Date: Thursday, January 24, 2019

Next week: New Professor test session.

* After MSDS: - Cool job
  - Pay off student loans
* Want to learn: - Everything
  - Stats & Bayes way of thinking
  - How Bayes compares to "Jeff's stats"
  - Relationship w/ deep learning
* Tips: - Cookies! (Thanks Diane)
  - Examples
  - In class labs
  - You do you, sir

# Chapter 1

Key components:
1) Bayesian state of mind
   - Bayes theorem
2) Bayesian vs. frequentist (aka "Jeff's stats")
3) Bayesian Modeling Inference
   - First models: Poisson, Exponential
   - Examples: a) De-bugging code
     b) Coin flipping
     c) Change in text behavior (aka changepoint problem)

# i) Bayesian state of mind

* Main idea: update prior beliefs with data.

* In all statistics, we aim to model an event or occurrence that is uncertain.

* Uncertainty is what leads to our use of probability models or distributions.

**Example:** De-bugging code

we write a script and want to know the likelihood there is a bug in it.

- uncertain event: bug or not

- prior: I know how often I have a bug the first in all off my previous scripts.

  (JR says 0.99 prob of bug the first time)

A: script has a bug

$\mathbb{P}(A) = 0.99$

- data: test current script on 3 examples

  X: # of times we get an error/bug

$f(X|A,\theta)$

- posterior belief: given our success rate on 3 new examples, what is the likelihood my code has a bug?

$\mathbb{P}(A|X)$

Bayesian inference makes big use of conditioning and Bayes' rule.

$$\mathbb{P}(A|X) = \frac{\mathbb{P}(X|A)\,\mathbb{P}(A)}{\mathbb{P}(X)} \propto \mathbb{P}(X|A)\,\mathbb{P}(A)$$

$$= f(X|A,\theta)\,\mathbb{P}(A)$$

marginalizing constant
that only deals w/ X.
single number, won't change.

"combine"

Prior beliefs "+" data $\longrightarrow$ Posterior beliefs

$\mathbb{P}(A)$ $\quad$ $f(X|A,\theta)$

distributed as

$A \sim \text{Bern}(0.99)$

**$f(X|A,\theta)$:** data generating density for the # of times we have a success (no bugs) given our prior beliefs. (Binomial$(3,\theta)$)

**$\theta$:** The probability of having no bugs.
(Unknown but we can guess using our prior beliefs.)

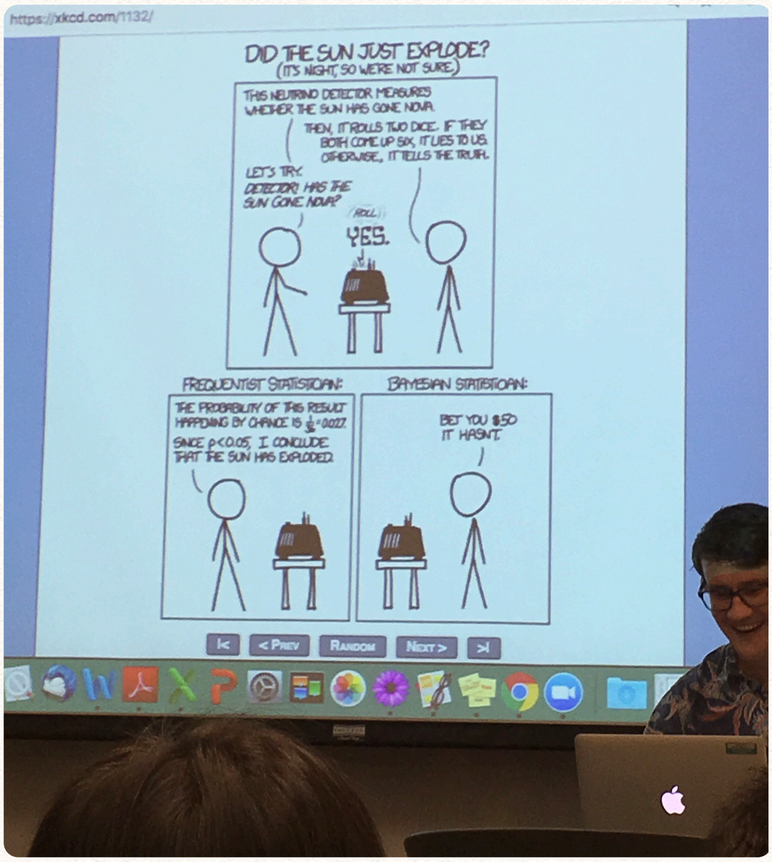
**$\mathbb{P}(A|X)$:** Posterior distribution of having a bug in our script given our new runs.

**Key points:**

1) Bayesian inference is completely done using the posterior distribution. (Prediction, hypothesis tests, etc.)

2) To get that, we specify (hopefully natural) models to our prior beliefs and data generating process.

3) Posterior distributions often cannot be written down in an useable form.
   ↳ This requires us to use simulation from the posterior (which we can do using MCMC).




## 2) Bayes vs. Frequentist ("Jeff's" stats)

Consider our "bug in the code" event A.

**Frequentist perspective:** The likelihood our code has a bug is the "long-run frequency of times we have a bug in our code". That is, we imagine we have run an infinite # of scripts and the prob. of a bug is the proportion of times we had a bug in these scripts.

**Bayesian perspective:** Likelihood our code has a bug is an updated belief in our prior knowledge using new data. In this case, we update our prior prob of 0.99 using density describing 3 successes of current code. Leads to "posterior belief".

**Frequentists:** data — 99/100 had bugs before

0/3 have bugs now

Prob of bug: 99/103

(weighting data the same way)
*distribution

$$\mathbb{P}(A|X) \propto \underbrace{f(X|A,\theta)}_{\text{weight of data}} * \underbrace{\mathbb{P}(A)}_{\text{weight of prior beliefs}}$$

**Notes:** 1) As more data becomes available, our prior beliefs are "washed out". In fact, the probabilities converge to frequentist beliefs.

2) With little data, our prior outweights our insight from data.

* James thinks it is healthy to look at both frequentist and Bayesian methods as tools in your tool belt
  ↳ use them where needed

# 3) Bayesian modeling & inference

**Example:** Change in texting behavior.

**Question:** What is the change point in mean texts received in our data?

**Data:** Counts of texts received each day ($C_i$)

**Distribution:** $C_i \sim \text{Poisson}(\lambda)$

$\lambda$ = mean number of counts

A change in # of texts implies there is a time $\tau$ so that $\lambda = \begin{cases} \lambda_1, & t < \tau \\ \lambda_2, & t \geq \tau \end{cases}$

**Prior:** Specify a distribution for the parameter(s) of the data generating process. Here, this means having priors for $\lambda_1$ and $\lambda_2$.

Note:
If you have
- non-negative values
- continous

↳ exponential distribution is a good way to start

$\lambda_1 \sim \text{Exp}(\alpha)$

$\lambda_2 \sim \text{Exp}(\alpha)$

↑ hyperparameter

Also need prior for when the change occurs (i.e. $\tau$)

$\tau \sim$ Discrete Uniform(1,70) [all days are equaly likely]

## * Magical MCMC

It allows us to simulate from the posterior distributions for

$\mathbb{P}(\tau \mid \underset{\sim}{C}, \alpha, \lambda_1, \lambda_2)$ and

$\mathbb{P}(\lambda_j \mid \underset{\sim}{C}, \alpha, \tau)$

↑ Text counts

which gives us distributions for each.

We can then simplify our findings by summarizing the distributions using the mean, median, or most likely value.

Next week: Chapter 2.
More details will be posted on Slack.