

Stat 343: Optimization in Stan

Mites

This example is adapted from Example A in Chapter 8.7 of Rice. Here's a quote from the book:

“Let us consider an example from Bliss and Fisher (1953). From each of 6 apple trees in an orchard that was sprayed, 25 leaves were selected. On each of the leaves, the number of adult female red mites was counted. Intuitively, we might conclude that this situation is too heterogeneous for a Poisson model to fit; the rates of infestation might be different on different trees and at different locations on the same tree.”

The code below reads in the data, creates a histogram of the data with the pmf from a Poisson model overlaid on top, and calculates the mean and sample variance of the data. If $X \sim \text{Poisson}(\lambda)$ then $E(X) = \text{Var}(X) = \lambda$. In this case, the variance is about twice the mean and the fit does not look good in the plot; these are signs that the Poisson model is not a good fit for these data.

```
mites <- read_csv("http://www.evanlray.com/data/rice/Chapter%208/mites.csv")

## Parsed with column specification:
## cols(
##   num_mites_per_leaf = col_double()
## )

head(mites)

## # A tibble: 6 x 1
##   num_mites_per_leaf
##             <dbl>
## 1                 4
## 2                 4
## 3                 0
## 4                 2
## 5                 1
## 6                 1

mean(mites$num_mites_per_leaf)

## [1] 1.146667

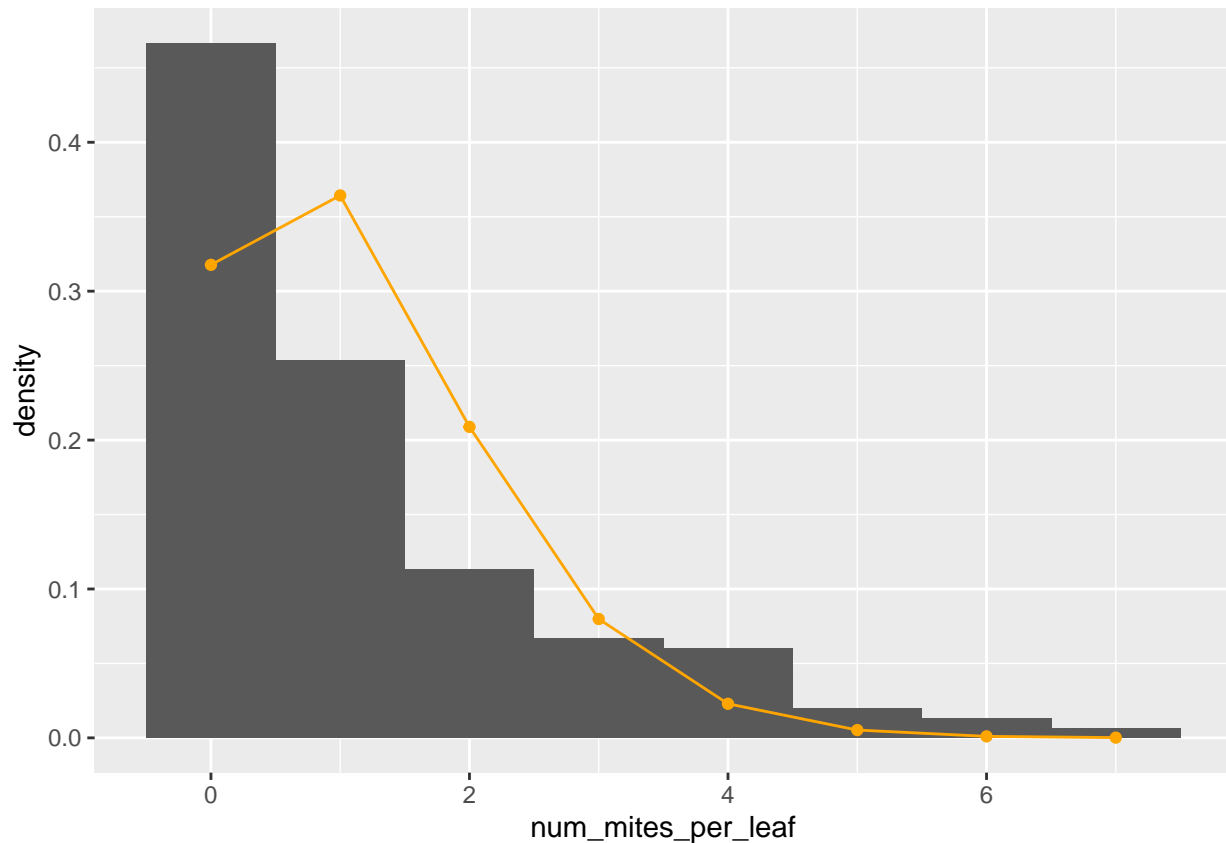
var(mites$num_mites_per_leaf)

## [1] 2.273647

x <- seq(from = 0, to = max(mites$num_mites_per_leaf), by = 1)

poisson_fit <- data.frame(
  x = x,
  fit = dpois(x, lambda = mean(mites$num_mites_per_leaf))
)

ggplot() +
  geom_histogram(data = mites, mapping = aes(x = num_mites_per_leaf, y = ..density..), binwidth = 1, color = "blue", fill = "blue") +
  geom_line(data = poisson_fit, mapping = aes(x = x, y = fit), color = "orange") +
  geom_point(data = poisson_fit, mapping = aes(x = x, y = fit), color = "orange")
```



A common alternative to the Poisson for count data when the variance is larger than the mean is the Negative Binomial distribution. We'll explore using this model

$$X_i^{\text{iid}} \sim \text{Negative Binomial}(\mu, \phi),$$

where X_i is the observed number of mites on leaf number i .

1. Set up model definition in stan

I have set up a skeleton of the stan file, included in this repository. Edit that file now to add necessary declarations and model statements for this model to the data, parameters, and model blocks.

2. Perform estimation

You will need to load the rstan package, set up a list with the data for the stan model, compile the model, and call `optimizing` to obtain the maximum likelihood estimates of the model parameters.

```
# Load the rstan package
library(rstan)
```

```
## Loading required package: StanHeaders
## rstan (Version 2.19.2, 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)
```

```

# Set up list with data Stan will need to know about
stan_data <- list(
  n = nrow(mites),
  x = mites$num_mites_per_leaf
)

# Compile the Stan model definition to an executable. Takes a few seconds to run.
mites_model_compiled <- stan_model(file = "mites_model.stan")

# Call Stan to do optimization
mites_fit <- optimizing(mites_model_compiled,
  data = stan_data,
  seed = 8742,
  init = "random"
)

```

```

mites_fit

## $par
##      mu      phi
## 1.146673 1.024600
##
## $value
## [1] -128.0874
##
## $return_code
## [1] 0
##
## $theta_tilde
##      mu      phi
## [1,] 1.146673 1.0246

```

3. Make a plot

Add a new layer to the plot below showing the pdf of the negative binomial distribution corresponding to the maximum likelihood parameter estimates.

```

x <- seq(from = 0, to = max(mites$num_mites_per_leaf), by = 1)

# Modify the code below to calculate values of the pmf from a
# negative binomial distribution with the parameter estimates from Stan.
# The function to use is dbinom(x, mu = mu, size = phi),
# but you need to extract the estimates of mu and phi from your mites_fit object.
neg_binomial_fit <- data.frame(
  x = x,
  fit = dbinom(x, mu = mites_fit$par["mu"], size = mites_fit$par["phi"])
)

ggplot() +
  geom_histogram(data = mites, mapping = aes(x = num_mites_per_leaf, y = ..density..), binwidth = 1, color = "blue") +
  geom_line(data = neg_binomial_fit, mapping = aes(x = x, y = fit), color = "orange") +
  theme_bw()

```

