A modern Bayesian Workflow

Peadar Coyle - PyMC3 committer, Blogger and Data Scientist

Signal Media Research Seminar November 2018

@springcoil

www.probabilisticprogrammingprimer.com

Overview

Lots of problems are “small data” or “heteogeneous data” problems. Examples include insurance, agriculture, pharmaceuticals, finance and sports.

Traditional ML models such as XGBoost or Random Forests **DON’T** incorporate domain expertise or work well with small data.

Increasingly models will be deployed in regulated industries - and in a post GDPR world interpretability will matter more. If you work with healthcare data, finance data, insurance you should add Bayesian Statistics to your toolkit.

We’ll discuss how to debug Bayesian models, using modern techniques such as NUTS. This is PyMC3 specific but the techniques apply to Rainier, Stan and BUGS.

If you want some exclusive notebooks

I’ve put together some exclusive notebooks on Probabilistic Programming **only** for this audience

If you want access to them - just sign up to http://eepurl.com/dFZZGb

You can look at my course - www.probabilisticprogrammingprimer.com

Why this matters

To handle large scale problems or ‘big data’ problems in a Bayesian Inference framework - we need to use Hamiltonian samplers.

Hamiltonian samplers work well under ‘certain conditions’. These ‘conditions’ are often swept under the carpet.

Without following these ‘certain conditions’ your inference will be wrong and the decisions you make on the basis on that analysis will be wrong :)

Your job is to inform better decisions

You might think that your job is to understand the ‘truth’ about reality or whatever.

All **science** is about making better decisions.

If your inference is wrong - then your decisions will be wrong.

What are the applications?

Basically anywhere you need to understand uncertainty, handle domain specific knowledge or handle small heterogeneous data.

Marketing is a good use case, A/B testing, survey data, pricing modelling and many use cases in terms of risk modelling.

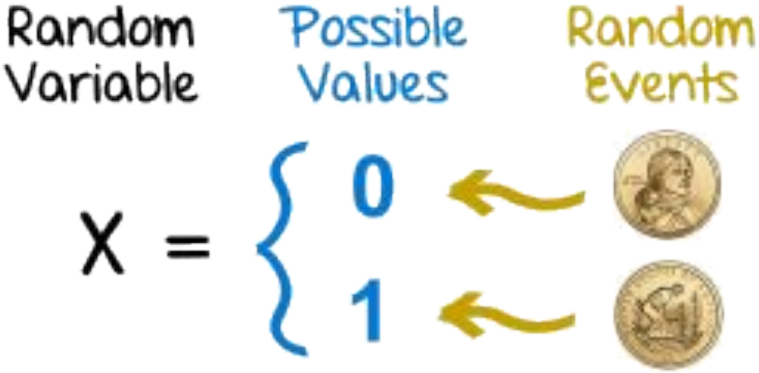
Political forecasting and sports analytics are also good use cases.

Who uses PyMC3 or Stan?

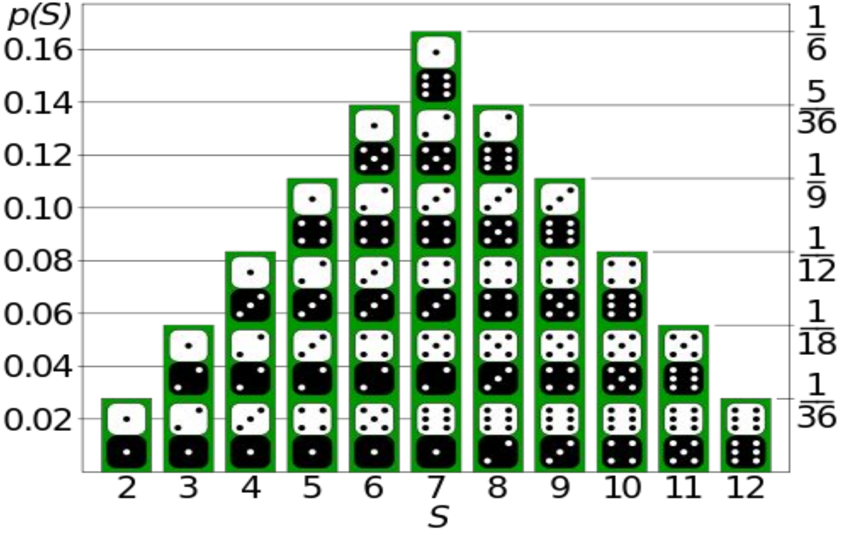
Google, Allianz, Amazon,

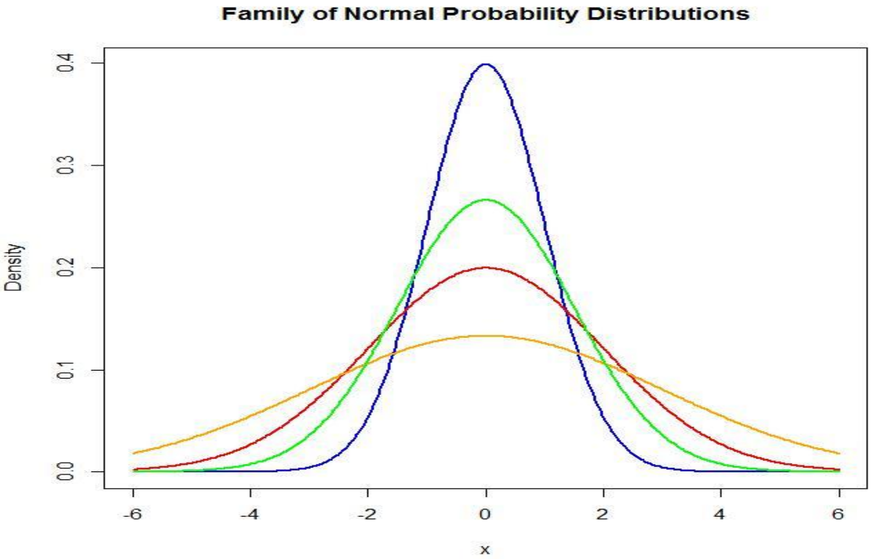
Zopa, Facebook, Channel 4

Random Variables



Probability Distribution





Probability Theory

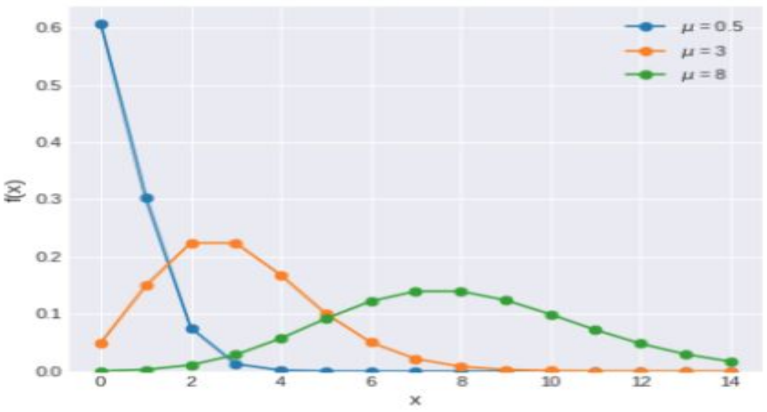
GRAD SCHOOL FLASHBACK!!!!!

- Measure Theory
- Integrals and expectations
- Mappings
- Algebra

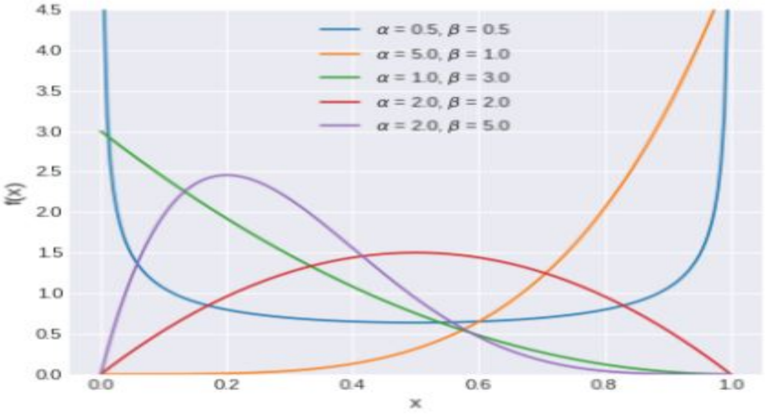
https://betanalpha.github.io/assets/case_studies/probability_theory.html

Probability Distributions

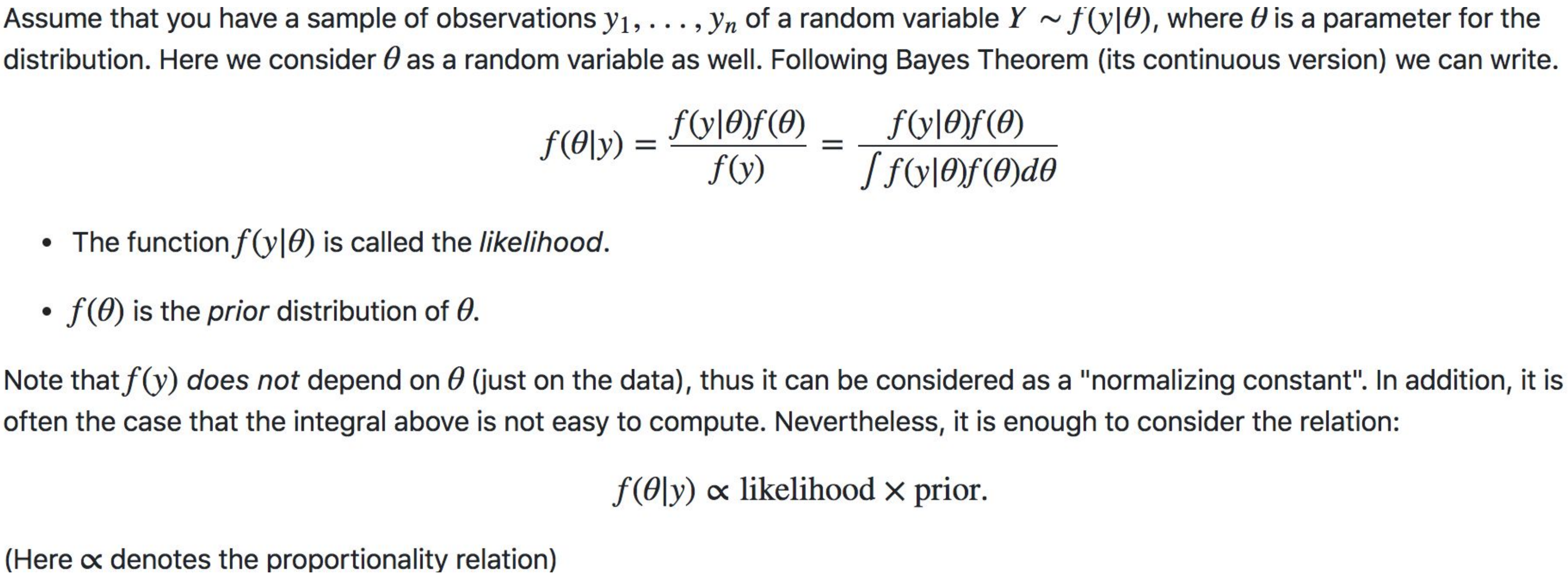
$$f(x \mid \mu) = \frac{e^{-\mu}\mu^{x}}{x!}$$



$$f(x \mid \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$



Bayesian Stats



Assume that you have a sample of observations $y_1, \ldots, y_n$ of a random variable $Y \sim f(y|\theta)$, where θ is a parameter for the distribution. Here we consider θ as a random variable as well. Following Bayes Theorem (its continuous version) we can write.

$$f(\theta|y) = \frac{f(y|\theta)f(\theta)}{f(y)} = \frac{f(y|\theta)f(\theta)}{\int f(y|\theta)f(\theta)d\theta}$$

- The function $f(y|\theta)$ is called the *likelihood*.
- $f(\theta)$ is the *prior* distribution of θ.

Note that $f(y)$ *does not* depend on θ (just on the data), thus it can be considered as a "normalizing constant". In addition, it is often the case that the integral above is not easy to compute. Nevertheless, it is enough to consider the relation:

$$f(\theta|y) \propto \text{likelihood} \times \text{prior}.$$

(Here $\propto$ denotes the proportionality relation)

Why is Bayesian Inference hard?

Because **Integration** is hard!!!!

What are the key ideas?

We know theoretically that MCMC will converge.

We have no guarantee that will happen in practical computational time.

You **don’t** have infinite computation time - even if you work for Google or Amazon.

We need ways of diagnosing biases in our MCMC estimators to tell us if our model is wrongly specified.

Basic workflow

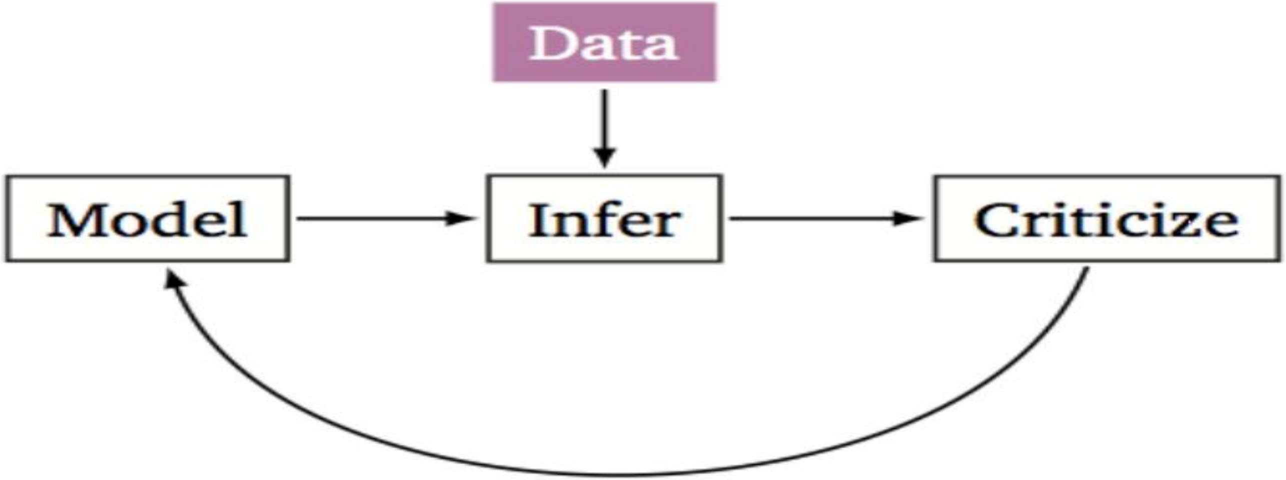
(We'll see a notebook now) https://github.com/springcoil/modernbayesianworkflow

1. Build a model
2. Diagnose the model by looking for 'divergences'
3. Reparameterize the model

General folk theorem of Statistical Computing

- *If the model takes a long time to fit, your model is poorly specified*

Box loop



Summary

Bayesian models aren’t so hard. You can build them now, and they’re important in cases with **small data, domain knowledge and where interpretability matters**.

Don’t just depend on one evaluation metric, if we depended on R-hat we’d not realise our models were wrong.

Personal opinion: **The next big thing in Data Science is Probabilistic Programming.**

Thank you - You may want to check out

www.probabilisticprogrammingprimer.com