Notes on the book Statistical Rethinking

Jason

February 20, 2019

Chapter 1: The Golem of Prague

- Limitations of *deductive falsification*
  1. "Hypotheses are not models. The relations among hypotheses and different kinds of models are complex. Many models correspond to the same hypothesis, and many hypotheses correspond to a single model. This makes strict falsification impossible." (p. 4)
  2. "Measurement matters. Even when we think the data falsify a model, another ob- server will debate our methods and measures. They don't trust the data. Sometimes they are right." (p.4)
- Main topics covered in this book...
  1. Bayesian data analysis
  2. Multilevel models
  3. Model comparison using information criteria

Chapter 2: Small Worlds and Large Worlds

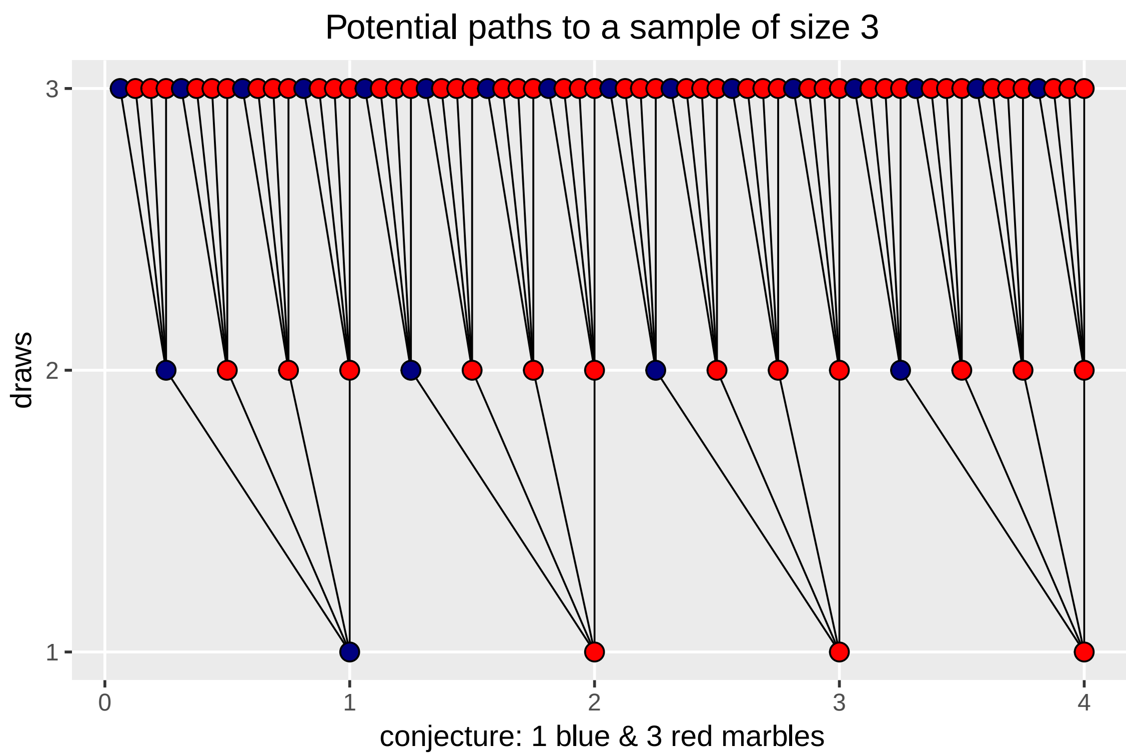
Useful Example.

Suppose there is a bag containing marbles. We *know* there are four marbles in the bag and that each marble may be either red or blue.

- *Prior belief:* The Bayesian approach always begins with a prior belief about the conjecture – i.e., how many red marbles are in the bag. To incorporate a prior belief we must assign a plausiblity to each possibility. Imagine that we met the owner of the bag and they told us that there is a company that makes these bags and the do so in a way that ensure that there is a 15% chance that a bag has only 1 blue marble, a 50% chance that a bag as 2 blue marbles, a 15% chance that a bag has 3 blue marbles, a 10% chance that a bag has all blue marbles, and a 10% chance that the bag does not contain any blue marbles. We could use this as our prior belief about the contents of our bag.

- *Data:* Suppose we repeat the following steps 3 times: shake the bag (randomly distributing the marbles), blindly draw out a marble, note the color, and put the marble back in the bag. Now, suppose this exercise produced the following sequence of marbles ● ● ● in that exact order (*so sequence matters*).

- *Likelihood:* We can think of the likelihood as a way to count all of the possible ways of producing this sample **given** a conjecture of what the bag looks like.Consider one possible conjecture: 🔵🔴🔴🔴. How likely is the observed sample 🔵🔴🔵 given the conjecture? To answer this question, we count the possible ways our conjecture can generate the sample. Then we will compare the total number of ways across different conjectures as a way to evaluate which conjecture is the most likely candidate for generating 🔵🔴🔵. The following plot shows all of the possible samples that (the conjecture) 🔵🔴🔴🔴 can produce:



From this figure we see that there are 3 possible ways to generate the observed data. The following table lists all of the ways that the 5 possible conjectures could generate the sample 🔵🔴🔵:

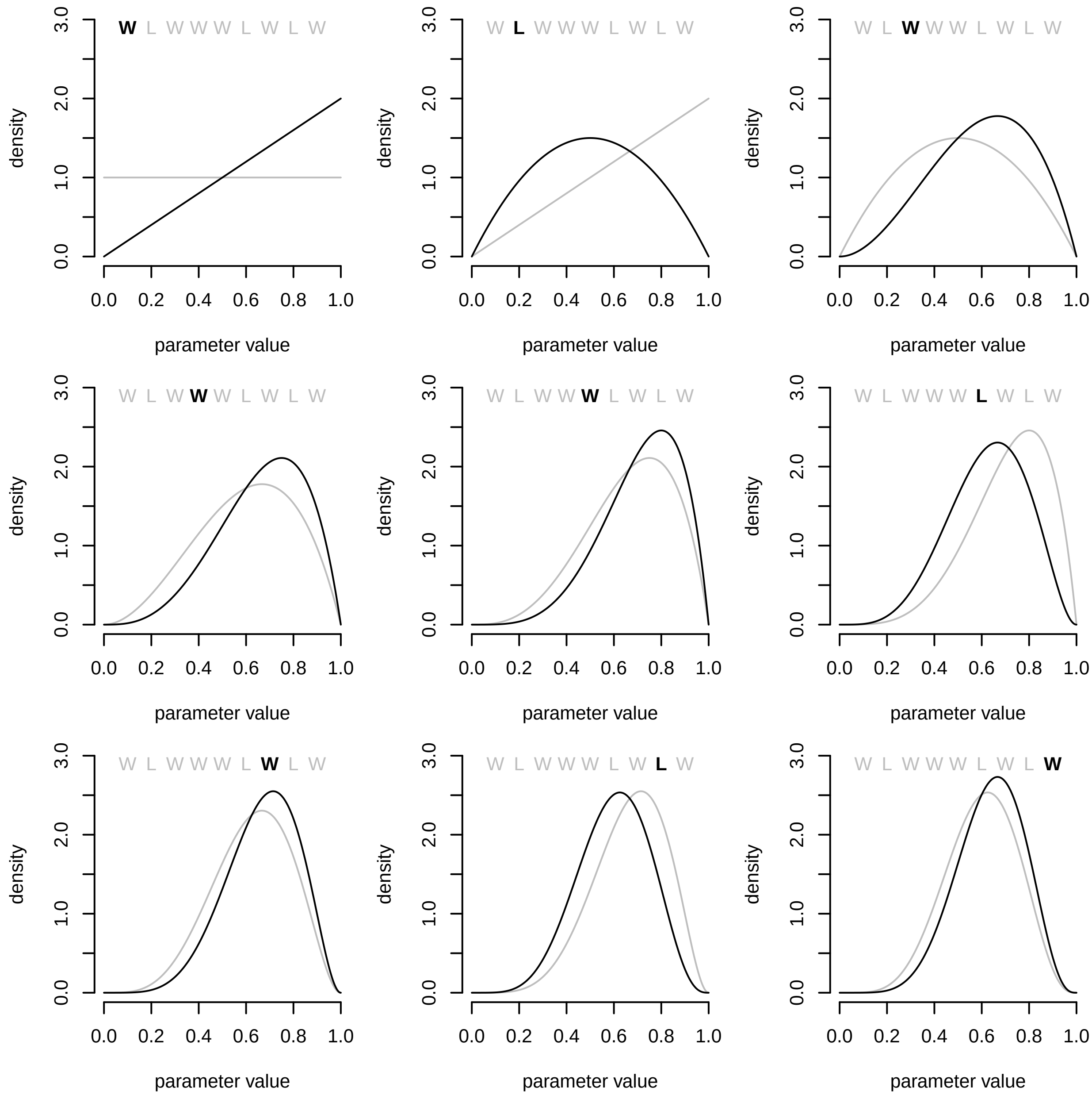
Conjecture	**# of ways to produce 🔵🔴🔵**	**% of Total**
🔵🔴🔴🔴	$1 \times 3 \times 1 = 3$	$\frac{3}{20} = .15$
🔵🔵🔴🔴	$2 \times 2 \times 2 = 8$	$\frac{8}{20} = .40$
🔵🔵🔵🔴	$3 \times 1 \times 3 = 9$	$\frac{9}{20} = .45$
🔵🔵🔵🔵	$4 \times 0 \times 4 = 0$	$\frac{0}{20} = .00$
🔴🔴🔴🔴	$0 \times 4 \times 0 = 0$	$\frac{0}{20} = .00$

- *Bayesian updating & posterior distribution:* We update our prior information using the data, our likelihood, and multiplication. The result is a probability distribution for our model parameters, which we can use to make inferences.

Conjecture	Prior	Likelihood of 🔵🔴🔵	$\propto$ Posterior	Rescaled Posterior
🔵🔴🔴🔴	.15	$\frac{3}{20} = .15$	$.15 \times .15 = .0225$	$\frac{.0225}{.275} = .082$
🔵🔵🔴🔴	.50	$\frac{8}{20} = .40$	$.50 \times .40 = .2000$	$\frac{.2}{.275} = .723$
🔵🔵🔵🔴	.15	$\frac{9}{20} = .45$	$.15 \times .35 = .0525$	$\frac{.0525}{.275} = .191$
🔵🔵🔵🔵	.10	$\frac{0}{20} = .00$	$.10 \times .00 = .0000$	$\frac{0}{.275} = .000$
🔴🔴🔴🔴	.10	$\frac{0}{20} = .00$	$.10 \times .00 = .0000$	$\frac{0}{.275} = .000$

Another example of Bayesian updating.

Consider a small globe that we toss up in the air (so that it spins) and when we catch it we note if our right index finger is covering a portion of water (W) or land (L). Supposes we toss the globe 9 times and obtain the following data: W, L, W, W, W, L, W, L, W. Consider the proportion of the globe that is covered with water, the parameter that we want to study. If our prior belief is that all of the proportions from 0 to 1 are equally likely, then the following figure shows what happens as we toss the ball, observe on outcome, and update our prior beliefs.



Evaluating posterior

In the previous figure, we used an analytic solution to show the result of updating our beliefs. For many problems, such an analytic solution does not exist and we need to use other methods to perform inference on the parameters of interest. Three such methods are covered in this text:

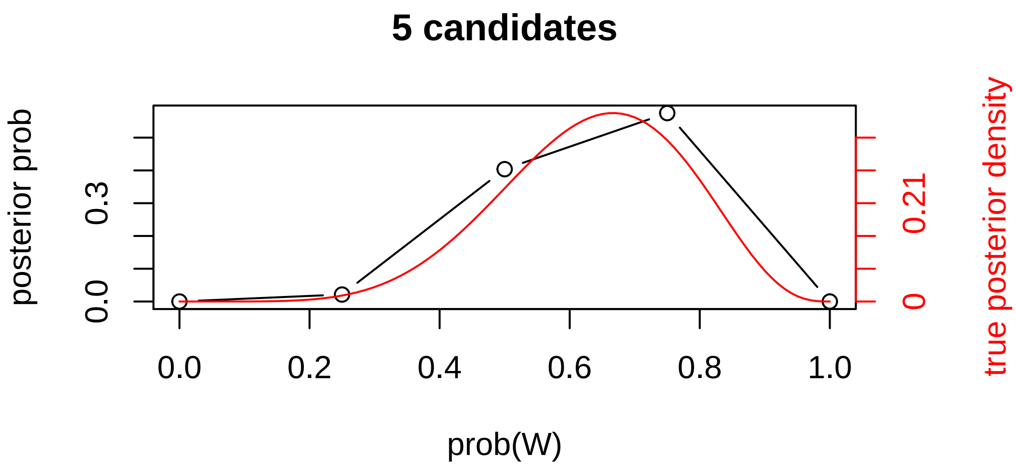
1. Grid approximation
2. Quadratic approximation
3. Monte carlo markov chain (MCMC)

Here we illustrate the grid and quadratic approximations using the globe example. With the grid approach, we begin by creating an R object that contains possible values, or "candidate" values, for the parameter being estimated (i.e., the probability that our experiment results in a W – water). Next, we create a vector containing the prior probabilities for the candidate values and multiply this with the values of our likelihood function, evaluated at our candidate values. The resulting product is an unstandardized posterior, which can easily be converted into a proper probability measure by dividing each value by the sum. Here are two examles with different levels of precision.

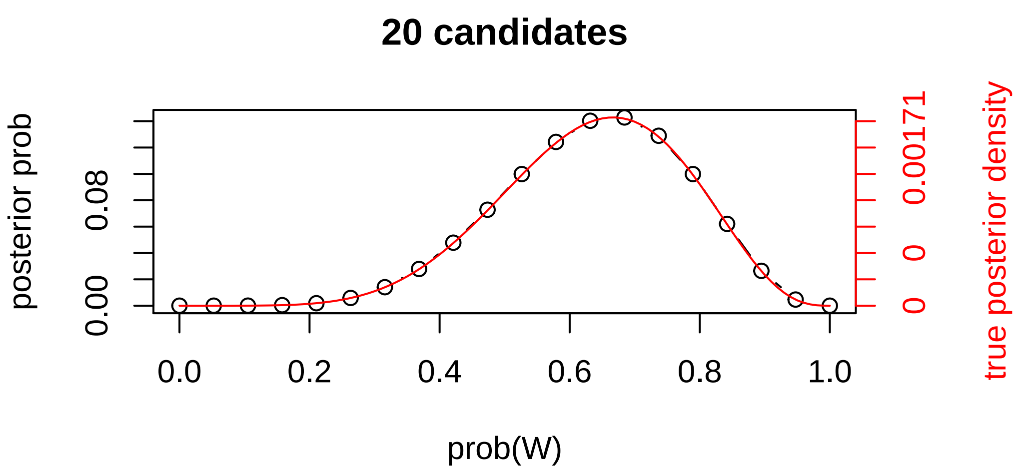
```
cand_1 <- seq(0, 1, length = 5)
prior_1 <- rep(1, 5)
like_1 <- dbinom(6, 9, prob = cand_1)
unstd_post_1 <- like_1 * prior_1
post_1 <- unstd_post_1 / sum(unstd_post_1)

cand_2 <- seq(0, 1, length = 20)
prior_2 <- rep(1, 20)
like_2 <- dbinom(6, 9, prob = cand_2)
unstd_post_2 <- like_2 * prior_2
post_2 <- unstd_post_2 / sum(unstd_post_2)

## check out results
post_true <- dbeta(seq(0, 1, .01), shape1 = 7, shape2 = 4)
par(oma = c(4,4,4,4))
plot(cand_1, post_1, type = 'b', main = '5 candidates',
     xlab = 'prob(W)', ylab ='posterior prob')
lines(seq(0, 1, .01), post_true * (max(post_1)/max(post_true)), col = 2)
axis(side = 4, col = 2, col.axis = 2, at = seq(0, .5, .1),
     lab = round(dbeta(seq(0, .5, .1), 7, 4), 2))
mtext("true posterior density", side = 4, col = 2, line = 1, outer = TRUE)
```




```r
par(oma = c(4,4,4,4))
plot(cand_2, post_2, type = 'b', main = '20 candidates',
     xlab = 'prob(W)', ylab ='posterior prob')
lines(seq(0, 1, .01), post_true * (max(post_2)/max(post_true)), col = 2)
axis(side = 4, col = 2, col.axis = 2, at = seq(0, .14, .02),
     lab = round(dbeta(seq(0, .14, .02), 7, 4), 5))
mtext("true posterior density", side = 4, col = 2, line = 1, outer = TRUE)
```




```r
## predict some new data to show another comparisons...
```

Chapter 3

Chapter 4

Quadratix approximation for the basic linear regression...

```r
library(rethinking)
```

```
## Loading required package: rstan
```

```
## Loading required package: StanHeaders
```

```
## Warning: package 'StanHeaders' was built under R version 3.5.2
```

```
## rstan (Version 2.18.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
##
```

```
## Attaching package: 'rstan'

## The following object is masked from 'package:tidyr':
##
##     extract

## Loading required package: parallel

## rethinking (Version 1.59)

##
## Attaching package: 'rethinking'

## The following object is masked from 'package:purrr':
##
##     map
```

```r
data(Howell1)
str(Howell1)
```

```
## 'data.frame':    544 obs. of  4 variables:
##  $ height: num  152 140 137 157 145 ...
##  $ weight: num  47.8 36.5 31.9 53 41.3 ...
##  $ age   : num  63 63 65 41 51 35 32 27 19 54 ...
##  $ male  : int  1 0 0 1 0 1 0 1 0 1 ...
```

```r
posterior <- function(x) { # x = c(140, 4)
  mu_prior <- dnorm(x[1], mean = 178, sd = 20)
  sd_prior <- dunif(x[2], 0, 50)
  likelihood <- dnorm(x = Howell1$height[Howell1$age > 18],
                      mean = x[1], sd = x[2], log = TRUE)
  logPost <- log(mu_prior) + sum(likelihood) + log(sd_prior)
  -logPost
}
posterior(x = c(140, 40))
```

```
## [1] 1633.64
```

```r
result <- optim(par = c(140, 40), fn = posterior, hessian = TRUE)
```

```
## Warning in dnorm(x = Howell1$height[Howell1$age > 18], mean = x[1], sd =
## x[2], : NaNs produced
```

```r
result$par
```

```
## [1] 154.653477   7.763159
```

```r
sqrt(diag(solve(result$hessian)))
```

```
## [1] 0.4172597 0.2951588
```

How does this do for a skewed posterior? An example. . .