Bayesian statistics and why it is a good fit for biology

Sytems Biology for Scientific Computing: week one

Introduction

General format

1. 25-35mins 'theory' aka slides
2. 25-35mins group computer work

Slide topics:

1. What is Bayesian statistical inference?
2. Why is it useful in general?
3. Why is it useful in systems biology?
4. The big challenge

Computer goals

Set up git/ssh, python, cmdstanpy and cmdstan

What is Bayesian statistical inference?

Probability function

A function that can measure the water in a jug.

i.e. $p : S \to [0, 1]$ where:

- $p(S) = 1$
- For disjoint $A, B \in S$

  $p(A \cup B) = p(A) + p(B)$

Statistical Inference

In: facts about a ~~spoonful~~ sample

Out: propositions about a ~~soup~~ population

e.g.

- spoonful not salty $\rightarrow$ soup not salty
- no carrots in spoon $\rightarrow$ no carrots in soup

Figure 1: A nice soup

Bayesian statistical inference

Statistical inference resulting in a probability.

e.g.

- spoon $\rightarrow p(\text{soup not salty}) = 99.9\%$
- spoon $\rightarrow p(\text{no carrots in soup}) = 95.1\%$

Non-Bayesian inferences:

- spoon $\rightarrow$ Best estimate of [salt] is 0.1mol/l
- $p_{null}(\text{spoon}) = 4.9\% \rightarrow$ no carrots (p=0.049)

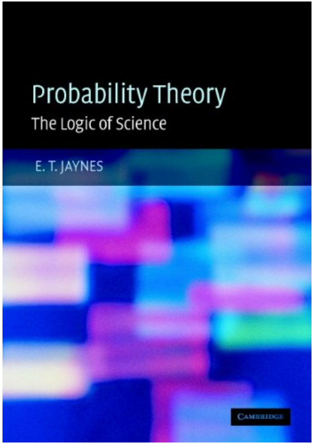
Why is Bayesian statistical inference useful in general?

The philosophical reason

Bayesian inference can be interpreted in terms of information and plausible reasoning.

e.g. "According to the model..."

- "...x is highly plausible."
- "...x is more plausible than y."
- "...the data doesn't contain enough information for firm conclusions about x."



Mathematical reason

Bayesian inference is old!

This means

- it is well understood mathematically.
- conceptual surprises are relatively rare.
- there are many compatible frameworks.

Figure 2: Laplace, who did Bayesian inference in the 1780s

General practical reason

Probabilities decompose nicely:

$$p(\theta, y) = p(\theta)p(y \mid \hat{y}(\theta))$$

- $p(\theta)$: nice form for *background* information, e.g. anything non-experimental
- $\hat{y}(\theta)$: nice form for *structural* information, e.g. physical laws
- $p(y \mid \hat{y}(\theta))$: nice form for *measurement* information, e.g. instrument accuracy

Why is Bayesian inference useful in systems biology?

Regression models: good for describing measurements

Idea: measured value systematically but noisily depends on the true value e.g.

$y \sim N(\hat{y}, \sigma)$

Bayesian inference lends itself to regression models that accurately describe details of the measurement process. e.g.

- heteroskedasticity $y \sim N(\hat{y}, \sigma(\hat{y}))$
- non-negativity $y \sim LN(\ln \hat{y}, \sigma)$ (also compositionality)
- unknown bias $y \sim N(\hat{y} + q, \sigma)$

Multi-level models: good for describing sources of variation

Measurement model:

$$y \sim binomial(K, logit(ability))$$

Gpareto model:

$$ability \sim GPareto(m, k, s)$$

Normal model:

$$ability \sim N(\mu, \tau)$$

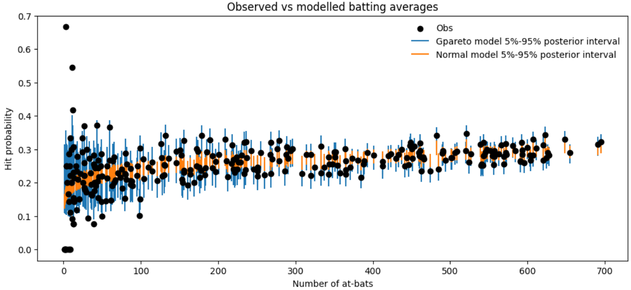


Figure 3: plot from https://github.com/teddygroves/baseball

Generative models: good for representing structural information

Information about hares (u) and lynxes (v):

$$\frac{d}{dt}u = (\alpha - \beta v)u$$
$$\frac{d}{dt}v = (-\gamma + \delta u)v$$

i.e. a deterministic function turning α, β, γ, δ, $u(0)$ and $v(0)$ into $u(t)$ and $v(t)$.

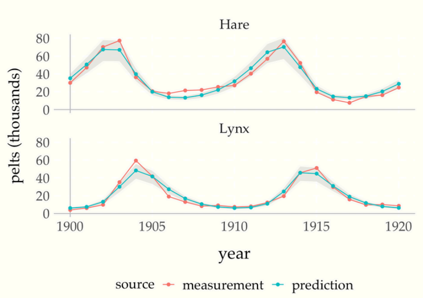


Figure 4: From a Stan case study

The big challenge

The big challenge

$p(\theta \mid y)$ is easy to evaluate but hard to integrate.

This is bad as we typically want something like

$$p([salt] < 0.1, spoon = s)$$

which is equivalent to

$$\int_0^{0.1} p([salt], spoon = s) d[salt]$$

The solution: MCMC

Strategy:

1. Find a series of numbers that
   - quickly finds the high-probabiliy region in parameter space
   - reliably matches its statistical properties
2. Do sample-based approximate integration.

It (often) works!

We can tell when it doesn't work!

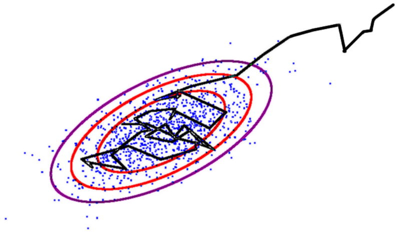


Figure 5: An image I found online

Computer setup

Things to set up

Python

```
python -m venv .venv --prompt=sbsc
```

Git and ssh

```
git clone git@github.com:teddygroves/systems_biology_for_scientific_computing.git
```

Things to set up

Cmdstanpy and cmdstan

```
from cmdstanpy import CmdStanModel

filename = "example_stan_program.stan"

code = "data {} parameters {real t;} model {t ~ std_normal();}"

with open(filename, "w") as f:

    f.write(code)

model = CmdStanModel(stan_file=filename)

mcmc = model.sample()
```

Next time

Next time

Theory

Hamiltonian Monte Carlo: - what? - why? - diagnostics

Computer

Stan, cmdstanpy, arviz: - formats - workflow - write a model