COMS30035, Machine learning: Probabilistic Graphical Models

James Cussens

School of Computer Science
University of Bristol

7th October 2024

The chain rule

- For any joint distribution $P(x_1, \dots, x_n)$ we have:

$$P(x_1, \dots, x_n) = P(x_1)P(x_2|x_1)\dots P(x_n|x_1, \dots x_{n-1}) \qquad (1)$$

- This just follows from the definition of conditional probability.
- Note that we can re-order the the variables at will e.g. $P(x_1, \dots, x_n) = P(x_2)P(x_1|x_2)\dots P(x_n|x_1, \dots x_{n-1})$

Conditional independence

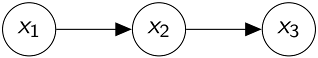
- For any joint distribution over random variables x_1, x_2, x_3 we always have:

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \tag{2}$$

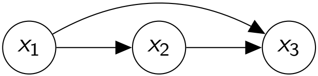
- Now suppose that for some particular probability distribution P we have that: $P(x_3|x_1, x_2) = P(x_3|x_2)$.
- In other words for the distribution P, x_3 is independent of x_1 conditional on x_2.
- Intuition: Once I know the value of x_2 (no matter what that value might be) then knowing x_1 provides no information about x_3.
- Then $P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_2)$
- *Probabilistic graphical models (PGMs)* provide a graphical representation of how a joint distribution factorises when there are conditional independence relations.

Bayesian networks

- The most commonly used PGM is the *Bayesian network*.
- If we have $P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_2)$
- Then this factorisation of the joint distribution is represented by the following directed acyclic graph (DAG):

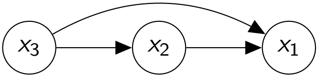


For a distribution with no conditional independence relations a suitable BN representation would be:



$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$

or



$P(x_1, x_2, x_3) = P(x_3)P(x_2|x_3)P(x_1|x_2, x_3)$

Bayesian network terminology

- If there is an arrow from A to B in a Bayesian network we say that A is a *parent* of B and B is a *child* of A.
- The set of parents of a node x_k is denoted (by Bishop) like this: pa_k.
- Note that any directed acyclic graph (DAG) determines pa_k for each node x_k in that DAG (and conversely the collection of parent sets determine the DAG).
- A Bayesian network with parent sets pa_k for random variables $x_1, \ldots, x_K$ represents a joint distribution which factorises as follows:

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k | \mathrm{pa}_k) \tag{3}$$

BN structure and parameters

- For a BN to represent a given joint distribution we need to specify:
  1. the DAG (*the structure of the BN*)
  2. the conditional probability distributions $p(x_k|\mathrm{pa}_k)$ (*the parameters of the BN*)
- A given DAG represents a **set** of joint distributions: each distribution in the set corresponds to a choice of values for the conditional distributions $p(x_k|\mathrm{pa}_k)$.
- We will see that it is possible to 'read off' conditional independence relations that are true for a distribution represented by a BN, just by using the DAG.

BNs represent machine learning models

- We will use BNs to represent machine learning models.
- Later we will see how to use such a representation to 'automatically' do Bayesian machine learning.
- Let's start with a BN to represent Bayesian polynomial regression [Bis06, §8.1.1].
- In a Bayesian approach we have to define a *prior probability distribution* over parameters which (is supposed to) represent our beliefs about their values prior to observing the data.

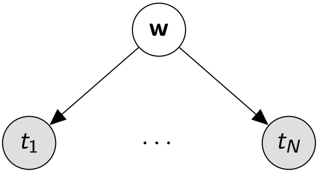
Polynomial regression model

To begin with let's just focus on the joint distribution $p(\mathbf{t}, \mathbf{w})$ where $\mathbf{w}$ is the vector of polynomial coefficients and $\mathbf{t}$ is the observed (output) data.

$p(\mathbf{t}, \mathbf{w})$ can be factorised as follows (since we assume the data is i.i.d.)

$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^{N} p(t_n|\mathbf{w}) \tag{4}$$

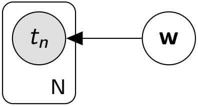
and so has the corresponding BN:



where the dots represent the t_n that have not been explicitly represented in the BN. I have shaded the t_1 and t_n nodes to indicate that the values of these random variables are observed (since they are data).

Plate notation

- Using dots to represent BN nodes we don't wish to explicitly represent is a bit yucky.
- Instead we use *plate notation* to represent BNs with many nodes:



- The plate around t_n represents a set of nodes $t_1, \ldots, t_N$ all of which have $\mathbf{w}$ as their (single) parent.
- Bishop [Bis06, Fig 8.4] labels the plate with N (the number of nodes 'in' the plate). Other authors label plates with an index (here it would be n). We will stick with Bishop's notation to be consistent with the textbook.

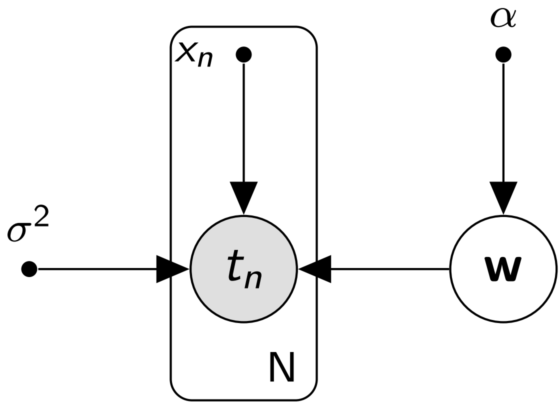
A fuller description

The full Bayesian polynomial regression model contains:

1. The input data $\mathbf{x} = (x_1, \ldots, x_N)^T$
2. The observed outputs $\mathbf{t} = (t_1, \ldots, t_N)^T$
3. The parameter vector $\mathbf{w}$.
4. A hyperparameter α.
5. The noise variance σ^2.

- We don't care how $\mathbf{x}$ is distributed and we would probably just set α to some value.
- So we would typically consider $\mathbf{x}$, α and also σ^2 as parameters of the model rather than random variables.
- But it is also useful represent these quantities in the BN.
- This leads us to more notation for BNs

A complete BN representation for the polynomial regression model



Using BNs to represent ML models

- Machine learning research papers frequently use Bayesian networks to graphically represent machine learning models.
- They represent *the data-generating process*.
- Here's an example from NeurIPS 2019 [BS19].

Differentially private Bayesian linear regression

3.1 Privacy mechanism

Using the Laplace mechanism to release the noisy sufficient statistics $\mathbf{z}$ results in the model shown in Figure 1. This is the same model used in non-private linear regression except for the introduction of $\mathbf{z}$, which requires the exact sufficient statistics $\mathbf{s}$ to have finite sensitivity. A standard assumption in literature [Awan and Slavkovic, 2018, Sheffet, 2017, Wang, 2018, Zhang et al., 2012] is to assume $\mathbf{x}$ and y have known a priori lower and upper bounds, $(a_{\mathbf{x}}, b_{\mathbf{x}})$ and (a_y, b_y), with bound widths $w_{\mathbf{x}} = b_{\mathbf{x}} - a_{\mathbf{x}}$ (assuming, for simplicity, equal bounds for all covariate dimensions) and $w_y = b_y - a_y$, respectively. We can then reason about the worst case influence of an individual on each component of $\mathbf{s} = [X^T X, X^T \mathbf{y}, \mathbf{y}^T \mathbf{y}]$, recalling that $\mathbf{s} = \sum_i t(\mathbf{x}^{(i)}, y^{(i)})$, so that $[\Delta_{(X^T X)_{jk}}, \Delta_{(Xy)_j}, \Delta_{y^2}] = [w_{\mathbf{x}}^2, w_{\mathbf{x}} w_y, w_y^2]$. The number of unique elements[2] in $\mathbf{s}$ is $[d(d+1)/2,\ d,\ 1]$, so $\Delta_{\mathbf{s}} = w_{\mathbf{x}}^2 d(d+1)/2 + w_{\mathbf{x}} w_y d + w_y^2$. The noisy sufficient statistics fit for public release are

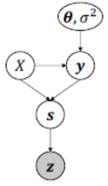


Figure 1: Private regression model.

Another example

- An example from a paper on 'causal representation learning' [LML+23]

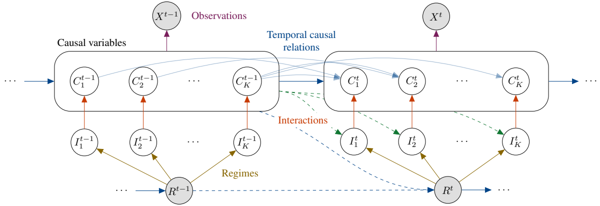


Figure 2: A representation of our assumptions. Observed variables are shown in gray (X^τ and R^τ) and latent variables in white. Optional causal edges are shown as dashed lines. A latent causal variable C_i^t has as parents a subset of the causal factors at the previous time step $C^{t-1} = \{C_1^{t-1}, \ldots, C_K^{t-1}\}$, and its latent binary interaction variable I_i^t. The interaction variables are determined by an observed regime variable R^t and potentially by the variables from the previous time step C^{t-1} (*e.g.*, in a collision). The regime variable can be a dynamical process over time as well, for example, by depending on the previous time step. The observation X^τ is a high-dimensional entangled representation of all causal variables C^τ at time step τ.

Naive Bayes

- In a naive Bayes model for classification [Bis06, p. 380] the observed variables $\mathbf{x} = (x_1, \dots x_D)$ are assumed independent conditional on the class variable $\mathbf{z}$:

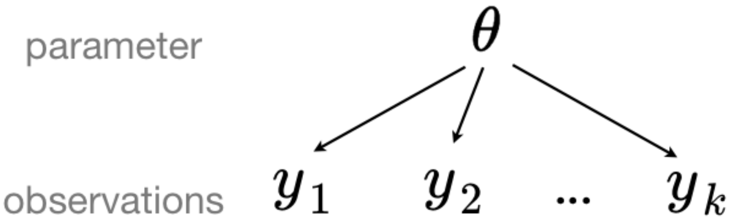
$$P(\mathbf{x}, \mathbf{z}) = P(\mathbf{z})P(\mathbf{x}|\mathbf{z}) = P(\mathbf{z}) \prod_{i=1}^{D} P(x_i|\mathbf{z}) \tag{5}$$

- Let's have a look at a naive Bayes model. [Mur23, p. 163].
- And a latent variable model [Mur23, p. 159].

Hierarchical Linear Regression

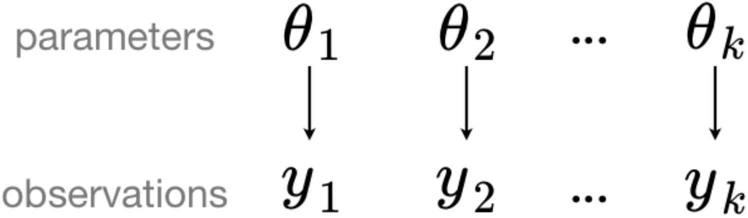
Here's a nice example of using Bayesian networks to represent different approaches to a linear regression problem where there is extra 'structure'.

Standard regression (abbreviated)



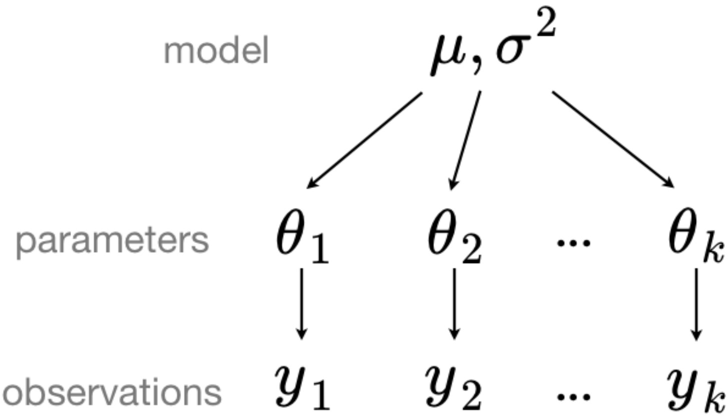
$$P(\theta, y) = P(\theta) \prod_{i=1}^{k} P(y_i|\theta)$$

Separate regressions (abbreviated)



$$P(\theta, y) = \prod_{i=1}^{k} P(y_i|\theta_i)P(\theta_i)$$

Hierarchical regression (abbreviated)



$$P(\theta, y, \mu, \sigma^2) = P(\mu, \sigma^2) \prod_{i=1}^{k} P(y_i|\theta_i) P(\theta_i|\mu, \sigma^2)$$

Conditional independence

- A random variable x is independent of another random variable y *conditional on* a set of random variables S if and only if:

$$P(x, y|S) = P(x|S)P(y|S) \tag{6}$$

Equivalently:

$$P(x|S) = P(x|y, S) \tag{7}$$

- The DAG for a BN encodes conditional independence relations.
- Some of the following slides are modified versions of those made available by David Barber,
- who has written a great (freely available) book on Bayesian machine learning [Bar12]

Conditional independence

- A random variable x is independent of another random variable y *conditional on* a set of random variables S if and only if:

$$P(x, y|S) = P(x|S)P(y|S) \tag{6}$$

Equivalently:

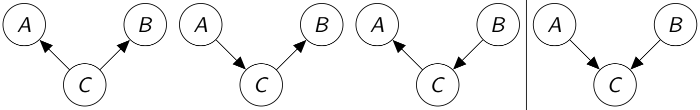
$$P(x|S) = P(x|y, S) \tag{7}$$

- The DAG for a BN encodes conditional independence relations.
- Some of the following slides are modified versions of those made available by David Barber,
- who has written a great (freely available) book on Bayesian machine learning [Bar12]

Independence $\perp$ in Bayesian Networks – Part I

All Bayesian networks with three nodes and two links:

$A \perp B \mid C$ $\qquad$ $A \not\perp B \mid C$

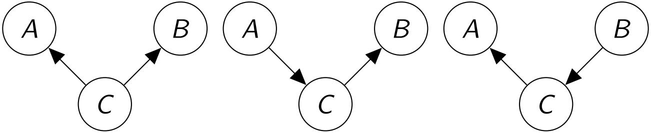


(a) (b) (c) (d)

- In (a), (b) and (c), A and B are conditionally independent given C.
  - (a) $p(A,B|C) = \frac{p(A,B,C)}{p(C)} = \frac{p(A|C)p(B|C)p(C)}{p(C)} = p(A|C)p(B|C)$
  - (b) $p(A,B|C) = \frac{p(A)p(C|A)p(B|C)}{p(C)} = \frac{p(A,C)p(B|C)}{p(C)} = p(A|C)p(B|C)$
  - (c) $p(A,B|C) = \frac{p(A|C)p(C|B)p(B)}{p(C)} = \frac{p(A|C)p(B,C)}{p(C)} = p(A|C)p(B|C)$
- In (d) the variables A, B are conditionally dependent given C, $p(A,B|C) \propto p(A,B,C) = p(C|A,B)p(A)p(B)$.

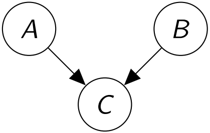
Independence $\perp$ in Bayesian Networks – Exercises

$$A \perp B \mid C$$



(a) (b) (c)

$$A \not\perp B \mid C$$

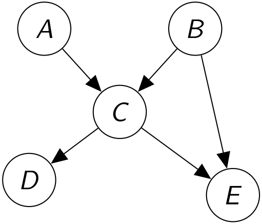


(d)

- Show that in (d), we have $A \perp B$.
- For each of (a), (b) and (c), assume that each variable is binary, and find parameters so that $A \not\perp B$

Paths and colliders

$$p(A, B, C, D, E) = p(A)p(B)p(C|A, B)p(D|C)p(E|B, C)$$

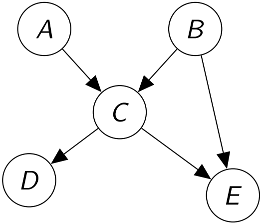


- A node is a *collider* on some path if both arrows point into it on that path.
- C is a collider on the path (A, C, B) but is not a collider on the path (A, C, E) or on any of the following paths: (A, C, E, B), (D, C, B) or (D, C, E).

d-separation

- If all paths from node x to node y are *blocked given nodes* S then x and y are *d-separated* by S.
- A path is blocked by S if at least one of the following is the case:
  1. there is a collider on the path that is not in S and none of its descendants are in S
  2. there is a non-collider on the path that is in S.
- If x and y are *d-separated* by S then $x \perp y | S$ for any probability distribution which factorises according to the DAG.
- Let's do some d-separation exercises.

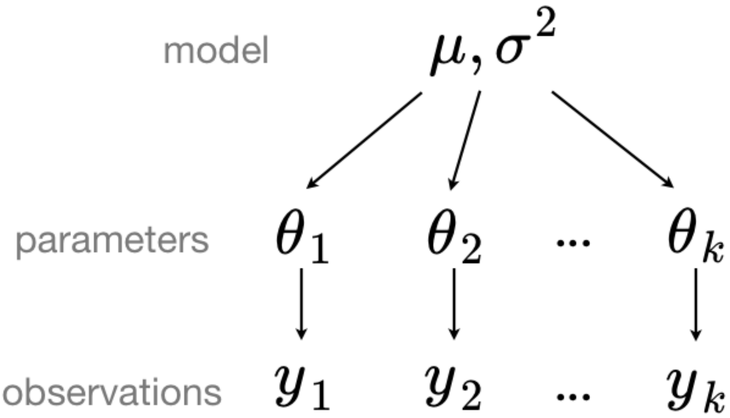
Checking for d-separation



A path is blocked by S if at least one of the following is the case:

1. there is a collider on the path that is not in S and none of its descendants are in S
2. there is a non-collider on the path that is in S.

Hierarchical regression revisited



$$P(\theta, y, \mu, \sigma^2) = P(\mu, \sigma^2) \prod_{i=1}^{k} P(y_i|\theta_i) P(\theta_i|\mu, \sigma^2)$$

Reading

- Bishop §8.1 up to §8.1.1.
- Bishop §8.2 up to §8.2.2.
- Murphy **Book 2** §4.2.1
- Murphy **Book 2** §4.2.4.1

Problems and quizzes

- Do the problems on slide L09: 22/26
- Quizzes:
  - Week 4: Intro to PGMs
  - Week 4: d-separation
  - Week 4: Bayesian Networks for Machine Learning

David Barber.
Bayesian Reasoning and Machine Learning.
Cambridge University Press, 2012.

Christopher M. Bishop.
Pattern Recognition and Machine Learning.
Springer, 2006.

Garrett Bernstein and Daniel R Sheldon.
Differentially private Bayesian linear regression.
In *Advances in Neural Information Processing Systems*, pages 525–535, 2019.

Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Efstratios Gavves.
BISCUIT: Causal representation learning from binary interactions.
In *Proc. UAI23*, 2023.

Kevin P. Murphy.
Probabilistic Machine Learning: Advanced Topics.
MIT Press, 2023.