



PyQB

Monga

Programming in Python¹

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

Academic year 2024/25, I semester

¹ © 2024 M. Monga. Creative Commons Attribuzione — Condividi allo stesso modo 4.0 Internazionale. <http://creativecommons.org/licenses/by-sa/4.0/deed.it>



PyQB

Monga

Lecture XXIV: Probabilistic programming



PyQB

Monga

How science works

Describing one single “scientific method” is problematic, but a schema many will accept is:

- ① Imagine a hypothesis
- ② Design (mathematical/convenient) **models** consistent with the hypothesis
- ③ Collect experimental data
- ④ Discuss the fitness of data given the models

It is worth noting that the falsification of models is not *automatically* a rejection of hypotheses (and, more obviously, neither a validation).



PyQB

Monga

The role of Bayes Theorem

In this discussion, a useful relationship between data and models is Bayes Theorem.

$$P(M, D) = P(M|D) \cdot P(D) = P(D|M) \cdot P(M)$$

Therefore:

$$P(M|D) = \frac{P(D|M) \cdot P(M)}{P(D)}$$

The plausibility of the model given some observed data, is proportional to the number of ways data can be *produced* by the model and the prior plausibility of the model itself.

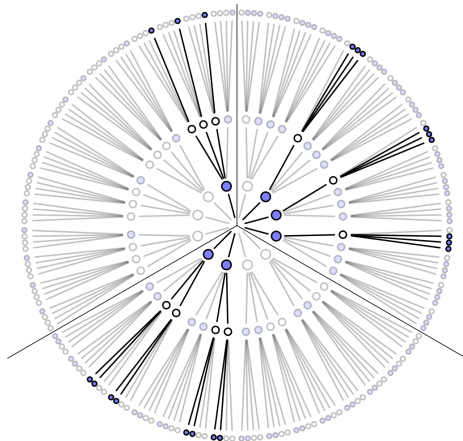
Simple example



PyQB

Monga

- Model: a bag with 4 balls in 2 colors B/W (but we don't know which of BBBB, BBBW, BBWW, BWWW, WWWW)
- Observed: BWB
- Which is the plausibility of BBBB, BBBW, BBWW, BWWW, WWWW?



Picture from: R. McElreath, Statistical Rethinking

Bayes Theorem is **counting**

155

A computational approach



PyQB

Monga

This Bayesian strategy is (conceptually) easy to transform in a computational process.

- 1 Code the models
- 2 Run the models
- 3 Compute the plausibility of the models based on observed data

156

Classical binomial example



PyQB

Monga

- Which is the proportion p of water covering Earth? The models are indexed by the float $0 < p < 1$
- Given p , the probability of observing some W, L in a series of **independent random observations** is:
$$P(W, L|p) = \frac{(W+L)!}{W!L!} p^W \cdot (1-p)^L$$
 (binomial distribution).
- Do we have an initial (prior) idea?
- Make observations, apply Bayes, update prior!

157

A conventional way of expressing the model



PyQB

Monga

$$W \sim \text{Binomial}(W + L, p)$$
$$p \sim \text{Uniform}(0, 1)$$

Probabilistic programming is systematic way of coding this kind of models, combining predefined statistical distributions and Monte Carlo methods for computing the posterior plausibility of parameters.

158

In principle you can do it by hand



PyQB

Monga

```
def dbinom(success: int, size: int, prob: float) -> float:
    fail = size - success
    return math.factorial(size)/(math.factorial(success)*math.factorial(fail))*prob**succ
    ↪ ess*(1-prob)**(fail)
```

Then,

```
W, L = 7, 3 # for example 'WWLLWWLWW'
p_grid = np.linspace(start=0, stop=1, num=20)
prior = np.ones(20)/20

likelihood = dbinom(W, n=W+L, p=p_grid)

unstd_posterior = likelihood * prior

posterior = unstd_posterior / unstd_posterior.sum()
```

Unfeasible with many variables!

159

PyMC



PyQB

Monga

```
import pymc as pm

W, L = 7, 3
earth = pm.Model()
with earth:
    p = pm.Uniform("p", 0, 1) # uniform prior
    w = pm.Binomial("w", n=W+L, p=p, observed=W)
    posterior = pm.sample(2000)

posterior['p']
```

160