Case Study/Application

Raven McKnight

1 Background

Metro Transit is the primary transit provider in the Minneapolis-Saint Paul metropolitan area. The agency is interested in predicting transit demand across the region in order to guide route planning and service provision. Currently, Metro Transit uses a set of 5 Transit Market Areas to estimate spatial demand. The Transit Market Areas are writen into the official Transportation Planning Policy (TPP) and are calculated using a simple linear regression. Following the notation of the TPP, the formula for determing Transit Market Areas is expressed

$$\begin{aligned}\text{Transit Market Index } &= 0.64 * \text{Population Density} + 0.20 * \text{Employment Density} + \\ &\quad 0.23 * \text{Intersection Density} + 0.11 * \text{Automobile Availability}\end{aligned}$$

where each predictor is logged and scaled by developed land acreage per census block group. Here, automobile availability refers to the number of adults over age 16 less the total number of automobiles available in a block group (scaled by developed land acreage). There is an additional indicator variable, omitted in this notation, for the census block group containing the MSP International Airport. For more documentation of the official Transit Market Areas, refere to the TPPs Appendix G.

Census block groups are split into 5 market areas based on the Transit Market Index described above, where the highest market area (Market Area 1) is expected to support high-frequency, all-day service and the lowest market area (Market Area 5) is expected to support peak commuter express service, if that.

The Transit Market Index values are geographically smoothed to create more-or-less concentric Transit Market Areas.

2 Data Description

2.1 Metro Transit Data

2.2 Covariates

3 Data Preparation

4 Modeling

4.1 Model 1: Poisson Regression

The baseline Poisson regression can be fit easily in Stan. This simplest model in this case study can be written

$$Y_i \sim \text{Poisson}(E_i \lambda_i)$$
$$\eta_i = log(\lambda_i) = \beta_0 + \sum_{k=1}^{19} x_k^T \beta_k$$
$$\beta_0, \beta \sim \text{Normal}(0, 1)$$

where Y_i is the number of boardings in census block group i in 2017, E_i is the number of bus stops made in block group i in 2017, and parameter vector x corresponds to the covariates discussed above. **Figure ...** shows this model written in Stan. The model was fit with four chains of 10,000 iterations each, although convergence was confirmed after only 2,000 iterations. After burn-in, or the warmup period, Model 1 contains 20,000 samples.

```
## data {
##   // number obs
##   int<lower=0> N;
##   // response
##   int<lower=0> y[N];
##   // "offset" (number of stops)
##   vector<lower=0>[N] E;
##   // number of covariates
##   int<lower=1> K;
##   // covariates
##   matrix[N, K] x;
## }
## transformed data {
##   vector[N] log_E = log(E);
## }
## parameters {
##   // intercept
##   real beta_0;
##   // covariates
##   vector[K] beta;
## }
## transformed parameters{
##   // latent function variables
##   vector[N] f;
##   f = log_E + beta_0 + x*beta;
## }
## model {
##   /// model
##   y ~ poisson_log(f);
##   // prior on betas
##   beta_0 ~ normal(0.0, 3);
##   beta ~ normal(0.0, 1);
## }
```

Table 1 reports 95% confidence intervals for each of the 19 parameters included in x as well as diagnostics $\hat{R}$ and n_eff. For Model 1, $n_eff > N$, which indicates "better than independent" (https://discourse.mc-stan.org/t/n-eff-bda3-vs-stan/2608/50, citing a forum? by stan developers?) draws. Simply put, the diagnostics suggest that we can trust this model.

Note that many β_k are close to zero, but few contain zero within their 95% confidence interval. The small magnitude of β_k estimates are because parameters x are normalized to have standard deviation equal to 1.

Based on the diagnostics above, we have reasonable confirmation that the MCMC algorithm fit the model

Parameter	Rhat	n_eff	mean	sd	2.5%	50%	97.5%
beta_0	1.00	25223	-1.38	0.00	-1.39	-1.38	-1.37
beta[1]	1.00	20629	0.05	0.01	0.04	0.05	0.07
beta[2]	1.00	22559	0.07	0.00	0.06	0.07	0.08
beta[3]	1.00	22078	-0.01	0.00	-0.02	-0.01	0.00
beta[4]	1.00	20946	-0.02	0.00	-0.02	-0.02	-0.01
beta[5]	1.00	20568	-0.02	0.00	-0.03	-0.02	-0.01
beta[6]	1.00	16294	0.21	0.01	0.20	0.21	0.22
beta[7]	1.00	20239	0.04	0.00	0.03	0.04	0.05
beta[8]	1.00	18051	-0.01	0.00	-0.02	-0.01	-0.00
beta[9]	1.00	21361	0.10	0.00	0.10	0.10	0.11
beta[10]	1.00	16390	-0.01	0.00	-0.01	-0.01	-0.00
beta[11]	1.00	17378	0.10	0.00	0.09	0.10	0.10
beta[12]	1.00	21918	-0.11	0.00	-0.12	-0.11	-0.11
beta[13]	1.00	26748	-0.08	0.00	-0.09	-0.08	-0.07
beta[14]	1.00	20707	-0.00	0.01	-0.01	-0.00	0.01
beta[15]	1.00	15443	-0.02	0.00	-0.03	-0.02	-0.01
beta[16]	1.00	15498	0.13	0.00	0.12	0.13	0.14
beta[17]	1.00	23302	-0.09	0.00	-0.09	-0.09	-0.08
beta[18]	1.00	26155	0.03	0.00	0.02	0.03	0.04
beta[19]	1.00	24639	0.01	0.00	0.01	0.01	0.02

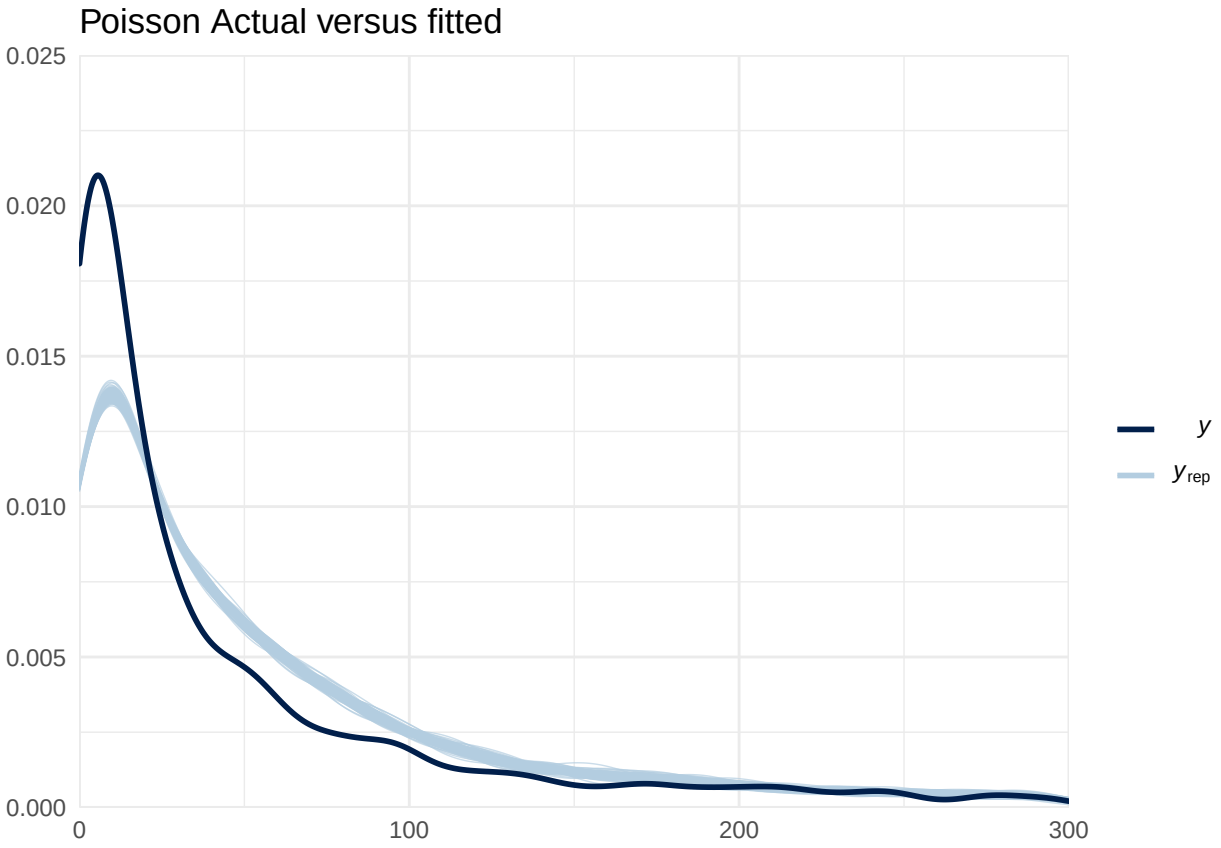
Table 1: Table 1: Parameter estimates and HMC/NUTS diagnostics for Model 1, the simple Poisson GLM

appropriately. Beyond convergence, however, we need to examine model fit. **Figure ___** shows the actual distribution of Y_i in dark blue with 100 samples from values of $\hat{Y}_i$ simulated by Model 1 in light blue. Immediately, it is clear that Model 1 is underperforming. Model 1 underestimates the number of block groups with low ridership and overestimates the number of block groups with high ridership, and the ridership within those block groups. Table 2 reports minima and maxima of Y_i and $\hat{Y}_i$ for comparison.

```
## Warning in `[.data.table`(mod_dat, , -c("GEOID", "daily_boards",
## "daily_stops", : column(s) not removed because not found: [daily_alights,
## num_interpolated, num_routes, daily_activity]
```

```
## Warning: Removed 8348 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 93 rows containing non-finite values (stat_density).
```



4.2 Horseshoe Prior

The model above does a decent job predicting ridership but there are many possible improvements. One shortcoming of the simplest Poisson model is that the normal priors on β make it challenging to determine which coefficients are truly significantly different from 0. This is where the regularized horseshoe prior comes into play.

This model can be expressed in Stan as pictured in **Figure. . . .** The Stan program for this model was adapted from **Vehtari..**.

4.3 Divergences

After running four chains for 10,000 iterations each, this model does produce some *divergences*. **This means I should explain what a divergence is** To be exact, this model produces 30 divergent transtions out of 20,000 post-warmup transitions, for a divergent rate of 0.15%. Divergencies are a diagnostic particular to the HMC and NUTS samplers used by Stan.

4.4 Variables Selected

Perhaps surprisingly, the use of horseshoe priors only suggests the removal of four variables: **,** , **,** . This is likely because the initial set of covariates were selected based on domain knowledge and therefore, it was unlikely for β to be exceedingly sparse. The application of the horseshoe prior would be more informative in a data set with more parameters, and with less insight into the selection of the initial covariate vector x.

4.5 Overdispersion

For computational efficiency, we can drop the horseshoe priors for this model. Armed with the information provided by our fist two models, we can simply give β tigher normal priors.

4.6 Spatial Structure