

Introduction to Bayesian inference for evolutionists & ecologists

Kent E. Holsinger

Department of Ecology & Evolutionary Biology, U-3043

University of Connecticut

Storrs, CT 06269-3043

© 2020 by Kent E. Holsinger

Why be Bayesian?

How worried should you be if you get a positive test for COVID-19?

- Abbott Alinity SARS-CoV2 Assay

SARS-CoV-2 concentration	Number tested	Number detected
1X to 2X LOD	20	20
20X LOD	20	20
Negative	31	0

- Do you believe it's perfect?
 - $P(\text{true positive}) = 0.976(0.913, 1.000)$
 - $P(\text{true negative}) = 0.970(0.888, 0.999)$

Why be Bayesian?

- Positive test \neq carrying virus
 - Positive and carrying virus (true positive)
 - Positive and not carrying virus (false positive)
- Assume prevalence (proportion of population carrying virus) is 3% and imagine that we test 1000 people

How much should I worry?

$$N_{\text{infected}} = 1000 \times 0.03 = 30$$

$$N_{\text{not infected}} = 1000 \times 0.97 = 970$$

$$N_{\text{infected and positive}} = N_{\text{infected}} \times 0.976$$

$$N_{\text{not infected and positive}} = N_{\text{not infected}} \times 0.030$$

How much should I worry?

$$\begin{aligned}N_{\text{infected}} &= 1000 \times 0.03 = 30 \\N_{\text{not infected}} &= 1000 \times 0.97 = 970 \\N_{\text{infected and positive}} &= N_{\text{infected}} \times 0.976 \\N_{\text{not infected and positive}} &= N_{\text{not infected}} \times 0.030 \\\frac{N_{\text{infected and positive}}}{N_{\text{positive}}} &= \frac{30 \times 0.976}{30 \times 0.976 + 970 \times 0.030} \\&= \frac{29.3}{29.3 + 29.1} \\&= 0.502\end{aligned}$$

Coin flip on whether you have COVID

Bayes' Rule

$$\begin{aligned} P(\text{infected}|\text{positive}) &= \frac{P(\text{positive}|\text{infected})}{P(\text{positive})} P(\text{infected}) \\ &= \frac{0.976}{(0.976)(0.03) + (0.030)(0.97)} (0.03) \\ &= 0.502 \end{aligned}$$

$$P(X|Y) = \frac{P(Y|X)}{P(X)} P(Y)$$

Bayes' Rule for inference

$$P(\theta|X) = \frac{P(X|\theta)}{P(X)}P(\theta)$$

θ = parameter

X = data

Bayes' Rule for inference

$$P(\theta|X) = \frac{P(X|\theta)}{P(X)}P(\theta)$$

θ = parameter

X = data

$P(X|\theta)$ = likelihood

Maximum likelihood estimate: value of θ that maximizes $P(X|\theta)$

Bayes' Rule for inference

$$P(\theta|X) = \frac{P(X|\theta)}{P(X)}P(\theta)$$

θ = parameter

X = data

$P(X|\theta)$ = likelihood

$P(\theta)$ = prior distribution of θ

$P(\theta|X)$ = posterior distribution of θ

Bayesian inference: based on posterior distribution, $P(\theta|X)$

Stan

A probabilistic language for Bayesian analysis

```
data {  
  int<lower=0> k;      // number of positives observed  
  int<lower=0> N;      // number in sample  
}  
  
parameters {  
  real<lower=0, upper=1> p;  // frequency of positives in the sample  
}  
  
model {  
  // likelihood  
  //  
  k ~ binomial(N, p);  
  
  // prior  
  //  
  p ~ uniform(0.0, 1.0);  
}
```

Posterior comparisons

We have a posterior distribution of LMA in *Protea eximia* and *Protea punctata*

1. Set $i = 1$
2. Take one sample at random from the posterior distribution of *Protea eximia*, one from the posterior distribution of *Protea punctata*, take the difference and record it as δ_i .
3. Increment i by 1, and return to step 2 until i is a “big” number.
4. The set of δ_i you have is now a sample from the posterior distribution of the *difference*. If the 95% credible interval is strictly positive, you have evidence that the LMA of *P. eximia* is bigger than the LMA of *P. punctata*.

Scaling variables

`scale()` sets mean to 0, standard deviation to 1

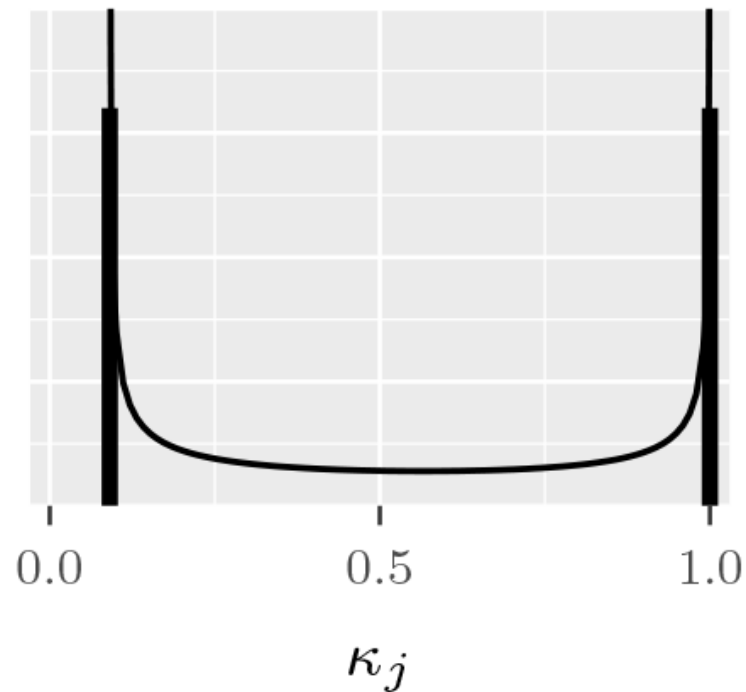
Interpretation of regression coefficient:

$$y_i = \beta_0 + \beta_1 x_i$$

a 1-sd unit change in x leads to a β_1 -sd unit change in y

Horseshoe prior

Shrinkage factor: How much a regression coefficient is “shrunk” towards 0. Lies between 0 and 1.



Pironen & Vehtari, *Electronic Journal of Statistics* 11:5018-5051; 2017.

Leave-one out cross-validation

Cross-validation: Leaving some data out of the model, then “cross-validating” the model by seeing how well you can predict the data you left out.

Leave-one out cross-validation: Leave each observation out of the model, fit the model, predict the observation, and compare prediction to observed *for every observation in the data set*.

`loo()` provides a sneaky way of doing this while only fitting the model once.