d-separation

- A "trail" in a DAG is an *undirected path* in the graph.
- Assume X, Y, Z are sets of variables. An "active trail" from X to Y given Z is one where, for every v-structure $x_{i-1} \rightarrow x_i \leftarrow x_{i+1}$ in the trail, x_i or a decendant is in Z, and no other node in the trail is in Z. (**Illustration**).
- We say X and Y are *d-separated* given Z if there is no active trail between any $x \in X$ and $y \in Y$ given Z.
- Theorem: If X and Y are d-separated given Z in a Bayesian network representation of a stochastic model, then $X \coprod Y \mid Z$.
- Theorem: If X and Y are *not* d-separated given Z in a DAG, then there exists a stochastic model where X and Y are not conditionally independent given Z that has the DAG as a Bayesian network.
- See Koller & Friedman: "Probabilistic Graphical Models" for more details.

A way to check d-separation

- Note: The dependency between X and Y given Z is not changed if you remove from a network a child that is not i X, Y, or Z and has no children on its own.
- Doing this repeatedly will lead to a network where all nodes that do not have children are either in X, Y, or Z.
- In this network, you still have to check in the same way whether each trail is active. But there may be fewer trails to check. (FIXED 2021-10-18)
- **Examples**

MSA101/MVE187 2021 Lecture 15
Applying Bayesian statistics

Petter Mostad

Chalmers University

October 18, 2021

Overview

- How to find a suitable stochastic model
- Bayesian model selection
- A connection between Bayesian Learning and Machine Learning
- An example of a paper using Bayesian modelling

The Bayesian paradigm

- Start with data Y_{data} and what you want to predict Y_{pred}, and
  1. describe a joint stochastic model $\pi(Y_{data}, Y_{pred})$,
  2. compute the conditional distribution $\pi(Y_{pred} \mid Y_{data})$.
- The entire course has focused on step 2. What about step 1?
- Below, we describe some advice on finding models.
- We then discuss *Bayesian model selection*, one way to select between alternative possible models.

Advice on statistical modelling

- Always start with data and a clear question.
- Always plot and explore your data, so you understand it as best you can.
- Understand the known science of what is going on as best as you can, to make a realistic model.
- In complicated models:
  1. Start with a Bayesian Network for variables needed to describe a model. Use causality as a guide!
  2. *Then* choose either fixed distributions, or distributions with uncertain parameters, to relate the variables.
- *Elicitation* for constructing informative priors. (Example: Use of `beta.select` in LearnBayes package).
- **Break for examples**

Bayesian model selection

- Assume you are considering n different models connecting your data Y_d with your prediction Y_p.
- Let λ have possible values $1, \ldots, n$ and let $\pi(Y_p, Y_d \mid \lambda = i)$ indicate model i.
- If you specify a prior belief in each model, you can use a combined *weighted model*

$$\pi(Y_p, Y_d) = \sum_{i=1}^{n} \pi(\lambda = i)\pi(Y_p, Y_d \mid \lambda = i)$$

  with weights $w_i = \pi(\lambda = i)$.
- We get

$$\begin{aligned}
\pi(Y_p \mid Y_d) &= \frac{\pi(Y_p, Y_d)}{\pi(Y_d)} = \frac{\sum_{i=1}^{n} \pi(\lambda = i)\pi(Y_d \mid \lambda_i)\pi(Y_p \mid Y_d, \lambda_i)}{\sum_{j=1}^{n} \pi(Y_d \mid \lambda = j)} \\
&= \sum_{i=1}^{n} \left(\frac{\pi(\lambda = i)\pi(Y_d \mid \lambda = i)}{\sum_{j=1}^{n} \pi(\lambda = j)\pi(Y_d \mid \lambda = j)} \right) \pi(Y_p \mid Y_d, \lambda = i)
\end{aligned}$$

Bayesian model selection

- The prediction $\pi(Y_p \mid Y_d)$ using the weighted model uses a weighting of the predictions $\pi(Y_p \mid Y_d, \lambda = i)$ from each individual model, where the weights are updated from $w_i = \pi(\lambda = i)$ to

$$w_i' = \frac{\pi(\lambda = i)\pi(Y_d \mid \lambda = i)}{\sum_{j=1}^{n} \pi(\lambda = j)\pi(Y_d \mid \lambda = j)}.$$

- The value $\pi(Y_d \mid \lambda = i)$ is the probability of observing the data Y_d given model i.
- Except the notation, formulas are exactly the same as when using *mixtures of conjugate priors* (see Lecture 3).
- If one posterior weight w_i' is close to 1, we may approximate by *discarding* all models but model i. *The procedure becomes a model selection procedure*.
- Note: When $n = 2$ we get that $w_2'/w_1' = w_2/w_1 \cdot \pi(Y_d \mid \lambda = 2)/\pi(Y_d \mid \lambda = 1)$.
- To use the formulas in practice, we need to be able to compute $\pi(Y_d \mid \lambda = i)$ for all models i.

Bayesian model selection

- Note: The ideas above cannot be used (directly) to compare a model i with an *improper prior*: Then $\pi(Y_d \mid y = i)$ cannot be computed.
- Note: An improper prior should not be interpreted as a limit of a sequence of proper priors.
- Note: How to determine if models are good apriori? (How to determine prior weights w_i?)
  - May use simulation from the prior model and compare with what is "expected".
- Examples
  - Simulate from the prior of a stochastic model for tree growth.
  - Simulate from the prior of a stochastic model for geological faults.
  - Simulate from the prior of a stochastic model for image noise.

Example of Bayesian model selection

- The data consists of counts c_i, $i = 1, \ldots, n$, with $S = \sum_{i=1}^{n} c_i$.
- Model 1: $(i = 1, \ldots, n)$

$$\begin{aligned} \lambda &\sim \text{Gamma}(1, 1) \\ c_i \mid \lambda &\sim \text{Poisson}(\lambda) \end{aligned}$$

- Model 2: $(i = 1, \ldots, n)$

$$\begin{aligned} p &\sim \text{Uniform}(0, 1) \\ \lambda_0, \lambda_1 &\sim \text{Gamma}(1, 1) \\ \pi(c_i \mid p, \lambda_0, \lambda_1) &= p\,\text{Poisson}(c_i; \lambda_1) + (1 - p)\,\text{Poisson}(c_i; \lambda_0) \end{aligned}$$

- **Break to compute** $\log \pi(c \mid$ **Model 1**$)$**.**
- **Break to compute** $\log \pi(c \mid$ **Model 2**$)$**.**
- As $\pi(c \mid \text{Model 2})/\pi(c \mid \text{Model 1}) = \exp(-2247.885 + 2270.421) = 6128386058$, we see that the second model fits the data much better. Overwhelms any reasonable value for w_2/w_1!

Example: Continued

- Consider Model 3:

$$\pi(c_i) = \hat{p}\,\text{Poisson}(c_i; \hat{\lambda}_1) + (1 - \hat{p})\,\text{Poisson}(c_i; \hat{\lambda}_0)$$

  where $(\hat{p}, \hat{\lambda}_0, \hat{\lambda}_1)$ is the mode of the `logpost` function.
- We get, using R,

$$\log \pi(c \mid \text{Model 3}) = \text{logpost}(\hat{p}, \hat{\lambda}_0, \hat{\lambda}_1) = -2243.493$$

  so

$$\pi(c \mid \text{Model 3})/\pi(c \mid \text{Model 2}) = \exp(-2243.493+2247.885) = 80.8$$

  Should model 3 be preferred to model 2?
- NO: The **prior** probability for Model 3 is quite low, so w_3/w_2 should cancel out the factor 80.8 above.
- Ignoring this leads to **overfitting**, a serious problem in non-Bayesian statistics.

Discussion: Finding a model

- In Bayesian model selection above, we *start with* a model: A weighted mixture of models.
- The modelling question then becomes: How do we get this mixture model in the first place?
- By definition: The initial modelling procedure cannot be based on a model.
- My view:
  - The initial models considered must be based on contextual knowledge and previous experience.
  - In practice, several possible models should be considered, and compared, if possible, with Bayesian model choice.
  - Many other paradigms for model selection exist: They are all somehow related to the basic idea of Bayesian model selection: Comparing likelihoods.

Comparing Bayesian learning and machine learning (ML)

- Bayesian statistics and computation is an important part of ML technology.
- However, the Bayesian paradigm (as used in this course) is generally not used in ML.
- What happens if we apply the Bayesian paradigm to an ML task, and compare approaches?
- For concreteness, we look at the basic problem of classifying digits (0 - 9) from images, using the MNIST data set.
- Using the Bayesian paradigm, Y_{data} is the set of images and their classifications, and Y_{pred} is the classification of a new image. We want to define a joint distribution on these, and then use $\pi(Y_{pred} \mid Y_{data})$.
- Using ML, you may for example choose a neural network ending with a softmax layer used to give probabilities for the 10 classification outcomes. You also choose a particular stochastic algorithm for training of that network, to obtain a single neural network, which you then use for prediction.
- Is it possible to compare or connect the two approaches?

Machine learning as Bayesian inference

- The neural network parameters should be identified with θ, the chosen parameter of the Bayesian model.
- The likelihood defined by the data is the same in both approaches. We also have conditional independence of the observations, and of any new prediction, given the parameter θ.
- In Bayesian inference one would find a posterior for θ (i.e., a posterior on the set of networks) and average over it for predictions.
- In ML one uses (most often) a single network for predictions.
- To make a comparison, we assume the Bayesian approach is to sample a *single* $\hat{\theta}$ from the posterior.
- The Bayesian approach will sample $\hat{\theta}$ from a distribution whose logdensity is

$$\text{Loglikelihood}(\theta) + \text{Prior}(\theta) \tag{1}$$

  where in ML `Loglikelihood` is the negative of the `Loss` and `Prior` is the negative of a regularization term.
- By comparison, ML will use a similar Equation 1 and a stochastic algorithm, but also test- and validation-data, to produce a NN $\hat{\theta}$.

Some important questions (for me)

1. Given an NN, can we establish a clear correspondence

$$\text{Prior}(\theta) \text{ functions} \leftrightarrow \text{Stochastic ML algorithm producing } \hat{\theta}$$

2. Is such a correspondence of practical use when developing new algorithms / models?

- Note: Priors need to be more advanced than currently used regularization terms.
- Note: Simulation in the posterior is not straight-forward in the relevant high dimensions.

Some work on these questions

- My PhD student Anton Johansson and I are investigating connections like those above.
- We define *geometric properties* of θ which can then be used in a `Prior` function.
- We also investigate how geometric properties vary and change when running ML algorithms.
- Anton will present some results fairly soon.