Improving multilevel regression and poststratification with structured priors

Yuxiang Gao, University of Toronto
Lauren Kennedy, Columbia University
Daniel Simpson, University of Toronto

Contents

1 Introduction

We propose structured priors for multilevel regression and poststratification (MRP), which aim to reduce estimation bias by introducing more intelligent shrinkage of posterior estimates. These proposed priors are Gaussian Markov random fields (GMRF) that model the underlying structure of categorical covariates. In our paper, we show through simulation studies that structured priors do indeed reduce posterior MRP bias, with a secondary benefit of reduced posterior variance for the underlying structured covariates.

2 Overview of MRP

Suppose that the population can be split into K categorical variables and that the k^{th} categorical variable has J_k categories. Hence the population can be represented by $\prod_{k=1}^{K} J_k$ cells. For every cell j, there is a known population size N_j.

2.1 Step 1: Multilevel regression step

Fit the hierarchical logistic regression model below to get estimated population averages θ_j for every cell $j \in \{1, \ldots, J\}$. The hierarchical logistic regression portion of MRP has a set of varying intrecepts $\{\alpha_j^k\}_{j=1}^{J_k}$ for each categorical covariate k, which have the effect of partially pooling each θ_j towards a globally-fitted regression model, $X_j\beta$, with sparse cells benefiting the most from this regularization.

$$
\begin{aligned}
\Pr(y_i = 1) &= \text{logit}^{-1}\left(X_i\beta + \sum_{k=1}^{K} \alpha^k_{j[i]}\right), \text{ for } i = 1, \dots, n \\
\alpha^k_j \mid \sigma^k &\overset{\text{ind.}}{\sim} \text{N}(0, (\sigma^k)^2), \text{ for } k = 1, \dots, K, \ j = 1, \dots, J_k \\
\sigma^k &\sim \text{N}_+(0, 1), \text{ for } \text{ for } k = 1, \dots, J \\
\beta &\sim \text{N}(0, 1),
\end{aligned}
$$

2.2 Step 2: Poststratification step

Using the known population sizes N_j of each cell j, poststratify to get posterior preference probabilities at the subpopulation level. The poststratification portion of MRP adjusts for nonresponse in the population by taking into account the sizes of every cell l relative to the total population size $N = \sum_{j=1}^{J} N_j$. Another way to interpret poststratification is as a weighted average of cell-wise posterior preferences, where the weighting scheme is determined by the size of each cell in the population. Smaller cells get downweighted and larger cells get upweighted. The final result is a more accurate estimate in the presence of non-representative data.

$\theta_S = \dfrac{\sum_{j \in S} N_j \theta_j}{\sum_{j \in S} N_j}$, where S is some subset of the population defined based on the poststratification variables.

3 Proposed improvement for MRP using structured prior distributions

The classical specification of MRP has independertly normally distributed random effects for categorical covariates. We will proceed to as follows to specify structured prior distributions for such categorical covariates in a MRP model:

Case 1: If w e do not want to model any structure in a categorical covariate, we model its varying intercepts as independently normally distributed.

Case 2: If there is underying structure we would like to model in a covariate, and spatial smoothing using this structure seems reasonable for the outcome of interest, then we use an apprpriate GMRF as a prior distribution for this batch of varying intercepts.

More complex prior structure allows for nonuniform information-borrowing in the presence of nonrepresentative surveys from a population. The proposal of using structured priors aims to reduce bias for MRP estimates in extremely nonrepresentative data regimes.

4 Simulation setup with structured priors

For the purpose of explaining our proposed method of MRP using structured priors, we'll assume that the population can be described by 2 covariates of age and income, where age ranges from 21–80 and income has 4 categories. Let $\alpha^{\text{Age Cat.}}_{j[i]}$ be the age category random effect for the i^{th} individual, and let $\alpha^{\text{Income}}_{j[i]}$ be the income random effect for the i^{th} individual.

For all three prior specifications of MRP, the link function exists as

$$\Pr(y_i = 1) = \text{logit}^{-1}\left(\beta_0 + \alpha^{\text{Age Cat.}}_{j[i]} + \alpha^{\text{Income}}_{j[i]}.\right), \text{ for } i = 1, \dots, n. \tag{1}$$

All three prior specifications of MRP will have independent random effects for the income covariate, as seen in Equation (2).

$$\begin{aligned} \alpha_j^{\text{Income}} \mid \sigma^{\text{Income}} &\overset{\text{ind.}}{\sim} \text{N}(0, (\sigma^{\text{Income}})^2), \text{ for } j = 1, \ldots, 4 \\ \sigma^{\text{Income}} &\sim \text{N}_+(0, 1). \end{aligned} \tag{2}$$

The *baseline specification* is the classical prior specification used in MRP and has independent random effects for all categorical covariates. This will specify independent random effects for the age category covariate as seen in Equation (3):

$$\begin{aligned} \alpha_j^{\text{Age Cat.}} \mid \sigma^{\text{Age Cat.}} &\overset{\text{ind.}}{\sim} \text{N}(0, (\sigma^{\text{Age Cat.}})^2), \text{ for } j = 1, \ldots, 12 \\ \sigma^{\text{Age Cat.}} &\sim \text{N}_+(0, 1) \end{aligned} \tag{3}$$

4.1 Structured prior specifications

A straightforward random variable that models the ordinal structure of age category is a first-order autoregressive process, and we propose this as our prior distribution for the covariate age category. This is defined as the *autoregressive specification* in Equation (4) below. The prior distribution imposed on ρ has support in $[-1, 1]$, resulting in stationary for the autoregressive process.

$$\begin{aligned} \alpha_1^{\text{Age Cat.}} \mid \rho, \sigma^{\text{Age Cat.}} &\sim \text{N}(0, \frac{1}{1-\rho^2}(\sigma^{\text{Age Cat.}})^2) \\ \alpha_j^{\text{Age Cat.}} \mid \alpha_{j-1}^{\text{Age Cat.}}, \ldots, \alpha_1^{\text{Age Cat.}}, \rho, \sigma^{\text{Age Cat.}} &\sim \text{N}(\rho\alpha_{j-1}^{\text{Age Cat.}}, (\sigma^{\text{Age Cat.}})^2), \text{ for } j = 2, \ldots, 12 \\ \sigma^{\text{Age Cat.}} &\sim \text{N}_+(0, 1) \\ (\rho + 1)/2 &\sim \text{Beta}(0.5, 0.5). \end{aligned} \tag{4}$$

The third proposed structured prior distribution for age category is a first-order random walk process. This will be defined as the *random walk specification* in Equation (5). Note that the random walk specification is a special case of the autoregressive specification when ρ is fixed as 1. The constraint $\sum_{j=1}^{12} \alpha_j^{\text{Age Cat.}} = 0$ ensures that the joint distribution for the first-order random walk process is identifiable.

$$\begin{aligned} \alpha_j^{\text{Age Cat.}} \mid \alpha_{j-1}^{\text{Age Cat.}}, \ldots, \alpha_1^{\text{Age Cat.}}, \sigma^{\text{Age Cat.}} &\sim \text{N}(\alpha_{j-1}^{\text{Age Cat.}}, (\sigma^{\text{Age Cat.}})^2), \text{ for } j = 2, \ldots, J \\ \sigma^{\text{Age Cat.}} &\sim \text{N}_+(0, 1) \\ \sum_{j=1}^{J} \alpha_j^{\text{Age Cat.}} &= 0. \end{aligned} \tag{5}$$

4.2 Defining the poststratification matrix for the population

Table 1 is the poststratification matrix that defines the population, when the probability of response for people of ages 61 - 80 is $p = 0.1$.

Table 1: Full poststratification matrix

Age	Income	Population	Probability of response	True preference	Age Category
21	1	166667	0.025	0.09975049	1
22	1	166667	0.025	0.12345263	1
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
80	4	666667	0.225	0.7064137	12

4.3 Fitting the three prior specifications on various survey data regimes

For every different value of p, we will retrieve a survey sample of size sample_size from the poststratification matrix, and then fit the three prior specifications of MRP for that sample. We will perform this runs times for every p. Running the script poststrat_pipeline_testing_age3_v2_markdown.R performs this.

Near the top of the script poststrat_pipeline_testing_age3_v2_markdown.R, we can modify the grid the p ranges over. This is the variable r. Currently, the setup has $p \in \{0.1, 0.2, \ldots, 0.9\}$. Changing this p results in the sample ranging from under-representing to over-representing people of ages 61-80. As well, age is discretized into age_grouping_multiplier. At the top of the script poststrat_pipeline_testing_age3_v2_markdown.R, we have

```
save_ridgeplots = TRUE
sample_size = 100
runs = 1
r = 0.9
store_stanobjects = FALSE # do you want to store fitted stan model after each run?
income_multiplier = 1 # partitions income into more categories
age_grouping_multiplier = 12 # number of age categories. make sure this can divide 60
```

Running poststrat_pipeline_testing_age3_v2_markdown.R with the current setup takes a full day to run and posterior summary statistics for each p is saved as .RData files. Increasing sample_size to 500 will result in the script taking two full days to run.

4.3.1 Sampling scheme

The sampling scheme is based on the following lines of code below, which is repeated for each simulation iteration in every p:

```
# get a sample from the poststratrification matrix
sample_ = sample(length(p_age) * length(p_income),
                 sample_size,
                 replace = TRUE,
                 prob = (poststrat_final$p_response * strat_final$N)/
                   sum((poststrat_final$p_response * poststrat_final$N)) )

# get response of sample
if (response_binary == TRUE) { # binary response
  y_sample_ = rbinom(sample_size, 1,
                     poststrat_final$true_pref[sample_])
}else{ # normal response
  y_sample_ = rnorm(sample_size,
                    logit(poststrat_final$true_pref[sample_]), response_normal_sd)
}

# get covariates for every row of sample
age_sample = poststrat_final[sample_, 1]
income_sample = poststrat_final[sample_, 2]

sample_final = data.frame(pref = y_sample_,
                          age = age_sample,
                          income = income_sample)



sample_final_ = inner_join(x = sample_final, y = age_group,
                           by = "age")
```

```
# stan needs numeric entries, not factors
sample_final_$age_cat = as.numeric(as.character(sample_final_$age_cat))
```

`sample_final_` returns a sample of binary responses from Table 1. We will use this sample to fit the three prior specifications.

4.3.2 Fitting the baseline prior specification

The baseline specification is fit as `fit_baseline`:

```
    # posterior sampling of baseline specification
    fit_baseline = sampling(m_baseline,
                            data = list(N = dim(sample_final_)[1],
                                        N_groups_age = age_grouping_multiplier,
                                        N_groups_income = 4 * income_multiplier,
                                        age = sample_final_$age_cat,
                                        income = sample_final_$income,
                                        y = sample_final_$pref),
                            iter=iterations, chains=num_chains,
                            control=list(max_treedepth=15, adapt_delta=0.99),
                            seed = 21,
                            chain_id = num_chains*3*(k-1) + 1 +
                              num_chains*3*(r_numericindex[p_counter] - 1)*(runs - 1) +
                              num_chains*3*(r_numericindex[p_counter] - 1))
```

4.3.3 Fitting the autoregressive prior specification

The autoregressive specification is fit as `fit_ar`:

```
    # posterior sampling of autoregressive specification
    fit_ar = sampling(m_ar,
                      data = list(N = dim(sample_final_)[1],
                                  N_groups_age = age_grouping_multiplier,
                                  N_groups_income = 4 * income_multiplier,
                                  age = sample_final_$age_cat,
                                  income = sample_final_$income,
                                  y = sample_final_$pref),
                      iter=iterations, chains=num_chains,
                      control=list(max_treedepth=15, adapt_delta=0.99),
                      seed = 21,
                      chain_id = num_chains*3*(k-1) + 5 +
                        num_chains*3*(r_numericindex[p_counter] - 1)*(runs - 1) +
                        num_chains*3*(r_numericindex[p_counter] - 1))
```

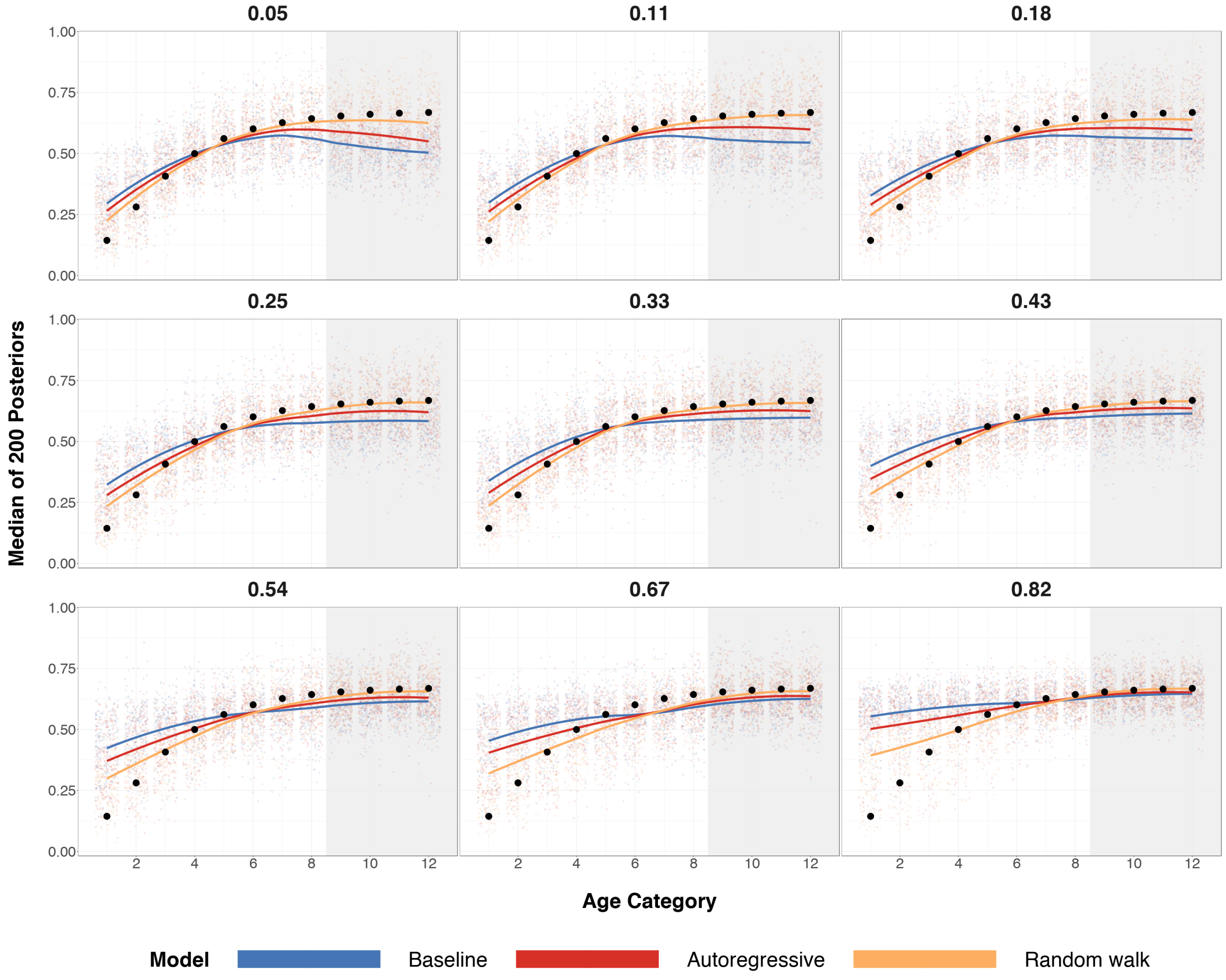
4.3.4 Fitting the random walk prior specification

The random walk specification is fit as `fit_rw`:

```
    # posterior sampling of random walk specification
    fit_rw = sampling(m_rw,
                      data = list(N = dim(sample_final_)[1],
                                  N_groups_age = age_grouping_multiplier,
                                  N_groups_income = 4 * income_multiplier,
                                  age = sample_final_$age_cat,
                                  income = sample_final_$income,
                                  y = sample_final_$pref),
                      iter=iterations, chains=num_chains,
                      control=list(max_treedepth=15, adapt_delta=0.99),
                      seed = 21,
                      chain_id = num_chains*3*(k-1) + 9 +
                        num_chains*3*(r_numericindex[p_counter] - 1)*(runs - 1) +
                        num_chains*3*(r_numericindex[p_counter] - 1))
```

4.4 Summarizing the simulations

After running poststrat_pipeline_testing_age3_v2_markdown.R, running threemodelwriteup_v2_posteriorvariance_markdown.R with the same configurations will produce plots that summarize the simulations. The below figure named allmedians_facet_200_12_100_.png is produced after running it.



From allmedians_facet_200_12_100_.png, one can see that the two structured prior specifications reduce posterior MRP bias regardless of non-response pattern if there is an underlying pattern.

5 Conclusion

The simulation studies shown in this markdown file is a small portion of the numerous simulations studies we conducted for our paper. We tested the sensitivity of MRP posterior estimates through perturbations of survey sample size, number of age categories and true preference curves. Our simulation studies had shown structured priors, namely the autoregressive and random walk specifications, to outperform the baseline specification in MRP through absolute bias reduction in poststratification cell estimates. We show that structured priors weather even extreme nonresponse patterns when compared to traditional random effects used in MRP. This is as expected since structured priors enable intelligent information-borrowing and shrinkage in posterior MRP estimates.

Furthermore, we applied our method of structured priors in MRP to a non-representative data set in real life. This real data analysis had showed that more intelligent shrinkage through structured priors has smoothing effects in posterior estimates, similar to nonparametric regression methods such as GP regression.