Received: 4 August 2021 | Revised: 9 October 2021 | Accepted: 7 November 2021

DOI: 10.1002/sam.11563

WILEY

RESEARCH ARTICLE

Bag of little bootstraps for massive and distributed longitudinal data

Xinkai Zhou[1] | **Jin J. Zhou**[2] | **Hua Zhou**[1,3]

[1]Department of Biostatistics, University of California, Los Angeles, California, USA

[2]Department of Medicine, University of California, Los Angeles, California, USA

[3]Department of Computational Medicine, University of California, Los Angeles, California, USA

Correspondence
Hua Zhou, Department of Biostatistics, University of California, Los Angeles, CA, USA.
Email: huazhou@ucla.edu

Funding information
Division of Mathematical Sciences, Grant/Award Number: DMS-2054253; National Heart, Lung, and Blood Institute, Grant/Award Number: HL150374; National Human Genome Research Institute, Grant/Award Number: HG006139; National Institute of Diabetes and Digestive and Kidney Diseases, Grant/Award Number: DK106116; National Institute of General Medical Sciences, Grant/Award Number: GM141798

Abstract

Linear mixed models are widely used for analyzing longitudinal datasets, and the inference for variance component parameters relies on the bootstrap method. However, health systems and technology companies routinely generate massive longitudinal datasets that make the traditional bootstrap method infeasible. To solve this problem, we extend the highly scalable bag of little bootstraps method for independent data to longitudinal data and develop a highly efficient Julia package `MixedModelsBLB.jl`. Simulation experiments and real data analysis demonstrate the favorable statistical performance and computational advantages of our method compared to the traditional bootstrap method. For the statistical inference of variance components, it achieves 200 times speedup on the scale of 1 million subjects (20 million total observations), and is the only currently available tool that can handle more than 10 million subjects (200 million total observations) using desktop computers.

KEYWORDS

bags of little bootstraps, big data, EMR, linear mixed models, longitudinal data, parallel and distributed computing

1 | INTRODUCTION

Linear mixed models (LMMs) are powerful tools for analyzing longitudinal data, which are ubiquitous in medical research and E-commerce applications. For example, electronic medical records (EMR) data contains longitudinal measurements from the same patient over time. However, there are two challenges in applying LMMs to today's problems. The first one is the massive sample size of modern datasets. For instance, the UCLA Health System alone has over 2.5 million *annual* patient visits. Analyzing such datasets with LMMs is challenging, especially if the goal is to make statistical inference on the variance component parameters. For example, to test if subjects have different slopes for a covariate, one needs to test whether the corresponding random effect has zero variance. Statistical tests based on asymptotics are dubious because the limiting distribution of random effect parameters is difficult to derive. Therefore, researchers rely on the bootstrap method [7], which eliminates the need for asymptotics, but is computationally intensive. Specifically, running the traditional bootstrap method on LMMs has a computational cost of $O\left(BNq^3\right)$, where B is the number of bootstrap replicates, N is the number of subjects, and q is the number of random effect parameters. When N is on the scale of millions, the bootstrap method is prohibitively slow.

The second challenge relates to distributed datasets. Modern datasets are often stored at multiple locations:

internet companies that harvest large volumes of data store them across data centers worldwide to save data transfer costs; medical centers that collaborate in multisite studies try to avoid sending data over the internet due to security and privacy concerns. However, to fit LMMs and use the traditional bootstrap method, one has to either move the distributed datasets to one place or communicate model parameters and their derivatives continuously between data centers, which incur high data transfer costs.

To overcome these challenges, we extend the bag of little bootstraps (BLB) method [12] to the longitudinal data setting. It has a computational cost of $O\left(Bbq^3\right)$ where $b \ll N$, so it is capable of fitting and making statistical inference of LMMs on massive longitudinal datasets using a fraction of the time compared with the traditional bootstrap method. Moreover, by using the BLB framework, our software, MixedModelsBLB.jl, provides a solution to the analysis of distributed longitudinal datasets.

2 | METHOD

2.1 | Model and notation

Given a longitudinal dataset with N independent clusters (the word "cluster" is used interchangeably with "subjects" in this paper), let $\boldsymbol{y}_i \in \mathbb{R}^{n_i}$ be the observed response vector of length n_i from subject i, and $\boldsymbol{X}_i \in \mathbb{R}^{n_i \times p}$ and $\boldsymbol{Z}_i \in \mathbb{R}^{n_i \times q}$ be the observed covariates for the fixed and random effect parameters, respectively. Consider an LMM of the form

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \epsilon_i, \tag{1}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ denotes the fixed effect parameters, $\boldsymbol{b}_i \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ denotes the random effect for the i-th subject, $\boldsymbol{\Sigma}$ is a $q \times q$ covariance matrix, and $\epsilon_i \sim N\left(\boldsymbol{0}, \sigma_0^2 \boldsymbol{I}_{n_i}\right)$ denotes the random error. $\boldsymbol{b}_i$ and ϵ_i are jointly independent. $\boldsymbol{\Sigma}$ and σ_0^2 are the variance component parameters.

2.2 | Statistical inference for LMMs

For fixed effect parameters, statistical inference is usually based on the asymptotic distribution of $\widehat{\boldsymbol{\beta}}$. This approach relies on approximations that may not be accurate when the data are unbalanced or when the residuals have nonconstant variance [1, 9]. For distributed datasets, the asymptotic approach is difficult to implement and is potentially costly because it involves transferring parameters and their derivatives between different data centers.

Statistical inference of variance component parameters is more challenging. For testing if a random effect should be included in the model, one needs to test the hypothesis that the corresponding random effect variance equals zero. Since zero lies on the boundary of the parameter space of variance, the usual regularity condition that the parameter should be an interior point of the parameter space is not met. Testing such hypotheses involves using complex asymptotic or exact null distributions [5, 6, 14], which makes it cumbersome to use in practice.

Following the notation in References [12, 16], let $\boldsymbol{w}_i = \left(\boldsymbol{y}_i, \boldsymbol{X}_i, \boldsymbol{Z}_i\right) \sim P$ be independent and identically distributed (IID) for $i = 1, \ldots, N$, and let the corresponding empirical distribution be $\mathbb{P}_N = N^{-1}\sum_{i=1}^N \delta_{\boldsymbol{w}_i}$. $\theta(P) = \left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \sigma_0^2\right)$ denotes all model parameters, $\widehat{\theta}_N = \widehat{\theta}_N\left(\mathbb{P}_N\right)$ is an estimate of $\theta(P)$. In its essence, statistical inference of $\widehat{\theta}_N = \widehat{\theta}_N\left(\mathbb{P}_N\right)$ is a summary, denoted by $\xi\left\{Q_N(P)\right\}$, of the distribution $Q_N(P)$ of $u\left(\mathbb{P}_N, P\right)$, which is a function of $\widehat{\theta}_N$ and its form depends on our inferential goal. For example, if we want to quantify the variance of $\widehat{\theta}_N$, then $u\left(\mathbb{P}_N, P\right) = \widehat{\theta}_N$ and ξ is the variance. In practice, since P and $Q_N(P)$ are unknown, we cannot calculate $\xi\left\{Q_N(P)\right\}$ directly, but we can estimate it using the observed dataset. The asymptotic approach is one way to perform the estimation where we replace $Q_N(P)$ with the asymptotic distribution of $\theta(P)$. An alternative approach is the bootstrap method [7], which replaces $Q_N(P)$ by its bootstrap approximation.

Given IID data $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N$ and its empirical distribution $\mathbb{P}_N$, the bootstrap method first samples N data points with replacement from $\mathbb{P}_N$, which has empirical distribution function $\mathbb{P}_N^*$. From the bootstrap sample, $u\left(\mathbb{P}_N^*, \mathbb{P}_N\right)$ can be calculated. This process is repeated many times to obtain $\mathbb{Q}_N^*$, which is the empirical distribution of the u's and serves to approximate $Q_N(P)$. Finally, we use $\xi\left(\mathbb{Q}_N^*\right)$ as an estimate of $\xi\left\{Q_N(P)\right\}$.

However, the bootstrap method is computationally expensive for large datasets, especially for longitudinal data. In addition, it is awkward to apply the bootstrap method to distributed datasets because resampling requires access to the full data. To solve these problems, we extend the BLB method [12], which was developed for cross-sectional data, to the longitudinal data setting.

Given a longitudinal dataset with N clusters and a subset size $b < N$, the BLB method first samples s subsets, each consisting of b clusters. The sampling is done without replacement and uniformly at random. Let $I_1, \ldots, I_s \subset \{1, \ldots, N\}$ denote the clusters that are in each subset, where $|I_j| = b \text{for} 1 \le j \le s$. Further let $\mathbb{P}_{N,b}^{(j)} = b^{-1}\sum_{i \in I_j} \delta_{\mathbf{w}_i}$ denote the empirical distribution for subset j. Then, for each subset, it samples N clusters with replacement to obtain the bootstrap sample and calculates $u\left(\mathbb{P}_{N,b}^*, \mathbb{P}_{N,b}^{(j)}\right)$, where $\mathbb{P}_{N,b}^*$ denotes the empirical distribution of the bootstrap sample. Resampling is repeated B times and the empirical distribution of the u-values on subset j is denoted by $\mathbb{Q}_{N,j}^*$. Finally, BLB estimate of $\xi\left\{Q_N(P)\right\}$ is

given by

$$s^{-1}\sum_{j=1}^{s}\xi\left(\mathbb{Q}_{N,j}^{*}\right),$$

where $\xi\left(\mathbb{Q}_{N,j}^{*}\right)$ serves as an approximation of $\xi\left\{Q_N\left(\mathbb{P}_{N,b}^{(j)}\right)\right\}$.

The fact that BLB operates on subsets rather than the entire dataset confers two advantages. First, it is more amenable to parallel processing than the bootstrap method. Since each subset is much smaller than the full dataset, we can parallelize at the subset level such that multiple CPU cores can work on multiple subsets at the same time. Secondly, to analyze datasets stored at multiple data centers, BLB can treat each data center as a subset or take further subsets at each data center, perform analysis on each subset, and obtain the final statistical inference by aggregating parameter estimates from different data centers. Since the final parameter estimates are all we need to transfer between data centers, BLB avoids moving raw data over the internet and incurs minimal communication costs. In contrast, the bootstrap method requires that we either move distributed datasets to one place, which poses security and privacy concerns, or communicate large amounts of intermediate parameter estimates and their derivatives, which incurs high communication costs. We note that in order for BLB to work in distributed data settings, one needs to be comfortable with the assumption that subjects from different data centers are IID samples from the population of interest. When certain variables demonstrate spatial heterogeneity, we expect more variability in the corresponding estimates; see Section S3 in supporting information for a simulation experiment.

Another feature of BLB is the way it generates bootstrap samples. Given a subset with b clusters, it samples N clusters ($N > b$) with replacement to form a bootstrap sample. Doing so offers three advantages. First, it makes BLB automatic in the sense that re-scaling of the resulting estimates is not needed because the u-values are calculated on datasets that are of the same size as the original data. This contrasts to methods such as subsampling [13] and M out of N bootstrap [3]. Both methods estimate parameters on datasets that are smaller than the original data, and thus require re-scaling the estimates. The second advantage is that storing BLB resamples requires $O(b)$ rather than $O(N)$ memory because each resample has its support on b distinct clusters. In fact, resampling N clusters from b clusters amounts to generating a weight vector from an N-trial uniform multinomial distribution over b objects, so each resample can be compactly represented by b clusters and a length-b vector denoting the number of repeats of each cluster. The third advantage is that for estimators that can work with a weighted data representation, the computational time using BLB resamples scales as $O(b)$ rather than $O(N)$. Many commonly used estimators, including maximum likelihood estimators (MLE) and general M-estimators, fall into this category. This means that we can use either MLE or generalized estimating equations (GEE) to estimate model parameters. Finally, BLB for longitudinal data enjoys the same consistency and higher-order correctness guarantee as BLB for IID data. Theoretical analysis of BLB is similar to that of bootstrap and follows from standard empirical process results. Using weak convergence of the bootstrapped empirical process [Reference 16, theorem 3.6.3], Kleiner et al. [12] showed that size n resamples from $\mathbb{P}_{N,b}^{(j)}$ behave asymptotically as if they were drawn directly from P. This together with the delta method for bootstrap [Reference 17, theorem 23.9] yields the consistency of each individual $\xi\left\{Q_N\left(\mathbb{P}_{N,b}^{(j)}\right)\right\}$ as $b, n \to \infty$. Consistency of BLB is then obtained by using the continuous mapping theorem [17]. This analysis assumes that the sampling units are IID, which is satisfied in the longitudinal setting because our sampling units are clusters and we assume that clusters are IID. Similar arguments can be made for the proof of higher-order correctness.

Consistency and higher-order correctness of BLB for longitudinal data hold for estimators that are Hadamard differentiable. Since M-estimators are generally Hadamard differentiable [16, 17] and both MLE and GEE produce M-estimators, these theoretical properties hold with either MLE or GEE.

In the following sections we present results obtained by MLE. GEE results, which are implemented through an approach called WiSER [8], are presented in Section S4 in supporting information.

3 | COMPUTATIONAL STRATEGY

A key component of Algorithm 1 is fitting LMMs, and we do so by maximizing the log-likelihood using the Fisher scoring algorithm. For model (1), the log-likelihood for the i-th cluster is

$$\begin{aligned}\ell_i = &-\frac{n_i}{2}\log(2\pi) - \frac{1}{2}\text{logdet}\left(\boldsymbol{Z}_i\boldsymbol{\Sigma}\boldsymbol{Z}_i' + \sigma_0^2\boldsymbol{I}_{n_i}\right)\\ &-\frac{1}{2}\left(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}\right)'\left(\boldsymbol{Z}_i\boldsymbol{\Sigma}\boldsymbol{Z}_i' + \sigma_0^2\boldsymbol{I}_{n_i}\right)^{-1}\left(\boldsymbol{y}_i - \boldsymbol{X}_i\boldsymbol{\beta}\right).\end{aligned}$$

Identifying a good starting point is crucial for fast convergence. In practice, we initialize $\boldsymbol{\beta}$ and σ_0^2 with least-squares solutions

$$\boldsymbol{\beta}^{(0)} = \left(\sum_i \boldsymbol{X}_i^T\boldsymbol{X}_i\right)^{-1}\left(\sum_i \boldsymbol{X}_i^T\boldsymbol{y}_i\right)$$

Algorithm 1. BLB for LMM.

Input: Clustered data $\boldsymbol{w}_1, ..., \boldsymbol{w}_N$; b: number of clusters in the subset; s: number of subsets; r: number of bootstrap samples within each subset; u: estimate of LMM parameters; ξ: summary of the distribution of u
Output: An estimate of $\xi\{Q_N(P)\}$

for $j \in 1$ *to* s **do**
  Randomly sample a set $I = \{i_1, ..., i_b\}$ of b indices without replacement from $\{1, ..., N\}$
  // Empirical distribution of the j-th subset
  $\mathbb{P}^{(j)}_{N,b} \leftarrow b^{-1} \sum_{i \in I} \delta_{\boldsymbol{w}_i}$
  // Approximate $\xi\{Q_N(\mathbb{P}^{(j)}_{N,b})\}$ by $\xi(\mathbb{Q}^*_{N,j})$
  for $k \in 1$ *to* r **do**
    Sample $(n_1, ..., n_b) \sim \text{Mult}(N, \mathbf{1_b}/b)$
    // Empirical distribution of the BLB re-sample
    $\mathbb{P}^*_{N,k} \leftarrow N^{-1} \sum_{l=1}^{b} n_l \delta_{\boldsymbol{w}_{i_l}}$
    Fit model using MLE or GEE on the re-sample to get $u^*_{N,k} \leftarrow u(\mathbb{P}^*_{N,k}, \mathbb{P}^{(j)}_{N,b}) = \hat{\theta}_N(\mathbb{P}^*_{N,k})$.
  end
  // Empirical distribution of the u-values on subset j
  $\mathbb{Q}^*_{N,j} \leftarrow r^{-1} \sum_{k=1}^{r} \delta_{u^*_{N,k}}$
  $\xi^*_{N,j} \leftarrow \xi(\mathbb{Q}^*_{N,j})$
end
// The BLB estimate of $\xi\{Q_N(P)\}$ averages $\xi^*_{N,j}$ from subsets
Return $s^{-1} \sum_{j=1}^{s} \xi(\mathbb{Q}^*_{N,j})$

$$\sigma_0^2 = \left(\sum_i \boldsymbol{r}_i^{(0)T} \boldsymbol{r}_i^{(0)} \right) / \left(\sum_i n_i \right),$$

where $\boldsymbol{r}_i^{(0)} = \boldsymbol{y}_i - \boldsymbol{X}_i \boldsymbol{\beta}^{(0)}$. To initialize $\boldsymbol{\Sigma}$, we minimize

$$\sum_i \| \boldsymbol{r}_i^{(0)} \boldsymbol{r}_i^{(0)T} - \boldsymbol{Z}_i \boldsymbol{\Sigma} \boldsymbol{Z}_i^T \|_{\mathrm{F}}^2,$$

which gives

$$\text{vec}\, \boldsymbol{\Sigma}^{(0)} = \left(\sum_i \boldsymbol{Z}_i^T \boldsymbol{Z}_i \otimes \boldsymbol{Z}_i^T \boldsymbol{Z}_i \right)^{-1} \left(\sum_i \boldsymbol{Z}_i^T \boldsymbol{r}_i^{(0)} \otimes \boldsymbol{Z}_i^T \boldsymbol{r}_i^{(0)} \right).$$

Besides a good starting point, we also need to evaluate the gradient and the Fisher information matrix efficiently by exploiting structures in these quantities. For example, by using the Woodbury structure in the marginal covariance $\boldsymbol{Z}_i \boldsymbol{\Sigma} \boldsymbol{Z}_i' + \sigma_0^2 \boldsymbol{I}_{n_i}$, we can avoid the storage and decomposition of potentially large $n_i \times n_i$ matrices. See Section S2 in supporting information for detailed derivation and the implementation strategy.

4 | SOFTWARE

Our implementation, `MixedModelsBLB.jl`, is an open-source Julia package available at https://github.com/xinkai-zhou/MixedModelsBLB.jl. Users can run the software on Julia v1.5 or later, or use Docker without installing Julia. The package is compatible with a wide range of data inputs, including data frames and datasets that are too large to fit in memory. Furthermore, it works with a variety of nonlinear programming solvers such as Ipopt [18], NLopt [11], and KNITRO [4]. Finally, when the user has access to multiple CPU cores, parallel processing can be turned on to gain further efficiency by processing BLB subsets simultaneously.

We illustrate it on the `sleepstudy` example data [2]. The BLB estimates and the confidence intervals are printed. In addition, parameter estimates from all iterations are returned in an object of type `blbEstimates` for further analyses. See https://github.com/xinkai-zhou/MixedModelsBLB.jl for detailed documentation.

5 | SIMULATION STUDY

This section presents two simulation experiments. The first one compares the statistical performance between BLB and bootstrap. The second simulation applies BLB to ultra large data sets to demonstrate its scalability.

In the first simulation, we define the relative error of the confidence intervals as $|c - c_0|/c_0$, where c is the estimated confidence interval width and c_0 is the true confidence interval width. We then compare the relative error of the confidence intervals between BLB and the bootstrap method. To calculate c_0, we generate 1000 datasets of size N from the underlying data generating distribution P, compute $\widehat{\theta}_N$ on each of them, and use these estimates to calculate confidence intervals and c_0. To calculate c, we simulate one dataset of size N from P, run BLB and bootstrap, and record the parameter estimates as well as the cumulative processing time (after each bootstrap resample or BLB subset has been processed). To reduce the variation in c induced by a particular dataset, we repeat this process on five simulated datasets and average the resulting relative errors and processing times. We present the trajectory of relative error versus time, where the relative error is averaged over variance components parameters. Note that the time axis provides a single-number summary of parameters b (subset size), s (number of subsets), and r (number of bootstrap iterations on a given subset) for BLB, and of r (number of bootstrap iterations) for bootstrap. We used our package `MixedModelsBLB.jl` for BLB and the `MixedModels.jl` package for bootstrap. Parallel

Listing 1. Illustrating software usage on the `sleepstudy` data.

```
using MixedModelsBLB, JuliaDB, StatsModels, Random
datatable = JuliaDB.loadtable("test/data/sleepstudy.csv")
blb_ests = blb_full_data(
        MersenneTwister(1),
        datatable;
        feformula    = @formula(Reaction ~ 1 + Days),
        reformula    = @formula(Reaction ~ 1),
        id_name      = "id",
        cat_names    = Array{String,1}(),
        subset_size = 10,
        n_subsets    = 20,
        n_boots      = 500,
        solver       = Ipopt.IpoptSolver(print_level=0),
        verbose      = false,
        nonparametric_boot = true
    )
#Bag of Little Boostrap (BLB) for linear mixed models.
#Number of subsets: 20
#Number of grouping factors per subset: 10
#Number of bootstrap samples per subset: 500
#Confidence interval level: 95%

Variance Components parameters

             Estimate  CI Lower  CI Upper

(Intercept)  1202.18    426.24   2087.30
Residual      826.92    513.64   1180.74


Fixed-effect parameters

             Estimate   CI Lower  CI Upper

(Intercept)  250.88     237.66    263.59
Days          10.79       8.11     13.50
```

processing was turned off for both methods because the primary focus of this experiment is statistical performance.

We generate data under two settings. In the first one, non-intercept entries of $\boldsymbol{X}_i, \boldsymbol{Z}_i$, and ϵ_i are drawn independently from the standard normal distribution. In the second one, $\boldsymbol{X}_{i_k,j} \sim \Gamma(1+5(j-1)/(p-1),2) - 2\Gamma(1+5(j-1)/(p-1),2)$, $\boldsymbol{Z}_{i_k,j} \sim \Gamma(1+5(j-1)/(q-1),2) - 2\Gamma(1+5(j-1)/(q-1),2)$, and $\epsilon_{i_k} \sim \Gamma(1,2) - 2$ independently for $k = 1, \ldots, n_i, j = 1, \ldots, p$. In both settings, $N = 20{,}000, n_i = 10$ for all i, $p = 100$, and $q = 2$. For BLB, we set the subset size to be $b = N^\gamma$ where $\gamma \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, and the number of Monte Carlo iterations to be $r = 200$.

Figure 1 shows the results. For all subset sizes, BLB converges to low relative error faster than bootstrap. When the subset size is small ($\gamma = 0.5, 0.6, 0.7$), it takes a very short time for BLB to process each subset, and it takes no more than 10–20 subsets for BLB to reach low relative error (each hinge corresponds to a subset for BLB). When the subset size is larger ($\gamma = 0.8, 0.9$), it takes longer to process each subset, but only a small number of subsets (3–5) is needed to achieve low relative error.

Besides the comparison with bootstrap, we also examined the subsampling method [13] as an alternative. However, we observed similar divergence in relative error for smaller subset sizes as reported by Reference [12]. See Section S5 in supporting information for more details.

The second simulation experiment compares the scalability of BLB and bootstrap. Since data generating distributions do not affect scalability, we only consider the standard normal case. We choose $N = 1$ million, $n_i = 20$, $p = 20$, and $q = 2$. The truth is obtained by simulating 200 instead of 1000 datasets due to the bigger sample size. For bootstrap, we set the number of Monte Carlo iterations $r = 400$. For BLB, we set $b = N^{0.6} \approx 3981, s = 10$, and $r = 200$. For both procedures, we turn on parallel processing. Specifically, BLB uses 10 worker nodes and bootstrap uses two threads. We cannot use 10 threads for bootstrap because it makes a copy of the model object and the bootstrap sample on each thread, so it would quickly exhaust the memory on our computer (64 GB) if we use more than two threads. Figure 2 shows the simulation result. We see that BLB finishes all calculations within 170 s, which is more than 200 times faster than bootstrap, and achieves lower relative error (0.0534 vs. 0.0603, or an 11% reduction). A rough calculation shows that even if our computer has more memory (> 300 GB) so that bootstrap can run

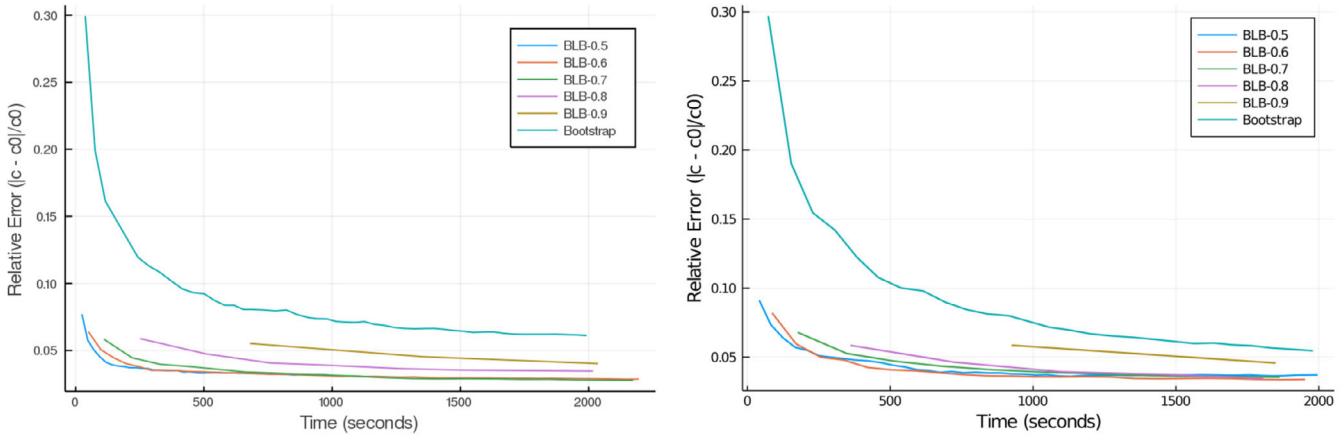

FIGURE 1 Relative error versus processing time for BLB and bootstrap under normal (left) and gamma (right) data generating distributions

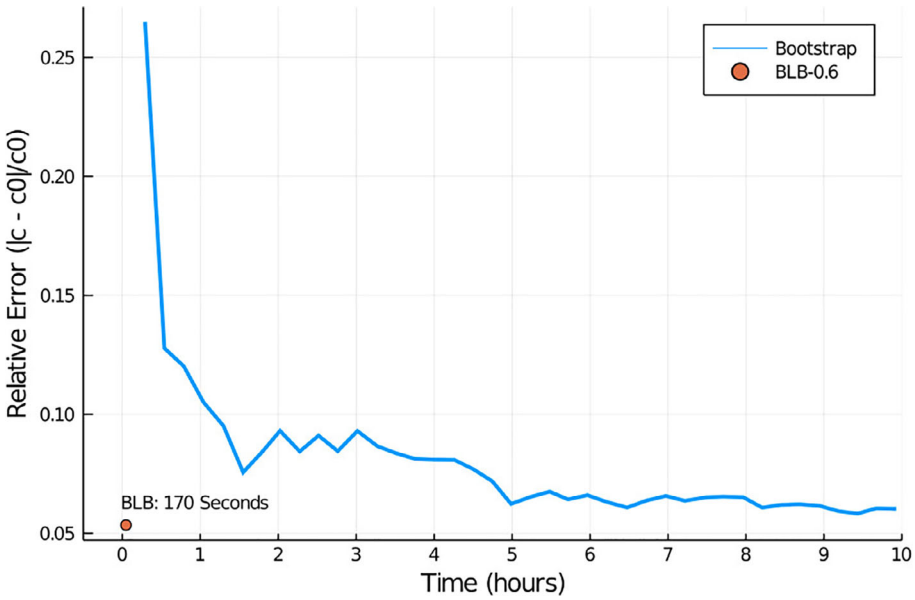

FIGURE 2 Relative error versus processing time on $N = 1$ million subjects and 20 million total observations. BLB subset size was set to $b = N^{0.6} \approx 3981$

with 10 threads, it would still take 2 h and thus be much slower than BLB.

To see how BLB compares with bootstrap on even larger data sets, we simulated a data set with $N = 10$ million, $n_i = 20$, $p = 20$, and $q = 2$ using the same data generating distribution as above. The entire data set contains 200 million records, and the CSV file takes 79 GB disk space. For BLB, we set $b = N^{0.6} \approx 15850, s = 10$, and $r = 200$. BLB finishes all computation within 22 min. On the other hand, since the data set exceeds our computer's memory limit, we are unable to run bootstrap.

Besides these two experiments, we also examined the relationship between the number of bootstrap samples on each subset (r) and relative error; see Section S6 in supporting information for details.

6 | REAL DATA

In this section, we apply `MixedModelsBLB.jl` to the Action to Control Cardiovascular Risk in Diabetes trial (ACCORD) dataset [10]. The ACCORD study examined whether the intensive therapy that targets normal glycated hemoglobin (HbA1c) levels (< 6.0%) would reduce cardiovascular events when compared with the standard therapy among patients with type 2 diabetes who had either established cardiovascular disease (CVD) or additional cardiovascular risk factors. A total of 12,251 patients aged 40–79 years participated; their glucose concentrations were measured every 4 months in the initial year and then annually up to a maximum of 84 months.

After data cleaning, our analytic dataset consists of 67,063 observations on 10,195 individuals. The outcome

TABLE 1 Ninety-five percent confidence intervals for the ACCORD data using an LMM that includes a random intercept, a random slope, and a covariance term between the random effects. We can see that all three methods give similar results, but BLB is much faster than the bootstrap

Method	BLB	Bootstrap	Wald
Fixed effect			
Intercept	(215.45, 232.02)	(216.72, 232.54)	(216.56, 232.56)
Visit number	(−0.26, −0.22)	(−0.27, −0.22)	(−0.27, −0.22)
BMI	(−0.28, −0.05)	(−0.28, −0.06)	(−0.28, −0.06)
Female	(−2.39, 0.23)	(−2.17, 0.41)	(−2.25, 0.42)
Baseline age	(−0.86, −0.67)	(−0.87, −0.68)	(−0.87, −0.67)
Race			
Black	(−10.42, −7.08)	(−10.26, −6.98)	(−10.30, −6.95)
Hispanic	(−4.12, 0.97)	(−4.80, 0.18)	(−4.83, 0.20)
Other	(−4.00, 0.32)	(−4.01, 0.13)	(−4.05, 0.10)
CVD history	(−0.87, 1.81)	(−0.32, 2.37)	(−0.34, 2.35)
Adjusted insulin (units/kg body weight)	(−12.23, −8.64)	(−12.06, −9.36)	(−12.18, −9.35)
Sulphonylureas	(−0.51, 1.72)	(−0.57, 1.48)	(−0.58, 1.47)
Metformin	(−7.35, −4.69)	(−7.27, −4.88)	(−7.23, −4.82)
Meglitinides	(−14.93, −12.58)	(−14.80, −12.41)	(−14.82, −12.37)
Thiazolidinediones	(−21.28, −19.29)	(−21.35, −19.56)	(−21.38, −19.59)
Variance components			
Intercept	(772.93, 883.30)	(809.81, 890.68)	
Visit number	(0.20, 0.28)	(0.21, 0.26)	
Intercept: visit number	(−6.17, −3.82)	(−6.20, −4.49)	
Residual	(1814.22, 1906.54)	(1837.48, 1884.00)	
Runtime (second)	230	2650	

of interest is fasting plasma glucose, and the covariates include gender, race, baseline age, BMI, visit number, baseline CVD history, adjusted insulin, and the type of therapy they received. We follow Siraj et al. [15] and use insulin units per body weight in kg (adjusted insulin) instead of raw total insulin units. In addition to random intercept, we also included a random slope for the visit number. Since the ground truth is not available for real data, we cannot compare methods using relative error. Instead, we present the 95% confidence intervals given by BLB, bootstrap, and the Wald method. Note that the Wald method can only produce confidence intervals for fixed effect parameters. For this analysis, we used a subset size of 1600 individuals ($\gamma = 0.8$) and ran BLB on 30 subsets, each with 200 bootstrap samples. The subset size was chosen so that we would not get too few observations for certain categories in the unevenly distributed race variable. A sensitivity analysis of other subset sizes is given in Section S7 in supporting information. For bootstrap, we ran it with 2000 bootstrap samples. Both methods used parallel processing. Table 1 shows the results. We find the visit number, BMI, baseline age, race, adjusted insulin, and certain oral medication classes to be significantly associated with fasting plasma glucose. We also find the random slope for visit number to be significant and should be included in the model. Finally, we note that BLB achieves similar inference compared with bootstrap, but uses much less time.

7 | CONCLUSION AND FUTURE WORK

We have developed an algorithm based on the BLB method for the statistical inference of fixed effect and variance component parameters of LMMs on large and distributed longitudinal datasets; we also developed a Julia software package MixedModelsBLB.jl for this purpose. Unlike the bootstrap method, which typically requires $O\left(BNq^3\right)$

computational cost, our method only costs $O\left(Bbq^3\right)$, where b is much smaller than N. The simulation and real data results demonstrate the efficiency and statistical performance of our method.

ACKNOWLEDGMENTS

This research was partially funded by grants from the National Institute of General Medical Sciences (GM141798, HZ), the National Human Genome Research Institute (HG006139, HZ and JJZ), the National Science Foundation (DMS-2054253, HZ and JJZ), the National Institute of Diabetes and Digestive and Kidney Disease (K01DK106116, JJZ), and the National Heart, Lung, and Blood Institute (R21HL150374, JJZ).

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

The Action to Control Cardiovascular Disease (ACCORD) data are available from NIH BioLINCC (2021) (https://biolincc.nhlbi.nih.gov/studies/accord/) by request.

ORCID

Hua Zhou https://orcid.org/0000-0003-1320-7118

REFERENCES

1. D. Bates, M. Mächler, B. Bolker, and S. Walker, *Fitting linear mixed-effects models using lme4*, J Stat Softw 67 (2015), no. 1, 1–48.
2. G. Belenky, N. J. Wesensten, D. R. Thorne, M. L. Thomas, H. C. Sing, D. P. Redmond, M. B. Russo, and T. J. Balkin, *Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: A sleep dose-response study*, J. Sleep Res. 12 (2003), no. 1, 1–12.
3. P. J. Bickel, F. Götze, and W. R. van Zwet, *Resampling fewer than n observations: Gains, losses, and remedies for losses*, Stat. Sin. 7 (1997), no. 1, 1–31.
4. R. H. Byrd, J. Nocedal, and R. A. Waltz, "*Knitro: An integrated package for nonlinear optimization*," *Large-scale nonlinear optimization*, Springer, Boston, MA, 2006, pp. 35–59.
5. C. Crainiceanu, "*Likelihood ratio testing for zero variance components in linear mixed models*," *Random effect and latent variable model selection*, Lecture Notes in Statistics, Vol 192, D. Dunson (ed.), Springer, New York, NY, 2008.
6. C. M. Crainiceanu and D. Ruppert, *Likelihood ratio tests in linear mixed models with one variance component*, J. R. Stat. Soc. Ser. B Stat Methodol. 66 (2004), no. 1, 165–185. https://doi.org/10.1111/j.1467-9868.2004.00438.x
7. B. Efron, *Bootstrap methods: Another look at the jackknife*, Ann. Stat. 7 (1979), no. 1, 1–26.
8. C. A. German, J. S. Sinsheimer, J. Zhou, and H. Zhou, *WiSER: Robust and scalable estimation and inference of within-subject variances from intensive longitudinal data*, Biometrics (2021). https://doi.org/10.1111/biom.13506.
9. U. Halekoh and S. Højsgaard, *A Kenward-Roger approximation and parametric bootstrap methods for tests in linear mixed models–the R package pbkrtest*, J. Stat. Softw. 59 (2014), no. 9, 1–30.
10. F. Ismail-Beigi, T. Craven, M. A. Banerji, J. Basile, J. Calles, R. M. Cohen, R. Cuddihy, W. C. Cushman, S. Genuth, R. H. Grimm Jr., B. P. Hamilton, B. Hoogwerf, D. Karl, L. Katz, A. Krikorian, P. O'Connor, R. Pop-Busui, U. Schubart, D. Simmons, H. Taylor, A. Thomas, D. Weiss, I. Hramiak, and ACCORD trial group, *Effect of intensive treatment of hyperglycaemia on microvascular outcomes in type 2 diabetes: An analysis of the ACCORD randomised trial*, Lancet 376 (2010), no. 9739, 419–430.
11. S. G. Johnson, *The NLopt nonlinear-optimization package*. http://github.com/stevengj/nlopt, 2020.
12. A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan, *A scalable bootstrap for massive data*, J R Stat Soc: Ser B: Stat Methodol 76 (2014), 795–816.
13. D. N. Politis, J. P. Romano, and M. Wolf, *Subsampling*, Springer Science & Business Media, New York, NY, 1999.
14. S. G. Self and K.-Y. Liang, *Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions*, J. Am. Stat. Assoc. 82 (1987), no. 398, 605–610.
15. E. S. Siraj, D. J. Rubin, M. C. Riddle, M. E. Miller, F.-C. Hsu, F. Ismail-Beigi, S.-H. Chen, W. T. Ambrosius, A. Thomas, W. Bestermann, J. B. Buse, S. Genuth, C. Joyce, C. S. Kovacs, P. J. O'Connor, R. J. Sigal, S. Solomon, and ACCORD Investigators, *Insulin dose and cardiovascular mortality in the accord trial*, Diabetes Care 38 (2015), no. 11, 2000–2008.
16. A. Van Der Vaart and J. Wellner, *Weak convergence and empirical processes: With applications to statistics*, Springer, New York, NY, 1996.
17. A. W. Van der Vaart, *Asymptotic statistics*, Vol 3, Cambridge University Press, Cambridge, MA, 2000.
18. A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program. 106 (2006), no. 1, 25–57.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: X. Zhou, J. J. Zhou, and H. Zhou, *Bag of little bootstraps for massive and distributed longitudinal data*, Stat. Anal. Data Min.: ASA Data Sci. J. (2021), 1–8. https://doi.org/10.1002/sam.11563