CPSC 440/540: Advanced Machine Learning

More MCMC; DAGs

Danica Sutherland (building on materials from Mark Schmidt)

University of British Columbia

Winter 2023

Last Time: Start of MCMC

- Want approximate samples from $\tilde{p} \propto p$, to estimate $\mathbb{E}_{X \sim p} f(X)$
  - Construct a Markov chain with stationary distribution p
  - Run for a long time, will get approximate samples from p
  - Burn-in period, and samples are highly correlated (sometimes thin them)
- Metropolis algorithm:
  - Start at x^0
  - Propose $\hat{x}^t = x^{t-1} + \mathcal{N}(0, \Sigma)$
  - Accept $(x^t = \hat{x}^t)$ with probability $\max\left(1, \tilde{p}(\hat{x}^t)/\tilde{p}(x^{t-1})\right)$
  - Otherwise reject, $x^t = x^{t-1}$
- Satisfies the detailed balance condition for reversibility:

$$\pi(s) q_{s \to s'} = \pi(s') q_{s' \to s} \qquad \text{for } \pi = p$$

  - Implies π is a stationary distribution of the chain – is unique if chain is ergodic

Metropolis-Hastings

- Metropolis algorithm is a special case of Metropolis-Hastings.
  - General version uses a general proposal distribution $q(\hat{x}^{t+1} \mid x^t) = q_{x^t \to \hat{x}^{t+1}}$.
  - In Metropolis, q is a Gaussian with mean x^t.

- Metropolis-Hastings accepts a proposed $\hat{x}^t$ if

$$u \leq \frac{\tilde{p}(\hat{x}^t)}{\tilde{p}(x^{t-1})} \cdot \frac{q(\hat{x}^t \to x^{t-1})}{q(x^{t-1} \to \hat{x}^t)}.$$

- These extra terms ensures reversibility (detailed balance) for asymmetric q.
  - If you're more likely to propose $x^{t-1} \to \hat{x}^t$ than the other way, less likely to accept.

- Eventually converges under very weak conditions, e.g. all $q(x^t \to \hat{x}^{t+1}) > 0$.
  - But practical convergence can change a lot with different q.

Metropolis-Hastings Example: Rolling Dice with Coins

- Say we want to sample from a fair 6-sided die.
  - $\Pr(X = c) = \frac{1}{6}$ for each $c \in \{1, \ldots, 6\}$.
  - But we don't have a die or a computer, just coins.

- Consider the following random walk on the numbers 1-6:
  - If $x = 1$, always propose 2.
  - If $x = 2$, 50% of the time propose 1 and 50% of the time propose 3.
  - If $x = 3$, 50% of the time propose 2 and 50% of the time propose 4.
  - If $x = 4$, 50% of the time propose 3 and 50% of the time propose 5.
  - If $x = 5$, 50% of the time propose 4 and 50% of the time propose 6.
  - If $x = 6$, always propose 5.

- Flip a coin: go up if it's heads, go down it it's tails.
  - Like a PageRank "random surfer" applied to this graph:

Metropolis-Hastings Example: Rolling Dice with Coins

- "Roll a die with a coin" by using random walk as transitions q in M-H:
  - $q_{1 \to 2} = 1$, $q_{2 \to 1} = \frac{1}{2}$, $q_{2 \to 3} = \frac{1}{2}$, $\ldots$, $q_{6 \to 5} = 1$

- If x is in the "middle" (2-5), we'll always accept the random walk.
  - If $x = 3$ and we propose $\hat{x} = 2$, then:

$$u < \frac{p(2)}{p(3)} \cdot \frac{q_{2 \to 3}}{q_{3 \to 2}} = \frac{1/6}{1/6} \cdot \frac{1/2}{1/2} = 1.$$

  - If $x = 2$ and we propose $\hat{x} = 1$, then we test $u < 2$ which is also always true.

  - If x is at the end $(1$ or $6)$, you accept with probability $1/2$:

$$u < \frac{p(2)}{p(1)} \cdot \frac{q_{2 \to 1}}{q_{1 \to 2}} = \frac{1/6}{1/6} \cdot \frac{1/2}{1} = \frac{1}{2}.$$

Metropolis-Hastings Example: Rolling Dice with Coins

- So Metropolis-Hastings modifies random walk probabilities:
  - If you're at the end (1 or 6), stay there half the time.
  - This accounts for the fact that 1 and 6 have only one neighbour.
    - Which means they aren't visited as often by the random walk.
- Could also be viewed as a random surfer in a different graph:

- You can think of Metropolis-Hastings as the modification that "makes the random walk have the right probabilities."
  - For any (reasonable) proposal distribution q.

Special Case: Gibbs Sampling

- An important special case of Metropolis-Hastings is Gibbs sampling.
  - Method to sample from a multi-dimensional distribution.
  - Probably the most common multi-dimensional sampler.

- Gibbs sampling starts with some x and then repeats:
  1. Choose a variable j uniformly at random.
  2. Update x_j by resampling it from its conditional distribution given everything else:

$$x_j^t \sim p(x_j \mid x_{-j}^{t-1}),$$

     where x_{-j} means "all variables except x_j".
     Keep other variables the same.

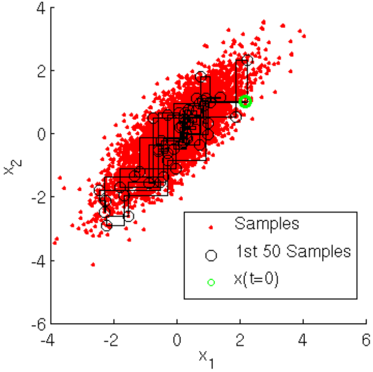
- A common variation is to cycle through the variables in order.

Gibbs Sampling in Action

- Start with some initial value: $x^0 = \begin{bmatrix} 2 & 2 & 3 & 1 \end{bmatrix}$.
- Select random index: $j = 3$.
- Sample variable j: $x^1 = \begin{bmatrix} 2 & 2 & 1 & 1 \end{bmatrix}$.
- Select random index: $j = 1$.
- Sample variable j: $x^2 = \begin{bmatrix} 3 & 2 & 1 & 1 \end{bmatrix}$.
- Select random index: $j = 2$.
- Sample variable j: $x^3 = \begin{bmatrix} 3 & 2 & 1 & 1 \end{bmatrix}$.
- ...
- Use the samples to form a Monte Carlo estimator.

Gibbs Sampling in Action: Multivariate Gaussian

- Gibbs sampling works for general distributions.
  - E.g., sampling from multivariate Gaussian by univariate Gaussian sampling.



https://theclevermachine.wordpress.com/2012/11/05/mcmc-the-gibbs-sampler

- Video: `https://www.youtube.com/watch?v=AEwY6QXWoUg`

Sampling from Conditionals

- For discrete X_j, the conditionals needed for Gibbs sampling have a simple form:

$$p(x_j = c \mid x_{-j}) = \frac{p(X_j = c, x_{-j})}{p(x_{-j})} = \frac{p(X_j = c, x_{-j})}{\sum_{c'} p(x_j = c', x_{-j})} = \frac{\tilde{p}(X_j = c, x_{-j})}{\sum_{c'} \tilde{p}(X_j = c', x_{-j})},$$

  where we can use unnormalized $\tilde{p}$ since Z is the same in numerator/denominator.
  - Last expression is easy to evaluate: just sum over values of x_j.

- For continuous X_j, replace the sum by an integral.
  - May be able to figure out quantile function for inverse transform sampling.
  - May need to use rejection sampling, especially in non-conjugate cases.

Gibbs Sampling as a Markov Chain

- The "Gibbs sampling Markov chain" if p is over 4 binary variables:
  - The states are the possible configurations of the four variables:
    - $[0\ 0\ 0\ 0], [0\ 0\ 0\ 1], [0\ 0\ 1\ 0]$, etc (there are $2^4 = 16$ of them).
  - The initial probability q is set to 1 for the initial state, and 0 for the others:
    - If you start at $[1\ 1\ 0\ 1]$, then $q(x^1 = [1\ 1\ 0\ 1]) = 1$ and $q(x^1 = [0\ 0\ 0\ 0]) = 0$.
  - The transition probabilities q are based on variable we choose and target p:
    - If we are at $[1\ 1\ 0\ 1]$ and choose coordinate randomly we have:

$$q([1\ 1\ 0\ 1] \to [0\ 0\ 1\ 1]) = 0 \quad \text{(Gibbs only updates one variable)}$$

$$q([1\ 1\ 0\ 1] \to [1\ 0\ 0\ 1]) = \underbrace{\frac{1}{d}}_{j \text{ is uniform}} \underbrace{\Pr(X_2 = 0 \mid X_1 = 1, X_3 = 0, X_4 = 1)}_{\text{from target distribution } p}.$$

  - Not homogeneous if cycling, but homogeneous if add "last variable" to state.

Gibbs is Metropolis-Hastings

- For random coordinates, proposal is $q_{x \to \hat{x}} = \frac{1}{d} \sum_{j=1}^{d} \mathbb{1}(\hat{x}_{-j} = x_{-j}) p(\hat{x}_j \mid x_{-j})$
- When $\hat{x}_{-j} = x_{-j}$, acceptance probability is min of 1 and

$$\begin{aligned}
\frac{p(\hat{x})}{p(x)} \cdot \frac{q_{\hat{x} \to x}}{q_{x \to \hat{x}}} &= \frac{p(\hat{x}_j \mid \hat{x}_{-j}) p(\hat{x}_{-j})}{p(x_j \mid x_{-j}) p(x_{-j})} \cdot \frac{\frac{1}{d} p(x_j \mid \hat{x}_{-j})}{\frac{1}{d} p(\hat{x}_j \mid x_{-j})} \\
&= \frac{p(\hat{x}_j \mid x_{-j}) p(x_{-j})}{p(x_j \mid x_{-j}) p(x_{-j})} \cdot \frac{p(x_j \mid x_{-j})}{p(\hat{x}_j \mid x_{-j})} \qquad (x_{-j} = \hat{x}_{-j}) \\
&= 1
\end{aligned}$$

- Detailed balance is satisfied; also need ergodicity for unique stationary dist

Metropolis-Hastings

bonus!

- Common choices for proposal distribution q in Metropolis-Hastings:
  - Metropolis et al. originally used random walks: $x^t = x^{t-1} + \epsilon$ for $\epsilon \sim \mathcal{N}(0, \Sigma)$.
  - Hastings originally used independent proposal: $q(x^t \mid x^{t-1}) = q(x^t)$.
    - Usually not a good choice in high dimensions.
  - Gibbs sampling updates single variable based on conditional.
  - Block Gibbs sampling:
    - If you can sample multiple variables at once Gibbs sampling tends to work better.
  - Collapsed Gibbs sampling (Rao-Blackwellization):
    - MCMC provably works better at sampling marginals of a joint distribution.
    - "Try to integrate over variables you do not care about."

- Unlike rejection sampling, high acceptance rate is not always good:
  - High acceptance rate may mean we're not moving very much.
  - Low acceptance rate definitely means we're not moving very much.
  - Designing good proposals q is an "art".

Advanced Monte Carlo Methods

bonus!

- "Adaptive MCMC": tries to update q as we go. Needs to be done carefully.
- "Particle MCMC": use particle filter to make proposal.

- Auxiliary-variable sampling: introduce variables to sample bigger blocks:
  - E.g., introduce z variables in mixture models.
  - Also used in Bayesian logistic regression (beginning with Albert and Chib).

- Trans-dimensional MCMC:
  - Needed when dimensionality of problem can change on different iterations.
  - Most important application is probably Bayesian feature selection.

- Hamiltonian Monte Carlo:
  - Faster-converging method based on Hamiltonian dynamics (using $\nabla \log p$).

- Population MCMC:
  - Run multiple MCMC methods, each having different "move" size.
  - Large moves do exploration and small moves refine good estimates.

Outline

Higher-Order Markov Models

- Markov models use a density of the form

$$p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2)p(x_4 \mid x_3)\cdots p(x_d \mid x_{d-1}).$$

- They support efficient computation but Markov assumption is strong.

- A more flexible model would be a second-order Markov model,

$$p(x) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2, x_1)p(x_4 \mid x_3, x_2)\cdots p(x_d \mid x_{d-1}, x_{d-2}),$$

  or even higher-order models.

- General case is called directed acyclic graphical (DAG) models:
  - They allow dependence on any subset of previous features.

DAG Models

- As in Markov chains, DAG models use the chain rule to write

$$p(x_1, x_2, \dots, x_d) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1, x_2) \cdots p(x_d \mid x_1, x_2, \dots, x_{d-1}).$$

- We can alternately write this as:

$$p(x_1, x_2, \dots, x_d) = \prod_{j=1}^{d} p(x_j \mid x_{1:j-1}).$$

- In Markov chains, we assumed x_j only depends on previous x_{j-1} given past.
- In DAGs, x_j can depend on any subset of the past $x_1, x_2, \dots, x_{j-1}$.

DAG Models

- We often write joint probability in DAG models as

$$p(x_1, x_2, \ldots, x_d) = \prod_{j=1}^{d} p(x_j \mid x_{\text{pa}(j)}),$$

where $\text{pa}(j)$ are the "parents" of feature j.
  - For Markov chains the only "parent" of j is $(j-1)$.
  - If we have k parents we only need 2^{k+1} parameters (for binary states).

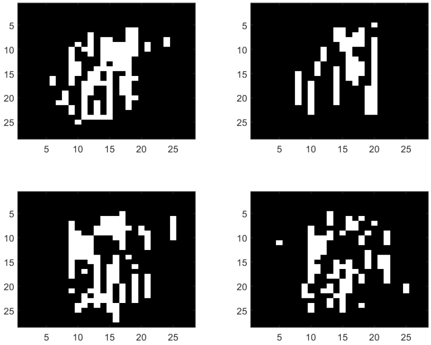
- This corresponds to a set of conditional independence assumptions,

$$p(x_j \mid x_{1:j-1}) = p(x_j \mid x_{\text{pa}(j)}),$$

that we're independent of previous non-parents given the parents.

MNIST Digits with Markov Chains

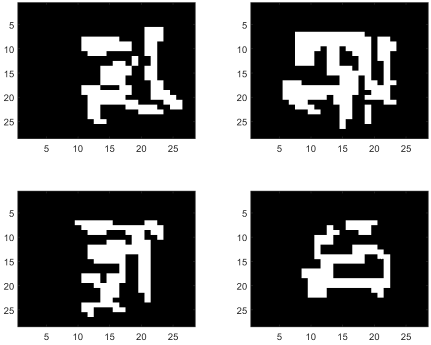
- Recall trying to model digits using an inhomogeneous Markov chain:



Only models dependence on pixel above, not on 2 pixels above nor across columns.

MNIST Digits with DAG Model (Sparse Parents)

- Samples from a DAG model with 8 parents per feature:



Parents of (i,j) are 8 other pixels in the neighbourhood ("up by 2, left by 2"):

$$\{(i{-}2,j{-}2),(i{-}1,j{-}2),(i,j{-}2),(i{-}2,j{-}1),(i{-}1,j{-}1),(i,j{-}1),(i{-}2,j),(i{-}1,j)\}.$$

DAG Models

- "Graphical" name comes from visualizing parents/features as a graph:
  - We have a node for each feature j.
  - We place an edge into j from each of its parents.
- This graph is not just a visualization tool:
  - Can be used to test arbitrary conditional independences ("d-separation").
  - Graph structure tells us whether message passing is efficient ("treewidth").

Graph Structure Examples

With product of independent distributions we have

$$p(x) = \prod_{j=1}^{d} p(x_j),$$

so $\text{pa}(j) = \varnothing$ and the graph is:

Graph Structure Examples

With Markov chain we have

$$p(x) = p(x_1) \prod_{j=2}^{d} p(x_j \mid x_{j-1}),$$

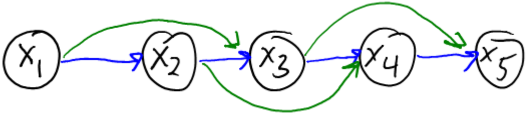
so $\text{pa}(j) = \{j-1\}$ and the graph is:

Graph Structure Examples

With second-order Markov chain we have

$$p(x) = p(x_1)p(x_2 \mid x_1)\prod_{j=3}^{d} p(x_j \mid x_{j-1}, x_{j-2}),$$

so $\text{pa}(j) = \{j-2, j-1\}$ and the graph is:

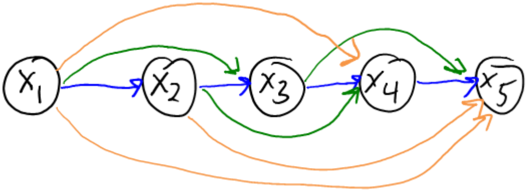


Graph Structure Examples

With a fully general distribution we have

$$p(x) = \prod_{j=1}^{d} p(x_j \mid x_{1:j-1}).$$

so $\mathrm{pa}(j) = \{1, 2, \ldots, j-1\}$ and the graph is:

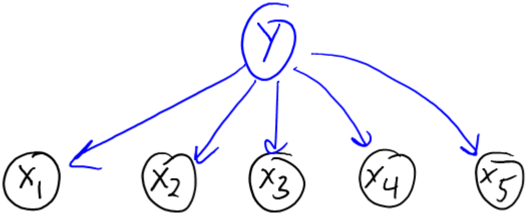


Graph Structure Examples

In naive Bayes (or GDA with diagonal Σ) we add an extra variable y and use

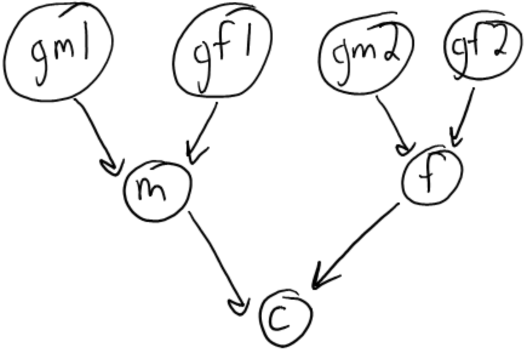
$$p(y, x) = p(y) \prod_{j=1}^{d} p(x_j \mid y),$$

which has $\text{pa}(y) = \emptyset$, $\text{pa}(x_j) = y$:



Graph Structure Examples

We can consider genetic phylogeny (family trees):



The "parents" in the graph are an individual's biological parents.

- Independence assumption: only depend on grandparent's genes through parents.

First DAG Model

bonus!

- DAGs were first used to analyze inheritance in guinea pigs (1920):

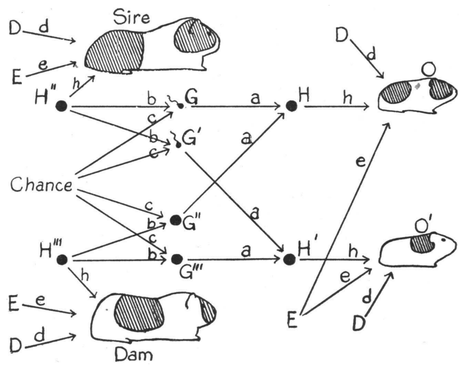


FIG. 5.

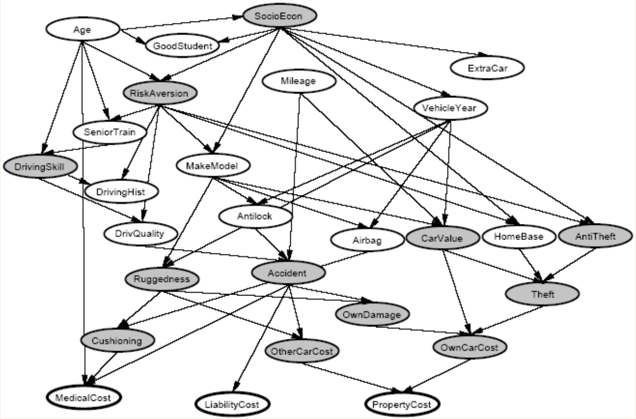
Diagram illustrating the casual relations between litter mates (O, O′) and between each of them and their parents. H, H′, H″, H,‴ represent the genetic constitutions of the four individuals, G, G′, G″, and G‴ that of four germ cells. E represents such environmental factors as are common to litter mates. D represents other factors, largely ontogenetic irregularity. The small letters stand for the various path coefficients.

https://www.pnas.org/doi/pdf/10.1073/pnas.6.6.320

Example: Vehicle Insurance

bonus!

- Want to predict bottom three "cost" variables, given observed and unobserved values:

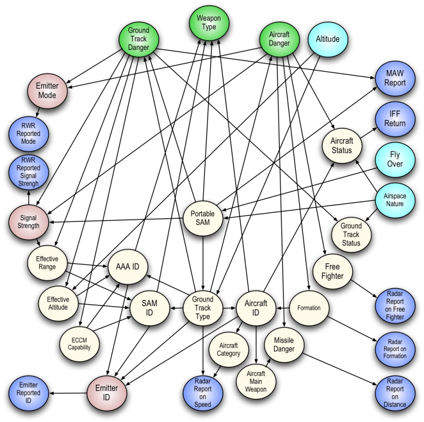


https://www.cs.princeton.edu/courses/archive/fall10/cos402/assignments/bayes

Example: Radar and Aircraft Control

bonus!

- Modeling multiple planes and radar signals:

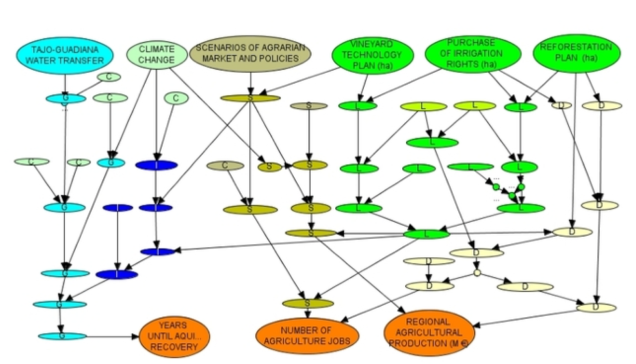


https://pr-owl.org/basics/bn.php

Example: Water Resource Management

bonus!

- Dependencies in environmental monitor and susatainability issues:



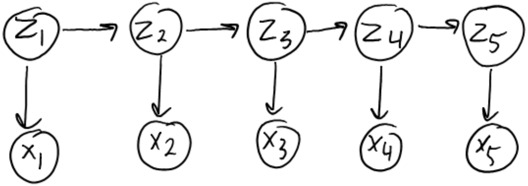
https://www.jstor.org/stable/26268156

Outline

Density Estimators vs. Relationship Visualizers

- In machine learning, DAGs are often used in two different ways:
  1. As a multivariate density estimation method.
     - We'll cover inference and learning in DAGs next time.
  2. As a way to describe the relationships we are modeling.
     - All independence assumptions we have used in 340/440 have DAG representation*.
     - Includes product of Bernoullis and naive Bayes, but also IID and prior vs. hyper-prior.
     - *Except multivariate Gaussians (which can use "undirected" independence).

- For example, later we will talk about hidden Markov models (HMMs):



- The graph and variable names already give you an idea of what this model does:
  - Hidden variables Z_j follow a Markov chain; feature X_j depends on Z_j.

Extra Conditional Independences in Markov Chains

- Markov assumption in Markov chains: $X_j \perp\!\!\!\perp X_1, X_2, \ldots, X_{j-2} \mid X_{j-1}$ for all j

- This implies other independences, like $X_j \perp\!\!\!\perp X_1, X_2, \ldots, X_{j-3} \mid X_{j-2}$.
  - We didn't assume this directly; it follows from assumptions we made.
  - We can use this property to easily compute $p(x_j \mid x_{j-2}, x_{j-3}, \ldots, x_1)$:

$$\begin{aligned}
p(x_j \mid x_{j-2}, x_{j-3}, \ldots x_1) &= p(x_j \mid x_{j-2}) \\
&= \sum_{x_{j-1}} p(x_j, x_{j-1} \mid x_{j-2}) \\
&= \sum_{x_{j-1}} p(x_j \mid x_{j-1}, x_{j-2}) p(x_{j-1} \mid x_{j-2}) \\
&= \sum_{x_{j-1}} \underbrace{p(x_j \mid x_{j-1})}_{\text{transition prob}} \underbrace{p(x_{j-1} \mid x_{j-2})}_{\text{transition prob}}.
\end{aligned}$$

- Mathematically showing extra independence assumptions is tedious (see bonus).
- But all conditional independences implied by a DAG can seen in the graph.

D-Separation: From Graphs to Conditional Independence

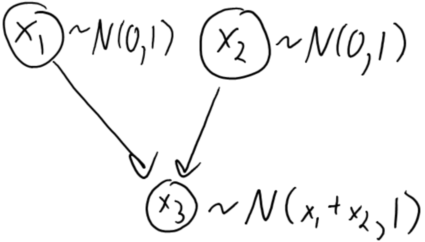
- In DAGs: variables A and B are conditionally independent given C if:
  - "D-separation blocks all undirected paths in the graph from any variable in A to any variable in B."
- In the special case of product of independent models our graph is:

- Here there are no paths to block, which implies the variables are independent.
- Checking paths in a graph tends to be faster than tedious calculations.

D-Separation as Genetic Inheritance

- The rules of d-separation are intuitive in a simple model of gene inheritance:
  - Each node/person has single number, which we'll call a "gene".
  - If you have no parents, your gene is a random number.
  - If you have parents, your gene is a sum of your parents plus noise.

- For example, think of something like this:



- Graph corresponds to the factorization $p(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3 \mid x_1, x_2)$.
  - In this model, does $p(x_1, x_2) = p(x_1)p(x_2)$? (Are X_1 and X_2 independent?)

D-Separation as Genetic Inheritance

- Genes of people are independent if knowing one says nothing about the other.
- Your gene is dependent on your parents:
  - If I know your parent's gene, I know something about yours.
- Your gene is independent of your (unrelated) friends:
  - If you know your friend's gene, it doesn't tell me anything about you.
- Genes of people can be conditionally independent given a third person:
  - Knowing your grandparent's gene tells you something about your gene.
  - But grandparent's gene isn't useful if you know parent's gene.

D-Separation Case 0 (No Paths and Direct Links)

Are genes in person x independent of the genes in person y?

- No path: X and Y are not related (independent).

  We have $X \perp\!\!\!\perp Y$: there are no paths to be blocked.

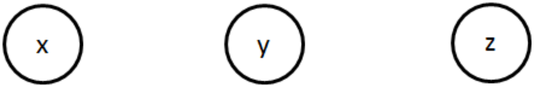
- Direct link: X is the parent of Y.

  We have $X \not\!\perp\!\!\!\perp Y$: knowing X tells you about Y (direct paths aren't blockable).
  - And similarly knowing Y tells you about X.

D-Separation Case 0 (No Paths and Direct Links)

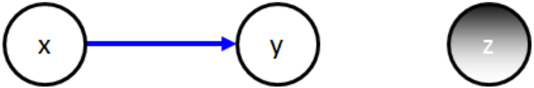
Neither case changes if we have a third independent person Z:

- No path: If X and Y are independent,



  We have $X \perp\!\!\!\perp Y$: adding Z doesn't make a path.

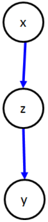
- Direct link: X is the parent of Y,



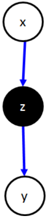
  We have $X \not\perp\!\!\!\perp Y \mid z$: adding Z doesn't block path.
  - We'll use **black or shaded** nodes to denote values we condition on (in this case Z).
    - We sometimes also call the nodes that we condition on the "observations".

D-Separation Case 1: Chain

- Case 1: X is the grandparent of Y.
  - If Z is the parent we have:



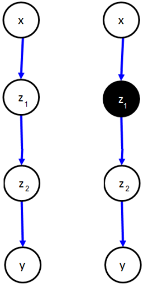
    We have $X \not\perp\!\!\!\perp Y$: knowing X would give information about Y because of Z
  - But if Z is *observed*:



    In this case $X \perp\!\!\!\perp Y \mid Z$: knowing Z "breaks" dependence between X and Y.

D-Separation Case 1: Chain

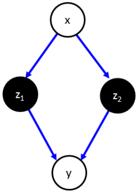
- The same logic holds for great-grandparents:



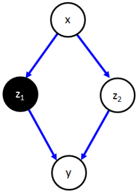
- We have $X \not\perp\!\!\!\perp Y$ (left), but $X \perp\!\!\!\perp Y \mid Z_1$ (right).
  - We also have $X \perp\!\!\!\perp Y \mid Z_2$ and that $X \perp\!\!\!\perp Y \mid Z_1, Z_2$.
- This case lets you test any independence in Markov chains.
  - "Variables are independent conditioned on any variable in betweeen".

D-Separation Case 1: Chain

- Consider weird case where parents Z_1 and Z_2 share parent X:
  - If Z_1 and Z_2 are observed:



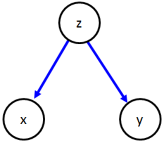
    We have $X \perp\!\!\!\perp Y \mid Z_1, Z_2$: knowing both parents breaks dependency.
  - But if only Z_1 is *observed*:



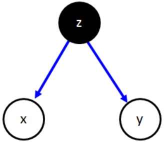
    We have $X \not\perp\!\!\!\perp Y \mid Z_1$: dependence still “flows” through Z_2.

D-Separation Case 2: Common Parent

- Case 2: X and Y are siblings.
  - If Z is a common unobserved parent:



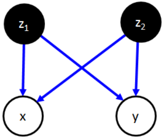
    We have $X \not\perp\!\!\!\perp Y$: knowing X would give information about Y.
  - But if Z is *observed*:



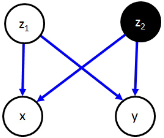
    In this case $X \perp\!\!\!\perp Y \mid Z$: knowing z "breaks" dependence between X and Y.
- This is the type of independence used in naive Bayes.

D-Separation Case 2: Common Parent

- Case 2: X and Y are siblings.
  - If Z_1 and Z_2 are common observed parents:



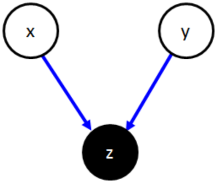
    We have $X \perp\!\!\!\perp Y \mid Z_1, Z_2$: knowing Z_1 and Z_2 breaks dependence between X and Y.
  - But if we only observe Z_2:



    Then we have $X \not\perp\!\!\!\perp Y \mid Z_2$: dependence still "flows" through Z_1.

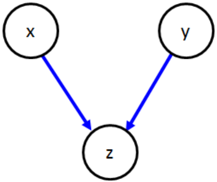
D-Separation Case 3: Common Child

- Case 3: X and Y share a child Z:
  - If we observe Z then we have:



We have $X \not\perp\!\!\!\perp Y \mid Z$: if we know Z, then knowing X gives us information about Y. (Sometimes called "explaining away.")

  - But if Z is not observed:

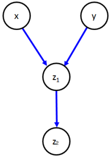


We have $X \perp\!\!\!\perp Y$: if you don't observe Z then X and Y are independent.

- Different from Case 1 and Case 2: **not** observing the child blocks the path.

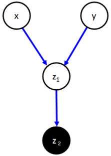
D-Separation Case 3: Common Child

- Case 3: X and Y share a child Z_1:
  - If there exists an unobserved grandchild Z_2:



We have $X \perp\!\!\!\perp Y$: the path is still blocked by not knowing Z_1 or Z_2.

  - But if Z_2 is observed:

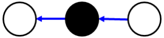


We have $X \not\perp\!\!\!\perp Y \mid Z_2$: grandchild creates dependence even with unobserved child.

- Case 3 needs to consider descendants of child.

D-Separation Summary (MEMORIZE)

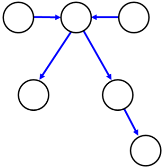
- Checking whether DAG implies A is independent of B given C:
  - Consider each undirected path from any node in any A to any node in B.
    - Ignoring directions and observations.
  - Use directions/observations, check if any of below hold somewhere along each path:
    1. P includes a "chain" with an observed middle node (e.g., Markov chain):

    2. P includes a "fork" with an observed parent node (e.g., naive Bayes):

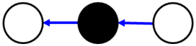
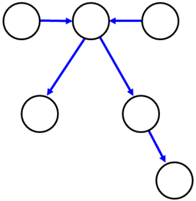
       

    3. P includes a "v-structure" or "collider" (e.g., genetic inheritance):

       where the "child" and all its descendants are unobserved.
  - If all paths are blocked by one of above, DAG implies the conditional independence.

D-Separation Summary (MEMORIZE)

- We say that A and B are d-separated (conditionally independent) given C if *all undirected paths* from A to B are "blocked" because *one* of the following holds *somewhere* on the path:
  1. P includes a "chain" with an observed middle node (e.g., Markov chain):
  
  2. P includes a "fork" with an observed parent node (e.g., naive Bayes):
  
  3. P includes a "v-structure" or "collider" (e.g., genetic inheritance):
  

  where the "child" and all its descendants are unobserved.

Alarm Example

- Case 1:
  - Earthquake $\not\perp\!\!\!\perp$ Call.
  - Earthquake $\perp\!\!\!\perp$ Call $\mid$ Alarm.
- Case 2:
  - Alarm $\not\perp\!\!\!\perp$ Stuff Missing.
  - Alarm $\perp\!\!\!\perp$ Stuff Missing $\mid$ Burglary.

Alarm Example

- Case 3:
  - Earthquake $\perp\!\!\!\perp$ Burglary.
  - Earthquake $\not\perp\!\!\!\perp$ Burglary | Alarm.
    - "Explaining away": knowing one parent can make the other less/more likely.
- Multiple Cases:
  - Call $\not\perp\!\!\!\perp$ Stuff Missing.
  - Earthquake $\perp\!\!\!\perp$ Stuff Missing.
  - Earthquake $\not\perp\!\!\!\perp$ Stuff Missing | Call.

Discussion of D-Separation

- D-separation lets you say if conditional independence is implied by assumptions:

$$(A \text{ and } B \text{ are d-separated given } C) \Rightarrow A \perp\!\!\!\perp B \mid C.$$

- However, there might be extra conditional independences in the distribution:
  - These would depend on specific choices of the DAG parameters.
    - For example, if we set Markov chain parameters so that $p(x_j \mid x_{j-1}) = p(x_j)$.
  - Or some *orderings* of the chain rule may reveal different independences.
  - Lack of d-separation doesn't imply dependence.
    - Just that it's not guaranteed to be independent by the graph structure.

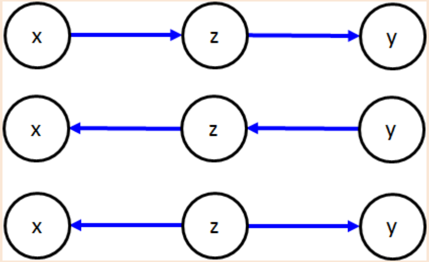
- Instead of restricting to $\{1, 2, \ldots, j-1\}$, can have general parent choices.
  - So x_2 could be a parent of x_1.
- As long the graph is acyclic, there exists a valid ordering (chain rule makes sense).

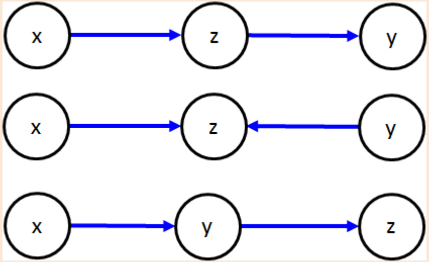
(all DAGs have a "topological order" of variables where parents are before children)

Non-Uniqueness of Graph and Equivalent Graphs

bonus!

- Note that some graphs imply same conditional independences:
  - Equivalent graphs: same v-structures and other (undirected) edges are the same.
  - Examples of 3 *equivalent* graphs (left) and 3 non-equivalent graphs (right):





Beware of the "Causal" DAG

bonus!

- It can be helpful to use the language of causality when reasoning about DAGs.
  - You'll find that they give the correct causal interpretation based on our intuition.

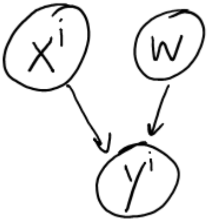
- However, keep in mind that the arrows are not necessarily causal.
  - "A causes B" can have the same graph as "B causes A"!

- There is work on causal DAGs which add semantics to deal with "interventions".
  - But these require assuming that the arrow directions are causal.
    - Fitting a DAG to observational data doesn't imply anything about causality.

Outline

Tilde Notation as a DAG

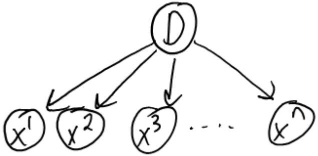
- When we write
$$y^i \sim \mathcal{N}(w^\top x^i, 1),$$
this can be interpreted as a DAG model:



- "The variables on the right of $\sim$ are the parents of the variables on the left".
  - We can see our standard $X \perp w$ assumption in the graph.
    - Common child case: w only depends on X if we know y.

IID Assumption as a DAG

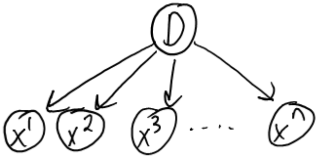
- During week 1, our first independence assumption was the IID assumption:



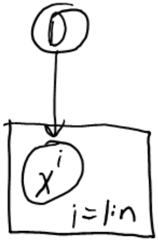
- Training/test examples come independently from data-generating process D.
- But D is unobserved, so knowing about some x^i tells us about the others.
  - This why the IID assumptions lets us learn.

Plate Notation

- Graphical representation of the IID assumption:

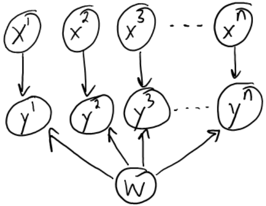


- It's common to represent repeated parts of graphs using plate notation:

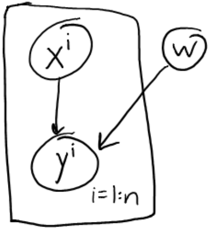


Tilde Notation as a DAG

- If the x^i are IID then we can represent linear regression as



or



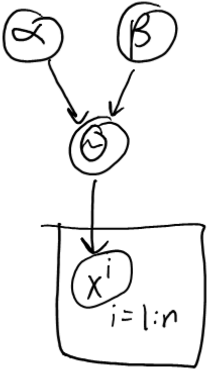
- From d-separation on this graph we have $p(\mathbf{y} \mid \mathbf{X}, w) = \prod_{i=1}^{n} p(y^i \mid x^i, w)$.
  - Our standard assumption that data is independent given parameters.

- We often omit the data-generating distribution D.
  - But if you want to learn it, then you should remember that it's there.

- Note that plate reflects parameter tying: that we use same w for all i.

IID Bernoulli-Beta Model

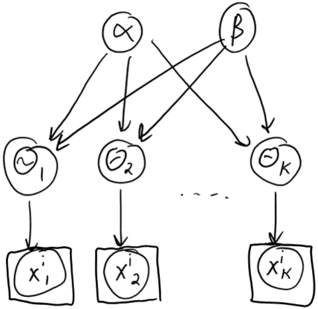
- The Bernoulli-beta model as a DAG (with parameters and hyper-parameters):



- Notice data is independent of hyper-parameters given parameters.
  - This is another of our standard independence assumptions.

Non-IID Bernoulli-Beta Model

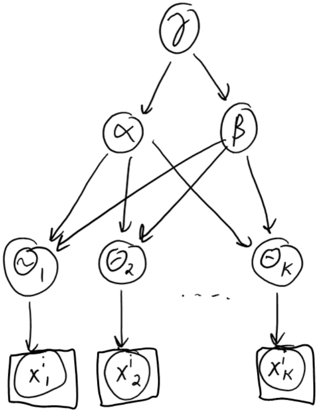
- The non-IID variant we considered with grouped data:



- DAG reflects that we do not tie parameters across all training examples.
- Notice that if you fix α and β then you can't learn across groups:
  - The θ_j are d-separated given α and β.

Non-IID Bernoulli-Beta Model

- Variant of the previous model with a hyper-hyper-parameter:

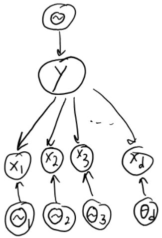


- Which is needed to avoid degeneracy.
- Better version uses nested plates.

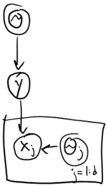
Naive Bayes with DAGs/Plates

- For naive Bayes we have

$$y^i \sim \mathrm{Cat}(\theta), \quad X^i \mid (Y^i = c) \sim \mathrm{Cat}(\theta_c).$$



- Or in plate notation as

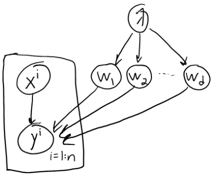


Bayesian Linear Regression as a DAG

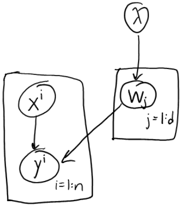
- In Bayesian linear regression we assume

$$y^i \sim \mathcal{N}(w^\mathsf{T} x^i, 1), \quad w_j \sim \mathcal{N}(0, 1/\lambda),$$

which we can interpret as a DAG model:



- Or introducing a second plate over parmaeters:



Summary

- Metropolis-Hastings: MCMC method allowing arbitrary "proposals".
  - Accept/reject samples based on proposal and target probabilities.
- Gibbs sampling: Samples each variable conditioned on all others.
  - Special case of Metropolis-Hastings MCMC method.
- DAG models factorize joint distribution into product of conditionals.
  - Usually we assume conditionals depend on small number of "parents".
  - Most models we've seen can be represented as DAGs.
  - Plate notation helps us do this efficiently.

- D-separation allows us to test conditional independences based on graph.
  - Conditional independence follows if all undirected paths are "blocked".
  - Observed values in chain or parent block paths.
  - Unobserved children (with no observed grandchildren) also blocks paths.

- Next time: learning with DAGs.

Extra Conditional Independences in Markov Chains

bonus!

- Proof that x_j is independent of $\{x_1, x_2, \dots, x_{j-3}\}$ given x_{j-2} in Markov chain:

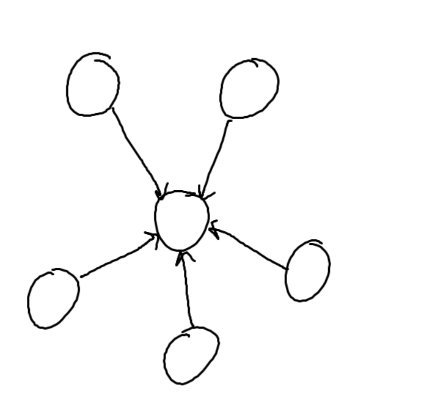
$$
\begin{aligned}
p(x_j \mid x_{j-2}, x_{j-3}, \dots, x_1) &= \frac{p(x_j, x_{j-2}, x_{j-3}, \dots, x_1)}{p(x_{j-2}, x_{j-3}, \dots, x_1)} \quad \text{(def'n cond. prob.)} \\
&= \frac{\sum_{x_{j-1}} p(x_j, x_{j-1}, x_{j-2}, \dots, x_1)}{p(x_{j-2} \mid x_{j-3}, x_{j-4}, \dots, x_1)p(x_{j-3} \mid x_{j-4}, x_{j-5}, \dots, x_1) \cdots p(x_1)} \quad \text{(marg. and chain rule)} \\
&= \frac{\sum_{x_{j-1}} p(x_j \mid x_{j-1}, x_{j-2})p(x_{j-1} \mid x_{j-2}) \dots p(x_2 \mid x_1)p(x_1)}{p(x_{j-2} \mid x_{j-3})p(x_{j-3} \mid x_{j-4}) \cdots p(x_1)} \quad \text{(chain rule and Markov)} \\
&= \frac{p(x_1)p(x_2 \mid x_1) \cdots p(x_{j-2} \mid x_{j-3}) \sum_{x_{j-1}} p(x_j \mid x_{j-1}, x_{j-2})p(x_{j-1} \mid x_{j-2})}{p(x_{j-2} \mid x_{j-3})p(x_{j-3} \mid x_{j-4}) \cdots p(x_1)} \quad \text{(take terms outside} \\
&= \sum_{x_{j-1}} p(x_j \mid x_{j-1}, x_{j-2})p(x_{j-1} \mid x_{j-2}) \quad \text{(cancel out in numerator/denominator)} \\
&= \sum_{x_{j-1}} p(x_j, x_{j-1} \mid x_{j-2}) \quad \text{(product rule)} \\
&= p(x_j \mid x_{j-2}) \quad \text{(marg rule)}.
\end{aligned}
$$

- Similar steps could be used to show $X_j \perp\!\!\!\perp X_{j+2} \mid X_{j+1}$,
and a variety of other conditional independences like $X_1 \perp\!\!\!\perp X_{10} \mid X_5$.

Conditional Independence in Star Graphs

bonus!

- Consider the following star graph:

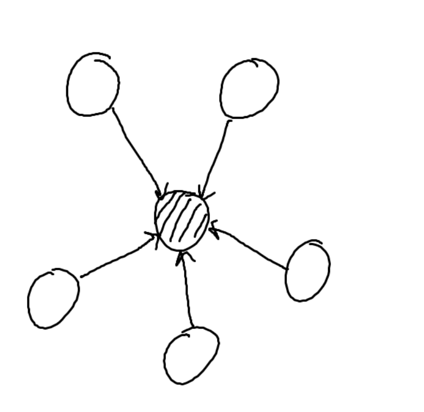


- "5 aliens get together and make a baby alien".
    - Unconditionally, the 5 aliens are independent.

Conditional Independence in Star Graphs

bonus!

- Consider the following star graph:

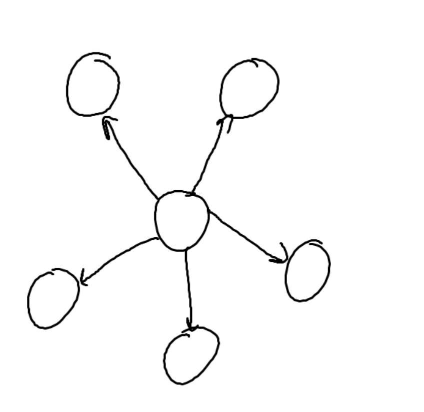


- "5 aliens get together and make a baby alien".
  - Conditioned on the baby, the 5 aliens are dependent.

Conditional Independence in Star Graphs

bonus!

- Consider the following star graph:

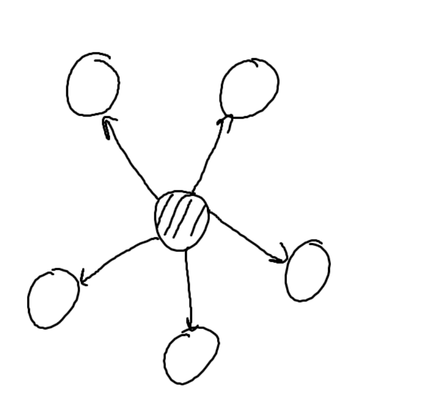


- "An organism produces 5 clones".
  - Unconditionally, the 5 clones are dependent.

Conditional Independence in Star Graphs

bonus!

- Consider the following star graph:



- "An organism produces 5 clones".
  - Conditioned on the original, the 5 clones are independent.