Appendix to Ordered Beta Regression: A Parsimonious, Well-Fitting Model for Survey Sliders and Visual Analog Scales

Robert Kubinec
New York University Abu Dhabi

March 7th, 2020

1 Models Without Degenerate Responses

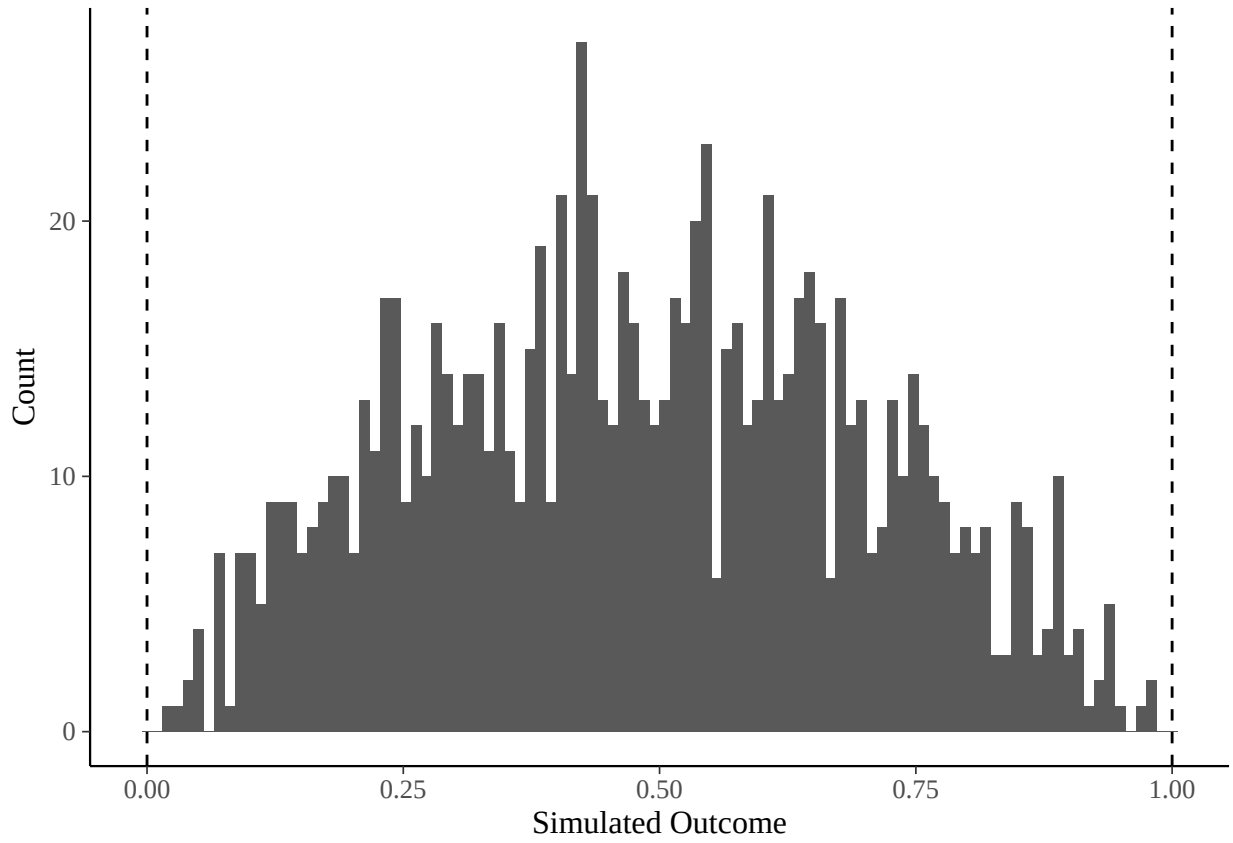
Not only is it possible to fit the ordered beta regression mdoel to data without observations at the bounds, but it is advisable to do so if there is even a remote chance that such observations could be observed. For example, it may well be that a certain realization of the data contains observations that just so happened to not reach the bounds and are in the (0.01,0.99) interval. We could imagine this arising in a feeling thermometer/VAS scale where respondents' preferences tend to fairly clustered around the midpoint of the scale. However, a future sample of this same data could end up with observations at the bounds. It would be problematic in this case to fit only a Beta regression to the current data as the estimates would later be incomparable to estimates of future data with observations at the bounds.

While this scenario does not necessarily need to happen, it is enough of a motivation to fit the ordered beta regression model even in situations where there are no observations at the bounds (or perhaps only at one bound). The costs of doing so, both in terms of inference and computation, are quite low. Because the cutpoints were assigned a weakly informative prior, *they are identified without any data.* As a result, if a model is fit without any observations on the bounds, the cutpoints will end up in the far corners of the distribution, say at 0.001 and 0.999, but they will still exist and the posterior predictive distribution can produce them with some small probability. If future data was added to the sample incorporating observations at the bounds, the combined estimates would be interpretable and the cutpoints would adjust to handle the new data.

To demonstrate this, I simulate data from a model with widely spaced cutpoints where I remove any of the few observations that end up at the bounds:

```r
N <- 1000

X <- rnorm(N,runif(1,-2,2),1)

X_beta <- -1
eta <- X*X_beta

# ancillary parameter of beta distribution
# high clustering
phi <- 70

# predictor for ordered model
mu1 <- eta
# predictor for beta regression
mu2 <- eta

# wide cutpoints on logit scale
cutpoints <- c(-8,8)

# probabilities for three possible categories (0, proportion, 1)
low <- 1-plogis(mu2 - cutpoints[1])
middle <- plogis(mu2-cutpoints[1]) - plogis(mu2-cutpoints[2])
high <- plogis(mu2 - cutpoints[2])

# we'll assume the same eta was used to generate outcomes

out_beta <- rbeta(N,plogis(mu1) * phi, (1 - plogis(mu1)) * phi)

# now determine which one we get for each observation
outcomes <- sapply(1:N, function(i) {
  sample(1:3,size=1,prob=c(low[i],middle[i],high[i]))
})

# now combine binary (0/1) with proportion (beta)
```

```r
final_out <- sapply(1:length(outcomes),function(i) {
  if(outcomes[i]==1) {
    return(0)
  } else if(outcomes[i]==2) {
    return(out_beta[i])
  } else {
    return(1)
  }
})

# remove residual 1/0s
remove_degen <- final_out>0 & final_out<1
final_out <- final_out[remove_degen]
X <- X[remove_degen]

tibble(x=final_out) %>%
ggplot(aes(x=final_out)) +
  geom_histogram(bins=100) +
  geom_vline(xintercept = 0,linetype=2) +
  geom_vline(xintercept = 1,linetype=2) +
  theme(panel.grid=element_blank(),
        panel.background=element_blank()) +
  ylab("Count") +
  xlab("Simulated Outcome")
```



We can then model this distribution as follows:

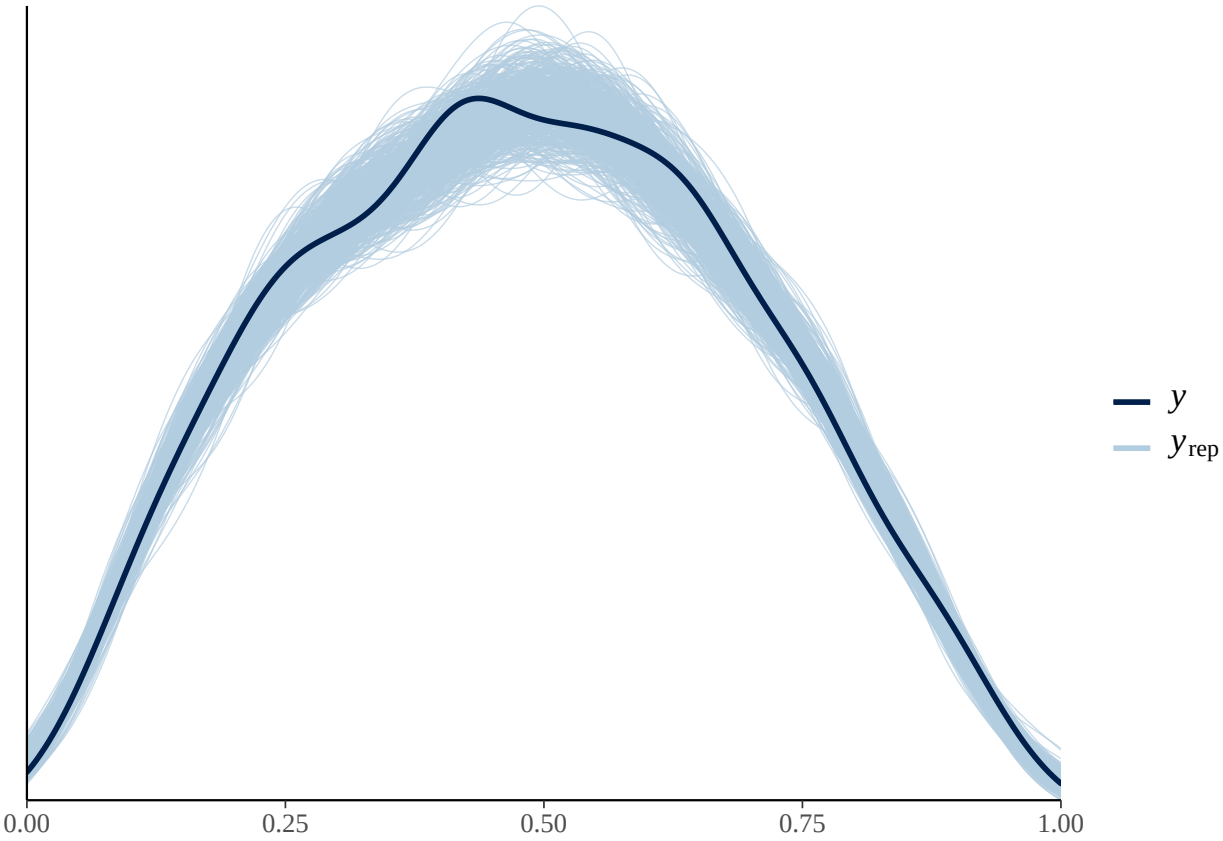
```
to_bl <- list(N_degen=sum(final_out %in% c(0,1)),
              N_prop=sum(final_out>0 & final_out<1),
              X=1,
              outcome_prop=final_out[final_out>0 & final_out<1],
              outcome_degen=final_out[final_out %in% c(0,1)],
              covar_prop=as.matrix(X),
              covar_degen=as.matrix(X[final_out %in% c(0,1)]),
              N_pred_degen=sum(final_out %in% c(0,1)),
              N_pred_prop=sum(final_out>0 & final_out<1),
              indices_degen=array(dim=0),
              indices_prop=1:(sum(final_out>0 & final_out<1)),
              run_gen=1)


 fit_model <- sampling(ord_beta_mod,data=to_bl,seed=random_seed,
                       refresh=0,
```

```
                        chains=1,cores=1,iter=1000,pars=c("regen_all","X_beta","ord_log","cutpoints"))

  cutpoints <- as.matrix(fit_model,"cutpoints")

  print(fit_model,c("X_beta","cutpoints"))
```

```
## Inference for Stan model: beta_logit.
## 1 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=500.
##
##               mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## X_beta[1]    -1.01    0.00 0.01 -1.03 -1.02 -1.01 -1.01 -0.99   561 1.01
## cutpoints[1] -7.39    0.07 0.94 -9.79 -7.85 -7.28 -6.73 -5.95   186 1.00
## cutpoints[2]  7.16    0.04 0.92  5.69  6.49  7.00  7.68  9.39   529 1.00
##
## Samples were drawn using NUTS(diag_e) at Sat Mar  7 16:13:28 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

We can see from the model results that our coefficient `X_beta` was estimated without bias (equal to -1). The cutpoints were estimated with a little bit of bias due to the censoring we did on the outcome variable, but are still quite close to the original values. Furthermore, they are estimated at extremes – the lower cutpoint is 6×10^{-4} and the upper cutpoint is 0.9992. As this example indicates, there is no reason not to fit a model with no observations at the bounds. The cutpoints are still identified and the model converges without a problem. Furthermore, we can then still simulate observations at the bounds from the posterior predictive distribution:

```
ppc_dens_overlay(final_out,as.matrix(fit_model,"regen_all"))
```


0.00
0.25
0.50
0.75
1.00
y
y rep