Sparse Multilevel Regression (and Poststratification (sMRP))*

Max Goplerud† Shiro Kuriwaki‡ Marc Ratkovic§ Dustin Tingley¶

July 19, 2018

Abstract

Multilevel models have long played an important role in a variety of social sciences. We extend this framework by bring to bear recent developments in the machine learning literature to allow for considerable flexibility. We introduce a sparse regression framework that covers both the linear case as well as a logit model for binary outcome data. We leverage recent computational tricks based on data-augmentation to dramatically speed up estimation times with equal or better performance compared to existing approaches. We apply our model in the context of multilevel modelling with post-stratification which has become a common tool for survey researchers.

Key Words: multilevel models, sparse Bayesian modeling, data augmentation, multilevel regression with post-stratification, public opinion

*We would like to thank Christopher Warshaw for early discussions about the MRP literature. Questions and comments can be sent to Dustin Tingley. Paper not for redistribution without permission of authors.

†PhD Candidate, Department of Government, Harvard University, Email: goplerud@g.harvard.edu, URL: http://www.mgoplerud.com

‡PhD Candidate, Department of Government, Harvard University, Email: kuriwaki@g.havard.edu, URL: http://www.shirokuriwaki.com

§Assistant Professor, Department of Politics, Princeton University, Princeton NJ 08544. Phone: 608-658-9665, Email: ratkovic@princeton.edu, URL: http://scholar.princeton.edu/ratkovic

¶Professor of Government, Harvard University, Email: dtingley@gov.harvard.edu, URL: http://scholar.harvard.edu/dtingley

1 Introduction

Data where units cluster into known groupings are commonly encountered across a range of disciplines (Hox, Moerbeek and van de Schoot, 2017; Gelman and Hill, 2006; Gelman, 2006*a*; Singer and Willett, 2003). For example, on a national survey, respondents may be grouped by their state or region, or in a study of youth, students may be clustered within schools. Because data is often clustered in real world applications, building this structure into a regression model leads to a variety of improvements. Incorporating this known grouping can lead to dramatic improvements in the efficiency of estimated outcomes, by separating out within-group and across-group variation. This improvement carries through to prediction, where the predicted values contain a contribution from the group-level effect as well as the unit-level characteristics. Ignoring either can lead to misleading models.

While multilevel regression has grown in popularity, it has been constrained in practice by considering only a handful of covariates. As we move to a hierarchical model to more accurately represent the data, we argue that we should also allow for more complexity in the covariates and interactions we allow in the model as well. We illustrate here, through applied and simulation evidence, that there is indeed room for improvement.

To allow for flexibility in the hierarchical model, we provide a unified framework for sparse multilevel regression by connecting a variant of the LASSO model (Tibshirani, 1996; Zou, 2006; Ratkovic and Tingley, 2017) with multilevel regression. Furthermore, by leveraging some recent estimation "tricks" (Polson, Scott and Windle, 2013), we can easily extend estimation to the logistic regression. Consequently, we incorporate the gains from variable selection through the LASSO into the multilevel regression context, while allowing for binary outcomes and accommodating observation (e.g. survey) weights.[1]

One application of multilevel models is in the public opinion literature, where multilevel regression with post-stratification (MRP) has become increasingly popular. Popularlized by Park, Gelman and Bafumi (2004), MRP has two steps. First, a multilevel regression model is fit to public opinion data. Individual features of an individual are treated as 'normal' covariates ($\boldsymbol{X}$) and other features, such as the state someone is from, is treated as a random effect. Second, in the post-stratification

[1] Applications of LASSO type models to the context of multilevel regressions are relatively sparse (e.g., Bondell, Krishna and Ghosh, 2010; Ibrahim et al., 2011; Pan and Huang, 2014; Groll and Tutz, 2014).

step the results are combined with information about the distribution of individual characteristics within geographic units to obtain local level estimates of the survey outcome of interest. Typically this requires the joint distribution of the set of variables used in the survey model, though recent work suggests alternative strategies are useful (Leemann and Wasserfallen, 2017). The use of MRP extends beyond political poll applications, and includes work in epidemiology (Zhang et al., 2014, 2015). In this paper, we show how our sparse approach to multilevel models can be linked up to the post-stratfication step, enabling what we call Sparse Multilevel Regression and Post-Stratification (sMRP). This enables researchers to consider many potential variables that predict an outcome, including interactions between variables, when fitting an MRP model.[2]

The structure of the paper is as follows. In section 2 we briefly review multilevel models and MRP. Next in section 3 we introduce our modelling approach and section 4 provides performance simulations that highlight the usefulness of our approach. Section 5 applies our approach to real data where we show the advantages of our multilevel model. Section 6 concludes and highlights areas for future work.

2 Multilevel Regression

2.1 Multilevel models: a primer

Before introducing our approach we briefly review multilevel regression models as well as their application to survey data with post-stratification. We refer readers to Gelman and Hill (2006) for a more complete discussion of multilevel models. Throughout this paper, we will use a common running example: Attempting to predict the state-level support for the Democratic party using a non-representative national survey with known survey weights.

In order to estimate these state-level outcomes, a naive approach would be to subset the survey into each state and run a separate regression to predict state-level support. This runs into problems, however, given that some states may have a small number of observations.[3] This likely would lead to state-by-state regressions overfitting the data, perhaps by having variables that perfectly predict the outcomes inside the state, and thus lead to poor out of sample performance.

[2] Using sparse models with post-stratification is rare. For example, Si et al. (2017) use a structured prior to induce sparsity in the context of jointly estimating survey weights and post-stratifying.

[3] Of course, if one had a sufficiently large survey for each state, the 'first best' approach would be to stratify one's sample as that allows for most flexibly modelling the heterogeneity across units.

A common solution to this problem is multilevel modelling (Gelman and Hill, 2007). In the context of extrapolating survey results to smaller geographic units, the traditional framework is the following: Assume that the effect of our covariates (e.g. age, income, race) are *constant* across states; we refer to these as *fixed effects*. However, we can allow for level-differences in the support for the Democrats by adding the state as a predictor variable. Again, to avoid issues with possible overfitting and to enable some 'partial pooling' of information across the coefficients (Gelman and Hill, 2007), the traditional framework for multilevel regression includes the state dummy as a *random effect* that may, itself, be a function of some state-level covariates (Park, Gelman and Bafumi, 2004).[4] To outline this more formally, consider the following simple example: Our survey contains some number of observations i. We use the notation $j[i]$ to represent the state j that observation i is nested in. A simple formulation of our model of interest is shown below

$$Pr(y_i = 1|\boldsymbol{x}_i, \alpha_{j[i]}) = \frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta} + \alpha_{j[i]})}{1 + \exp(\boldsymbol{x}_i^T\boldsymbol{\beta} + \alpha_{j[i]})}; \quad \alpha_j \sim N(0, \sigma_\alpha^2) \tag{1}$$

In words, this is a logistic regression to predict the outcome (supporting the Democratic party) where some covariates $\boldsymbol{x}_i$ are included linearly and a random effect for the state $\alpha_{j[i]}$. A motivation for this paper is to introduce an estimation strategy that can handle a proliferation of covariates, including interactions between them. This framework can be easily generalized to include much more complex random effects, although we focus on this simple and common case for exposition. Our software that accompanies this package incorporates multiple random effects.

Random Effects As noted above, a key feature in the literature on multilevel regression (and hence those using MRP) is to include random effects for a geographic region. In our running example, including a random effect for state is standard; this is a way of estimating a level-difference between states—conditional on all other covariates, a voter in Massachusetts may be more likely to support the Democrats than a voter in Texas. The key benefit of the random effect approach is that the level shift in each state j is not estimated solely using the observations in that state; rather, the level of the random effect is a weighted average of the data coming from state j and the global average calculated across all states. This encourages 'shrinkage' where states with limited numbers of observations have random effect estimates that are closer to the global average (Gelman

[4] The traditional model includes *all* categorical variables as random effects, although simply putting some adaptive regularization, as our proposed framework does, onto those coefficients will avoid problems of separation.

and Hill, 2007). Below we revisit exactly how we utilize random effects and fixed effects in our model, as terminological slippage is large (Gelman et al., 2005; Gelman and Hill, 2007).[5]

2.2 Multilevel regression with *post-stratification*

In this paper we introduce a way to estimate multilevel models that model the outcome linearly or with a logistic function in the case of a binary outcome variable. Below we show how this naturally links up with a particular application of multilevel models, multilevel regression with *post-stratification*.

A multilevel model, on its own, is insufficient to predict state-level public opinion as it is based on a survey that, while hopefully nationally representative, is unlikely to exactly mirror the distribution of the population in specific geographic regions (i.e., the stratification units). Thus, Park, Gelman and Bafumi (2004) introduced multilevel regression with post-stratification (MRP) in order to obtain better estimates.[6] Given some model that we think predicts the survey data well, we combine that model with information about the known population distribution within the unit of interest—in this case, the states.[7] This model is then combined with information known about the within-state population distribution of variables that were used in the model to predict the outcome variable *within each region*.

In our running example of predicting Democratic support, traditional applications use variables such as age, ethnicity, gender, income, and education for which the joint distribution at the state level is known from census data. In a standard regression context, these would be entered as 'fixed effects' (i.e. turned into a data matrix $\boldsymbol{X}$). However, the traditional approach in the multilevel setting is to include all of these covariates as separate random effects, i.e. have a random effect for each level of each variable. A justification behind this approach is that the pooling of information across the coefficients in the random effects approach provides a small amount of regularization that improves performance. While that has been used to great effect, we show it is possible to do even better by incorporating more recent priors that allow for the identification of relevant interactions

[5] The sparse regression model we introduce below incorporates random effects, although it does not yet perform any sort of variable selection/regularization on the random effects.

[6] See also Breidt and Opsomer (2008) for a similar idea.

[7] It is important to note that there is no need for this procedure to inherently involve multilevel modelling, but the tendency in this literature has been to use this framework and thus we follow that trend. With our framework, another viable strategy is to interact the dummies for the geographic units with the covariates and give this to the regularization procedure, described shortly, and thus have no random effects at all. Whether this outperforms the procedure in this paper is a question for future research.

and leading to massive gains in performance while also being much faster to estimate than earlier models that included the interactions directly as random effects (e.g., Ghitza and Gelman, 2013).[8]

In the social sciences MRP has been employed extensively (Park, Gelman and Bafumi, 2006; Lax and Phillips, 2009*b*; Wang et al., 2015; Warshaw and Rodden, 2012; Tausanovitch and Warshaw, 2013; Ghitza and Gelman, 2013; Howe et al., 2015; Lax and Phillips, 2009*a*, 2012; Kastellec, Lax and Phillips, 2010; Tausanovitch and Warshaw, 2014; Mildenberger et al., 2016). And recently outside of the United States (Lauderdale et al., 2017). Often these models are fit using the non-Bayesian `lmer` or `glmer` routines in the lme4 R package (Bates and Sarkar, 2008). Some scholars use Bayesian methods via stan (Carpenter et al., 2017).

Along the way there have been various innovations. One is to encourage researchers to consider interactions between covariates in order to get better predictions. This, too, is a motivation of the current paper. For example, Ghitza and Gelman (2013) allow for interactions between variables by specifying each level of the interaction as a random effect. We provide a variable selection framework with some computational advantages.[9]

We provide an alternative estimation strategy to this standard approach. By leveraging a recent model for variable selection (Ratkovic and Tingley, 2017), we can examine a large possible number of interactions—beyond what was possible in Ghitza and Gelman (2013)—while retaining computational tractability (unlike tree based methods, e.g. Montgomery and Olivella (2018)) and performing competitively with state-of-the-art machine learning methods for uncovering non-linear relationships.

[8] A different strategy that is complementary to ours but we do not explore here is trying to regularize the random effects themselves, see work by Kinney and Dunson (2007); Bondell, Krishna and Ghosh (2010); Ibrahim et al. (2011); Fan and Li (2012).

[9] The potential value of using variable selection methods is implicitly highlight by the authors. "Here, however, we are modeling based on only three factors (ethnicity, income, and state), but the survey adjustments use several other variables, including sex, age, and education. Ultimately we want to fit a complex model including all these predictors, but for now we must accept that our regression does not include all the weighting variables." Montgomery and Olivella (2018) follow up on this in a reanalysis of the Ghitza and Gelman (2013) data to illustrate the use of random forests, "To illustrate, consider a model that allows for interactions between state, ethnicity, income, age, sex, education, marriage status and whether a person has children-the full array of demographic variables contained in the Ghitza and Gelman data, which could not be included in their MRP implementation *for computational reasons*" (emphasis added, p. 13).

3 Sparse Multilevel Modelling

Next we introduce our approach to multilevel regression that integrates with previous work on sparse regression techniques. In particular we extend the LassoPLUS framework of Ratkovic and Tingley (2017) to the multilevel context. We further innovate by allowing survey weights, extending the approach to a logit model, and introduce a *fast* approximate EM algorithm made feasible by utilizing recent developments in Bayesian statistics (Polson, Scott and Windle, 2013).

A Sparse Bayesian Regression Model We update LassoPLUS model in Ratkovic and Tingley (2017) which is a Bayesian LASSO (Park and Casella, 2008) model that has a sparse posterior mode and endogenous tuning parameters. It also incorporates the adaptive properties of the adaptive LASSO (Zou, 2006). The method was initially designed for the normal regression model, in the situation where the researcher had a large number of possible covariates but did not know which ones entered the model. The model is constructed so as to estimate the coefficients such that the estimation error is optimal [10], while zeroing out irrelevant variables. For a further introduction to LASSO models like ours, and related regression based variable selection tools, see Hastie, Tibshirani and Friedman (2010).

In this paper we extend the framework to both binary outcomes and a hierarchical structure in the data. We therefore extend the model to the hierarchical logistic regression, which is the workhorse model for applications like MRP.

Formally, we assume a set of p covariates $\mathbf{x}_i$ with associated parameters β, with the intercept β_0. We also assume a random effect that takes on one of $j \in \{1, 2, \ldots, J\}$ values, where $\alpha_{j[i]}$ indicates that observation i is in cluster j and takes on value α_j. We also include known survey weights δ_i.

[10] In that the estimation error decreases no slower than a provably optimal rate as a function of the sample size n and number of covariates p, the minmax rate.

The hierarchical representation of our model is now

$$\Pr(y_i|\mathbf{x}_i, \boldsymbol{\beta}, \alpha_{j[i]}) = \frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta} + \beta_0 + \alpha_{j[i]})^{y_i}}{1 + \exp(\boldsymbol{x}_i^T\boldsymbol{\beta} + \beta_0 + \alpha_{j[i]})} \tag{2}$$

$$\beta_k|\lambda, w_k \sim DE\left(\lambda w_k\right) \tag{3}$$

$$\lambda^2|n, p, \rho \sim \Gamma\left(\{n \times \log(p)\}^{.25} - p, \rho\right) \tag{4}$$

$$w_k|\gamma \sim \text{generalizedGamma}(1, 1, \gamma) \tag{5}$$

$$\gamma \sim \exp(1) \tag{6}$$

$$\alpha_j \sim \mathcal{N}(0, \sigma_\alpha^2) \tag{7}$$

$$\sigma_\alpha^2 \propto 1/\sigma_\alpha^2 \tag{8}$$

$$\beta_0 \propto 1 \tag{9}$$

where $DE(a)$ denotes the double exponential density, $\Gamma(a, b)$ denotes the Gamma distribution with shape a and rate b, and generalizedGamma(a, b, c) the generalized Gamma density $f(x; a, d, p) = \frac{p/a^d}{\Gamma(d/p)} x^{d-1} \exp\{-(x/a)^p\}$. We have two remaining prior parameters. We set $\rho = 1$ in the generalized Gamma density. The prior term on λ^2 governs the asymptotic properties of the estimator both as the sample size (n) and number of covariates (p) grows. A key concern in any Bayesian model is how to select these prior parameters. In the least squares setting, taking $\lambda^2 \propto n \times \log(p)$ gives us an optimal (minmax) prediction rate of $\lambda \approx \sqrt{n \log(p)}$. We have found this rate to be too aggressive in the logistic regression setting so we adopted the prior above and found that it performed well in our simulations and applied examples.[11]

Random effects As mentioned earlier, the exact definition and explanation of what random effects are doing in a model is quite variable. We, please reader, are not trying to wade into a terminological morass on this point.[12] Instead, we briefly describe the role of these two different parameters in our model.

In our model, there is a simple distinction: When we say 'random effect' α_j, we mean a quantity that has a normal prior whose variance is random (i.e. σ_α^2 is a parameter in our model) and shared across all random effects α_j. A random effect would be specified by a researcher providing a single variable with some number of discrete levels (e.g. 'state') to the function; it is *exactly* analogous to

[11] We are currently working to derive a rate-optimal value for this term.

[12] See Gelman and Hill (2007) for a more detailed discussion on this topic.

random effects in `lmer` (Bates and Sarkar, 2008). The 'fixed effects' $\boldsymbol{\beta}$ correspond to the variables that the researcher provides in the form of a rectangular matrix ($\mathbf{X}$) as is standard in regression. From the researcher's point of view, it is entered in an identical way to the 'fixed effects' in `lmer`. However, the key innovation of this paper is to place the lassoPLUS prior (outlined above) on these coefficients to induce sparsity; by contrast, the traditional non-sparse set up places independent normal priors (whose variance is fixed) on each coefficient. This traditional case with a flat prior can be thought of a limiting case of the lassoPLUS prior when there is no sparsity.[13] As within any multilevel model, the decision of what variables to treat as fixed effects versus random effects is a substantive choice.

Incorporating Survey Weights In practice most surveys come with survey weights to allow for estimation on a representative sample. Incorporating weights into our model is straightforward; for our purposes, we rely on the provided weights from the surveying organizations, although one could in principle estimate the weights using methods such as those outlined in Caughey and Hartmann (2016). As is standard, we include weights as a multiplicative factor on the log-likelihood scale. Weights do not change the computational complexity of the model at all; rather, they can be trivially folded into the model updates as outlined in Appendix A.

Construction of Interaction Terms In the standard all variables are entered in additively. This of course may not be appropriate. The outcome variable might be better modelled as including interactions between covariates, as in Ghitza and Gelman (2013). However, a naive approach of including all possible interactions as fixed effects will likely lead to over-fitting and thus poor performance. Ghitza and Gelman (2013) adopt a different solution; using the traditional approach of including the predictors as random effects, they include random effects for each level of the interaction. For example, their ethnicity variable has five levels and their income variable has four levels; their random effect on 'income x ethnicity' thus has 20 levels. Their framework, however, only includes a limited number of interactions and, as the number of random effects increases, the computational time increases enormously.

We suggest a different approach that leverages the sparse model outlined above; in the fixed effect

[13] As the lassoPLUS prior implies some pooling of information between the coefficients via the shared regularization parameters (λ, γ) to calibrate the level of sparsity, it differs from the 'no pooling' case commonly associated with fixed effects. Thus, it could be thought of as 'partial pooling' in a way that induces sparsity, rather than the normal 'partial pooling' associated with a 'random effect'.

portion of our model, we include a large number of interactions between the covariates (all pairwise interactions, by default) and apply the sparse modelling strategy outlined above to eliminate many irrelevant interactions. To help resolve the high degree of collinearity of the data entered into the fixed effects portion of the regression, we partial out the lower order terms as part of a pre-processing step (see Ratkovic and Tingley (2017) for discussion). Open source software provided with this paper constructs interactions automatically, or researchers can create their own interactions.

Estimation The complete model we seek to estimate, with random effects and survey weights, is a slight adaptation of Equation 1. We add the possibility of survey weights (δ_i) on a multiplicative log-scale by raising each term in the likelihood to the power of the weight. We write this in the canonical non-Bayesian formulation where our likelihood to optimize integrates out the random effects.

$$l(\boldsymbol{\beta}, \sigma_\alpha^2) = \int \prod_i \left[\frac{\exp(\boldsymbol{x}_i^T \boldsymbol{\beta} + \alpha_{j[i]})^{y_i}}{1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta} + \alpha_{j[i]})}; \right]^{\delta_i} \prod_j \phi(\alpha_j; 0, \sigma_\alpha^2) d\alpha_j \tag{10}$$

Estimation of this model is challenging using standard methods for basically two reasons. First, the raw likelihood function itself has an integral. In the linear case, one can use a standard EM algorithm to address this (Laird and Ware, 1982; Meng and Van Dyk, 1998); or, one can approximate the integral using various forms of quadrature methods (Bates and Sarkar, 2008). Second, in the non-linear case, the standard EM solution does not work. And in addition to having to rely on quadrature methods, it is also necessary to rely on some iterative procedure, possibly unstable and dependent on starting values, for finding the maximum likelihood of the $\boldsymbol{\beta}$ (e.g. Fischer Scoring or Newton-Raphson). This is because there is no closed form solution for the coefficients of a multivariate logistic regression. In the fully Bayesian framework, the problems are compounded as the logistic link is intractable and thus the simple updates of a linear random effects model Gibbs Sampler are not applicable.

As a result, the non-linearity of the logistic random effects model leads to large computational burdens for researchers. For example, using the non-Bayesian approach employed in the lme4 package (i.e., approximating the integral with quadrature), the main model presented in Ghitza and Gelman (2013) takes 40-60 minutes on a 3.1Ghz laptop. We instead use recent developments in Bayesian "data augmentation" to avoid some of these problems and rapidly increase the speed of computation for this non-linear model. Data augmentation transforms a seemingly intractable

problem into a simple one by showing that there is some latent variable that, if known, would make the problem tractable. Since the latent variable is unobservable, data augmentation estimation algorithms iterate between updating the latent variable based on the observed data and a current guess to the parameters and then updates the parameters based on the latent variable. As the particular form of data augmentation we are using (Polya Gamma - Polson, Scott and Windle (2013)) is relatively new and has seen limited use in the social sciences, (see, e.g., Goplerud, 2018; Goplerud et al., 2018), we outline it briefly.

Data Augmentation Primer - The Probit case Explaining Polya-Gamma data augmentation to a new reader can be difficult. However, showing its conceptual similarity to the latent variable interpretation of a probit regression can be helpful. After providing intuitions, we shift to the logit case which is the more common model in the MRP literature.

Consider the likelihood implied by a probit regression.

$$\ell(\boldsymbol{\beta}) = \prod_{i=1}^{N} \Phi(\boldsymbol{x}_i^T \boldsymbol{\beta})^{y_i} \left(1 - \Phi(\boldsymbol{x}_i^T \boldsymbol{\beta})\right)^{1-y_i} \tag{11}$$

There is no closed-form solution to this problem and thus common optimization techniques rely on iterative procedures like Fisher Scoring. However, there is a different way to find the maximum likelihood estimates of $\boldsymbol{\beta}$: data augmentation. This refers to a broad procedure that casts a difficult problem into a simpler one if two conditions are satisfied: (i) the difficult problem can be written equivalently as one involving the observed data y_i and some latent variable z_i for which, if we knew both (y_i, z_i) for all observations, it would be easy to solve; (ii) the conditional distribution of the unobserved z_i is tractable given y_i and the parameters of the model (Tanner and Wong, 1987). For the probit case, data augmentation relies on the well-known identity that:

$$P(y_i = 1) = \Phi(\boldsymbol{x}_i^T \boldsymbol{\beta}) = \int \phi(z_i; \boldsymbol{x}_i^T \boldsymbol{\beta}, 1) \left[I(z_i > 0)^{y_i} + I(z_i < 0)^{1-y_i} \right] dz_i \tag{12}$$

This says, we can think of a probit link as having some latent variable z_i that, if positive, $y_i = 1$ and if negative, $y_i = 0$. We can thus write the probability of y_i in terms of the latent variable as shown on the second line. Further, and crucially, even though z_i is fundamentally unobservable, its distribution given the observed data, i.e. $z_i|y_i$, has a simple distribution (truncated normal).

The reason for writing a model in terms of some latent variable z_i is that, if we knew z_i, we

could estimate $\boldsymbol{\beta}$ in a single step by performing linear regression of z_i on $\boldsymbol{x}_i$. However, since z_i is unobserved, we have to use an iterative procedure. The details vary but the intuition is that we plug in some guess as to the z_i that is based on the observed data y_i and the current iteration of the parameters and then perform a linear regression of that estimate of z_i on the data $\boldsymbol{x}_i$ to update the $\boldsymbol{\beta}$. In the non-Bayesian case, this procedure is an implementation of the Expectation Maximization algorithm (Dempster, Laird and Rubin, 1977) and can be shown to deterministically converge to the maximum likelihood estimate of the $\boldsymbol{\beta}$. In the fully Bayesian case, this procedure (a Gibbs Sampler) will eventually give us a sample of $\boldsymbol{\beta}$ that converges to the true posterior distribution.

Polya-Gamma augmentation primer-the logistic case While multilevel models can be framed using a probit link, the many scholars and especially the MRP literature has almost exclusively adopted the logistic regression model in both academic and industry usage. Estimating using a logistic link, however, was typically assumed to be rather complicated: Bayesian models had to rely on algorithms that had internal tuning parameters (e.g. Metropolis-Hastings) that necessarily resulted in some inefficiencies in the sampler. This slowed down the estimation—especially as the size of the data increases—resulting in models that took a very long time evaluate. Non-Bayesian implementations (e.g. Bates and Sarkar (2008)) also run into computational difficulties as they needed to perform increasingly complex numerical integration as the number of random effects increases. Further, attempts to estimate models quickly, e.g., by variational inference, turn on approximations that may lead to rather poor out-of-sample performance (see our simulations below).

However, a recent development in data augmentation by Polson, Scott and Windle (2013) elegantly resolves all of these issues. To quote the authors, their form of data augmentation allows models involving logistic links, including but not limited to multilevel logistic regression, to be estimated using "simple, effective methods for posterior inference that (1) circumvent the need for analytic approximations, numerical integration, or Metropolis Hastings; and (2) outperform other known data-augmentation strategies, both in ease of use and in computational efficiency" (p. 1339). Their analysis shows that Polya-Gamma augmentation which can be estimated via a Gibbs Sampler is much more efficient than traditional samplers in terms of higher effective sample sizes and lower autocorrelations for any given run of a sampler (p. 1347). In practice, this means that researchers can likely run their samplers for much shorter lengths of time to achieve sufficiently good mixing, and can estimate even more complex models in a reasonable time.

For our purposes, this data augmentation also admits an Expectation Maximization algorithm to find the posterior mode that is directly analogous to the probit outlined above that is orders of magnitude faster than full Bayesian samplers but have competitive performance. We show this in Figures 1 and 2 better performance and run times of at least x100 faster than a state-of-art Hamiltonian Monte Carlo implementation of the multilevel logit model.

It is not an overstatement to suggest that this development has revolutionized estimation of non-linear Bayesian models and is rapidly becoming the 'default' option—analogously to the response to the development of probit data augmentation. It is now standard in new papers performing tasks such as logistic regression, multinomial regression (Polson, Scott and Windle, 2013), ideal point estimation (Goplerud, 2018), topic models (Jianfei et al., 2013), and others. This development has yet to penetrate deeply into the social sciences (with some exceptions, e.g. Goplerud (2018); Goplerud et al. (2018)), and thus we explain it here in some detail.

As outlined above, the crucial identity in the probit augmentation is showing that the probit link can be written as the integral over some latent variable such that the marginal distribution of y_i has the correct form. Polson, Scott and Windle (2013) showed that the following identity exists for the logistic link:

$$P(y_i = 1) = \frac{\exp(x)^a}{(1+\exp(x))^b} \tag{13a}$$

$$P(y_i = 1) = \frac{\exp(x)^a}{(1+\exp(x))^b} = 2^{-b}\int_0^\infty \exp\left([a-b/2]x - \omega x^2/2\right) f(\omega|b,0)d\omega; \quad \omega \sim PG(b,0) \tag{13b}$$

Here, the augmentation variable ω serves as the tool to turn the logistic link into a tractable form—note that, conditional on the augmentation variable, the log-likelihood is linear. Unlike in the probit case, ω does not have a clear latent variable interpretation and thus it should be thought of as more of a book-keeping trick than having some inherent meeting. In the second line, 'PG' denotes a Polya-Gamma random variable with shape parameter $b > 0$ and scale $c \in \mathbb{R}$. The distribution of the augmentation variables ω cannot be expressed in a simple form, but they can be represented as an infinite weighted sum of independent gamma random variables:

$$\omega \sim PG(b,c); \quad \omega = \frac{1}{2\pi^2}\sum_{n=1}^{\infty}\frac{Z_n}{(n-1/2)^2 + c^2/(4\pi^2)}; \quad Z_n \sim^{iid} Gamma(b,1) \tag{14}$$

The key benefit of this augmentation is that, despite its complicated density, their mean is known in closed form and they are conditionally conjugate, i.e. $p(\omega|x) \sim PG(1, x)$,[14] and thus inference is straightforward. In an analogous fashion to the probit case, we update our estimate of the Polya-Gamma variables using the identity noted (equivalent to updating the z_i in the probit case). Then, one can perform a weighted least squares to update $\boldsymbol{\beta}$ using the second line of Equation 13.

For our purpose–and here a critical contribution of the present paper–this data augmentation approach means we can adapt the estimation algorithms in Ratkovic and Tingley (2017) for fitting sparse regression models. The highly competitive performance of that model against other machine learning methods in their paper suggests that it will also have large gains, that we demonstrate below, when applied to the MRP case.[15]

This adaptation is straightforward as the Ratkovic and Tingley (2017) framework does not require a separate re-derivation for the logistic case. Rather, we can simply perform a Polya-Gamma step at the beginning of each iteration of their algorithm and then perform all of their updates exactly as described in their paper and the relevant appendices.

Estimation via the EM algorithm As outlined in detail in Appendix C, our EM implementation relies on the following idea: If we were trying to estimate a standard multilevel logistic regression, we could first augment based on the Polya-Gamma random variables. If those were known, the augmented model can be written as a linear mixed effects model that can, itself, be estimated via an exact EM algorithm (Laird and Ware, 1982; Meng and Van Dyk, 1998). The representation of LASSOplus as a mixture of normal random variables, see Appendix B, shows that we can perform a marginally modified version of this linear mixed effect update to similarly estimate the EM algorithm for our proposed model.

A slight complication occurs, however, because calculating the expectations of the Polya-Gamma variables in the presence of random effects is, unfortunately, not trivial. We rely on an approximation of a proper EM algorithm, i.e., approximating the E-Step based on a point mass of high density,

[14] The mean of a Polya-Gamma b, c variable is Polson, Scott and Windle (2013):

$$\frac{b}{2c}\tanh(c/2)$$

[15] Of course, other machine learning methods, e.g. the horseshoe, could be adapted to rely on Polya-Gamma augmentation. For example, recently Makalic and Schmdit (2016) propose a Gibbs Sampler to do this.

that performs extremely well in our simulations and on actual data.[16]

The benefits of the EM algorithm are numerous: It is very fast (see Figure 2) and performs well in our simulations and on actual data versus fully Bayesian methods. Further, it is able to scale to datasets of a size at which fully Bayesian methods (e.g. a Gibbs Sampler) would be unable to feasibly run. However, the EM algorithm is not without costs; first, it returns only a point estimate and thus calculating measures of uncertainty require further work, either by bootstrapping or using analytical approximations such as those in Ratkovic and Tingley (2017). Moreover, the EM algorithm is approximate in that because the method relies on an approximate E-Step as noted above.[17] This may trouble researchers who care about a model with guaranteed convergence, although if the model's predictions perform well on held-out data or other validation metrics, the approximations are perhaps less troubling.

Post-Stratification Researchers wishing to do a post-stratification step can take our estimates of the coefficients and random effects from the model and calculate the predicted probability of the outcome ($\tilde{p}$)) for each unique combination of covariates ($\tilde{\boldsymbol{x}}$) in the dataset used to fit the model. In practice this is done through a simple predict function in the statistical software. We then construct a weighted average for each post-stratified unit (e.g. state) based on the distribution of the $\tilde{\boldsymbol{x}}$ in that state. In our sparse procedure, where we interact all of these covariates together, the interactions of those covariates alone does not complicate the mechanics of post-stratification. It simply means that the predicted outcome for each combination of covariates is calculated ($\tilde{p}$) in a more complex way accounting for the interactions, but the post-stratification proceeds *identically* to the linear additive case as that is simply a weighted average of the predictions based on the distribution of the covariate profiles $\tilde{\boldsymbol{x}}$ in the population.

The classical applications of MRP are limited by the fact that we have to *know* the distribution of $\tilde{\boldsymbol{x}}$ in order to post-stratify. For example, using the canonical case, if we are extrapolating to

[16] In slightly more technical terms, we perform the correct M-Step implied by our EM algorithm, but we must rely on an approximate E-step because of the mix of both Polya-Gamma variables and random effects. A variety of solutions are possible besides the one we employed, e.g., quadrature methods, but the approximate solution is faster, scales to multiple random effects with ease, and is computationally tractable for the cases that are commonly applied to random effects. We also note, related to our comments above, that one could simply include the random effects as 'fixed effects' and use the regularization inherent in lassoPLUS to stabilize their values. We further stress that in the case of a model with no random effects, the EM algorithm is exact (no approximations needed) and has the usual convergence guarantees.

[17] Another slight point to note is that the EM algorithm maximizes the joint posterior of $\boldsymbol{\beta}, \sigma^2_\alpha, \gamma, \lambda$ versus the typical desire to find the posterior mode over $\boldsymbol{\beta}, \sigma^2_\alpha$ alone.

states using census data, our regression cannot use a variable (such as partisanship) for which the joint distribution with all of the other covariates (e.g. age) is unknown in the population.

One benefit of our approach, though, is that we can add in other covariates (for which we only have a marginal distribution for) and include their complex interactions with the existing variables in a regularized way that will often increase the model's performance without falling prey to overfitting. To do this, however, one needs to rely on an approximate method for post-stratification. Leemann and Wasserfallen (2017) consider the case where the joint population distribution assumes independence between the underlying variables. They then approximate the joint distribution by the product of the marginal distributions [18]

Importantly, we note that the independence assumption is only necessary for a variable whose joint distribution with the others is unknown. For example, we can post-stratify using the correct joint distribution for age, education, and ethnicity—but we would create synthetic joint distributions assuming independence with additional variables, such as partisanship. Thus, the joint distribution of the census variables, marginalizing away the census variables, remains exact.[19] We are aware this is, at best, a rather crude approximation; however, as Leemann and Wasserfallen (2017) point out it may provide markedly better performance.[20] We discuss future directions that might improve this step of the process in the conclusion, although we note that one can stick only to variables for which the joint is known and still see improvements by including a large combination of interactions of those variables.

4 Performance Simulation Evidence

Next we investigate the performance of our modelling approach across several different contexts. We vary the number of candidate fixed effects, whether the true data generating process has interactions

[18] They suggest another procedure, 'adjusted synthetic distributions', that involves using the survey to estimate correlations between the variables when post-stratifying. In our running example, one can use the fact that partisanship, education, and age are all measured in the survey to better create joint distributions that reflect the observed correlations when post-stratifying.

[19] More formally, one could define a random variable $\tilde{X}$ that takes on all possible combinations of values in the known joint data, e.g., 20 if our two variables are ethnicity and income as coded above. With *that* variable, we then use an independent assumption with the marginal distribution on, say, partisanship to create the synthetic joint.

[20] Ornstein (2017) provides some theoretical justification for this approximation: If the underlying first stage model is additively linear in the included covariates (and one estimates a linear model), then the synthetic method outlined above will return the same post-stratified estimates as classic MRP if the true joint was known. Most applications of MRP rely on non-linear models, however, and thus further work to see whether similar guarantees can be obtained is an interesting question for future research.

between fixed effects, and whether or not the random effects are correlated with the fixed effects. Here we focus only on the case of a binary outcome using logistic regression.[21]

Simulation Structure We use several simulation setups to vary the complexity of the data generation process. There are two dimensions along which we increase the complexity: adding interaction terms to the model between fixed effects and allowing a correlation between the random effects and our covariates. We assess how well each method performs when confronting these two scenarios. Throughout we fix the sample size at $n = 1000$ and conduct 500 Monte Carlo simulations per setting.

The underlying structure of the simulation has a vector of covariates $\mathbf{x}_i$ drawn from a multivariate normal, where the correlation between each pair of covariates is 0.5 and the data are generated as

$$y_i \overset{\text{i.i.d.}}{\sim} \text{logit}^{-1}\left(-1 + \mathbf{x}_i^\top \beta + a_{j[i]} + b_{k[i]}\right) \tag{15}$$

where a and b denote random effects. We adopt the notation of Gelman and Hill (2006), where $a_{j[i]}$ denotes that observation i is in group $j[i]$ with level a_j, and similarly with b_k. Across settings, $j \in \{1, 2, \ldots, 8\}$ and $k \in \{1, 2, \ldots, 25\}$ with each group of approximately equal size. Across settings, $a_1 = 0.25$, $a_3 = -.25$, and $a_4 = .1$, and $b_1 = .25$, $b_3 = -.25$, $b_4 = .1$, and $b_{12} = .1$

When we estimate the model, we also include a set of other covariates that in truth do not impact the outcome. Throughout all simulations we vary the number of these covariates entered into the model over $p \in \{10, 25, 50, 100\}$.

In the first, "No interactions/No RE correlation", setting, we simulate data from a model with a set of random effects and a set of covariates that enter linearly into the model. That is, there are no interactions among the fixed covariates, and the groups are assigned completely at random so there is no correlation between between the random effects and covariates.[22]

This setting serves as a benchmark, as all of the assumptions for the proposed method, `STAN`, and `glmer` are satisfied. We expect `glmer` to perform well in the low p setting but to degrade as p increases, due to the lack of regularization. We expect the `STAN` model with the horseshoe prior to perform better than `glmer`, due to the impact of the prior. The `glmmlasso` should also perform

[21] In the case of a linear regression we found that the efficacy of our approach was even stronger than what we report below.

[22] Formally, we take $\beta = [1, .8, .6, .4, .2, 0, \ldots, 0]$, and observations are assigned to clusters in $\{a, b\}$ completely at random.

reasonably well given its use of regularization.

In our second simulation setting, "Interactions/No RE correlation" the fixed effects that in truth impact the outcome also impact the outcome in an interactive way. Here, all two-way interactions of these fixed effect terms have an impact on the outcome. Fixed effects (p) that do not, in truth, impact the outcome linearly also do not impact the outcome in any interactive way. However, because in practice an analyst does not know ex ante what is the right model specification we still enter into the estimation model all two way interactions. Hence the estimation model contains all two way interactions between all fixed effects. Importantly, we note that while our software allows the analyst to automatically calculate these interaction terms as an option, other candidate models do not. We none the less pass this larger covariate space to these other estimators.

In the third setting, "No interactions/RE Correlation", we induce some correlation between cluster assignment in a, b (the random effects) and the covariates. We keep the same setup as in our first setting, but we assign units to a, b based on coarsened covariates.[23] We set the correlation to be relatively high in order to differentiate this simulation setting from the first one. In our final simulation setting, "Interactions/RE correlation", we simply add the same correlation structure to the model where we had interactions between fixed effects.

Evaluation Metric We evaluate the model performance via deviance, which is -2 times the log-likelihood.

$$dev(\widehat{p}) = -2 \times \sum_{i=1}^{n} p_i \log \widehat{p}_i + (1 - p_i) \log(1 - \widehat{p}_i) \tag{16}$$

The model deviance is the standard measure of the performance of a model, relative to the true probabilities p_i. In a normal regression model with error variance 1, it reduces to the residual sum of squares. For the logistic regression, it takes a different functional form but inherits the same basic property that lower values are preferable. We then scale the deviance metric by the sample size, which in our simulations is $n = 1000$.

[23] Specifically, we take the third covariate $\mathbf{x}_3$ and sort it from largest to smallest. We then place units in the top octile in group a_1, those in the next in a_2, and so on to a_8. For the next random effect, we sort $\mathbf{x}_5$, place units in the top $1/25^{th}$ in b_1, the next in b_2, and so on to b_{25}. In this way, the random effects are not independent of the covariates. Regressing the covariate on the categories as fixed effects gives an R^2 of approximately 0.9, so the categories are highly collinear with the covariate.

Comparisons In practice, most researchers utilize the glmer function in the `lmer4` package to fit multilevel logistic regression model (Bates and Sarkar, 2008). Thus we benchmark our performance against status quo practice. In addition to including this model in our simulations, we also include two models from the STAN library (Carpenter et al., 2017): a variational version of the logit multilevel model with a horseshoe prior (Carvalho, Polson and Scott, 2010) and a version of the same model but fit using a full run of their Hamiltonian Monte Carlo routine.[24] Finally, we include the `glmmLasso` model (Groll, 2017; Groll and Tutz, 2014) which uses gradient ascent to apply regularization in the logistic random effects framework.[25]

Results We present our results starting with Figure 1 which presents the average model deviance for each simulation setting and number of candidate covariate (p). The best performing models throughout all the simulation settings were our proposed method (black line) and the rstan model with horseshoe prior fit with a Hamiltonian Markov chain sampler (rstanHMC). The rstan model fit via a variational approximation did the worst throughout the simulation settings (rstan). The `glmmlasso` model performed well at low p but its performance quickly degraded as the number of candidate variables increased.[26]

Given the competitive performance of rstanHMC Figure 2 presents the ratio of the average amount of time to fit the rstanHMC model to our proposed method. The results are dramatic, with our proposed methods regularly over 100 times as fast with no compromise in performance (as shown in Figure 1). In some sense, this comparison is unfair:[27] Our proposed method gives only a point estimate whereas rstanHMC gives a full sample of the posterior distribution for calculating uncertainty.[28] However, if one is merely interested in point prediction, our method is far superior—much faster and equally good performance. Indeed, the fastest alternative option (HMC via variational inference) does markedly worse than our method and is roughly comparable in terms

[24] The horseshoe prior is a prior similar to the LASSOplus, in that it will both aggressively zero out small effects while leaving larger effects approximately unbiased.

[25] We also fit a standard logit model including the random effects as fixed effects. The performance of this model unsurprisingly was not competitive and so we exclude it below.

[26] The results are presented on an expected deviance scale, so multiplying a difference by $n = 1000$ will be on a χ^2 scale. Lower values are to be preferred.

[27] We are currently working on the Gibbs Sampler implementation of our model that we believe would have both better performance than our EM method and would still have speed gains on HMC because of the computational gains inherent in using Polya-Gamma data augmentation outlined above.

[28] Note that the bootstrap procedure can be applied to our method to get estimates of uncertainty in times that are competitive with rstanHMC—especially noting that the bootstrap is trivially parallelized.

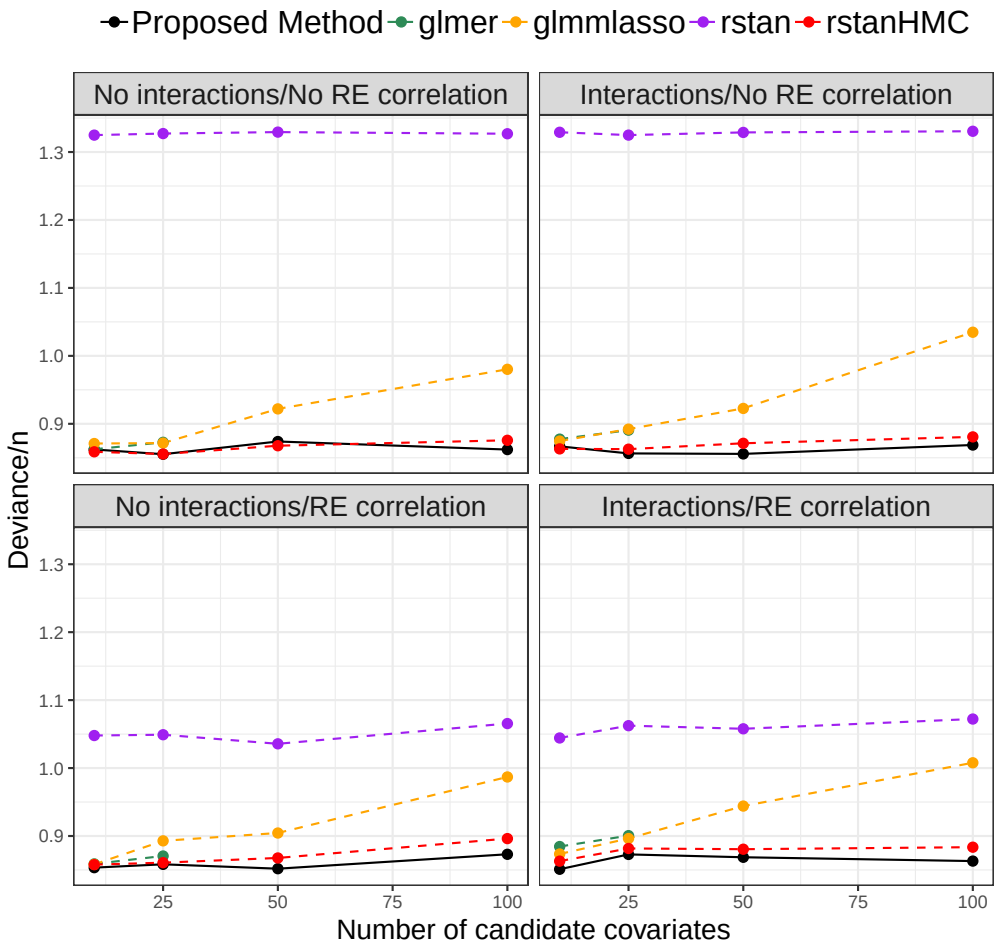

Figure 1: **Deviance estimates**. Average deviance, divided by sample size, by simulation. Each pane represents a different simulation setting and the x-axis displays the number of additional candidate covariates (p). Results for `glmer` with $p > 25$ are omitted due to non-convergence.

of speed.[29]

5 Application

Finally, we evaluate our approach in practice by re-analyzing data from previous work using multilevel modelling with post-stratification. For this we turn to the analyses and replication data provided by Ghitza and Gelman (2013).

Using a series of Pew polls conducted during the Fall of 2008[30], Ghitza and Gelman (2013) predict

[29] A plausible explanation for why our method outperforms the variational method is that said method is not guaranteed to maximize the *true* posterior while our method targets the true posterior directly.

[30] Pew polls were largely done by random digit dial with a supplemental cell number. Ghitza and Gelman (2013) provide a cleaned version of about ten different Pew polls, standardizing demographic variables and selecting

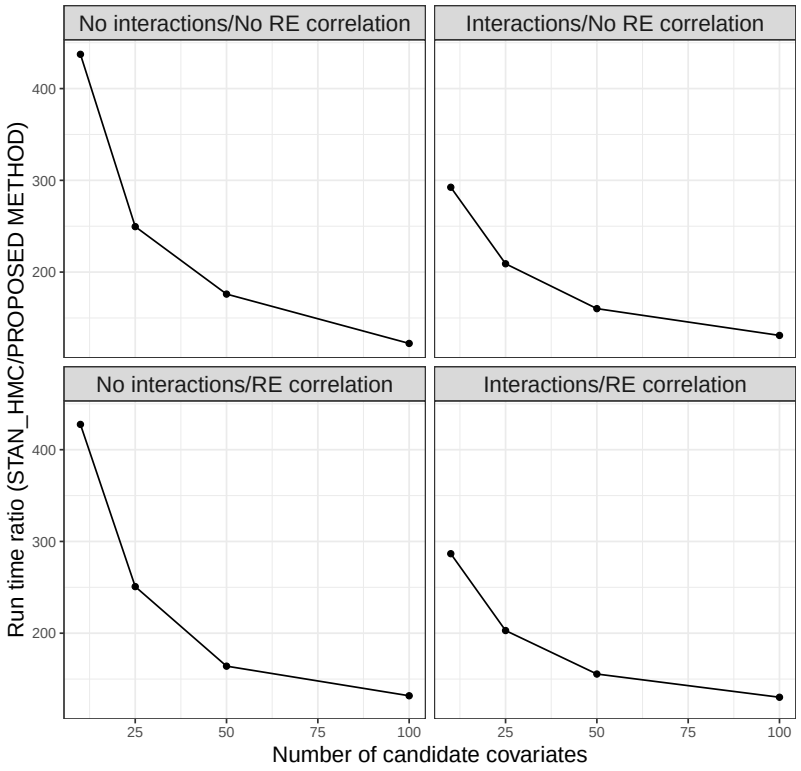

Figure 2: **Ratio of average simulation time for rstanHMC to EM version of proposed method.** Results show that even in the most complicated settings ($p = 100$) the proposed method is 100 times faster yet still maintains equal or better performance as shown in Figure 1.

state-level estimates of the 2008 Republican presidential vote share using approximate marginal maximum likelihood estimates by `glmer`. They estimated a broad range of models. We consider the two specifications they focus on as benchmarks. The first, what we refer to as their simple model below is a specification with random-effects for state, race, income, and age.[31] By not modeling any of the covariates as fixed effects, the model avoids estimating coefficients on demographic variables by the usual maximum likelihood, instead using a prior for each of the four variables and computing the parameters for the underlying Gaussian distribution of random effects.

They also estimate a "deep interaction" specification where random effects are estimated for 18 groupings, each grouping being some combination of the demographic variables. Below we refer

only registered voters.

[31] In R code, the specification is `glmer(y ~ (1 | eth) + (1 | inc) + (1 | age) + (1 | stt), family = binomial(link = "logit"))`. Here `y` is the binary outcome for intending to vote for McCain, `stt` is a categorical variable for state, `eth` is a categorical variable for race, `inc` is a categorical variable for income group, `age` is a categorical variable for age group. In addition, `reg` is a categorical variable for the Census region and `z.inc` is the income grouping transformed into a numerical Z-score.

to this as their full model. Even by starting with six categorical variables, the number of possible coefficients easily reaches the order of hundreds once interactions between each variable and random effects for each interaction are included. Drawing on previous work that shows how the effect of income on political attitudes vary considerably by state (Gelman et al., 2007), the authors choose to include multiple terms involving income.[32] All categorical variables are entered as random effects. Ghitza and Gelman (2013) also translate income group into a continuous Z-score, and compute its fixed effects with state-level income Z-score. Finally they estimate separate coefficients of the continuous variable, Z-score of income, within each categorical group of region, state, race, and age. This additional specification allows for the effect of income on vote choice to vary by geographic group, instead of the impact of geographic-groups affecting estimates by a common factor. This is what is often called a varying slopes specification[33]. Other random effects do not include varying slopes, but compute varying intercepts for interactions between categorical variables. Even this extensive model is not the most complex specification that could be specified with the amount of data in the replication dataset.

5.1 Analysis

Software developed as part of the `sparsereg` development package implements our proposed method and the necessary workflow. This enables the estimation of a variety of multilevel models. The workflow also enables easy routines facilitating the post-stratification step, rather than hand rolled processes.

Figure 3 shows a summary of our approach using the R syntax. The `sparsereg` software allows users to estimate logit regressions with arbitrarily many varying-intercept random effects, using the same interface at that of the existing R packages `lme4` and `stanarm`.

There are two main data-intensive tasks when generating MRP models, and our codebase facilitates each of these steps as well as the final step of combining the two. First, the analyst builds a predictive model for the quantity of interest at the survey respondent level. While the MRP notation implies that this regression be a multilevel model, in fact the model can be any form of

[32] In R code, the specification is `glmer(y ~ z.inc * z.incstt + z.inc * z.trnprv + (1 + z.inc | reg) + (1 + z.inc | stt) + (1 + z.inc | eth) + (1 + z.inc | age) + (1 | inc) + (1 | reg.eth) + (1 | reg.inc) + (1 | reg.age) + (1 | stt.eth) + (1 | stt.inc) + (1 | stt.age) + (1 | eth.inc) + (1 | eth.age) + (1 | inc.age) + (1 | stt.eth.inc) + (1 | stt.eth.age) + (1 | stt.inc.age) + (1 | eth.inc.age),family = binomial(link = "logit"))`. See footnote 31 for description of terms.

[33] Our framework does not yet support varying slopes but will.

Figure 3: **Pseudo-code describing MRP workflow with `sparsereg`**. MRP estimation comprises two distinct steps. First, analysts fit a regression model predicting the individual-level quantity of interest (Step 1a). We employ the widely used lmer syntax for specifying random effects: `(1| state)` indicates that random effects for each state are modeled. The model is then used to generated predicted values for an out of sample schedule of all demographic and state combinations (Step 2). A post-stratification target that dictates the relative weight of each demographic profile in each geography is prepared beforehand (Step 1b). These predicted values are then averaged to the state-level by using the target distribution as weights (Step 3).

Step 1a: Fit multi-level model

```
# EM = TRUE allows for fast estimation via data augmentation
fit <- sparsereg(vote ~ age + income + race + (1|state), type = "logit", EM = TRUE)
```

Step 1b: Prepare post-stratification target

```
 # count census cells
cell_size <- count_cellsize(census, popvar = pop2008, geovar = state, race, age, income)
# add on additional variables if synthetic
synth_size <- synth_cellsize(cell_size, marginals)
```

Step 2: Predict onto each demographic cell

```
pred_cell <- predict(fit, newdata = synth_size)
```

Step 3: Aggregate to state level using post-stratification target distribution

```
df_geo <- sum_to_geo(pred_cell, geovar = state)
```

regression as long as it predicts the outcome well. We estimate a logit model with random effects using the estimator described in Section 3. A demographic target distribution for post-stratification must be prepared by the analyst as well. This task becomes data intensive as model complexity increases, because the target distribution must estimate joint probabilities for every variable in the regression. We provide a convenience function that easily aggregates a census-type dataset to the desired cell distribution. We also provide a convenience function to incorporate additional marginal distributions and extend these distributions into a synthetic distribution using the simplifying independence assumption (Leemann and Wasserfallen, 2017). Once the individual-level regression model is fit, the analyst can generate post-stratified estimates by generating out of sample predictions for each of the demographic cells in the target distribution. Because each cell is associated with its relative frequency within each state, the MRP estimation can be thought of as a weighted mean of regression predicted values with the relative prevalence of each cell in a particular state as weights.

Synthetic Post-stratification Post-stratification targets are based on the 5-percent sample of the 2010 Census, provided in Ghitza and Gelman (2013). However, the Census does not ask a resident's religion. In order to make advantage of the added predicted power of accounting for religion at the regression stage, we construct a synthetic post-stratification target distribution as described in Section 3.

For estimates of the marginal religiosity at the state-level, we gather data from the 2007 Pew Religious Landscape Survey. The main survey was conducted in the 48 continental states and DC, largely by random digit dialing with a small cellphone supplement, and collected over 35,000 responses on detailed questions about a voter's religious affiliation. We then categorized religion into three categories to mirror the main survey data, collapsing some small categories[34], computing their estimated fraction within each state by using Pew's survey weights. To create a synthetic joint distribution, we apply the assumption that the distribution of religion is independent with the other 5 variables: for example, if we estimate that 62 percent of the adult population of Massachusetts is mainline protestant or Catholic, then all demographic cells (e.g. middle-aged white women with a college degree, middle-aged white men with a college-degree) are also assumed to be 62 percent mainline Protestant or Catholic. This description highlights two sources of potential bias introduced. First is that the original dataset, while large, is a survey nonetheless, and selection bias may not cancel out (Meng, 2018). Another is that we make a simplifying approximation that marginal distributions of religion are independent from the other joint distributions. Despite these potential biases, improvements from adding a relevant variable and its interactions may, or may not, be able to offset that bias.

5.2 Estimation Results

We evaluate our estimation in two stages: first at the multilevel model as we have done with simulated data in Section 4, and next at the aggregated state level after post-stratification. In the post-stratification stage, we also attempt include new predictors that are likely predictive of the outcome but are not part of the post-stratification target. For example, neither Ghitza and Gelman (2013) nor Montgomery and Olivella (2018) modelled religion due to limitations in the post-stratification dataset, but we attempt include this through our synthetic approach.

[34] One category for {Born-again Protestant, Mormon}, another for {Mainline Protestant, Catholic}, and another for {Jewish, other religion, and non-religious}

Deviance Does a more complex fitted model improve accuracy, or do more covariates induce overfitting? In Figure 4 we show how our model's prediction improves as model complexity increases. We consider three sets of variables in addition to random effects for state, which is always included: {age, income, race}; {age, income, race, sex, education, marital status}, and {age, income, race, sex, education, marital status, religion}. All variables are categorical, and each contains 2 to 6 levels. For each set of these variables, we further enter them into the regression in one of three ways: without any interactions, with all pairwise combinations as interactions, and with all triple sets as interactions. Together, we consider nine specifications whose number of considered coefficients range from 10 to 679. We estimate these models on a non-missing subset of replication data (n = 16,722). To assess both in-sample and out-of-sample fit, the full sample is split in half and the model is fit on one half.

The Figure shows that using the proposed method, model fit improves (i.e., deviance decreases) with model complexity. Both the number of covariates and their interactions matter. Both in-sample fit and out-of-sample fit improve, while in-sample fit improves more clearly. Even with close to 1,000 coefficients, the proposed method indicates positive returns with relatively fast computation time.

State-level Estimates with sMRP An accurate individual-level model is only the first part of Multi-level Regression Poststratification. Once a regression model is estimated, these must be aggregated to the geographic level of interest, weighting each demographic cell according to their fraction in each geography. How accurate are the state-level estimates that the models estimated in Figure 4?[35]

We re-estimate each model with the full sample of $n = 16,722$, then aggregate up to the state level using post-stratification target. As an approximate ground truth, we use McCain's vote share on election day, and compute the root mean squared error (RMSE)[36]. In Figure 5 we show the RMSE associated with estimates of the 48 continental states[37].

[35] Before comparing predicted MRP estimates with observed data, it is worth emphasizing that unlike our simulations, no ground truth exists in our application. While election results are population values, these measure a candidate's support among the population that turned out to vote. Here we estimate state-level support with pre-election survey data using a binary variable, effectively assuming that everyone in the survey sample votes, and vote intention does not change between the poll and election day.

[36] For states $s = 1, ..., 48$, and estimates of McCain vote share $\hat{\mu}_s$ corresponding to truth μ_s, RMSE $= \sqrt{\frac{1}{48}\sum_{s=1}^{48}(\mu_s - \hat{\mu}_s)^2}$

[37] We omit DC because it is an outlier in terms of democratic vote share, and omit Alaska and Hawaii because

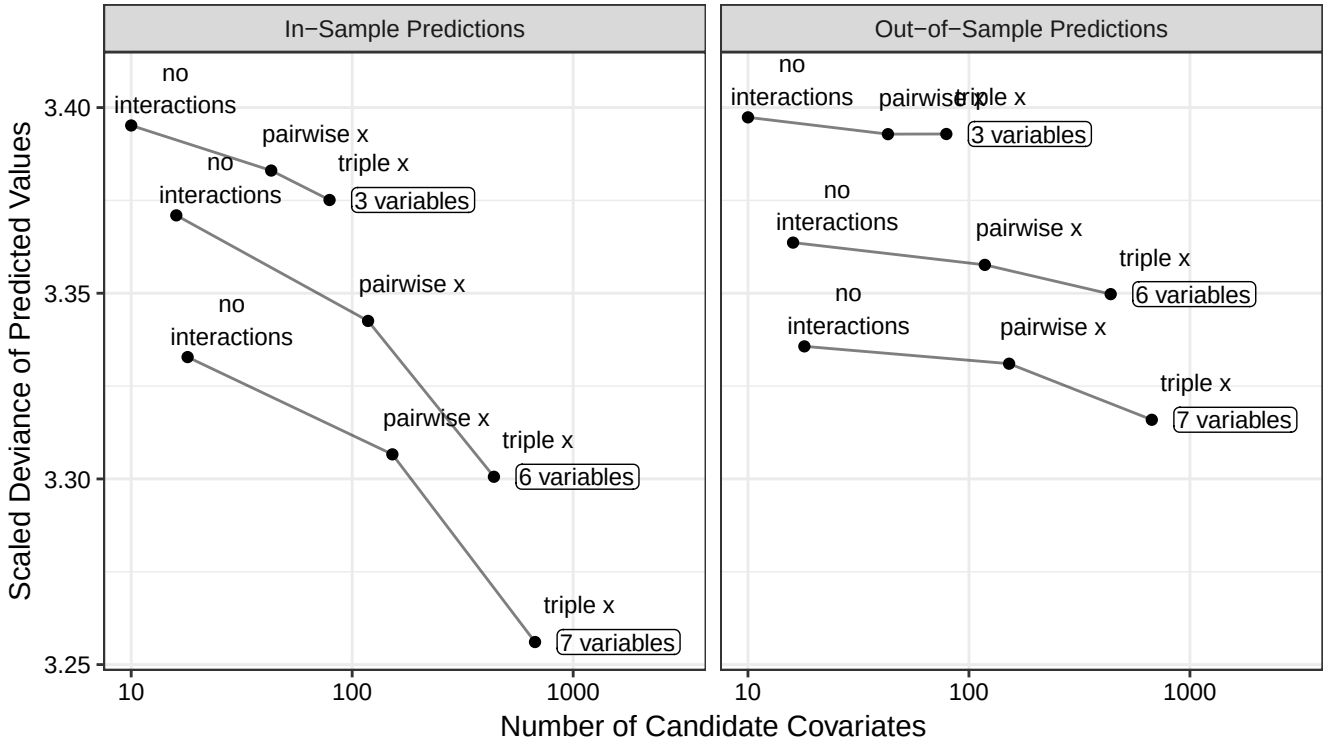

Figure 4: **Improvement in model prediction as number of coefficients increase.** Each point is a model specification, plotted by their mode complexity (on the log scale) and deviance of fitted values. Deviance is computed as in equation 16 and divided by the sample size ($n = 8,361$). All models were estimated from a random half of the the same replication dataset from Ghitza and Gelman (2013), generating both in-sample and out-of-sample (the remaining half) predictions. "3 variables" estimate fixed effects for age, income, and race. "6 variables" adds sex, education, and marital status, and "7 variables" further add religion. All models include random effects for state. "no interactions" indicates that these variables are entered without any interaction, "pairwise x" indicates that all pairwise interactions between the variables considered were entered, and "triple x" indicates that all triple interactions between the variables were entered.

Figure 5 is set up in the same way as Figure 4 but with 48-state RMSE as the outcome variable. Smaller RMSE indicates a closer fit between estimates and the final electin outcome. For comparison, Ghitza and Gelman's GG-simple and GG-full models are re-estimated using the same subset of the Pew survey, and their RMSEs from their unadjusted estimates are shown in dotted lines.[38]

phone polling is significantly unreliable in these two states. Moreover, most of the Pew political surveys in the replication data as well as the Pew religious landscape survey we use for synthetic post-stratification do not poll Alaska and Hawaii in their main sampling frame mainly for this reason. These three geographies are often omitted as outliers, for example in Ghitza and Gelman (2013) and Kiewiet De Jonge, Langer and Sinozich (2018).

[38] Because their main goal is inference on small subgroups instead of model validation, Ghitza and Gelman (2013) re-adjust their `lmer` estimates after model estimation so that state-level estimates match almost perfectly with the state-level McCain vote share (769). For comparisons of model validation, we recompute the MRP estimates before adjustment.

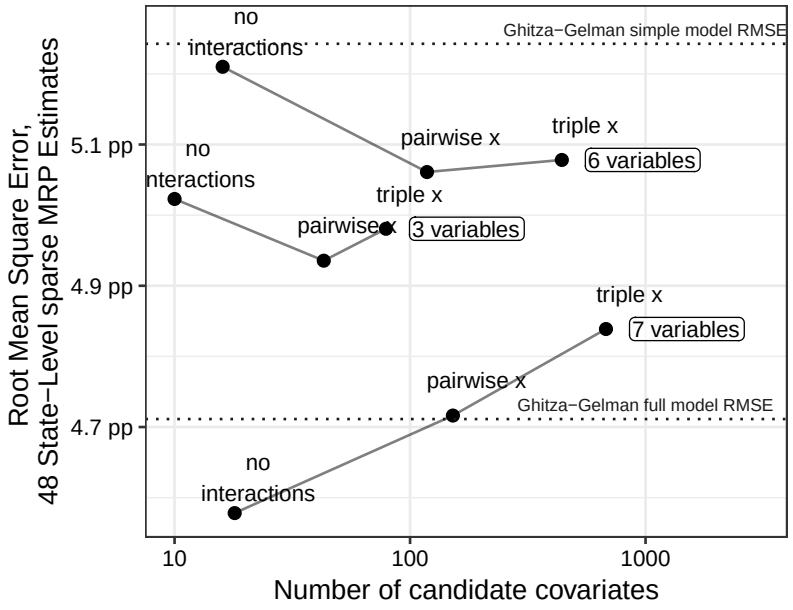

Figure 5: **Change in RMSE of state-level estimates.** The RMSE associated with the post-stratified state-level estimates from the models in Figure 4, recomputed on the full sample. The RMSEs from Ghitza and Gelman (2013)'s two models are indicated with dotted lines. For comparison, RMSEs from raw poll averages are 5.9 percentage points without weights and 6.6 percentage points with weights.

MRP estimates using 7 variables has the lowest RMSE, although more variables do not necessarily correspond to better RMSE: The simplest model of 3 fixed effects maintains a lower RMSE than one with 6 variables. `sparsereg`'s RMSEs are generally competitive with those from Ghitza and Gelman (2013), and substantially faster to estimate.

The pattern of the 7 variables case is interesting to note because the inclusion of the seventh variable, religion, necessitates extrapolating the post-stratification target distribution using the synthetic method. If either the new dataset computing state-level marginal proportions of religion *or* the independence assumption contains any bias with respect to the true joint distribution in the population, these biases may offset the improvement in individual-level fit shown in Figure 4.

We also note that differences in performance among these models is substantially not large. A difference in RMSE of 0.2 or 0.4 percentage points can be caused by a negligible shifts in certain states with no consistent pattern. To visually inspect the difference between estimates, we present the state-level estimates of the models with the lowest and highest complexity in Figure 6.

In sum, estimation results from Ghitza and Gelman (2013) highlights several important lessons for applied researchers estimating MRP models. First, the variables on the order of 100 or 1000 can

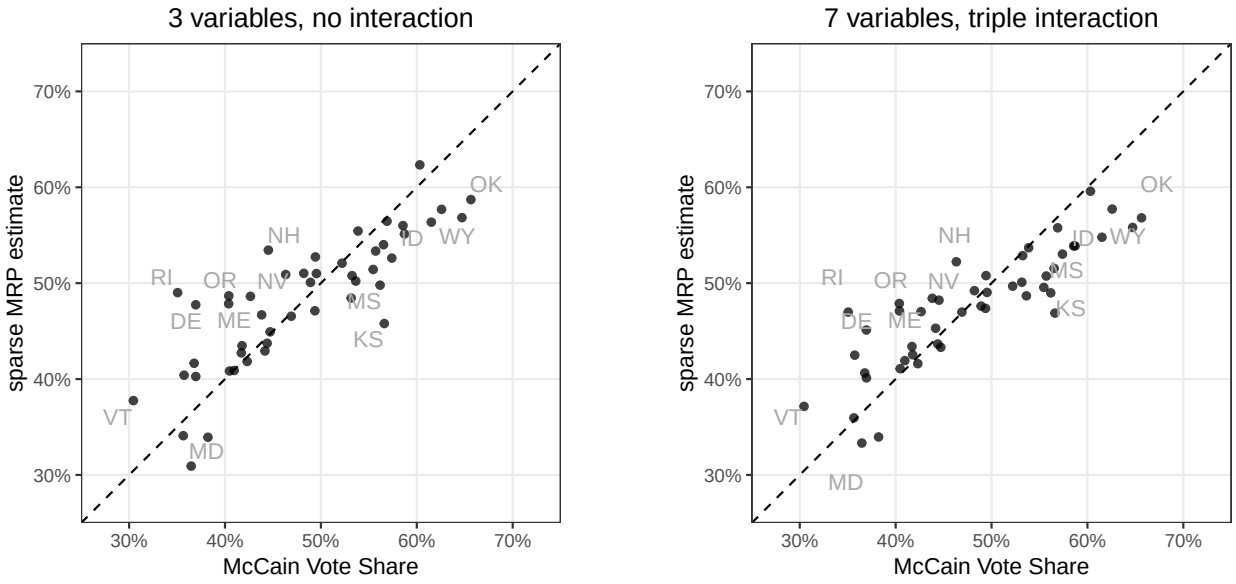

Figure 6: **A visual inspection of state-level MRP estimates**. The state-level estimates for the model with the smallest number of covariates (left) and the largest umber of covariates (right) are shown along the election outcome. States where the difference between the estimate and outcome is larger than 5 percentage points are labelled. Model names, indicated in the title, correspond to those in Figure 5.

be included in the first stage regression model if appropriate regularization is applied. Second, even variables whose joint distribution in the population is unknown can be included in the regression model and extrapolated by a synthetic joint distribution, which can be beneficial as long as the improvement in model fit from the new variable offset any new bias that the synthetic estimate introduces. Third, while better models at the regression stage certainly help the estimation of geographic-level estimates, the gains may be less than proportionally large.

6 Conclusion

This paper provides a framework for estimating sparse multilevel models. Our frameworks accommodates both the standard linear model but also draws on recent computational tools to extend to the logit case. Our data augmentation approach for the logit model produces computational gains, making it faster to estimate non-linear multilevel models, especially as the number of parameters grows. Our approach can be combined with a stratification step to produce a highly competitive methodology for researchers using multilevel regression and post-stratification (MRP).

We are currently working to extend the model in several ways. We believe we have a complete

Gibbs version done, but we are still testing it. Within the current case of linear and logit models, there is the natural extension to allow for random slopes rather than random intercepts. A second path for extension is to utilize Polya-Gamma augmentation to handle the case of a multinomial model. Multinomial models have come up in the MRP literature (Lauderdale et al., 2017), and we can provide a tractable way to estimate these models. Similar extensions for count data could be made as well (Hanretty, 2017). Of course, given that we are operating in a Bayesian framework we could build out more complicated prior structures. For example, following Si et al. (2017), one could modify the prior structure to ensure that higher-order interactions are more aggressively penalized. Additionally, if researchers or practitioners have informative priors for certain variables this could be built in.

We are also investigating different ways to conduct the post-stratification step given that there might be many variables predicting the outcome of interest (which becomes more feasible to explore with our approach). We suspect there might be fertile engagement with other literatures such as work on ecological inference (King, Tanner and Rosen, 2004; Flaxman, Wang and Smola, 2015; Flaxman et al., 2016; Rosenman and Viswanathan, 2018) to push the usefulness of post-stratification steps. .

References

Armagan, Artin and David Dunson. 2011. "Sparse variational analysis of linear mixed models for large data sets." *Statistics & probability letters* 81(8):1056–1062.

Bates, D and D Sarkar. 2008. "The lme4 Package, 2006." *URL http://cran. r-project. org* .

Bondell, Howard D, Arun Krishna and Sujit K Ghosh. 2010. "Joint Variable Selection for Fixed and Random Effects in Linear Mixed-Effects Models." *Biometrics* 66(4):1069–1077.

Breidt, F Jay and Jean D Opsomer. 2008. "Endogenous post-stratification in surveys: classifying with a sample-fitted model." *The annals of statistics* pp. 403–427.

Carpenter, Bob, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li and Allen Riddell. 2017. "Stan: A probabilistic programming language." *Journal of statistical software* 76(1).

Carvalho, C, N Polson and J Scott. 2010. "The Horseshoe Estimator for Sparse Signals." *Biometrika* 97:465–480.

Caughey, Devin and Erin Hartmann. 2016. "Target Selection as Variable Selection: Using the Lasso to Select Auxiliary Vectors for the Construction of Survey Weights." *working paper* .

Dempster, Arthur P, Nan M Laird and Donald B Rubin. 1977. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the royal statistical society. Series B (methodological)* pp. 1–38.

Fan, Yingying and Runze Li. 2012. "Variable selection in linear mixed effects models." *Annals of statistics* 40(4):2043.

Flaxman, Seth, Dougal Sutherland, Yu-Xiang Wang and Yee Whye Teh. 2016. "Understanding the 2016 US Presidential Election using ecological inference and distribution regression with census microdata." *arXiv preprint arXiv:1611.03787* .

Flaxman, Seth R, Yu-Xiang Wang and Alexander J Smola. 2015. Who supported Obama in 2012?: Ecological inference through distribution regression. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM pp. 289–298.

Gelman, Andrew. 2006*a*. "Multilevel (hierarchical) modeling: what it can and cannot do." *Technometrics* 48(3):432–435.

Gelman, Andrew. 2006*b*. "Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)." *Bayesian Analysis* 1(3):515–534.

Gelman, Andrew, Boris Shor, Joseph Bafumi and David Park. 2007. "Rich State, Poor State, Red State, Blue State: What's the Matter with Connecticut?" *Quarterly Journal of Political Science* 2(4):345–367.

Gelman, Andrew and Jennifer Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

Gelman, Andrew and Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge: Cambrdige University Press.

Gelman, Andrew et al. 2005. "Analysis of varianceâwhy it is more important than ever." *The annals of statistics* 33(1):1–53.

Ghitza, Yair and Andrew Gelman. 2013. "Deep interactions with MRP: Election turnout and voting patterns among small electoral subgroups." *American Journal of Political Science* 57(3):762–776.

Goplerud, Max. 2018. "A Multinomial Framework for Ideal Point Estimation." *Political Analysis* Forthcoming.

Goplerud, Max, Shiro Kuriwaki, Marc Ratkovic and Dustin Tingley. 2018. "Generalized Sparse Modelling." *working paper* .

Groll, Andreas. 2017. *glmmLasso: Variable Selection for Generalized Linear Mixed Models by L1-Penalized Estimation*. R package version 1.5.1.
URL: *https://CRAN.R-project.org/package=glmmLasso*

Groll, Andreas and Gerhard Tutz. 2014. "Variable selection for generalized linear mixed models by L 1-penalized estimation." *Statistics and Computing* 24(2):137–154.

Hanretty, Chris. 2017. "Areal interpolation and the UK's referendum on EU membership." *Journal of Elections, Public Opinion and Parties* 27(4):466–483.

Hastie, Trevor, Robert Tibshirani and Jerome Friedman. 2010. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

Howe, Peter D, Matto Mildenberger, Jennifer R Marlon and Anthony Leiserowitz. 2015. "Geographic variation in opinions on climate change at state and local scales in the USA." *Nature Climate Change* 5(6):596.

Hox, Joop J, Mirjam Moerbeek and Rens van de Schoot. 2017. *Multilevel analysis: Techniques and applications*. Routledge.

Ibrahim, Joseph G, Hongtu Zhu, Ramon I Garcia and Ruixin Guo. 2011. "Fixed and random effects selection in mixed effects models." *Biometrics* 67(2):495–503.

Jianfei, Chen, Jun Zhu, Zi Wang, Xun Zheng and Bo Zhang. 2013. "Scaleable Inference for Logistic-Normal Topic Models." *Advances in Neural Information Processing Systems* pp. 2445–2453.

Kastellec, Jonathan P, Jeffrey R Lax and Justin H Phillips. 2010. "Public opinion and Senate confirmation of Supreme Court nominees." *The Journal of Politics* 72(3):767–784.

Kiewiet De Jonge, Chad P, Gary Langer and Sofi Sinozich. 2018. "Predicting State Presidential Election Results Using National Tracking Polls and Multilevel Regression With Poststratification (MRP)." *Public Opinion Quarterly* (July).

King, Gary, Martin A Tanner and Ori Rosen. 2004. *Ecological inference: New methodological strategies*. Cambridge University Press.

Kinney, Satkartar K and David B Dunson. 2007. "Fixed and random effects selection in linear and logistic models." *Biometrics* 63(3):690–698.

Laird, Nan M and James H Ware. 1982. "Random-effects models for longitudinal data." *Biometrics* pp. 963–974.

Lauderdale, Benjamin E, Delia Bailey, Jack Blumenau and Douglass Rivers. 2017. "Model-Based Pre-Election Polling for National and Sub-National Outcomes in the US and UK." *Working paper* .

Lax, Jeffrey R and Justin H Phillips. 2009*a*. "Gay rights in the states: Public opinion and policy responsiveness." *American Political Science Review* 103(3):367–386.

Lax, Jeffrey R and Justin H Phillips. 2009*b*. "How should we estimate public opinion in the states?" *American Journal of Political Science* 53(1):107–121.

Lax, Jeffrey R and Justin H Phillips. 2012. "The democratic deficit in the states." *American Journal of Political Science* 56(1):148–166.

Leemann, Lucas and Fabio Wasserfallen. 2017. "Extending the use and prediction precision of subnational public opinion estimation." *American Journal of Political Science* 61(4):1003–1022.

Makalic, Enes and Daniel F. Schmdit. 2016. "A Simple Sampler for the Horseshoe Estimator." *IEEE Signal Processing Letters* 23(1):179–182.

Meng, Xiao-Li. 2018. "Statistical paradises and paradoxes in Big Data (I): Law of large populations, Big Data paradox, and 2016 US presidential election." *Annals of Applied Statistics* .

Meng, Xiao-Li and David Van Dyk. 1997. "The EM Algorithmâan Old Folk-song Sung to a Fast New Tune." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59(3):511–567.

Meng, Xiao-Li and David Van Dyk. 1998. "Fast EM-type implementations for mixed effects models." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 60(3):559–578.

Mildenberger, Matto, Peter Howe, Erick Lachapelle, Leah Stokes, Jennifer Marlon and Timothy Gravelle. 2016. "The distribution of climate change public opinion in Canada." *PloS one* 11(8):e0159774.

Montgomery, Jacob M and Santiago Olivella. 2018. "Tree-based models for political science data." *American Journal of Political Science* .

Ornstein, Joseph T. 2017. "Subnational Public Opinion Estimation Using MrsP.".

Pan, Jianxin and Chao Huang. 2014. "Random effects selection in generalized linear mixed models via shrinkage penalty function." *Statistics and Computing* 24(5):725–738.

Park, David K, Andrew Gelman and Joseph Bafumi. 2004. "Bayesian multilevel estimation with poststratification: state-level estimates from national polls." *Political Analysis* 12(4):375–385.

Park, David K, Andrew Gelman and Joseph Bafumi. 2006. "State-level opinions from national surveys: Poststratification using multilevel logistic regression." *Public opinion in state politics* pp. 209–28.

Park, Trevor and George Casella. 2008. "The bayesian lasso." *Journal of the American Statistical Association* 103(482):681–686.

Polson, Nicholas G, James G Scott and Jesse Windle. 2013. "Bayesian inference for logistic models using Pólya–Gamma latent variables." *Journal of the American statistical Association* 108(504):1339–1349.

Ratkovic, Marc and Dustin Tingley. 2017. "Sparse Estimation and Uncertainty with Application to Subgroup Analysis." *Political Analysis* 1(25):1–40.

Rosenman, Evan and Nitin Viswanathan. 2018. "Using Poisson Binomial GLMs to Reveal Voter Preferences." *arXiv preprint arXiv:1802.01053* .

Si, Yajuan, Rob Trangucci, Jonah Sol Gabry and Andrew Gelman. 2017. "Bayesian hierarchical weighting adjustment and survey inference." *arXiv preprint arXiv:1707.08220* .

Singer, Judith and BJ Willett. 2003. *Applied longitudinal data analysis: modeling change and event occurrence*. Oxford University Press.

Tanner, Martin A. and Wing Hung Wong. 1987. "The Calculation of Posterior Distributions by Data Augmentation." *Journal of the American Statistical Association* 82:528–540.

Tausanovitch, Chris and Christopher Warshaw. 2013. "Measuring constituent policy preferences in congress, state legislatures, and cities." *The Journal of Politics* 75(2):330–342.

Tausanovitch, Chris and Christopher Warshaw. 2014. "Representation in municipal government." *American Political Science Review* 108(3):605–641.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society, Series B.* 58:267–88.

Wang, Wei, David Rothschild, Sharad Goel and Andrew Gelman. 2015. "Forecasting elections with non-representative polls." *International Journal of Forecasting* 31(3):980–991.

Warshaw, Christopher and Jonathan Rodden. 2012. "How should we measure district-level public opinion on individual issues?" *The Journal of Politics* 74(1):203–219.

Wei, Greg C.G. and Martin A. Tanner. 1990. "A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms." *Journal of the American Statistical Association* 85(411):699–704.

Zhang, Xingyou, James B Holt, Hua Lu, Anne G Wheaton, Earl S Ford, Kurt J Greenlund and Janet B Croft. 2014. "Multilevel regression and poststratification for small-area estimation of population health outcomes: a case study of chronic obstructive pulmonary disease prevalence using the behavioral risk factor surveillance system." *American journal of epidemiology* 179(8):1025–1033.

Zhang, Xingyou, James B Holt, Shumei Yun, Hua Lu, Kurt J Greenlund and Janet B Croft. 2015. "Validation of multilevel regression and poststratification methodology for small area estimation of health indicators from the behavioral risk factor surveillance system." *American journal of epidemiology* 182(2):127–137.

Zou, Hui. 2006. "The Adaptive Lasso and Its Oracle Properties." *Journal of the American Statistical Association* 101(476):1418–1429.

Online Appendix

A Survey Weights

B Gibbs Sampling with Data Augmentation

In this Appendix, we do not fully derive the results for the sparse model as these are extensively documented in Ratkovic and Tingley (2017). Rather, we outline the steps needed to augment the data such that one can then do one 'sparsereg update' by which we mean, draw all of the variables from their full conditionals as outlined in Ratkovic and Tingley (2017). The results here are done for a single random effect; this can be generalized easily to multiple random effects either by the way outlined in C or by sampling them sequentially as is standard in a Gibbs Sampler.

1. For each observation i, draw one Polya-Gamma random variable with the following form:

$$\omega_i \sim PG(\delta_i, \boldsymbol{x}_i^T\boldsymbol{\beta} + \alpha_{j[i]})$$

   That is, draw one Polya-Gamma random variable where the shape parameter is the survey weight δ_i–one in the unweighted case–and the scale paramter is the linear predictor, i.e. the fixed effects plus all random effects. We would encourage researchers to standardize the weights such that the maximum value is $\delta_i = 1$ as this speeds the sampler for the Polya-Gamma random variables.

2. Second, recall from the Polya-Gamma augmentation noted above that we can write the complete data log-likelihood as follows:

$$\ln p(y_i|\omega_i, \boldsymbol{\beta}, \alpha_j) \propto \sum_i \delta_i(y_i - 1/2)(\boldsymbol{x}_i^T\boldsymbol{\beta} + \alpha_{j[i]}) - \omega_i(\boldsymbol{x}_i^T\boldsymbol{\beta} + \alpha_{j[i]})^2/2$$

   As expected, this looks very similar to the complete data log-likelihood for a linear regression and can be turned into exactly that. Define the Polya-Gamma weighted outcome as $\tilde{y}_i$ and the Polya-Gamma weighted covariates as $\tilde{x}_i$. By distributed through the ω_i, we can write the above as

$$\ln p(y_i|\omega_i, \boldsymbol{\beta}, \alpha_j) \propto \sum_i \tilde{y}_i(\tilde{\boldsymbol{x}}_i^T\boldsymbol{\beta} + \sqrt{\omega_i}\alpha_{j[i]}) - (\tilde{\boldsymbol{x}}_i^T\boldsymbol{\beta} + \tilde{z}_i\alpha_{j[i]})^2/2$$

$$\tilde{y}_i = \frac{\delta_i(y_i - 1/2)}{\sqrt{\omega_i}}; \quad \tilde{x}_i = \sqrt{\omega_i}\boldsymbol{x}_i$$

With further re-arrangement and ignoring all terms that do not depend on $\boldsymbol{\beta}$ or α_j,

$$\ln p(y_i|\omega_i, \boldsymbol{\beta}, \alpha_j) \propto -1/2\left(\tilde{y}_i - [\tilde{\boldsymbol{x}}_i^T\boldsymbol{\beta} + \sqrt{\omega_i}\alpha_{j[i]}]\right)^2$$

This is the kernel of a normal likelihood. Thus, one can simply perform the linear sparsereg updates where the outcome is the weighted $\tilde{y}_i$ and the covariates are the $\tilde{\boldsymbol{x}}_i$. The random effect is also weighted by $\sqrt{\omega_i}$.

3. Following Ratkovic and Tingley (2017), do one sparsereg update, i.e. update the $\boldsymbol{\beta}$, the regularization parameters (λ, γ) as well as the auxiliary variables needed for those updates.

4. The random effects α_j can be sampled from the corresponding normal distribution, where $\sum_{i:j[i]=j}$ is notation for summing over all of the i in group j:

$$\alpha_j \sim N\left(\frac{\sum_{i:j[i]=j}\left(\tilde{y}_i - \tilde{\boldsymbol{x}}_i^T\boldsymbol{\beta}\right)\sqrt{\omega_i}}{\sigma_\alpha^2 + \sum_{i:j[i]=j}\omega_i}, \frac{1}{\sigma_\alpha^2 + \sum_{i:j[i]=j}\omega_i}\right)$$

5. σ_α^2 can be sampled from its posterior distribution. All of the usual conjugacies apply. For example, if we place an inverse-gamma prior $IG(a_\alpha, b_\alpha)$ on σ_α^2, the posterior is inverse-gamma by standard conjugacy rules, where N_j is the number of groups (levels) of j.

$$\sigma_\alpha^2 \sim IG\left(\frac{N_j}{2} + a_\alpha, \quad \frac{1}{2}\sum_j \alpha_j^2 + b_\alpha\right)$$

Other choices for the variance, e.g. the half-Cauchy (Gelman, 2006*b*), can be implemented and sampled in the standard way.

C EM Estimation with Data Augmentation

The EM algorithm used in the paper exploits the following result: We wish to optimize the posterior on $p(\boldsymbol{\beta}, \sigma_\alpha^2 | \boldsymbol{y}, \boldsymbol{X})$ having integrated out the random effets α_j. That is, recalling Equation 17 from above where p_{SR} denotes the LassoPLUS prior described above.

$$p(\boldsymbol{\beta}, \sigma_\alpha^2 | \boldsymbol{X}, \boldsymbol{y}) \propto p(\sigma_\alpha^2) p_{SR}(\boldsymbol{\beta}) \int \prod_i \left[\frac{\exp(\boldsymbol{x}_i^T \boldsymbol{\beta} + \alpha_{j[i]})^{y_i}}{1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta} + \alpha_{j[i]})}; \right]^{\delta_i} \prod_j \phi(\alpha_j; 0, \sigma_\alpha^2) d\alpha_j \tag{17}$$

However, this is intractable. If we augment with the Polya-Gammas, we can write the posterior as follows:

$$p(\boldsymbol{\beta}, \sigma_\alpha^2 | \boldsymbol{X}, \boldsymbol{y}) \propto p(\sigma_\alpha^2) p_{SR}(\boldsymbol{\beta}) \cdot \int_\alpha \prod_i \left[\int_\omega \exp(\delta_i [y_i - 1/2/2](\boldsymbol{x}_i^T \boldsymbol{\beta} + \alpha_{j[i]}) - \omega_i (\boldsymbol{x}_i^T \boldsymbol{\beta} + \alpha_{j[i]})^2/2) f(\omega_i | \delta_i, 0) d\omega_i \right] \prod_j \phi(\alpha_j; 0, \sigma_\alpha^2) d\alpha_j \tag{18}$$

Flipping the order of integration, we can write the augmented posterior as:

$$p(\boldsymbol{\beta}, \sigma_\alpha^2 | \boldsymbol{X}, \boldsymbol{y}) \propto p(\sigma_\alpha^2) p_{SR}(\boldsymbol{\beta}) \cdot \int_\omega \left[\int_\alpha \prod_i \exp(\delta_i [y_i - 1/2/2](\boldsymbol{x}_i^T \boldsymbol{\beta} + \alpha_{j[i]}) - \omega_i (\boldsymbol{x}_i^T \boldsymbol{\beta} + \alpha_{j[i]})^2/2) \prod_j \phi(\alpha_j; 0, \sigma_\alpha^2) d\alpha_j \right] \prod_i f(\omega_i | \delta_i, 0) d\omega_i \tag{19}$$

We can re-arrange the inner integral to resemble a linear-mixed effects model—the exact linear mixed effects model used in the Gibbs Sampler. That is, if the ω_i were known, we have a linear mixed effect model model as our complete data likelihood. As outlined below, the EM algorithm can be used on *this* 'inner' complete data likelihood to deterministically increase it. Thus, conditional on the ω_i, our procedure can be thought of as a generalized EM algorithm (Dempster, Laird and Rubin, 1977). One could perform multiple EM-steps on the linear mixed effect component, although we found that in practice this did not speed convergence. Note further that since the inner M-Step will, eventually, maximize the complete data (log)-likelihood, this should inherit the desirable properties of the EM algorithm.

The one issue with this procedure, however, is the fact that calculating the expectation of ω_i depends itself on the random effect as the Polya-Gamma augmentation above shows. We tried a

variety of methods to address this but found that a solution in the vein of an "EM-type" algorithm as noted in Wei and Tanner (1990, p. 700): Specifically, they suggest that we pick some plausible, high posterior density, distribution of ω_i as an approximate E-Step. In short, we calculate the expectation of ω_i using the *past* best guess at the mean of the random effect α_j.

Somewhat more formally, our E-Step requires knowing the distribution of $p(\{\omega_i\}, \{\alpha_j\}|\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{y})$. As the above re-arrangement shows: $p(\alpha_j|\{\omega_i\}, \boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{y})$ is tractable (normal), but $p(\omega_i|\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{y})$ is not. Thus, we approximate this second density as Polya-Gamma around a lagged value of $\boldsymbol{\alpha}$, i.e. $p(\omega_i|\boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{y}) \sim PG(\delta_i, \boldsymbol{x}_i^T\boldsymbol{\beta} + \alpha_j^{(t-1)})$. Thus, we avoid an integral but use a value of α_j that is lagged and also ignore its variability that would feed into the expectation of ω_i.

While this method performs extremely well on the simulations reported in the paper and others—in a simple case, it gives extremely similar results to `glmer` for plausibly sized random effects, we note that it does require an approximation and thus should be used either to get a sense of how well, roughly, the algorithm will perform and then be used as starting values for a Gibbs Sampler. As this likely puts us very near the posterior mode, the Gibbs Sampler can likely be run for a short time to approximate convergence and then inference can be conducted. Other papers that apply regularization to logistic random effects framework rely on a Monte Carlo approximation to the E-Step (Ibrahim et al., 2011), a gradient ascent algorithm (Groll and Tutz, 2014), or a variational approach (Armagan and Dunson, 2011). We found that the first approach does not scale competitive versus a Gibbs Sampler. We also note that the second uses a number of approximations to the posterior and relies on an optimization method that, could, be unstable depending on the starting values. The third is clearly not guaranteed to optimize the posterior as it is a variational method. Thus, we note that while our method has approximate E-Step, we note that all other updates are *exact* and inherit guarantees on increasing the actual posterior. As noted above, without random effects, our procedure is *exact*. Thus, if one wanted an EM algorithm that was guaranteed convergence to the posterior mode, one could include the random effects as 'fixed effects' and apply regularization to them directly or simply place a ridge-type prior to prevent overfitting.

The full algorithm is outlined below; the relevant sparsereg EM updates are located in the online appendix for Ratkovic and Tingley (2017).

- Initalize the algorithm with some $\boldsymbol{\beta}^{(0)}$, $[\sigma_\alpha^2]^{(0)}$ and mean of the random effects, $\boldsymbol{\mu}_\alpha$. We suggest $\boldsymbol{\mu}_\alpha = \mathbf{0}$.

- For $t \in \{1, \cdots, T\}$ iterations:

  - (Approximate) Outer E-Step: Calculate the expectation of the ω_i given the fixed effect $\boldsymbol{x}_i^T \boldsymbol{\beta}$ and the posterior mean of the random effect $E[\alpha_{j[i]}|-]$.

$$\omega_i^* = E[\omega_i | y_i, \boldsymbol{x}_i] = \frac{1}{2\left(\boldsymbol{x}_i^T \boldsymbol{\beta}^{(t-1)} + [\boldsymbol{\mu}_\alpha^{(t-1)}]_{j[i]}\right)} \tanh\left(\left(\boldsymbol{x}_i^T \boldsymbol{\beta}^{(t-1)} + [\boldsymbol{\mu}_\alpha^{(t-1)}]_{j[i]}\right)/2\right) \quad (20)$$

  - EM Algorithm For the Complete Data Likelihood:

    * Inner E-Step: Using the ω_i^* calculated above, the distribution of the vector of stacked random effects $\boldsymbol{\alpha}$ can be written as follows (Meng and Van Dyk, 1998). Note that its prior variance is written as $N(0, \sigma_\alpha^2 \boldsymbol{I})$ where $\boldsymbol{I}$ is the identity matrix with dimensionality equal to the number of unique values of the random effects.

      To be clear, this means that $\boldsymbol{\alpha}$ has a dimensionality of $P = \sum_k N_k$ where N_k is the number of levels of each random effect. For example, if we had two random effects (state and education), we would have $\boldsymbol{\alpha}$ as a 54 (50 + 4) length vector. For simplicity, we stack all levels of random effect together, i.e. all fifty states, all four levels of education. We do this for the following reason. If the random effects are crossed (which they are in most social science applications), we can define a (sparse) $\boldsymbol{z}_i$ that is a P length vector with mostly zeros but a '1' corresponding to each random effect that applies to an observation. In the case of state and education random effects, two elements of $\boldsymbol{z}_i$ would equal one for each observation. Thus $\boldsymbol{z}_i^T \boldsymbol{\alpha}$ pulls out the sum of the random effects for each unit. As the prior for $\boldsymbol{\alpha}$ encodes an assumption that random effects are independent, it must be a diagonal matrix with the variance of each random effect on the corresponding diagonal. Thus, the first 50 elements would be σ_α^2, the final four elements would be $\sigma_{education}^2$. Define this 'stacked' distributional assumption for the random effect as $\boldsymbol{T}$. It, however, only consists of K distinct parameters where K is the number of random effects.

      This set up allows for easily dealing with the types of random effects that occur in MRP settings, although it is worth noting that this step will require the inversion of $P \times P$ matrices. This is usually fast; for an extremely large number of random effects, the code should be adapted to rely on various inversion tricks for a large

matrix (e.g. Woodbury). Note that $\boldsymbol{\Sigma}_\alpha^{(t)}$ and $\boldsymbol{T}$ is only ever used in update σ_α^2 and thus does not need to be stored. Further, $\boldsymbol{Z}$ and $\boldsymbol{T}$ are highly sparse and thus can be stored easily using sparse matrix representations that both speed computation and ensure scalability.

Given this set up, the inner E-step for $\boldsymbol{\alpha}^{(t)}$ can be written by noting its mean and variance, following Meng and Van Dyk (1998). Define $\tilde{\boldsymbol{z}}_i^* = \sqrt{\omega_i^*}\boldsymbol{z}_i$ and $\tilde{\boldsymbol{Z}}$ stacks the scaled data vertically.

$$\boldsymbol{\alpha} \sim N\left([\boldsymbol{\mu}_\alpha]^{(t)}, \boldsymbol{\Sigma}_\alpha^{(t)}\right) \tag{21a}$$

$$\boldsymbol{W} = \left(\boldsymbol{I} + \tilde{\boldsymbol{Z}}\boldsymbol{T}^{(t-1)}\tilde{\boldsymbol{Z}}^T\right)^{-1} \tag{21b}$$

$$\boldsymbol{\Sigma}_\alpha^{(t)} = \boldsymbol{T}^{(t-1)} - \boldsymbol{T}^{(t-1)}\tilde{\boldsymbol{Z}}^T\boldsymbol{W}\tilde{\boldsymbol{Z}}\boldsymbol{T}^{(t-1)} \tag{21c}$$

$$\boldsymbol{\mu}_\alpha^{(t)} = \boldsymbol{T}^{(t-1)}\tilde{\boldsymbol{Z}}^T\boldsymbol{W}\left(\tilde{\boldsymbol{s}} - \tilde{\boldsymbol{X}}\boldsymbol{\beta}^{(t-1)}\right) \tag{21d}$$

If the model includes multiple random effects, this can be easily accommodated by stacking them together and changing the random effect from $\alpha_{j[i]}$ into $\boldsymbol{z}_i^T\boldsymbol{\alpha}$ where $\boldsymbol{z}_i$ is a sparse vector with '1's to pull out the corresponding elements of $\boldsymbol{\alpha}$. The prior covariance matrix on $\boldsymbol{\alpha}$ in this case is block-diagonal. Our code, as written, can implement an arbitrary number of random effects subject to the ability to invert the matrix.

* Conditional M-Step (1): Update $[\sigma_\alpha^2]^{(t)}$ using the standard linear mixed effect update. Verbally, for all levels j of the random effect, we average $E[\alpha_j^2] = E[\alpha_j]^2 + Var(\alpha_j)$ as well as adding in the prior. To do this, we select the relevant elements of $\boldsymbol{\mu}_\alpha^{(t)}$ and $\boldsymbol{\Sigma}_\alpha^{(t)}$. In the case of a single random effect, this is all elements of $\boldsymbol{\mu}_\alpha^{(t)}$ and all diagonal elements of $\boldsymbol{\Sigma}^{(t)})\alpha$. Formally, the update is:

$$(\sigma_\alpha^2)^{(t)} = \frac{a_\alpha + \sum_j \left([\boldsymbol{\mu}^{(t)}\alpha]_j + [\boldsymbol{\Sigma}^{(t)}]_{j,j}\right)}{N_j + b_\alpha} \tag{22}$$

* Conditional M-Step (2): Update $\boldsymbol{\beta}^{(t)}$ using the weighted sparsereg updates where $\alpha_{j[i]}$ is replaced with $\boldsymbol{\mu}_\alpha^{(t)}$ from above. To speed convergence, our default setting performs a new E-step for the ω_i (i.e. using the updated $\boldsymbol{\beta}$) before doing this M-

Step. While again approximate, this is roughly justified by reference to the AECM algorithm (Meng and Van Dyk, 1997). For point of reference, if the M-Step was a simple linear regression, i.e. *not* penalized, it would be as follows:

$$\boldsymbol{\beta}^{(t)} = \left(\tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}}\right)^{-1} \left(\sum_i \left[\tilde{s}_i - \boldsymbol{z}_i^T \boldsymbol{\mu}_\alpha^{(t)}\right] \tilde{\boldsymbol{x}}_i\right) \tag{23}$$