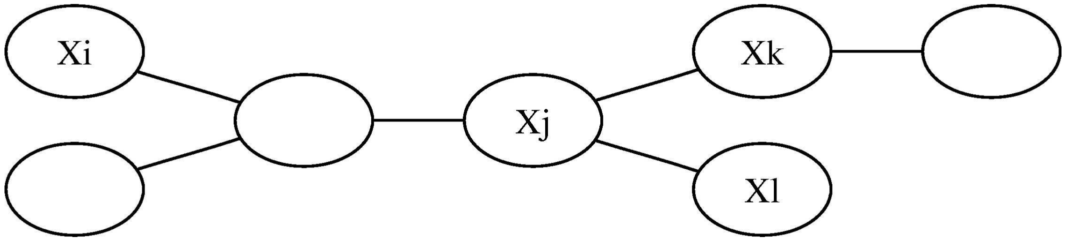
Finding a Good Decomposition

- *Graph partitioning* or *Hypergraph partitioning*, see Alpert and Kahng 1995.
- Mincut in the dual space mostly finds trivial cuts where one side is almost empty.
- "Balanced" Mincut, forcing X_1, X_2 to be similar sizes is NP-complete.
- Local search works poorly (compared with others).
- Spectral methods (approximate task with maximum eigenvector computation) works quite well.

Overview

- Independence and Problem Decomposition
- Undirected Graphs
- **A Tree of Cliques**
- Directed Graphs
- A Catalogue of Graphical Forms

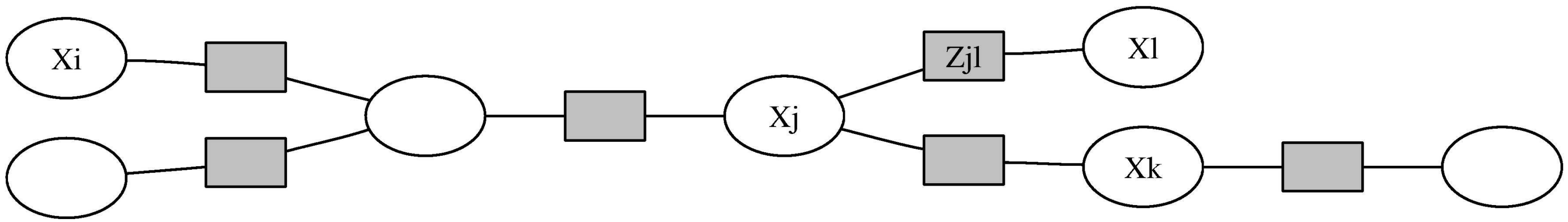
A Tree of Cliques



- Lets put variable sets in the nodes instead of single variables, but restrict it to be a tree (no cycles).
- The variable sets cannot be unrestricted: independence statements likes $\{a\} \perp\!\!\!\perp \{a,b\} | \{c,d\}$ should not be allowed (i.e., a is independent of itself.
- For any connected subtree, each split must form a consistent independence statement for its two sides. The necessary and sufficient conditions are:

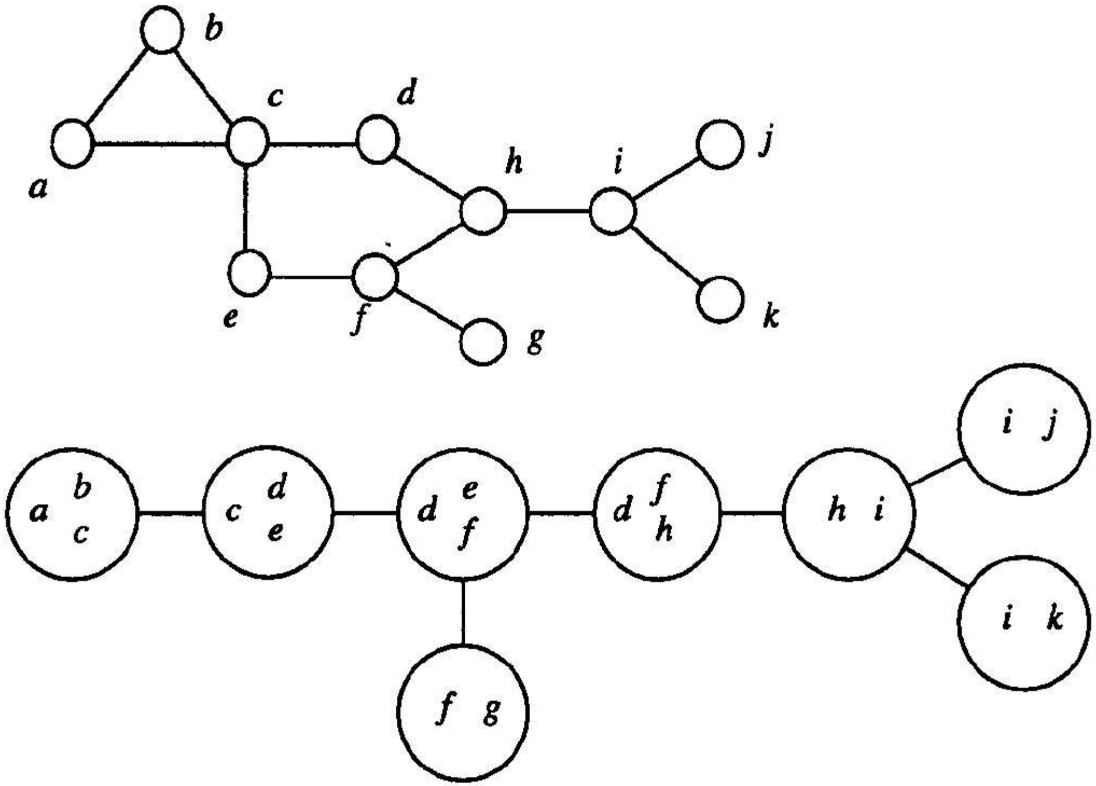
  if node X_j is on the path between nodes X_i and X_k
  then $X_i \cap X_k \subseteq X_j$
- Under these conditions, this is called a *clique tree*, where the X_j are called cliques.

Multiple Decompositions



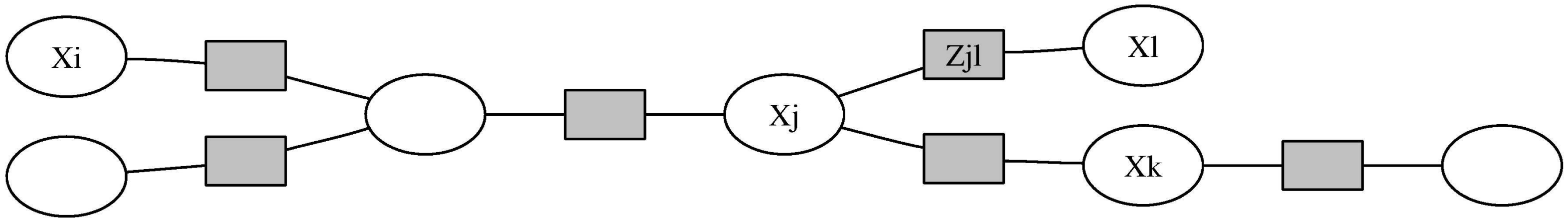
- Alternatively, generalize independence to a tree with nodes (sets X_j) as ovals and separating sets as boxes.
- For any connected subtree, each separating set must form a consistent independence statement for its two sides. The necessary and sufficient conditions are:
  - if separating set $Z_{j,l}$ separates nodes X_j and X_l, then $Z_{j,l} = X_j \cap X_l$,
  - if node X_j is on the path between nodes X_i and X_k then $X_i \cap X_k \subseteq X_j$

Multiple Decompositions, example



Example from "A Tourist Guide through Treewidth", H.L. Bodlaender. Shows a good clique tree for the corresponding undirected graph, i.e., every clique in the undirected graph is a subset of a clique in the clique tree.

Multiple Decompositions, cont.

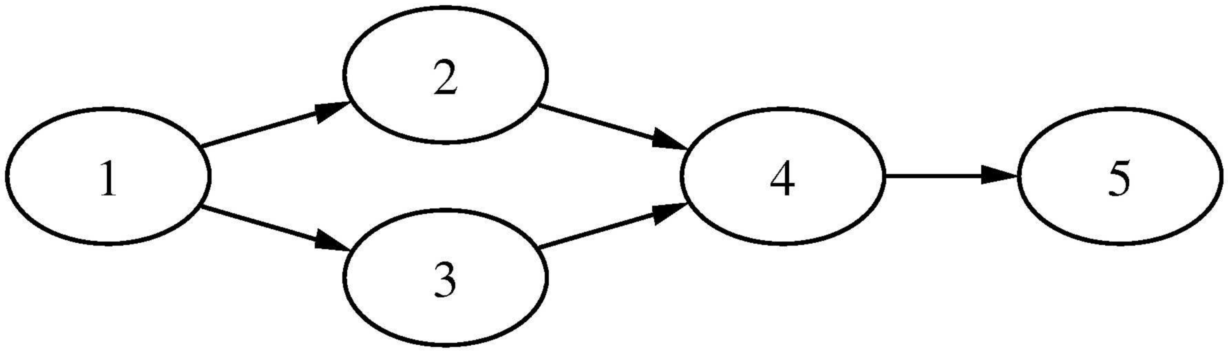


- Computation on a clique tree is like the two-node case of simple independence: summary statistics need to go in each direction across every separating set so that every clique task is consistent with the global task.
- The *tree-width* of the clique tree is $T = \max_j |X_j| - 1$, one off the size of the largest clique.
- Many NP-complete problems solvable in $O(C2^T)$ for C the number of cliques in a clique tree for the problem, since the computation on each clique is $O(2^T)$

Overview

- Independence and Problem Decomposition
- Undirected Graphs
- A Tree of Cliques
- **Directed Graphs**
- A Catalogue of Graphical Forms

Directed Graph, example



Local Markov Property: $3 \perp\!\!\!\perp 2 | 1$; $4 \perp\!\!\!\perp 1 | 2, 3$; $5 \perp\!\!\!\perp 1, 2, 3 | 4$

Functional form:

$$p(1)p(2|1)p(3|1)p(4|2,3)p(5|4)$$

Directed Graph

> **NB.** English language purists like to point out that the Directed Acyclic Graph (DAG) is in fact an Acyclic Directed Graph (ADG).

For a directed graph on variables X, the following are equivalent:

Local Markov Property: for all $x \in X$,

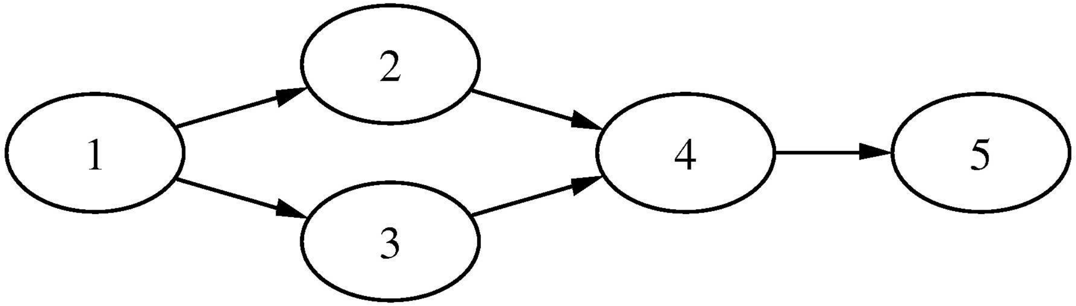
$$\{x\} \perp\!\!\!\perp (X - \text{descendants}(x) - \text{parents}(x) - \{x\}) \,|\, \text{parents}(x)$$

Functional Form:

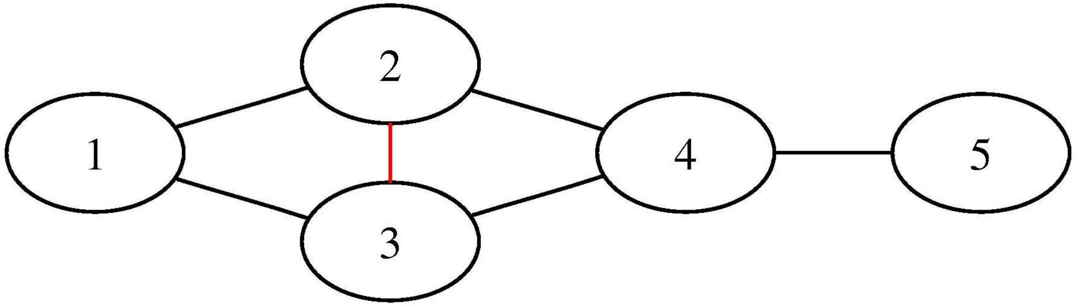
$$p(X) \;=\; \prod_{x \in X} p(x|\text{parents}(x)) \;.$$

For the corresponding Global Markov Property, we need another definition, later ...

Directed to Undirected Graph



$$p(1)p(2|1)p(3|1)p(4|2,3)p(5|4)$$



(we added an arc between 2 and 3, 4's parents)

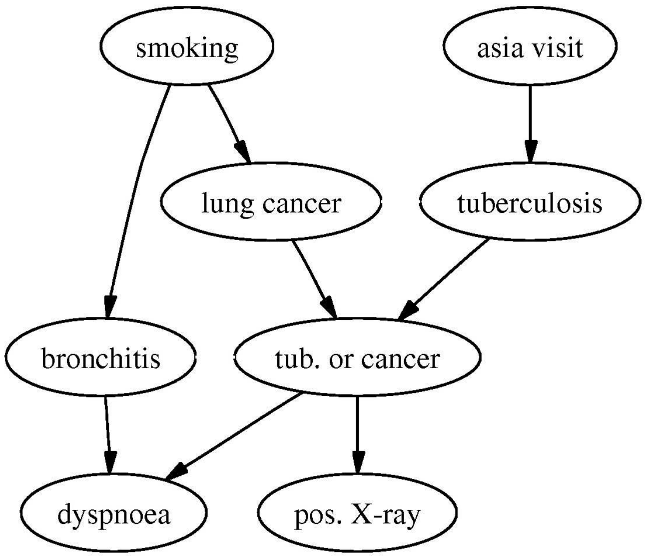
Moralizing a Directed Graph

- We look at the functional form of the DAG as if it were for an undirected graph,

$$\prod_{x \in X} p(x|\text{parents}(x)) \longrightarrow \prod_{C \in \mathscr{C}} f_C(X_C) \ .$$

i.e. all the sets $\{x\} \cup \text{parents}(x)$ need to be a clique in the undirected graph.
- We need to make sure that *every two common parents have an arc between them*.
- Converted a directed to an undirected graph (preserving potential dependencies) is thus called *moralizing*, as we "marry" unconnected common parents.

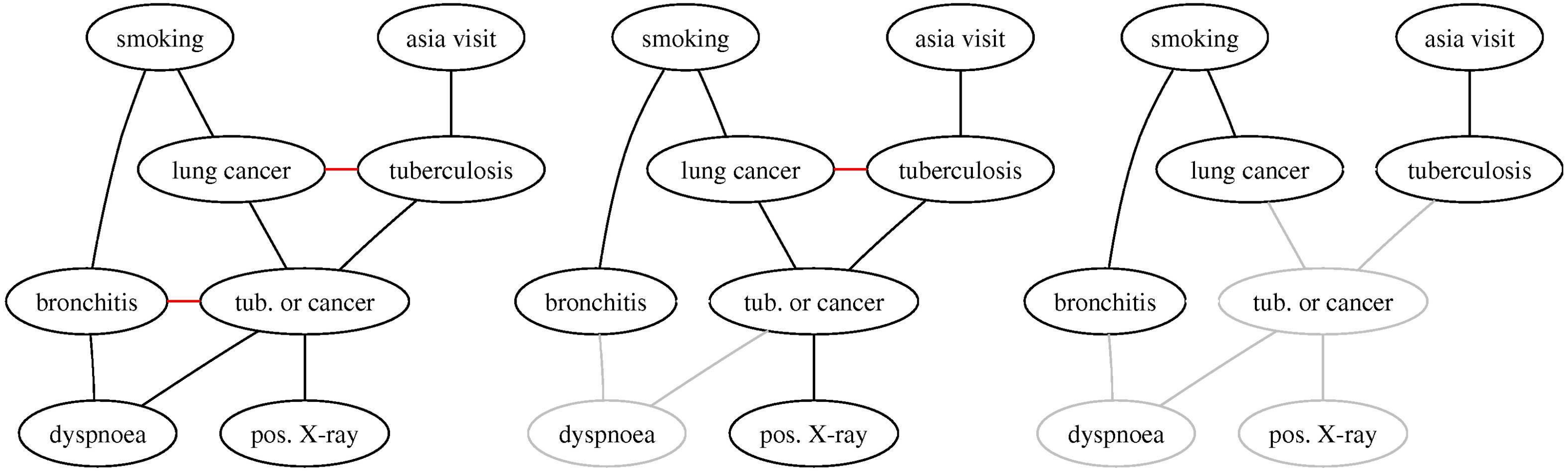
Directed to Many Undirected Graphs



Depending on the ancestral sets used, different undirected graphs can be obtained.

Directed to Many Undirected Graphs

Red arcs show the moral arcs added to parents. The light sections have been removed from the graph to produce each case.



Directed Graph, Independence

- The Global Markov Property defines how to test for independence:

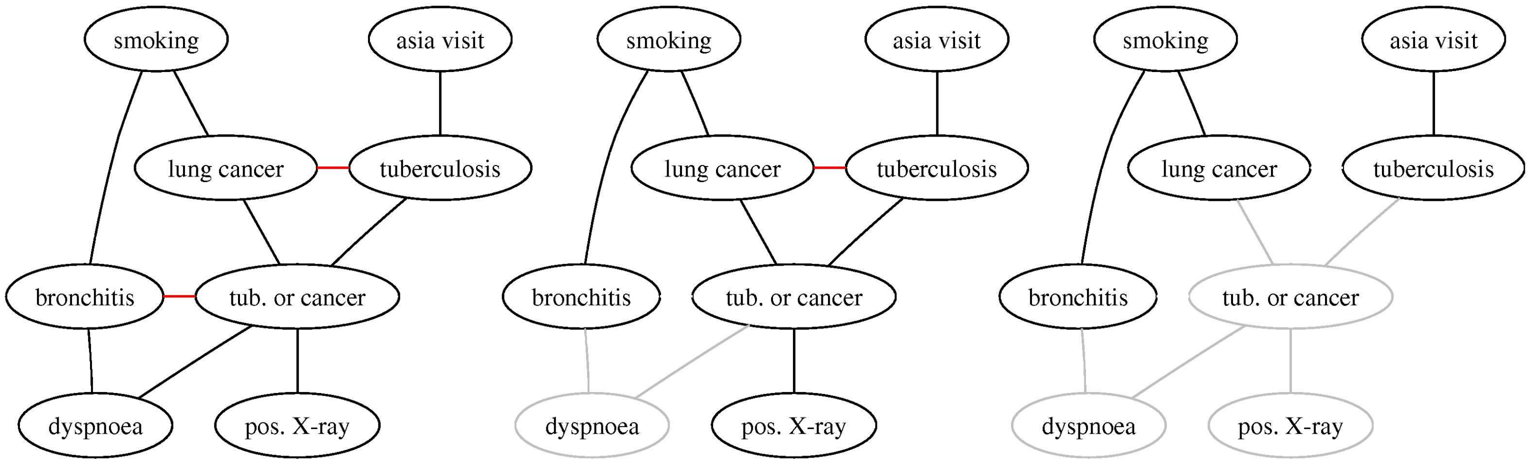
  $X_1 \perp\!\!\!\perp X_2 | Z$ iff X_1 is separated from X_2 in the undirected graph formed by moralizing the graph on the smallest ancestral set containing X_1, X_2 and Z.
- An equivalent formulation is the *d-separation* criterion, used in the AI literature.

Independence, example

Does Z separate X_1 and X_2 in any of the moralized graphs on the ancestral sets?

"asia visit" $\perp\!\!\!\perp$ "smoking", but not if "pos. X-ray" is given.

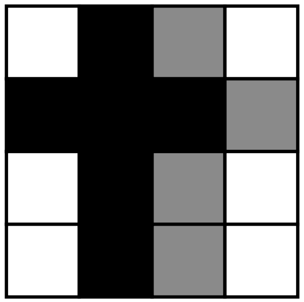
"asia visit" $\perp\!\!\!\perp$ "bronchitis" given "lung cancer", but not if "dyspnoea" is also given.

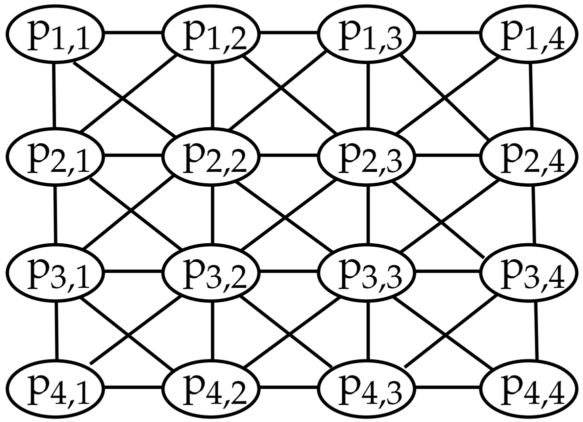


Overview

- Independence and Problem Decomposition
- Undirected Graphs
- A Tree of Cliques
- Directed Graphs
- **A Catalogue of Graphical Forms**

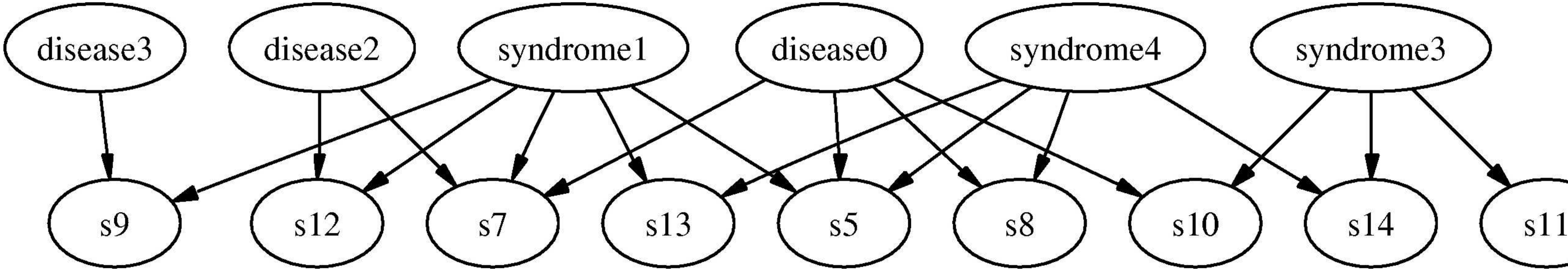
Image Models





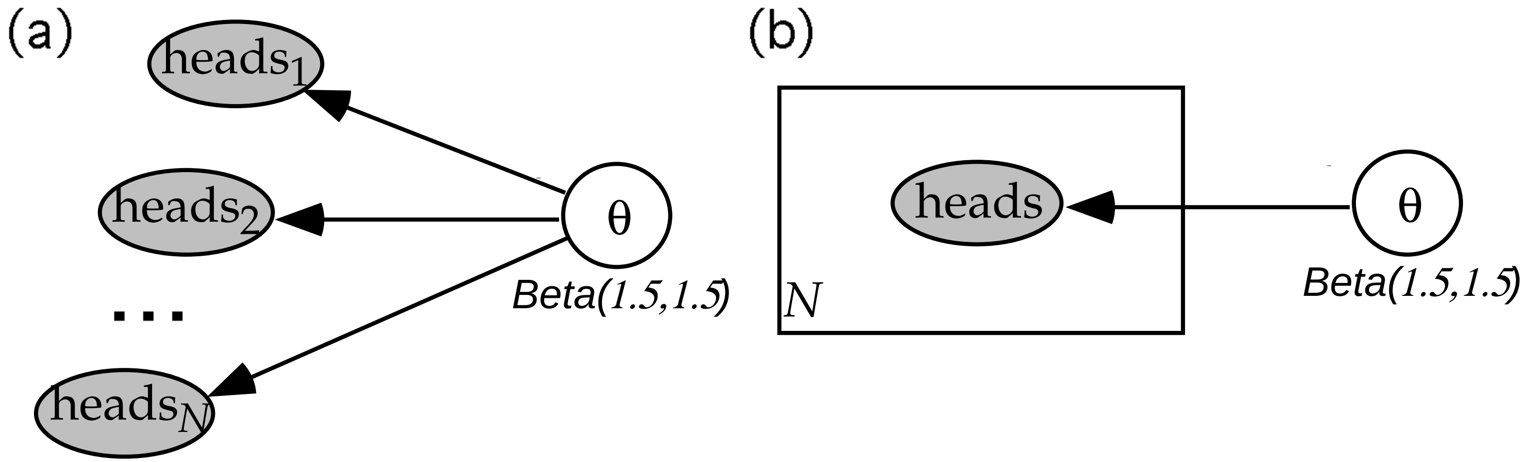
Simple 4×4 image. Top graph says all pixels influenced only by their neighbour's values. Has checkered history in image analysis, but becoming more successful.

Expert Systems: 2 Level Belief Nets



Model layered with 2 sets of variables: diseases and syndromes in first level causing symptoms in the second level. Special algorithms used for this structure. *i.e.* QMR-DT

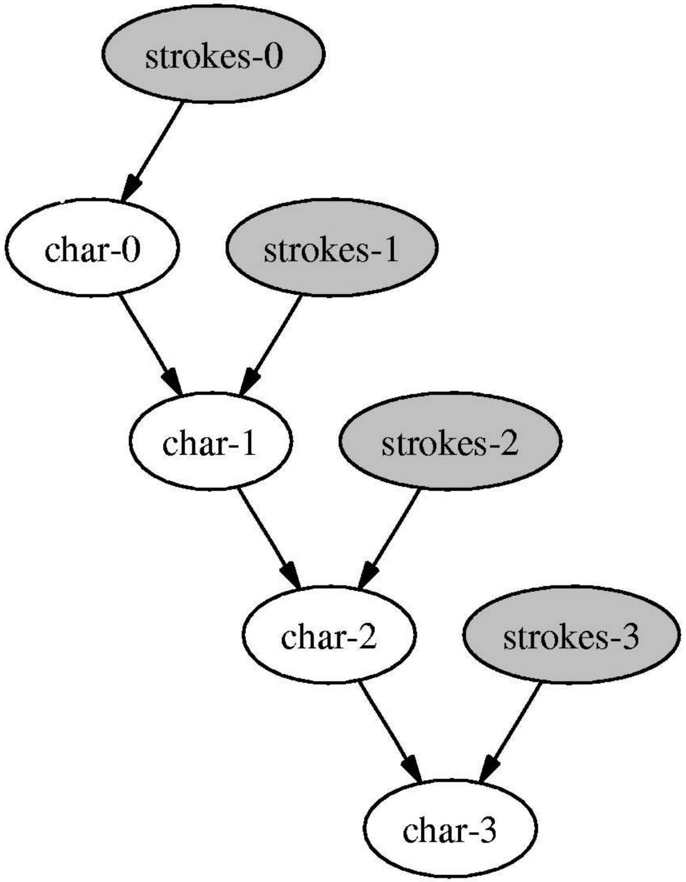
Estimating the Bias of a Die/Coin



What is the bias of the coin, as given by θ?. Estimate from N coin tosses. Observed (sampled) data is shaded. Unknown parameter left unshaded.

General versions model *independent and identically distributed* variables in sampling.

Character Recognition



The observed data (again shaded) is the character strokes. The unknown data one wishes to predict is the underlying characters. All is sequential. Called *Hidden Markov Models*.

More Models

- clustering
- sequential models,
- simple decision models
- principle components analysis
- diagnostic models

See the other online slide sets.

Next Week

- Review sections I, II and VI of Aji and McEliece.
- Review Bishop's tutorial Part I to see how you are going.