Lecture 2:

Directed Graphical Models

Sam Roweis

January 7, 2004

Conditional Independence

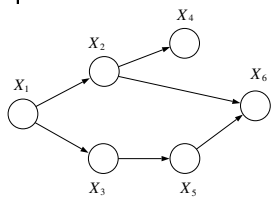
- Notation: $\mathbf{x}_A \perp \mathbf{x}_B | \mathbf{x}_C$
  Definition: two (sets of) variables $\mathbf{x}_A$ and $\mathbf{x}_B$ are conditionally independent given a third $\mathbf{x}_C$ if:
  $$\mathsf{P}(\mathbf{x}_A, \mathbf{x}_B | \mathbf{x}_C) = \mathsf{P}(\mathbf{x}_A | \mathbf{x}_C)\mathsf{P}(\mathbf{x}_B | \mathbf{x}_C) \qquad \forall \mathbf{x}_C$$
  which is equivalent to saying
  $$\mathsf{P}(\mathbf{x}_A | \mathbf{x}_B, \mathbf{x}_C) = \mathsf{P}(\mathbf{x}_A | \mathbf{x}_C) \qquad \forall \mathbf{x}_C$$
- Only a subset of all distributions respect any given (nontrivial) conditional independence statement. The subset of distributions that respect all the CI assumptions we make is the *family of distributions consisitent with our assumptions*.
- Probabilistic graphical models are a powerful, elegant and simple way to specify such a family.

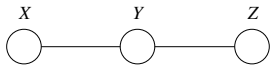
Joint Probabilities

- Goal 1: represent a joint distribution $\mathsf{P}(\mathbf{X}) = \mathsf{P}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ compactly even when there are many variables.
- Goal 2: efficiently calculate marginal and conditionals of such compactly represented joint distributions.
- Notice: for n discrete variables of arity k, the naive (table) representation is HUGE: it requires k^n entries.
- We need to make some *assumptions* about the distribution.
  One simple assumption: independence == complete factorization:
  $$\mathsf{P}(\mathbf{X}) = \prod_i \mathsf{P}(\mathbf{x}_i)$$
- But the independence assumption is too restrictive.
  So we make *conditional independence* assumptions instead.

Probabilistic Graphical Models

- Probabilistic graphical models represent large joint distributions compactly using a set of "local" relationships specified by a graph.
- Each random variable in our model corresponds to a graph node.
- There are directed/undirected *edges* between the nodes which tell us qualitatively about the *factorization* of the joint probability.
- There are *functions* stored at the nodes which tell us the quantitative details of the pieces into which the distribution factors.





- Graphical models are also known as Bayes(ian) (Belief) Net(work)s.

Directed Graphical Models

- Consider *directed acyclic graphs* over n variables.
- Each node has (possibly empty) set of parents π_i.
- Each node maintains a function $f_i(\mathbf{x}_i; \mathbf{x}_{\pi_i})$ such that $f_i > 0$ and $\sum_{\mathbf{x}_i} f_i(\mathbf{x}_i; \mathbf{x}_{\pi_i}) = 1 \; \forall \pi_i$.
- Define the joint probability to be:

$$\mathsf{P}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \prod_i f_i(\mathbf{x}_i; \mathbf{x}_{\pi_i})$$

  Even with no further restriction on the the f_i, it is always true that

$$f_i(\mathbf{x}_i; \mathbf{x}_{\pi_i}) = \mathsf{P}(\mathbf{x}_i | \mathbf{x}_{\pi_i})$$

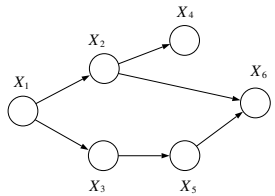
  so we will just write

$$\mathsf{P}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \prod_i \mathsf{P}(\mathbf{x}_i | \mathbf{x}_{\pi_i})$$

- Factorization of the joint in terms of *local conditional probabilities*. Exponential in "fan-in" of each node instead of in total variables n.

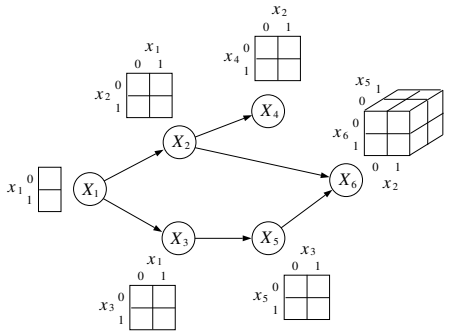
Example DAG

- Consider this six node network:



The joint probability is now:

$$\mathsf{P}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6) = \mathsf{P}(\mathbf{x}_1)\mathsf{P}(\mathbf{x}_2|\mathbf{x}_1)\mathsf{P}(\mathbf{x}_3|\mathbf{x}_1)\mathsf{P}(\mathbf{x}_4|\mathbf{x}_2)\mathsf{P}(\mathbf{x}_5|\mathbf{x}_3)\mathsf{P}(\mathbf{x}_6|\mathbf{x}_2, \mathbf{x}_5)$$



Conditional Independence in DAGs

- If we order the nodes in a directed graphical model so that parents always come before their children in the ordering then the graphical model implies the following about the distribution:

$$\{\mathbf{x}_i \perp \mathbf{x}_{\widetilde{\pi}_i} | \mathbf{x}_{\pi_i}\} \forall i$$

  where $\mathbf{x}_{\widetilde{\pi}_i}$ are the nodes coming before $\mathbf{x}_i$ that are not its parents.
- In other words, the DAG is telling us that each variable is conditionally independent of its non-descendants given its parents.
- Such an ordering is called a "topological" ordering.

Missing Edges

- Key point about directed graphical models:
  Missing edges imply conditional independence
- Remember, that by the chain rule we can always write the full joint as a product of conditionals, given an ordering:

$$\mathsf{P}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \ldots) = \mathsf{P}(\mathbf{x}_1)\mathsf{P}(\mathbf{x}_2|\mathbf{x}_1)\mathsf{P}(\mathbf{x}_3|\mathbf{x}_1, \mathbf{x}_2)\mathsf{P}(\mathbf{x}_4|\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)\ldots$$

- If the joint is represented by a DAGM, then some of the conditioned variables on the right hand sides are missing. This is equivalent to enforcing conditional independence.
- Start with the "idiot's graph": each node has all previous nodes in the ordering as its parents.
- Now remove edges to get your DAG.
- Removing an edge into node i eliminates an argument from the conditional probability factor $p(\mathbf{x}_i | \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{i-1})$

Even more structure

- Surprisingly, once you have specified the basic conditional independencies, there are other ones that follow from those.
- In general, it is a hard problem to say which extra CI statements follow from a basic set. However, in the case of DAGMs, we have an efficient way of generating *all* CI statements that *must be true* given the connectivity of the graph.
- This involves the idea of *d-separation* in a graph.
- Notice that for specific (numerical) choices of factors at the nodes there may be even more conditional independencies, but we are only concerned with statements that are always true of every member of the family of distributions, no matter what specific factors live at the nodes.
- Remember: the graph alone represents a *family of joint distributions consistent with its CI assumptions*, not any specific distribution.

Chain

- Q: When we condition on $\mathbf{y}$, are $\mathbf{x}$ and $\mathbf{z}$ independent?

$$\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{y}|\mathbf{x})\mathsf{P}(\mathbf{z}|\mathbf{y})$$

which implies

$$\begin{aligned}\mathsf{P}(\mathbf{z}|\mathbf{x},\mathbf{y}) &= \frac{\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z})}{\mathsf{P}(\mathbf{x},\mathbf{y})} \\ &= \frac{\mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{y}|\mathbf{x})\mathsf{P}(\mathbf{z}|\mathbf{y})}{\mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{y}|\mathbf{x})} \\ &= \mathsf{P}(\mathbf{z}|\mathbf{y})\end{aligned}$$

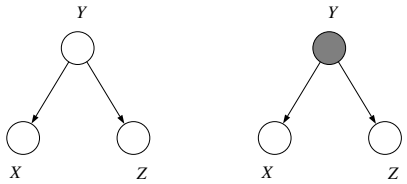
and therefore $\mathbf{x} \perp \mathbf{z}|\mathbf{y}$

- Think of $\mathbf{x}$ as the past, $\mathbf{y}$ as the present and $\mathbf{z}$ as the future.

D-separation

- D-separation, or *directed-separation* is a notion of connectedness in DAGMs in which two (sets of) variables may or may not be connected conditioned on a third (set of) variable.
- D-connection implies conditional dependence and d-separation implies conditional independence.
- In particular, we say that $\mathbf{x}_A \perp \mathbf{x}_B|\mathbf{x}_C$ if every variable in A is d-separated from every variable in B conditioned on all the variables in C.
- To check if an independence is true, we can cycle through each node in A, do a depth-first search to reach every node in B, and examine the path between them. If *all* of the paths are d-separated, then we can assert $\mathbf{x}_A \perp \mathbf{x}_B|\mathbf{x}_C$.
- Thus, it will be sufficient to consider triples of nodes. (Why?)
- Pictorially, when we condition on a node, we shade it in.

Common Cause



$\mathbf{y}$ is the common cause of the two independent effects $\mathbf{x}$ and $\mathbf{z}$

- Q: When we condition on $\mathbf{y}$, are $\mathbf{x}$ and $\mathbf{z}$ independent?

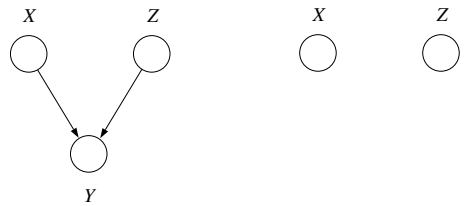
$$\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathsf{P}(\mathbf{y})\mathsf{P}(\mathbf{x}|\mathbf{y})\mathsf{P}(\mathbf{z}|\mathbf{y})$$

which implies

$$\begin{aligned}\mathsf{P}(\mathbf{x},\mathbf{z}|\mathbf{y}) &= \frac{\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z})}{\mathsf{P}(\mathbf{y})} \\ &= \frac{\mathsf{P}(\mathbf{y})\mathsf{P}(\mathbf{x}|\mathbf{y})\mathsf{P}(\mathbf{z}|\mathbf{y})}{\mathsf{P}(\mathbf{y})} \\ &= \mathsf{P}(\mathbf{x}|\mathbf{y})\mathsf{P}(\mathbf{z}|\mathbf{y})\end{aligned}$$

and therefore $\mathbf{x} \perp \mathbf{z}|\mathbf{y}$

Explaining Away



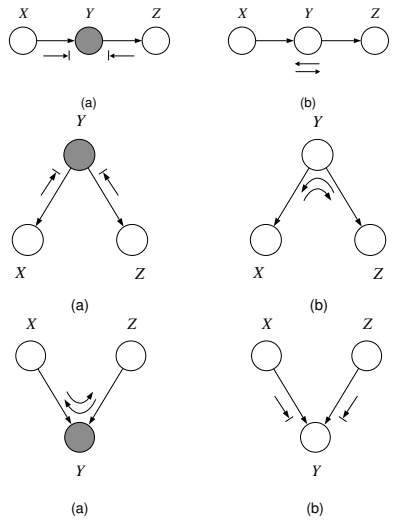
- Q: When we condition on $\mathbf{y}$, are $\mathbf{x}$ and $\mathbf{z}$ independent?

$$\mathsf{P}(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{z})\mathsf{P}(\mathbf{y}|\mathbf{x},\mathbf{z})$$

- $\mathbf{x}$ and $\mathbf{z}$ are *marginally independent*, but given $\mathbf{y}$ they are *conditionally dependent*.
- This important effect is called *explaining away* (Berkson's paradox.)
- For example, flip two coins independently; let $\mathbf{x}$=coin1,$\mathbf{z}$=coin2. Let $\mathbf{y}$=1 if the coins come up the same and $\mathbf{y}$=0 if different.
- $\mathbf{x}$ and $\mathbf{z}$ are independent, but if I tell you $\mathbf{y}$, they become coupled!

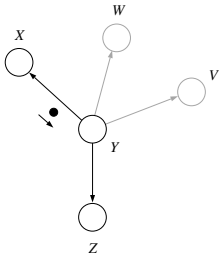
Bayes-Ball Rules

- The three cases we considered tell us rules:



Bayes Ball Algorithm

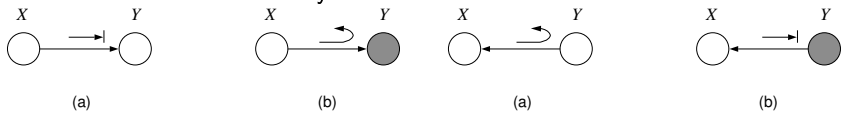
- To check if $\mathbf{x}_A \perp \mathbf{x}_B | \mathbf{x}_C$ we need to check if every variable in A is d-separated from every variable in B conditioned on all vars in C.
- In other words, given that all the nodes in $\mathbf{x}_C$ are clamped, when we wiggle nodes $\mathbf{x}_A$ can we change any of the node $\mathbf{x}_B$?
- The *Bayes-Ball Algorithm* is a such a d-separation test. We shade all nodes $\mathbf{x}_C$, place balls at each node in $\mathbf{x}_A$ (or $\mathbf{x}_B$), let them bounce around according to some rules, and then ask if any of the balls reach any of the nodes in $\mathbf{x}_B$ (or $\mathbf{x}_A$).



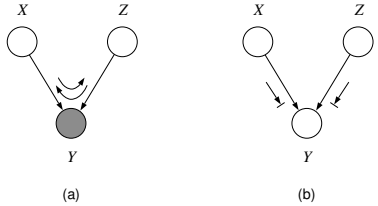
So we need to know what happens when a ball arrives at a node $\mathbf{Y}$ on its way from $\mathbf{X}$ to $\mathbf{Z}$.

Bayes-Ball Boundary Rules

- We also need the boundary conditions:

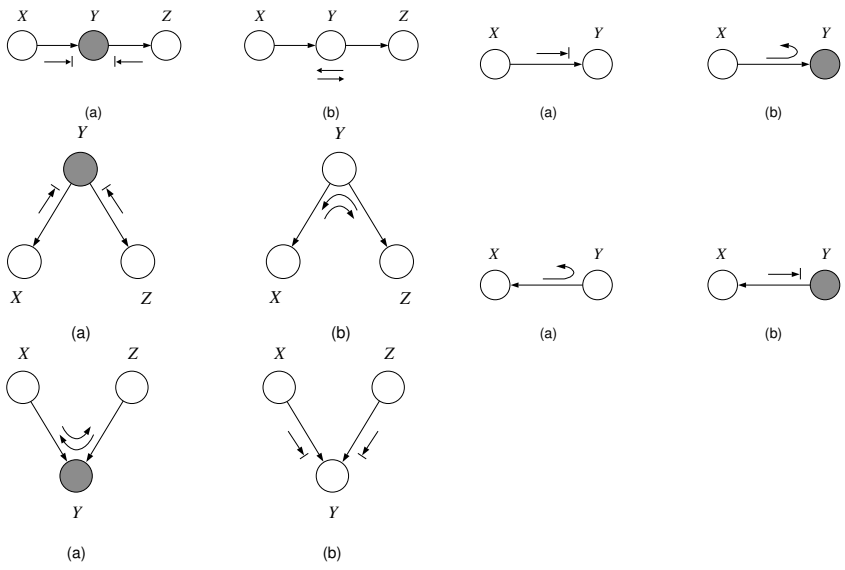


- Here's a trick for the explaining away case: If $\mathbf{y}$ *or any of its descendants* is shaded, the ball passes through.



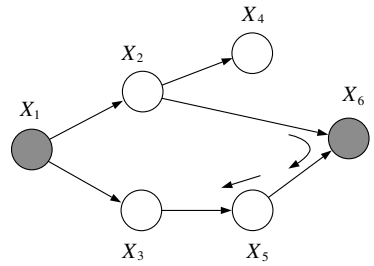
- Notice balls can travel opposite to edge directions.

Canonical Micrographs



Examples of Bayes-Ball Algorithm

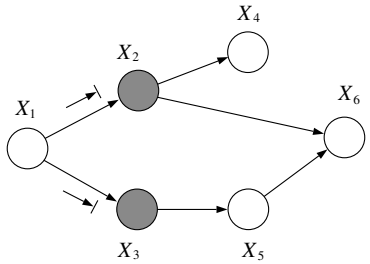
$$\mathbf{x}_2 \perp \mathbf{x}_3 | \{\mathbf{x}_1, \mathbf{x}_6\} \quad ?$$



Notice: balls can travel opposite to edge directions.

Examples of Bayes-Ball Algorithm

$$\mathbf{x}_1 \perp \mathbf{x}_6 | \{\mathbf{x}_2, \mathbf{x}_3\} \quad ?$$



Families of Distributions

- Consider two families of distributions.
- One is generated by all possible settings of the conditional probability tables in the DAGM form:

$$\mathsf{P}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \prod_i \mathsf{P}(\mathbf{x}_i | \mathbf{x}_{\pi_i})$$

- The other is generated by finding *all* the conditional independencies implied by a DAGM and eliminating any joint distributions which violate them.
- A version of the amazing *Hammersley-Clifford Theorem* (1971) states that these two families are exactly the same.