Efficient moment calculations for variance components in large unbalanced crossed random effects models

Katelyn Gao
Stanford University

Art B. Owen
Stanford University

January 2016

Abstract

Large crossed data sets, described by generalized linear mixed models, have become increasingly common and provide challenges for statistical analysis. At very large sizes it becomes desirable to have the computational costs of estimation, inference and prediction (both space and time) grow at most linearly with sample size.

Both traditional maximum likelihood estimation and numerous Markov chain Monte Carlo Bayesian algorithms take superlinear time in order to obtain good parameter estimates. We propose moment based algorithms that, with at most linear cost, estimate variance components, measure the uncertainties of those estimates, and generate shrinkage based predictions for missing observations. When run on simulated normally distributed data, our algorithm performs competitively with maximum likelihood methods.

1 Introduction

Modern electronic activity generates enormous data sets with an unbalanced crossed random effects structure. The factors are customer IDs, URLs, product IDs, cookies, IP addresses, news stories, tweets, and query strings, among others. These variables could be treated as fixed effects, plain categorical variables that just happen to have a large number of levels. But in many cases, the specific category levels are evanescent. Customers turn over at some rate, cookies get deleted at an even faster rate, products or news stories grow in popularity but then fade. In such cases it is more realistic to treat such variables as random effects. We want our inferences to apply to the population from which the future and observed levels of those variables are sampled. Furthermore, for realism we should treat data in the same level of a factor as correlated.

The statistically efficient way to treat data sets with crossed random effects is through generalized linear mixed models (GLMMs), maximizing the likelihood with respect to both the parameters and the random effects. However, the cost of these computations is dominated by a Cholesky decomposition that takes time cubic in the number of distinct levels and space quadratic in that number; see Bates (2014) or Raudenbush (1993). Such costs are infeasible for big data.

It has been suggested to us that stochastic gradient descent (SGD) could provide an alternative way to maximize the likelihood. However, SGD approaches have only been

developed for data that can be split into independent subsets, which is not possible for data sets with crossed random effects.

With GLMMs infeasible, it is natural to consider the Gibbs sampler and other Markov Chain Monte Carlo (MCMC) methods. But, as shown in Section 2, those methods in the crossed random effects context has computational cost that is superlinear in the sample size. This is very different from the great success that MCMC has on hierarchical models for data with a nested structure. See for instance Gelman et al. (2012), Snijders (2014) and Yu and Meng (2011).

With both likelihood and Bayesian methods running into difficulties, we turn to the method of moments. It seems ironic to use a 19th century method in this era of increased computer power. But data growth has been outpacing processing power for single-threaded computation, so it is appropriate to revisit methods from an earlier time when the data was large compared to the available computing power. A compelling advantage of the method of moments is that it is easily parallelizable. It also makes very weak assumptions, has no tuning parameters, and does not require cumbersome diagnostics.

We are motivated by generalized linear mixed models with linear predictors but we focus the present paper on a very special case. We consider a setting with identity link, just two factors that are both random, and intercept only regression. In this paper, we assume that the data follows the model

Model 1. *Two-factor crossed random effects:*

$$\begin{aligned} &Y_{ij} = \mu + a_i + b_j + e_{ij}, \quad i, j \in \mathbb{N} \quad \textit{where} \\ &a_i \overset{\text{iid}}{\sim} (0, \sigma_A^2), \quad b_j \overset{\text{iid}}{\sim} (0, \sigma_B^2), \quad e_{ij} \overset{\text{iid}}{\sim} (0, \sigma_E^2) \quad \textit{and} \\ &\mathbb{E}(a_i^4) < \infty, \quad \mathbb{E}(b_j^4) < \infty, \quad \mathbb{E}(e_{ij}^4) < \infty \end{aligned} \tag{1}$$

In the available data we only see N of the Y_{ij}, where $1 \leqslant N < \infty$, in R distinct rows (i's) and C distinct columns (j's). We assume that observations are missing completely at random. See Section 7.1 for comments on informative missingness. Note that we do not make any distributional assumptions.

We choose this model because it is the simplest case that exhibits the intrinsic difficulty of the large unbalanced crossed random effects setting, even though it may not describe real-world data well. Our goal is not to resolve the issue of analyzing massive crossed data sets via GLMMs in one go. Instead, we consider a simple GLMM for crossed data and study parameter estimation in that model, which is still a challenging problem.

Let $\theta = (\sigma_A^2, \sigma_B^2, \sigma_E^2)^{\mathsf{T}}$ be the vector of variance components. Our first task is to get an unbiased estimate $\hat{\theta}$ of θ at computational cost $O(N)$ and using additional storage that is $O(R + C)$, which is often sublinear in N.

Our second and more challenging task is to find the variance of $\hat{\theta}$, $\mathrm{Var}(\hat{\theta} \mid \theta, \kappa)$. This variance depends on both θ and the vector of kurtoses of the random effects $\kappa = (\kappa_A, \kappa_B, \kappa_E)^{\mathsf{T}}$. We develop formulas $V(\theta, \kappa)$ approximating $\mathrm{Var}(\hat{\theta} \mid \theta, \kappa)$ that can be computed in $O(N)$ time and $O(R + C)$ storage, given values for θ and κ. After developing an estimate $\hat{\kappa}$ that can be computed in $O(N)$ time and $O(R+C)$ space, we let $\widehat{\mathrm{Var}}(\hat{\theta}) = V(\hat{\theta}, \hat{\kappa})$ be our plug-in estimate of the variance of $\hat{\theta}$.

Notice that in order to achieve the complexity bounds, we choose to over-estimate $\mathrm{Var}(\hat{\theta})$. Specifically, we require the functions V to satisfy $\mathrm{diag}(V(\theta, \kappa)) \geqslant \mathrm{diag}(\mathrm{Var}(\hat{\theta} \mid \theta, \kappa))$. There

is a trade-off in selecting V though; the less conservative it is, the more time needed to compute it.

For large data sets we might suppose that $\mathrm{Var}(\hat{\theta})$ is necessarily very small and getting exact values is not important. While this may be true, it is wise to check. The effective sample size (as defined in Lavrakas (2008)) in model (1) might be as small as R or C if the row or column effects dominate. Moreover, if the sampling frequencies of rows or columns are very unequal, then the effective sample size can be much smaller than R or C. For example, the Netflix data set (Bennett and Lanning, 2007) has $N \doteq 10^8$. But there are only about 18,000 movies and so for statistics dominated by the movie effect the effective sample size might be closer to 18,000. That the movies do not appear equally often would further reduce the effective sample size. Indeed, Owen (2007) shows that for some linear statistics the variance could be as much as 50,000 times larger than a formula based on IID sampling would yield. That factor is perhaps extreme but it would translate a nominal sample size of 10^8 into an effective sample size closer to 2,000.

An outline of this paper is as follows. Section 2 describes the difficulties with Gibbs sampling and other MCMC algorithms for crossed random effects, as suggested by theoretical results and shown through simulations. Section 3 introduces further notation and assumptions. Section 4 presents our linear-cost algorithm to estimate θ and conservatively approximate the variance of that estimate. Section 5 studies how knowledge of σ_A^2, σ_B^2, and σ_E^2 can be used to construct shrinkage predictions of unknown Y_{ij}. Section 6 illustrates the methods in Section 4 on both simulated Gaussian data and real world data. Section 7 concludes the paper and discusses informative missingness. The appendix, Section 8, has a proof of convergence rates for MCMC methods and tables of their simulation results. A supplement, Sections 10–20, develops the variance formulas for our moment estimates and provides proofs of our theorems about prediction. We conclude this section with a few more pointers to the literature.

Our procedure to find variance component estimates are similar to those of Henderson (1953) as described in Searle et al. (2009, Chapter 5). Some differences are that we use U-statistics, and that we find variance component estimates and variances of those estimates in time and space $O(N)$. For one of Henderson's algorithms, even the point estimates require superlinear computation in inverting $R \times R$ or $C \times C$ matrices. Moreover, the majority of Searle et al. (2009) considers Gaussian data which makes the kurtoses zero. Gaussian variables are not a reasonable assumption in our target applications and so we develop kurtosis estimates.

For crossed random effects models with missing data Clayton and Rasbash (1999) propose an alternating imputation-posterior (AIP) algorithm, which they show has good performance on fairly large data sets. It may be termed a 'pseudo-MCMC' method since it alternates between sampling the missing data from its distribution given the parameter estimates and sampling the parameters from a distribution centered on the maximum likelihood estimates. Because of this last step, we do not consider AIP to be scalable to Internet size problems.

In our model (1), for simplicity the variance components are homoscedastic. Alternatively, we could allow them to be heteroscedastic; see Owen (2007) or Owen and Eckles (2012), who study bootstrap variance estimates for means and smooth functions of means. The latter paper also considers a more complex model in the sense that there are more than

two factors as well as interactions among factors.

2 MCMC for large crossed data

In this section we consider some common MCMC methods to estimate the parameters σ_A^2, σ_B^2, and σ_E^2 of model (1). For this section only, we assume that a_i, b_j and e_{ij} are normally distributed.

Balanced data is a fully sampled $R\times C$ matrix with Y_{ij} for rows $i = 1, \dots, R$ and columns $j = 1, \dots, C$. We present some analyses for the balanced case with interspersed remarks on how the general unbalanced case behaves. The balanced case allows sharp formulas that we find useful and that case is the one we simulate. In particular, we can obtain convergence rates for some MCMC algorithms.

To estimate σ_A^2, σ_B^2, and σ_E^2 we sample from the posterior distribution given the data: $\pi = p(\mu, a, b, \sigma_A^2, \sigma_B^2, \sigma_E^2 \mid Y)$ where a is the vector of a_i and b is the vector of b_j. Let

$$S^{(t)} = \begin{pmatrix}\mu^{(t)} & a^{(t)\mathsf{T}} & b^{(t)\mathsf{T}} & \sigma_A^{2(t)} & \sigma_B^{2(t)} & \sigma_E^{2(t)}\end{pmatrix}^{\mathsf{T}}, \quad \text{for } t \geqslant 1$$

denote the resulting chain. While MCMC is effective for hierarchical random effects models, it scales badly for crossed random effects models as we see here. In limits where $R, C \to \infty$, the dimension of our chain $S^{(t)}$ approaches infinity. Convergence rates of many MCMC methods slow down as the dimension of the chain increases, making them ineffective for high dimensional parameter spaces.

The MCMC methods we consider go over the entire data set at each iteration. There are alternative samplers that save computation time by only looking at subsets of data at each iteration. However, so far those approaches are developed for IID data and not the crossed random effects setting.

2.1 Gibbs sampling

In each iteration of Gibbs sampling (Geman and Geman, 1984), we draw from the conditional posteriors of μ, a, b, σ_A^2, σ_B^2, and σ_E^2 in turn. For elucidation, let us consider the problem of Gibbs sampling from the 'smaller' distribution $\phi = p(a, b \mid \mu, \sigma_A^2, \sigma_B^2, \sigma_E^2, Y)$. At iteration $t+1$, we sample $a^{(t+1)} \sim p(a \mid b^{(t)}, \mu, \sigma_A^2, \sigma_B^2, \sigma_E^2, Y)$ and $b^{(t+1)} \sim p(b \mid a^{(t+1)}, \mu, \sigma_A^2, \sigma_B^2, \sigma_E^2, Y)$, which are normal distributions with diagonal covariance matrices. Let $X^{(t)}$ be the resulting chain.

Roberts and Sahu (1997) give the following definition.

Definition 2.1. Let $\theta^{(t)}$, for integer $t \geqslant 0$ be a Markov chain with stationary distribution h. Its convergence rate is the minimum number ρ such that

$$\lim_{t\to\infty} \mathbb{E}_h\big((\mathbb{E}_h(f(\theta^{(t)}) \mid \theta^{(0)}) - \mathbb{E}_h(f(\theta)))^2\big) r^{-t} = 0$$

holds for all measurable functions f such that $\mathbb{E}_h(f(\theta)^2) < \infty$ and all $r > \rho$.

Theorem 2.1. *Let ρ be the convergence rate of $X^{(t)}$ to ϕ, as in Definition 2.1. Then,*

$$\rho = \frac{\sigma_B^2}{\sigma_B^2 + \sigma_E^2/R} \times \frac{\sigma_A^2}{\sigma_A^2 + \sigma_E^2/C}.$$

Proof. See Section 8.1. □

We see that $\rho \to 1$ as $R, C \to \infty$, outside of trivial cases with σ_A^2 or σ_B^2 equal to zero. If R and C grow proportionately then $\rho = 1 - \alpha/\sqrt{N} + O(1/N)$ for some $\alpha > 0$. We can therefore expect the Gibbs sampler to require at least some constant multiple of $\sqrt{N}$ iterations to approximate the target distribution sufficiently. When the data are not perfectly balanced numerical computation of ρ shows that Gibbs still mixes increasingly slowly as $N \to \infty$. But in that case, the sampler requires $O(N)$ computation per iteration. In sum, Gibbs takes $O(N^{3/2})$ work to sample from ϕ, which is not scalable.

Because sampling from ϕ can be viewed as a subproblem of sampling from π, we believe that the Gibbs sampler that draws from π, which also requires $O(N)$ time per iteration, will exhibit the same slow convergence and hence require superlinear computation time.

2.2 Other MCMC algorithms

The Gibbs sampler is widely used for problems like this, where the full conditional distributions are tractable. But there are other MCMC algorithms that one could use. Here we consider random walk Metropolis (RWM), Langevin diffusion, and Metropolis adjusted Langevin (MALA). They also have difficulties scaling to large data sets.

At iteration $t+1$ of RWM, a Gaussian random walk proposal $S^{(t+1)} \sim \mathcal{N}(S^{(t)}, \sigma^2 I)$ for $\sigma^2 > 0$ is made and the step is taken with the Metropolis-Hastings acceptance probability. If the target distribution is a product distribution of dimension d, the chain $\tilde{S}^{(t)} \equiv S^{(dt)}$ (i.e. the chain formed by every dth state of the chain $S^{(t)}$) converges to a diffusion whose solution is the target distribution. We may interpret this as a convergence time for the algorithm that grows as $O(d)$ (Roberts and Rosenthal, 2001).

For our problem, evaluating the acceptance probability requires time at least $O(N)$, so the overall algorithm then takes $O(N(R+C))$ time. This is at best $O(N^{3/2})$, as we found for Gibbs sampling, and could be worse for sparse data where $N \ll RC$. Our target distribution is not of product form, and we have no reason to expect that RWM mixes orders of magnitude faster here than for a distribution of product form. Indeed, it seems more likely that mixing would be faster for product distributions than for distributions with more complicated dependence patterns such as ours.

At iteration $t+1$, Langevin diffusion steps $S^{(t+1)} \sim \mathcal{N}(S^{(t)} + (h/2)\nabla \log \pi(S^{(t)}), hI)$ for $h > 0$. As $h \to 0$, the stationary distribution for this process converges to π, as shown for general target distributions in (Liu, 2004). Because $h \neq 0$ in practice, the Langevin algorithm is biased. To correct this, the MALA algorithm uses the Metropolis-Hastings algorithm with the Langevin proposal $S^{(t+1)}$. When the target distribution is a product distribution of dimension d, the chain $\tilde{S}^{(t)} \equiv S^{(d^{1/3}t)}$ converges to a diffusion with solution π; the convergence time grows as $O(d^{1/3})$ (Roberts and Rosenthal, 2001). With similar reasoning as for RWM, the computation time is $O(N(R+C)^{1/3})$, which is at best $O(N^{1+1/6})$.

2.3 Simulation results

We carried out simulations of the four algorithms described above, as well as five others: the block Gibbs sampler ('Block'), the reparameterized Gibbs sampler ('Reparam.'), the

Method	Gibbs	Block	Reparam.	Lang.	MALA	Indp.	RWM	RWM Sub.	pCN
CPU sec.	3432	15046	4099	2302	4760	2513	2141	2635	1966
med μ	0.97	1.02	1.04	0.99	0.96	2.39	1.55	1.07	1.53
med σ_A^2	1.96	1.99	2.02	1.90	1.95	1.78	2.01	1.96	1.99
med σ_B^2	0.51	0.50	0.50	0.40	0.50	2.94	0.51	0.50	0.49
med σ_E^2	1.00	1.00	1.00	65.22	2.66	0.15	0	0.93	0
ACF(μ)	801	790	694	1	2501	5000+	1133	1656	1008
ACF(σ_A^2)	1	1	1	122	2656	5000+	1133	989	912
ACF (σ_B^2)	1	1	1	477	2514	5000+	1133	855	556
ACV(σ_E^2)	1	1	1	385	3062	5000+	1518	1724	621

Table 1: Summary of simulation results for cases with $R = C = 1000$. The first row gives CPU time in seconds. The next four rows give median estimates of the 4 parameters. The next four rows give the number of lags required to get an autocorrelation below 0.5.

independence sampler ('Indp.'), RWM with subsampling ('RWM Sub.'), and the pCN algorithm of Hairer et al. (2014). Descriptions of these five algorithms are given below with discussions of their simulation results. Every algorithm was implemented in MATLAB and run on a cluster using 4GB memory.

For each algorithm and a range of values of R and C, we generated balanced data from model (1) with $\mu = 1$, $\sigma_A^2 = 2$, $\sigma_B^2 = 0.5$, and $\sigma_E^2 = 1$. We ran 20,000 iterations of the algorithm, retaining the last 10,000 for analysis. We record the CPU time required, the median values of μ, σ_A^2, σ_B^2, and σ_E^2, and the number of lags needed for their sample auto-correlation functions (ACF) to go below 0.5.

The entire process is repeated in 10 independent runs. Table 1 presents median values of the recorded statistics over the 10 runs for the case $R = C = 1000$. Tables 2 through 6 of the appendix collect corresponding results at a range of (R, C) sizes.

Block Gibbs, which updates a and b together to try to improve mixing, has computation time superlinear in the number of observations. Also to improve mixing, reparameterized Gibbs scales the random effects to have equal variance. This gives an algorithm equivalent to the conditional augmentation of Van Dyk and Meng (2001). For all three Gibbs-type algorithms, the parameter estimates are good but μ mixes slower as R and C increase, while the variance components do not exhibit this behavior.

The computation times of Langevin diffusion ('Lang.') and MALA are approximately linear in the number of observations. However, σ_E^2 tends to explode for large data sets in Langevin diffusion, while the chain does not mix well in MALA.

The independent sampler is a Metropolis-Hastings algorithm where the proposal distribution is fixed. We propose $\mu \sim \mathcal{N}(1, 1)$, $a = \mathcal{N}(0, I_R)$, $b = \mathcal{N}(0, I_C)$, and $\sigma_A^2, \sigma_B^2, \sigma_E^2 \sim$ InvGamma$(1, 1)$. The computation time grows linearly with the data size. The parameters do not mix well, and their estimates are not good. It is possible that better results would be obtained from a different proposal distribution, but it is not clear how best to choose one in practice.

RWM and RWM with subsampling, the latter of which updates a subset of parameters at each iteration, both have computation time linear in the number of observations. Neither

algorithm mixed well, and for RWM σ_E^2 tended to go to zero in large data sets.

The pCN algorithm is Metropolis-Hastings where the proposals are Gaussian random walk steps shrunken towards zero: $S^{(t+1)} \sim \mathcal{N}(\sqrt{1-\sigma^2}S^{(t)}, \sigma^2 I)$, for $\sigma^2 \leq 1$. Hairer et al. (2014) show that under certain conditions on the target distribution, the convergence rate of this algorithm does not slow with the dimension of the distribution. We include it here, even though our π does not satisfy those conditions. The computation time grows linearly with the data size. However, the estimates for μ and σ_E^2 are not good, and those for σ_E^2 even get worse as the data size increases. None of the parameters seem to mix well.

In summary, for large data sets each algorithm mixes increasingly slowly or returns flawed estimates of μ and the variance components. We have also simulated some unbalanced data sets and slow mixing is once again the norm, with worse performance as R and C grow.

3 Further notation and assumptions

In this section, we go over pertinent notation and assumptions about the pattern of observations. Our data are realizations from model (1).

We refer to the first index of Y_{ij} as the 'row' and the second as the 'column'. We use integers i, i', r, r' to index rows and j, j', s, s' for columns. The actual indices may be URLs, customer IDs, or query strings and are not necessarily the integers we use here.

The variable Z_{ij} takes the value 1 if Y_{ij} is observed and 0 otherwise. We assume that there can be at most one observation in position (i, j).

The sample size is $N = \sum_{ij} Z_{ij} < \infty$. The number of observations in row i is $N_{i\bullet} = \sum_j Z_{ij}$ and the number in column j is $N_{\bullet j} = \sum_i Z_{ij}$. The number of distinct rows is $R = \sum_i 1_{N_{i\bullet}>0}$ and there are $C = \sum_j 1_{N_{\bullet j}>0}$ distinct columns. In the following, all of our sums over rows are only over rows i with $N_{i\bullet} > 0$, and similarly for sums over columns. We state this because there are a small number of expressions where omitting rows without data changes their values. This convention corresponds to what happens when one makes a pass through the whole data set.

Let Z be the matrix containing Z_{ij}. Of interest are $(ZZ^{\mathsf{T}})_{ii'} = \sum_j Z_{ij}Z_{i'j}$, the number of columns for which we have data in both rows i and i', and $(Z^{\mathsf{T}}Z)_{jj'}$. Note that $(ZZ^{\mathsf{T}})_{ii'} \leqslant N_{i\bullet}$ and furthermore

$$\sum_{ir}(ZZ^{\mathsf{T}})_{ir} = \sum_{jir} Z_{ij}Z_{rj} = \sum_j N_{\bullet j}^2, \quad \text{and} \quad \sum_{js}(Z^{\mathsf{T}}Z)_{js} = \sum_i N_{i\bullet}^2.$$

Two other useful idioms are

$$T_{i\bullet} = \sum_j Z_{ij}N_{\bullet j} \quad \text{and} \quad T_{\bullet j} = \sum_i Z_{ij}N_{i\bullet}. \tag{2}$$

$T_{i\bullet}$ is the total number of observations in all of the columns j that are represented in row i.

Our notation allows for an arbitrary pattern of observations. Some special cases are as follows. A balanced crossed design can be described via $Z_{ij} = 1_{i\leqslant R}1_{j\leqslant C}$. If $\max_i N_{i\bullet} = 1$ but $\max_j N_{\bullet j} > 1$ then the data have a nested structure with rows nested in columns. If $\max_i N_{i\bullet} = \max_j N_{\bullet j} = 1$, then the observed Y_{ij} are IID.

Some patterns are difficult to handle. For example, if all the observations are in the same row or column, some of the variance components are not identifiable. We are motivated by problems that are not such worst cases.

The quantities

$$\epsilon_R = \max_i N_{i\bullet}/N, \quad \text{and} \quad \epsilon_C = \max_j N_{\bullet j}/N \tag{3}$$

measure the extent to which a single row or column dominates the data set. We expect that these are both small and in limiting arguments, where $N \to \infty$, we may assume that

$$\max(\epsilon_R, \epsilon_C) \to 0. \tag{4}$$

It is also often reasonable to suppose that $\max_i T_{i\bullet}/N$ and $\max_j T_{\bullet j}/N$ are both small.

In many data sets, the average row and column sizes are large, but much smaller than N. One way to measure the average row size is N/R. Another way to measure it is to randomly choose an observation and inspect its row size, obtaining an expected value of $(1/N)\sum_i N_{i\bullet}^2$. Similar formulas hold for the average column size. Therefore, we assume that as $N \to \infty$

$$\max(R/N, C/N) \to 0 \tag{5}$$

and

$$\begin{aligned} &\min\Big(\frac{1}{N}\sum_i N_{i\bullet}^2, \frac{1}{N}\sum_j N_{\bullet j}^2\Big) \to \infty, \quad \text{and} \\ &\max\Big(\frac{1}{N^2}\sum_i N_{i\bullet}^2, \frac{1}{N^2}\sum_j N_{\bullet j}^2\Big) \to 0. \end{aligned} \tag{6}$$

Notice that

$$\frac{1}{N^2}\sum_i N_{i\bullet}^2 \leqslant \frac{1}{N^2}\sum_i N_{i\bullet}(\epsilon_R N) \leqslant \epsilon_R, \quad \text{and} \quad \frac{1}{N^2}\sum_j N_{\bullet j}^2 \leqslant \epsilon_C \tag{7}$$

and so the second part of (6) merely follows from (3) and (4).

While the average row count may be large, many of the rows corresponding to newly seen entities can have $N_{i\bullet} = 1$. In our analysis, it is not necessary to assume that all of the rows and columns contain at least some minimum number of observations. Thus, we avoid losing information by the practice of iteratively removing all rows and columns with few observations.

As a demonstration of the validity of our assumptions, the Netflix data has $N =$ 100,480,507 ratings on $R =$ 17,770 movies by $C =$ 480,189 customers. Therefore $R/N \doteq$ 0.00018 and $C/N \doteq 0.0047$. It is sparse with $N/(RC) \doteq 0.012$. It is not dominated by a single row or column because $\epsilon_R \doteq 0.0023$ and $\epsilon_R = 0.00018$ even though one customer has rated an astonishing 17,653 movies. Similarly

$$\frac{N}{\sum_i N_{i\bullet}^2} \doteq 1.78 \times 10^{-5}, \qquad \frac{\sum_j N_{\bullet j}^2}{N^2} \doteq 0.00056,$$

$$\frac{N}{\sum_j N_{\bullet j}^2} \doteq 0.0015, \quad \text{and} \qquad \frac{\sum_j N_{\bullet j}^2}{N^2} \doteq 6.43 \times 10^{-6}$$

so that the average row or column has size $\gg 1$ and $\ll N$.

There are various possible data storage models. We consider the log-file model with a collection of (i, j, Y_{ij}) triples, which for the purposes of this paper we assume are stored at the same location. A pass over the data proceeds via an iteration over all (i, j, Y_{ij}) triples in the data set. Such a pass may generate intermediate values that we assume can be retained for further computations.

4 Moment estimates of variance components

Here we develop a method of moments estimate $\hat{\theta}$ for $\theta = (\sigma_A^2, \sigma_B^2, \sigma_E^2)^\mathsf{T}$ that requires one pass over the data. We also find an expression for $\mathrm{Var}(\hat{\theta} \mid \theta, \kappa)$ and describe how to obtain an approximation of it after a second pass over the data.

Naturally, we would also want to estimate μ, and there are a number of ways to do so. The simplest is to let $\hat{\mu} = \bar{Y}_{\bullet\bullet}$, the sample mean. From Owen and Eckles (2012),

$$\mathrm{Var}(\bar{Y}_{\bullet\bullet}) = \sigma_A^2 \frac{\sum_r N_{r\bullet}^2}{N^2} + \sigma_B^2 \frac{\sum_s N_{\bullet s}^2}{N^2} + \frac{\sigma_E^2}{N} \leqslant \epsilon_R \sigma_A^2 + \epsilon_C \sigma_B^2 + \frac{\sigma_E^2}{N}. \tag{8}$$

The upper bound in (8) is tight for balanced data, but otherwise it can be very conservative. We anticipate that $1 \gg \epsilon_R, \epsilon_C \gg 1/N$ holds for our motivating applications as it did in the examples of Owen and Eckles (2012). The properties of this estimator has been well-studied in the literature, so in this paper we focus on estimating the variance components.

4.1 U-statistics for variance components

We use U-statistics in our method of moments estimators. The usual unbiased sample variance estimate can be formulated as a U-statistic, which is more convenient to analyze. We use the following U-statistics:

$$\begin{aligned} U_a &= \frac{1}{2} \sum_{ijj'} N_{i\bullet}^{-1} Z_{ij} Z_{ij'} (Y_{ij} - Y_{ij'})^2, \\ U_b &= \frac{1}{2} \sum_{jii'} N_{\bullet j}^{-1} Z_{ij} Z_{i'j} (Y_{ij} - Y_{i'j})^2, \quad \text{and} \\ U_e &= \frac{1}{2} \sum_{iji'j'} Z_{ij} Z_{i'j'} (Y_{ij} - Y_{i'j'})^2. \end{aligned} \tag{9}$$

To understand U_a note that for each row i, the quantities $Y_{ij} - \mu - a_i$ are IID with variance $\sigma_B^2 + \sigma_E^2$. Thus, U_a is a weighted sum of within-row unbiased estimates of $\sigma_B^2 + \sigma_E^2$. The explanation for U_b is similar, while U_e is a proportional to the sample variance estimate of all N observations.

Lemma 4.1. *Let Y_{ij} follow the two-factor crossed random effects model* (1) *with the observation pattern Z_{ij} as described in Section 3. Then the U-statistics defined at* (9) *satisfy*

$$\begin{aligned}
\mathbb{E}(U_a) &= (\sigma_B^2 + \sigma_E^2)(N - R)\\
\mathbb{E}(U_b) &= (\sigma_A^2 + \sigma_E^2)(N - C), \quad \textit{and}\\
\mathbb{E}(U_e) &= \sigma_A^2(N^2 - \sum_i N_{i\bullet}^2) + \sigma_B^2(N^2 - \sum_j N_{\bullet j}^2) + \sigma_E^2(N^2 - N).
\end{aligned}$$

Proof. See Section 11.1 of the supplement. □

To obtain unbiased estimates $\hat\sigma_A^2$, $\hat\sigma_B^2$, and $\hat\sigma_E^2$ given values of the U-statistics, we solve the 3×3 system of equations

$$M\begin{pmatrix}\hat\sigma_A^2\\ \hat\sigma_B^2\\ \hat\sigma_E^2\end{pmatrix} = \begin{pmatrix}U_a\\ U_b\\ U_e\end{pmatrix}, \quad \text{for} \quad M = \begin{pmatrix} 0 & N-R & N-R\\ N-C & 0 & N-C\\ N^2 - \sum_i N_{i\bullet}^2 & N^2 - \sum_j N_{\bullet j}^2 & N^2 - N\end{pmatrix} \tag{10}$$

For our method to return unique and meaningful estimates, the determinant of M

$$\begin{aligned}
\det M &= (N-R)(N-C)\Big(N^2 - \sum_i N_{i\bullet}^2 - \sum_j N_{\bullet j}^2 + N\Big)\\
&\geqslant (N-R)(N-C)(N^2(1-\epsilon_R-\epsilon_C) + N)
\end{aligned}$$

must be nonzero. This is true when no row or column has more than half of the data, and at least one row and at least one column has more than one observation.

To compute the U-statistics, notice that $U_a = \sum_i S_{i\bullet}$, where $S_{i\bullet} = \sum_j Z_{ij}(Y_{ij} - \bar Y_{i\bullet})^2$ and $\bar Y_{i\bullet} = (1/N_{i\bullet})\sum_j Z_{ij}Y_{ij}$. In one pass over the data and time $O(N)$, we compute $N_{i\bullet}$, $\bar Y_{i\bullet}$, and $S_{i\bullet}$ for all R observed levels of i using the incremental algorithm described in the next paragraph. We can also compute N, R and C in such a pass if they are not known beforehand.

Chan et al. (1983) show how to compute both $Y_{i\bullet} = N_{i\bullet}\bar Y_{i\bullet}$ and $S_{i\bullet}$ in a numerically stable one pass algorithm. At the initial appearance of an observation in row i, with corresponding column $j = j(1)$, set $N_{i\bullet} = 1$, $Y_{i\bullet} = Y_{ij}$ and $S_{i\bullet} = 0$. After that, at the kth appearance of an observation in row i, with corresponding column $j(k)$,

$$N_{i\bullet} \leftarrow N_{i\bullet} + 1, \qquad Y_{i\bullet} \leftarrow Y_{i\bullet} + Y_{ij(k)}, \quad \text{and} \qquad S_{i\bullet} \leftarrow S_{i\bullet} + \frac{(k \times Y_{ij(k)} - Y_{i\bullet})^2}{k(k-1)}. \tag{11}$$

Chan et al. (1983) give a detailed analysis of roundoff error for update (11) as well as generalizations that update higher moments from groups of data values.

In that same pass over the data, U_e and the analogous quantities needed to compute U_b ($S_{\bullet j}$, $\bar Y_{\bullet j}$, $N_{\bullet j}$) are also computed using the incremental algorithm. Finally, in additional time $O(R + C)$, we calculate $\sum_i S_{i\bullet}$, $\sum_j S_{\bullet j}$, $\sum_i N_{i\bullet}^2$, and $\sum_j N_{\bullet j}^2$. Now, we have U_a, U_b, U_e, and all the entries of M.

Given U_a, U_b, U_e, and M we can calculate $\hat\sigma_A^2$, $\hat\sigma_B^2$, and $\hat\sigma_E^2$ in constant time. Therefore, finding our method of moments estimators takes $O(N)$ time overall.

4.2 Variances of the estimators

In this section we present how to estimate the covariance matrix of $\hat{\theta} = (\hat{\sigma}_A^2, \hat{\sigma}_B^2, \hat{\sigma}_E^2)^{\mathsf{T}}$.

4.2.1 True variance of $\hat{\theta}$

This section discusses the finite sample covariance matrix of $\hat{\theta}$. Theorem 4.1 below gives the exact variances and covariances of our U-statistics.

Theorem 4.1. *Let Y_{ij} follow the random effects model* (1) *with the observation pattern Z_{ij} as described in Section 3. Then the U-statistics defined at* (9) *have variances*

$$\begin{aligned}\mathrm{Var}(U_a) = {} & \sigma_B^4(\kappa_B+2)\sum_{ir}(ZZ^{\mathsf{T}})_{ir}(1-N_{i\bullet}^{-1})(1-N_{r\bullet}^{-1}) \\ & + 2\sigma_B^4\sum_{ir}N_{i\bullet}^{-1}N_{r\bullet}^{-1}(ZZ^{\mathsf{T}})_{ir}((ZZ^{\mathsf{T}})_{ir}-1)) + 4\sigma_B^2\sigma_E^2(N-R) \\ & + \sigma_E^4(\kappa_E+2)\sum_i N_{i\bullet}(1-N_{i\bullet}^{-1})^2 + 2\sigma_E^4\sum_i(1-N_{i\bullet}^{-1}),\end{aligned} \tag{12}$$

and

$$\begin{aligned}\mathrm{Var}(U_b) = {} & \sigma_A^4(\kappa_A+2)\sum_{js}(Z^{\mathsf{T}}Z)_{js}(1-N_{\bullet j}^{-1})(1-N_{\bullet s}^{-1}) \\ & + 2\sigma_A^4\sum_{js}N_{\bullet j}^{-1}N_{\bullet s}^{-1}(Z^{\mathsf{T}}Z)_{js}((Z^{\mathsf{T}}Z)_{js}-1)) + 4\sigma_A^2\sigma_E^2(N-C) \\ & + \sigma_E^4(\kappa_E+2)\sum_j N_{\bullet j}(1-N_{\bullet j}^{-1})^2 + 2\sigma_E^4\sum_j(1-N_{\bullet j}^{-1}),\end{aligned} \tag{13}$$

and $\mathrm{Var}(U_e)$ *equals*

$$\begin{aligned}& 2\sigma_A^4((\sum_i N_{i\bullet}^2)^2 - \sum_i N_{i\bullet}^4) + \sigma_A^4(\kappa_A+2)(N^2\sum_i N_{i\bullet}^2 - 2N\sum_i N_{i\bullet}^3 + \sum_i N_{i\bullet}^4) \\ & + 2\sigma_B^4((\sum_j N_{\bullet j}^2)^2 - \sum_j N_{\bullet j}^4) + \sigma_B^4(\kappa_B+2)(N^2\sum_j N_{\bullet j}^2 - 2N\sum_j N_{\bullet j}^3 + \sum_j N_{\bullet j}^4) \\ & + 2\sigma_E^4N(N-1) + \sigma_E^4(\kappa_E+2)N(N-1)^2 \\ & + 4\sigma_A^2\sigma_B^2(N^3 - 2N\sum_{ij}Z_{ij}N_{i\bullet}N_{\bullet j} + \sum_{ij}N_{i\bullet}^2N_{\bullet j}^2) \\ & + 4\sigma_A^2\sigma_E^2(N^3 - N\sum_i N_{i\bullet}^2) + 4\sigma_B^2\sigma_E^2(N^3 - N\sum_j N_{\bullet j}^2).\end{aligned} \tag{14}$$

Their covariances are

$$\mathrm{Cov}(U_a, U_b) = \sigma_E^4(\kappa_E + 2)\sum_{ij} Z_{ij}(1 - N_{i\bullet}^{-1})(1 - N_{\bullet j}^{-1}), \tag{15}$$

$$\begin{aligned}\mathrm{Cov}(U_a, U_e) = {} & 2\sigma_B^4\Big(\sum_i N_{i\bullet}^{-1}T_{i\bullet}^2 - \sum_{ij} Z_{ij}N_{i\bullet}^{-1}N_{\bullet j}^2\Big) \\ & + \sigma_B^4(\kappa_B + 2)\sum_{ij} Z_{ij}(N - N_{\bullet j})N_{\bullet j}(1 - N_{i\bullet}^{-1}) \\ & + 2\sigma_E^4(N - R) + \sigma_E^4(\kappa_E + 2)(N - R)(N - 1) \\ & + 4\sigma_B^2\sigma_E^2 N(N - R), \quad \textit{and}\end{aligned} \tag{16}$$

$$\begin{aligned}\mathrm{Cov}(U_b, U_e) = {} & 2\sigma_A^4\Big(\sum_j N_{\bullet j}^{-1}T_{\bullet j}^2 - \sum_{ij} Z_{ij}N_{\bullet j}^{-1}N_{i\bullet}^2\Big) \\ & + \sigma_A^4(\kappa_A + 2)\sum_{ij} Z_{ij}(N - N_{i\bullet})N_{i\bullet}(1 - N_{\bullet j}^{-1}) \\ & + 2\sigma_E^4(N - C) + \sigma_E^4(\kappa_E + 2)(N - C)(N - 1) \\ & + 4\sigma_A^2\sigma_E^2 N(N - C).\end{aligned} \tag{17}$$

Proof. Equation (12) is proved in Section 12.2 of the supplement and then equation (13) follows by exchanging indices. Equation (14) is proved in Section 12.7 of the supplement. Equation (15) is proved in Section 13 of the supplement. Equation (16) is proved in Section 14 of the supplement and then equation (17) follows by exchanging indices. □

Now we consider $\mathrm{Var}(\hat{\theta})$. From (10)

$$\mathrm{Var}(\hat{\theta}) = M^{-1}\mathrm{Var}\begin{pmatrix} U_a \\ U_b \\ U_e \end{pmatrix}(M^{-1})^{\mathsf{T}}. \tag{18}$$

We show in Section 4.2.2 that while $\mathrm{Var}(U_e)$ and the covariances of the U-statistics may be exactly computed in time $O(N)$, $\mathrm{Var}(U_a)$ and $\mathrm{Var}(U_b)$ cannot. Therefore, we approximate $\mathrm{Var}(U_a)$ and $\mathrm{Var}(U_b)$ such that when we apply formula (18) we get conservative estimates of $\mathrm{Var}(\hat{\sigma}_A^2)$, $\mathrm{Var}(\hat{\sigma}_B^2)$, and $\mathrm{Var}(\hat{\sigma}_E^2)$ (the values of primary interest).

For intuition on what sort of approximation is needed, we give a linear expansion of $\mathrm{Var}(\hat{\theta})$ in terms of the variances and covariances of the U-statistics. Letting $\epsilon = \max(\epsilon_R, \epsilon_C, R/N, C/N)$ we have that as $\epsilon \to 0$

$$M = \begin{pmatrix} N & & \\ & N & \\ & & N^2 \end{pmatrix}\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}(1 + O(\epsilon))$$

and so

$$M^{-1} = \begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}\begin{pmatrix} N^{-1} & & \\ & N^{-1} & \\ & & N^{-2} \end{pmatrix}(1 + O(\epsilon)).$$

It follows that

$$\begin{aligned}
\hat{\sigma}_A^2 &= (U_e/N^2 - U_a/N)(1 + O(\epsilon)), \\
\hat{\sigma}_B^2 &= (U_e/N^2 - U_b/N)(1 + O(\epsilon)), \quad \text{and} \\
\hat{\sigma}_E^2 &= (U_a/N + U_b/N - U_e/N^2)(1 + O(\epsilon)).
\end{aligned} \tag{19}$$

Disregarding the $O(\epsilon)$ terms,

$$\begin{aligned}
\mathrm{Var}(\hat{\sigma}_A^2) &\doteq \mathrm{Var}(U_e)/N^4 + \mathrm{Var}(U_a)/N^2 - 2\mathrm{Cov}(U_a, U_e)/N^3, \\
\mathrm{Var}(\hat{\sigma}_B^2) &\doteq \mathrm{Var}(U_e)/N^4 + \mathrm{Var}(U_b)/N^2 - 2\mathrm{Cov}(U_b, U_e)/N^3, \quad \text{and} \\
\mathrm{Var}(\hat{\sigma}_E^2) &\doteq \mathrm{Var}(U_a)/N^2 + \mathrm{Var}(U_b)/N^2 + \mathrm{Var}(U_e)/N^4 \\
&\qquad - 2\mathrm{Cov}(U_a, U_e)/N^3 - 2\mathrm{Cov}(U_b, U_e)/N^3 + 2\mathrm{Cov}(U_a, U_b)/N^2.
\end{aligned} \tag{20}$$

In light of equation (20), to find computationally attractive but conservative approximations of $\mathrm{Var}(\hat{\theta})$ in finite samples, we use over-estimates of $\mathrm{Var}(U_a)$ and $\mathrm{Var}(U_b)$. We discuss how to do so in Section 4.2.2.

In practice, when obtaining $\widehat{\mathrm{Var}}(\hat{\theta})$, unless we are in the asymptotic situation described in Section 4.2.3, we plug in $\hat{\sigma}_A^2$, $\hat{\sigma}_B^2$, $\hat{\sigma}_E^2$, and estimates of the kurtoses into the covariance matrix of the U-statistics where $\mathrm{Var}(U_a)$ and $\mathrm{Var}(U_b)$ have been replaced by their over-estimates. Then we apply equation (18). We discuss estimating the kurtoses in Section 4.2.4.

4.2.2 Computable approximations of $\mathrm{Var}(U)$

First, we show how to obtain over-estimates of $\mathrm{Var}(U_a)$ in time $O(N)$; the case of $\mathrm{Var}(U_b)$ is similar. In addition to $N - R$, $\mathrm{Var}(U_a)$ contains the following quantities

$$\sum_{ir} (ZZ^\mathsf{T})_{ir}(1 - N_{i\bullet}^{-1})(1 - N_{r\bullet}^{-1}) \qquad \sum_{ir} N_{i\bullet}^{-1} N_{r\bullet}^{-1} (ZZ^\mathsf{T})_{ir}((ZZ^\mathsf{T})_{ir} - 1)$$

$$\sum_{i} N_{i\bullet}(1 - N_{i\bullet}^{-1})^2, \quad \text{and} \qquad \sum_{i} (1 - N_{i\bullet}^{-1}).$$

The third and fourth quantities above can be computed in $O(R)$ work after the first pass over the data.

The first quantity is a sum over i and r, and cannot be simplified any further. Computing it takes more than $O(N)$ work. Since its coefficient $\sigma_B^4(\kappa_B + 2)$ is nonnegative, we must use an upper bound to obtain an over-estimate of $\mathrm{Var}(U_a)$. We have the bound

$$\begin{aligned}
\sum_{ir} (ZZ^\mathsf{T})_{ir}(1 - N_{i\bullet}^{-1})(1 - N_{r\bullet}^{-1}) &\leqslant \sum_{ij} \sum_{r} Z_{ij} Z_{rj} (1 - N_{i\bullet}^{-1}) \\
&= \sum_{j} N_{\bullet j}^2 - \sum_{ij} Z_{ij} N_{\bullet j} N_{i\bullet}^{-1},
\end{aligned}$$

which can be computed in $O(N)$ work in a second pass over the data. Other weaker bounds may be obtained without the second pass. An example is

$$\sum_{ir} (ZZ^\mathsf{T})_{ir}(1 - N_{i\bullet}^{-1})(1 - N_{r\bullet}^{-1}) \leqslant \sum_{ij} \sum_{r} Z_{ij} Z_{rj} = \sum_{j} N_{\bullet j}^2$$

which can be computed in $O(C)$ work.

For the same reason the second quantity cannot be computed in time $O(N)$ and we upper bound it via $(ZZ^{\mathsf{T}})_{ir} \leqslant N_{r\bullet}$, getting

$$\begin{aligned}\sum_{ir} N_{i\bullet}^{-1} N_{r\bullet}^{-1} (ZZ^{\mathsf{T}})_{ir}((ZZ^{\mathsf{T}})_{ir} - 1) &\leqslant \sum_{ir} N_{i\bullet}^{-1} N_{r\bullet}^{-1} (ZZ^{\mathsf{T}})_{ir}(N_{r\bullet} - 1) \\ &= \sum_{ij} Z_{ij} N_{i\bullet}^{-1} N_{\bullet j} - \sum_{ir} N_{i\bullet}^{-1} N_{r\bullet}^{-1} (ZZ^{\mathsf{T}})_{ir} \\ &\leqslant \sum_{ij} Z_{ij} N_{i\bullet}^{-1} N_{\bullet j}\end{aligned}$$

which can be computed in $O(N)$ work on a second pass.

All but one expression in $\mathrm{Var}(U_e)$ (see (14)) can be computed in $O(R + C)$ time after the first pass over the data. The one expression is

$$N^3 - 2\sum_{ij} Z_{ij} N_{i\bullet} N_{\bullet j} + \Big(\sum_i N_{i\bullet}^2\Big)\Big(\sum_j N_{\bullet j}^2\Big). \tag{21}$$

The second term in (21) requires a second pass over the data in time $O(N)$, because it is the sum over i and j of a polynomial of Z_{ij}, $N_{i\bullet}$, and $N_{\bullet j}$. The quantity in (21) alternatively can be expressed as

$$\sum_i \sum_j (N_{i\bullet} N_{\bullet j} - N Z_{ij})^2, \tag{22}$$

which shows that it is a kind of unnormalized test for row versus column independence in the observation process. Equation (22) is numerically more stable than (21) but requires $O(RC)$ computation which is ordinarily too expensive.

With the same reasoning as for the second term of (21), we see that $\mathrm{Cov}(U_a, U_b)$ can be computed in a second pass over the data in time $O(N)$. This reasoning also shows that we can compute nearly every term in $\mathrm{Cov}(U_a, U_e)$ in a second pass over the data; the exception is

$$\sum_i N_{i\bullet}^{-1} T_{i\bullet}^2 \tag{23}$$

We compute $T_{i\bullet}$ for each i in a second pass over the data. But, we must use additional time $O(R)$ to get (23). Nevertheless, the total computation time is still $O(N)$. Symmetrically $\mathrm{Cov}(U_b, U_e)$ can be computed in time $O(N)$ as well.

4.2.3 Asymptotic approximation of $\mathrm{Var}(\hat{\theta})$

Under asymptotic conditions, we may obtain simple, analytic approximate expressions for the covariance matrix of our method of moments estimators.

Theorem 4.2. *As described in Section 3, suppose that*

$$N_{i\bullet} \leqslant \delta N, \quad N_{\bullet j} \leqslant \delta N, \quad R \leqslant \delta N, \quad C \leqslant \delta N, \quad N \leqslant \delta \sum_i N_{i\bullet}^2, \quad \textit{and} \quad N \leqslant \delta \sum_j N_{\bullet j}^2,$$

hold for the same small $\delta > 0$ and that

$$0 < \kappa_A + 2, \kappa_B + 2, \kappa_E + 2, \sigma_A^4, \sigma_B^4, \sigma_E^4 < \infty.$$

Suppose additionally that

$$\sum_{ij} Z_{ij} N_{i\bullet}^{-1} N_{\bullet j} \leqslant \delta \sum_i N_{i\bullet}^2, \quad \textit{and} \qquad \sum_{ij} Z_{ij} N_{i\bullet} N_{\bullet j}^{-1} \leqslant \delta \sum_j N_{\bullet j}^2 \tag{24}$$

hold. Then

$$\begin{aligned}
\mathrm{Var}(U_a) &= \sigma_B^4(\kappa_B + 2) \sum_j N_{\bullet j}^2 (1 + O(\delta)) \\
\mathrm{Var}(U_b) &= \sigma_A^4(\kappa_A + 2) \sum_i N_{i\bullet}^2 (1 + O(\delta)), \quad \textit{and} \\
\mathrm{Var}(U_e) &= \Big(\sigma_A^4(\kappa_A + 2) N^2 \sum_i N_{i\bullet}^2 + \sigma_B^4(\kappa_B + 2) N^2 \sum_j N_{\bullet j}^2\Big)(1 + O(\delta)).
\end{aligned}$$

Similarly

$$\begin{aligned}
\mathrm{Cov}(U_a, U_b) &= \sigma_E^4(\kappa_E + 2) N (1 + O(\delta)), \\
\mathrm{Cov}(U_a, U_e) &= \sigma_B^4(\kappa_B + 2) N \sum_j N_{\bullet j}^2 (1 + O(\delta)), \quad \textit{and} \\
\mathrm{Cov}(U_b, U_e) &= \sigma_A^4(\kappa_A + 2) N \sum_i N_{i\bullet}^2 (1 + O(\delta)).
\end{aligned}$$

Finally $\hat\sigma_A^2$, $\hat\sigma_B^2$ and $\hat\sigma_E^2$ are asymptotically uncorrelated as $\delta \to 0$ with

$$\begin{aligned}
\mathrm{Var}(\hat\sigma_A^2) &= \sigma_A^4(\kappa_A + 2) \frac{1}{N^2} \sum_j N_{i\bullet}^2 (1 + O(\delta)) \\
\mathrm{Var}(\hat\sigma_B^2) &= \sigma_B^4(\kappa_B + 2) \frac{1}{N^2} \sum_j N_{\bullet j}^2 (1 + O(\delta)), \quad \textit{and} \\
\mathrm{Var}(\hat\sigma_E^2) &= \sigma_E^4(\kappa_E + 2) \frac{1}{N} (1 + O(\delta)).
\end{aligned}$$

Proof. See Section 16 of the supplement. □

We think that the typical $N_{\bullet j}$ is large, so $\sum_i N_{i\bullet}^2 = \sum_{ij} Z_{ij} N_{i\bullet}$ ought to be much larger than $\sum_{ij} Z_{ij} N_{i\bullet} N_{\bullet j}^{-1}$. A similar argument applies for $N_{i\bullet}$. Thus, the additional bounds in (24) seem very reasonable. However, it is possible that the pairs where $Z_{ij} = 1$ with large $N_{i\bullet}$ may have small $N_{\bullet j}$ and vice versa. Dyer and Owen (2011) report such a head-to-tail affinity in several data sets but it would have to be quite extreme for (24) to require a large δ.

The variance of $\hat\sigma_E^2$ is the same variance we would have gotten had $\sigma_A^2 = \sigma_B^2 = 0$ held. Similar remarks apply for $\hat\sigma_A^2$ and $\hat\sigma_B^2$.

4.2.4 Estimating kurtoses

Under a Gaussian assumption, $\kappa_A = \kappa_B = \kappa_E = 0$. If however the data have heavier tails than this, a Gaussian assumption will lead to underestimates of $\mathrm{Var}(\hat{\theta})$. Therefore, we will estimate the kurtoses by U-statistics.

Let $\mu_{A,4} = \mathbb{E}(a_i^4) = (\kappa_A + 3)\sigma_A^4$, $\mu_{B,4} = \mathbb{E}(b_i^4) = (\kappa_B + 3)\sigma_B^4$, and $\mu_{E,4} = \mathbb{E}(e_{ij}^4) = (\kappa_E + 3)\sigma_E^4$. The fourth moment U-statistics we use are

$$\begin{aligned} W_a &= \frac{1}{2}\sum_{ijj'} N_{i\bullet}^{-1} Z_{ij} Z_{ij'} (Y_{ij} - Y_{ij'})^4 \\ W_b &= \frac{1}{2}\sum_{iji'} N_{\bullet j}^{-1} Z_{ij} Z_{i'j} (Y_{ij} - Y_{i'j})^4, \quad \text{and} \\ W_e &= \frac{1}{2}\sum_{iji'j'} Z_{ij} Z_{i'j'} (Y_{ij} - Y_{i'j'})^4. \end{aligned} \tag{25}$$

Theorem 4.3. *Let Y_{ij} follow the random effects model* (1) *with the observation pattern Z_{ij} as described in Section 3. Then the statistics defined at* (25) *have means*

$$\begin{aligned} \mathbb{E}(W_a) &= (\mu_{B,4} + 3\sigma_B^4 + 12\sigma_B^2\sigma_E^2 + \mu_{E,4} + 3\sigma_E^4)(N - R) \\ \mathbb{E}(W_b) &= (\mu_{A,4} + 3\sigma_A^4 + 12\sigma_A^2\sigma_E^2 + \mu_{E,4} + 3\sigma_E^4)(N - C), \quad \textit{and} \\ \mathbb{E}(W_e) &= (\mu_{A,4} + 3\sigma_A^4 + 12\sigma_A^2\sigma_E^2)(N^2 - \sum_i N_{i\bullet}^2) \\ &\quad + (\mu_{B,4} + 3\sigma_B^4 + 12\sigma_B^2\sigma_E^2)(N^2 - \sum_j N_{\bullet j}^2) \\ &\quad + (\mu_{E,4} + 3\sigma_E^4)(N^2 - N) + 12\sigma_A^2\sigma_B^2(N^2 - \sum_i N_{i\bullet}^2 - \sum_j N_{\bullet j}^2 + N). \end{aligned}$$

Proof. See Section 17 of the supplement. □

Using Theorem 4.3, we compute estimates $\hat{\mu}_{A,4}$, $\hat{\mu}_{B,4}$, and $\hat{\mu}_{E,4}$, by solving the 3×3 system of equations

$$M \begin{pmatrix} \hat{\mu}_{A,4} \\ \hat{\mu}_{B,4} \\ \hat{\mu}_{E,4} \end{pmatrix} = \begin{pmatrix} W_a - m_a \\ W_b - m_b \\ W_e - m_e \end{pmatrix}, \tag{26}$$

where M is the same matrix that we used for the U-statistics in equation (10), with

$$\begin{aligned} m_a &= (3\hat{\sigma}_B^4 + 12\hat{\sigma}_B^2\hat{\sigma}_E^2 + 3\hat{\sigma}_E^4)(N - R), \\ m_b &= (3\hat{\sigma}_A^4 + 12\hat{\sigma}_A^2\hat{\sigma}_E^2 + 3\hat{\sigma}_E^4)(N - C), \quad \text{and} \\ m_e &= (3\hat{\sigma}_A^4 + 12\hat{\sigma}_A^2\hat{\sigma}_E^2)(N^2 - \sum_i N_{i\bullet}^2) + (3\hat{\sigma}_B^4 + 12\hat{\sigma}_B^2\hat{\sigma}_E^2)(N^2 - \sum_j N_{\bullet j}^2) \\ &\quad + 3\hat{\sigma}_E^4(N^2 - N) + 12\hat{\sigma}_A^2\hat{\sigma}_B^2(N^2 - \sum_i N_{i\bullet}^2 - \sum_j N_{\bullet j}^2 + N). \end{aligned}$$

We compute the statistics (25) via

$$\begin{aligned} W_a &= \sum_i \Big(\sum_j Z_{ij}(Y_{ij} - \bar{Y}_{i\bullet})^4 + 3N_{i\bullet}^{-1} S_{i\bullet}^2 \Big) \\ W_b &= \sum_j \Big(\sum_i Z_{ij}(Y_{ij} - \bar{Y}_{\bullet j})^4 + 3N_{\bullet j}^{-1} S_{\bullet j}^2 \Big), \quad \text{and} \\ W_e &= N \sum_{ij} Z_{ij}(Y_{ij} - \bar{Y}_{\bullet\bullet})^4 + 3S_{\bullet\bullet}^2, \end{aligned} \tag{27}$$

where $\bar{Y}_{\bullet\bullet} = N^{-1} \sum_{ij} Z_{ij} Y_{ij}$ and $S_{\bullet\bullet} = \sum_{ij} Z_{ij}(Y_{ij} - \bar{Y}_{\bullet\bullet})^2$.

Therefore, the kurtosis estimates $\hat{\kappa}$ requires $R + C + 1$ new quantities

$$\sum_j Z_{ij}(Y_{ij} - \bar{Y}_{i\bullet})^4, \quad \sum_i Z_{ij}(Y_{ij} - \bar{Y}_{\bullet j})^4, \quad \text{and} \quad \sum_{ij} Z_{ij}(Y_{ij} - \bar{Y}_{\bullet\bullet})^4 \tag{28}$$

beyond those used to compute $\hat{\theta}$. These can be computed in a second pass over the data after $\bar{Y}_{i\bullet}$, $\bar{Y}_{\bullet j}$ and $\bar{Y}_{\bullet\bullet}$ have been computed in the first pass. They can also be computed in the first pass using update formulas analogous to the second moment formulas (11). Such formulas are given by Pébay (2008), citing an unpublished paper by Terriberry.

Because the kurtosis estimates are used in formulas for $\widehat{\mathrm{Var}}(\hat{\theta})$ and those formulas already require a second pass over the data, it is more convenient to compute the sample fourth moments via (28) in a second pass. By a similar argument as in Section 4.1, obtaining $\hat{\kappa}_A$, $\hat{\kappa}_B$, and $\hat{\kappa}_E$ has space complexity $O(R + C)$ and time complexity $O(N)$, and is therefore scalable.

4.3 Algorithm summary

For clarity of exposition, here we gather all of the steps in our algorithm to estimate σ_A^2, σ_B^2, and σ_E^2 and the variances of those estimators. An outline is shown in Figure 1. We assume that all of the computations below can be done with large enough variable storage that overflow does not occur. This may require an extended precision representation beyond 64 bit floating point, such as that in the python package mpmath (Johansson, 2010).

The first task is to compute $\hat{\theta}$. In a first pass over the data compute counts N, R, C, row values $N_{i\bullet}$, $\bar{Y}_{i\bullet}$, $S_{i\bullet}$ for all unique rows i in the data set, and column values $N_{\bullet j}$, $\bar{Y}_{\bullet j}$, $S_{\bullet j}$ for all unique columns j in the data set as well as $\bar{Y}_{\bullet\bullet}$ and $S_{\bullet\bullet}$. Incremental updates are used as described in (11).

Then compute

$$U_a = \sum_i S_{i\bullet}, \quad U_b = \sum_j S_{\bullet j}, \quad \text{and} \quad U_e = N S_{\bullet\bullet},$$

the matrix M from (10) and then $\hat{\theta} = (\hat{\sigma}_A^2, \hat{\sigma}_B^2, \hat{\sigma}_E^2)^{\mathsf{T}} = M^{-1}(U_a, U_b, U_e)^{\mathsf{T}}$ in time $O(R+C)$.

The second task is to compute approximately the variance of $\hat{\theta}$. A second pass over the data computes the centered fourth moments in (28). Then one calculates the fourth order U-statistics of equation (27), solves (26) for the centered fourth moments, and converts them to kurtosis estimates, all in time $O(R + C)$.

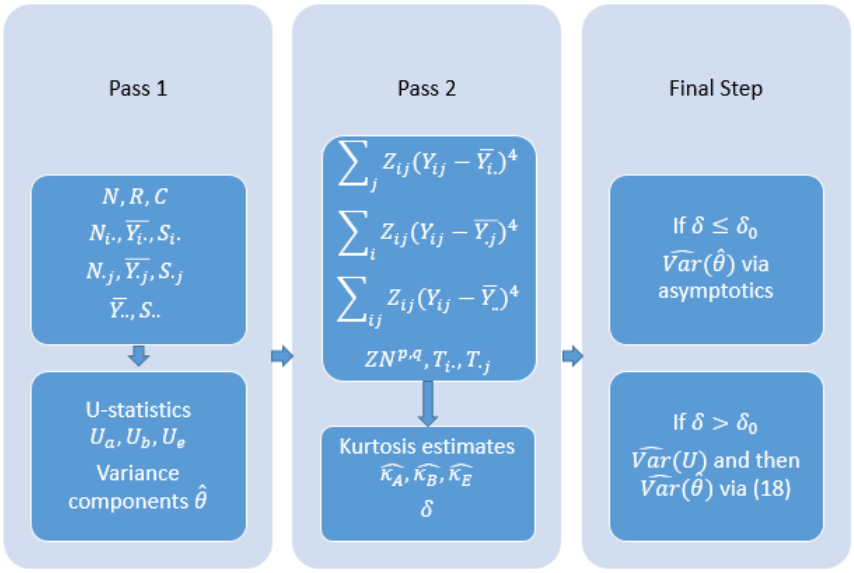


Figure 1: Schematic of our algorithm. The expressions in the smallest boxes are the values computed at each step. The threshold δ_0 is chosen at the discretion of the data analyst and varies between applications.

In the second pass over the data, also compute

$$\mathrm{ZN}^{p,q} \equiv \sum_{ij} Z_{ij} N_{i\bullet}^p N_{\bullet j}^q \tag{29}$$

for

$$\binom{p}{q} \in \left\{ \binom{-1}{-1}, \binom{1}{-1}, \binom{-1}{1}, \binom{1}{1}, \binom{-1}{2}, \binom{2}{-1} \right\}$$

as well as $T_{i\bullet}$ and $T_{\bullet j}$ of equation (2) for all i and j in the data.

Now we may verify whether the limiting approximations in Theorem 4.2 hold. Specifically, compute

$$\delta = \max\left(\epsilon_R, \epsilon_C, \frac{R}{N}, \frac{C}{N}, \frac{N}{\sum_i N_{i\bullet}^2}, \frac{N}{\sum_j N_{\bullet j}^2}, \frac{\sum_{ij} Z_{ij} N_{i\bullet}^{-1} N_{\bullet j}}{\sum_i N_{i\bullet}^2}, \frac{\sum_{ij} Z_{ij} N_{i\bullet} N_{\bullet j}^{-1}}{\sum_j N_{\bullet j}^2}\right)$$

If $\delta \leq \delta_0$, where δ_0 is a user-specified threshold, then we may use

$$\widehat{\mathrm{Var}} \begin{pmatrix} \hat{\sigma}_A^2 \\ \hat{\sigma}_B^2 \\ \hat{\sigma}_E^2 \end{pmatrix} \doteq \frac{1}{N^2} \begin{pmatrix} \hat{\sigma}_A^4(\hat{\kappa}_A + 2) \sum_j N_{\bullet j}^2 & & \\ & \hat{\sigma}_B^4(\hat{\kappa}_B + 2) \sum_i N_{i\bullet}^2 & \\ & & \hat{\sigma}_E^4(\hat{\kappa}_E + 2) N \end{pmatrix}.$$

Otherwise, then more work must be done in the second pass. Some of these next computations require even more bits per variable than are needed to avoid overflow, because they involve subtraction in a way that will lose precision.

In this case, estimate the variances of the U-statistics. To estimate the variances of U_a and U_b, we apply the upper bounds discussed in Section 4.2.2 to (12) and (13) and plug in $\hat{\sigma}_A^2$, $\hat{\sigma}_B^2$, $\hat{\sigma}_E^2$, $\hat{\kappa}_A$, $\hat{\kappa}_B$, and $\hat{\kappa}_E$, calculating using time and space $O(R+C)$

$$\begin{aligned}\widehat{\mathrm{Var}}(U_a) = \hat{\sigma}_B^4(\hat{\kappa}_B+2)\Big(\sum_j N_{\bullet j}^2 - \mathrm{ZN}^{-1,1}\Big) + 2\hat{\sigma}_B^4\Big(\mathrm{ZN}^{-1,1} - R\sum_i N_{i\bullet}^{-1}\Big) \\ + 4\hat{\sigma}_B^2\hat{\sigma}_E^2(N-R) + \hat{\sigma}_E^4(\hat{\kappa}_E+2)\sum_i N_{i\bullet}(1-N_{i\bullet}^{-1})^2 + 2\hat{\sigma}_E^4\sum_i(1-N_{i\bullet}^{-1})\end{aligned}$$

and

$$\begin{aligned}\widehat{\mathrm{Var}}(U_b) = \hat{\sigma}_A^4(\hat{\kappa}_A+2)\Big(\sum_i N_{i\bullet}^2 - \mathrm{ZN}^{1,-1}\Big) + 2\hat{\sigma}_A^4\Big(\mathrm{ZN}^{1,-1} - C\sum_j N_{\bullet j}^{-1}\Big) \\ + 4\hat{\sigma}_A^2\hat{\sigma}_E^2(N-C) + \hat{\sigma}_E^4(\hat{\kappa}_E+2)\sum_j N_{\bullet j}(1-N_{\bullet j}^{-1})^2 + 2\hat{\sigma}_E^4\sum_j(1-N_{\bullet j}^{-1}).\end{aligned}$$

To estimate $\mathrm{Var}(U_e)$ and the covariances of the U-statistics, we again plug in the variance component and kurtosis estimates into Theorem 4.1 without approximation. We get $\widehat{\mathrm{Var}}(U_e)$ from (14), using $\mathrm{ZN}^{1,1}$ from the second pass over the data. We get $\widehat{\mathrm{Cov}}(U_a, U_e)$ from (16) using $\mathrm{ZN}^{-1,1}$, $\mathrm{ZN}^{-1,2}$ and $T_{i\bullet}$, and $\widehat{\mathrm{Cov}}(U_b, U_e)$ from (17) using $\mathrm{ZN}^{1,-1}$, $\mathrm{ZN}^{2,-1}$ and $T_{\bullet j}$. We get $\widehat{\mathrm{Cov}}(U_a, U_b)$ from (15) using $\mathrm{ZN}^{-1,-1}$. It can be easily verified that these calculations also take time and space $O(R+C)$.

The final plug-in estimator of variance is

$$\widehat{\mathrm{Var}}\begin{pmatrix}\hat{\sigma}_A^2\\ \hat{\sigma}_B^2\\ \hat{\sigma}_E^2\end{pmatrix} = M^{-1}\begin{pmatrix}\widehat{\mathrm{Var}}(U_a) & \widehat{\mathrm{Cov}}(U_a,U_b) & \widehat{\mathrm{Cov}}(U_a,U_e)\\ \widehat{\mathrm{Cov}}(U_b,U_a) & \widehat{\mathrm{Var}}(U_b) & \widehat{\mathrm{Cov}}(U_b,U_e)\\ \widehat{\mathrm{Cov}}(U_e,U_a) & \widehat{\mathrm{Cov}}(U_e,U_b) & \widehat{\mathrm{Var}}(U_e)\end{pmatrix}(M^{-1})^{\mathsf{T}} \tag{30}$$

where M is the matrix in (10).

Aggregating the computation times and counting the number of intermediate values we must calculate, we see that our algorithm takes time $O(N)$ and space $O(R+C)$.

5 Predictions

Here we consider an application of variance component estimation to the prediction of a missing observation Y_{ij} at given values of i and j in model (1). An equivalent problem is predicting the expected value at those levels of the factors, $\mu + a_i + b_j = \mathbb{E}(Y_{ij} \mid a_i, b_j)$.

5.1 Best linear predictor

A gold standard is the best linear predictor (BLP), (Searle et al., 2009, Chapter 7.3), which minimizes the MSE over the class of all predictors of the form $\hat{Y}_{ij}(\lambda) = \sum_{rs}\lambda_{rs}Z_{rs}Y_{rs}$, where λ is the vector of all λ_{rs}. In this section, we characterize the weights λ_{rs}^* of the BLP. We begin with the MSE

$$L(\lambda) = \mathbb{E}((\hat{Y}_{ij}(\lambda) - Y_{ij})^2) \tag{31}$$

Lemma 5.1. *The MSEs for the linear predictor* $\sum_{rs}\lambda_{rs}Z_{rs}Y_{rs}$ *are*

$$\begin{aligned}L(\lambda) &= \mu^2\Big(1-\sum_{rs}\lambda_{rs}Z_{rs}\Big)^2+\sigma_A^2+\sigma_B^2+\sigma_E^2\\ &\quad+\sigma_A^2\sum_{rss'}\lambda_{rs}\lambda_{rs'}Z_{rs}Z_{rs'}+\sigma_B^2\sum_{rsr'}\lambda_{rs}\lambda_{r's}Z_{rs}Z_{r's}+\sigma_E^2\sum_{rs}\lambda_{rs}^2Z_{rs}\\ &\quad-2\Big(\sigma_A^2\sum_s\lambda_{is}Z_{is}+\sigma_B^2\sum_r\lambda_{rj}Z_{rj}+\sigma_E^2\lambda_{ij}^2Z_{ij}\Big).\end{aligned}\tag{32}$$

Proof. See Section 18.1 of the supplement. □

The weights λ_{rs}^* of the BLP must satisfy the stationarity condition $\partial L(\lambda_{rs}^*)/\partial\lambda = 0$. As shown in Section 18.2 of the supplement, when $Z_{rs}=0$, the condition holds no matter the value of λ_{rs}^*. When $Z_{rs}=1$, the condition becomes

$$\sigma_E^2\lambda_{rs}^* = \mu^2\Big(1-\sum_{r's'}\lambda_{r's'}^*Z_{r's'}\Big)+\sigma_A^2\Big(1_{i=r}-\sum_{s'}\lambda_{rs'}^*Z_{rs'}\Big)+\sigma_B^2\Big(1_{j=s}-\sum_{r'}\lambda_{r's}^*Z_{r's}\Big)\tag{33}$$

We can compute λ_{rs}^* by solving an $N\times N$ system of equations but that ordinarily costs $O(N^3)$ time. Shortcuts are possible if there is a special pattern in the Z_{ij}, such as balanced data, but we don't know of any faster way to solve (33) for general Z. Therefore, we consider a smaller class of linear predictors called shrinkage predictors.

5.2 Shrinkage predictors

It is reasonable to suppose that the most important observations for predicting Y_{ij} are those in its row and column. Therefore we consider predicting Y_{ij} through a linear combination of the overall average, the average in row i, and the average in column j. We use estimators of the form

$$\hat{Y}_{ij}(\lambda) = \lambda_0\sum_{rs}Z_{rs}Y_{rs}+\lambda_a\sum_s Z_{is}Y_{is}+\lambda_b\sum_r Z_{rj}Y_{rj}\tag{34}$$

where $\lambda = \begin{pmatrix}\lambda_0 & \lambda_a & \lambda_b\end{pmatrix}^{\mathsf{T}}$. Then t$\lambda_0$, λ_a, and λ_b are chosen to minimize $L(\lambda)$. By writing (34) in terms of row and column totals we avoid complicated treatments for the situation where row or column means are unavailable because $N_{i\bullet}=0$ or $N_{\bullet j}=0$ (or both). As an example, if $\min(N_{i\bullet},N_{\bullet j})>0$, then the predictor $\hat{Y}_{ij}=\bar{Y}_{i\bullet}+\bar{Y}_{\bullet j}-\bar{Y}_{\bullet\bullet}$ (from Theorem 5.3 below) has $\lambda_0=-1/N$, $\lambda_a=1/N_{i\bullet}$ and $\lambda_b=1/N_{\bullet j}$.

Lemma 5.2. *The MSEs for the linear predictor* (34) *are*

$$\begin{aligned}L(\lambda) &= \mu^2\big(1-\lambda_0 N-\lambda_a N_{i\bullet}-\lambda_b N_{\bullet j}\big)^2+\lambda_0^2\Big(\sigma_A^2\sum_r N_{r\bullet}^2+\sigma_B^2\sum_s N_{\bullet s}^2+\sigma_E^2 N\Big)\\
&+\lambda_a^2\Big(\sigma_A^2 N_{i\bullet}^2+\sigma_B^2 N_{i\bullet}+\sigma_E^2 N_{i\bullet}\Big)+\lambda_b^2\Big(\sigma_A^2 N_{\bullet j}+\sigma_B^2 N_{\bullet j}^2+\sigma_E^2 N_{\bullet j}\Big)+\sigma_A^2+\sigma_B^2+\sigma_E^2\\
&-2\lambda_0\Big(\sigma_A^2 N_{i\bullet}+\sigma_B^2 N_{\bullet j}+\sigma_E^2 Z_{ij}\Big)-2\lambda_a\Big(\sigma_A^2 N_{i\bullet}+\sigma_B^2 Z_{ij}+\sigma_E^2 Z_{ij}\Big)\\
&-2\lambda_b\Big(\sigma_A^2 Z_{ij}+\sigma_B^2 N_{\bullet j}+\sigma_E^2 Z_{ij}\Big)+2\lambda_0\lambda_a\Big(\sigma_A^2 N_{i\bullet}^2+\sigma_B^2\sum_s Z_{is}N_{\bullet s}+\sigma_E^2 N_{i\bullet}\Big)\\
&+2\lambda_0\lambda_b\Big(\sigma_A^2\sum_r Z_{rj}N_{r\bullet}+\sigma_B^2 N_{\bullet j}^2+\sigma_E^2 N_{\bullet j}\Big)+2\lambda_a\lambda_b Z_{ij}\Big(\sigma_A^2 N_{i\bullet}+\sigma_B^2 N_{\bullet j}+\sigma_E^2\Big).\end{aligned}$$

Proof. See Section 18.3 of the supplement. □

Theorem 5.1. *The λ^* that minimizes the MSE $L=\mathbb{E}((\hat{Y}_{ij}-Y_{ij})^2)$ satisfies $H\lambda^*=c$, where*

$$c=\begin{pmatrix}N & N_{i\bullet} & N_{\bullet j} & Z_{ij}\\ N_{i\bullet} & N_{i\bullet} & Z_{ij} & Z_{ij}\\ N_{\bullet j} & Z_{ij} & N_{\bullet j} & Z_{ij}\end{pmatrix}\begin{pmatrix}\mu^2\\ \sigma_A^2\\ \sigma_B^2\\ \sigma_E^2\end{pmatrix},\quad \textit{and}\quad H=\begin{pmatrix}H_{11} & H_{12} & H_{13}\\ * & H_{22} & H_{23}\\ * & * & H_{33}\end{pmatrix}$$

is a symmetric matrix with upper triangular elements

$$\begin{aligned}H_{11}&=\mu^2N^2+\sigma_A^2\sum_r N_{r\bullet}^2+\sigma_B^2\sum_s N_{\bullet s}^2+\sigma_E^2N\\
H_{12}&=\mu^2NN_{i\bullet}+\sigma_A^2N_{i\bullet}^2+\sigma_B^2T_{i\bullet}+\sigma_E^2N_{i\bullet}\\
H_{13}&=\mu^2NN_{\bullet j}+\sigma_A^2T_{\bullet j}+\sigma_B^2N_{\bullet j}^2+\sigma_E^2N_{\bullet j}\\
H_{22}&=\mu^2N_{i\bullet}^2+\sigma_A^2N_{i\bullet}^2+\sigma_B^2N_{i\bullet}+\sigma_E^2N_{i\bullet}\\
H_{23}&=\mu^2N_{i\bullet}N_{\bullet j}+\sigma_A^2Z_{ij}N_{i\bullet}+\sigma_B^2Z_{ij}N_{\bullet j}+\sigma_E^2Z_{ij},\quad \textit{and}\\
H_{33}&=\mu^2N_{\bullet j}^2+\sigma_A^2N_{\bullet j}+\sigma_B^2N_{\bullet j}^2+\sigma_E^2N_{\bullet j}.\end{aligned}$$

Proof. See Section 18.4 of the supplement. □

Given estimates of μ and θ we can plug them in to get estimates of the optimal λ for prediction at (i,j). Assuming that the algorithm to compute $\hat{\theta}$ and its variance has been executed, all of c and most of H can be computed using quantities found in the first pass over the data. All of the quantities (2) are available after a second pass.

Therefore, since solving $H\lambda^*=c$ takes time $O(1)$, λ^* for predicting a given Y_{ij} can be found in time $O(N)$. If we wanted to find λ^* for k different sets of i and j, the computation cost is $O(N+k)$; we simply would have to store k different H's and c's.

Predicting a missing Y_{ij} using Theorem 5.1 is simple. Next we look at some special cases to understand how it performs.

Special case: Y_{ij} in new row and new column

In this case, $N_{rj} = N_{is} = 0$ for any r, s, and $N_{i\bullet} = N_{\bullet j} = 0$. The only nonzero entry of H is $H_{11} = \mu^2 N^2 + \sigma_A^2 \sum_r N_{r\bullet}^2 + \sigma_B^2 \sum_s N_{\bullet s}^2 + \sigma_E^2 N$, and the only nonzero entry of c is $c_1 = \mu^2 N$. Hence $\lambda_a^* = \lambda_b^* = 0$ and

$$\lambda_0^* = \frac{\mu^2 N}{\mu^2 N^2 + \sigma_A^2 \sum_r N_{r\bullet}^2 + \sigma_B^2 \sum_s N_{\bullet s}^2 + \sigma_E^2 N}.$$

The prediction $\hat{Y}_{ij}$ is then a shrinkage

$$\lambda_0^* Y_{\bullet\bullet} = N\lambda_0^* \bar{Y}_{\bullet\bullet} = \frac{\mu^2}{\mu^2 + \sigma_A^2 \sum_r N_{r\bullet}^2/N^2 + \sigma_B^2 \sum_s N_{\bullet s}^2/N^2 + \sigma_E^2/N} \bar{Y}_{\bullet\bullet}.$$

In practice we would plug in estimates of μ and the variance components. As we would expect, this estimate is very close to $\bar{Y}_{\bullet\bullet}$ for large N, when $\hat{\mu} \neq 0$ and the limits (6) hold. In that case, the corresponding MSE is $L \doteq \sigma_A^2 + \sigma_B^2 + \sigma_E^2$, which can be verified to be approximately the same as the MSE of the BLP.

Special case: Y_{ij} in new row but old column

Suppose that $Z_{is} = 0$ for any s but $\exists r$ where $Z_{rj} = 1$, so $N_{i\bullet} = 0$ and $N_{\bullet j} > 0$. We would expect most of the weight to be on $\bar{Y}_{\bullet j}$, the average in the column containing Y_{ij}. This is indeed the case if $T_{\bullet j}$ is not large compared to N, that is, if the rows that are co-observed with column j do not comprise a large fraction of the data.

Let c_k denote the kth entry of c and $H_{k\ell}$ be the entry of H in row k and column ℓ. In this case, c_2 is zero as is the second row and second column of H. Therefore, without loss of generality we can take $\lambda_a^* = 0$ and $\tilde{\lambda}^* = \begin{pmatrix} \lambda_0^* & \lambda_b^* \end{pmatrix}^\mathsf{T}$ can be computed by solving the system $\tilde{H}\tilde{\lambda}^* = \tilde{c}$, where

$$\tilde{H} = \begin{pmatrix} H_{11} & H_{13} \\ H_{31} & H_{33} \end{pmatrix} \quad \text{and} \quad \tilde{c} = \begin{pmatrix} c_1 \\ c_3 \end{pmatrix}.$$

The following theorem describes the relative size of λ_0^* and λ_b^* in the big data limit.

Theorem 5.2. *Suppose that we are predicting Y_{ij} where $N_{i\bullet} = 0$ but $N_{\bullet j} > 0$. Assume that $0 < \mu^2, \sigma_A^2, \sigma_B^2, \sigma_E^2 < \infty$ and that $T_{\bullet j} \equiv \sum_r N_{r\bullet} Z_{rj} \leqslant \eta N$. Then*

$$\frac{\lambda_0^*}{\lambda_b^*} = \frac{1}{N} \frac{\sigma_A^2 + \sigma_E^2}{\sigma_B^2} (1 + O(\eta))$$

as $\eta \to 0$.

Proof. See Section 18.5 of the Supplement. □

Note that λ_0^* is the coefficient of a sum of N observations, while λ_b^* is the coefficient of a sum of $N_{\bullet j}$ observations. Therefore, to more equitably compare the importances of the overall average and the column average for predicting Y_{ij}, we consider the ratio

$$\frac{N\lambda_0^*}{N_{\bullet j}\lambda_b^*} \approx \frac{\sigma_A^2 + \sigma_E^2}{\sigma_B^2 N_{\bullet j}}.$$

We may interpret this as the column j average being some multiple of $N_{\bullet j}$ times as important as the overall average. This makes sense because the more data we have in column j, the better estimate we would be able to get of $\mu + b_j$; the overall average only tells us about μ. Also, note that the larger σ_E^2 is relative to σ_B^2, the more weight we put on the overall average; we do not trust using only the column average.

Special case: large $N_{i\bullet}$ and large $N_{\bullet j}$

Next we show that if both row i and column j have a very large number of observations, and the observation matrix Z is not too extreme, then $\hat{Y}_{ij}$ is approximately $\bar{Y}_{i\bullet} + \bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet}$ as we might expect. As a result, the customized weights in Theorem 5.1 are most useful for cases where one or both of $N_{i\bullet}$ and $N_{\bullet j}$ are not very large.

Theorem 5.3. *Suppose that* $1/\eta \leqslant N_{i\bullet} \leqslant \eta N$ *and* $1/\eta \leqslant N_{\bullet j} \leqslant \eta N$ *both hold for some* $\eta \in (0, 1)$ *and that* $0 < \mu^2, \sigma_A^2, \sigma_B^2, \sigma_E^2 < \infty$. *Then*

$$\hat{Y}_{ij} = (\bar{Y}_{i\bullet} + \bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet})(1 + O(\eta)), \quad \text{as } \eta \to 0.$$

Proof. See Section 19 in the supplement. □

6 Experimental Results

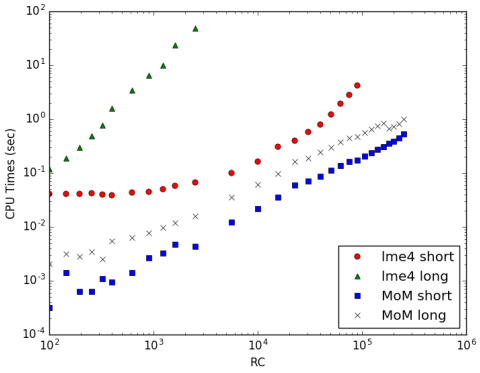
6.1 Simulations

First, we compare the performance of our method of moments algorithm ('MoM'), described in Section 4.3, to the commonly used R package for mixed models, lme4. lme4 computes the maximum likelihood estimates of the parameters under an assumption of normality.

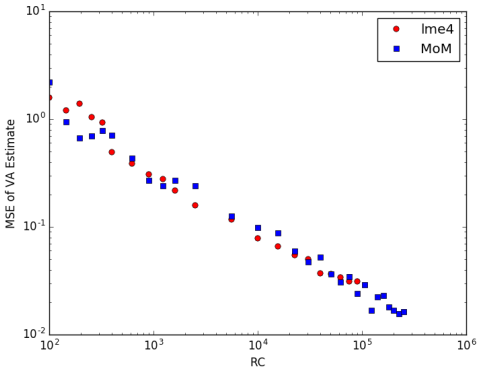
For our algorithm, we consider a range of data sizes, with $R = C$ ranging from 10 to 500. At each fixed value of $R = C$, for 100 iterations, we generate data according to model (1) with normally distributed random effects and $\sigma_A^2 = 2$, $\sigma_B^2 = 0.5$, and $\sigma_E^2 = 1$. Exactly 25 percent of the cells were randomly chosen to be observed. We measure the CPU time needed to obtain the variance component estimates $\hat{\sigma}_A^2$, $\hat{\sigma}_B^2$, and $\hat{\sigma}_E^2$ (labeled short) and the CPU time need to obtain the variance component estimates as well as upper bounds on the variances of those estimates (labeled long). In addition, we measure the mean squared errors of the variance component estimates. At the end, those five measurements were averaged over the 100 iterations.

With regard to lme4, our simulation steps are nearly the same, with the following differences. Due to the slowness of lme4, we only consider data sizes with $R = C$ up to 300. In addition, because lme4 finds the maximum likelihood variance component estimates, the variances of those estimates were computed asymptotically using the inverse expected Fisher information matrix. The simulation results are shown in Figure 2.

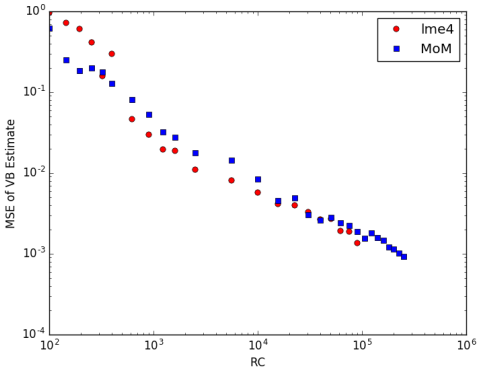
Note that lme4 always takes more time than our algorithm. From Figure 2a, we see that our method of moments algorithm takes time at most linear in the data size to compute both the variance component estimates and upper bounds on the variances of those estimates. For lme4 the computation time is clearly superlinear in the data size, for data sets large enough that the startup cost of the package is no longer dominant.



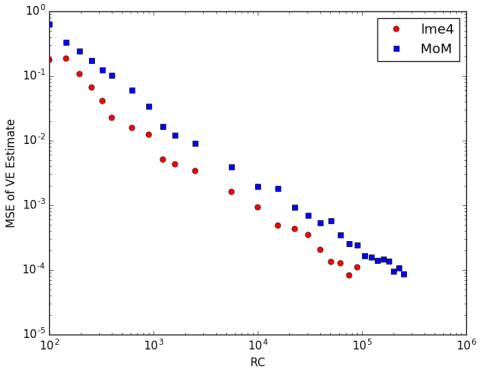
(a) CPU



(b) MSE of $\hat{\sigma}_A^2$



(c) MSE of $\hat{\sigma}_B^2$



(d) MSE of $\hat{\sigma}_E^2$

Figure 2: Simulation results: log-log plots of the five recorded measurements against $R * C$, which is proportional to the number of observations. The slope of a fitted line through the scatterplot describes the effect of the x-axis quantity on the y-axis quantity; a slope of 1 indicates a linear relationship, greater than 1 a superlinear relationship, and less than 1 a sublinear relationship.

The MSEs of $\hat{\sigma}_A^2$ for our algorithm and lme4 are comparable. Moreover, both decrease at most linearly with the data size. The same is true for the MSEs of $\hat{\sigma}_B^2$. However, the MSE of $\hat{\sigma}_E^2$ in lme4 is noticeably smaller than that of our algorithm; this appears to be the price we pay for the decreased computation time. In both cases, though, the MSE of $\hat{\sigma}_E^2$ decreases approximately linearly with the data size.

6.2 Real World Data

We illustrate our algorithm, coded in Python, on three real world data sets that are too large for lme4 to handle in a timely manner.

The first, from Yahoo!-Webscope (2015a), contains a random sample of ratings of movies by users, which are grades from A+ to F converted into a numeric scale. There are $211,231$ ratings by $7,642$ users on $11,916$ movies, filtered with the condition that each user rates at least ten movies. Only 0.23 percent of the user-movie matrix is observed.

The estimated variances of the user random effect, the movie random effect, and the error are 2.57, 2.86, and 7.68. The estimated kurtoses are -2, -2, and 6.56. Estimated upper bounds on the variances of the estimated variance components are 0.0030, 0.0018, and 0.0060.

The second data set, also from Yahoo!-Webscope (2015b), contains ratings of 1000 songs by 15400 users, on a scale of 1 to 5. The first group of 10000 users were randomly selected on the condition that they had rated at least 10 of the 1000 songs. The rest of the users were randomly selected from responders on a survey that asked them to rate a random subset of 10 of the 1000 songs. The songs were selected to have at least 500 ratings. Here, about 2 percent of the user-song pairs were observed.

The estimated variances of the user random effect, the song random effect, and the error are 0.97, 0.24, and 1.30. The estimated kurtoses are -2, -2, and 3.31. Estimated upper bounds on the variances of the estimated variance components are 4.5×10^{-5}, 10^{-5}, and 5.8×10^{-5}. For determining the rating, the user effect is dominant over the song effect.

The third data set from Last.fm (2015) contains the numbers of times artists' songs are played by about $360,000$ users. Only the counts for the top k (for some k) artists for each user is recorded. The users are randomly selected. This data set is extremely sparse; only about 0.03 percent of user-artist pairs are observed.

The estimated variances of the user random effect, the artist random effect, and the error are 1.65, 0.22, and 0.27. The estimated kurtoses are 0.019, -2, and 23.14. Estimated upper bounds on the variances of the estimated variance components are 1.68×10^{-5}, 4.06×10^{-7}, and 1.37×10^{-6}. The biggest source of variation in the number of plays is the user, not the artist. The kurtosis of the row effect is nearly zero, indicating possible normality.

In all three data sets at least one of the estimated kurtoses was -2, which would be unexpected if the model is correctly specified. However, if model (1) does not fit the data well, such behavior may occur. For example, the expected rating of a movie may not be additively decomposable into a movie effect, a user effect, and an error.

7 Conclusion

When traditional maximum likelihood or MCMC methods are used, with both theory and simulations, we have found that fitting large two-factor crossed unbalanced random effects models has costs that are superlinear in the number of data points, N. With the method of moments it is possible to get, in linear time, parameter estimates and somewhat conservative estimates of their variance. The space requirements are proportional to the number of distinct levels of the factors entities; this will often be sublinear in N. We also developed shrinkage predictors of missing data that utilize our method of moments estimates.

Through simulations on normally distributed data, we show that our method of moments estimates are competitive with maximum likelihood estimates. We trade off a small increase in the MSE of one variance component for a dramatic decrease in computation time as N gets large.

As stated in the introduction, the crossed random effects model we consider here is the simplest one for which we felt that there was no useful prior solution. We expect that richer models, which are the basis of our future work, will provide better fits to real world data.

In some cases we may be expecting a repeat observation in the ij-cell and then it may be possible to get a better estimate of $\mu + a_i + b_j$ than Y_{ij} is. Section 20 of the supplement considers this problem.

7.1 Informative Missingness

We have assumed throughout that the missingness pattern in Z_{ij} is not informative. But in many applications the observed values are likely to differ in some way from the missing values. For instance, in movie ratings data people may be more likely to watch and rate movies they believe they will like, and so missing values could be lower on average than observed ones. In general, the observed ratings may have both high and low values oversampled relative to middling values.

From observed values alone we cannot tell how different the missing values would be. To do so requires making untestable assumptions about the missingness mechanism. Even in cases where followup sampling can be made, e.g., giving some users incentives to make additional ratings, there will still be difficulties such as users refusing to make those ratings, or if forced, making inaccurate ratings. Methods to adjust for missingness have to be designed on a case by case basis, using whatever additional data and assumptions can be brought to bear. The uncertainties of the estimates from such methods can be quantified using, with further development, the techniques of this paper.

Acknowledgments

This work was supported by US NSF under grant DMS-1407397. KG was supported by US NSF Graduate Research Fellowship under grant DGE-114747. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

We would like to thank Brad Klingenberg for his motivation and encouragement during this project. We would also like to thank Rob Tibshirani for his suggestions about our

experiments, and Lester Mackey and Norm Matloff for some helpful discussions.

References

Bates, D. (2014). Computational methods for mixed models. `https://cran.r-project.org/web/packages/lme4/vignettes/Theory.pdf`.

Bennett, J. and Lanning, S. (2007). The Netflix prize. In *Proceedings of KDD Cup and Workshop 2007*.

Chan, T. F., Golub, G. H., and LeVeque, R. J. (1983). Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37(3):242–247.

Clayton, D. and Rasbash, J. (1999). Estimation in large cross random-effect models by data augmentation. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 162(3):425–436.

Dyer, J. S. and Owen, A. B. (2011). Visualizing bivariate long-tailed data. *Electronic Journal of Statistics*, 5:642–668.

Gelman, A., Van Dyk, D. A., Huang, Z., and Boscardin, J. W. (2012). Using redundant parameterizations to fit hierarchical models. *Journal of Computational and Graphical Statistics*.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Hairer, M., Stuart, A. M., and Vollmer, S. J. (2014). Spectral gaps for a Metropolis Hastings algorithm in infinite dimensions. *The Annals of Applied Probability*, 24(6):2455–2490.

Henderson, C. R. (1953). Estimation of variance and covariance components. *Biometrics*, 9(2):226–252.

Johansson, F. (2010). *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.14)*. `http://code.google.com/p/mpmath/`.

Last.fm (2015). Dataset - 360k users. `http://ocelma.net/MusicRecommendationDataset/lastfm-360K.html`. `http://www.last.fm/`.

Lavrakas, P. J. (2008). *Encyclopedia of Survey Research Methods: A–M*, volume 1. Sage Publications, Inc., Thousand Oaks, CA.

Liu, J. S. (2004). *Monte Carlo Strategies in Scientific Computing*. Springer, New York.

Owen, A. B. (2007). The pigeonhole bootstrap. *The Annals of Applied Statistics*, 1(2):386–411.

Owen, A. B. and Eckles, D. (2012). Bootstrapping data arrays of arbitrary order. *The Annals of Applied Statistics*, 6(3):895–927.

Pébay, P. (2008). Formulas for robust, one-pass parallel computation of covariances and arbitrary-order statistical moments. Technical Report SAND2008-6212, Sandia National Laboratories.

Raudenbush, S. W. (1993). A crossed random effects model for unbalanced data with applications in cross-sectional and longitudinal research. *Journal of Educational and Behavioral Statistics*, 18(4):321–349.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351–367.

Roberts, G. O. and Sahu, S. K. (1997). Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *Journal of the Royal Statistical Society. Series B*, pages 291–317.

Searle, S. R., Casella, G., and McCulloch, C. E. (2009). *Variance components*. John Wiley & Sons, New York.

Snijders, T. A. (2014). Multilevel analysis. In Lovric, M., editor, *International Encyclopedia of Statistical Science*, pages 879–882. Springer, Berlin.

Van Dyk, D. A. and Meng, X.-L. (2001). The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1).

Yahoo!-Webscope (2015a). Dataset ydata-ymovies-user-movie-ratings-train-v1_0. `http://research.yahoo.com/Academic_Relations`.

Yahoo!-Webscope (2015b). Dataset ydata-ymusic-rating-study-v1_0-train. `http://research.yahoo.com/Academic_Relations`.

Yu, Y. and Meng, X.-L. (2011). To center or not to center: That is not the question—an ancillarity–sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics*, 20(3):531–570.

8 Appendix

8.1 Proof of Theorem 2.1

In the balanced case we may assume that $i \in \{1, 2, \ldots, R\}$ and $j \in \{1, 2, \ldots, C\}$. The posterior distribution of the parameters is given by

$$p(\mu, a, b, \sigma_A^2, \sigma_B^2, \sigma_E^2 \mid Y) \propto \prod_{i=1}^{R} \frac{1}{\sqrt{2\pi\sigma_A^2}} \exp\Big(-\frac{a_i^2}{2\sigma_A^2}\Big) \prod_{j=1}^{C} \frac{1}{\sqrt{2\pi\sigma_B^2}} \exp\Big(-\frac{b_j^2}{2\sigma_B^2}\Big)$$
$$\times \prod_{i=1}^{R}\prod_{j=1}^{C} \frac{1}{\sqrt{2\pi\sigma_E^2}} \exp\Big(-\frac{(Y_{ij} - \mu - a_i - b_j)^2}{2\sigma_E^2}\Big)$$

$$\propto \sigma_A^{-R}\sigma_B^{-C}\sigma_E^{-RC}\exp\Big(-\frac{\sum_i a_i^2}{2\sigma_A^2}-\frac{\sum_j b_j^2}{2\sigma_B^2}-\frac{\sum_{ij}(Y_{ij}-\mu-a_i-b_j)^2}{2\sigma_E^2}\Big)$$

Then, ϕ is given by

$$p(a,b\mid\mu,\sigma_A^2,\sigma_B^2,\sigma_E^2,Y)\propto\exp\Big(-\frac{\sum_i a_i^2}{2}\Big(\frac{1}{\sigma_A^2}+\frac{C}{\sigma_E^2}\Big)-\frac{\sum_j b_j^2}{2}\Big(\frac{1}{\sigma_B^2}+\frac{R}{\sigma_E^2}\Big)-\frac{\sum_{ij}a_ib_j}{\sigma_E^2}\Big).$$

Therefore, the posterior distribution of a and b is a joint normal with precision matrix

$$Q=\begin{pmatrix}\dfrac{\sigma_E^2+C\sigma_A^2}{\sigma_A^2\sigma_E^2}I_R & \dfrac{1}{\sigma_E^2}1_R1_C^\intercal\\ \dfrac{1}{\sigma_E^2}1_C1_R^\intercal & \dfrac{\sigma_E^2+R\sigma_B^2}{\sigma_B^2\sigma_E^2}I_C\end{pmatrix}.$$

From Theorem 1 of Roberts and Sahu (1997), for the Gibbs sampler described in Section 2.1, we have the following result. Let $A=I-\mathrm{diag}(Q_{11}^{-1},Q_{22}^{-1})Q$, where Q_{11} denotes the upper left block of Q and Q_{22} denotes the lower right block. Let L be the block lower triangular part of A, and $U=A-L$. Then, the convergence rate ρ is given by the spectral radius of the matrix $B=(I-L)^{-1}U$. Now, we compute ρ. First

$$A=I-\begin{pmatrix}\dfrac{\sigma_A^2\sigma_E^2}{\sigma_E^2+C\sigma_A^2}I_R & 0\\ 0 & \dfrac{\sigma_B^2\sigma_E^2}{\sigma_E^2+R\sigma_B^2}I_C\end{pmatrix}Q=\begin{pmatrix}0 & -\dfrac{\sigma_A^2}{\sigma_E^2+C\sigma_A^2}1_R1_C^\intercal\\ -\dfrac{\sigma_B^2}{\sigma_E^2+R\sigma_B^2}1_C1_R^\intercal & 0\end{pmatrix}.$$

Next

$$L=\begin{pmatrix}0 & 0\\ -\dfrac{\sigma_B^2}{\sigma_E^2+R\sigma_B^2}1_C1_R^\intercal & 0\end{pmatrix}\quad\text{and}\quad U=\begin{pmatrix}0 & -\dfrac{\sigma_A^2}{\sigma_E^2+C\sigma_A^2}1_R1_C^\intercal\\ 0 & 0\end{pmatrix}$$

from which

$$\begin{aligned}B&=\begin{pmatrix}I_R & 0\\ \dfrac{\sigma_B^2}{\sigma_E^2+R\sigma_B^2}1_C1_R^\intercal & I_C\end{pmatrix}^{-1}U=\begin{pmatrix}I_R & 0\\ -\dfrac{\sigma_B^2}{\sigma_E^2+R\sigma_B^2}1_C1_R^\intercal & I_C\end{pmatrix}U\\&=\begin{pmatrix}0 & -\dfrac{\sigma_A^2}{\sigma_E^2+C\sigma_A^2}1_R1_C^\intercal\\ 0 & \dfrac{R\sigma_A^2\sigma_B^2}{(\sigma_E^2+C\sigma_A^2)(\sigma_E^2+R\sigma_B^2)}1_C1_C^\intercal\end{pmatrix}.\end{aligned}$$

Clearly, B has rank one. Then, its spectral radius must be equal to its nonzero eigenvalue, which is also the trace of B. Hence,

$$\rho=\frac{RC\sigma_A^2\sigma_B^2}{(\sigma_E^2+C\sigma_A^2)(\sigma_E^2+R\sigma_B^2)}$$

8.2 Simulation results

The results of our simulations described in Section 2 are presented here in Tables 2 through 6.

Method	Gibbs	Block	Reparam.	Lang.	MALA	Indp.	RWM	RWM Sub.	pCN
R=10 C=10	20	9	23	20	27	21	19	21	21
R=20 C=20	33	10	37	35	45	34	32	33	33
R=50 C=50	71	17	80	79	101	71	68	75	70
R=100 C=100	143	361	159	156	199	139	133	141	136
R=200 C=200	326	984	351	323	462	300	279	303	280
R=500 C=500	1157	2356	1205	955	1786	952	851	1019	817
R=1000 C=1000	3432	15046	4099	2302	4760	2513	2141	2635	1966
R=2000 C=2000	10348	88756	11434	6991	15836	7815	5712	9274	6006
R=50 C=100	105	287	121	112	151	103	101	107	102
R=10 C=200	138	316	167	139	200	138	137	142	138
R=100 C=1000	898	5148	964	807	1179	795	748	822	760

Table 2: Median CPU time in seconds.

Method	Gibbs	Block	Reparam.	Lang.	MALA	Indp.	RWM	RWM Sub.	pCN
R=10	0.72	0.94	1.27	1.07	1.18	2.40	0.76	0.74	1.51
C=10	26	29	24	178	689	1604	1252	1522	1392
R=20	0.81	1.02	1.01	1.07	0.94	2.89	1.69	1.08	1.47
C=20	34	43	26	75	841	1019	1674	1720	1765
R=50	1.09	0.91	0.98	0.98	1.04	2.97	1.66	1.70	1.58
C=50	83	84	75	8	610	5000+	1158	1681	1104
R=100	0.98	1.02	1.13	0.99	0.85	2.73	1.57	1.61	1.49
C=100	123	185	144	2	398	5000+	1145	1713	1522
R=200	1.01	1.02	1.03	1.01	0.95	3.22	1.60	1.31	1.52
C=200	257	346	272	1	1	1278	1508	1692	807
R=500	0.99	1.01	0.99	0.99	1.00	2.26	1.58	1.15	1.55
C=500	536	617	576	9	4	1572	924	1687	1613
R=1000	0.97	1.02	1.04	0.99	0.96	2.39	1.55	1.07	1.53
C=1000	801	790	694	1	2501	5000+	1133	1656	1008
R=2000	0.98	1.01	1.00	1.01	1.00	2.57	1.55	1.03	1.55
C=2000	672	721	771	1	5000+	1086	1176	1716	799
R=50	0.89	1.03	0.95	1.01	1.06	2.70	1.57	1.61	1.45
C=100	144	155	118	7	1095	5000+	1219	1725	1371
R=10	0.86	1.08	0.84	0.94	0.80	2.40	1.41	1.36	1.23
C=200	329	244	299	120	944	3339	1518	1657	1437
R=100	1.06	1.06	1.02	1.01	1.03	2.73	1.57	1.11	1.55
C=1000	573	536	672	1	1	3330	1161	1681	3333

Table 3: Median estimates of μ and lag when $\text{ACF}(\hat{\mu}) \leq 0.5$.

Method	Gibbs	Block	Reparam.	Lang.	MALA	Indp.	RWM	RWM Sub.	pCN
R=10	2.76	2.49	2.05	2.07	2.45	2.39	1.88	2.05	1.38
C=10	1	1	1	898	768	1604	759	606	1232
R=20	2.00	2.06	1.65	1.89	2.32	1.48	1.96	1.76	2.00
C=20	1	1	1	930	829	850	873	822	1083
R=50	1.94	1.96	2.17	1.77	2.21	1.44	2.06	2.03	1.95
C=50	1	1	1	797	720	5000+	1035	1032	1079
R=100	2.21	2.14	2.23	1.88	1.87	1.11	2.19	1.92	1.95
C=100	1	1	1	649	398	5000+	994	917	1522
R=200	2.09	2.09	2.10	2.08	1.99	1.16	2.02	2.12	2.01
C=200	1	1	1	410	437	1281	1598	673	1135
R=500	1.97	2.12	1.99	1.64	1.96	1.07	2.02	2.01	1.97
C=500	1	1	1	407	197	1572	895	826	1599
R=1000	1.96	1.99	2.02	1.90	1.95	1.78	2.01	1.96	1.99
C=1000	1	1	1	122	2656	5000+	1133	989	912
R=2000	1.97	2.00	2.03	1.94	1.99	1.04	2.01	2.00	1.99
C=2000	1	1	1	69	5000+	1086	1181	1262	1161
R=50	2.22	2.29	2.05	2.24	1.98	1.10	2.00	1.96	2.09
C=100	1	1	1	948	672	5000+	1103	787	1005
R=10	2.34	1.74	3.05	2.70	2.72	0.88	1.89	1.43	1.16
C=200	1	1	1	891	1023	3309	1492	724	988
R=100	2.04	2.03	2.14	1.98	1.98	1.46	1.90	1.87	2.05
C=1000	1	1	1	512	450	3329	985	1086	3333

Table 4: Median estimates of σ_A^2 and lag when ACF($\hat{\sigma_A^2}$) ≤ 0.5

Method	Gibbs	Block	Reparam.	Lang.	MALA	Indp.	RWM	RWM Sub.	pCN
R=10	0.66	0.81	0.88	0.46	0.89	1.47	0.45	0.43	0.45
C=10	1	1	1	382	638	1604	1214	956	1297
R=20	0.54	0.45	0.44	0.43	0.44	1.55	0.49	0.46	0.57
C=20	1	1	1	261	410	978	937	1217	704
R=50	0.49	0.49	0.49	0.49	0.53	1.35	0.49	0.43	0.48
C=50	1	1	1	123	138	5000+	1308	786	1463
R=100	0.51	0.54	0.49	0.46	0.48	0.84	0.52	0.47	0.49
C=100	1	1	1	65	66	5000+	691	1169	1522
R=200	0.49	0.51	0.51	0.47	0.50	1.67	0.51	0.49	0.50
C=200	1	1	1	36	37	1266	1497	1241	831
R=500	0.51	0.49	0.50	0.28	0.47	1.56	0.50	0.48	0.47
C=500	1	1	1	770	16	1572	696	993	1619
R=1000	0.51	0.50	0.50	0.40	0.50	2.94	0.51	0.50	0.49
C=1000	1	1	1	477	2514	5000+	1133	855	556
R=2000	0.50	0.50	0.49	0.39	0.50	1.65	0.48	0.49	0.50
C=2000	1	1	1	224	5000+	1086	1220	830	1253
R=50	0.50	0.51	0.53	0.48	0.54	1.93	0.53	0.49	0.49
C=100	1	1	1	69	85	5000+	1378	910	1419
R=10	0.47	0.51	0.51	0.40	0.52	1.65	0.61	0.59	0.55
C=200	1	1	1	23	52	3332	1289	1004	1408
R=100	0.50	0.49	0.50	0.47	0.49	2.95	0.50	0.49	0.50
C=1000	1	1	1	6	8	3328	1345	962	3333

Table 5: Median estimates of σ_B^2 and lag when $\text{ACF}(\hat{\sigma_B^2}) \leq 0.5$

Method	Gibbs	Block	Reparam.	Lang.	MALA	Indp.	RWM	RWM Sub.	pCN
R=10	1.02	0.99	0.96	0.91	1.17	0.17	0.76	0.80	0.75
C=10	1	1	1	196	334	1604	1354	1329	1504
R=20	0.97	0.98	1.00	0.91	1.00	0.17	0.48	0.45	0.37
C=20	1	1	1	61	75	1218	1649	1614	1827
R=50	1.00	1.01	0.98	0.96	0.99	0.17	0	0.01	0
C=50	1	1	1	10	12	5000+	1107	1616	1466
R=100	1.00	1.00	1.00	0.98	1.00	0.16	0	0.38	0
C=100	1	1	1	3	3	5000+	1199	1714	1532
R=200	1.00	1.00	1.00	1.01	1.01	0.21	0	0.66	0
C=200	1	1	1	1	1	1266	1626	1691	636
R=500	1.00	1.00	1.00	118.45	52.70	0.14	0	0.87	0
C=500	1	1	1	545	138	1572	834	1702	1616
R=1000	1.00	1.00	1.00	65.22	2.66	0.15	0	0.93	0
C=1000	1	1	1	385	3062	5000+	1518	1724	621
R=2000	1.00	1.00	1.00	115.59	1.05	0.18	0	0.97	0
C=2000	1	1	1	10	5000+	1021	1194	1702	1014
R=50	1.01	0.99	1.00	0.98	1.01	0.15	0	0.19	0
C=100	1	1	1	5	6	5000+	1676	1774	1442
R=10	0.99	0.99	1.01	0.92	0.99	0.17	0	0.55	0
C=200	1	1	1	12	15	3309	1570	1678	1279
R=100	1.00	1.00	1.00	3.50	3.46	0.19	0	0.87	0
C=1000	1	1	1	3	3	3330	1454	1699	3333

Table 6: Median estimates of σ_E^2 and lag when $\text{ACF}(\hat{\sigma_E^2}) \leq 0.5$