Scalable Algorithms for Learning High-Dimensional Linear Mixed Models

Zilong Tan  
Duke University  
ztan@cs.duke.edu

Kimberly Roche  
Duke University  
kimberly.roche@duke.edu

Xiang Zhou  
University of Michigan  
xzhousph@umich.edu

Sayan Mukherjee  
Duke University  
sayan@stat.duke.edu

Abstract

Linear mixed models (LMMs) are used extensively to model observations that are not independent. Parameter estimation for LMMs can be computationally prohibitive on big data. State-of-the-art learning algorithms require computational complexity which depends at least linearly on the dimension $p$ of the covariates, and often use heuristics that do not offer theoretical guarantees. We present scalable algorithms for learning high-dimensional LMMs with sublinear computational complexity dependence on $p$. Key to our approach are novel dual estimators which use only kernel functions of the data, and fast computational techniques based on the subsampled randomized Hadamard transform. We provide theoretical guarantees for our learning algorithms, demonstrating the robustness of parameter estimation. Finally, we complement the theory with experiments on large synthetic and real data.

1 INTRODUCTION

Linear mixed models (LMMs) are widely used in many real world applications ranging from longitudinal data analysis (Laird and Ware, 1982; Demidenko, 2013) and genome wide association studies (Kang et al., 2008; Lippert et al., 2011; Zhou, 2017) to recommender systems (Zhang et al., 2016). LMMs provide a flexible framework for modeling a wide range of data types, including clustered, longitudinal, and spatial data. Parameter estimation for LMMs is computationally prohibitive for big data, both for large sample size $n$ (Zhou and Stephens, 2014; Darnell et al., 2017; Perry, 2017) and for high-dimensional covariates $p$ (Schelldorfer et al., 2011). The main computational bottlenecks for parameter estimation arise from the non-convexity of the optimization problem (Kang et al., 2008; Perry, 2017) as well as the computational cost of matrix inversions (Zhou, 2017; Laird et al., 1987; Lindstrom and Bates, 1988; Bates et al., 2015). State-of-the-art methods for parameter estimation in LMMs require computational complexity that depends at least linearly on $p$: (i) $O(nkp)$ for the setting $n > p$ with a rank $k$ covariance matrix (Zhou, 2017; Darnell et al., 2017); and (ii) $O(n^2p)$ per iteration for $p \gg n$ (Schelldorfer et al., 2011, 2014; Jakubík, 2015).

In this paper, we present scalable algorithms with sublinear computational complexity in $p$, making the proposed approach useful for high-dimensional LMMs. In addition, we provide a theoretical analysis for our approach that states provable error guarantees between the estimated and ground-truth parameters.

Two sets of parameters are estimated in LMMs, the fixed-effects coefficients and the variances for the unobservable random effects and noise. The random-effects variance is generally assumed to have a certain structure, such as a block-diagonal matrix (Laird and Ware, 1982; Demidenko, 2013). To estimate both sets of parameters, an expectation maximization (EM) algorithm is typically used (Laird et al., 1987; Bates et al., 2015) to handle the latent random-effect variable. The M-step in the EM algorithm incurs high computational costs due to matrix inversions. Newton-Raphson has been used to reduce the number of iterations required for parameter estimates to converge (Lindstrom and Bates, 1988); however, each iteration is still costly due to matrix inversions. A recent research focus is to avoid matrix inversions at each iteration. For instance, when $n > p$ a spectral algorithm is available (Kang et al., 2008; Lippert et al., 2011; Patterson and Thompson, 1971). The state-of-the-art algorithm (Darnell et al., 2017) further improved the computational complexity of the spectral algorithm using randomized singular value decomposition (Darnell et al., 2017).

While approximate learning algorithms (Zhou, 2017;
Darnell et al., 2017) are efficient, few provide provable guarantees in terms of estimation accuracy. Recently, a guaranteed non-iterative algorithm was proposed in (Perry, 2017), which runs in $O \left( n (p + d)^4 \right)$ time for $d$ random effects. Inference with guarantees for high-dimensional LMMs, i.e., $p \gg n$, typically incurs greater computational complexity due to the regularization required to address high-dimensional data (Schell dorfer et al., 2011, 2014). In the high-dimensional setting, most algorithms perform block coordinate descent (Schell dorfer et al., 2011, 2014). In this paper, we show that efficiency and provable guarantees can be achieved simultaneously for learning high-dimensional LMMs.

There are two key ideas we use in our efficient algorithms. The first idea is to propose an approximate estimator that relies on an $n \times n$ kernel matrix (§ 3) which can be computed efficiently using the subsampled randomized Hadamard transform (SRHT) (Tropp, 2011). This reduces the linear complexity dependence on $p$. Unlike some other approximation algorithms (Lu et al., 2013), the proposed estimator also has the advantage of recovering the fixed-effects coefficients for all $p$ dimensions as opposed to the reduced dimensions. This allows us to provide effect sizes in terms of the original covariates, a requirement in many applications. The second idea is the introduction of approximate variance components (AVCs) to replace variance components when estimating the fixed-effects coefficients. These AVCs have a closed-form expression and are fast to compute.

We apply our novel approach to LMMs with a both general covariances as well as a block-diagonal covariances for the random effects. The former can be viewed as a special case of the latter with a single block, and has been adopted in genome-wide association studies (Kang et al., 2008; Lippert et al., 2011; Zhou, 2017). LMMs with a block-diagonal covariance structure have been widely used for modeling repeated measures data (Laird and Ware, 1982). We propose a non-iterative algorithm for the general covariance setting and a fast EM variant for the block-diagonal setting.

### Contribution

Our main contribution is providing a class of approximation algorithms for parameter inference in high-dimensional LMMs with provable guarantees. In Table 1, we state the computational complexity for several standard and state-of-the-art parameter inference algorithms. In the table and in this paper, $n$ is the sample size, $p$ is the number of covariates, $k$ is the rank of the covariance matrix, $s$ are the number of subsamples, and $\epsilon$ is the approximation error. Our method is the only one that is sublinear in $p$, and can be a $n / \log p$ magnitude faster than the others (discussed in § 4.1). In addition to theoretical advantages, we demonstrate the empirical accuracy and speed of our method on both synthetic and real data in § 6.

### Notation

We denote the maximum and minimum eigenvalues of a matrix $A$ by $\lambda_{\text{max}}(A)$ and $\lambda_{\text{min}}(A)$, respectively. Similarly, we denote the maximum and minimum singular values respectively by $\sigma_{\text{max}}(A)$ and $\sigma_{\text{min}}(A)$. $A^†$ represents the Moore–Penrose pseudoinverse of $A$, and $\kappa(A)$ denotes the condition number of $A$. The superscripted notation $y^{(i)}$ refers to the copy of $y$ for group $i$. We write the spectral norm of a matrix as $\| \cdot \|_2$, the Frobenius norm as $\| \cdot \|_F$, and the Ky Fan $k$-norm $\| \cdot \|_k$ (the sum of the $k$ largest singular values) as $\| \cdot \|_k$.

### Organization

Section 2 provides the background on standard LMMs. In section 3, we formulate the $L_2$-regularized LMMs and present approximate estimators based on a kernel matrix. Section 4 describes fast computational techniques for the approximate estimators. In section 5, we provide theoretical guarantees for our estimators. Section 6 reports empirical evidence of the speed and accuracy of our methods, and section 7 concludes this paper.

### Table 1: Computational complexity for parameter inference

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>REML (LIPPERT ET AL., 2011)</td>
<td>$O(n^2p)$</td>
</tr>
<tr>
<td>Moments (PERRY, 2017)</td>
<td>$O(n(p+q)^3)$</td>
</tr>
<tr>
<td>Subsampling (ZHOU, 2017)</td>
<td>$O(ps^2)$</td>
</tr>
<tr>
<td>RSVD (DARNELL ET AL., 2017)</td>
<td>$O(pmk)$</td>
</tr>
<tr>
<td>This work</td>
<td>$O(n^2(k+\log p)\log k)$</td>
</tr>
</tbody>
</table>

Consider a regression problem with $n$ observations, where $y \in \mathbb{R}^n$ denotes the response vector and $X \in \mathbb{R}^{n \times p}$ represents the covariance matrix with $p$ covariates. The standard LMM is given by

$$ y = X \beta + Z \gamma + c1 + e \quad \text{with}$$

$$ \begin{bmatrix} \gamma \\ e \end{bmatrix} \sim \text{MVN} \left( 0, \begin{bmatrix} \Lambda & 0 \\ 0 & \sigma^2 I \end{bmatrix} \right),$$

where $\beta \in \mathbb{R}^p$ is the fixed-effect coefficient vector, $Z \in \mathbb{R}^{n \times q}$ is a full-rank random-effects design matrix, $\gamma \in \mathbb{R}^q$ is the random-effect coefficient vector, $c$ is the intercept, and $e \in \mathbb{R}^n$ is the noise vector. The parameters to be estimated are the fixed-effects coefficients $\beta$, and variance components $\Lambda$ and $\sigma^2$.
In general, the variables $X, y, \gamma,$ and $e$ in (1) correspond to observations from $m$ classes, and are grouped by the following structure (Laird and Ware, 1982):

$$
\begin{bmatrix}
X^{(1)} \\
X^{(2)} \\
\vdots \\
X^{(m)}
\end{bmatrix}, \quad
\begin{bmatrix}
y^{(1)} \\
y^{(2)} \\
\vdots \\
y^{(m)}
\end{bmatrix}, \quad
\begin{bmatrix}
\gamma^{(1)} \\
\gamma^{(2)} \\
\vdots \\
\gamma^{(m)}
\end{bmatrix}, \quad
\begin{bmatrix}
e^{(1)} \\
e^{(2)} \\
\vdots \\
e^{(m)}
\end{bmatrix},
$$

where $^{(i)}$ denote the variables specific to group $i$, whose dimensions are $X^{(i)} \in \mathbb{R}^{n_i \times p}$, $\gamma^{(i)} \in \mathbb{R}^d$, and $y^{(i)}, e^{(i)} \in \mathbb{R}^{n_i}$. The LMM assumes that $\gamma^{(i)}$ corresponding to distinct classes are independent. In particular, the random-effects design matrix $Z$ and the random-effects covariance are block-diagonal

$$
Z = \begin{bmatrix}
Z^{(1)} & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & Z^{(m)} & 0 \\
0 & \cdots & 0 & H
\end{bmatrix}, \quad
\Lambda = \begin{bmatrix}
H & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & H & 0 \\
0 & \cdots & 0 & H
\end{bmatrix}
$$

with $Z^{(i)} \in \mathbb{R}^{n_i \times d}$, $H \in \mathbb{R}^{d \times d}$, and $q = md$.

**Computational challenges** Parameter inference in LMMs aims to accurately recover $\mathcal{P} := \{\beta, \Lambda, \sigma^2\}$ from $\{X, y, Z\}$. This is straightforward if $\Lambda$ is given. When $\Lambda$ is unknown, inference can be computationally challenging even in the standard setting where $n > p$ (Laird and Ware, 1982; Lippert et al., 2011; Zhou, 2017; Laird et al., 1987; Lindstrom and Bates, 1988; Patterson and Thompson, 1971; Zhang et al., 2011).

First, parameter estimation problem is non-convex for both maximum likelihood and restricted maximum likelihood (REML) (Laird et al., 1987; Patterson and Thompson, 1971; Harville, 1974). For instance, the methods using REML (Kang et al., 2008; Lippert et al., 2011) project the data onto two uncorrelated parts, and then estimate the fixed-effects and variance components separately on each part. This has the advantage of giving unbiased estimates of the variance components. However, the REML likelihood function is a non-convex function which involves the eigenvalues of the variance of the projected data (Patterson and Thompson, 1971).

Second, regularization is typically required to support the high-dimensional setting, which adds further computational overheads (Lippert et al., 2011; Zhou, 2017; Schelldorfer et al., 2011, 2014; Jakubik, 2015). To address these challenges, we develop novel approximate estimators that are efficient to compute ($\S$ 4), and have provable accuracy guarantees ($\S$ 5).

### 3 APPROXIMATE ESTIMATORS FOR HIGH-DIMENSIONAL LMMS

In this section, we consider an $L_2$-regularized LMM to support the high-dimensional setting $p > n$, and develop efficient approximate estimators for the parameters.

Standard parameter estimation algorithms for LMMs such as (Kang et al., 2008; Laird et al., 1987; Bates et al., 2015) do not support the high-dimensional setting $p > n$. We consider introducing the $L_2$ regularization on the fixed-effects coefficients, which can be viewed as adding the prior $\beta \sim \mathcal{N}(0, \Phi)$. The $L_2$-regularized LMM has the following log-likelihood

$$
\log p(y | X; V) \propto -\frac{1}{2} \beta^T \Phi \beta - \frac{1}{2} \log \det V
$$

with the marginal variance $V := Z \Lambda Z^T + \sigma^2 I$.

Parameter estimation of an LMM is typically iterative and computationally prohibitive, especially in the high-dimensional setting (Darnell et al., 2017; Perry, 2017; Schelldorfer et al., 2011). To improve the computational efficiency, we propose dual as well as approximate estimators. These estimators are non-iterative and have reduced computational complexity, as we will show in $\S$ 4.

#### 3.1 FIXED-EFFECT COEFFICIENTS

We first derive the estimators for the fixed-effects coefficients $\hat{\beta}$ and $\hat{\gamma}$, which are the maximizers of the log-likelihood (2). A dual estimator of $\beta$ is then given for use in the high-dimensional setting. Using the partial derivatives, it is straightforward to show

$$
(X^T V^{-1} X + \Phi^{-1}) \hat{\beta} = X^T V^{-1} (y - c1)
$$

$$
\hat{\gamma} = \frac{1^T V^{-1} y - 1^T V^{-1} X \hat{\beta}}{1^T V^{-1} 1}.
$$

Let $L = I - 11^T V^{-1} (1^T V^{-1} 1)^{-1}$, we obtain

$$
\hat{\beta} = (X^T V^{-1} LX + \Phi^{-1})^{-1} X^T V^{-1} Ly.
$$

The dual estimator using $X \Phi X^T$ was proposed in (Saunders et al., 1998) where the authors used Lagrange multipliers to obtain the following estimator for ridge regression:

$$
\hat{\beta}_{\text{Dual}} = \Phi X^T (V + X \Phi X^T)^{-1} y.
$$

Here, $\Phi$ is set to be diagonal, and the above estimator (6) can be evaluated in $O (n^2 p)$ time, a significant improvement when $p \gg n$. However, the computational bottleneck becomes evaluating the kernel matrix $X \Phi X^T$. 

For the zero intercept case $\hat{c} = 0$, the dual estimator (6) is equivalent to (5) from the following variant of the Woodbury identity $(U^{-1} + A^T V^{-1} A)^{-1} = U A^T (A U A^T + V)^{-1}$. The dual estimator can be generalized to any intercept,

$$
\hat{\beta} = \Phi X^T V^{-1} L (X \Phi X^T V^{-1} L + I)^{-1} y. \tag{6}
$$

Computing the dual estimator (6) takes $O(n^2 p)$ time as opposed to $O(p^3)$ time required by (5). This complexity will be further improved in §4 for the setting $p \gg n$.

### 3.2 APPROXIMATE VARIANCE COMPONENTS

The variance components $\Lambda$ and $\sigma^2$ are typically estimated using an iterative EM algorithm with a perturbation cost $O(p^3)$ (Laird et al., 1987; Lindstrom and Bates, 1988) or an exhaustive grid search for the solution of a system of eigenvalue equations (Kang et al., 2008; Lippert et al., 2011). We consider an approximate non-iterative estimator based on the key observation that the optimization of the (2) has a simple closed-form solution if carried out with respect to $M = V + X\Phi X^T$. We will estimate $M$ and use it as a proxy for estimating $\Lambda$ as well as $\sigma^2$. The variance components inferred using $M$ are referred to as the approximate variance components (AVCs). While AVCs may be used as variance components estimates under certain circumstances, the main purpose is to serve as fast replacements in estimating fixed-effects coefficients.

**Proxy component estimation** To perform the REML estimation of the variance components in terms of $M$, we first rewrite the log-likelihood (2) as

$$
l(\beta, V) = -\frac{1}{2} \log \det V - \frac{1}{2} (y - \hat{c} 1)^T M^{-1} (y - \hat{c} 1) - \frac{1}{2} \beta^T Q (\beta - \hat{\beta}(V)) \tag{7}
$$

where $Q = X^T V^{-1} X + \Phi^{-1}$ and $\hat{\beta}(V) = (X^T V^{-1} X + \Phi^{-1})^{-1} X^T V^{-1} (y - \hat{c} 1)$. Here, the estimate $\hat{\beta}$ depends on $V$, and is consistent with the estimate given by (5). The $\hat{c}$ in (7) can be set to the mean response, or estimated based on a prior distribution as in (Zhou et al., 2013).

Then, the REML estimator for the variance components is based on marginalizing the fixed effects $\beta$ (Harville, 1974). It follows that

$$
l_p(V) \propto \log \int_{\mathbb{R}^p} \exp(l(\beta, V)) \, d\beta \\
\propto -\frac{1}{2} \log \det V - \frac{1}{2} \log \det Q - \frac{1}{2} (y - \hat{c} 1)^T M^{-1} (y - \hat{c} 1). \tag{8}
$$

From Sylvester’s determinant theorem, one observes that $\det(M) = \det(\Phi) \det(V) \det(Q)$. Thus, we arrive at

$$
l_p(V) \propto -\frac{1}{2} \log \det M - \frac{1}{2} (y - \hat{c} 1)^T M^{-1} (y - \hat{c} 1). \tag{8}
$$

Now, what we have achieved through (8) is a simple closed-form REML estimate of $V$, rather than the non-convex or iterative updates for $\Lambda$ and $\sigma^2$ in state-of-the-art LMM parameter estimation algorithms. Unconstrained maximization of (8) with respect to $M$ results in the closed-form equality

$$
Z \lambda Z^T + \sigma^2 I = (y - \hat{c} 1) (y - \hat{c} 1)^T - X \Phi X^T, \tag{9}
$$

for an optimal $M$. Note that $Z \lambda Z^T$ is positive semidefinite, whereas the right hand side has at most one positive eigenvalue. Thus, this optimal $M$ may not be achievable and the unbiased estimate of $\Lambda$ may possibly have negative eigenvalues. The issue of negative variance estimates in linear mixed models is an open problem (Demenkno, 2013) and beyond the scope of this paper. One resolution is to introduce a Gamma prior on $\Lambda$ (Chung et al., 2013). For unbiased estimation, we allow $\Lambda$ to have negative eigenvalues, and intuitively we refer to the variance estimators obtained this way as approximate variance components.

**Approximate variance estimators** Assume that $Z$ has full column rank and let $S = (y - \hat{c} 1) (y - \hat{c} 1)^T - X \Phi X^T$. The approximate variance components $\hat{\Lambda}_{\text{AVC}}$ and $\hat{\sigma^2}_{\text{AVC}}$ can be obtained via

$$
\arg \min_{\Lambda, \sigma^2} \|Z \Lambda Z^T - S + \sigma^2 I\|_F^2. \tag{10}
$$

Optimizing with respect to $\Lambda$ yields

$$
\Lambda_* = Z^\dagger (S - \sigma^2 I) Z^\dagger, \tag{11}
$$

where $Z^\dagger = (Z^T Z)^{-1} Z^T$. The estimators are computed by substituting $\Lambda_*$ into (10) and optimizing with respect to $\sigma^2$:

$$
\hat{\sigma^2}_{\text{AVC}} = \frac{\text{tr} [S (I - ZZ^\dagger)]}{n - q} \tag{12}
$$

$$
\hat{\Lambda}_{\text{AVC}} = Z^\dagger S Z^\dagger - \hat{\sigma^2}_{\text{AVC}} (Z^T Z)^{-1}.
$$
Consider the parameterization $\Lambda = \theta D$ in (Kang et al., 2008; Lippert et al., 2011) with a fixed symmetric positive semi-definite $D$, the solution to (10) is written as

$$\Lambda_{\kappa} = \frac{\text{tr} (G (S - \sigma^2 I))}{\text{tr} (G^2)} D$$

with $G = ZDZ^\top$. Substituting into (10), we obtain

$$\hat{\sigma}_2^2_{\text{AVC}} = \frac{1}{n - \alpha} \left[ \text{tr} (S) - \frac{\text{tr} (GS)}{\text{tr} (G^2)} \right],$$

where $\alpha = \text{tr} (G^2) / \text{tr} (G^2)$. Combined with (13), we arrive at

$$\hat{\Lambda}_{\text{AVC}} = \frac{\text{tr} (G (S - \hat{\sigma}_2^2_{\text{AVC}} I))}{\text{tr} (G^2)} D.$$ (15)

While AVCs may be used as variance components estimates under certain circumstances, the main purpose is to speed up estimating the fixed-effect coefficients. The complexity for computing the AVCs is $O(n^3)$, if $S$ is given. Like the dual fixed-effects estimator (6), the computational bottleneck of AVCs also lies in evaluating $X\Phi X^\top$.

4 FAST COMPUTATIONAL ALGORITHMS

In this section, we further improve the computational complexity $O\left( n^2 p \right)$ of the proposed approximate estimators in the high-dimensional setting $p \gg n$, where the computation bottleneck lies in evaluating the kernel $X\Phi X^\top$. We adopt the subsampled randomized Hadamard transform (SRHT) (Tropp, 2011) to compute the kernel matrix efficiently. In particular, the high-dimensional data is first projected into lower dimensions using SRHT, and the parameters of the LMM are then estimated using the projected data. However, there are two main challenges involved: 1) the estimated parameter $\hat{\beta}$ now corresponds to the projected data of reduced dimensions, whereas the coefficients of the full original covariates are desired; and 2) the impact of applying the SRHT on the accuracy of parameter estimation needs to be justified. The techniques developed in this section recovers the coefficients to the full covariates from the SRHT projected data with high accuracy, as will be shown in § 5.

4.1 NON-ITERATIVE ALGORITHM FOR GENERAL LMMS

In this subsection, we provide a fast algorithm for parameter estimation in case of a general covariance matrix. Algorithm 1 takes as input the matrices $X$ and $\Phi$

![Algorithm 1](image)

which will be typically diagonal) and an approximation error $\epsilon$ described in § 5. Both an approximation to the kernel matrix $X\Phi X^\top$ and the SRHT matrix $\Pi$ are computed. The computational efficiency of the algorithm is a result of replacing $X$ with the smaller transform $A$ in subsequent operations. Additionally, the structure of the SRHT allows for a divide-and-conquer scheme to compute $A = X\sqrt{\Phi}\Pi^\top$ in $O\left( np \log p \right)$ time. Note that the matrix $W_p$ is not formed explicitly. The computation $AA^\top$ requires $O\left( n^2 \log n \right)$ time, which becomes dominant setting $\epsilon \leq Cn \sqrt{\frac{\log n}{p \log p}}$ for some universal constant $C$. Thus, the overall runtime for the algorithm is $O\left( \frac{n^3 \log n}{\epsilon^2} \right)$ for dense full-rank $X$, and will be faster if $X$ is of low rank. The quality of the approximation depends on $\epsilon$, which will be discussed in § 5.

Given the approximate kernel matrix, it is straightforward to compute the AVCs $\hat{\Lambda}_{\text{AVC}}$ and $\hat{\sigma}_2^2_{\text{AVC}}$ via (12). The coefficients for the fixed-effects can also be computed efficiently using the following estimator

$$\hat{\beta} = \sqrt{\Phi}\Pi^\top A^\top \hat{V}^{-1} L \left( I + AA^\top \hat{V}^{-1} L \right)^{-1} y.$$ (16)

Given the approximate kernel matrix and $A$, computing $A^\top \hat{V}^{-1} L \left( I + AA^\top \hat{V}^{-1} L \right)^{-1} y$ takes time $O \left( \max \left\{ n^2 s, n^3 \right\} \right)$ and multiplication of this vector by $\sqrt{\Phi}\Pi^\top$ is $O \left( p \log p \right)$ due to the structure of the SRHT matrix as well as the fact that $\sqrt{\Phi}$ is diagonal. The resulting complexity in computing (16) is $O \left( \max \left\{ n^2 s, n^3, p \log p \right\} \right)$.
Approximating the kernel matrix using the SRHT was proposed for ridge regression in (Lu et al., 2013), a special case of our setting. A method for estimating the full set of fixed-effects coefficients was not provided in (Lu et al., 2013). Instead, a reduced set of \( m \) fixed-effects coefficients corresponding to the transformed covariate matrix \( XL^\top \) was reported. For many applications, a major point of using an LMM is to estimate the effect-size of the fixed-effect coefficients, so computing \( \hat{\beta} \) is essential to the problem.

### 4.2 FAST EM FOR MULTI-GROUP LMMS

For efficient parameter estimation in \( L_2 \)-regularized LMMs with repeated measurements, we extend the EM algorithm for the low-dimensional setting \( n \geq p \) (Laird et al., 1987) by combining the kernel estimators and Algorithm 1. While this high-dimensional EM variant is iterative, we show that the per-iteration computational cost is scalable in \( p \).

The log-likelihood of the \( L_2 \)-regularized LMM (17) can be rewritten in terms of class-specific variables as

\[
\log p\left( y, \gamma, \beta \mid X; \sigma^2, \Lambda \right) \\
\propto -\frac{1}{2} \hat{\beta}^\top \Phi^{-1} \beta - \frac{n}{2} \log \sigma^2 - \frac{m}{2} \log \det H \\
- \frac{1}{2} \sum_{i=1}^m \gamma(i)^\top H^{-1} \gamma(i) - \frac{e^\top e}{2\sigma^2},
\]

where \( e = y - c1 - X\beta - Z\gamma \).

From the above log-likelihood, the posterior distribution of \( \beta \) conditioned on the data and parameter estimates \( \hat{\beta} := \left\{ \hat{c}, \hat{\sigma}^2, \hat{H} \right\} \) is multivariate normal with mean \( \Phi X^\top \hat{M}^{-1} (y - c1) \) and covariance \( \Phi - \Phi X^\top \hat{M}^{-1} X \Phi \). Similarly, the posterior distribution of the vector of latent variables \( \gamma \) is multivariate normal with mean \( \hat{\Lambda} Z^\top \hat{M}^{-1} (y - c1) \) and covariance \( \hat{\Lambda} - \hat{\Lambda} Z^\top \hat{M}^{-1} Z \hat{\Lambda} \). Denote by \( \hat{\gamma}_i \) the mean of the posterior distribution of \( \gamma_i \), we also obtain the following posterior distributions of class-specific latent variable \( \gamma(i) \):

\[
\mathcal{N}\left( \hat{\gamma}(i), \hat{H} - \hat{H} Z(i)^\top \left( \hat{M}^{-1} \right)^{(i)} Z(i)^\top \hat{H} \right). \quad (18)
\]

Note that \( \cdot(i) \) represents the block matrix corresponding to group \( i \). These posteriors are used in the E-step, discussed next.

**E-step** In the E-step, we derive the expectation of the log-likelihood (17) with respect to the aforementioned posterior distribution of \( \beta \) and \( \gamma(i) \):

\[
\mathbb{E}_{\beta, \gamma \mid y, \hat{\beta}} \left[ \log p\left( y, \gamma, \beta \mid X; \sigma^2, H \right) \right],
\]

We only need to consider terms in the expectation that involve \( c, \sigma^2 \), and \( H \). Denote by \( \hat{\Sigma}_{\gamma(i)} \) the variance of \( \hat{\gamma}(i) \) as follows holds

\[
\hat{\Sigma}_{\gamma(i)} = \mathbb{E}_{\beta, \gamma \mid y, \hat{\beta}} \left( \gamma(i)^\top H^{-1} \gamma(i) \right) = \mathbb{E}_{\beta, \gamma \mid y, \hat{\beta}} \left( \gamma(i)^\top H^{-1} \gamma(i) \right) + \text{tr} \left( \hat{\Sigma}_{\gamma(i)} H^{-1} \right). \]

Using the previously derived posterior distributions, we get

\[
\mathbb{E}\left( e \mid y, \hat{\beta} \right) = y - X\hat{\beta} - Z\hat{\gamma} - c1 \\
\text{cov}\left( e \mid y, \hat{\beta} \right) = \hat{\sigma}^2 I - \hat{\sigma}^4 \hat{M}^{-1}.
\]

Thus, we arrive at

\[
\mathbb{E}_{\beta, \gamma \mid y, \hat{\beta}} \left( e^\top e \right) = e^\top e + \hat{\sigma}^2 I - \hat{\sigma}^4 \hat{M}^{-1},
\]

\[
h \text{where } \hat{e} := \mathbb{E}\left( e \mid y, \hat{\beta} \right).
\]

**M-step** We now update the parameter estimates with the maximizers of the expectation from the E-step. First, observe that the \( \beta \) estimate from the posterior distribution is the same as the the dual estimator developed in § 3. To maximize the expectation with respect to \( H \) and \( \sigma^2 \), we take the partial derivatives with respect to \( H^{-1} \) and \( \sigma^{-2} \), and set them to zero. This gives the following M-step updates:

\[
\hat{H} \leftarrow \frac{1}{m} \sum_{i=1}^m \left( \hat{\gamma}(i) \hat{\gamma}(i)^\top + \hat{\Sigma}_{\gamma(i)} \right) \\
\hat{\sigma}^2 \leftarrow \hat{\sigma}^2 + \frac{1}{m} \left[ \hat{e}^\top \hat{e} - \hat{\sigma}^2 \text{tr} \left( \hat{M}^{-1} \right) \right].
\]

The fast version of the above EM variant uses Algorithm 1 for computing the kernel. Note that the original \( X \) is no longer needed after the SRHT projection. This provides additional space advantages as data \( X \) can be preprocessed, and the Hadamard transform in Step 1 requires a small constant amount of memory. Overall, the per-iteration computational complexity of the EM algorithm is \( O\left( \max \{ n^2 s_e, n^3 \} \right) \).

## 5 THEORETICAL GUARANTEES

In this section, we provide an analysis of the difference in the parameters estimated via the approximate algorithms versus minimizing the \( L_2 \)-regularized LMM. We are not proving consistency of our estimator—convergence of the parameter estimates to the population quantity. Consistency results for LMMs and-regularized LMMs were provided in (Schelldorfer et al., 2011; Cui et al., 2004; Hall and Yao, 2003). See the supplementary materials for proofs of the theorems in this section.

**Theorem 1** (Fixed-effect norm error). Let \( \hat{\beta} \) be the fixed-effect coefficients estimated by (5) and \( \hat{\beta}' \) be the fixed-effect coefficients estimated by the approximate proce-
The main finding of the experiments is that the proposed multigroup variant of arLMM-EM and (12), the proposed multi-group variant of arLMM-AVC based on (16) and (12), the proposed multi-group variant of arLMM-EM based on (19), the standard REML (Bates et al., 2015), L1-regularized LMM lmmlasso (Schellendorf et al., 2011), and CovexLasso using both L1- and L2-regularization (Jakubík, 2015).

### Synthetic data generation

The simulation is based on synthetic training and validation sets sampled from a fixed LMM distribution. The design matrices as well as the parameters for the fixed LMM are randomly generated. Specifically,

\[
X_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1) \quad Z_{ij}^{(k)} \overset{i.i.d.}{\sim} \mathcal{U}(0, 1) \quad \gamma^{(k)} \overset{i.i.d.}{\sim} \mathcal{N}(0, K^T K) \quad K_{ij} \overset{i.i.d.}{\sim} \mathcal{N}(0, 1) \quad \beta \sim \mathcal{N}(0, I) \quad \sigma^2 \sim \mathcal{U}(0, d). 
\]

Note that there are \(d\) random-effect variables with covariance \(K^T K\). Thus, the random-effect design matrix \(Z \in \mathbb{R}^{n \times q}\), \(q = md\), will be block-diagonal with diagonal blocks \(Z^{(k)}\). Given the number of observations \(n\), we randomly sample \(n_k\) observations for each group \(k\), where the fractions \(n_k/n\) are specified by the Dirichlet distribution with the concentration parameters \((1, 1, \ldots)^T\).

#### Overdetermined settings

Let us first consider the standard setting \(n > p\), which are supported by many parameter estimation algorithms of LMMs. We evaluate the performance of arLMM-AVC and arLMM-EM in a variety of \(p, d, \) and \(m\) settings. The parameter estimates obtained using the proposed methods are compared with the estimates given by the standard REML (see e.g., (Kang et al., 2008; Lippert et al., 2011; Laird et al., 1987; Bates et al., 2015)) which is known to produce unbiased estimates.
Figure 1: Comparing the performance of parameter estimation on synthetic data with \( n = 1,000 \) observations. Note that \( \text{arLMM-AVC} \) is only applicable to the single group setting. This figure shows that the \( \text{arLMM-EM} \) and \( \text{arLMM-AVC} \) achieve comparable estimation performance as \( \text{REML} \).

Figure 1 shows the error for the fitted parameters using 1,000 observations sampled from the underlying LMM. The average results are reported over 10 runs on independently generated datasets. These generated datasets have the same number of observations \( n = 1,000 \) but different settings of \( p, d, \) and \( m \).

As shown in Figure 1, \( \text{arLMM-EM} \) and \( \text{arLMM-AVC} \) exhibit comparable estimation accuracy as the standard \( \text{REML} \). Note that \( \text{arLMM-AVC} \) is applicable only when \( m = 1 \) (the first row of Figure 1). Since \( \text{arLMM-AVC} \) is based on non-iterative approximation to the variance components, the error is slighted higher than the others as expected.

High-dimensional (underdetermined) setting We also examined the performance of our model in the high-dimensional setting where we are interested in variable selection based on the fixed-effects coefficients. In Table 2, we specify the three regimes for which we generate simulated data: an overdetermined LMM, a moderate-dimensional LMM, and a high-dimensional LMM. Each regime is characterized by \( n, p, d, \) and \( m, \) and an extra parameter \( s, \) the number of non-zeros in the ground-truth \( \beta_{\text{True}}. \) Since \( m > 1 \) we did not apply \( \text{arLMM-AVC}. \)

Table 2: Regimes of data.

<table>
<thead>
<tr>
<th></th>
<th>((n, p, d, m, s))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>((100, 1000, 5, 3, 10))</td>
</tr>
<tr>
<td>Mod</td>
<td>((200, 10^4, 5, 3, 10))</td>
</tr>
<tr>
<td>High</td>
<td>((10^8, 10^8, 10, 100, 100))</td>
</tr>
</tbody>
</table>

Figure 2 reports variable selection results for \( \text{arLMM-EM}, \text{lmmlasso} \) (Schelldorfer et al., 2011), and \( \text{ConvexLasso}. \) All the settings in Table 2 have sparse ground-truth \( \beta_{\text{True}}. \) Figure 2 shows the fraction of the signal (non-zeros in \( \beta_{\text{True}} \)) recovered in the estimate \( \hat{\beta}. \) We varied the regularization parameters to obtain \( \hat{\beta} \) with different sparsity \( \| \hat{\beta} \|_0. \) The entries with the largest magnitude of \( \hat{\beta} \) is considered the signal in these evaluations. As can be seen, \( \text{arLMM-EM} \) delivers a competitive signal recovery ratio for \( p = 10^3, 10^4, \) and scales to considerably large dimensions \( n = 10^4 \) and \( p = 10^6, \) which the other two methods cannot handle.

6.2 GENOME WIDE ASSOCIATION STUDIES

LMMs have been used extensively for mapping traits in statistical genetics. The problem formulation is that of regressing a quantitative or categorical trait onto a high-dimensional vector of 450,000 single nucleotide polymorphisms (SNPs), or locations of discrete genetic variation, for each subject included in the study. The random effects are driven by population structure or the pairwise similarity or relatedness between individuals.

We compare our approximate estimator to the performance of a state-of-the-art estimator called \( \text{BSLMM} \) (Bayesian sparse linear mixed model) (Zhou et al., 2013). Specifically, we run \( \text{BSLMM} \) in its ridge-regression with mixed models setting, the fastest setting of the package for a fair comparison. In this setting \( \text{BSLMM} \) is computing the maximum a posteriori estimate of the regularized LMM. We compare performance on the Wellcome Trust Case Control Consortium (WTCCC) dataset of 14,000 cases of 7 diseases - bipolar disorder (BD), coronary artery disease (CAD), Crohn’s disease (CD), hypertension (HT), rheumatoid arthritis (RA), type 1 diabetes (T1D), and type 2 diabetes (T2D) - and 3,000 shared controls. This dataset characterizes over 450,000 single nu-
Table 2: Comparing the prediction performance as well as the runtime of BSLMM and arLMM-AVC on the WTCCC dataset. Corr ($\hat{\beta}_{\text{BSLMM}}, \hat{\beta}_{\text{arLMM-AVC}}$) denotes the correlation between the fixed-effect coefficient estimates given by BSLMM and arLMM-AVC.

<table>
<thead>
<tr>
<th>DISEASE</th>
<th>TIME (MIN)</th>
<th>AUC</th>
<th>CORR ($\hat{\beta}<em>{\text{BSLMM}}, \hat{\beta}</em>{\text{arLMM-AVC}}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD</td>
<td>115.8</td>
<td>0.6520</td>
<td>0.9898</td>
</tr>
<tr>
<td>CAD</td>
<td>161.0</td>
<td>0.5999</td>
<td>0.9776</td>
</tr>
<tr>
<td>CD</td>
<td>110.3</td>
<td>0.6260</td>
<td>0.9862</td>
</tr>
<tr>
<td>HT</td>
<td>120.6</td>
<td>0.5956</td>
<td>0.9766</td>
</tr>
<tr>
<td>RA</td>
<td>147.4</td>
<td>0.6173</td>
<td>0.9834</td>
</tr>
<tr>
<td>T1D</td>
<td>120.0</td>
<td>0.6846</td>
<td>0.9939</td>
</tr>
<tr>
<td>T2D</td>
<td>155.3</td>
<td>0.6003</td>
<td>0.9783</td>
</tr>
</tbody>
</table>

7 CONCLUSIONS

State-of-the-art parameter inference in LMMs requires computational complexity which depends at least linearly on the number of covariates $p$ and generally relies on heuristics. In this paper, we presented scalable learning algorithms which have sublinear computational complexity in $p$ and provide theoretical guarantees for the accuracy of parameter estimation. Our approach combines novel approximate estimators that use a kernel matrix of the observations and the subsampled randomized Hadamard transform. Experiments on synthetic and real data corroborate the theory.
References


