CSC 412/2506 Winter 2018

Probabilistic Learning and Reasoning

Lecture 3: Directed Graphical Models and Latent Variables

Based on slides by Richard Zemel

Learning outcomes

- What aspects of a model can we express using graphical notation?
- Which aspects are not captured in this way?
- How do independencies change as a result of conditioning?
- Reasons for using latent variables
- Common motifs such as mixtures and chains
- How to integrate out unobserved variables

Joint Probabilities

- Chain rule implies that any joint distribution equals

$$p(x_{1:D}) = p(x_1)p(x_2|x_1)p(x_3|x_1,x_2)p(x_4|x_1,x_2,x_3)...p(x_D|x_{1:D-1})$$

- Directed graphical model implies a restricted factorization

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:

$$P(\mathbf{x}_A, \mathbf{x}_B | \mathbf{x}_C) = P(\mathbf{x}_A | \mathbf{x}_C) P(\mathbf{x}_B | \mathbf{x}_C) \;;\; \forall \mathbf{x}_C$$

  which is equivalent to saying

$$P(\mathbf{x}_A | \mathbf{x}_B, \mathbf{x}_C) = P(\mathbf{x}_A | \mathbf{x}_C) \;;\; \forall \mathbf{x}_C$$

- 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 consistent with our assumptions*.
- Probabilistic graphical models are a powerful, elegant and simple way to specify such a family.

Directed Graphical Models

- Consider *directed acyclic graphs* over *n* variables.
- Each node has (possibly empty) set of parents π_i

- We can then write

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

- Hence we factorize the joint in terms of *local conditional probabilities*
- Exponential in "fan-in" of each node instead of in N

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:

$$\left\{\mathbf{x}_i \perp \mathbf{x}_{\tilde{\pi}_i} \middle| \mathbf{x}_{\pi_i}\right\} \; ; \; \forall i$$

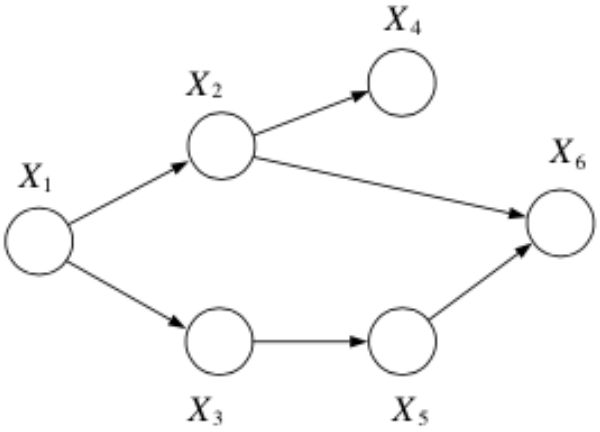
where $\mathbf{x}_{\tilde{\pi}_i}$ are the nodes coming before $\mathbf{x}_i$ that are not its parents

- 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

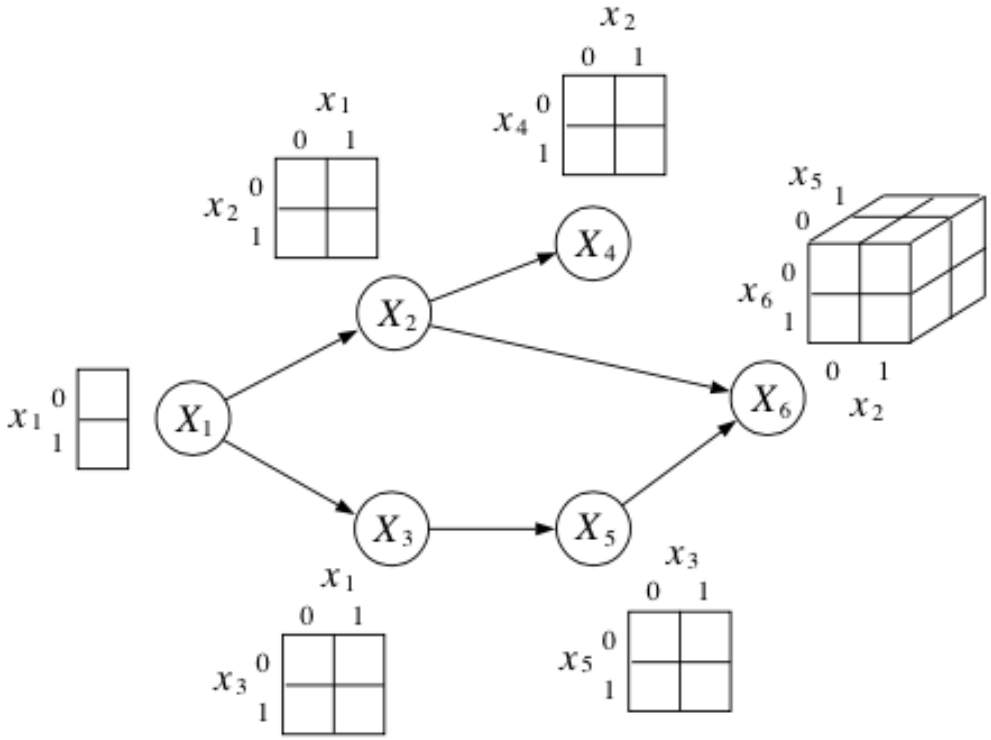
Example DAG

Consider this six node network:



The joint probability is now:

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



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:

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

- 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 $P(\mathbf{x}_i|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{i-1})$

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 $x_A \perp x_B | x_C$ 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 $x_A \perp x_B | x_C$
- Thus, it will be sufficient to consider triples of nodes. (Why?)
- Pictorially, when we condition on a node, we shade it in.

Chain

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

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

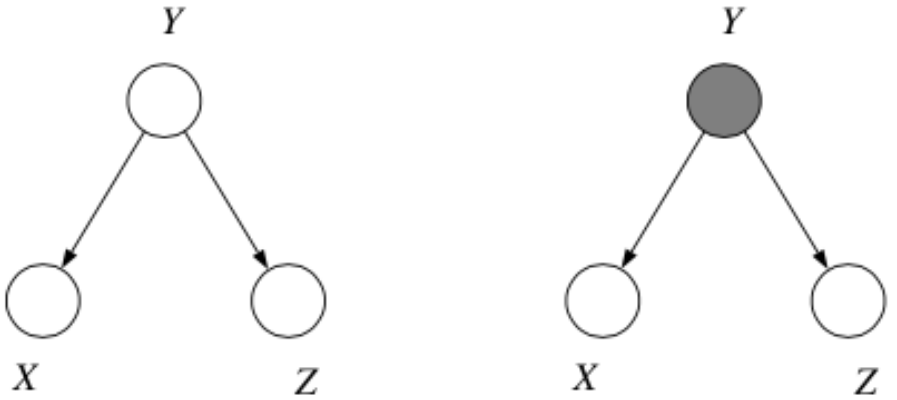
which implies

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

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

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

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?

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

which implies

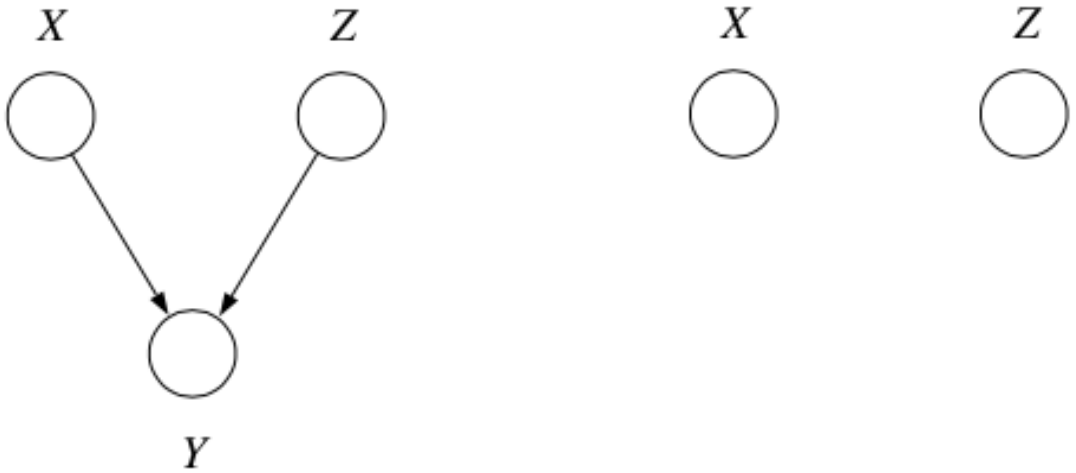
$$P(\mathbf{x}, \mathbf{z}|\mathbf{y}) = \frac{P(\mathbf{x}, \mathbf{y}, \mathbf{z})}{P(\mathbf{y})}$$

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

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

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

Explaining Away



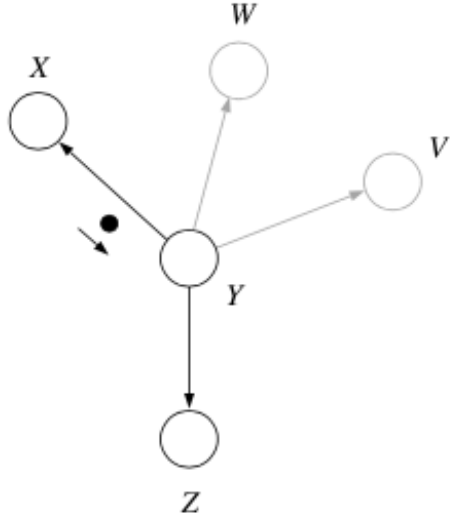
- Q: When we condition on **y**, are **x** and **z** independent?

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

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

Bayes-Ball Algorithm

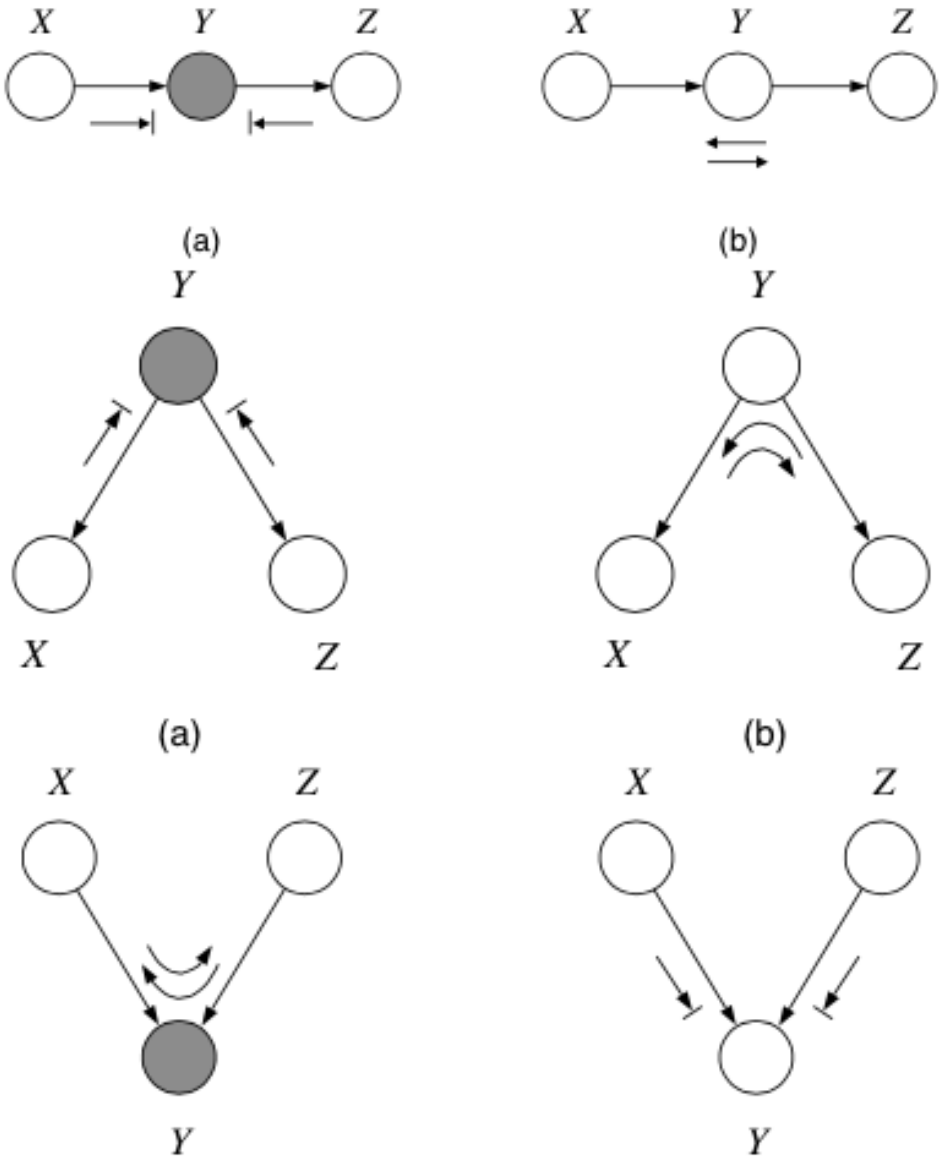
- To check if $\mathbf{x}_A \perp \mathbf{x}_B | \mathbf{x}_C$ we need to check if every variable in A is d-separated from every variable in B conditioned on all vars in C.
- In other words, given that all the nodes in $\mathbf{x}_C$ are clamped, when we wiggle nodes $\mathbf{x}_A$ can we change any of the nodes in $\mathbf{x}_B$?
- 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 y on its way from x to z.

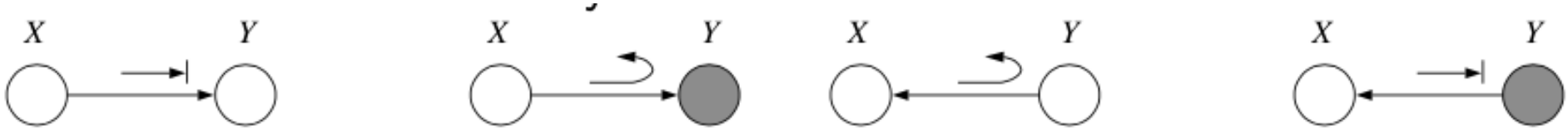
Bayes-Ball Rules

- The three cases we considered tell us rules:

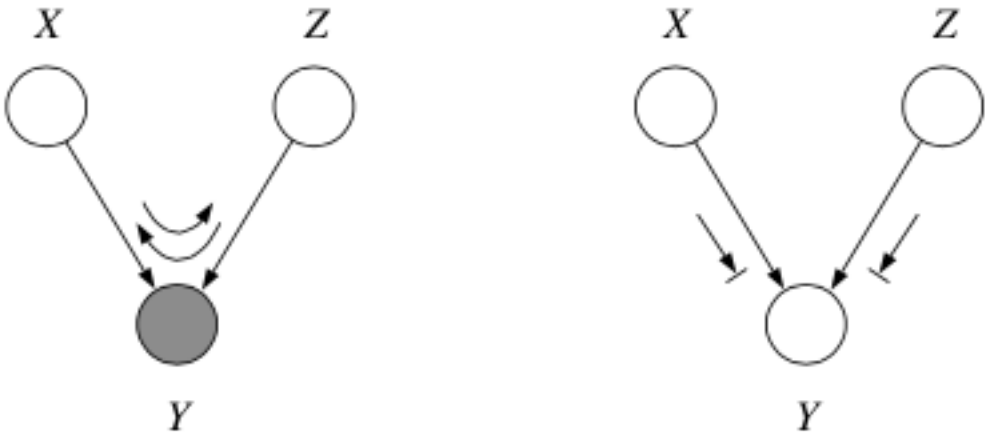


Bayes-Ball Boundary Rules

- We also need the boundary conditions:

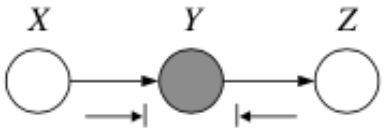


- Here's a trick for the explaining away case: If **y** *or any of its descendants* is shaded, the ball passes through.

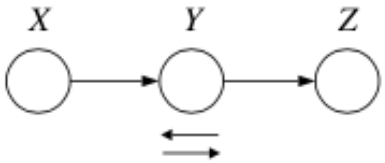


- Notice balls can travel opposite to edge directions.

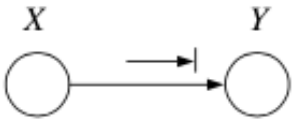
Canonical Micrographs



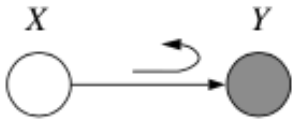
(a)



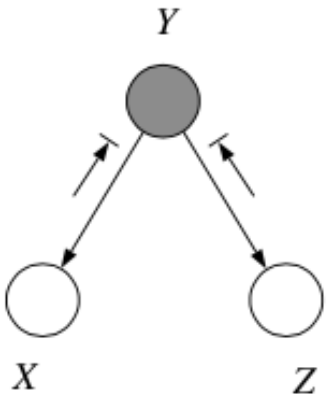
(b)



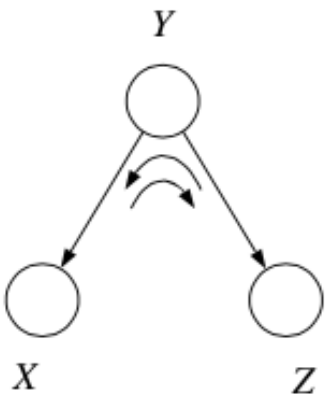
(a)



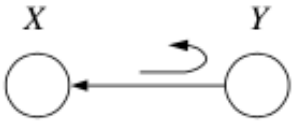
(b)



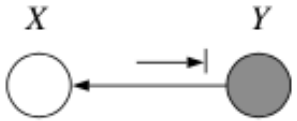
(a)



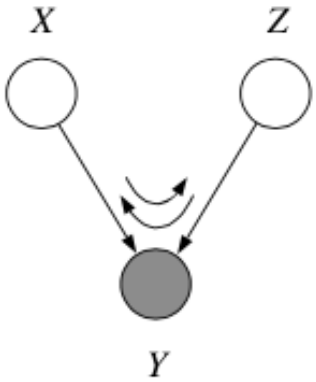
(b)

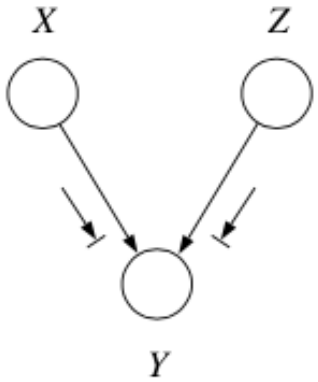


(a)



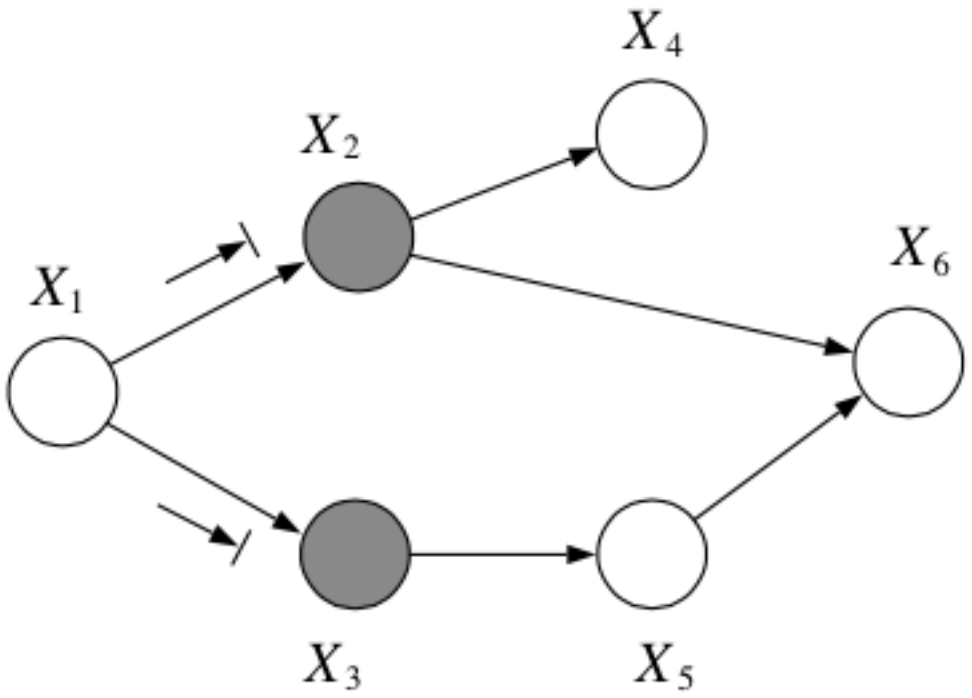
(b)





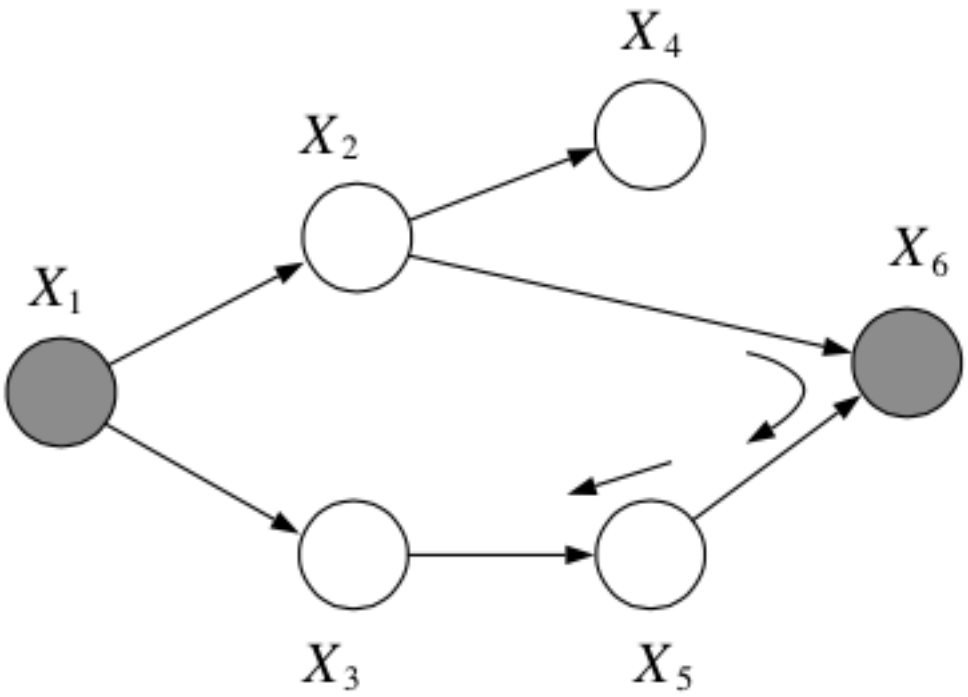
Examples of Bayes-Ball Algorithm

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



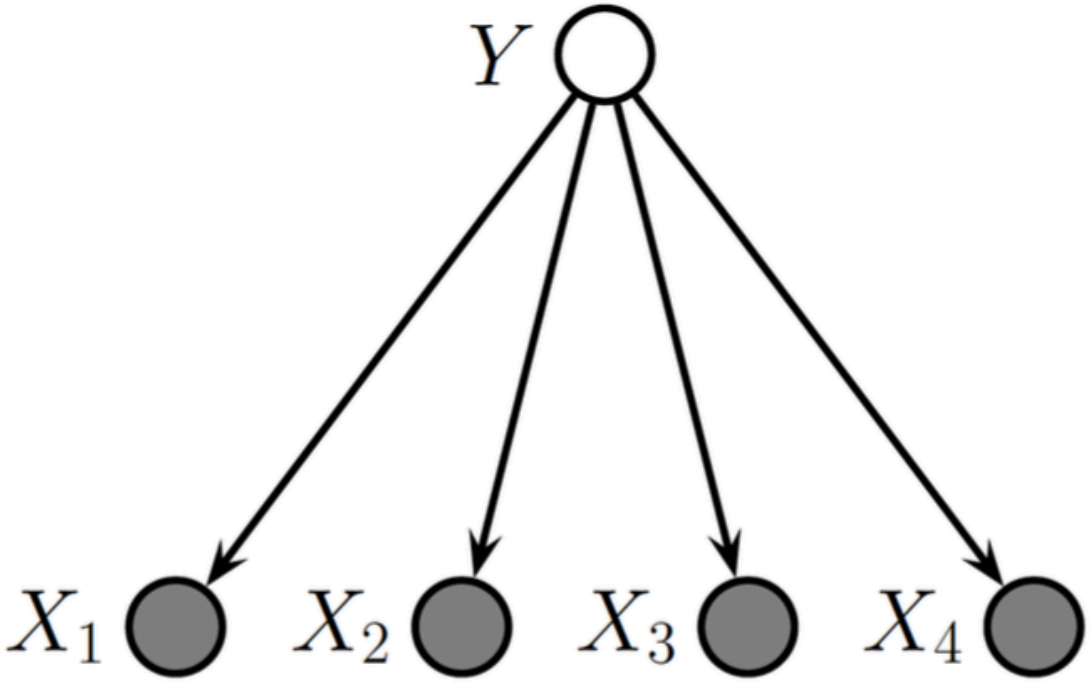
Examples of Bayes-Ball Algorithm

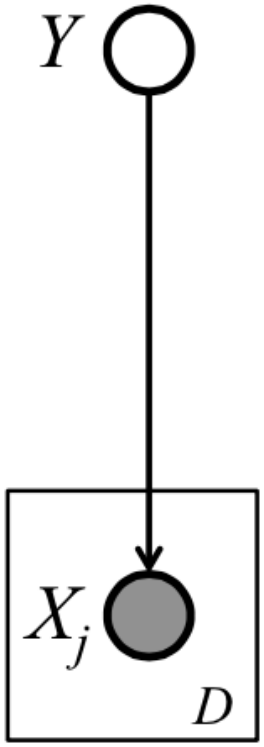
$$\mathbf{x}_2 \perp \mathbf{x}_3 | \{\mathbf{x}_1, \mathbf{x}_6\} \quad ?$$



- Notice: balls can travel opposite to edge direction

Plates





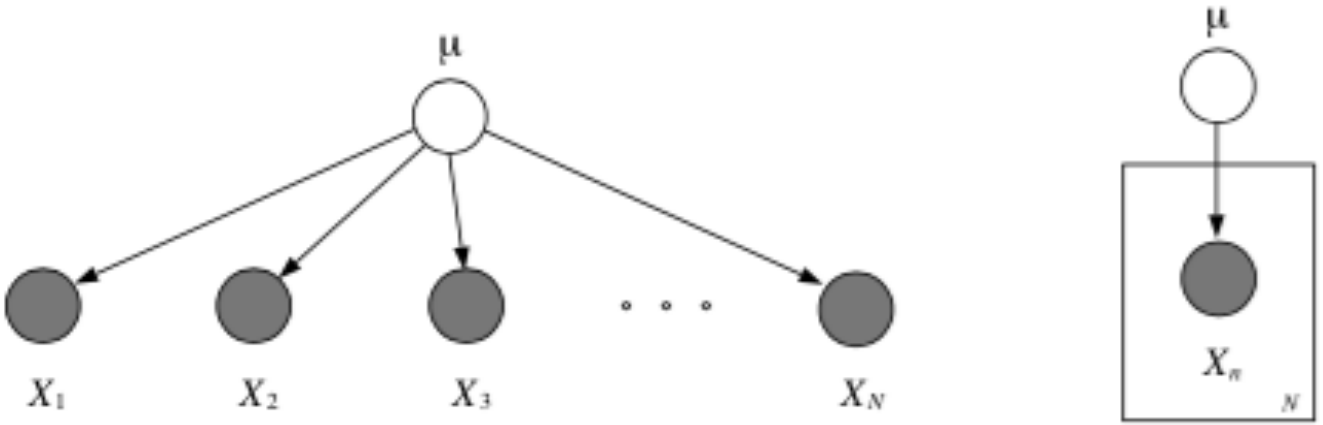
Plates denote replication of random variables

Plates & Parameters

- Since Bayesian methods treat parameters as random variables, we would like to include them in the graphical model.
- One way to do this is to repeat all the iid observations explicitly and show the parameter only once.
- A better way is to use plates, in which repeated quantities that are iid are put in a box.

Plates: Macros for Repeated Structures

- Plates are like "macros" that allow you to draw a very complicated graphical model with a simpler notation.
- The rules of plates are simple: repeat every structure in a box a number of times given by the integer in the corner of the box (e.g. *N*), updating the plate index variable (e.g. *n*) as you go.
- Duplicate every arrow going into the plate and every arrow leaving the plate by connecting the arrows to each copy of the structure.

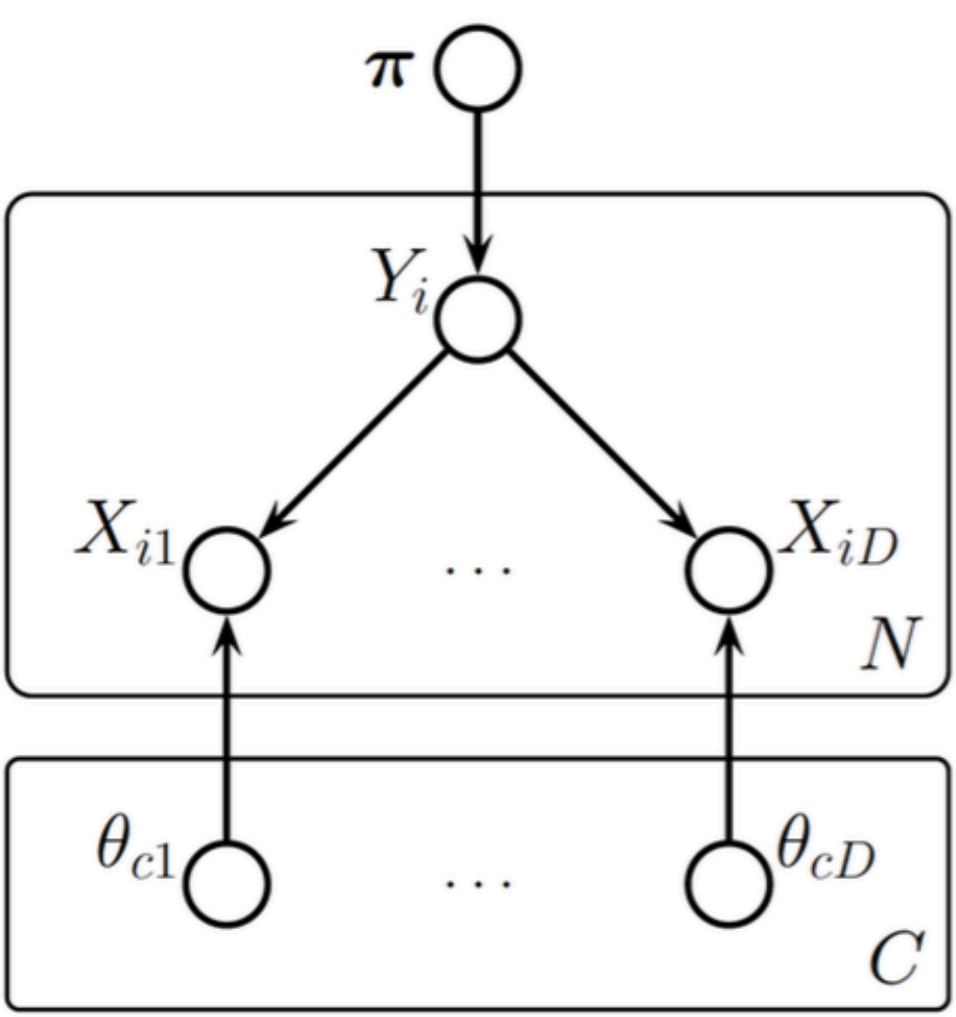


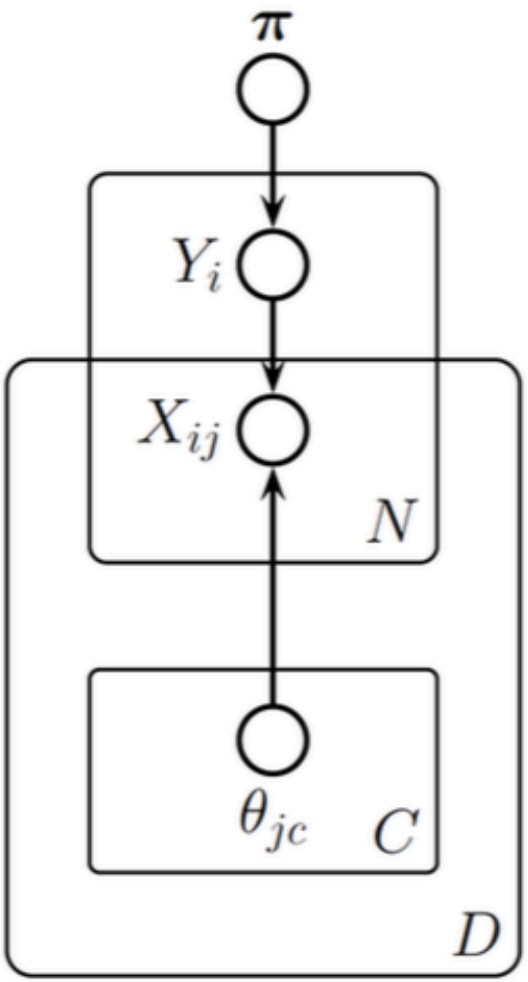
Nested/Intersecting Plates

- Plates can be nested, in which case their arrows get duplicated also, according to the rule: draw an arrow from every copy of the source node to every copy of the destination node.

- Plates can also cross (intersect), in which case the nodes at the intersection have multiple indices and get duplicated a number of times equal to the product of the duplication numbers on all the plates containing them.

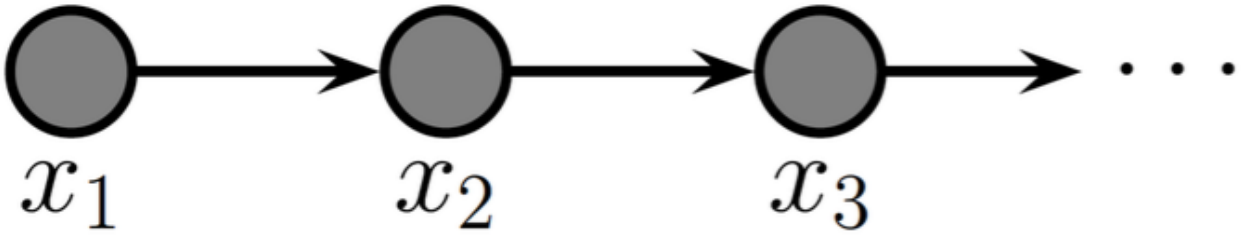
Example: Nested Plates





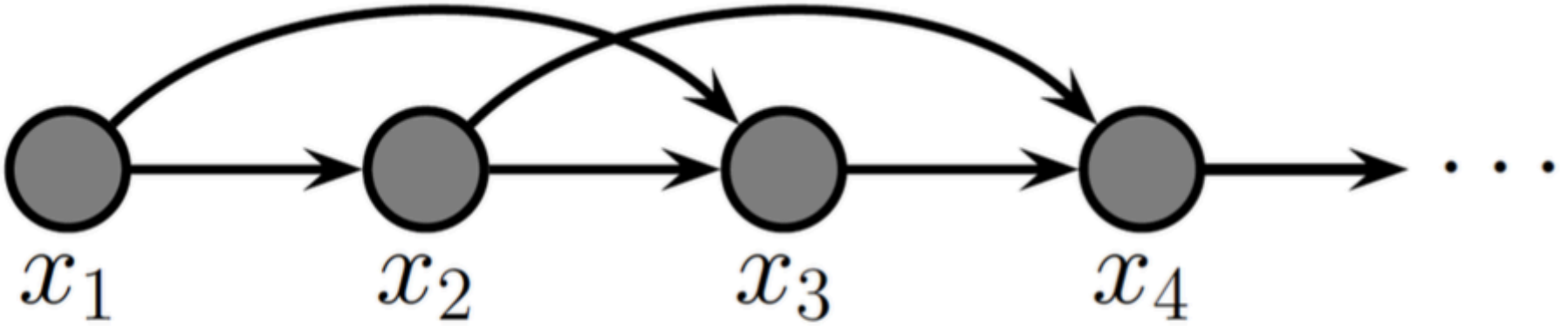
$$p(\pi)\left[\prod_{c=1}^{C}\prod_{j=1}^{D}p(\theta_{cj})\right]\prod_{i=1}^{N}\left[p(y_i \mid \pi)\prod_{j=1}^{D}p(x_{ij} \mid y_i, \theta_{j1}, \ldots, \theta_{jC})\right]$$

Example DAGM: Markov Chain



$$p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2)p(x_4 \mid x_3)\cdots$$

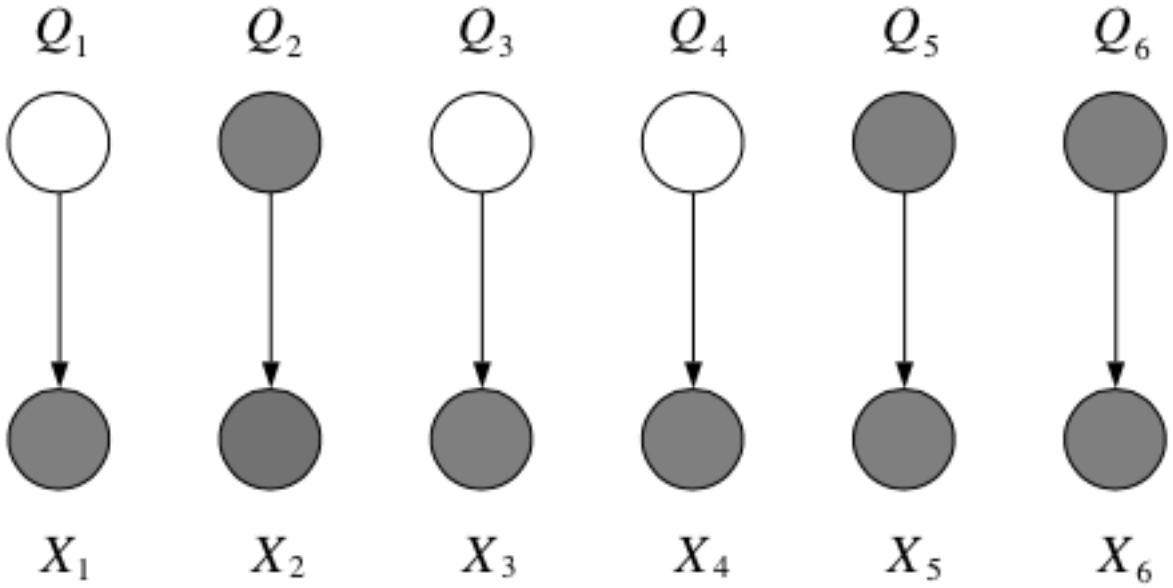
- *Markov Property*: Conditioned on the present, the past and future are independent



$$p(\mathbf{x}_{1:T}) = p(x_1, x_2)p(x_3|x_1, x_2)p(x_4|x_2, x_3)\ldots = p(x_1, x_2)\prod_{t=3}^{T} p(x_t|x_{t-1}, x_{t-2})$$

Unobserved Variables

- Certain variables Q in our models may be *unobserved*, either some of the time or always, either at training time or at test time



- Graphically, we will use shading to indicate observation

Partially Unobserved (Missing) Variables

- If variables are occasionally unobserved they are *missing data*, e.g., undefined inputs, missing class labels, erroneous target values
- In this case, we can still model the joint distribution, but we define a new cost function in which we sum out or marginalize the missing values at training or test time:

$$\ell(\theta; \mathcal{D}) = \sum_{\text{complete}} \log p(\mathbf{x}^c, \mathbf{y}^c | \theta) + \sum_{\text{missing}} \log p(\mathbf{x}^m | \theta)$$

$$= \sum_{\text{complete}} \log p(\mathbf{x}^c, \mathbf{y}^c | \theta) + \sum_{\text{missing}} \log \sum_{\mathbf{y}} p(\mathbf{x}^m, \mathbf{y} | \theta)$$

Recall that $p(x) = \sum_q p(x, q)$

Latent Variables

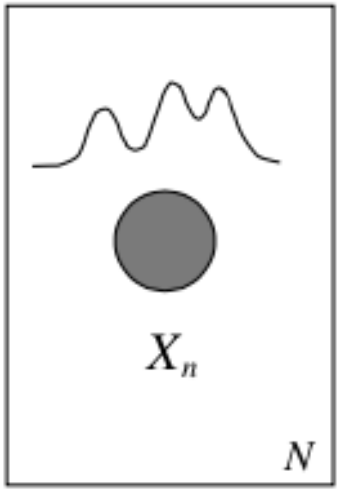
- What to do when a variable $\mathbf{z}$ is always unobserved? Depends on where it appears in our model. If we never condition on it when computing the probability of the variables we *do* observe, then we can just forget about it and integrate it out. e.g., given $\mathbf{y}, \mathbf{x}$ fit the model $p(\mathbf{z}, \mathbf{y}|\mathbf{x}) = p(\mathbf{z}|\mathbf{y})p(\mathbf{y}|\mathbf{x}, \mathbf{w})p(\mathbf{w})$. (In other words if it is a leaf node.)

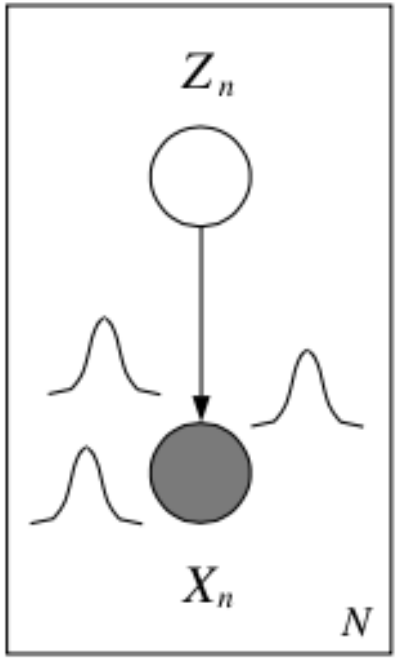
- But if z is conditioned on, we need to m□odel it:

  e.g. given $\mathbf{y}, \mathbf{x}$ fit the model $p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{y}|\mathbf{x}, \mathbf{z})p(\mathbf{z})$

Where Do Latent Variables Come From?

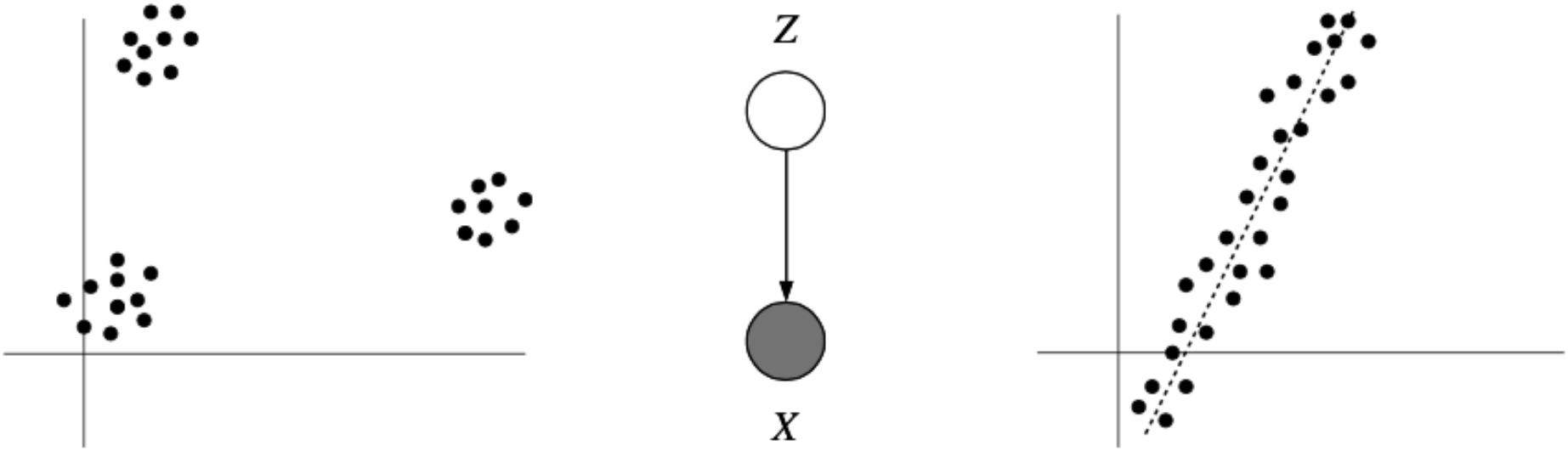
- Latent variables may appear naturally, from the structure of the problem, because something wasn't measured, because of faulty sensors, occlusion, privacy, etc.
- But also, we may want to intentionally introduce latent variables to model complex dependencies between variables without looking at the dependencies between them directly.
  This can actually simplify the model (e.g., mixtures).





Latent Variables Models & Regression

- You can think of clustering as the problem of classification with missing class labels.



- You can think of factor models (such as factor analysis, PCA, ICA, etc.) as linear or nonlinear regression with missing inputs.

Why is Learning Harder?

- In fully observed iid settings, the probability model is a product, thus the log likelihood is a sum where terms decouple.
  (At least for directed models.)

$$\ell(\theta; \mathcal{D}) = \log p(\mathbf{x}, \mathbf{z}|\theta)$$

$$= \log p(\mathbf{z}|\theta_z) + \log p(\mathbf{x}|\mathbf{z}, \theta_x)$$

- With latent variables, the probability already contains a sum, so the log likelihood has all parameters coupled together via z:

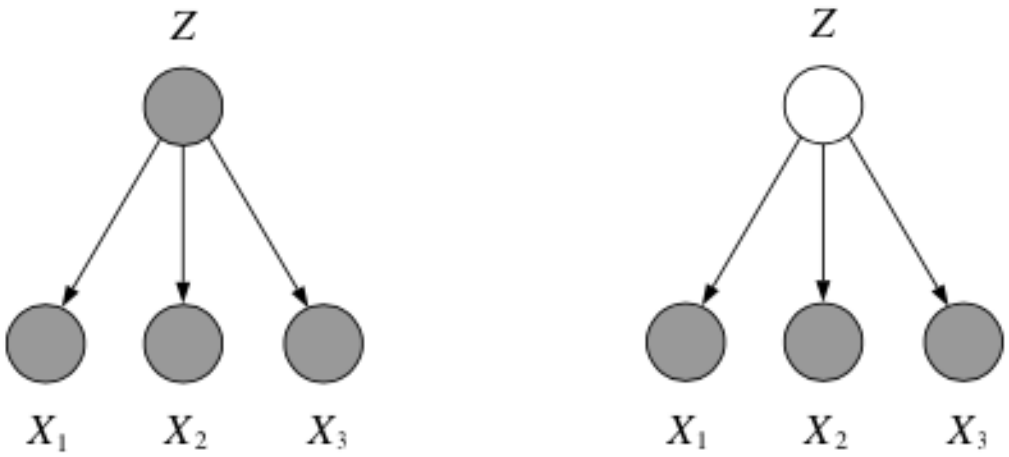
$$\ell(\theta; \mathcal{D}) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\theta)$$

$$= \log \sum_{\mathbf{z}} p(\mathbf{z}|\theta_z) + \log p(\mathbf{x}|\mathbf{z}, \theta_x)$$

(Just as with the partition function in undirected models)

Why is Learning Harder?

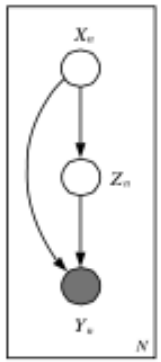
- Likelihood $\log \sum_{\mathbf{z}} p(\mathbf{z}|\theta_z) + \log p(\mathbf{x}|\mathbf{z}, \theta_x)$ couples parameters:

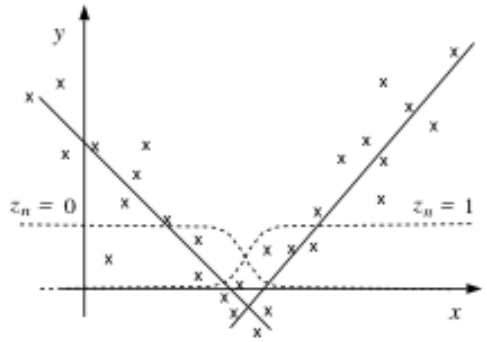


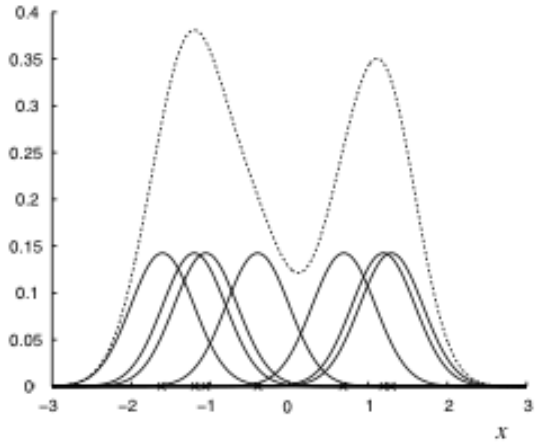
- We can treat this as a black box probability function and just try to optimize the likelihood as a function of θ (e.g. gradient descent). However, sometimes taking advantage of the latent variable structure can make parameter estimation easier.
- Good news: soon we will see how to deal with latent variables.
- Basic trick: put a tractable distribution on the values you don't know.
  Basic math: use convexity to lower bound the likelihood.

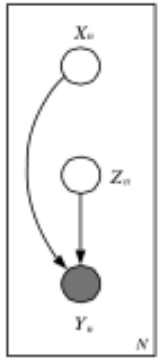
Mixture Models

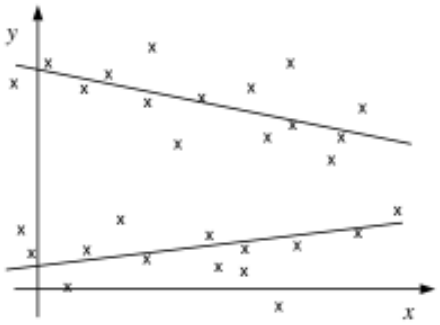
- Most basic latent variable model with a single discrete node *z*.
- Allows different submodels (experts) to contribute to the (conditional) density model in different parts of the space.
- Divide & conquer idea: use simple parts to build complex models (e.g., multimodal densities, or piecewise-linear regressions).

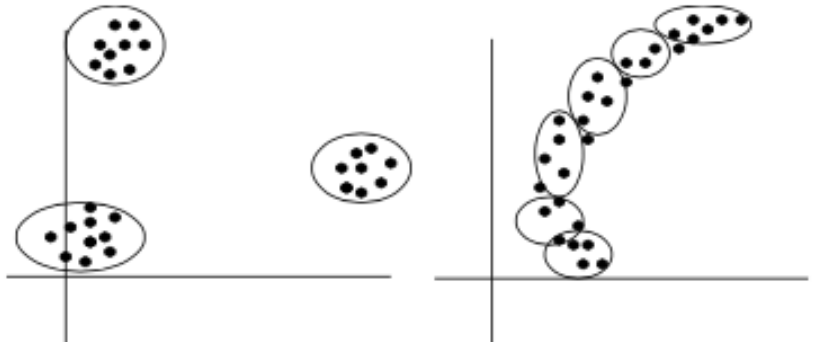












Mixture Densities

- Exactly like a classification model but the class is unobserved and so we sum it out. What we get is a perfectly valid density:

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} p(z=k|\theta_z)p(\mathbf{x}|z=k,\theta_k)$$
$$= \sum_{k} \alpha_k p_k(\mathbf{x}|\theta_k)$$

where the "mixing proportions" add to one: $\sum_k \alpha_k = 1$.

- We can use Bayes' rule to compute the posterior probability of the mixture component given some data:

$$p(z=k|\mathbf{x},\theta) = \frac{\alpha_k p_k(\mathbf{x}|\theta_k)}{\sum_j \alpha_j p_j(\mathbf{x}|\theta_j)}$$

these quantities are called *responsibilities*.

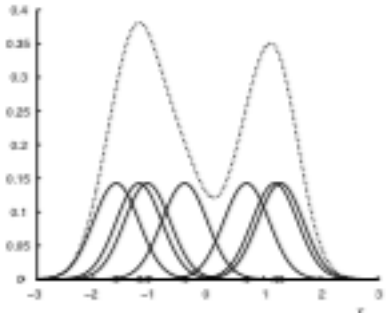
Example: Gaussian Mixture Models

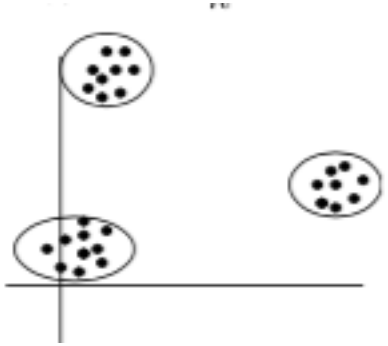
- Consider a mixture of K Gaussian components:

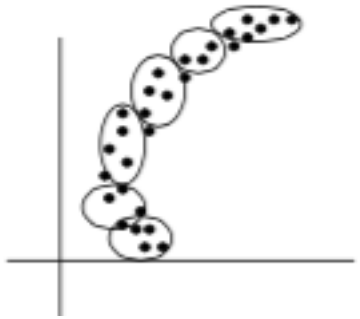
$$p(\mathbf{x}|\theta) = \sum_k \alpha_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$$p(z = k|\mathbf{x}, \theta) = \frac{\alpha_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)}{\sum_j \alpha_j \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)}$$

$$\ell(\theta; \mathcal{D}) = \sum_n \log \sum_k \alpha_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k)$$







- Density model: $p(x|\theta)$ is a familiarity signal.
  Clustering: $p(z|\mathbf{x}, \theta)$ is the assignment rule, $-l(\theta)$ is the cost.

Example: Mixtures of Experts

- Also called conditional mixtures. Exactly like a class-conditional model but the class is unobserved and so we sum it out again:

$$p(\mathbf{y}|\mathbf{x},\theta) = \sum_{k=1}^{K} p(z=k|\mathbf{x},\theta_z)p(\mathbf{y}|z=k,\mathbf{x},\theta_k)$$
$$= \sum_{k} \alpha_k(\mathbf{x}|\theta_z)p_k(\mathbf{y}|\mathbf{x},\theta_k)$$

where $\sum_k \alpha_k(\mathbf{x}) = 1 \quad \forall \mathbf{x}$

- Harder: must learn α(x) (unless chose z independent of x).
- We can still use Bayes' rule to compute the posterior probability of the mixture component given some data:

$$p(z=k|\mathbf{x},\mathbf{y},\theta) = \frac{\alpha_k(\mathbf{x})p_k(\mathbf{y}|\mathbf{x},\theta_k)}{\sum_j \alpha_j(\mathbf{x})p_j(\mathbf{y}|\mathbf{x},\theta_j)}$$

this function is often called the *gating function*.

Example: Mixtures of Linear Regression Experts

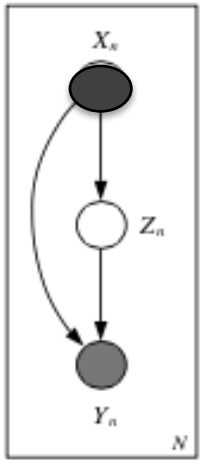
- Each expert generates data according to a linear function of the input plus additive Gaussian ?noise:

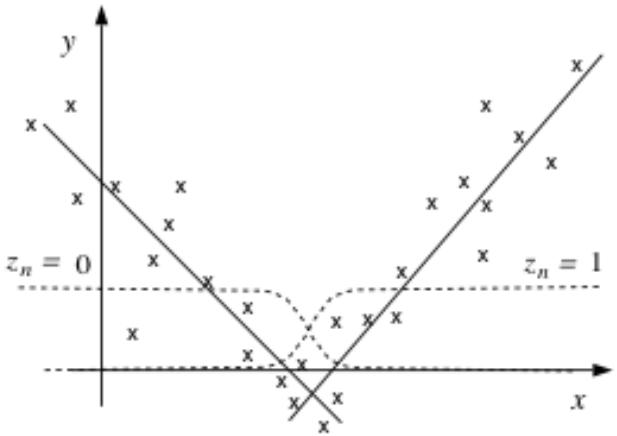
$$p(y|\mathbf{x},\theta) = \sum_k \alpha_k \mathcal{N}(y|\beta_k^T\mathbf{x}, \sigma_k^2)$$

- The "gate" function can be a softmax classification machine

$$\alpha_k(\mathbf{x}) = p(z=k|\mathbf{x}) = \frac{e^{\eta_k^T\mathbf{x}}}{\sum_j e^{\eta_j^T\mathbf{x}}}$$

- Remember: we are not modeling the density of the inputs x





Gradient Learning with Mixtures

- We can learn mixture densities using gradient descent on the likelihood as usual. The gradients are quite interesting:

$$\ell(\theta) = \log p(\mathbf{x}|\theta) = \log \sum_k \alpha_k p_k(\mathbf{x}|\theta_k)$$

$$\frac{\partial \ell}{\partial \theta} = \frac{1}{p(\mathbf{x}|\theta)} \sum_k \alpha_k \frac{\partial p_k(\mathbf{x}|\theta_k)}{\partial \theta}$$

$$= \sum_k \alpha_k \frac{1}{p(\mathbf{x}|\theta)} p_k(\mathbf{x}|\theta_k) \frac{\partial \log p_k(\mathbf{x}|\theta_k)}{\partial \theta}$$

$$= \sum_k \alpha_k \frac{p_k(\mathbf{x}|\theta_k)}{p(\mathbf{x}|\theta)} \frac{\partial \ell_k}{\partial \theta_k} = \sum_k \alpha_k r_k \frac{\partial \ell_k}{\partial \theta_k}$$

- In other words, the gradient is the *responsibility weighted sum* of the individual log likelihood gradients

Parameter Constraints

- If we want to use general optimizations (e.g., conjugate gradient) to learn latent variable models, we often have to make sure parameters respect certain constraints (e.g., $\sum_k \alpha_k = 1$, $\sum_k \alpha_k$ positive definite)
- A good trick is to re-parameterize these quantities in terms of unconstrained values. For mixing proportions, use the softmax:

$$\alpha_k = \frac{\exp(q_k)}{\sum_j \exp(q_j)}$$

- For covariance matrices, use the Cholesky decomposition

$$\Sigma^{-1} = A^T A \qquad |\Sigma|^{-1/2} = \prod_i A_{ii}$$

where A is upper diagonal with positive diagonal

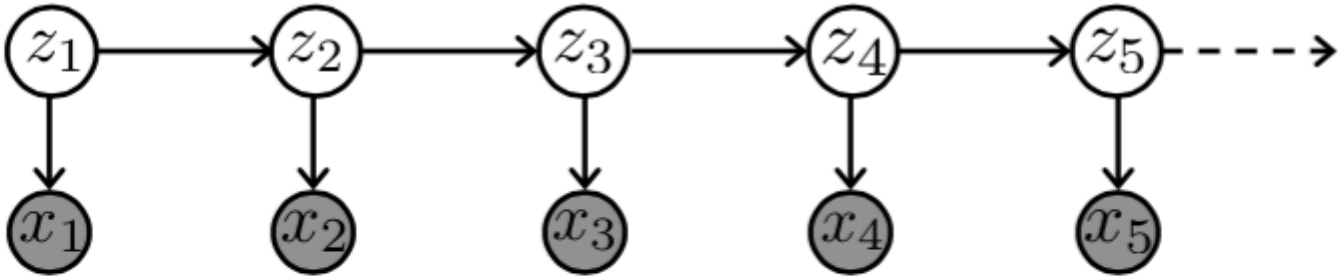
$$A_{ii} = \exp(r_i) > 0 \quad A_{ij} = a_{ij} \ (j > i) \quad A_{ij} = 0 \ (j < i)$$

Logsumexp

- Often you can easily compute $b_k = \log p(\mathbf{x}/z = k, \theta_k)$,
  but it will be very negative, say -10^6 or smaller.
- Now, to compute $l = \log p(\mathbf{x}/\theta)$ you need to compute $\log \sum_k e^{b_k}$ (e.g., for calculating responsibilities at test time or for learning)
- Careful! Do not compute this by doing log(sum(exp(b))). You will get underflow and an incorrect answer.
- Instead do this:
  - Add a constant exponent B to all the values b_k such that the largest value equals zero: B = max(b).
  - Compute log(sum(exp(b - B))) + B.
- Example: if $\log p(x/z = 1) = -120$ and $\log p(x/z = 2) = -120$, what is $\log p(x) = \log [p(x/z = 1) + p(x/z = 2)]$?
  Answer: $\log[2e^{-120}] = -120 + \log 2$.
- Rule of thumb: never use log or exp by itself

Hidden Markov Models (HMMs)

- A very popular form of latent variable model



$$p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) = p(\mathbf{z}_{1:T})p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) = \left[p(z_1)\prod_{t=2}^{T} p(z_t|z_{t-1})\right]\left[\prod_{t=1}^{T} p(\mathbf{x}_t|z_t)\right]$$

- Z_t → Hidden states taking one of K discrete values
- X_t → Observations taking values in any space

Example: discrete, M observation symbols $B \in \Re^{K \times M}$

$$p(x_t = j|z_t = k) = B_{kj}$$

Inference in Graphical Models

x_E → Observed evidence variables (subset of nodes)

x_F → unobserved query nodes we'd like to infer

x_R → remaining variables, extraneous to this query but part of the given graphical representation

$$p(x_E, x_F) = \sum_{x_R} p(x_E, x_F, x_R) \qquad R = V \setminus \{E, F\}$$

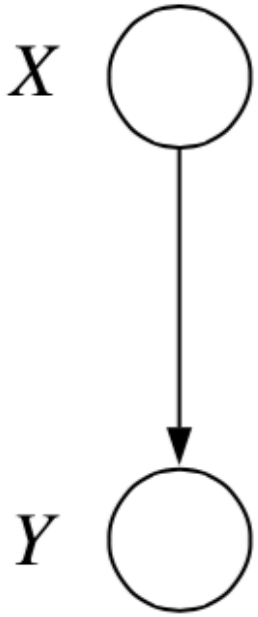
Maximum a Posteriori (MAP) Estimates

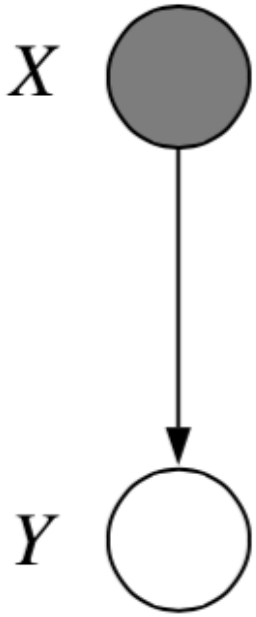
$$\hat{x}_F = \arg\max_{x_F} p(x_F \mid x_E) = \arg\max_{x_F} p(x_E, x_F)$$

Posterior Marginal Densities

$$p(x_F \mid x_E) = \frac{p(x_E, x_F)}{p(x_E)} \qquad p(x_E) = \sum_{x_F} p(x_E, x_F)$$

Inference with Two Variables





$$p(x, y) = p(x)p(y \mid x)$$

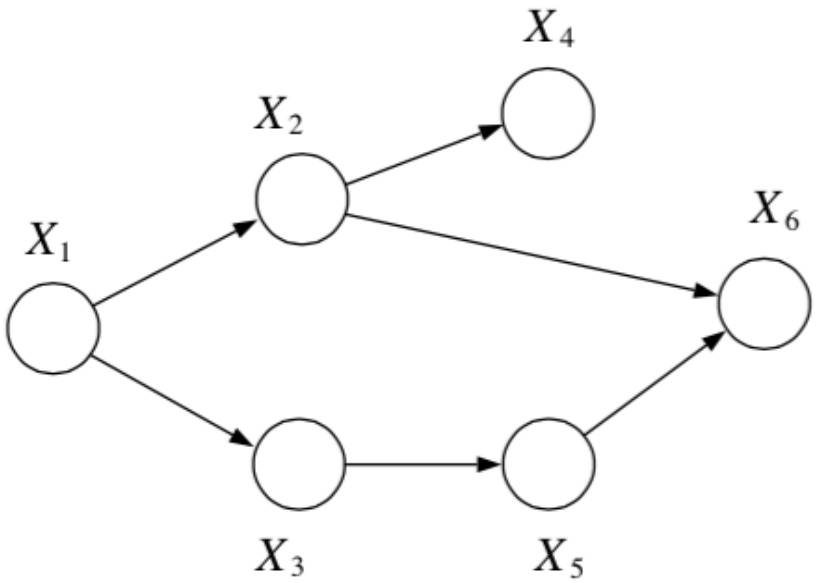
Table look-up

$$p(y|x = \bar{x})$$

Bayes' Rule

$$p(x|y = \bar{y}) = \frac{p(\bar{y}|x)p(x)}{p(\bar{y})}$$

Naïve Inference



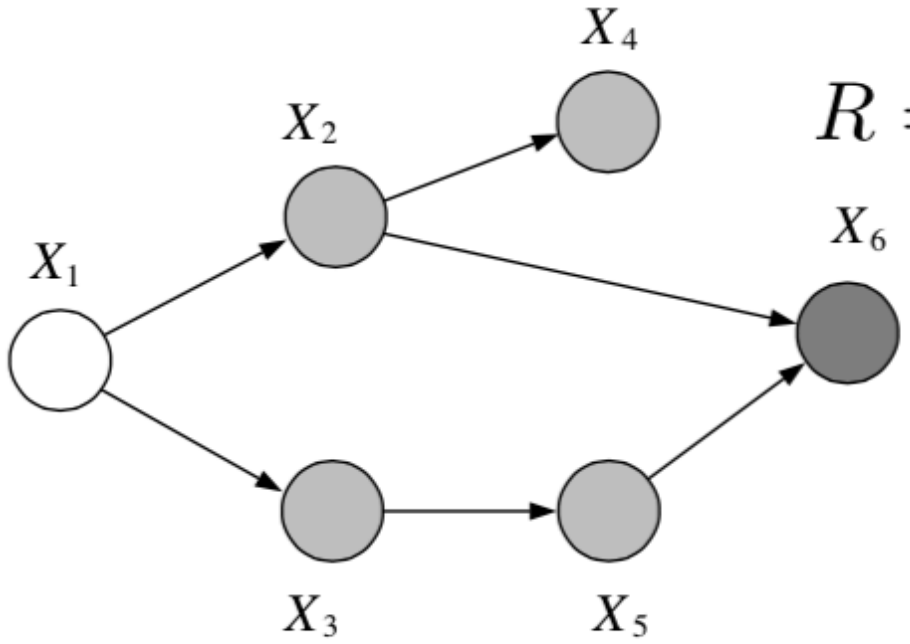
- Suppose each variable takes one of *k* discrete values

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

- Costs O*(k)* operations to update each of O*(k^5)* table entries
- Use factorization and distributed law to reduce complexity

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

Inference in Directed Graphs



$F = \{1\}$

$p(x_1 \mid \bar{x}_6) = ?$

$R = \{2, 3, 4, 5\}$

$E = \{6\}$

$X_6 = \bar{x}_6$

$$p(x_1, \bar{x}_6) = \sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_5} p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(\bar{x}_6 \mid x_2, x_5)$$

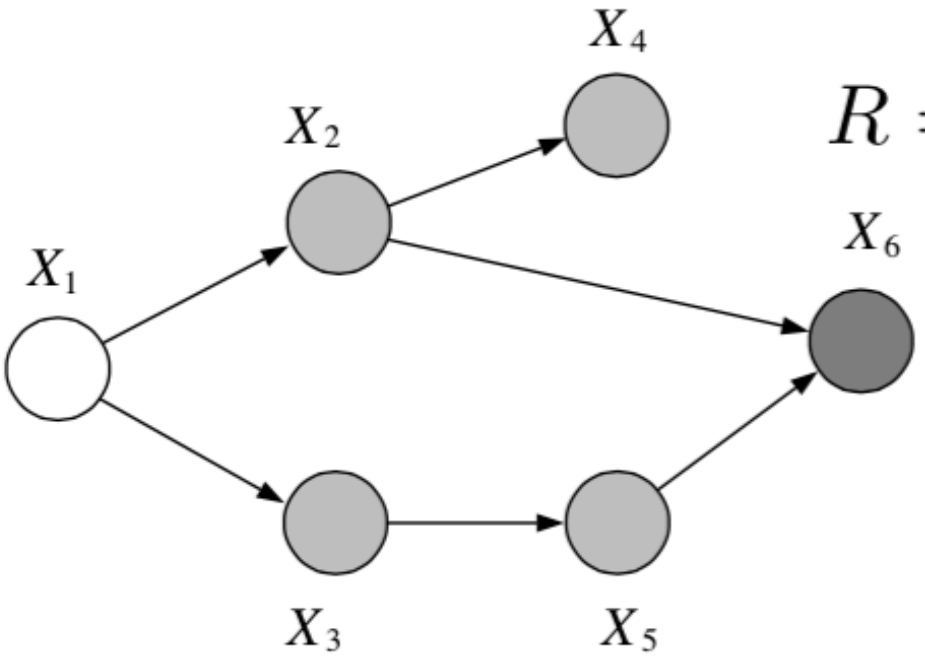
$$= p(x_1)\sum_{x_2} p(x_2 \mid x_1)\sum_{x_3} p(x_3 \mid x_1)\sum_{x_4} p(x_4 \mid x_2)\sum_{x_5} p(x_5 \mid x_3)p(\bar{x}_6 \mid x_2, x_5)$$

$$= p(x_1)\sum_{x_2} p(x_2 \mid x_1)\sum_{x_3} p(x_3 \mid x_1)\sum_{x_4} p(x_4 \mid x_2)m_5(x_2, x_3)$$

Inference in Directed Graphs

$F = \{1\}$

$p(x_1 \mid \bar{x}_6) = ?$



$R = \{2, 3, 4, 5\}$

$E = \{6\}$

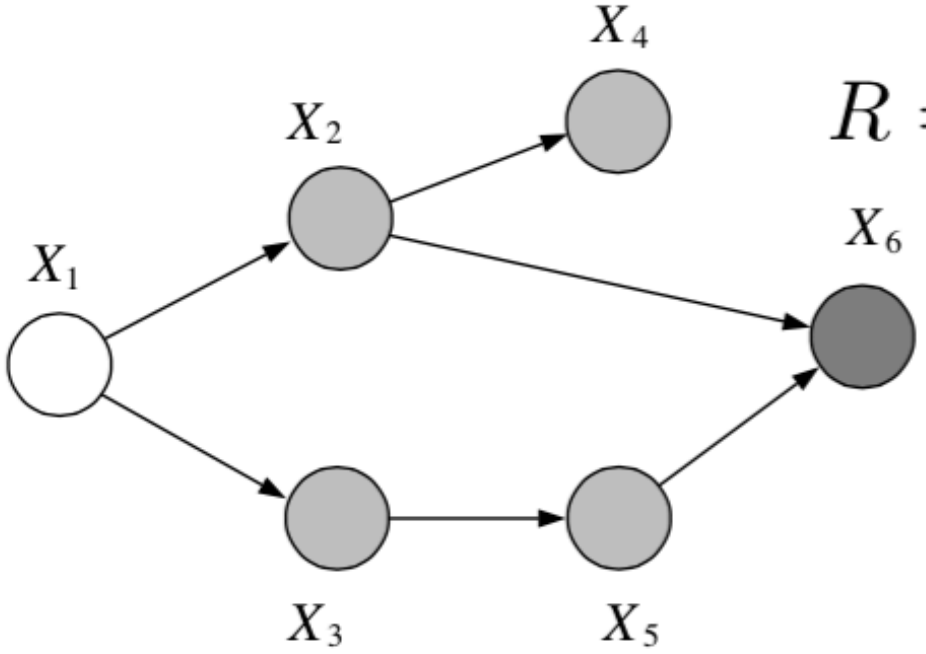
$X_6 = \bar{x}_6$

$$p(x_1, \bar{x}_6) = \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} p(x_1) p(x_2 \mid x_1) p(x_3 \mid x_1) p(x_4 \mid x_2) p(x_5 \mid x_3) p(\bar{x}_6 \mid x_2, x_5)$$

$$= p(x_1) \sum_{x_2} p(x_2 \mid x_1) \sum_{x_3} p(x_3 \mid x_1) m_5(x_2, x_3) \sum_{x_4} p(x_4 \mid x_2)$$

$$= p(x_1) \sum_{x_2} p(x_2 \mid x_1) m_4(x_2) \sum_{x_3} p(x_3 \mid x_1) m_5(x_2, x_3).$$

Inference in Directed Graphs



$R = \{2, 3, 4, 5\}$

$F = \{1\}$

$E = \{6\}$

$p(x_1 \mid \bar{x}_6) = ?$

$X_6 = \bar{x}_6$

$$
\begin{aligned}
p(x_1, \bar{x}_6) &= \sum_{x_2}\sum_{x_3}\sum_{x_4}\sum_{x_5} p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(\bar{x}_6 \mid x_2, x_5) \\
&= p(x_1)\sum_{x_2} p(x_2 \mid x_1)m_4(x_2)m_3(x_1, x_2) \\
&= p(x_1)m_2(x_1).
\end{aligned}
$$

$$p(x_1 \mid \bar{x}_6) = \frac{p(x_1)m_2(x_1)}{\sum_{x_1} p(x_1)m_2(x_1)}$$

$$p(\bar{x}_6) = \sum_{x_1} p(x_1)m_2(x_1)$$

Learning outcomes

- What aspects of a model can we express using graphical notation?
- Which aspects are not captured in this way?
- How do independencies change as a result of conditioning?
- Reasons for using latent variables
- Common motifs such as mixtures and chains
- How to integrate out unobserved variables

Questions?

- Thursday: Tutorial on automatic differentiation
- This week: Assignment 1 released