Bayesian Model Specification

3: Bayesian Qualitative/Quantitative Inference

David Draper

Department of Applied Mathematics and Statistics
University of California, Santa Cruz

and (1 Jul 2014–30 Sep 2015) *eBay Research Labs*

{draper@ams.ucsc.edu, davdraper@ebay.com}
www.ams.ucsc.edu/∼draper

Short Course (Day 3)
University of Reading (UK)

27 Nov 2014

Bayesian Qual/Quant Inference

Recall from our earlier discussion that if I judge **binary** $(y_1, \ldots, y_n)$ to be part of **infinitely exchangeable sequence**, to be **coherent** my joint predictive distribution $p(y_1, \ldots, y_n)$ must have simple **hierarchical** form

$$\begin{array}{rcl} \theta & \sim & p(\theta) \\ (y_i|\theta) & \stackrel{\text{IID}}{\sim} & \text{Bernoulli}(\theta), \end{array}$$

where $\theta = P(y_i = 1) =$ **limiting value of mean of** y_i in infinite sequence.

Writing $s = (s_1, s_2)$ where s_1 and s_2 are the **numbers of 0s and 1s**, respectively in $(y_1, \ldots, y_n)$, this is **equivalent** to the model

$$\begin{array}{rcl} \theta_2 & \sim & p(\theta_2) \\ (s_2|\theta_2) & \sim & \text{Binomial}(n, \theta_2), \end{array} \qquad (1)$$

where (in a slight change of notation) $\theta_2 = P(y_i = 1)$; i.e., in this simplest case the form of the **likelihood function** (Binomial(n, θ_2)) is determined by **coherence**.

The **likelihood function** for θ_2 in this model is

$$l(\theta_2|y) = c\,\theta_2^{s_2}(1 - \theta_2)^{n - s_2} = c\,\theta_1^{s_1}\theta_2^{s_2}, \qquad (2)$$

from which it's evident that the **conjugate prior** for the **Bernoulli/Binomial likelihood** (the choice of prior having the property that the **posterior** for θ_2 has the same **mathematical form** as the **prior**) is the family of **Beta**(α_1, α_2) densities

$$p(\theta_2) = c\,\theta_2^{\alpha_2 - 1}(1 - \theta_2)^{\alpha_1 - 1} = c\,\theta_1^{\alpha_1 - 1}\theta_2^{\alpha_2 - 1}. \qquad (3)$$

for some $\alpha_1 > 0, \alpha_2 > 0$.

Bayesian Qual/Quant Inference

With this prior the **conjugate updating rule** is evidently

$$\left\{ \begin{array}{c} \theta_2 \sim \text{Beta}(\alpha_1, \alpha_2) \\ (s_2 | \theta_2) \sim \text{Binomial}(n, \theta_2) \end{array} \right\} \rightarrow (\theta_2 | y) \sim \text{Beta}(\alpha_1 + s_1, \alpha_2 + s_2), \tag{4}$$

where s_1 (s_2) is the **number of 0s (1s)** in the data set $y = (y_1, \ldots, y_n)$.

Moreover, given that the **likelihood** represents a **(sample) data set** with s_1 0s and s_2 1s and a **data sample size** of $n = (s_1 + s_2)$, it's clear that

(a) the **Beta**(α_1, α_2) prior acts like a **(prior) data set** with α_1 0s and α_2 1s and a **prior sample size** of $(\alpha_1 + \alpha_2)$, and

(b) to achieve a relatively **diffuse** (**low-information-content**) prior for θ_2 (if that's what **context** suggests I should aim for) I should try to specify α_1 and α_2 **not far from 0**.

Easy **generalization** of all of this: suppose the y_i take on $l \geq 2$ **distinct values** $v = (v_1, \ldots, v_l)$, and let $s = (s_1, \ldots, s_l)$ be the **vector** of **counts** ($s_1 = \#(y_i = v_1)$ and so on).

If I judge the y_i to be part of an **infinitely exchangeable sequence**, then to be **coherent** my joint predictive distribution $p(y_1, \ldots, y_n)$ must have the **hierarchical** form

$$\begin{array}{rcl} \theta & \sim & p(\theta) \\ (s|\theta) & \sim & \text{Multinomial}(n, \theta), \end{array} \tag{5}$$

where $\theta = (\theta_1, \ldots, \theta_l)$ and θ_j is the **limiting relative frequency** of v_j values in the infinite sequence.

Bayesian Qual/Quant Inference

The **likelihood** for (vector) θ in this case has the form

$$l(\theta|y) = c \prod_{j=1}^{l} \theta_j^{s_j}, \tag{6}$$

from which it's evident that the **conjugate prior** for the **Multinomial likelihood** is of the form

$$p(\theta) = c \prod_{j=1}^{l} \theta_j^{\alpha_j - 1}, \tag{7}$$

for some $\alpha = (\alpha_1, \ldots, \alpha_l)$ with $\alpha_j > 0$ for $j = 1, \ldots, l$; this is the **Dirichlet**(α) distribution, a **multivariate generalization** of the Beta family.

Here the **conjugate updating rule** is

$$\left\{ \begin{array}{c} \theta \sim \text{Dirichlet}(\alpha) \\ (s|\theta) \sim \text{Multinomial}(n, \theta) \end{array} \right\} \rightarrow (\theta|y) \sim \text{Dirichlet}(\alpha + s), \tag{8}$$

where $s = (s_1, \ldots, s_l)$ and s_j is the **number of** v_j **values** ($j = 1, \ldots, l$) in the data set $y = (y_1, \ldots, y_n)$.

Furthermore, by **direct analogy** with the $l = 2$ case,

(a) the **Dirichlet**(α) prior acts like a **(prior) data set** with α_j v_j values ($j = 1, \ldots, l$) and a **prior sample size** of $\sum_{j=1}^{l} \alpha_j$, and

(b) to achieve a relatively **diffuse (low-information-content**) prior for θ (if that's what **context** suggests I should aim for) I should try to choose all of the α_j **not far from 0**.

Bayesian Qual/Quant Inference

To **summarize**:

(A) if the **data vector** $y = (y_1, \ldots, y_n)$ takes on l **distinct** values $v = (v_1, \ldots, v_l)$ (**real numbers or not**) and I judge (my uncertainty about) the infinite sequence $(y_1, y_2, \ldots)$ to be **exchangeable**, then (by a **representation theorem** of de Finetti) **coherence** compels me (i) to **think about** the quantities $\theta = (\theta_1, \ldots, \theta_l)$, where θ_j is the **limiting relative frequency** of the v_j values in the infinite sequence, and (ii) to **adopt** the Multinomial model

$$\begin{aligned} \theta & \sim p(\theta) \\ p(y_i|\theta) & = c \prod_{j=1}^{l} \theta_j^{s_j}, \end{aligned} \qquad (9)$$

where s_j is the **number** of y_i values equal to v_j;

(B) if context suggests a **diffuse** prior for θ a convenient (**conjugate**) choice is **Dirichlet**(α) with $\alpha = (\alpha_1, \ldots, \alpha_l)$ and all of the α_j **positive but close to 0**; and

(C) with a **Dirichlet**(α) prior for θ the **posterior** is **Dirichlet**(α'), where $s = (s_1, \ldots, s_l)$ and $\alpha' = (\alpha + s)$.

Note, remarkably, that the v_j values themselves **make no appearance** in the model; this modeling approach is **natural** with **categorical** outcomes but can also be used when the v_j are **real numbers**.

For example, for **real-valued** y_i, if (as in the **IHGA case study** in Part 1) interest focuses on the **(underlying population) mean** in the infinite sequence $(y_1, y_2, \ldots)$, this is $\mu_y = \sum_{j=1}^{l} \theta_j \, v_j$, which is just a **linear function** of the θ_j with **known coefficients** v_j.

Bayesian Qual/Quant Inference

This fact makes it possible to draw an **analogy** with the **distribution-free** methods that are at the heart of **frequentist non-parametric** inference: when your **outcome variable** takes on a **finite number** of **real** values v_j, **exchangeability** compels a **Multinomial likelihood** on the **underlying frequencies** with which the v_j occur; you are not required to build a **parametric model** (e.g., normal, lognormal, ...) on the y_i values themselves.

In this sense, therefore, model (14)—particularly with the **conjugate Dirichlet** prior—can serve as a kind of **low-technology Bayesian non-parametric** modeling: this is the basis of the **Bayesian bootstrap** (Rubin 1981).

Moreover, if you're **in a hurry** and you're already familiar with `WinBUGS` you can readily carry out **inference** about quantities like μ_y above in that environment, but there's **no need to do MCMC** here: **ordinary Monte Carlo** (MC) sampling from the **Dirichlet**(α') posterior distribution is perfectly **straightforward**, e.g., in `R`, based on the following **fact**:

To generate a **random draw** $\theta = (\theta_1, \ldots, \theta_l)$ from the **Dirichlet**(α') distribution, with $\alpha' = (\alpha'_1, \ldots, \alpha'_l)$, **independently draw**

$$g_j \stackrel{\text{indep}}{\sim} \Gamma(\alpha'_j, \beta), \quad j = 1, \ldots, l \tag{10}$$

(where $\Gamma(a, b)$ is the **Gamma distribution** with parameters a and b) and compute

$$\theta_j = \frac{g_j}{\sum_{m=1}^{l} g_j}. \tag{11}$$

Any $\beta > 0$ will do in this calculation; $\beta = 1$ is a **good choice** that leads to **fast random number generation**.

Bayesian Qual/Quant Inference

The **downloadable version** of `R` doesn't have a **built-in function** for making **Dirichlet draws**, but it's easy to write one:

```
rdirichlet = function( n.sim, alpha ) {

  l = length( alpha )

  theta = matrix( 0, n.sim, l )

  for ( j in 1:l ) {

    theta[ , j ] = rgamma( n.sim, alpha[ j ], 1 )

  }

  theta = theta / apply( theta, 1, sum )

  return( theta )

}
```

The **Dirichlet**(α) distribution has the following **moments**: if $\theta \sim$ Dirichlet(α) then

$$E(\theta_j) = \frac{\alpha_j}{\alpha_0}, \ V(\theta_j) = \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_0 + 1)}, \ C(\theta_j, \theta_{j'}) = -\frac{\alpha_j \alpha_{j'}}{\alpha_0^2(\alpha_0 + 1)},$$

where $\alpha_0 = \sum_{j=1}^{l} \alpha_j$ (note the **negative correlation** between components of θ).

This can be used to **test** the function above:

Bayesian Qual/Quant Inference

```
> alpha = c( 5.0, 1.0, 2.0 )

> alpha.0 = sum( alpha )

> test = rdirichlet( 100000, alpha )    # 15 seconds at 550 Unix MHz

> apply( test, 2, mean )

[1] 0.6258544 0.1247550 0.2493905

> alpha / alpha.0

[1] 0.625 0.125 0.250

> apply( test, 2, var )

[1] 0.02603293 0.01216358 0.02071587

> alpha * ( alpha.0 - alpha ) / ( alpha.0^2 * ( alpha.0 + 1 ) )

[1] 0.02604167 0.01215278 0.02083333

> cov( test )

             [,1]         [,2]         [,3]
[1,]  0.026032929 -0.008740319 -0.017292610
[2,] -0.008740319  0.012163577 -0.003423259
[3,] -0.017292610 -0.003423259  0.020715869

> - outer( alpha, alpha, "*" ) / ( alpha.0^2 * ( alpha.0 + 1 ) )

             [,1]         [,2]         [,3]
[1,] -0.043402778 -0.008680556 -0.017361111
[2,] -0.008680556 -0.001736111 -0.003472222     # ignore diagonals
[3,] -0.017361111 -0.003472222 -0.006944444
```

Bayesian Qual/Quant Inference

Example: re-analysis of **IHGA data** from Part 1; recall **policy** and **clinical interest** focused on $\eta = \frac{\mu_E}{\mu_C}$.

	Number of Hospitalizations										
Group	0	1	2	3	4	5	6	7	n	Mean	SD
Control	138	77	46	12	8	4	0	2	287	0.944	1.24
Experimental	147	83	37	13	3	1	1	0	285	0.768	1.01

In this **two-independent-samples** setting I can apply de Finetti's representation theorem **twice, in parallel**, on the C and E data.

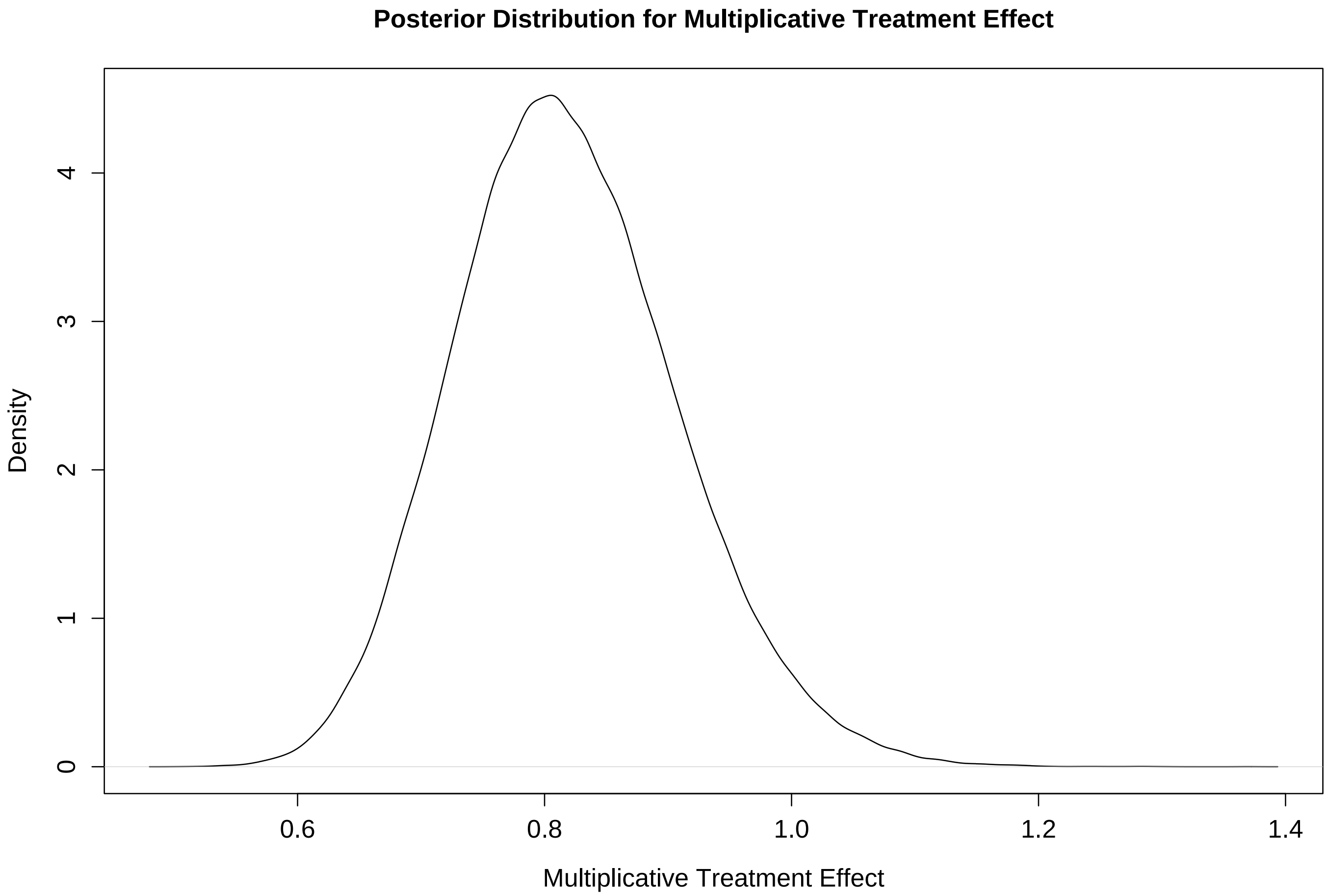
I don't know much about the **underlying frequencies** of $0, 1, \ldots, 7$ hospitalizations under C and E **external** to the data, so I'll use a **Dirichlet**$(\epsilon, \ldots, \epsilon)$ **prior** for both θ_C and θ_E with $\epsilon = 0.001$, leading to a **Dirichlet**$(138.001, \ldots, 2.001)$ **posterior** for θ_C and a **Dirichlet**$(147.001, \ldots, 0.001)$ **posterior** for θ_E (other small positive choices of ϵ yield **similar results**).

```
> alpha.C = c( 138.001, 77.001, 46.001, 12.001, 8.001, 4.001, 0.001,
    2.001 )

> alpha.E = c( 147.001, 83.001, 37.001, 13.001, 3.001, 1.001, 1.001,
    0.001 )

> theta.C = rdirichlet( 100000, alpha.C )   # 17 sec at 550 Unix MHz

> theta.E = rdirichlet( 100000, alpha.E )   # also 17 sec

> print( post.mean.theta.C = apply( theta.C, 2, mean ) )

[1] 4.808015e-01 2.683458e-01 1.603179e-01 4.176976e-02 2.784911e-02
[6] 1.395287e-02 3.180905e-06 6.959859e-03

> print( post.SD.theta.C <- apply( theta.C, 2, sd ) )

[1] 0.0294142963 0.0261001259 0.0216552661 0.0117925465 0.0096747630
[6] 0.0069121507 0.0001017203 0.0048757485
```

Bayesian Qual/Quant Inference

```
> print( post.mean.theta.E <- apply( theta.E, 2, mean ) )

[1] 5.156872e-01 2.913022e-01 1.298337e-01 4.560130e-02 1.054681e-02
[6] 3.518699e-03 3.506762e-03 3.356346e-06

> print( post.SD.theta.E <- apply( theta.E, 2, sd ) )

[1] 0.029593047 0.026915644 0.019859213 0.012302252 0.006027157
[6] 0.003501568 0.003487824 0.000111565

> mean.effect.C <- theta.C %*% ( 0:7 )

> mean.effect.E <- theta.E %*% ( 0:7 )

> mult.effect <- mean.effect.E / effect.C

> print( post.mean.mult.effect <- mean( mult.effect ) )

[1] 0.8189195

> print( post.SD.mult.effect <- sd( mult.effect ) )

[1] 0.08998323

> quantile( mult.effect, probs = c( 0.0, 0.025, 0.5, 0.975, 1.0 ) )

       0%      2.5%       50%     97.5%      100%
0.5037150 0.6571343 0.8138080 1.0093222 1.3868332

> postscript( "mult.effect.ps" )

> plot( density( mult.effect, n = 2048 ), type = ’l’, cex.lab = 1.25,
    xlab = ’Multiplicative Treatment Effect’, cex.axis = 1.25,
    main = ’Posterior Distribution for Multiplicative Treatment Effect’,
    cex.main = 1.25 )

> dev.off( )
```

Bayesian Qual/Quant Inference



Model	Posterior Mean	Posterior SD	Central 95% Interval
REPR	0.830	0.0921	$(0.665, 1.02)$
Dir-Mult	0.819	0.0900	$(0.657, 1.01)$

In this example the **low-tech BNP**, **Dirichlet-Multinomial**, **exchangeability-plus-diffuse-prior-information** model has **reproduced** the **parametric REPR results** almost exactly and without a **complicated search through model space** for a **"good"** model.

$\boxed{\textbf{NB}}$ This **approach** is an **application** of the **Bayesian bootstrap** (Rubin 1981), which (for **complete validity**) includes the **assumption** that the **observed** y_i **values form** a **complete set** of $\{$**all possible values the outcome** y **could take on**$\}$.

Bayesian Qual/Quant Inference

This is **clearly not true** in the **IHGA case study**, and yet **in that case** the **Bayesian qualitative/quantitative inferential approach** did a **terrific job** of **reproducing what we will later see** is an **excellent parametric model** for the **IHGA data, without any parametric modeling assumptions**.