Bayesian Prediction and Post-Stratification

Roberto Cerina

June 2019

1 Overview

In this workshop we are going to be reviewing Prediction and Post-Stratification (PPS), a general technique which can allow us to make population-level inference from non-representative samples. Throughout we will describe how the technique can be applied to public opinion research, and in particular the monitoring of voting intentions. That said, the technique is general enough that the examples in this workshop can easily be re-purposed for multiple applications in social science, epidemiology etc. Figure 1 provides a stylized overview of the main steps involved in the procedure.

Sampling $\longrightarrow$ Prediction $\longrightarrow$ Stratification

Figure 1: Stylized summary of Prediction and Post-Stratification Strategy.

We begin this workshop with an introduction to Bayesian methods, as these will be the backbone of the modeling techniques in this workshop. We then move on to Multilevel Regression and Post Stratification (MRP), describe the technique and follow an example based on real-life, non-representative data obtained via the Amazon Mechanical Turk (AMT) platform. After the exposition of these concepts is concluded, we will provide you with the AMT data and ask you to apply these methods for a workshop task.

2 Bayesian Modeling Primer

We prefer working under the Bayesian paradigm when fitting parametric models. This is because of the benefits with respect to natural propagation of uncertainty, ease of parameter interpretation and modeling flexibility. In this section we provide the reader with a set of essential introductory concepts in Bayesian statistics; this review is by no means exhaustive. For more details consult Gelman[9], Kruschke[17], Baio[2] and others.

2.1 Bayes Theorem

Bayes theorem is a formula which allows us to update our beliefs over the uncertainty about an event due to the acquisition of novel information. Consider parameter θ, the National vote share of the Republican candidate for the 2020 election. Based on our knowledge of the current political climate, campaign dynamics, and historical performance, we have a prior belief over the distribution of that parameter $\mathrm{P}(\theta)$. After observing voting intentions y_i from $i = \{1, ..., 1000\}$ respondents to a nationally representative opinion poll, with distribution $\mathrm{P}(y_i \mid \theta)$, we would like to update our beliefs about θ. We do so using the following formula:

$$\mathrm{P}(\theta|y) = \frac{\mathrm{P}(y|\theta) \times \mathrm{P}(\theta)}{\mathrm{P}(y)}; \tag{1}$$

where $\mathrm{P}(\theta|y)$ is referred to as the *posterior* distribution, which is proportional to the product of the likelihood of having observed the data we did $\mathrm{P}(y|\theta)$ and the prior we held before observing the data $\mathrm{P}(\theta)$. Note that the likelihood $\mathrm{P}(y|\theta)$ is conditional on the population parameter θ.

2.2 Data Generating Process and Predictive Distributions

In the Bayesian paradigm, θ exists in a state of uncertainty, and drives the data generating process of voting intentions; this means voting intentions are stochastically dependent from θ, and the more intentions we observe, the more we learn about the parameter that is generating them. This relationship is made clear using Directed Acyclic Graphs (DAGs), as shown in Figure 2. The DAG can be examined from two angles: following the solid lines, we see stochastic dependency, i.e. the direction and flow of the data generating process; following the dotted lines, we see information flow that leads to the Bayesian updating of our priors. The new information carried by the dotted line is propagated back to θ to obtain $\mathrm{P}(\theta|y)$, and then used to update the predictive distribution of y - namely $\mathrm{P}(y^*|y)$.

2.3 Conjugate Models

There exist convenient combinations of priors and likelihoods such that posteriors are of the same form as the priors, which allows us to exploit analytical solutions to formulate a posterior. More

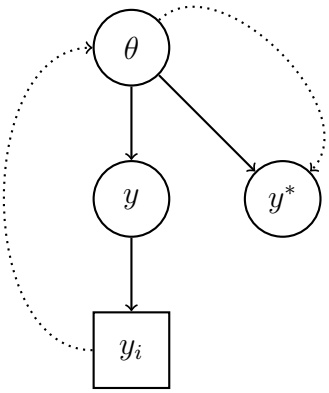

Figure 2: A graphical representation of the data generating process, Bayesian updating and the generation of predictive distributions under the Bayesian paradigm. Circular nodes represent stochastic quantities; rectangular nodes represent observed realizations. The figure is loosely reproduced from Baio[2].

formally, a prior distribution $\mathrm{P}(\theta|\phi) = f(\phi)$, where ϕ stands for the hyper-parameters of that distribution, is said to be *conjugate* for observed likelihood $\mathrm{P}(y|\theta)$ if its posterior is of the same function form, $\mathrm{P}(\theta|\phi) = f(\phi^*)$. In these cases, all is required to obtain the posterior is to update the hyper-parameters from ϕ to ϕ^*.

Looking back at our polling example, a most useful conjugate model is the beta-binomial. We could describe our prior belief over parameter θ, the national vote share of the Republican party, using a Beta distribution of the form: $\theta \sim \mathrm{Beta}(\alpha, \beta)$; this is a reasonable choice, as the Beta distribution ranges from zero to one, and the hyper-parameters α and β can be set to model a vast array of distributional shapes. Our individual voting intentions y_i can reasonably be thought to have been generated by a Bernoulli distribution, such that $y_i|\theta \sim \mathrm{Bernoulli}(\theta)$, where θ can be interpreted as the national-level probability of voting to the Republican party (hence equivalent to the vote share). Then, the posterior can be simply calculated to be $\theta|y \sim \mathrm{Beta}(\alpha + \sum_i y_i, \beta - \sum_i y_i)$. See the MIT Open Course Handout[21] on conjugacy, for more details over these computations.

2.4 MCMC Methods

Conjugate models are however a limited modeling tool in practice - in most circumstances, we may want to specify prior information that goes beyond the simple functional forms available in the conjugate family; many useful statistical models, such as logistic regression, do not have conjugate priors. To overcome these issues me use Monte Carlo Markov Chain (MCMC) methods[9]; these are a set of methods to sample from the posterior of generic probability distributions; the *Markov chain*[2] is at the heart of the method. We define a Markov chain as a sequence of random variables

$X_0, \theta_1, \theta_2, ...$ for which the distribution of its future values depends only on its current state, after conditioning for past and present values. More concisely, a sequence of random variables is a Markov chain if:

$$\mathrm{P}(\theta_{t+1} \mid \theta_0, \theta_1, ..., \theta_t) = \mathrm{P}(\theta_{t+1} \mid \theta_t). \tag{2}$$

In most practical cases (see Jackman[14] for a detailed analysis) we can initialize a Markov chain for some posterior probability distribution with some reasonable starting values, and after an a number of *burn-in* iterations, this chain will become independent of its initial state and its past values, and stabilize around the posterior probability distribution of interest. The most widely used iterative procedure available to researchers is the Gibbs sampler[4].

2.4.1 Gibbs Sampling

Consider a set of hyper-parameters $\theta = (\theta_1, \theta_2, ..., \theta_k)$ from a probability distribution function; for simplicity in exposition, we use a Gaussian distribution as example, where we have $\theta = (\mu, \tau)$. We observe data y, and we seek to learn from it and update our priors on μ and σ. We will now follow the steps of the Gibbs sampler to obtain a stable joint posterior distribution for these parameters.

1) *Initial values*: We assign starting values μ_0 and σ_0; these are generally arbitrary, though some consideration is taken to ensure they are not contradictory with the support of the prior distribution, and in line with the expected posterior (i.e. if you are setting a starting value for a probability parameter, a good starting point is the mean of your prior distribution; you should set a starting value bigger than one or smaller than zero).

2) *Iterative conditional sampling*: sample (as in picking balls at random from an urn):

i. a new value of μ, namely μ^1, from the conditional distribution $\mathrm{P}(\mu \mid \sigma^0, y)$;

ii. sample σ^1 from $\mathrm{P}(\sigma \mid \mu^1, y)$, the new conditional distribution informed by the new value of μ;

iii. sample μ^2 from $\mathrm{P}(\mu \mid \sigma^1, y)$;

iv. ...

3) continue iterating until the above chain converges to the joint posterior distribution $\mathrm{P}(\mu, \sigma \mid y)$, at which point further sampling will be akin to simulating realizations from the joint distribution.

Intuitively, you can see the Gibbs sampler works because there are only so many values for the parameters of interest that support the observed data y.

2.4.2 Convergence and Auto-Correlation

As we stated previously, it will take a few iterations for each parameter to reach convergence and forget its initial values; these iterations are usually called *burn-in*, and are discarded before making inference. To ensure a given chain has converged, we run multiple chains starting from different values, and monitor the mixing of the two chains. These concepts are illustrated in Figure 3 from Baio[2]; the chains are said to have converged if they are *mixing*. It may be useful to have a

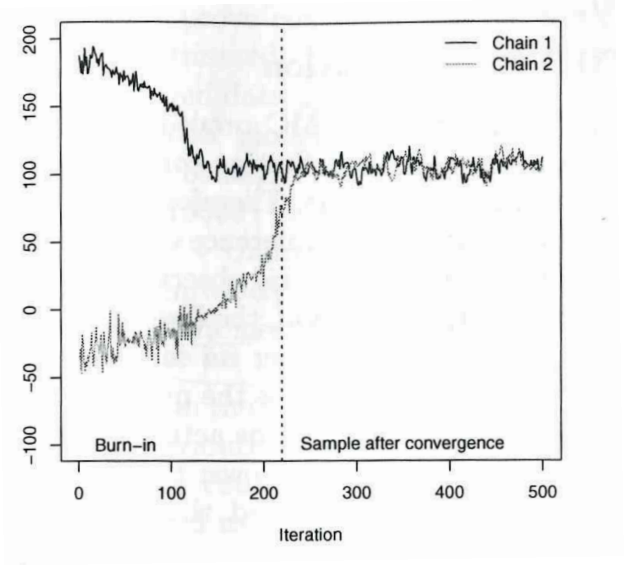

Figure 3: Graphical illustration of the concepts of convergence, burn-in and chain-mixing.

hard rule as per when to stop sampling; there are graphical methods one can use, which involve inspecting the chains and assessing they are nicely mixed; we will see how to do this in the following example. However, we may want a more formal method to assess convergence. This is provided by the Gelman-Rubin statistic[9]; if we were to calculate said statistic for the mean of the above normal distribution, we would do so as follows:

1) Estimate the posterior variance of the parameter of interest:

$$\widehat{\text{Var}}(\mu \mid y) = \frac{S-1}{S} W(\mu) + \frac{1}{s} B(\mu), \tag{3}$$

where $W(\mu)$ stands for the within-chain variance; $B(\mu)$ stands for the between-chains variance, and S is the size of the MCMC sample you are taking from your posterior distribution;

2) Calculate the Gelman-Rubin statistic to assess convergence:

$$\hat{R}(\mu) = \sqrt{\frac{\widehat{\text{Var}}(\mu \mid y)}{W(\theta_k)}}, \tag{4}$$

which is essentially a comparison of the within and between chain variance components. The accepted rule of thumb is that if $\hat{R} \leq 1.1$, we can assume the chain has converged.

Beyond forgetting the initial values, another issue with the chains could be auto-correlation; namely the tendency for a given sample realization to be more similar to the sample realizations that preceded it, than to any other realization. It is clear that this behaviour will be common in our chains, as any realisation is directly dependent on its current value; it is also clear that this would contradict our claim that we can take independent samples from the posterior, after convergence is reached. We can assess the impact of this by calculating the *effective sample size* (or simulation size) of our posterior simulations:

$$n_{\text{eff}} = \frac{S}{1 + 2\sum_{t}^{\infty} \rho_t} \tag{5}$$

where ρ_t is the correlation between iterations that are t time steps apart[2]; if there is no auto correlation, the effective number of simulations is equal to the actual number of simulations S. To tackle this issue we employ a technique called *thinning*. If we want S simulations from the posterior of a single chain, then we should run $f \times S$ simulations from the posterior, and discard every f^{th} value; we call f the thinning factor.

2.5 Learning Objectives

Given what we have covered thus far you should:

i. be familiar with Bayes theorem, the concepts of priors, likelihoods and posteriors;

ii. understand the Bayesian inferential procedure;

iii. able to use conjugate priors, and their limitations;

iv. know what a Gibbs sampler is, and why it works as a tool to obtain the joint posterior distribution of otherwise intractable posteriors;

v. be able to evaluate whether a chain has converged

3 Prediction

In this section we review the most common method for creating category predictions in the literature, namely multilevel regression, as well as how one can move from this to more general non-parametric methods which allow for wider model selection, hence enabling the researcher to make fewer assumptions when developing their modeling strategy. When multilevel regression is the smoothing algorithm of choice, prediction and post-stratification becomes the more familiar multilevel regression and post-stratification (MRP)[22, 19, 24, 18]. We note that researchers can use any prediction method at their disposal; the choice often falls to multilevel regression due to its shrinkage effect (which makes for better out-of-sample predictions - say when we estimate the vote choice distribution of categories we have not observed in the sample) and the ability to include hierarchical predictors, hence helping providing stable estimates of voting distributions for voter-categories of interest; in principle any method that provides some degree of regularization of the predictors can produce good results[1, 12], with more complex non-parametric methods providing greater advantages in highly heterogeneous contexts where there is reason to believe deep interactions[10] between the voter characteristics exist, and the voting population is heterogeneous enough across areas that these deep interactions would change area estimates.

The talk develops in the context of applying the method to vote-intention surveys, in particular with the aim of predicting the winner of a given election. We start with describing typical Bayesian Hierarchical modeling paradigm[7] under which we fit our models. Given the decomposition described below, we start with a model of turnout. Whilst we know responses to the typical turnout question on surveys to be biased, with this upward bias quantified at around 13.5 points for large representative surveys such as the American National Election Study[15], and 17 points for online surveys[13], recent work[16] has shown that modeling the survey questions can produce more accurate results than other kinds of turnout adjustments.

3.1 Prediction and Post-Stratification Problem Decomposition

It is of interest to estimate, for each voter-category of interest, the vote-choice distribution of individuals who will turn-out for the election. Recent applications of MRP[18] for the purpose of model-based opinion polling have taken to decomposing the joint distribution of vote choice and turnout as follows:

$$\mathrm{P}\left(V = j, T = 1 | X\right) = \mathrm{P}\left(V = j | T = 1, X\right) \times \mathrm{P}\left(T = 1 | X\right); \tag{6}$$

where X represents the set of covariates that define a given voter-category. This enables us to provide separate estimates for the turnout distribution, and vote-choice conditional on turnout, for each voter-category of interest. We could in principle estimate the joint distribution in a single step, say by considering turnout as part of the vote-choice survey question (with 'Would Stay Home' as

a vote choice option); we decide against this because this formulation allows us to more intuitively bring in information from from separate sources. For instance, we may want to use an election study to estimate turnout propensities.

3.2 Multilevel Regression

There are many ways in which people have taken to defining random effects[1]. In the Bayesian paradigm, since all parameters have prior distributions, all parameters are random in the strict sense. For the purpose of this exposition, we find it useful to define random effects as those that display some degree of shrinkage i.e. those whose effect is pulled towards the average of the groups. Hence an effect is random if its point estimate is the weighted average of within-group and between-group means, where the weights are dependent on the within-group and between-group variances (the more precise mean having higher weights); effects that display no shrinkage, are called fixed; see Gelman and Hill[7] for more a more formal definition. A typical Bayesian Hierarchical model to predict voter-category turnout can be specified as follows.

$$T_i \sim \text{Bernoulli}(\pi^T_{g[i]}); \tag{7}$$

$$\text{logit}(\pi^T_{g[i]}) = \eta^0 + \eta^{sex}\text{Female}_i + \eta^{age}_a + \eta^{race}_r + \eta^{edu}_e + \eta^{hinc}_h + \eta^{state}_s + \sum_d \beta_d x_{id}; \tag{8}$$

$$\eta^k_{j_k} \sim \text{N}(0, \tau^k_\eta); \tag{9}$$

$$\tau^k_\eta = \frac{1}{{\sigma^k_\eta}^2}; \tag{10}$$

$$\sigma^k_\eta \sim \text{Unif}(0, 5); \tag{11}$$

$$\beta_d \sim \text{N}(0, 0.001); \tag{12}$$

where k is the index of all individual level voter characteristics introduced in the model; j_k represents the j^{th} level of variable k; the prior distribution on σ^k is non-informative on the logit scale[6]; the *gender* variable has been reduced to a dichotomous Male/Female indicator, given that there are no gains from pooling if the number of groups in the multilevel structure is below 3; $x_{s[i]d}$ represents the d^{th} element of the area-level linear predictor that includes variables such as past area turnout, past area Republican share, etc.; the area predictor is estimated as a set of fixed effects. This model can easily be estimated via Monte Carlo Markov Chain methods[9], of which the Gibbs sampler is the fundamental backbone, implemented via `R` packages such as `JAGS`[23] or `STAN`[8][2]. In this workshop we privilege the use of `JAGS` out of simplicity, but readers should learn both languages.

An advantage of the Bayesian graphical modeling offered by `JAGS` is that we can take certain liberties in our model specification, such as linking multiple models together in non-linear ways, and

[1] See Gelman[5] for a complete discussion of this problem

[2] `STAN` makes use of another MCMC method, namely Hamiltonian Monte Carlo, which can provide benefits in speed for certain kinds of problems.

incorporating uncertainty in a hierarchical way. In practical terms, we can imagine there will be individuals in the sample which will have high levels of uncertainty around their predicted probability of turnout, and hence on any given realization from their turnout distribution they may be excluded from the sample of voters who will turn out, for whom we want to estimate vote-choice probabilities. We can model this behaviour in `JAGS` as follows:

$$T^*_{g[i]} \sim \text{Bernoulli}(\pi^T_g); \tag{13}$$

$$R_i \mid T^*_{g[i]} = 1 \sim \text{Bernoulli}(\pi^R_{g[i]}) \tag{14}$$

$$\text{logit}(\pi^R_{g[i]}) = \alpha^0 + \alpha^{sex}\text{Female}_i + \alpha^{age}_a + \alpha^{race}_r + \alpha^{hinc}_h + \alpha^{edu}_e + \alpha^{state}_s + \sum_d \gamma_d x_{id}; \tag{15}$$

$$\alpha^k_{j_k} \sim \text{N}(0, \tau^k_\alpha); \tag{16}$$

$$\tau^k_\alpha = \frac{1}{{\sigma^k_\alpha}^2}; \tag{17}$$

$$\sigma^k_\alpha \sim \text{Unif}(0, 5); \tag{18}$$

$$\gamma_d \sim \text{N}(0, 0.001); \tag{19}$$

where $T^*_g[i]$ is a novel indicator from the predictive distribution of T for each individual in the sample, generated by updated parameter π^T_g. T^*_i may take different values in each simulation, allowing us to incorporate turnout uncertainty in the vote choice model.

3.2.1 Example: Replicating 2016 State-Level Results with Amazon Mechanical Turks - Fitting the Model

We want generate a survey for predicting who is going to win the 2020 election amongst plausible democratic candidates and Donald Trump. In this survey we also ask retrospective questions about behaviour in the 2016 race. The goal of this example will be to replicate the 2016 results at the state and national level, from this unrepresentative survey of Americans.

We create the survey and post it on the Amazon Mechanical Turks (AMT) platform to collect responses from up to $1,500$ workers on the 11^{th} of June 2019. Respondents are paid \$0.5 per response; on top of that, AMT takes roughly 25% of the total fee, which includes the price for some pre-stratification (such as limiting respondents to members from the US). The sample took just over 3 hours to be collected, and the total cost for this survey was $\$1,050$. Figure 4 shows the summary of the Batch (this refers to the batch of Human Intelligence Task (HITs), which is the language AMT uses to describe the total number of surveys we have requested to be completed, with each survey being a single HIT).

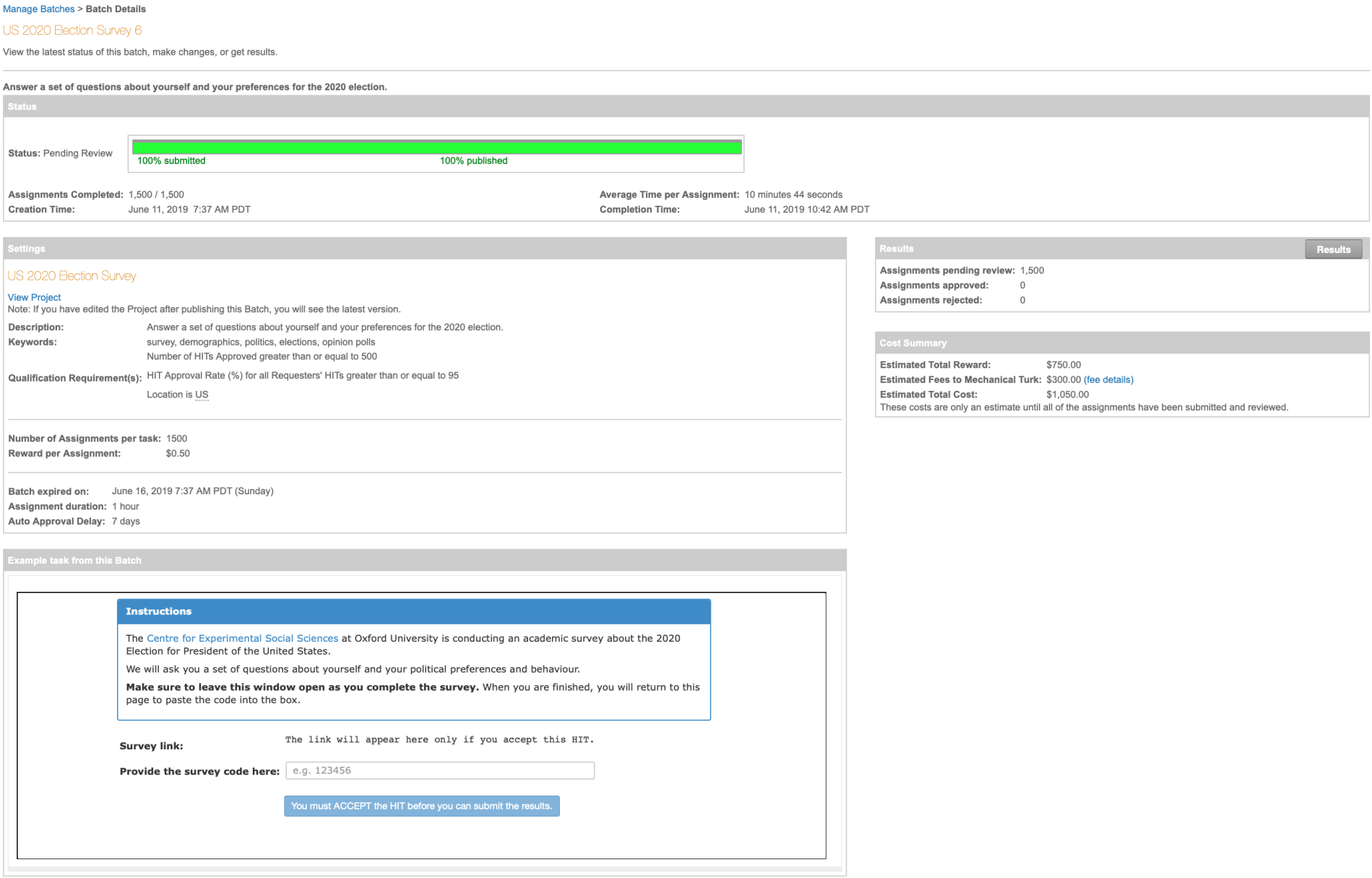

Figure 4: Summary of data collection logistics on the AMT platform.

The Mechanical Turks are presented with the following turnout question: *Did you vote for the Presidential election in 2016 ?* with potential answers {*Yes*, *No*, *Can't Remember/Don't Know*,*Was Not Eligible*}. We further ask: *Which candidate did you vote for President in 2016?* with potential answers {*Donald Trump*, *Hillary Clinton*, *Third Party*, *Can't Remember/Don't Know*, *Did Not Vote*}. Individuals who have a conflict in their turnout behaviour in the two questions above are assumed not to have turned out. Finally, we ask questions regarding voter characteristics, such as Gender, Age, Race, Education, Income and Location.

We want to learn from the survey who are the voter-categroies most likely to turn-out on election day, and who they will vote for. We specify a Bayesian hierarchical model similar what we have seen in this section, making amendments for allowing higher levels of complexity. Specifically, we introduce interactions effects for race and education and allow for multi-party (instead of two-party) vote choice.

Variables that are part of the state-level predictor include: `pct_hispanic`; `pct_black`; `pct_asian`; `pct_non_college_whites`; `pct_college_grads`; `median_income`; `R_pres_2016_pct`; `L_pres_2016_pct`; `G_pres_2016_pct`; `VAP_T_pres_2016_pct`. Note that we are using 2016 state level results as part of the prediction; this may sound circular but remember we are trying to replicate these results from the self-reported individual level voting preferences from the AMT sample, so there is not direct cycle here. We should also note that Lax and Phillips[19] point to the fact that the linear predictor should not include many variables. This is because in multilevel regression, introducing linear predictors that may be very noisy could lead to decreasing predictive accuracy. That said, though our predictor is substantial, we deplot Gibbs regularization to filter out noisy correlates.

In its hierarchical form, the turnout model can be written as follows:

$$T_i \sim \text{Bernoulli}(\pi^T_{g[i]}); \tag{20}$$

$$\text{logit}(\pi^T_{g[i]}) = \eta^0 + \eta^{sex}\text{Fem}_i + \eta^{age}_a + \eta^{race}_r + \eta^{edu}_e + \eta^{hinc}_h + \eta^{state}_s + \eta^{reg}_l + \sum_d \beta_d x_{id}; \tag{21}$$

$$\eta^k_{j_k} \sim \text{N}(0, \tau^k_\eta); \tag{22}$$

$$\tau^k_\eta \sim \text{Gamma}(0.001, 0.001); \tag{23}$$

$$\beta_d \sim \text{N}(0, 0.001); \tag{24}$$

$$\tag{25}$$

we note the change in prior on the variance: we use the common Gamma conjugate prior with small parameters to approximate a non-informative prior, though we recognized this may in fact bias the variance parameter; nevertheless, the computational speed gains from this construct are large enough, and the bias to be expected small enough in our context, that the hazard is justified.

The vote choice model is similar in nature, with the main difference being the use of a Cate-

gorical distribution instead of Bernoulli, to allow for multiple-choices beyond the binary Republican/Democrat, and the introduction of party-specific effects constrained to sum to zero for identification purposes; the hierarchical specification follows:

$$T^*_{g[i]} \sim \text{Bernoulli}(\pi^T_g - 0.17); \tag{26}$$

$$V_i \mid T^*_{g[i]} = 1 \sim \text{Categorical}(\pi^V_{g[i]1}, \pi^V_{g[i]2}, \pi^V_{g[i]3}) \tag{27}$$

$$\text{logit}(\pi^V_{g[i]}) = \alpha^{*0}_v + \alpha^{*sex}_v \text{Fem}_i + \alpha^{*age}_{va} + \alpha^{*race}_{vr} + \alpha^{*edu}_{ve} + \alpha^{*hinc}_{vh} + \alpha^{*state}_{vs} + \alpha^{*reg}_{vl} + \sum_d \beta_{vd} x_{id}; \tag{28}$$

$$\alpha^{*\ k}_{vj_k} = \alpha_{vj_k} - \bar{\alpha}_{j_k}; \tag{29}$$

$$\alpha_{vj_k} \sim \text{N}(0, \tau^k_\alpha); \tag{30}$$

$$\tau^k_\alpha \sim \text{Gamma}(0.001, 0.001); \tag{31}$$

$$\beta_{vd} \sim \text{N}(0, 0.01); \tag{32}$$

$$\tag{33}$$

where the mean-differencing which can be seen on equation-line (42) serves as a sum-to-zero constraint over the parties. This constraint is introduced to help convergence and identifiability of the parameters by constraining the support of the posterior distribution of the relevant α, but also because it makes sense from a modeling stand-point: it must be the case that (in a two-party toy example) if males are strongly in favour of republicans, they must be against the democrats. Notice also the -0.17 offset in the predictive distribution of turnout on equation-line (36); this is done to attempt a correction for over-reporting, assuming said over-reporting is uniform across categories of interest; the figure comes from the Pew[13] study cited previously. The code to fit this complex model follows.

```
# PART 4 - i) - REPLICATE 2016 RESULTS  # # # # # # # # # # # # # # # # # # # # # # #

# Now load data for JAGS
model_data = list(
  N = dim(AMT_counts_complete)[1],
  #V = as.integer(AMT_counts_complete$TRUMP_CLINTON_choice),
  V =
    cbind(ifelse(as.integer(AMT_counts_complete$TRUMP_CLINTON_choice)==1,1,0),
          ifelse(as.integer(AMT_counts_complete$TRUMP_CLINTON_choice)==2,1,0),
          ifelse(as.integer(AMT_counts_complete$TRUMP_CLINTON_choice)==3,1,0)),

  N_V = max(as.integer(AMT_counts_complete$TRUMP_CLINTON_choice),na.rm=TRUE),
  T = as.numeric(as.character(unlist(AMT_counts_complete$TRUMP_CLINTON_turnout))),
```

```
  ST_ID = as.numeric(gsub("\\)","",substr(AMT_counts_complete$state,2,3))),
  N_ST = max(as.numeric(gsub("\\)","",substr(AMT_counts_complete$state,2,3)))),
  RG_ID = as.integer(AMT_counts_complete$region),
  N_RG = max(as.integer(AMT_counts_complete$region)),
  FEM = as.numeric(substr(AMT_counts_complete$sex,2,2))-1,#female id
  AG_ID = as.integer(AMT_counts_complete$age),
  N_AG = max(as.integer(AMT_counts_complete$age)),
  RC_ID = as.integer(AMT_counts_complete$race),
  N_RC = max(as.integer(AMT_counts_complete$race)),
  ED_ID = as.integer(AMT_counts_complete$education),
  N_ED = max(as.integer(AMT_counts_complete$education)),
  IC_ID = as.integer(AMT_counts_complete$income),
  N_IC = max(as.integer(AMT_counts_complete$income)),
  # Need to bring in state-level predictor
  X = AMT_counts_complete[,grep("state",names(AMT_counts_complete))][,-1],
  N_X = dim(AMT_counts_complete[,grep("state",names(AMT_counts_complete))][,-1])[2])
)

# Load packages
library(R2jags)

model_code = '
model {
# Likelihood - Turnout
for(i in 1:N) {
# predictive distribution of memebers of the sample (useful to weight vote choice model later)
   T_star[i] ~ dbern(pi_T_star[i])
pi_T_star[i] <- ifelse( (pi_T[i] - 0.17)<=0,0,(pi_T[i] - 0.17) ) # assume 17% over-reporting across all categ

# estimate T prediction model
    T[i] ~ dbern(pi_T[i])
 pi_T[i] <- exp(eta_T[i])/(1+exp(eta_T[i]))
eta_T[i] <- alpha_T_0 +
# individual level random effects
            alpha_T_1[ST_ID[i]] + #state effect
            alpha_T_2[RG_ID[i]] + #region effect
            alpha_T_3[AG_ID[i]] + #age effect
            alpha_T_4[RC_ID[i]] + #race effect
            alpha_T_5[ED_ID[i]] + #education effect
```

```
            alpha_T_6[IC_ID[i]] + #income effect
            alpha_T_7*FEM[i] + #female effect

# state level predictor
            inprod(beta_T[1:N_X],X[i,])
}

# Priors - Turnout
# fixed effects
alpha_T_0 ~ dnorm(0,0.01)
alpha_T_7 ~ dnorm(0,0.01)

for(x in 1:N_X){
beta_T_star[x] ~ dnorm(0,tau_beta_T)
     beta_T[x] <- beta_T_star[x] * aux_beta_T
}
tau_beta_T ~ dgamma(0.0001,0.0001)
aux_beta_T ~ dnorm(0,0.01)

# random effects
for(st in 1:N_ST){
alpha_T_1[st] <- alpha_T_1_over[st] * aux_T[1]
alpha_T_1_over[st] ~ dnorm(0,(tau_T[1]))
}
for(rg in 1:N_RG){
alpha_T_2[rg] <- alpha_T_2_over[rg] * aux_T[2]
alpha_T_2_over[rg] ~ dnorm(0,(tau_T[2]))
}
for(ag in 1:N_AG){
alpha_T_3[ag] <- alpha_T_3_over[ag] * aux_T[3]
alpha_T_3_over[ag] ~ dnorm(0,(tau_T[3]))
}
for(rc in 1:N_RC){
alpha_T_4[rc] <- alpha_T_4_over[rc] * aux_T[4]
alpha_T_4_over[rc] ~ dnorm(0,(tau_T[4]))
}
for(ed in 1:N_ED){
alpha_T_5[ed] <- alpha_T_5_over[ed] * aux_T[5]
alpha_T_5_over[ed] ~ dnorm(0,(tau_T[5]))
```

```
}
for(ic in 1:N_IC){
alpha_T_6[ic] <- alpha_T_6_over[ic] * aux_T[6]
alpha_T_6_over[ic] ~ dnorm(0,(tau_T[6]))
}

for(i in 1:6){
tau_T[i] ~ dgamma(0.0001,0.0001)
aux_T[i] ~ dnorm(0,0.01)
}



# Likelihood - Vote Choice
for(i in 1:N) {
for(j in 1:N_V){
V[i,j] ~ dpois(nu_V[i,j])
nu_V[i,j] <- exp(eta_V[i,j])

eta_V[i,j] <- lambda[i] +
              alpha_V_0[j,T_star[i]+1] +
              # individual level random effects
              alpha_V_1[ST_ID[i],j,T_star[i]+1] + #state effect
              alpha_V_2[RG_ID[i],j,T_star[i]+1] + #region effect
              alpha_V_3[AG_ID[i],j,T_star[i]+1] + #age effect
              alpha_V_4[RC_ID[i],j,T_star[i]+1] + #race effect
              alpha_V_5[ED_ID[i],j,T_star[i]+1] + #education effect
              alpha_V_6[IC_ID[i],j,T_star[i]+1] + #income effect
              alpha_V_7[j,T_star[i]+1]*FEM[i] + #female effect

# state level predictor
              inprod(beta_V[1:N_X,j,T_star[i]+1],X[i,])
} }

# Priors - Vote Choice
for(i in 1:N){
lambda[i] ~ dgamma(0.0001,0.0001)
}

for(t in 1:2){ for(j in 1:N_V){
```

```
for(x in 1:N_X){
beta_V[x,j,t] <- (beta_V_over[x,j,t]-mean(beta_V_over[x,,t])) * aux_V[9]
beta_V_over[x,j,t] ~ dnorm(0,tau_beta_V[j])
}
} }

for(j in 1:N_V){
  tau_beta_V[j] ~ dgamma(0.0001,0.0001)
}

for(j in 1:N_V){ for(t in 1:2){
     alpha_V_0[j,t] <- alpha_V_0_over[j,t] * aux_V[7]
alpha_V_0_over[j,t] ~ dnorm(0,(tau_V[7,j]))

     alpha_V_7[j,t] <-  (alpha_V_7_over[j,t] -mean(alpha_V_7_over[,t] ))* aux_V[8]
alpha_V_7_over[j,t] ~ dnorm(0,(tau_V[8,j]))

# random effects
for(st in 1:N_ST){
     alpha_V_1[st,j,t] <- (alpha_V_1_over[st,j,t]-mean(alpha_V_1_over[st,,t])) * aux_V[1]
alpha_V_1_over[st,j,t] ~ dnorm(0,(tau_V[1,j]))
}

for(rg in 1:N_RG){
     alpha_V_2[rg,j,t] <- (alpha_V_2_over[rg,j,t]-mean(alpha_V_2_over[rg,,t])) * aux_V[2]
alpha_V_2_over[rg,j,t] ~ dnorm(0,(tau_V[2,j]))
}

for(ag in 1:N_AG){
     alpha_V_3[ag,j,t] <- (alpha_V_3_over[ag,j,t] -mean(alpha_V_3_over[ag,,t] ))* aux_V[3]
alpha_V_3_over[ag,j,t] ~ dnorm(0,(tau_V[3,j]))
}

for(rc in 1:N_RC){
     alpha_V_4[rc,j,t] <-  (alpha_V_4_over[rc,j,t]-mean(alpha_V_4_over[rc,,t])) * aux_V[4]
alpha_V_4_over[rc,j,t] ~ dnorm(0,(tau_V[4,j]))
}

for(ed in 1:N_ED){
```

```
alpha_V_5[ed,j,t] <- (alpha_V_5_over[ed,j,t]-mean(alpha_V_5_over[ed,,t])) * aux_V[5]
alpha_V_5_over[ed,j,t] ~ dnorm(0,(tau_V[5,j]))
}

for(ic in 1:N_IC){
alpha_V_6[ic,j,t] <- (alpha_V_6_over[ic,j,t]-mean(alpha_V_6_over[ic,,t])) * aux_V[6]
alpha_V_6_over[ic,j,t] ~ dnorm(0,(tau_V[6,j]))
}
} }

for(i in 1:9){ for(j in 1:N_V){
tau_V[i,j] ~ dgamma(0.0001,0.0001)
} }

for(i in 1:9){
aux_V[i] ~ dnorm(0,0.01)
}

}'

tmpf=tempfile()
tmps=file(tmpf,"w")
cat(model_code,file=tmps)
close(tmps)

# Monitor the following parameters
model_parameters =  c("T_star",
                      "alpha_T_0","alpha_T_1","alpha_T_2","alpha_T_3",
                      "alpha_T_4","alpha_T_5","alpha_T_6","alpha_T_7",
                      "beta_T",
                      "alpha_V_0","alpha_V_1","alpha_V_2","alpha_V_3",
                      "alpha_V_4","alpha_V_5","alpha_V_6","alpha_V_7",
                      "beta_V")

# Run JAGS model
model_run = jags.parallel(data = model_data,
                          parameters.to.save = model_parameters,
                          model.file = tmpf,
                          n.chains = 4,
```

```
        n.iter = 7000,
        n.burnin =6000,
        n.thin = 4,
        n.cluster = 4
)
```

The over-parametrisation to include a multiplicative factor in random-effect priors is designed to speed-up convergence[9]. Notice that the categorical distribution is replaced by a poisson with an individual level random effect, with gamma prior on the log-scale; this is a well known equivalency, explained in detailed in Lee et al.[20]. This model takes around $4hours$ hours to converge.

The first order of business as usual is to check the convergence diagnostics. Note that this model is huge, so it needs more than a few thousand runs to converge. Here we only ran the model for $7,000$ runs with a thinning factor of 4. The Gelman-Rubin statistic values for this model are shown in Figure 5. Most of the parameters are right below the 1.1 line. We can say this predictive model has largely converged; in the machine learning literature, where the Gibbs sampler is deemed too slow to be useful, and parameters are estimated through a search algorithm (of which the most famous is Stochastic Gradient Descent), the lack of convergence (and associated bias introduced in the parameters) is taken to be a prediction cost; this means that a model that does not converge but consistently provides good predictions is preferable to one that converges but provides worst predictions out-of-sample. Of course non-convergent models are more likely to be inconsistent, but we should keep this aspect in mind.

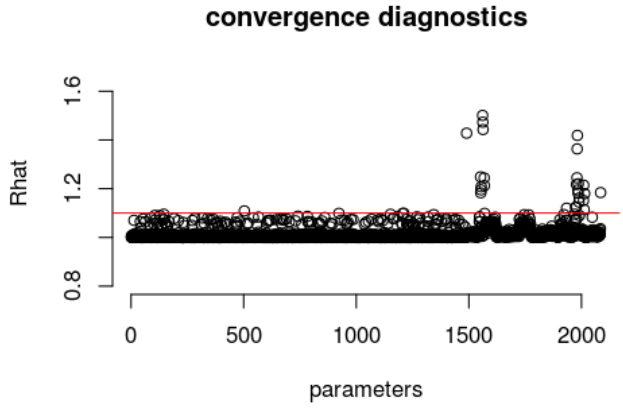

Figure 5: This model has largely converged, with the exception of a small number of parameters who would need another few hundred iterations.

We use this fitted model to produce predictions for voter-categories of interest; these are derived from a stratification-frame, which identifies the number of individuals in a given voter-category (cells). The stratification frame we use is derived from the American Community Survey (ACS) microdata available at https://usa.ipums.org/usa/. We break down the US population into the following voter characteristics: Sex (2 categories); Age (6 categories); Race (5 Categories); Education (4 Categories); Household Income (3

Categories); State (51 categories). This amounts to 36,720 cells, of which 29,905 have at least one individual in the ACS micro-data. Figure 6 shows the probability of sampling a member of a given cell in the population at large, and their cumulative sampling probability.

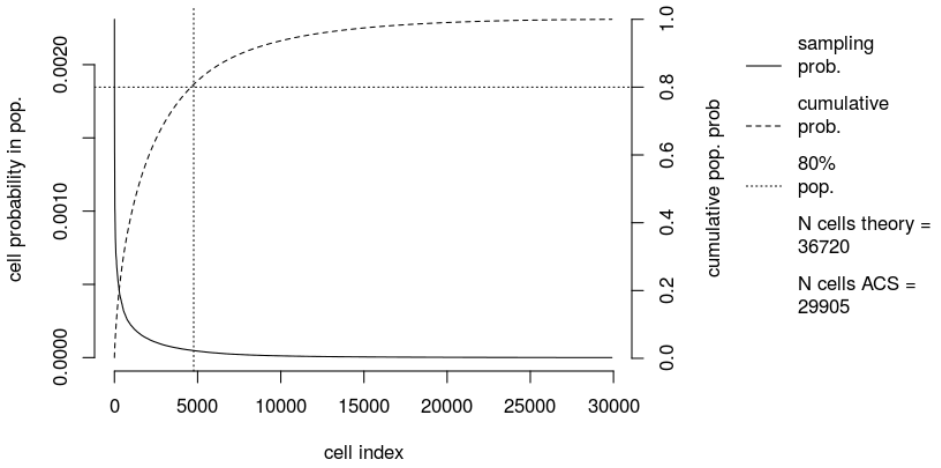

Figure 6: Relative and cumulative size of cells in the stratification frame.

We can see from the plot that the largest 5,000 cells make up roughly 80% of the voting age population of the United States. This has implication for a-priori sampling design; assuming the 'weird' groups (those constituted of very few members) are randomly dispersed across the areas of interest, we can ignore them; consequently, the implication is that it is ok for our sample to be non-representative of the whole population, so long as we have enough members of the largest categories to be able to predict accurately their voiting intentions.

Table 1 shows the sampling frame which we feed to the fitted model, and for which we obtain predictions on turnout and vote-choice behaviour. For each cell of the stratification frame in Table 1 we obtain a simulation matrix for their probability to turn-out on election day, $\mathrm{P}(T \mid X)$, and another, of the same size, for their probability of voting given they turn-out, $\mathrm{P}(V \mid T, X)$. In the next section we will see how these estimates are combined with the cell-counts in the stratification frame to produce area estimates.

3.3 Learning Objectives

Given what we have covered thus far you should:

i. be familiar with Bayesian multilevel regression and Hierarchical modeling;

ii. understand the role that shrinkage plays in making MRP a preferable method for out-of-sample predictions (i.e. for categories of interest that we do not directly observe in the sample);

iii. be able to use JAGS to fit a Bayesian Hierarchical model;

iv. understand what a stratification frame is, and how cells are distributed across it.

Cell Id.	State	Region	Sex	Age	Race	Education	Income	Cell Counts
1	(10) Florida	(3) South Region	(2) Female	(44-54]	(2) Hispanic	(4) College Grad	(1) [-$20,000-$49,999]	241
2	(10) Florida	(3) South Region	(2) Female	(44-54]	(1) White	(4) College Grad	(3) [$100,000-$9,999,999]	1800
3	(10) Florida	(3) South Region	(1) Male	(34-44]	(4) Asian or Pacific Islander	(4) College Grad	(1) [-$20,000-$49,999]	34
4	(10) Florida	(3) South Region	(1) Male	(64-max]	(1) White	(3) Some College	(1) [-$20,000-$49,999]	1466
5	(10) Florida	(3) South Region	(1) Male	(17-24]	(5) Other	(2) High School	(1) [-$20,000-$49,999]	55
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
29900	(9) District of Columbia	(3) South Region	(2) Female	(64-max]	(4) Asian or Pacific Islander	(2) High School	(1) [-$20,000-$49,999]	4
29901	(9) District of Columbia	(3) South Region	(2) Female	(64-max]	(3) Black	(2) High School	(1) [-$20,000-$49,999]	100
29902	(9) District of Columbia	(3) South Region	(1) Male	(24-34]	(3) Black	(1) Below Elementary	(1) [-$20,000-$49,999]	1
29903	(9) District of Columbia	(3) South Region	(1) Male	(54-64]	(2) Hispanic	(3) Some College	(1) [-$20,000-$49,999]	2
29904	(9) District of Columbia	(3) South Region	(2) Female	(34-44]	(2) Hispanic	(1) Below Elementary	(2) [$50,000-$99,999]	2
29905	(9) District of Columbia	(3) South Region	(2) Female	(34-44]	(5) Other	(4) College Grad	(1) [-$20,000-$49,999]	1

Table 1: A few randomly selected rows from the ACS derived stratification frame.

4 Stratification

The problem with the AMT sample is its inherent non-representativeness of the broader US population. Figure 7 shows the sample v. population plots for some key voter characteristics. We can see that, though the sample holds up on location, age, race and sex, it is very un-representative of education, over-sampling college-grads and under-sampling high-school grads; it is further unrepresentative of income, under-sampling high income households; and finally it is extremely poor in terms of age, with the young heavily over-represented, and the old being under-represented. Finally the sample respondents (unsurprisingly, given the other inconsistencies) are more likely to have voted for third-parties in 2016, and more heavily more likely to have voted for Clinton over Trump. The problems outlined in the previous set of Figures are exacerbated at

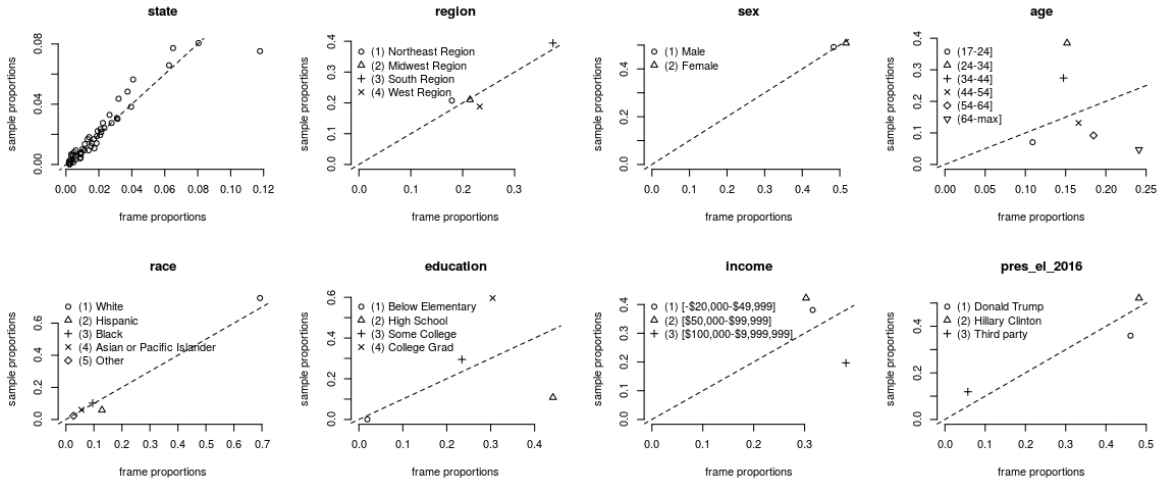

Figure 7: AMT Sample comparison with Population for basic characteristics.

the area-level, where we have no real hope of obtaining reliable estimates of support due to the tiny sample sizes per-area; consider for instance that we have no individuals from DC; only one individual from Wyoming and South Dakota; two for Hawaii, and so forth. To overcome this issue, we stratify the category-predictions

obtained by this model. We do so as follows:

$$V_{js} = \frac{\sum_g \mathrm{P}_g\left(V = j, T = 1 | X, S = s\right) \times Q_g}{\sum_g \mathrm{P}_g\left(T = 1 | X, S = s\right) \times Q_g}; \tag{34}$$

where Q_g represents the number of individuals which are part of group g, and V_{js} is the proportion of individuals voting for choice j in state s. We can also use the formula if we are interested in a higher area-level, namely the national level, as follows:

$$V_j = \frac{\sum_g \mathrm{P}_g\left(V = j, T = 1 | X\right) \times Q_g}{\sum_g \mathrm{P}_g\left(T = 1 | X\right) \times Q_g}; \tag{35}$$

where we effectively just treat state as another individual characteristic in X by which we stratify. The results of this effort in out AMT example can be evaluated in the following figures.

4.0.1 Example: Replicating 2016 State-Level Results with Amazon Mechanical Turks - Results

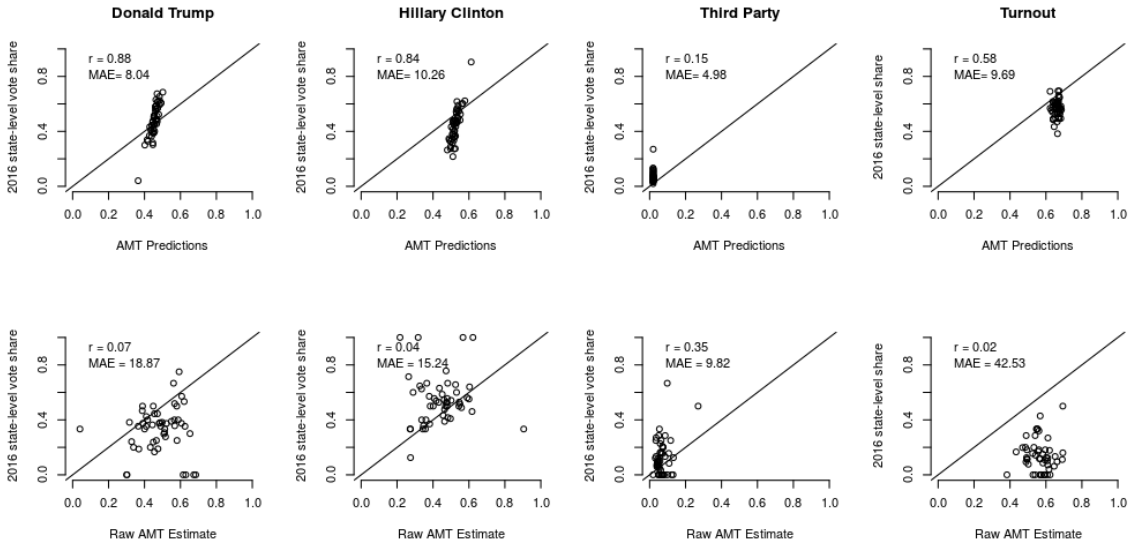

Figure 8: Scatter plots of observed v. predicted values for the quantities of interest in each state.

Figure 8 shows that, on average, the MRP reduced Mean Absolute Error for Trump predictions, when compared to the raw AMT sample estimates, by around 11 points; for Hillary and Third parties by around 5 points; for turnout by an astounding 30 plus points. Note that some of the gains on turnout are also due to the over-reporting offset in the model. Correlations for MRP estimates of vote-choice for Trump and Hillary are extremely high, suggesting the model is able to order the states reasonably well. It is evident that attenuation bias is an issue, clearly seen by the low-variance displayed by MRP point estimates. Remember that given this is a Bayesian estimation procedure, these point estimates come as the average of a number of simulations each.

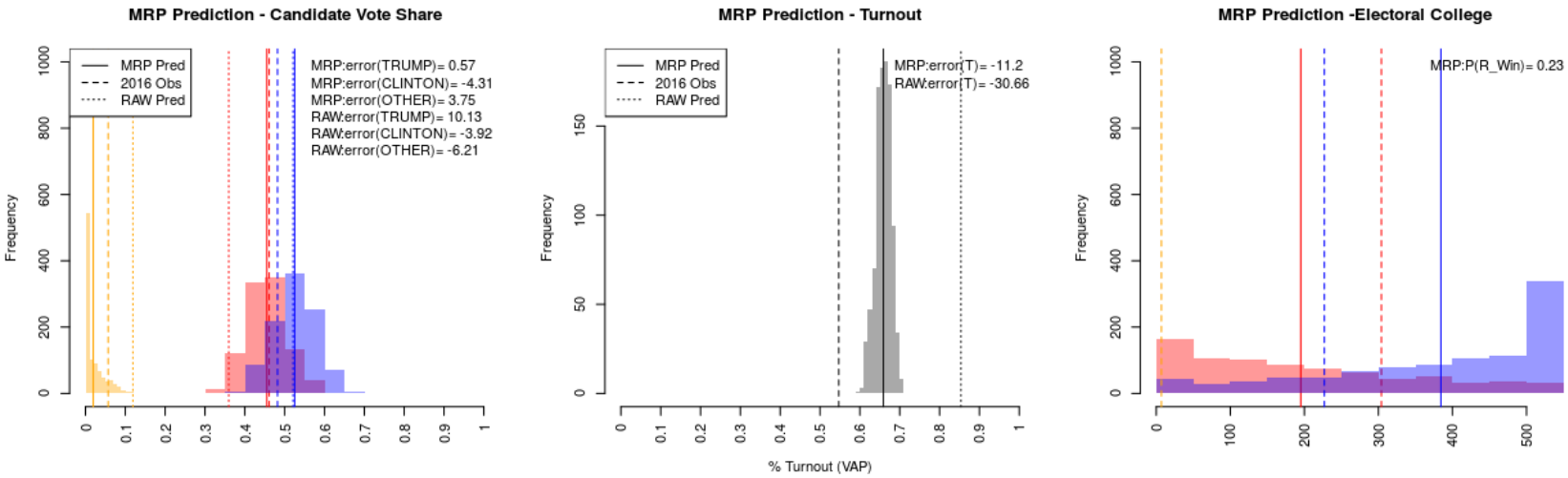

Figure 9: Outcome of the stratification effort at the National Level, including forecast for the electoral college with appropriate uncertainty.

Results for the National Level estimates are available in Figure 9. The MRP estimates are excellent on Trump, miscalculating its true vote-share by around half of a percentage point at the national level; this represents a 9.5 points gain over the raw-sample estimate. The MRP could not however help much with Clinton's national share, which is over-estimates by 4 percentage points in both raw estimates and MRP, though the MRP error is lower by about half a point. The third-party vote is under-estimated by 4 points by MRP, whilst it was over-estimated by about 6 points by the raw sample estimates, representing a gain of around 3 points in absolute terms. 2016 turnout is over-estimated by 11 points in our MRP estimates, a gain of around 20 points over the raw-sample estimate (though again here some of the work is being done by the over-reporting offset in the model). Finally, the MRP estimates would predict that Hillary was more likely to win the electoral college, though the probability assigned to a Trump win is substantial - namely 23/100.

Overall, we can be satisfied with our MRP estimates; they massively improve on raw-sample estimates, and reach satisfactory levels of accuracy at the national level.

We should note this was a replication excercise, and so we should not expect the vote-intention questions to be estimated as accurately, as they do not benefit from the post-hoc inclusion of true state-level vote shares. We further note that the level of accuracy at the state-level is unsatisfactory: a campaign could not make reasonable resource allocation with estimates that are on average 10 points off.

Finally, we note that we could have probably done alot better with a larger sample; $1,500$ individuals are not much for an MRP excercise, especially when trying to estimate multiple vote-choice options in heterogeneous contexts. We know from preliminary findings on low sample sizes that the absolute error of estimates coming from AMT samples tends to be much higher than simple random samples (roughly 7.5 percentage points compared to 2.5 in a random sample, for national-level estimates using $N = 1000$[11]); with this work we have already shown that AMT error can be well-below that, provided the linear predictor is strong enough. Findings on probability samples by Buttice and Highton[3] have shown that, though stratifying increases performance of national probability surveys to estimate state-level quantities under any sample size, performance varies significantly with typical survey sizes of $N \sim 1000$, and increases dramatically as sample size increases. Given these considerations, there is reason to believe higher sample sizes would better the estimates.

5 Workshop

You have been introduced to Multilevel Regression and Post-Stratification. In this workshop, you will get to apply the new techniques you have just learnt. The AMT survey did not stop at asking individuals who they voted for in 2016. We also asked horse-race questions conditional on the democratic candidate. For example: *If the US Presidential election were held tomorrow, the candidates being Donald Trump for the Republicans and Bernie Sanders for the Democrats, who would you vote for?*. These questions were asked for the following Democratic Candidates, in no precise order: Joe Biden; Kamala Harris; Elizabeth Warren; Bernie Sanders; Cory Booker; Pete Buttigieg; Beto O'Rourke and Amy Klobuchar. In the following hour and half, you will pick a candidate of interest amongst these, and replicate the analysis in the code on the github page. You can use the code in its entirety, but do make an effort to understand and execute the different steps.

i. Pick a small number (say four) of individual-level covariates, to construct a smaller stratification frame; plot the sampling probabilities and the cumulative probability of your sample frame;

ii. Further pick one state level predictor variable and two or three vote-choice predictors. Make sure these are ordered in the same way as the sampling frame and the AMT sample;

iii. Fit a turnout model using `JAGS` (or any other method, including frequentist ones, at your disposal; note that if you use frequentist approaches, you still need to produce prediction intervals or other estimates of uncertainty for the state-level post-stratified estimates). Ensure the model converges in the first 1000 to 2000 iterations (try out your initial idea; if it's slow it it crashes, this may mean you need simply the model);

iv. Fit a vote-choice model (same caveats as above);

v. Create predictions for each voter-category in the stratification frame. This will require you to understand the code below the model specification; pay particular attention to the decomposition;

vi. Stratify the predictions to obtain National-level results. If you feel like you got the hang of it, aggregate at the state level as well to produce state-level predictions. Ensure these have uncertainty estimates around them (either they are simulations or they have a standard deviation value with a set of coherent assumptions about their distribution or prediction bounds);

vi. Reproduce state-level or national level historgrams to summarize results;

vii. Check that your results align (or do not align!) with current polling at https://www.realclearpolitics.com.

References

[1] Regularized prediction and poststratification (the generalization of mister p). https://statmodeling.stat.columbia.edu/2018/05/19/regularized-prediction-poststratification-generalization-mister-p/. Accessed: 2019-06-09.

[2] Baio, G. *Bayesian methods in health economics*. Chapman and Hall/CRC, 2012.

[3] Buttice, M. K., and Highton, B. How does multilevel regression and poststratification perform with conventional national surveys? *Political Analysis 21*, 4 (2013).

[4] Casella, G., and George, E. I. Explaining the gibbs sampler. *The American Statistician 46*, 3 (1992), 167–174.

[5] Gelman, A., et al. Analysis of variance—why it is more important than ever. *The annals of statistics 33*, 1 (2005), 1–53.

[6] Gelman, A., et al. Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian analysis 1*, 3 (2006), 515–534.

[7] Gelman, A., and Hill, J. *Data analysis using regression and multilevelhierarchical models*, vol. 1. Cambridge University Press New York, NY, USA, 2012.

[8] Gelman, A., Lee, D., and Guo, J. Stan: A probabilistic programming language for bayesian inference and optimization. *Journal of Educational and Behavioral Statistics 40*, 5 (2015), 530–543.

[9] Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[10] Ghitza, Y., and Gelman, A. Deep interactions with mrp: Election turnout and voting patterns among small electoral subgroups. *American Journal of Political Science 57*, 3 (2013), 762–776.

[11] Goel, S., Obeng, A., and Rothschild, D. Non-representative surveys: Fast, cheap, and mostly accurate. In *Working Paper*. 2015.

[12] Goplerud, M., Kuriwaki, S., Ratkovic, M., and Tingley, D. Sparse multilevel regression (and poststratification [smrp]). *Unpublished manuscript, Harvard University* (2018).

[13] Igielnik, R., Keeter, S., Kennedy, C., and Spahn, B. Commercial voter files and the study of us politics. *Pew Research Center Report, Washington, DC, available at www. pewresearch. org/2018/02/15/commercial-voter-files-and-the-study-of-us-politics* (2018).

[14] Jackman, S. Estimation and inference via bayesian simulation: An introduction to markov chain monte carlo. *American journal of political science 44*, 2 (2000), 375–398.

[15] Jackman, S., and Spahn, B. Why does the american national election study overestimate voter turnout? *Political Analysis* (2019), 1–15.

[16] Kiewiet de Jonge, C. P., Langer, G., and Sinozich, S. Predicting state presidential election results using national tracking polls and multilevel regression with poststratification (mrp). *Public Opinion Quarterly 82*, 3 (2018), 419–446.

[17] Kruschke, J. *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan.* Academic Press, 2014.

[18] Lauderdale, B. E., Bailey, D., Blumenau, Y., and Rivers, D. Model-based pre-election polling for national and sub-national outcomes in the us and uk. *Unpublished Work). Available online: https://www. jackblumenau. com/papers/mrp_polling. pdf (accessed on 5 December 2018)* (2017).

[19] Lax, J. R., and Phillips, J. H. How should we estimate sub-national opinion using mrp? preliminary findings and recommendations. In *annual meeting of the Midwest Political Science Association, Chicago* (2013).

[20] Lee, J. Y., Green, P. J., and Ryan, L. M. On the" poisson trick" and its extensions for fitting multinomial regression models. *arXiv preprint arXiv:1707.08538* (2017).

[21] Orloff, J., and Bloom, J. Conjugate priors: Beta and normal, Spring 2014.

[22] Park, D. K., Gelman, A., and Bafumi, J. Bayesian multilevel estimation with poststratification: state-level estimates from national polls. *Political Analysis 12*, 4 (2004), 375–385.

[23] Plummer, M., et al. Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing* (2003), vol. 124, Vienna, Austria.

[24] Wang, W., Rothschild, D., Goel, S., and Gelman, A. Forecasting elections with non-representative polls. *International Journal of Forecasting 31*, 3 (2015), 980–991.