CPSC 540: Machine Learning

More DAGs

Mark Schmidt

University of British Columbia

Winter 2018

Last Time: Directed Acyclic Graphical (DAG) Models

- DAG models use a factorization of the joint distribution,

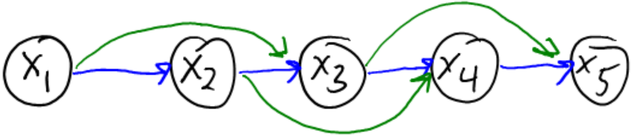
$$p(x_1, x_2, \ldots, x_d) = \prod_{j=1}^{d} p(x_j | x_{\text{pa}(j)}),$$

  where $\text{pa}(j)$ are the parents of node j.

- This assumes a Markov property (generalizing Markov property in chains),

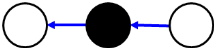
$$p(x_j | x_{1:j-1}) = p(x_j | x_{\text{pa}(j)}),$$

- We visualize the assumptions made by the model as a graph:



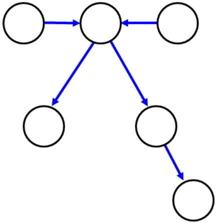
Last Time: D-Separation

- We say that A and B are d-separated (conditionally independent) if *all paths* P from A to B are "blocked" because *at least one* of the following holds:
  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., mixture of Bernoulli):

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



where "child" and all its descendants are unobserved.

Alarm Example

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

Alarm Example

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

Discussion of D-Separation

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

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

- However, there might be extra conditional independences in the distribution:
  - These would depend on specific choices of the $p(x_j \mid x_{\text{pa}(j)})$.
  - Or some *orderings* may reveal different independences.

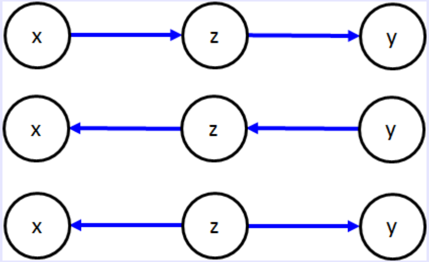
- Instead of restricting to $\{1, 2, \ldots, j-1\}$, consider general parent choices.
  - x_2 could be a parent of x_1.

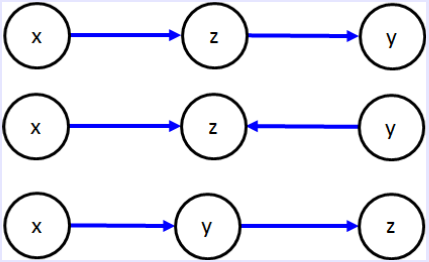
- As long the graph is acyclic, there exists a valid ordering (chain rule makes sense).

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

Non-Uniqueness of Graph and Equivalent Graphs

- 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):



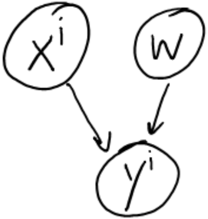


Discussion of D-Separation

- So the graph is not necessarily unique and is not the whole story.
- But, we can already do a lot with d-separation:
  - Implies every independence/conditional-independence we've used in 340/540.
- Here we start blurring distinction between data/parameters/hyper-parameters...

Tilde Notation as a DAG

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

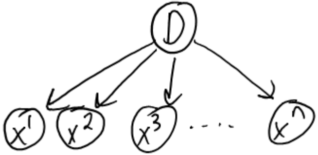


- "The variables on the right of $\sim$ are the parents of the variables on the left".
  - In this case, w only depends on X since we know y.

- Note that we're now including both data and parameters in the graph.
  - This allows us to see and reason about their relationships.

IID Assumption as a DAG

- On Day 2, our first independence assumption was the IID assumption:



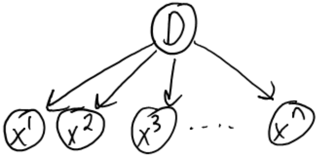
- Training/test examples come independently from data-generating process D.

- If we knew D, we wouldn't need to learn.
- But D is unobserved, so knowing about some x^i tells us about the others.

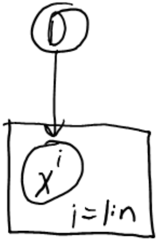
- We'll use this understanding later to relax the IID assumption.
  - Bonus: using this to ask "when does semi-supervised learning make sense?"

Plate Notation

- Graphical representation of the IID assumption:

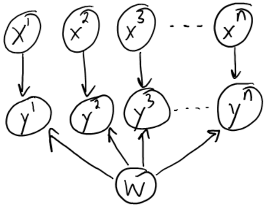


- It's common to represent repeated parts of graphs using plate notation:

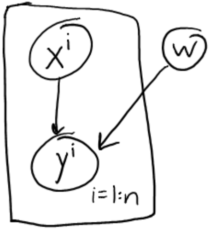


Tilde Notation as a DAG

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



or



- From d-separation on this graph we have $p(y \mid X, w) = \prod_{i=1}^{n} p(y^i \mid x^i, w)$.

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

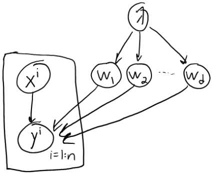
- Note that graph represents parameter tieing: that we use same w for all i.

Tilde Notation as a DAG

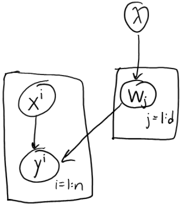
- When we do MAP estimation under the assumptions

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

  we can interpret it as the DAG model:



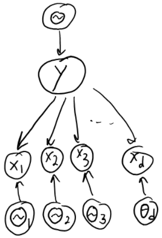
- Or introducing a second plate using:



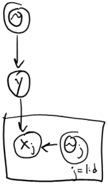
Other Models in DAG/Plate Notation

- For naive Bayes we have

$$y^i \sim \mathsf{Cat}(\theta), \quad x^i \mid y^i = c \sim Cat(\theta_c).$$



- Or in plate notation as



Outline

Parameter Learning in General DAG Models

- The log-likelihood in DAG models is separable in the conditionals,

$$\begin{aligned}\log p(x \mid \Theta) &= \log \prod_{j=1}^{d} p(x_j \mid x_{\mathsf{pa}(j)}, \Theta_j) \\ &= \sum_{j=1}^{d} \log p(x_j \mid x_{\mathsf{pa}(j)}, \Theta_j)\end{aligned}$$

- If each $p(x_j \mid x_{\mathsf{pa}(j)})$ has its own parameters Θ_j, we can fit them independently.
    - We've done this before: naive Bayes, Gaussian discriminant analysis, etc.

- Sometimes you want to have tied parameters $(\Theta_j = \Theta_{j'})$
    - Homogeneous Markov chains, Gaussian discriminant analysis with shared covariance.
    - Still easy, but need to fit $p(x_j \mid x_{\mathsf{pa}(j)}, \Theta_j)$ and $p(x_{j'} \mid x_{\mathsf{pa}(j')}, \Theta_j)$ together.

Tabular Parameterization in DAG Models

- To specify distribution, we need to decide on the form of $p(x_j \mid x_{\mathsf{pa}(j)}, \Theta_j)$.
- For discrete data a default choice is the tabular parameterization:

$$p(x_j \mid x_{\mathsf{pa}(j)}, \Theta_j) = \theta_{x_j, x_{\mathsf{pa}(j)}},$$

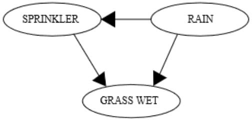
  as we did for Markov chains (but now with multiple parents).
- Intuitive: just need conditional probabilities of children given parents like

$$p(\text{“wet grass”} = 1 \mid \text{“sprinkler”} = 1, \text{“rain”} = 0),$$

  and MLE is just counting.

Tabular Parameterization Example

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

https://en.wikipedia.org/wiki/Bayesian_network

Some quantities can be directly read from the tables:

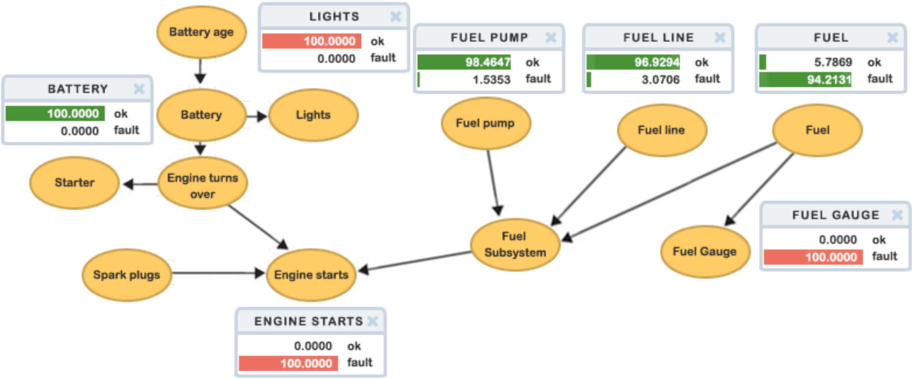
$$p(R = 1) = 0.2.$$

$$p(G = 1 \mid S = 0, R = 1) = 0.8.$$

Can calculate any probabilities using marginalization/product-rule/Bayes-rule (bonus).

Tabular Parameterization Example

Some companies sell software to help companies reason using tabular DAGs:



http://www.hugin.com/index.php/technology

Fitting DAGs using Supervised Learning

- But tabular parameterization requires too many parameters:
  - With binary states and k parents, need 2^{k+1} parameters.
- One solution is letting users specify a "parsimonious" parameterization:
  - Typically have a linear number of parameters.
  - For example, the "noisy-or" model: $p(x_j \mid x_{\mathsf{pa}(j)}) = 1 - \prod_{k \in \mathsf{pa}(j)} q_k$.
- But if we have data, we can use supervised learning.
  - Write fitting $p(x_j \mid x_{\mathsf{pa}(j)})$ as our usual $p(y \mid x)$.
  - We're predicting one column of X given the values of some other columns.

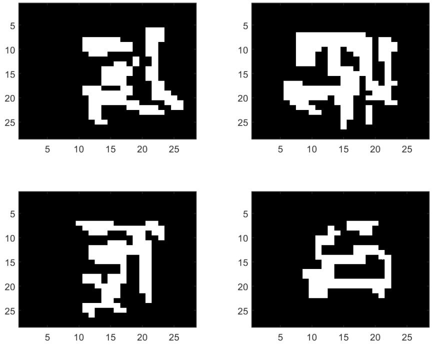
Fitting DAGs using Supervised Learning

- Fitting DAGs using supervised learning:
  - For $j = 1 : d$:
    1. Set $\bar{y}^i = x^i_j$ and $\bar{x}^i = x^i_{\text{pa}(j)}$.
    2. Solve a supervised learning problem using $\{\bar{X}, \bar{y}\}$.
  - Use the d regression/classification models as the density estimator.

- We can use our usual tricks:
  - Linear models, non-linear bases, regularization, kernel trick, random forests, etc.
  - With least squares it's called a Gaussian belief network.
  - With logistic regression it's called a sigmoid belief networks.
  - Don't need Markov assumptions to tractably fit these models.

MNIST Digits with Tabular DAG Model

- Recall our latest MNIST model using a tabular DAG:

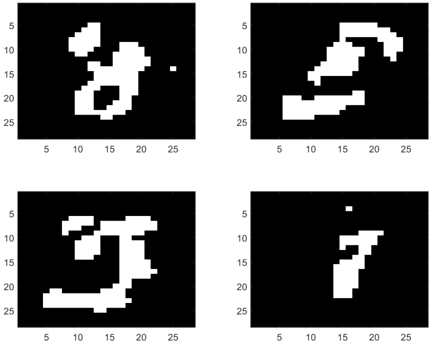


- This model is pretty bad because you only see 8 parents.

MNIST Digits with Sigmoid Belief Network

- Samples from sigmoid belief network:

(DAG with logistic regression for each variable)



where we use all previous pixels as parents (from 0 to 783 parents).

- Models long-range dependencies but has a linear assumption.

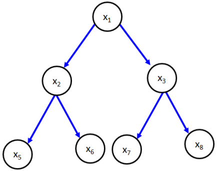
Inference in Forest DAGs

- If we try to generalize the CK equations to DAGs we obtain

$$p(x_j = s) = \sum_{x_{\text{pa}(j)}} p(x_j = s, x_{\text{pa}(j)}) = \sum_{x_{\text{pa}(j)}} \underbrace{p(x_j = s \mid x_{\text{pa}(j)})}_{\text{given}} p(x_{\text{pa}(j)}).$$

which works if each node has at most one parent.

  - Such graphs are called trees (connected), or forests (disconnected).
    - Also called "singly-connected".



  - Forests allow efficient message-passing methods as in Markov chains.
    - In particular, decoding and univariate marginals/conditionals in $O(dk^2)$.
    - Message passing applied to tree-structured graphs is called belief propagation.

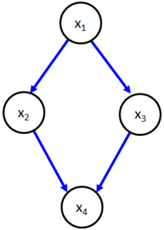
Inference in General DAGs

- If we try to generalize the CK equations to DAGs we obtain

$$p(x_j = s) = \sum_{x_{\mathsf{pa}(j)}} p(x_j = s, x_{\mathsf{pa}(j)}) = \sum_{x_{\mathsf{pa}(j)}} \underbrace{p(x_j = s \mid x_{\mathsf{pa}(j)})}_{\text{given}} p(x_{\mathsf{pa}(j)}).$$

- What goes wrong if nodes have multiple parents?
  - The expression $p(x_{\mathsf{pa}(j)})$ is a joint distribution depending on multiple variables.

- Consider the non-tree graph:



Inference in General DAGs

- We can compute $p(x_4)$ in this non-tree using:

$$\begin{aligned} p(x_4) &= \sum_{x_3}\sum_{x_2}\sum_{x_1} p(x_1, x_2, x_3, x_4) \\ &= \sum_{x_3}\sum_{x_2}\sum_{x_1} p(x_4 \mid x_2, x_3)p(x_3 \mid x_1)p(x_2 \mid x_1)p(x_1) \\ &= \sum_{x_3}\sum_{x_2} p(x_4 \mid x_2, x_3) \underbrace{\sum_{x_1} p(x_3 \mid x_1)p(x_2 \mid x_1)p(x_1)}_{M_{23}(x_2,x_3)} \end{aligned}$$

- Dependencies between $\{x_1, x_2, x_3\}$ mean our message depends on two variables.

$$\begin{aligned} p(x_4) &= \sum_{x_3}\sum_{x_2} p(x_4 \mid x_2, x_3)M_{23}(x_2, x_3) \\ &= \sum_{x_3} M_{34}(x_3, x_4), \end{aligned}$$

Inference in General DAGs

- With 2-variable messages, our cost increases to $O(dk^3)$.
- If we add the edge $x_1 -> x_4$, then the cost is $O(dk^4)$.

  (the same cost as enumerating all possible assignments)
- Unfortunately, cost is not as simple as counting number of parents.
  - Even if each node has 2 parents, we may need huge messages.
  - Decoding is NP-hard and computing marginals is $\#$P-hard in general.
  - We'll see later that maximum message size is "treewidth" of a particular graph.
- On the other hand, ancestral sampling is easy:
  - We can obtain Monte Carlo estimates of solutions to these NP-hard problems.

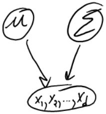
Summary

- Plate Notation lets us compactly draw graphs with repeated patterns.
  - There are fancier versions of plate notation called "probabilistic programming".
- Parameter learning in DAGs:
  - Can fit each $p(x_j \mid x_{\text{pa}(j)})$ independently.
  - Tabular parameterization, or treat as supervised learning.
- Inference in DAGs:
  - Ancestral sampling and Monte Carlo methods work as before.
  - Message-passing message sizes depend on graph structure.
- Next time: trying to discover the graph structure from data.

Other Models in DAG/Plate Notation

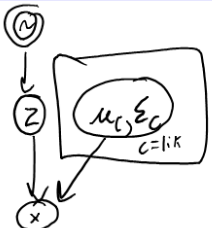
- In a full Gaussian model for a single x we have

$$x^i \sim \mathcal{N}(\mu, \Sigma).$$

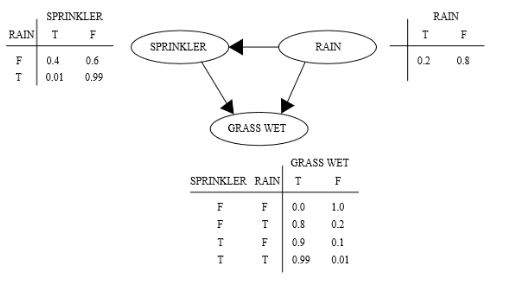


- For mixture of Gaussians we have

$$z^i \sim \mathsf{Cat}(\theta), \quad x^i \mid z^i = c \sim \mathcal{N}(\mu_c, \Sigma_c).$$



Tabular Parameterization Example



	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

`https://en.wikipedia.org/wiki/Bayesian_network`

Can calculate any probabilities using marginalization/product-rule/Bayes-rule, for example:

$$p(G=1 \mid R=1) = p(G=1, S=0 \mid R=1) + p(G=1, S=1 \mid R=1) \qquad \left(p(a \mid c) = \sum_b p(a, b \mid c)\right)$$

$$= p(G=1 \mid S=0, R=1)p(S=0 \mid R=1) + p(G=1 \mid S=1, R=1)p(S=1 \mid R=1)$$

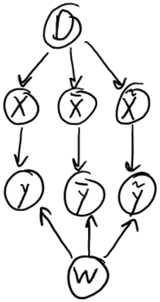
$$= 0.8(0.99) + 0.99(0.01) = 0.81.$$

Does Semi-Supervised Learning Make Sense?

- Should unlabeled examples always help supervised learning?
  - No!
- Consider choosing unlabeled features $\bar{x}^i$ uniformly at random.
  - Unlabeled examples collected in this way will not help.
  - By construction, distribution of $\bar{x}^i$ says nothing about $\bar{y}^i$.
- Example where SSL is not possible:
  - Try to detect food allergy by trying random combinations of food:
    - The actual random process isn't important, as long as it isn't affected by labels.
    - You can sample an infinite number of $\bar{x}^i$ values, but they says nothing about labels.
- Example where SSL is possible:
  - Trying to classify images as "cat" vs. "dog.:
    - Unlabeled data would need to be images of cats or dogs (not random images).
    - Unlabeled data contains information about what images of cats and dogs look like.
    - For example, there could be clusters or manifolds in the unlabeled images.

Does Semi-Supervised Learning Make Sense?

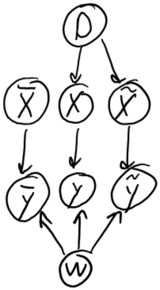
- Let's assume our semi-supervised learning model is represented by this DAG:



- Assume we observe $\{X, y, \bar{X}\}$ and are interested in test labels $\tilde{y}$:
  - There is a dependency between y and $\tilde{y}$ because of path through w.
    - Parameter w is tied between training and test distributions.
  - There is a dependency between X and $\tilde{y}$ because of path through w (given y).
    - But note that there is also a second path through D and $\tilde{X}$.
  - There is a dependency between $\bar{X}$ and $\tilde{y}$ because of path through D and $\tilde{X}$.
    - Unlabeled data helps because it tells us about data-generating distribution D.

Does Semi-Supervised Learning Make Sense?

- Now consider generating $\bar{X}$ independent of D:



- Assume we observe $\{X, y, \bar{X}\}$ and are interested in test labels $\tilde{y}$:
  - Knowing X and y are useful for the same reasons as before.
  - But knowing $\bar{X}$ is not useful:
    - Without knowing $\bar{y}$, $\bar{X}$ is d-separated from $\tilde{y}$ (no dependence).

Beware of the "Causal" DAG

- It can 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" has the same graph as "B causes A".
- There is work on causal DAGs which add semantics to deal with "interventions".
  - But these require extra assumptions: fitting a DAG to observational data doesn't imply anything about causality.