*The Annals of Applied Statistics*
2019, Vol. 13, No. 4, 2260–2288
https://doi.org/10.1214/19-AOAS1251

# FITTING A DEEPLY NESTED HIERARCHICAL MODEL TO A LARGE BOOK REVIEW DATASET USING A MOMENT-BASED ESTIMATOR

BY NINGSHAN ZHANG, KYLE SCHMAUS AND PATRICK O. PERRY

*New York University, Stitch Fix and Oscar Health*

We consider a particular instance of a common problem in recommender systems, using a database of book reviews to inform user-targeted recommendations. In our dataset, books are categorized into genres and subgenres. To exploit this nested taxonomy, we use a hierarchical model that enables information pooling across across similar items at many levels within the genre hierarchy. The main challenge in deploying this model is computational. The data sizes are large and fitting the model at scale using off-the-shelf maximum likelihood procedures is prohibitive. To get around this computational bottleneck, we extend a moment-based fitting procedure proposed for fitting single-level hierarchical models to the general case of arbitrarily deep hierarchies. This extension is an order of magnitude faster than standard maximum likelihood procedures. The fitting method can be deployed beyond recommender systems to general contexts with deeply nested hierarchical generalized linear mixed models.

## 1. Introduction.

Given a dataset of books, users and user reviews of those books, consider the problem of recommending books to users. This problem is a specific instance of the recommender system problem common in commercial applications (Adomavicius and Tuzhilin (2005)). In our context the following data are available:

- A collection of 38,659 books, each with an author title, genre, subgenre and sub-subgenre, a taxonomy scraped from `amazon.com` by McAuley, Pandey and Leskovec (2015). Figure 1 shows the first two levels of the book genre hierarchy.
- A set of 157,638 ratings of the books made by 38,085 users taken from the BookCrossing dataset, an anonymized collection of book reviews harvested from `bookcrossing.com` by Ziegler et al. (2005).
- User age and location (continent) included with the BookCrossing dataset.

Appendix A gives descriptive statistics for the dataset, including descriptions of how the ratings are distributed between user age groups and continents. Most ratings are from users, aged 20–40 years, in the United States.

Received May 2018; revised November 2018.
*Key words and phrases.* Generalized linear mixed model, hierarchical model, method of moments, recommender system.

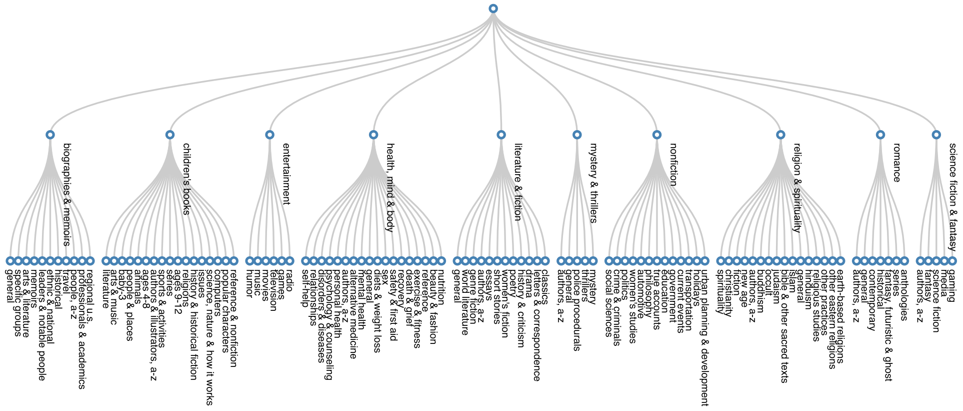

FIG. 1. *First two levels of the book genre hierarchy for the* 10 *largest subgenres.*

Others have built recommender systems for the BookCrossing dataset (Agarwal and Chen (2010), Weng et al. (2008), Zhang, Cao and Yeung (2010)). Our application is unique in that we will attempt to leverage the book genre hierarchy to improve recommendations.

Rather than solving the book recommendation problem directly, we will attempt to solve a proxy problem—for each book-user pair predict whether the user would like the book if they rated it. We can use the solution to the proxy problem to solve the original problem by recommending books with the highest predicted "like" probabilities. This proxy approach to the recommendation system is common (Adomavicius and Tuzhilin (2005), Ansari, Essegaier and Kohli (2000)); it ignores information inherent in a user's selection of which books to review, but despite this it often gives reasonable downstream results.

One strategy for solving a recommender system problem is the content-based approach, using user attributes together with book-specific parameters to make recommendations. Another strategy is the collaborative approach, recommending books that are liked by similar users. We prefer instead a variant of the hierarchical-model-based approach advocated by Condliff, Lewis and Madigan (1999) and Ansari, Essegaier and Kohli (2000) that combines the content-based and collaborative strategies.

The simplest form of the hierarchical model approach is a flat item hierarchy with each item sitting directly under the root. In our context, the flat model would assign a random effect vector $u_i$ to each book $i$ that relates the popularity of the book to user- and context-specific covariate vector along with a fixed-effect vector $\beta$ that relates book popularity to another covariate vector (possibly the same). For a particular user review of a book $i$, let $y$ be a binary indicator of whether the user liked the book, and let $x_0$ and $x_1$ be the user-context covariate vectors associate fixed and random effects, respectively. The link between the effects, the covariates and the response is

$$\operatorname{logit}\Pr(y = 1 \mid u_i) = \beta^T x_0 + u_i^T x_1. \tag{1}$$

The random effect vectors are independent multivariate normal random vectors with mean *zero* and covariance matrix $\Sigma_1$:

$$u_i \sim \mathcal{N}(0, \Sigma_1). \tag{2}$$

Equation (1) demonstrates the content-based aspect of the model, where user attributes ($x_0$ and $x_1$) are linked to preferences. Equation (2) introduces the collaborative-based features: books with abundant data will have strongly identified random effects $u_i$; others will have posterior means (conditional on the available review data) determined in part by the effects of similar items through the covariance matrix $\Sigma_1$.

In our application, we have a richer hierarchy with books nested under author, sub-subgenre, subgenere and genre. We can exploit this hierarchy in the model formulation by allowing for a random effect vector at each node in the hierarchy and not just at the leaves. Doing so allows for information pooling in random effect estimates across the levels in the hierarchy. We elaborate on this benefit in Section 2.

Despite its appeal, the hierarchical modeling and more general mixed modelling approaches to recommender systems have long considered infeasible at commercial scale due to the high computational demands of fitting the model (Agarwal (2008), Naik et al. (2008)). Recent progress has expanded the scope of application of these models. Gao and Owen (2016), Gao and Owen (2017) proposed a moment-based approach for estimating the parameters of a crossed effects model; Perry (2017) proposed a moment-based approach for fitting a flat hierarchical model; Tan et al. (2018) proposed a kernel-based approach for fitting a linear flat hierarchical model, and Zhang et al. (2016) developed a parallelized maximum likelihood fitting algorithm that can exploit multiple computing cores.

To fully exploit the deeply nested book hierarchy in a computationally efficient manner, we will in the sequel develop a moment-based fitting method for hierarchical models of arbitrary depth. Note that deeper models involve more parameters and thus require more data to learn. In practice, one should perform model selection to choose the depth of the hierarchical model. For example, in our book recommendation application we used out-of-sample prediction performance on a development set to determine the best number of hierarchies.

The rest of the paper is organized as follows. In Section 2 we elaborate on the benefits of using a hierarchical model in our context. Next, in Section 3 we introduce the details of our model, using framework suitable for describing general hierarchical models. Our fitting method proceeds in two passes. In the first pass, we use the available data to get initial parameter estimates at the leaves of the tree, and then we propagate information in these estimates up to the root. In the second pass, we use the accumulated information to refine the estimates back down from the root back to the leaves. We describe these procedures in Section 4 and Section 5, respectively. After investigating the performance of our method in simulations in Section 6, we apply the procedure to our dataset in Sections 7 and 8. We conclude with a short discussion in Section 9.

Our fitting procedure is implemented in the `mbest` R package and is available at https://cran.r-project.org/package=mbest.

## 2. Local and global approaches.

This hierarchical modeling approach interpolates two extremes, a "global" and a "local" approach. The global approach would have a single parameter vector shared by all books and fit by lumping the reviews for all books together. The local approach would have a different parameter vector for each book and fit in isolation using only the reviews of the corresponding book. The global approach corresponds to setting $u_i = 0$ for all books $i$; the local approach corresponds to treating $u_i$ as nonrandom.

The appeal of the hierarchical model can be demonstrated by comparing the performance of a global and a local model. We perform this comparison at the author level. The global model will have a single parameter vector shared by all authors; the local model will have a different parameter vector for each author. Both models use the same set of covariates, described later in Section 3. We fit both models on a training set then evaluate on a test set. In the global model, the probability that a user likes an item is described as

$$\operatorname{logit} \Pr(y = 1) = \beta^T x,$$

where $x$ is the vector of user covariates. The predictions from this model are the same for all authors. In the local model, the probability that the user likes book $i$ is described as

$$\operatorname{logit} \Pr(y = 1) = \beta_i^T x.$$

This takes a similar form to the global model, but the coefficients $\beta_i$ depend on the book author $i$.

TABLE 1
*Average misclassification rates of local author-specific* ($\mathrm{Error}_\mathrm{a}$) *and global* ($\mathrm{Error}_\mathrm{g}$) *models* (*standard deviations in parentheses*)

| # Author Ratings | $\mathrm{Error}_\mathrm{a}$ (%) | $\mathrm{Error}_\mathrm{g}$ (%) | $\mathrm{Error}_\mathrm{a} - \mathrm{Error}_\mathrm{g}$ (%) |
|---|---|---|---|
| [10, 20] | 38.52 (0.53) | 35.16 (0.52) | 3.36 |
| (20, 50] | 37.15 (0.42) | 34.91 (0.41) | 2.24 |
| (50, 100] | 34.76 (0.48) | 34.20 (0.48) | 0.55 |
| (100, 1000] | 32.15 (0.40) | 32.32 (0.40) | −0.16 |

Table 1 shows the test set misclassification rates for the two fitted models, grouped by the number of ratings per author. For authors with 100 or fewer ratings, the global model performs better on average than the local model. Only in the group of authors with large number of ratings (more than 100) does the local model perform better.

What is needed is an adaptive model that can interpolate between these two extremes. For authors with abundant data, use the local model; for authors with little or no data, use the global model; for others, use some combination of the two models.

The hierarchical model achieves the interpolation between local and global models automatically. For items $i$ with abundant data, the posterior distribution (conditional on the data) for the random effect vector $u_i$ is concentrated around the local coefficient estimate that uses only the reviews for item $i$. For items with no data, the posterior distribution for $u_i$ is diffuse with mean zero; the predictions for item $i$ are determined, mostly, by the fixed effect vector $\beta$ shared globally by all items. As the number of reviews for item $i$ increases, the posterior distribution for $u_i$ and the corresponding predictions for item $i$ interpolate between these local and global extremes.

The flat item hierarchy shares information across all items. In our application, we have a deep hierarchy of books. In Section 3, we will show how to exploit this deep hierarchy by a using a model with a random effect vector at each node in the hierarchy. Predictions for the items at the leaves involve the random effects on the nodes on the path from the root of the hierarchy to the item. Information pooling occurs at all levels of the hierarchy, with siblings in a subtree pooled to estimate the posterior distribution of their random effects. A hierarchy node with a subtree of abundant data will have an estimated random effect close to what would come in a fitted model. For other nodes, the estimate will involve information pooling across other nodes at the same level in the tree.

## 3. Modeling framework.

For our proxy problem, the goal is to estimate, for a given book and user, the probability that the user would like the book conditional on the user rating the book. We have a set of user and context covariates for each review and a response $y$ indicating whether the user liked the book. We also have a deeply nested hierarchy of books nested under author, sub-subgenere, subgenre and genre. In what follows, we describe a model that leverages the book hierarchy in a way that facilitates information pooling for the estimates across the levels in the hierarchy.

To introduce the model, we first need to be more precise about what we mean by a hierarchy in the context of our problem and other similar settings. For us, a "depth-$d$ hierarchy" is a tree where all leaves have depth $d$. In such a hierarchy, we label the nodes of the tree by unique strings of natural numbers:

- the root of the hierarchy gets labeled by the empty string, denoted $*$;

- the children of the root get labeled by the length-1 strings $1, 2, \dots, M_*$;
- in general, if $i$ is the label of a node, we let $M_i$ denote its number of children; we label these children by the strings obtained from concatenating the label $i$ with the child identifiers: $i1, i2, \dots, iM_i$.

In this labeling scheme, each node other than the root has a label that can be represented as $ij$, where $i$ is the node's parent and $j$ is a natural number in the range $1, 2, \dots, M_i$.

For a node $i$, we denote its depth by $|i|$, equal to its distance from the root; this is also equal to the length of its label. For any depth $l$ in the range $1, \dots, d$, we let $N_l$ denote the set of nodes with depth $l$. Finally, for node $i$ of depth $l$ and for $0 \le k \le l$, we let $\pi(i, k)$ denote its ancestor at depth $k$ in the hierarchy, setting $\pi(i, |i|) = i$.

In the context of our application, the leaves of the hierarchy are authors. The internal nodes are genres and subgenres. We observe data for each author (ratings for that author by users); our goal is to relate these ratings to the user covariates, using the structure of the book hierarchy to inform our predictions.

In general terms, the hierarchical model supposes that at each leaf node $i \in N_d$ we observe a response vector $y_i$ of length $n_i$. The behavior of this response vector is linked to observable covariates through a vector of nonrandom fixed effects identified with the root of the hierarchy, and a set of random effects identified with the nodes in the hierarchy on the path from the root to the leaf $i$. Different levels of the hierarchy may use different sets of user and book features to predict the user's probability of liking the book. We denote these features by $X_{i0}, \dots, X_{id}$, where $X_{il}$, a matrix of dimension $n_i \times q_l$, contains the features used by level-$l$ of the hierarchical model to predict the response vector $y_i$. To link these features to the response, we posit the existence of a fixed effect vector $\beta$ of dimension $q_0$ identified with the root of the hierarchy, along with a random effect vector $u_i$ at every other node in the tree such that $u_i$ has dimension $q_l$ when $i$ is at depth $l$. The distribution of the response $y_i$ is determined by some function of the linear predictor $\eta_i$, defined as

$$\eta_i = X_{i0}\beta + \sum_{l=1}^{d} X_{il} u_{\pi(i,l)}. \tag{3}$$

This predictor involves the fixed effect vector $\beta$ and the random effects of all nodes on the path from the leaf $i$ to the root.

In our application, we have a binary response vector $y_i$ with entries indicating whether the user liked the book. We use the canonical logistic link, supposing that for $k = 1, \dots, n_i$; this predictor relates to the response as

$$\operatorname{logit} \Pr(y_{ik} = 1 \mid u) = \eta_{ik}, \tag{4}$$

where $u$ without subscript denotes the collection of all random effects. We further suppose that the components of the vector $y_i$ are independent of each other conditional on $u$. In an application with a continuous response vector $y_i$, we would instead typically specify that $y_i$ has independent Gaussian components with mean and variance given by

$$\mathbb{E}(y_{ik} \mid u) = \eta_{ik}, \qquad \operatorname{var}(y_{ik} \mid u) = \phi \tag{5}$$

for $k = 1, \dots, n_i$ and for some dispersion parameter $\phi$. The hierarchical model is not limited to these two settings, and in principle a modeler could specify any link between the linear predictor $\eta_i$ and the mean of the response $y_i$.

To endow our model with a mechanism that allows borrowing strength across similar items in the hierarchy, we model the $d$ populations of random effects at the levels of the hierarchy. We treat these populations as independent. For the population of level-$l$ random effects, $l = 1, \dots, d$, we suppose that each item $u_i$ is an independent draw from a multivariate normal distribution with mean-zero and covariance matrix $\Sigma_l$ for some $q_l \times q_l$ covariance matrix $\Sigma_l$:

$$u_i \sim \mathcal{N}(0, \Sigma_l) \quad \text{for } i \in N_l. \tag{6}$$

We further suppose that all random effects $u$ are independent of each other.

In the sequel, we discuss estimation for the depth-$d$ hierarchical model. That estimation procedes in two stages: first, estimate the model parameters $\beta$ and $\Sigma_1, \dots, \Sigma_d$. Next, use the model parameters to get empirical Bayes estimates of the random effects $\{\hat{u}_i\}$. The empirical Bayes estimation procedure is the part of the model estimation that leverages information across different levels of the hierarchy. Our estimates of the covariance matrices $\Sigma_1, \dots, \Sigma_d$ allow us to impute components of particular random effects vectors when we only have information about a subset of their components.

**4. Fitting procedure.** Frequentist hierarchical models like the one described in the previous section are often fit via maximum likelihood methods (Bates et al. (2013)). However, these fitting algorithms can be prohibitively slow for large datasets like those that appear in commercial-scale settings (Agarwal (2008), Naik et al. (2008)). Perry (2017) got over the computational hurdle in a depth-1 hierarchical model by using a moment-based estimation procedure, adapted from an earlier procedure due to Cochran (1937). Here we will extend Perry's (2017) procedure to handle hierarchy of arbitrary depth.

Throughout the section we will assume that the model described in (3) and (6) is in force. For the response data $y_i$ and the leaf nodes $i \in N_d$, we will allow for both the logistic regression case from (4) and the normal response case from (5). Our estimators extend naturally to any generalized linear model at the leaves with shared dispersion parameter $\phi$.

The estimation procedure is easiest to describe if we reparametrize. To do so, for any node $i$, let $b_i$ denote the vector of fixed and random effects on the path from the root up to and including $i$. Specifically, set $b_* = \beta$ and for node $ij$ with parent $i$ define recursively

$$b_{ij} = (b_i, u_{ij}).$$

For depth $l = 1, \ldots, d$, let $p_l$ be the total number of fixed and random effects up to and including depth $l$,

$$p_l = q_0 + q_1 + \ldots + q_l,$$

if $i \in N_l$, then $b_i$ has $p_l$ components.

Now, for leaf node $i \in N_d$ define the matrix obtained by concatenating the columns of feature matrices $X_i = [X_{i0}\ X_{i1} \cdots X_{id}]$, so $X_i$ has dimension $n_i \times p_d$. In this reparametrized form, the linear predictor at leaf node $i$ is

$$\eta_i = X_i b_i. \tag{7}$$

This form makes the hierarchical model look somewhat like a standard generalized linear model, but the effect vector $b_i$ includes both nonrandom and random components—the fixed effect vector $\beta$ and the random effects $u_{\pi(i,l)}$ on the path from the root to the leaf $i$.

The estimation procedure for the hierarchical model is defined by repeatedly pruning the tree by reducing the leaves to a set of estimates at their parents. The high level description of the procedure is as follows:

1. Produce estimates $\hat{b}_i$ of $b_i$ at each leaf node $i \in N_d$. Set $l = d$.
2. We have at hand estimates $\hat{b}_{ij}$ of $b_{ij} = (b_i, u_{ij})$ for each node $ij \in N_l$. For each $i \in N_{l-1}$, combine the child estimates, $\hat{b}_{ij}$ for $j = 1, \ldots, M_i$ to produce an estimate $\hat{b}_i$ of $b_i$ and an estimate $\hat{\Sigma}_{li}$ of $\Sigma_l$.
3. Combine estimates $\hat{\Sigma}_{li}$ for $i \in N_{l-1}$ to produce a final estimate $\bar{\Sigma}_l$.
4. If $l = 1$, set $\bar{\beta} = \hat{b}_*$ to be the final estimate of the fixed effects and stop. Otherwise, go to Step 2 with the level $l$ decreased to $l - 1$.

In settings with a dispersion parameter $\phi$, we handle this parameter analogously to $\Sigma_d$.

When the fitting procedure terminates, we will have final estimates $\bar{\beta}$ and $\bar{\Sigma}_1, \ldots, \bar{\Sigma}_d$ of the fixed effects and the random effect covariance matrices. The rest of this section is devoted to detailing the individual steps of the fitting procedure.

The computational cost of estimating $\hat{b}_i$ at all leaf nodes $N_d$ (step 1) is of order $O\{Np_d^2\}$, where $N$ denotes the total number of observations. For any level $l \in \{1, 2, \ldots, d\}$, it takes $O(|N_l|p_l^3)$ operations to compute $\hat{b}_i$ for all nodes $i \in N_{l-1}$, followed by $O(|N_l|(p_l^3 + q_l^4) + |N_{l-1}|q_l^6)$ to compute $\bar{\Sigma}_l$, where $|N_l|$ denotes the number of nodes on level $l$. In total, the fitting procedure takes at most $O(Np_d^2 + \sum_{l=1}^{d}(|N_l|(p_l^3 + q_l^4) + |N_{l-1}|q_l^6))$ operations.

4.1. *Step* 1: *Estimate parameters at the leaves.* The first step in the estimation procedure is to use the data $y_i$ at each leaf $i \in N_d$ to produce an estimate $\hat{b}_i$ of $b_i$, the vector of fixed and random effects on the path from the root to the leaf.

Recall that $\eta_i = X_i b_i$. We will explicitly handle cases where the combined predictor matrix $X_i$ is rank degenerate. We will only require that, conditional on $b_i$, the estimate $\hat{b}_i$ has negligible bias outside the null space of $X_i$ and is approximately normally distributed with known covariance matrix.

First, we handle the normal model (5), where for $k = 1, \dots, n_i$ the components of the response satisfy $y_{ik} = \eta_{ik} + \varepsilon_{ik}$ for a mean-zero Gaussian error vector $\varepsilon_i$ with independent components $\varepsilon_{ik}$ for $k = 1, \dots, n_i$ having unknown variance $\phi$. In this case we set

$$\hat{b}_i = (X_i^T X_i)^{\dagger} X_i^T y_i,$$

where † denotes pseudo-inverse. When $X_i$ has full rank, the estimate $\hat{b}_i$ is the unique least-squares estimate of $b_i$; otherwise, the least-squares estimate is not unique and $\hat{b}_i$ is one of the vectors minimizing the squared Euclidean norm $\|y_i - X_i \hat{b}_i\|^2$.

To define the estimate of the dispersion parameter $\phi$, we let $r_i$ denote the rank of $X_i$. When $r_i < n_i$ we set

$$\hat{\phi}_i = \frac{1}{n_i - r_i} \|y_i - X_i \hat{b}_i\|^2;$$

otherwise, we set $\hat{\phi}_i = 0$. We combine the estimates $\hat{\phi}_i$ across all the leaves to get a single estimate for the dispersion parameter:

$$\bar{\phi} = \frac{\sum_{i \in N_d} (n_i - r_i) \hat{\phi}_i}{\sum_{i \in N_d} (n_i - r_i)}.$$

Next we derive the properties of the estimate $\hat{b}_i$. First, let $X_i = U_i D_i V_i^T$ denote a compact singular value decomposition where $D_i$ is a diagonal matrix of dimension $r_i \times r_i$ with positive diagonal entries. Then

$$\begin{aligned} \hat{b}_i &= V_i D_i^{-1} U_i^T y_i \\ &= V_i V_i^T b_i + V_i D_i^{-1} U_i^T \varepsilon_i. \end{aligned}$$

Thus,

$$D_i V_i^T (\hat{b}_i - b_i) = e_i,$$

where $e_i = U_i^T \varepsilon_i$ is a mean-zero Gaussian random vector of $r_i$ independent components, each with variance $\phi$. If we set $Z_i = \bar{\phi}^{-1/2} D_i V_i^T$, then the quantity $Z_i(\hat{b}_i - b_i)$ is approximately mean-zero normal with identity covariance matrix.

For the logistic regression model (4), we proceed analogously, but we use the Firth's biased-reduced estimator (Firth (1993)) in place of the least squares estimator for $b_i$. This is a refinement of the maximum likelihood estimator that is well defined even when the responses are perfectly separated by a linear combination of the predictors. In cases where $X_i$ is rank-degenerate, there are multiple such estimators; we arbitrarily take $\hat{b}_i$ to be one of them. The properties of the estimator are like those of the maximum likelihood. As the sample size $n_i$ increases, the estimator is asymptotically unbiased with covariance equal to the inverse information matrix. In the case of rank-deficient feature matrix $X_i$, the information matrix takes the form $I(b_i) = V_i D_i(b_i) V_i^T$ where $V_i$ is the matrix of right singular vectors of $X_i$. In this case, if we set $Z_i = D_i^{1/2}(\hat{b}_i) V_i^T$, then $Z_i(\hat{b}_i - b_i)$ is approximately mean-zero normal with identity covariance matrix.

In both the normal and the logistic regression case, we can find an estimator $\hat{b}_i$ and a matrix $Z_i$ with full column rank $r_i$ such that, conditional on $b_i$, the quantity $Z_i(\hat{b}_i - b_i)$ is approximately normal with identity covariance. In the logistic regression case, the quality of the normal approximation depends on the sample size $n_i$ being large.

4.2. *Step* 2: *Combine the estimates at level $l$.* We now suppose that for some level $l$, for each node $ij \in N_l$ we have a matrix $Z_{ij}$ of full row rank $r_{ij}$, such that conditional on $b_{ij}$,

$$\mathbb{E}\{Z_{ij}(\hat{b}_{ij} - b_{ij}) \mid b_{ij}\} = 0, \qquad \operatorname{cov}\{Z_{ij}(\hat{b}_{ij} - b_{ij}) \mid b_{ij}\} = I.$$

For linear models, these two conditions hold exactly; in nonlinear models these will only hold approximately, with the quality of the approximation depending on the size of the sample used to estimate $\hat{b}_{ij}$. We will show how to combine estimates $\hat{b}_{i1}, \dots, \hat{b}_{iM_i}$ to get an estimate $\hat{b}_i$ of $b_i$ for node $i \in N_{l-1}$ and an estimate $\hat{\Sigma}_{il}$ of $\Sigma_l$.

Recall that $b_{ij} = (b_i, u_{ij})$ for each node $ij \in N_l$, where $u_{ij}$ is the random effects on level $l$, and $b_i$, $u_{ij}$ are of length $p_{l-1}$, $q_l$, respectively. Let $Z_{ij} = U_{ij} D_{ij} V_{ij}^T$ be a compact singular value decomposition. When the context is clear, for simplicity we denote $V_{ij1}$ as the first $p_{l-1}$ rows of $V_{ij}$ and denote $V_{ij2}$ as the last $q_l$ rows of $V_{ij}$.

We have the following (unconditional) moment equations:

$$\mathbb{E}(V_{ij}^T \hat{b}_{ij}) = V_{ij}^T (b_i, 0) = V_{ij1}^T b_i, \tag{8}$$

$$\operatorname{cov}(V_{ij}^T \hat{b}_{ij}) = D_{ij}^{-2} + V_{ij}^T \begin{bmatrix} 0 & 0 \\ 0 & \Sigma_l \end{bmatrix} V_{ij} = D_{ij}^{-2} + V_{ij2}^T \Sigma_l V_{ij2}. \tag{9}$$

The moment equations (8) and (9) hold for any node $ij \in N_l$; therefore, by standard moment matching method, we want to take empirical mean of terms on the left-hand side and set parameters on the right-hand side to match it. However, we cannot do this right away, since the dimension of $V_{ij}^T$, or equivalently the rank $r_{ij}$, may vary by nodes $ij \in N_l$. To overcome this, we augment the moment equations to have same dimension across nodes $ij \in N_l$. In particular, with any choice of symmetric positive-definite matrix $W_{ij}$ for each node $ij \in N_l$, we have

$$\mathbb{E}(V_{ij1} W_{ij} V_{ij}^T \hat{b}_{ij}) = V_{ij1} W_{ij} V_{ij1}^T b_i, \tag{10}$$

$$\operatorname{cov}(V_{ij2} W_{ij} V_{ij}^T \hat{b}_{ij}) = V_{ij2} W_{ij} (D_{ij}^{-2} + V_{ij2}^T \Sigma_l V_{ij2}) W_{ij} V_{ij2}^T. \tag{11}$$

We use the semiweighted scheme for choosing $W_{ij}$ as described by Perry (2017).

Now, the moment equations have consistent dimension across all nodes. For every node $ij \in N_l$, equation (10) has dimension $p_{l-1} \times 1$, and (11) has dimension $q_l \times q_l$. Based on equation (10), we define the moment-based estimator $\hat{b}_i$ as

$$\hat{b}_i = \Omega_i^{\dagger} \sum_{j=1}^{M_i} V_{ij1} W_{ij} V_{ij}^T \hat{b}_{ij}, \quad \Omega_i = \sum_{j=1}^{M_i} V_{ij1} W_{ij} V_{ij1}^T,$$

where † denotes pseudoinverse. Based on equation (11), the moment-based estimator $\hat{\Sigma}_{il}$ should satisfy

$$\sum_{j=1}^{M_i} (V_{ij2} W_{ij} V_{ij}^T \hat{b}_{ij} - V_{ij2} W_{ij} V_{ij1}^T b_i)(V_{ij2} W_{ij} V_{ij}^T \hat{b}_{ij} - V_{ij2} W_{ij} V_{ij1}^T b_i)^T$$

$$= \sum_{j=1}^{M_i} V_{ij2} W_{ij} D_{ij}^{-2} W_{ij} V_{ij2}^T + \sum_{j=1}^{M_i} V_{ij2} W_{ij} V_{ij2}^T \hat{\Sigma}_{il} V_{ij2} W_{ij} V_{ij2}^T.$$

In practice, we do not have access to the true $b_i$ to compute $\hat{\Sigma}_{il}$ in the above equation, instead we use the empirical estimate $\hat{b}_i$. If the result $\hat{\Sigma}_{il}$ is not positive semidefinite, we project it onto the cone of positive semidefinite matrices and obtain the final estimate.

Let $\Omega_i = V_i D_i V_i^T$ denote the eigendecomposition of the positive semidefinite matrix $\Omega_i$. Let $\Omega_i^{1/2}$ denote the symmetric square root of $\Omega_i$. Perry's (2017) results imply that, subject to assumptions on the sample size and conditional on $b_i$, the quantity $\Omega_i^{1/2}(\hat{b}_i - b_i)$ is approximately normally distributed with

$$\mathbb{E}\{\Omega_i^{1/2}(\hat{b}_i - b_i) \mid b_i\} = 0, \qquad \operatorname{cov}\{\Omega_i^{1/2}(\hat{b}_i - b_i) \mid b_i\} \approx V_i V_i^T.$$

The error in the approximation tends to zero as the number of child nodes $M_i$ increases. In addition, since $V_i^T \Omega_i^{1/2} = V_i^T V_i D_i^{1/2} V_i^T = D_i^{1/2} V_i^T$, we can rewrite the above results as

$$\mathbb{E}\{D_i^{1/2} V_i^T (\hat{b}_i - b_i) \mid b_i\} = 0, \qquad \operatorname{cov}\{D_i^{1/2} V_i^T (\hat{b}_i - b_i) \mid b_i\} \approx I. \tag{12}$$

Perry (2017) details the precise assumptions required for these results along with the quality of the approximations.

4.3. *Step* 3: *Combine the level-l covariance estimates.* At the end of Step 2 in the procedure, we have an estimates $\hat{\Sigma}_{il}$ of $\Sigma_l$ for each $i \in N_{l-1}$. In Step 3, we combine these estimates to produce a final estimate $\bar{\Sigma}_l$ by taking a weighted average: of $\hat{\Sigma}_{il}$ over all nodes $i \in N_{l-1}$,

$$\bar{\Sigma}_l = \frac{\sum_{i \in N_{l-1}} M_i \hat{\Sigma}_{il}}{\sum_{i \in N_{l-1}} M_i}.$$

Nodes with higher numbers of children $M_i$ get more weights.

4.4. *Step* 4: *Recurse or stop*. If we are at the root of the tree, so that $l = 0$ and we have an estimate $\hat{b}_*$, then we terminate the estimation procedure by setting our final estimate of the fixed effects to $\bar{\beta} = \hat{b}_*$. Otherwise, we decrement $l$ to $l - 1$ and go to Step 2 with $Z_i = D_i^{1/2} V_i^T$ that are used in equation (12).

## 5. Empirical Bayes random effect estimates.

At the end of the fitting procedure described in Section 4, we have estimate $\bar{\beta}$ of the fixed effect vector and estimates $\bar{\Sigma}_1, \dots, \bar{\Sigma}_d$ of the random effect covariance matrices. We also have at each node $i$ in the hierarchy a preliminary estimate $\hat{b}_i$ of $b_i$, the fixed and random effects on the path from the root to node $i$. These preliminary estimates do not share information across the hierarchy; estimate $\hat{b}_i$ is determined only from the data at the leaves descending from $i$. We can improve the estimates by replacing each $\hat{b}_i$ with an empirical Bayes estimate $\bar{b}_i$ that pools information across the hierarchy.

The information-pooling algorithm works top-down from the root. It starts by setting $\bar{b}_* = \bar{\beta}$. Then, at depth-1 nodes $j \in N_1$, the procedure uses $\bar{b}_*$ and $\bar{\Sigma}_1$ together with $\hat{b}_j$ to get a refined estimate $\bar{b}_j$. This process repeats, level by level, until we get refined estimates at the leaves.

The full procedure is as follows:

1. Set $\bar{b}_* = \bar{\beta}$ and set $l = 0$.
2. If $l = d$, stop.
3. For each node $i \in N_l$ we have a refined estimate $\bar{b}_i$. For each child $ij$ for $j = 1, \dots, M_i$, we have a preliminary estimate $\hat{b}_{ij}$. Use $\bar{b}_i$ together with $\hat{b}_{ij}$ and $\bar{\Sigma}_{l+1}$ to produce a refined estimate $\bar{b}_{ij}$.
4. Increment $l$ to $l + 1$ and go to Step 2.

After applying this procedure, we have a refined estimate $\bar{b}_i$ at each node in the tree. The estimates at each leaf $i$ can be used to make refined estimates of the linear predictors ($\bar{\eta}_i = X_i \bar{b}_i$), or they can be used to make predictions for new data.

To complete the description of the procedure, we need to explain Step 3 in more detail. In this step we have at our disposal $\bar{b}_i$, $\bar{\Sigma}_{l+1}$ and $\hat{b}_{ij}$ for node $i \in N_l$ and its children. Further, we have a matrix $Z_{ij}$ of full column rank $r_{ij}$ such that $Z_{ij}(\hat{b}_{ij} - b_{ij})$ is approximately distributed as a multivariate normal with identity covariance matrix.

By definition, $b_{ij} = (b_i, u_{ij})$ where $u_{ij}$ is the random effect vector for node $ij \in N_{l+1}$, and $b_i$, $u_{ij}$ are of length $p_l$, $q_{l+1,}$ respectively. We denote $Z_{ij1}$ as the first $p_l$ columns of $Z_{ij}$ and denote $Z_{ij2}$ as the last $q_{l+1}$ columns of $Z_{ij}$. Conditional on $b_i$, we have the following (approximate) Bayesian linear regression model for any node $j \in N_{l+1}$:

$$\begin{aligned} u_{ij} &\sim \mathcal{N}(0, \Sigma_{l+1}), \\ Z_{ij}\hat{b}_{ij} &= Z_{ij1} b_i + Z_{ij2} u_{ij} + e_{ij}, \\ e_{ij} &\sim \mathcal{N}(0, I). \end{aligned}$$

The empirical Bayes estimate $\hat{u}_{ij}$ is an estimate of the posterior mean of $u_{ij}$ conditional on the observed data $Z_{ij}\hat{b}_{ij}$, acquired by using plug-in estimates $\bar{b}_i$ and $\hat{\Sigma}_{l+1}$ for $b_i$ and $\Sigma_{l+1}$.

To derive the posterior distribution define $Y_{ij} = Z_{ij}\hat{b}_{ij} - Z_{ij1}b_i$, noting that the conditional distribution $u_{ij} \mid \{Z_{ij}\hat{b}_{ij}, b_i\}$ is the same as that of $u_{ij} \mid \{Y_{ij}, b_i\}$. By Bayes rule, then the posterior density of $u_{ij}$ satisfies

$$\begin{aligned} p(u_{ij} \mid Y_{ij}, b_i) &\propto p(Y_{ij} \mid u_{ij}, b_i) p(u_{ij}) \\ &\propto \exp\left\{-\frac{1}{2}(Y_{ij} - Z_{ij2}u_{ij})^T (Y_{ij} - Z_{ij2}u_{ij}) - \frac{1}{2}u_{ij}^T \Sigma_{l+1}^{-1} u_{ij}\right\} \\ &\propto \exp\left\{-\frac{1}{2}\left[u_{ij}^T (Z_{ij2}^T Z_{ij2} + \Sigma_{l+1}^{-1}) u_{ij} - 2Y_{ij}^T Z_{ij2} u_{ij}\right]\right\}. \end{aligned}$$

Thus, the posterior distribution is that of a multivariate Gaussian with expected value given by

$$\mathbb{E}(u_{ij} \mid Z_{ij}\hat{b}_{ij}, b_i) = (Z_{ij2}^T Z_{ij2} + \Sigma_{l+1}^{-1})^{-1} Z_{ij2}^T (Z_{ij}\hat{b}_{ij} - Z_{ij1}b_i).$$

The empirical Bayes estimate of $u_{ij}$ comes from using this expression in conjunction with plug-in estimates for $\Sigma_{l+1}$ and $b_i$:

$$\hat{u}_{ij} = (Z_{ij2}^T Z_{ij2} + \hat{\Sigma}_{l+1}^{-1})^{-1} Z_{ij2}^T (Z_{ij}\hat{b}_{ij} - Z_{ij1}\bar{b}_i).$$

The refined estimate of $b_{ij}$ is $\bar{b}_{ij} = (\bar{b}_i, \hat{u}_{ij})$. Note that $\bar{b}_{ij}$ pools information across the hierarchy by using $\bar{b}_i$ and $\hat{\Sigma}_{l+1}$ which contain information from all the nodes in the hierarchy.

## 6. Simulation.

To evaluate our proposed estimation method, we compare its performance with three other procedures:

- `glmer`, a maximum likelihood procedure, implemented as part of the `lme4` R package (Bates et al. (2013)).
- `glmer.split`, a data-splitting estimation procedure, which randomly splits the dataset into 10 subsets, computes estimates on each of them separately using `glmer` and then combines the estimates by averaging them. We implemented the procedure ourselves in R; the algorithm is based on procedures proposed by Huang and Gelman (2005), Gebregziabher et al. (2012) and Scott et al. (2013).
- `sgd`, which uses stochastic gradient descent to maximize a regularized version of the $h$-likelihood. We implemented the procedure in a combination of C and R; the algorithm is based on procedures proposed by Koren, Bell and Volinksy (2009) and Dror, Koenigstein and Koren (2011). We choose the regularization parameters by cross validation.

In evaluating the methods, we look at both the quality of their estimates and the time it takes to compute them. We do not include the tuning parameter cross-validation time in the timing results.

We perform two sets of simulations: one for a two-level logistic regression model, and one for a two-level linear regression model. The setup and results for both simulations are similar, so we only include the logistic regression results here. Appendix B contains the linear regression results.

Following the notation in Section 3, we set the number of groups on the first level to $|N_1| = 50$ and the number of groups on the second level (the leaves) to $|N_2| = 500$. We simulate $N$ samples with $N$ ranging from 1000 to 100,000. We set the dimensions of fixed and random effect vectors to $q_0 = q_1 = q_2 = 5$. For each value of $N$, we draw 20 replicates according to the following procedure.

For each replicate, we draw a $q_0$-dimensional fixed effect vector $\beta$ with components $\beta_k, k = 1, \dots, q_0$ drawn independently from a heavy-tailed student's $t$-distribution with four degrees of freedom. We draw random effect covariance matrices $\Sigma_1$ and $\Sigma_2$ independently from an inverse Wishart distribution with shape $I$ and 10 degrees of freedom, scaled by 0.1.

We allocate the $N$ samples to the 50 groups and 500 subgroups in a way that approximates the highly skewed hierarchies in the Book Crossing dataset. In each replicate, we first draw sampling rates $\lambda_1, \dots, \lambda_{500}$ from a Pareto distribution with scale and shape parameters set to 1. Then, we allocate the $N$ samples to the 500 leaf nodes by drawing from a multinomial distribution with probability vector $(\lambda_1, \dots, \lambda_{500}) / \sum_{i=1}^{500} \lambda_i$. Similarly, we allocate the 500 leaf nodes to 50 groups using the same Pareto distribution and sampling scheme.

For every group node in the first level of the hierarchy, $i \in N_1$, we draw a $q_1$-dimensional random effect vector $u_i$ from multivariate Gaussian with mean zero and covariance matrix $\Sigma_1$. For every leaf node $i \in N_2$, we draw a $q_2$-dimensional random effect vector $u_i$ from multivariate Gaussian with mean zero and covariance $\Sigma_2$. Then, we randomly draw fixed effect predictor vectors $x_k$ for sample point $k = 1, \dots, N$ with independent elements taking values $+1$ and $-1$ with probability $1/2$ each. We use the same procedure to randomly draw random effect predictors $z_k$ for every sample point $k$ and let the two levels of the hierarchy share the same random effect predictors. Finally, for every sample $k$ in leaf node $ij \in N_2$, we draw response $y_k$ as Bernoulli with success probability

$$\mu_k = \operatorname{logit}^{-1}\{x_k^T \beta + z_k^T (u_i + u_{ij})\}.$$

To evaluate the quality of the estimators, we use the following loss functions:

- Fixed Effect Loss: $\|\beta - \hat{\beta}\|^2$;
- Random Effect Level-$l$ Covariance Loss: $\operatorname{tr}\{(\hat{\Sigma}_l \Sigma_l^{-1} - I)^2\}$;
- Random Effect Level-$l$ Loss: $|N_l|^{-1} \sum_{i \in N_l} \|\Sigma_l^{-1/2}(u_i - \hat{u}_i)\|^2$;

- Prediction Loss:

$$N^{-1}\sum_{k=1}^{N}\mu_k\log\frac{\mu_k}{\hat{\mu}_k}+(1-\mu_k)\log\frac{1-\mu_k}{1-\hat{\mu}_k},$$

where $\hat{\mu}_k=\operatorname{logit}^{-1}\{x_k^T\hat{\beta}+z_k^T(\hat{u}_i+\hat{u}_{ij})\}$ for sample $k$ in leaf $ij$.

We also measure the overall computation time for each, excluding the cross-validation time for tuning parameter selection.

We compare our method (`mhglm`) with the three other methods described above—`glmer`, `glmer.split` and `sgd`. Figure 2 shows the mean performance for each method, averaged over 20 replicates, with circle radii indicating standard

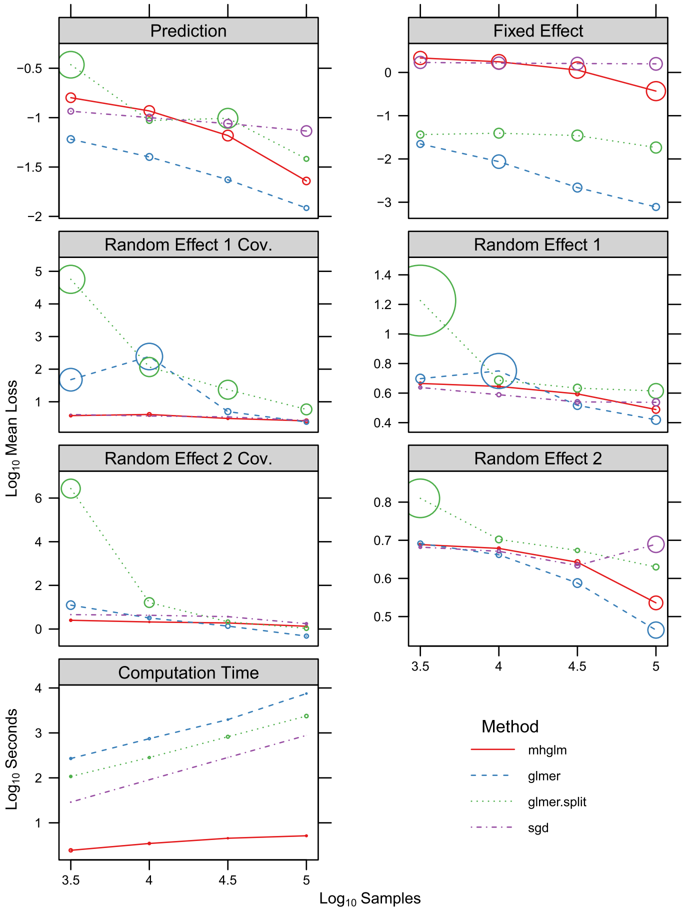

FIG. 2. *Performance for the multilevel logistic regression model. Radii indicate one standard error along y-axis (absent when smaller than line width). The proposed method* (`mhglm`) *is much faster than the three other methods, meanwhile it provides competitive prediction performance.*

errors along the vertical axes. For moderate to large sample sizes, there is a noticeable difference between the proposed method and other maximum likelihood based estimators. However, the proposed method still appears to be consistent, in the sense that its estimators improve with more samples. In terms of prediction loss, our proposed method outperformed both `sgd` and `glmer.split` and is only slightly worse than `glmer`.

The bottom panel compares the computation time for all methods. For large sample sizes, our proposed method is much faster than the other procedures by factors ranging from 100 to 1000, and the factor appears to grow exponentially as sample sizes increase.

In the context of this simulation, our proposed method is able to trade off a modest loss in prediction performance for a dramatic decrease in computation time. We can see that our proposed procedure will scale well to our book recommendation context and to commercial recommendation settings generally.

## 7. Application.

Having developed an estimation procedure for deeply nested hierarchical models in Sections 4 and 5 and having established its suitability in Section 6, we now return to our main application, fitting a model to data that allows us to predict whether or n**a**ot a user would like a book if they had rated it.

Recall from Section 1 that our dataset consists of two parts: a set of user ratings of books, and a hierarchy of these books. We treat each rating as an observation containing book and user identifiers along with a numerical score between one and 10. To accomplish the task of determining a "recommend" or "do not recommend" label and to smooth differences between user-specific rating scales, we binarize the ratings. In particular, we treat numerical scores of eight or above as "positive" and ratings below this as "negative" which gives us a binary classification dataset with balanced classes. We will model these binarized ratings using a hierarchical logistic regression model. We use the user demographic features together with the rating context to construct candidate predictors in the model, linked to the response through fixed and random effects. We use a subset of the book hierarchy for the structure of the hierarchical model.

### 7.1. *Candidate predictors.*

The first set of candidate predictors are converted from user demographic data. We bin users' ages into five groups, $(0, 26]$, $(26, 32]$, $(32, 38]$, $(38, 47]$ and $(47, 101]$, where each group has approximately the same number of ratings. Each age group is represented by one categorical variable. If age is missing, then all five indicators are zero. We aggregate the geographic feature into six groups by continent, North America, Europe, Oceania, Asia, South America and Africa. Similarly, each group is represented by one categorical variable. We have a total of 11 demographic predictors.

Our second set of candidate predictors is the time-varying predictors defined as functions of past user behavior. The first predictor `prev` is a user-specific binary indicator of whether the user's previous rating was positive. This is designed to

TABLE 2
*Predictor associated with one observation from* $\text{user}_i$ *on* $\text{book}_j$

| Predictor | Description |
|---|---|
| $\text{Age}_i$ | User-specific features: a five-component indicator vector for age range $(0, 26]$, $(26, 32]$, $(32, 38]$, $(38, 47]$, $(47, 101]$ |
| $\text{Geographic}_i$ | User-specific features: a six-component indicator vector for continent Africa, Asia, Europe, North America, Oceania, South America |
| $\text{Previous}_i$ | User-specific feature: a smoothed estimate of the log of proportion of positive ratings from $\text{user}_i$: $\log(p_i + 1)/(n_i + 2)$, where $p_i$ and $n_i$ are the number of positive ratings ($\geq 8$) and total number of ratings from $\text{user}_i$. |
| $\text{Distribution}_{ij}$ | User-book-specific feature: a smoothed estimate of the log of proportion of ratings in $\text{book}_j$'s genre from $\text{user}_i$: $\log(k_{ij} + 1)/(n_i + m)$, where $k_{ij}$ is number of ratings $\text{user}_i$ gives in $\text{book}_j$'s genre; $n_i$ is total number of ratings from $\text{user}_i$, and $m$ is total number of genres. |

capture a user's propensity to make positive ratings. The second predictor, `dist`, is user-category specific and computed as the smoothed log proportion of past ratings that the user gives in each category. This is designed to capture a users' tendency to give ratings to each category, revealing his or her relative preference among all categories. Table 2 gives detailed descriptions of all the predictors.

7.2. *Model selection.* We perform two forms of model selection. First, we need to choose which parts of the book hierarchy to use. Second, we need to choose which predictors to use. To carry out the model selection, we randomly partition the data into 80/10/10 percent chunks, for training/development/testing sets. We train various models on the training set, select a model with the best performance on the development set and finally compare the chosen model with other fitting methods on the testing set. In data processing, we only use training data to construct the new features.

We use all predictors for fixed effects, and we can fit these reliably given the large volume of data. However, we do not use all of these predictors on all random effects levels. Because they rely on ratings specific to the particular position in the hierarchy, fitting the random effects is much more difficult. The population structure of the random effects mitigates against some of this data sparsity, but there will still be situations where using coarser hierarchy makes the model less susceptible to overfitting. To guard against overfitting in the random effects terms of the model, we perform model selection by using out-of-sample prediction performance on the development set. To choose the specific subset of the predictors to use as random effects, we fit all possible combinations at all levels of the model, selecting the model with the lowest misclassification rate on the development set.

We start with a depth-1 model. Here we fit depth-1 models with all five possible grouping factors: genre, subgenre, sub-subgenre, author, book. Table 3 lists

TABLE 3
*Best performing model for all choices of grouping factor for one-level model. The standard deviations of the listed model errors are below* 0.004

| Group | Features | Error |
|---|---|---|
| author | geo | 0.3212 |
| book | age, geo | 0.3235 |
| sub-subgenre | prev, dist, age, geo | 0.3282 |
| subgenre | prev | 0.3288 |
| genre | 1 | 0.3291 |

the best one-level model using each grouping factor. We sort the performance by misclassification error on development dataset. We see that using author as the grouping factor and demographic information as the random effect features gives the best prediction performance on development set. Note that we did not get additional performance improvement by using a book-specific random effect model which suggests that we could potentially overfit the data by using too many groups.

To further take advantage of the five nested hierarchies, we also consider depth-2 models. Using the `lme4` modeling notation, we fit all models of the following form:

$$y \sim \texttt{age} + \texttt{geo} + \texttt{prev} + \texttt{dist} + (X_1|g_1) + (X_2|g_1 : g_2),$$

where grouping level $g_2$ is nested under $g_1$, and $X_1$ and $X_2$ are predictor matrices with columns taken from the candidate predictors. The notation indicates that the model has fixed effects corresponding to an intercept and predictors `age`, `geo`, `pref` and `dist`, random effect predictors $X_1$ at the first level, and random effect predictors $X_2$ at the second level.

We list the best performing depth-2 model for every combination of $(g_1, g_2)$ in Table 4 where we sort the performance by misclassification error on development dataset. The feature `1` indicates the feature of all ones (i.e., the intercept term). Note that we decrease the misclassification rate from 0.3212 to 0.3177 by adding an additional hierarchy subgenre on top of author. This improvement in predictive performance may seem small, but in practice such improvements can translate to big impacts when the corresponding models are deployed in commercial scale recommender system applications (Kohavi et al. (2014), Kramer, Guillory and Hancock (2014)). Thus, the small improvement of the two-level model over the one-level model can be meaningful.

The best two-level model is using subgenre and author as the two grouping factors. On subgenre level it uses `dist` and `age` as random effects features; on author level it uses `age` and `geo` as random features. It is a relatively simple model with competitive performance, and we will focus on this model throughout the rest of the paper.

TABLE 4

*Best performing model for all choices of grouping factors for two-level model. The standard deviations of the listed model prediction errors are below* 0.004

| $g_1$ | $g_2$ | $X_1$ | $X_2$ | Error |
|---|---|---|---|---|
| subgenre | author | `dist`, `age` | `age`, `geo` | 0.3177 |
| sub-subgenre | author | `age` | `dist`, `geo` | 0.3184 |
| genre | author | `dist`, `geo` | `age`, `geo` | 0.3189 |
| author | book | `geo` | `dist` | 0.3210 |
| sub-subgenre | book | `dist`, `age`, `geo` | `geo` | 0.3212 |
| subgenre | book | `prev`, `dist`, `age` | `age`, `geo` | 0.3218 |
| genre | book | `age`, `geo` | `age` | 0.3226 |
| subgenre | sub-subgenre | `age`, `geo` | `dist` | 0.3266 |
| genre | sub-subgenre | `dist`, `age` | `dist`, `geo` | 0.3269 |
| genre | subgenre | `prev`, `dist` | `prev` | 0.3287 |

## 8. Results.

8.1. *Performance.* In Section 7.2, the model that gave the best prediction performance on the development set used two levels of hierarchy, corresponding to “author” and “subgenere,” with author nested within subgenre. For fixed effect predictors, the model used an intercept along with `age`, `geo`, `prev` and `dist`. For random effect predictors at the first level in the hierarchy (subgenre), the model used an intercept along with `dist` and `age`; at the second level in the hierarchy (author), the model used an intercept along with `age` and `geo`. We do not include any book-specific features in this model which implies that all books under the same “subgenre $\triangleright$ author” are viewed equally for a given user. When we are recommending books to users under this model, we are in fact recommending authors under specific subgenres.

Having selected the model, we will now evaluate its performance on the held-out test set. We fit the model to the training dataset using our proposed moment-based procedure `mhglm` along with four competing methods: the `glmer` maximum likelihood procedure, the `sgd` stochastic gradient descent $h$-likelihood-based procedure described in Section 6, the `BSLMM` Bayesian sparse linear mixed model procedure by Zhou, Carbonetto and Stephens (2013) and the `arLMM` approximate ridge linear mixed model procedure by Tan et al. (2018). We compare their prediction performances on the held-out test dataset. We do not include the `glmer.split` method because `glmer` fails on the randomly splitted subsets due to sparsity.

Both `BSLMM` and `arLMM` involve computing a kernel matrix on the training data which makes it impossible to fit with all of the 100K+ data points in the training set. As a result, we run `BSLMM` and `arLMM` on 10,000 randomly sampled training data points. Furthermore, `BSLMM` and `arLMM` only work with one-level

TABLE 5
*Misclassification error and running time for five fitting methods*

| Fitting Method | Error | Error 95% Confidence Interval | Time (seconds) |
|---|---|---|---|
| mhglm | 0.3262 | [0.3189, 0.3335] | 55.14 |
| glmer | 0.3268 | [0.3195, 0.3341] | 44,790.23 |
| sgd | 0.3302 | [0.3229, 0.3376] | 2022.35 |
| BSLMM | 0.3325 | [0.3252, 0.3399] | 5355.50 |
| arLMM | 0.3792 | [0.3716, 0.3867] | 39,628.08 |

hierarchical models, thus we keep the second level (author) and skip the first level hierarchy (subgenre) in modeling BSLMM and arLMM. Additionally, since arLMM is designed for regression, we run arLMM with the original book ratings as labels and convert the predicted ratings into $\{0, 1\}$ in the same way as we binarize the true ratings.

Table 5 lists misclassification error, error's 95% confidence intervals and running time for all five methods. Most methods have comparable prediction performance with a misclassification rate of about 32.5%. Our proposed procedure mhglm slightly outperforms the other methods in overall misclassification error, but the difference is not statistically significant. The only exception is arLMM which performs poorly under the misclassification error. (If we evaluate arLMM using the root mean squared error (RMSE) of the predicted ratings, the performance gap is much smaller; arLMM has RMSE of 1.6828, meanwhile mhglm achieves RMSE of 1.6784 from two-level regression in 19.41 seconds.)

When we look at the running time, however, mhglm is faster than sgd by a factor of 45, faster than BSLMM by a factor of 100 and faster than glmer and arLMM by a factor of 1000. Fitting the model using our proposed method took under a minute; fitting using sgd took 33 minutes; fitting using BSLMM took 1.5 hours; fitting using arLMM took 11 hours, and fitting using glmer took 12.4 hours. Note, again, that BSLMM and arLMM use 10 times fewer data than other methods, but the running time is on the same order of magnitude as sgd and glmer respectively.

The results in Table 5 demonstrate two features of the mhglm fitting procedure. First, the prediction performance is comparable to that of the more established likelihood-based procedures. Second, mhglm is faster than these methods by at least an order of magnitude. This reduction in computation time enabled us to perform an exhaustive model selection search over all one- and two-level models. For the four predictors and the intercept, and for the five grouping levels, there were $5 \cdot 2^4 = 80$ 1-level models and $10 \cdot 2^4 \cdot 2^4 = 2560$ two-level models. Extrapolating from the timing results in Table 5, performing the search over these models using mhglm took approximately 40 hours; using sgd, BSLMM, arLMM, or glmer, the same search would take approximately 60 days, 160 days, 3.3 years or 3.75 years, respectively.

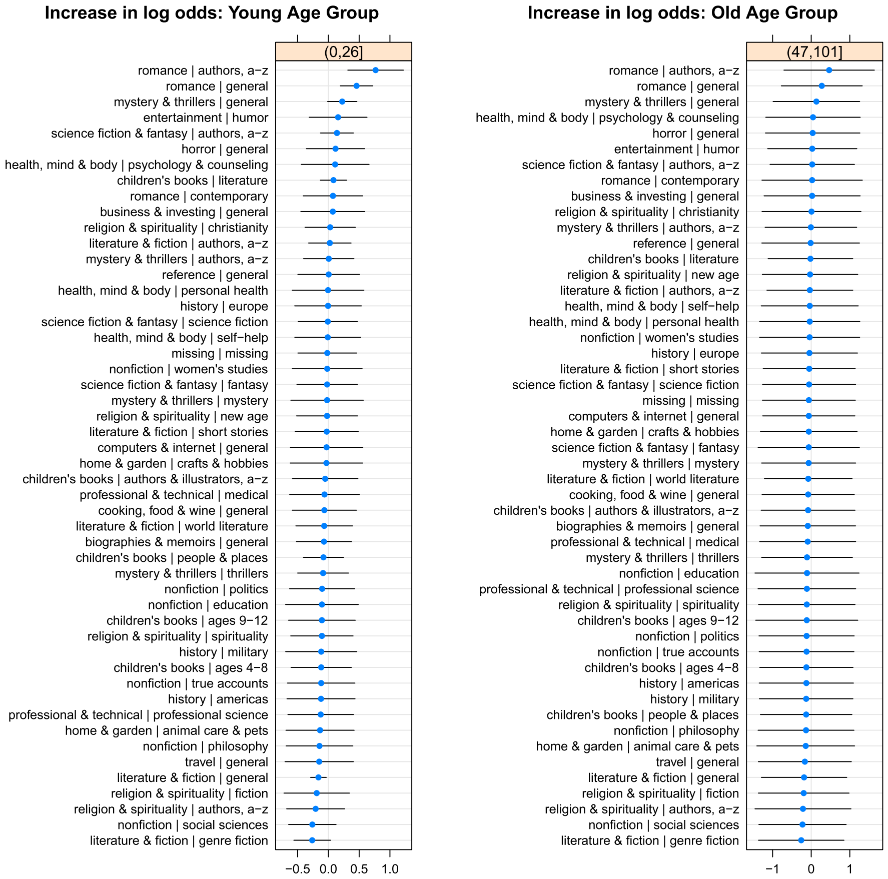

FIG. 3. *Given book subgenre. Everything else remains the same, the increase in log odds if user is young (left panel) or old (right panel). Error-bars show the ± estimated posterior standard deviation. Both figures show the top* 50 *subgenres with the most ratings. Among the young age group (left panel), their most favorite subgenre "romance" has significantly higher random effects than "literature & fiction | genre fiction" which is their least favorite subgenre. The pattern is less clear among the old age group (right panel) due to the large posterior standard deviations.*

8.2. *Fitted model.* To gain some insight into the predictions made by our fitted model, we investigate the empirical Bayes random effect estimates. Specifically, we investigate the age random effects at the subgenre and author levels.

In the context of the fitted model, given a book's subgenre we can compute the increase in log odds of a user liking the book if we change the user's age from "missing" to "known" while keeping all other predictors constant. In Figure 3 we show the change in log odds (±1 estimated posterior standard deviation) for young and old age groups a for the subgenres that have most ratings. For the old age group (47–101 years), the estimates have large estimated posterior standard deviations across the subgenres listed, making it difficult to identify a clear patter. For the young age group (0–26 years) there is some weak but meaningful signal. In this age group, there is a clear pattern in which of the common subgenres the users like and don’t like. “Romance” is their favorite subgenre, which has significantly higher random effects than “literature & fiction | genre fiction,” their least favorite subgenre.

Next, we perform a similar analysis but on the second level of the hierarchy, “author.” For every author we compute the increase in log odds of liking the book if we change the users’ age from “missing” to “known” while everything else remains the same. In Figure 4 we show the increase in log odds (± estimated

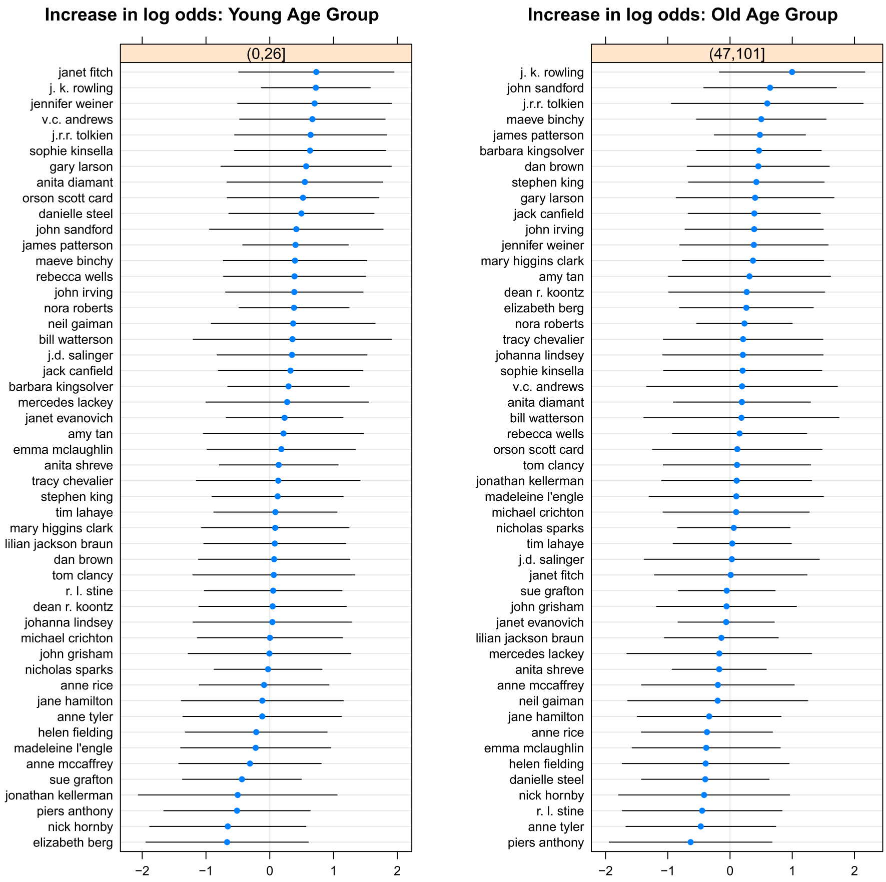

FIG. 4. *Given book author. Everything else remains the same, the increase in log odds if user is young (left panel) or old (right panel). Error-bars show the* ± *estimated posterior standard deviation. Both figures show top* 50 *authors with most ratings. The posterior standard deviations are much larger than that of the parent level (book subgenre). We observe some interesting patterns for authors such as J. K. Rowling (liked by both age groups) and Danielle Steel (opposite opinions from the two age groups).*

posterior standard deviation) for young and old age groups for the authors that have most ratings.

We observe a few interesting patterns:

- The estimated posterior standard deviations are much larger for random effects on the second level (author) for both young and old age groups.
- Some authors are consistent across different age groups. For instance, one would want to recommend J. K. Rowling to both young and age groups. Meanwhile, Nick Hornby and Piers Anthony are liked by neither group.
- Some authors have quite different behaviors across age groups. For instance, Danielle Steel has positive log odds increase if we know the user is within young age group, but negative log odds increase if user is among old age groups. An opposite example is Elizabeth Berg; we will suffer a decrease in log odds if user is young, meanwhile log odds will increase if the user is old.

The size of the estimated posterior standard deviations make clear that these associations are weak. Still, as demonstrated in Sections 7.2 and 8.1, there is enough signal in them to translate to a meaningful reduction in misclassification rate on the held-out development and test sets.

**9. Discussion.** The appeal of the deeply nested hierarchical model is that it facilitates information sharing across subtrees at all levels of the hierarchy. Nodes with abundant data effectively have their random effects estimated using only data at the leaves descending from them. Nodes with little or moderate data, however, benefit by having their estimated coefficients (random effects) shrunk toward the global mean. In our book recommendation application, we have demonstrated this advantage by showing that using two levels of hierarchy (author and subgenre) deliver increased prediction performance than using one or no levels.

The main hurdle in deploying hierarchical models in recommender systems applications like ours, and other contexts of similar scale, is that the time required to fit these models can be prohibitive. Perry (2017) extended a method original due to Cochran (1937) and proposed a partial solution to this problem, but his procedure is limited to single-level hierarchical models. Here, with our proposed `mglhm` method, we have shown how to fit a hierarchical model of arbitrary depth by repeatedly applying the single-level fitting procedure to prune the leaves of the hierarchy. We then showed how to propagate the estimates at the root of the hierarchy down through the nodes in the hierarchy to refine the random effect estimates.

In our book recommendation application, our proposed fitting procedure was faster than stochastic gradient descent by a factor of 45 and faster than the likelihood-based `glmer` procedure by a factor of 1000. This increase in computational speed enabled us to perform an exhaustive model selection search over all one- and two-level models, reducing the overall computation time from about 60 days using `sgd` (or 3.75 years using `glmer`) to about 40 hours. As our simulations in Section 6 demonstrated, the tradeoff in deploying our method is reduced statistical efficiency and prediction performance. However, in our application the loss in prediction performance was negligible.

Although our motivation was a book recommendation system, our proposed fitting procedure is general enough to handle hierarchical generalized linear models of arbitrary depth. We have incorporated our implementation of this procedure into the `mbest` R package, available on the Comprehensive R Archive Network (CRAN). The interface in this implementation is flexible enough to handle any deeply nested hierarchical generalized linear model.

## APPENDIX A: DATA DESCRIPTION

**A.1. Overview.** The BookCrossing dataset introduced in Section 1 contains 433,671 numerical ratings of 185,973 books from 77,805 users (Ziegler et al. (2005)). Each rating consists of a book id (ISBN), a user id, and a numerical score between one and 10, where one indicates extreme negative and 10 indicates extreme positive sentiment. We binarize the ratings so that ratings equal or above *eight* are considered positive and ratings below *eight* are considered negative. The threshold *eight* is chosen such that the two classes have comparable number of samples. We have user demographic information including age and location; Section A.2 reports some descriptive statistics about these features. We also know the book authors and titles.

We augment the book meta-data with a genre hierarchy scraped from Amazon.com by McAuley, Pandey and Leskovec (2015). In this meta-data, book titles are nested within authors within sub-subgenres within subgenres within genres. If the same author writes titles in multiple sub-subgenres, we treat the author as multiple, separate entities. Section A.3 describes the hierarchy in metadata.

In the raw dataset, more than half of the ratings cannot be matched to Amazon metadata. Dealing with this missing data is beyond the scope of the present treatment, so we remove samples with missing ratings or unmatched book IDs from consideration. This leaves us with 157,638 ratings of 38,659 books from 38,085 users.

**A.2. User demographic features.** The reported user age is a continuous variable, ranging from 15 to 100. The mean and standard deviation of user age are 36.4 and 12.6 respectively. Table 6 shows the number of users and ratings from each age range.

Most of the ratings comes from young or middle-aged users which makes it easier to estimate and predict for users from those age ranges.

User's location information is reported as his city, state, country and continent. Table 7 reports the number of users and ratings from the 10 most represented countries, and Table 8 reports the same information for each continent.

We can see that the vast majority of the ratings (85%) are from North America with Europe, the next-most-represented continent, receiving only 6% of ratings. This indicates that it's quite difficult to accurately estimate and predict for users from other than these two continents.

TABLE 6
*Number of users & ratings from each age range*

| Age Interval | # Users | # Ratings |
|---|---|---|
| ≤20 | 2664 | 8752 |
| (20, 30] | 6011 | 30,820 |
| (30, 40] | 5906 | 32,252 |
| (40, 50] | 3765 | 20,539 |
| (50, 60] | 2609 | 11,961 |
| (60, 70] | 952 | 2997 |
| >70 | 297 | 992 |

TABLE 7
*Top* 10 *countries with most ratings*

| Country | # Users | # Ratings |
|---|---|---|
| USA | 29,042 | 120,201 |
| Canada | 3619 | 14,592 |
| United Kingdom | 989 | 3622 |
| Australia | 632 | 2067 |
| Portugal | 181 | 1490 |
| Germany | 381 | 1189 |
| Spain | 187 | 1008 |
| Malaysia | 111 | 964 |
| Netherlands | 178 | 638 |
| New Zealand | 148 | 563 |

TABLE 8
*Number of users & ratings from each continent*

| Continent | # Users | # Ratings |
|---|---|---|
| North America | 32,722 | 135,059 |
| Europe | 2632 | 10,122 |
| Oceania | 780 | 2630 |
| Asia | 430 | 2272 |
| South America | 68 | 210 |
| Africa | 36 | 87 |

**A.3. Book hierarchy.** Every book is nested under a deep hierarchy:

$$\text{genre} \triangleright \text{subgenre} \triangleright \text{sub-subgenre} \triangleright \text{author} \triangleright \text{title}.$$

For example, the book *Harry Potter and the Chamber of Secrets* is nested as *Children's Books* ▹ *Literature* ▹ *Science Fiction, Fantasy, Mystery & Horror* ▹

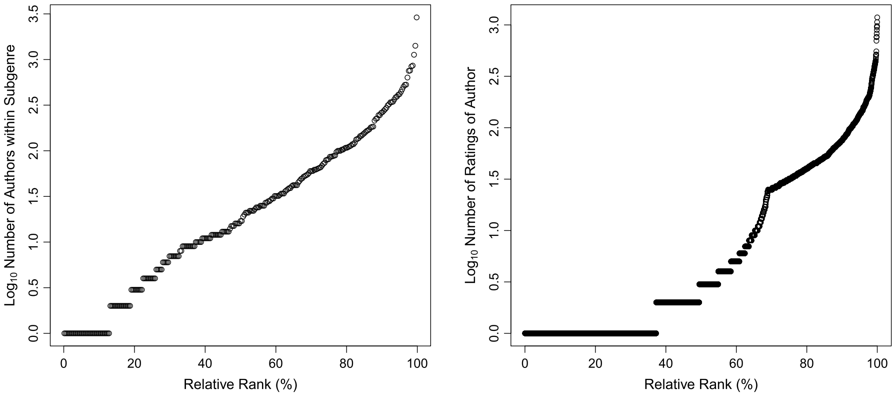

FIG. 5. *Left panel*: *Quantile plot of* $\mathrm{Log}_{10}$ *Number of Authors within Subgenres*. *Right panel*: *Quantile Plot of* $\mathrm{Log}_{10}$ *Number of Ratings of Authors*.

*J. K. Rowling* ▷ *Harry Potter and the Chamber of Secrets*. Figure 1 displays all hierarchies on the first two levels.

For modeling purposes, we only chose two out of five hierarchies. We omit the intermediate levels and use simplified hierarchy of subgenre ▷ author. Our first level of hierarchy subgenre has 1344 groups which captures the necessary amount of diversity across books using a reasonable amount of groups. We use author (nested under subgenre) as the second level of hierarchy which has 27,360 groups. We use book author instead of book title as the second level hierarchy, since the BookCrossing dataset is very sparse, such that most books have only a few number of ratings. Hierarchical models will not work well if most groups have very few samples which shrinks the overall results towards that of a simple "global" model.

Even for these carefully chosen hierarchies, the distribution of subgroups and samples are still highly skewed. We can see this skewness in Figure 5 which plots the quantiles of the number of authors per subgenre (left panel) and the number of ratings per author (right panel). Both plots are on $\log_{10}$ scale. 50% of subgenres have fewer than 17 authors and 90% of subgenres have fewer than 261 authors. At the other extreme, the largest subgenre (Literature & Fiction ▷ General) has 2893 authors. The distribution of ratings among authors are highly skewed as well: 50% of authors have only one rating, 90% of authors have less than nine ratings, meanwhile the mostly rated author (Sue Grafton) received 1183 ratings.

Hierarchical models gain its predictive power by pooling information across groups. The existence of large numbers of small groups will make learning model parameters and making good predictions difficult.

## APPENDIX B: TWO-LEVEL LINEAR MODEL SIMULATIONS

Here we perform a simulation study similar to the two-level logistic regression model study described in Section 6 but using a two-level linear regression model instead.

With all other simulation parameters drawn as described in Section 6, in the linear regression setup we draw response $k$ from a normal distribution with mean $\mu_k = x_k^T \beta + z_k^T (u_i + u_{ij})$ and variance $\phi = 1$ whenever sample $k$ belongs to leaf $ij$. We again compare our procedure with those three methods. We use the same loss for fixed and random effects as well as for the random effect covariance. For prediction loss, we use the mean squared error:

$$N^{-1} \sum_{k=1}^{N} \phi^{-1} (\mu_k - \hat{\mu}_k)^2,$$

where $\mu_k = x_k^T \beta + z_k^T (u_i + u_{ij})$ and $\hat{\mu}_k = x_k^T \hat{\beta} + z_k^T (\hat{u}_i + \hat{u}_{ij})$.

Figure 6 shows the mean loss, averages over 20 replicates, with circle radii indicating standard errors along the vertical axes. For moderate to large sample

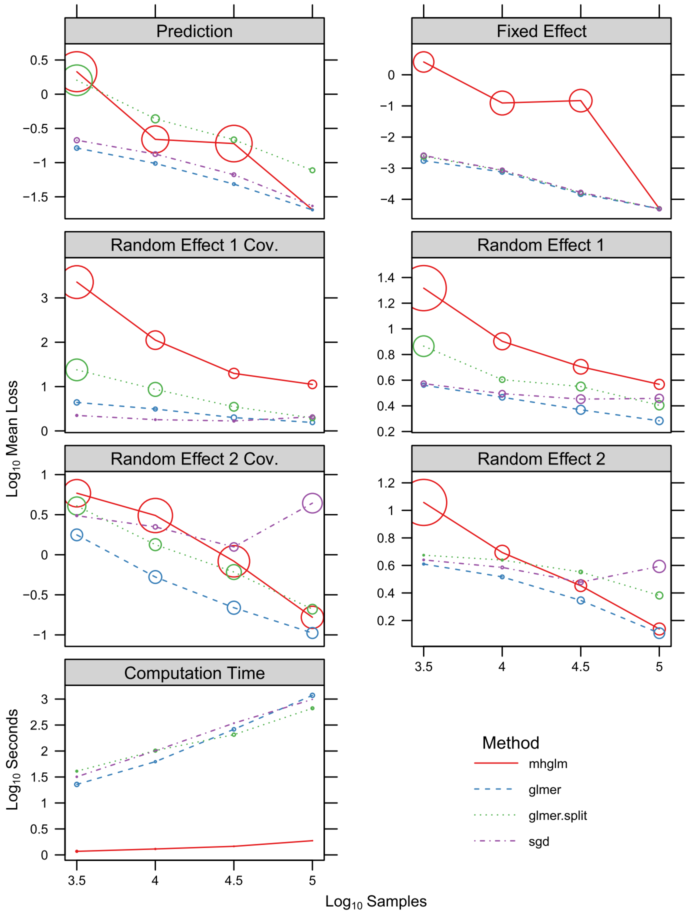

FIG. 6. *Performance for the multilevel linear regression model. Circle radii indicate one standard error along $y$-axis (absent when smaller than line width).*

sizes, there is a noticeable but decreasing difference between the proposed method and other maximum likelihood based estimators. However, the proposed method still appears to be consistent. In terms of computation time, this method again has improvement by factor ranging from 100 to 1000, and the factor appears to grow exponentially as sample sizes increase.

## REFERENCES

ADOMAVICIUS, G. and TUZHILIN, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17** 734–749.

AGARWAL, D. (2008). Statistical challenges in Internet advertising. In *Statistical Methods in e-Commerce Research. Statist. Practice* 3–17. Wiley, Hoboken, NJ. MR2503567

AGARWAL, D. and CHEN, B.-C. (2010). fLDA: Matrix factorization through latent Dirichlet allocation. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining. WSDM* '10 91–100. ACM, New York.

ANSARI, A., ESSEGAIER, S. and KOHLI, R. (2000). Internet recommendation systems. *J. Mark. Res.* **37** 363–375.

BATES, D., MAECHLER, M., BOLKER, B. and WALKER, S. (2013). lme4: Linear mixed-effects models using Eigen and S4. R package version 1.1-7.

COCHRAN, W. G. (1937). Problems arising in the analysis of a series of similar experiments. *Suppl. J. R. Stat. Soc.* **4** 102–118.

CONDLIFF, M. K., LEWIS, D. D. and MADIGAN, D. (1999). Bayesian mixed-effects models for recommender systems. In *ACM SIGIR* '99 *Workshop on Recommender Systems: Algorithms and Evaluation*.

DROR, G., KOENIGSTEIN, N. and KOREN, Y. (2011). Yahoo! Music recommendations: Modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the Fifth ACM Conference on Recommender Systems* 165–172. ACM, New York.

FIRTH, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80** 27–38. MR1225212

GAO, K. and OWEN, A. B. (2016). Estimation and inference for very large linear mixed effects models. ArXiv E-prints.

GAO, K. and OWEN, A. (2017). Efficient moment calculations for variance components in large unbalanced crossed random effects models. *Electron. J. Stat.* **11** 1235–1296. MR3635913

GEBREGZIABHER, M., EGEDE, L., GILBERT, G. E., HUNT, K., NIETERT, P. J. and MAULDIN, P. (2012). Fitting parametric random effects models in very large data sets with application to VHA national data. *BMC Med. Res. Methodol.* **12** 1–14.

HUANG, Z. and GELMAN, A. (2005). Sampling for Bayesian computation with large datasets. Unpublished.

KOHAVI, R., DENG, A., LONGBOTHAM, R. and XU, Y. (2014). Seven rules of thumb for web site experimenters. In *Proceedings of the* 20*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1857–1866. ACM, New York.

KOREN, Y., BELL, R. and VOLINKSY, C. (2009). Matrix factorization techniques for recommender systems. *Computer* **42** 30–37.

KRAMER, A. D., GUILLORY, J. E. and HANCOCK, J. T. (2014). Experimental evidence of massive-scale emotional contagion through social networks. *Proc. Natl. Acad. Sci. USA* **111** 8788–8790.

MCAULEY, J., PANDEY, R. and LESKOVEC, J. (2015). Inferring networks of substitutable and complementary products. In *Proceedings of the* 21*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 785–794.

NAIK, P., WEDEL, M., BACON, L., BODAPATI, A., BRADLOW, E., KAMAKURA, W., KREULEN, J., LENK, P., MADIGAN, D. M. et al. (2008). Challenges and opportunities in high-dimensional choice data analyses. *Mark. Lett.* **19** 201–213.

PERRY, P. O. (2017). Fast moment-based estimation for hierarchical models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **79** 267–291. MR3597973

SCOTT, S. L., BLOCKER, A. W., BONASSI, F. V., CHIPMAN, H. A., GEORGE, E. I. and MCCULLOCH, R. E. (2013). Bayes and big data: The consensus Monte Carlo algorithm. In *Bayes* 250.

TAN, Z., ROCHE, K., ZHOU, X. and MUKHERJEE, S. (2018). Scalable algorithms for learning high-dimensional linear mixed models. Available at arXiv:1803.04431.

WENG, L. T., XU, Y., LI, Y. and NAYAK, R. (2008). Exploiting item taxonomy for solving cold-start problem in recommendation making. In 2008 20*th IEEE International Conference on Tools with Artificial Intelligence* **2** 113–120.

ZHANG, Y., CAO, B. and YEUNG, D.-Y. (2010). Multi-domain collaborative filtering. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. *UAI*'10 725–732. AUAI Press, Arlington, VA.

ZHANG, X., ZHOU, Y., MA, Y., CHEN, B.-C., ZHANG, L. and AGARWAL, D. (2016). GLMix: Generalized linear mixed models for large-scale response prediction. In *Proceedings of the* 22*nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. *KDD* '16 363–372. ACM, New York.

ZHOU, X., CARBONETTO, P. and STEPHENS, M. (2013). Polygenic modeling with Bayesian sparse linear mixed models. *PLoS Genet.* **9** e1003264.

ZIEGLER, C.-N., MCNEE, S. M., KONSTAN, J. A. and LAUSEN, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the* 14*th International Conference on World Wide Web* 22–32. ACM, New York.

N. ZHANG
DEPARTMENT OF TECHNOLOGY,
OPERATIONS, AND STATISTICS
STERN SCHOOL OF BUSINESS
NEW YORK UNIVERSITY
44 WEST 4TH ST.
NEW YORK CITY, NEW YORK 10012
USA
E-MAIL: nzhang@stern.nyu.edu

K. SCHMAUS
STITCH FIX
1 MONTGOMERY TOWER
SAN FRANCISCO, CALIFORNIA 94104
USA
E-MAIL: kschmaus@stichfix.com

P. O. PERRY
OSCAR HEALTH
295 LAFAYETTE ST, 6TH FLOOR
NEW YORK CITY, NEW YORK 10012
USA
E-MAIL: pperry@hioscar.com