

Distributed Multilevel Modeling

David Afshartous and George Michailidis

Abstract

Multilevel modeling is a popular statistical technique for analyzing data in hierarchical format, and thus naturally fits within a distributed database framework. We consider the computational aspects of multilevel modeling across distributed databases. In addition, we consider these aspects under a generalization of the multilevel model where the distributed groups (or databases) are allowed to specify different models at both level-1 (individual) and level-2 (group). For a variety of scenarios, we develop the distributed computation algorithm for two-step Least Squares (LS) estimators and also for iterative MLE estimators of the parameters of interest; in particular, we determine the required data structure at each computing site, the necessary information (original data, cross-product matrices, coefficient vectors), and the order in which such information needs to be passed between sites. Finally, we discuss recursive updating, fault tolerance, and security issues.

Keywords: Multilevel models, distributed computing, estimation.

1 Introduction

In a variety of applied statistics problems, information or data of interest is distributed across multiple databases. There may be multiple reasons for this distribution, ranging from privacy and confidentiality, to organizational structure and/or regulatory requirements, to storage capabilities. An example of the former are patients' or students' records that are collected and kept at local databases, while an example of the latter are sales records of a supermarket chain that due to storage limitations cannot be integrated together. As mentioned by Karr et al. (2005), concerns about data confidentiality pose strong legal, regulatory or even physical barriers to literally integrating the databases. Indeed, a fundamental tradeoff arises between protecting the confidentiality of the subjects while at the same time trying to derive useful information from the data. Such issues are faced by many federal statistical agencies (e.g., Bureau of Labor Statistics, Census Bureau, National Center for Educational Statistics) and also within social science and health research (e.g., clinical trials and medical records). Nevertheless, many questions about the performance of the entire organization, or comparisons between units of an organization, can be posed that would require a 'pooling' of the data from these distributed sources. Consider as an example the case of educational data associated with school performance that is managed by state or private educational agencies. Research questions concerning student performance may be investigated separately at each database (how one student fares with respect to his fellow classmates), or may be investigated collectively on an integrated database formed from the individual databases (how a student fares with respect to students in other schools). A similar question can be posed for schools and in general for any unit of analysis for which measurements are contained in the database. Another aspect that involves an integrated analysis is the fact that most databases are dynamic in nature, with their records being updated at regular time intervals, and hence one would want to run different analyses as additional information is being accumulated.

Multilevel modeling is a statistical technique designed to facilitate inferences from hierarchical data; thus, this technique naturally fits into the distributed database framework if one considers the databases as "groups" of the multilevel model. Moreover, multilevel modeling is employed across a variety of applications, ranging from clinical trials to educational statistics to marketing studies (Bryk & Raudenbush, 2002). Indeed, given that the multilevel model may be cast as a mixed effects model, applications also include areas such as longitudinal growth studies and animal breeding (Rao, 1987; Robinson, 1991). Although we will primarily use the term multilevel model, we could just as well employ the term mixed effects model due to the fact that the multilevel model can be written as a mixed effects model.

Such an analysis often increases the stability of coefficient estimates and hence the validity of the associated inferences. If all the data can be transported to a single location, existing multilevel model estimation methods may be applied. In certain cases, however, this may not be feasible, either due to the size of the data or possibly due to privacy/confidentiality concerns. Thus, it is important that the researcher has available techniques that allow the application of multilevel modeling in a distributed database setting. If not, other options include: 1) perform separate analysis in the different groups, 2) ignore the structure and treat the data as one collective group, or 3) aggregate the data from the lower levels of the hierarchy and perform analysis on the aggregates. The potential pitfalls of these approaches, however, are well documented, e.g., large standard errors, independence assumption violations, and aggregation bias or the ecological fallacy, etc. See (Bryk & Raudenbush, 2002; Goldstein, 2002; de Leeuw & Kreft, 1986, 1995; Judge et al., 1988) for details.

In this paper, we focus on the on the computational aspects of distributed multilevel models (the underlying working hypothesis is a very large number of groups and/or a very large number of observations per group, coupled with limited computational resources available at each group's disposal). We consider this problem under a general formulation of the multilevel model that encompasses different models (including different subsets of variables) for level-1 and different models (including different subsets of variables) for level-2 and similarly for the error structures. Our main contribution is the development of the distributed computation algorithm for one-step and two-step Least Squares (LS) estimators and also for iterative MLE estimators of the parameters of interest under various model scenarios; in particular, we determine the required data structure at each computing site, the necessary information (original data, cross-product matrices, coefficient vectors), and the order in which such information needs to be passed between sites. As the results of the distributed algorithms would be the same as those for the non-distributed case, we do not illustrate the algorithms on an actual data set. Although our focus is mainly on developing algorithms for modeling without bringing data into a single repository, we also briefly consider and discuss the confidentiality and privacy aspects mentioned above.

The outline of the paper is as follows. In Section 2 we review the multilevel model and introduce notation to generalize the multilevel model to allow for different groups to specify their own level-1 and level-2 models. In Section 3 we develop the computational algorithms necessary for distributed estimation in multilevel models, both for non-iterative and iterative estimation methods. We emphasize that these algorithms provide the same final results as those obtained if the data was collected in a single repository; they are not approximations. In Section 4, we discuss recursive updating, fault tolerance, computational and security issues. Finally, we conclude with a brief

summary in Section 5.

2 Some Basics of Multilevel Modeling

In many scientific fields the data have a natural hierarchical structure, and therefore multilevel or mixed effects models are appropriate for capturing such structure. Specifically, one considers a level-1 or individual model for micro-units, and then a level-2 or group level model for macro-units. Multilevel models can handle more complicated hierarchical structures that involve several levels, but in most cases in practice one rarely goes beyond two levels and therefore we restrict attention to this setting as well. In an educational context, level-1 units could correspond to students and level-2 units to schools, while in a marketing context, customers and geographical regions, respectively. The level-1 model is a classical linear model for the micro-units, while the level-2 model relates the parameters of the level-1 model (i.e., the level-1 regression coefficients), to level-2 or group regressors. In essence, the level-1 within-group regression coefficients are modeled as conditionally exchangeable random variables, conditional upon the values of group-level variables. Various forms of this general framework exist, allowing for setting certain level-1 coefficients to be non-random and to allow the inclusion of different level-2 variables for the equations of different level-1 regression coefficients. There exists a large literature on multilevel and mixed effects models, spanning a variety of research disciplines. For more information on the history, motivation, and applied examples of multilevel and mixed effects models, see Raudenbush & Bryk, (2002) and Robinson, (1991).

Next, we formally introduce a 2-level model. In our formulation we emphasize the fact that different databases may contain measurements on different subsets of variables. For example, in an educational context some schools may offer students special education programs, or in a medical context some tests may not be available at some hospitals. Suppose we have J groups and n_j observations in the j th group. The level-1 equation for each group may be written as follows:

$$Y_j = X_j\beta_j + \epsilon_j. \tag{1}$$

Each X_j has dimensions $n_j \times p$, and contains the measurements on the p regressors for group j , and $\epsilon_j \sim N(0, \sigma^2\Psi_j)$, with Ψ_j usually taken as I_{n_j} , and β_j is a random p -vector containing the within-group regression coefficients for group j . Thus, we have the standard linear model for each group, except that the coefficients are now modeled as random variables. In order to generalize to the situation where the different groups may employ different choices of level-1 variables, we use

p_j instead of p above. For each of the random variables β_{js} ($j = 1, \dots, J$ and $s = 1, \dots, p_j$), we have a model of the form:

$$\beta_{js} = z'_{js} \gamma_s + r_{js} \quad (2)$$

where the vector z_{js} has q_s elements. This corresponds to the case where the level-2 model for each of the random level-1 coefficients is constant across groups. In matrix notation, we may consider β_s , a J -vector containing the level-1 regression coefficients for the s th micro-variable across the J groups:

$$\beta_s = Z_s \gamma_s + r_s, \quad (3)$$

where Z_s is a $J \times q_s$ matrix where each row contains the q_s level-2 covariates of the j th group for the s th level-1 regression coefficient. Since q_s is not necessarily constant for all s , we allow for the possibility that the regression coefficients for different level-1 variables are regressed on different sets of level-2 or group variables. Furthermore, to allow for the case where each group may choose different sets of level-2 variables to model the same level-1 coefficient, we replace q_s with q_{sj} . Since not every group necessarily chooses the s th micro-variable, we require that the corresponding entry in β_s be equal to zero to insure that β_s has J elements for all s . This will later simplify our data representations when performing distributed estimation.

In addition, we must make sure that Z_s and γ_s are altered accordingly. The full model is where all groups employ all level-2 variables for the given random slope variable s , i.e., for β_{sj} . If a total of q level-2 variables exist and all are used to model this level-1 coefficient, the matrix Z_s will be $J \times q$, with each row corresponding to all the level-2 information for a particular group. (Note that in the situation where no group uses all level-2 variables that exist, we may not set q to the maximum number of level-2 variables employed across the groups; we must still “expand” Z_s so that the number of columns is equal to the total number of variables, thereby allowing the column position to indicate variable identity.) If the s th level-1 variable is not used by a particular group, i.e., if we assume $\beta_{sj} = 0$ for a particular j , the j th row of Z_s is a row of zeros. Similarly, if a particular level-2 variable is not used by a group to model β_{sj} , then we require that the corresponding entry of Z_s be equal to zero. The Z_s matrix essentially contains information on how the J groups model the s th level-1 coefficient. The full model is given below, and zeroes must be inserted appropriately to reflect the modeling decisions.

$$\begin{pmatrix} \beta_{s1} \\ \beta_{s2} \\ \cdot \\ \cdot \\ \cdot \\ \beta_{sJ} \end{pmatrix} = \begin{pmatrix} 1 & Z_{11} & \dots & Z_{q1} \\ 1 & Z_{12} & \dots & Z_{q2} \\ 1 & \cdot & \dots & \cdot \\ 1 & \cdot & \dots & \cdot \\ 1 & \cdot & \dots & \cdot \\ 1 & Z_{1J} & \dots & Z_{qJ} \end{pmatrix} \begin{pmatrix} \gamma_{s0} \\ \gamma_{s1} \\ \cdot \\ \cdot \\ \cdot \\ \gamma_{sq} \end{pmatrix} + \begin{pmatrix} r_{s0} \\ r_{s1} \\ \cdot \\ \cdot \\ \cdot \\ r_{sq} \end{pmatrix}$$

Although this notation may seem cumbersome initially, we shall see that these “expanded” matrix forms provide savings in distributed estimation since positional information within matrices provides information with respect to model choice at different levels.

The assumptions on the level-2 disturbance term are as follows: (1) the disturbances have 0 expectation ($E(r_{js}) = 0$), (2) the disturbances in different groups are uncorrelated ($E(r_{js}r_{lt}) = 0$ for $j \neq l$), (3) the dispersion of the regression coefficients is the same in each group, e.g., the covariance between the level-1 intercept and level-1 slope is the same across groups ($E(r_{js}r_{jt}) = \tau_{st}$), and 4) the disturbances are uncorrelated with the level-1 disturbance ϵ_{ij} ($E(r_{js}\epsilon_{ij}) = 0$).

As in de Leeuw & Kreft (1986), we employ matrix direct sums to write equations in simpler form. Recall that if A_1, \dots, A_s are matrices, with matrix A_r having n_r rows and J_r columns, then the direct sum $A_1 \dot{+} \dots \dot{+} A_s$ is an $(n_1 + \dots + n_s) \times (J_1 + \dots + J_s)$ block diagonal matrix, with A_1, \dots, A_s as the diagonal blocks. Thus, in the restricted case where every group fits the same level-1 model, $X = X_1 \dot{+} \dots \dot{+} X_J$ is a matrix with $n = \sum n_j$ rows and Jp columns. In the general case where each group fits different level-1 models, we require that each X_j be expanded in an analogous manner to the expansion of Z_s . Specifically, if p is the total number of level-1 variables that exist, each X_j will be $n_j \times p$, but a zero column will exist for each level-1 variable that is not used by group j .

If we stack the J vectors y_j on top of each other to form the n -vector y , and in the same way form the Jp -vector β and the n -vector ϵ , then we can write our model of Equation (1) as

$$y = X\beta + \epsilon. \tag{4}$$

We may also translate the level-2 equation into matrix equation. Define the $p \times pq$ matrix Z_j , with q as before equal to the total number of level-2 variables, and $Z_j = z'_{j1} \dot{+} \dots \dot{+} z'_{jp}$, with

$z'_{js} = 0'$ if group j does not use level-1 variable s . Thus, we may now write:

$$\beta_j = Z_j \gamma + r_j, \quad (5)$$

where $E(r_j) = 0$, $E(r_j r'_j) = \tau$, and $E(r_j r'_l) = 0$, for $j \neq l$. The matrix Z_j may be viewed as an expanded matrix similar to Z_s , but in this case the expansion provides information on model choices at both level-1 and level-2. For example, if only two variables exist at both level-1 and level-2, and all level-2 variables are used to model each level-1 random coefficient, we have:

$$\begin{pmatrix} \beta_{0j} \\ \beta_{1j} \\ \beta_{2j} \end{pmatrix} = \begin{pmatrix} 1 & Z_{1j} & Z_{2j} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & Z_{1j} & Z_{2j} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & Z_{1j} & Z_{2j} \end{pmatrix} \begin{pmatrix} \gamma_{00} \\ \gamma_{01} \\ \gamma_{02} \\ \gamma_{10} \\ \gamma_{11} \\ \gamma_{12} \\ \gamma_{20} \\ \gamma_{21} \\ \gamma_{22} \end{pmatrix} + \begin{pmatrix} r_{0j} \\ r_{1j} \\ r_{2j} \end{pmatrix}.$$

To account for the general case, zero entries must be inserted accordingly. For instance, at level-1, if β_{0j} is not used, this is set equal to zero and the first row of Z_j and r_j must also be set equal to zero. On the other hand, if β_{0j} is used, but only Z_{2j} is used to model β_{0j} at level-2, then the second entry of row one in Z_j is set to zero. With respect to γ , this is viewed as a series of blocks for each of the level-1 coefficients and is independent of j . Thus, we only set a block to zero if the corresponding level-1 coefficient is not used for *all* j . Similarly, a component of a block, say γ_{01} , is only selectively set to zero if the corresponding level-2 variable Z_{1j} is not used by any group to model β_{0j} . In essence, the rows and columns of Z_j provide model choice information for both level-1 and level-2. For instance, if the s th row is equal to zero, then the s th variable was not included at level-1 by this group. If a column is equal to zero, then the corresponding level-2 variable was not included when modeling the corresponding level-1 coefficient for this group. This structure requires that the groups agree on the numbering scheme (which group assumes the first position, the second and so on), their corresponding sample sizes, and also on a numbering of the variables, so that the necessary information can be placed in the relevant position. Hence, this structure does not allow for the case where a group would want to mask its variable IDs for

confidentiality with respect to its chosen model; this would be an entirely different problem.

Stacking equations, we obtain:

$$\beta = Z\gamma + r, \tag{6}$$

where now $E(r) = 0$ and $E(rr') = \tau + \dots + \tau$ (J times).

One may once again combine these equations to yield the single equation format. Letting $U = XZ$ ($n \times q$), we have:

$$y = U\gamma + Xr + \epsilon, \tag{7}$$

2.1 Estimation

There exist several parameters which must be estimated in the multilevel model. The relative interest in the respective parameters is a function of one's research interests. For instance, one may be interested in estimates of the level-1 regression coefficients β_j , where we note that we are using the word estimation somewhat loosely given that these are random variables in the full multilevel model. In addition, one may be interested in the fixed effects γ , i.e., the regression coefficients of the level-2 model that specify the relationship between level-2 covariates and within-group regression coefficients. In the school performance example, these would be considered as 'school effects.' Finally, interest may center around the variance components, i.e., the level-1 variance σ_j^2 and the level-2 variance-covariance matrix τ , where the latter represents the dispersion of the level-1 regression coefficients, conditional upon level-2 covariate information.

The two main approaches to estimation are least squares (LS) and likelihood based methods. LS methods may be partitioned into one-step or two-step methods for estimation of the fixed effects, depending upon whether one adopts the single equation or two equation format of the multilevel model discussed previously. Following the developments and derivations given in Appendix A, we have the equations in Table 2.1 for the LS methods for the various objects of interest. Regarding the properties of the estimators, the LS methods provide unbiased estimates of all parameters of interest. As seen from the equations in the table, the one-step estimates of fixed effects may also be interpreted as a two-step estimator, where step one is the estimation of the level-1 coefficients and step two is the estimation of the level-2 coefficients. The two-step estimates may be produced via either a weighted or un-weighted approach, and we note that they have the same asymptotic distribution. As the two-step estimates are no longer linear in the observations, the simple calculus of bias no longer applies and one must resort to asymptotic methods to evaluate the estimates.

As discussed by de Leeuw & Kreft (1995), although both the one-step and two-step non-weighted LS estimates of fixed effects are unbiased, non-iterative, linear, and easy to implement,

Table 1: Multilevel Estimators

| Name | Estimator | Function of Data |
|---------------------------------------|---|---|
| within-group coefficients | $\hat{\beta}_j = (X_j'X_j)^{-1}X_j'y_j$ | $f(X_j, y_j)$ |
| within-group variance | $\hat{\sigma}_j^2 = e_j'e_j/(n_j - p)$ | $f(e_j)$ |
| 2-step fixed effects | $\hat{\gamma}_s = (Z_s'Z_s)^{-1}Z_s'\hat{\beta}_s$ $\hat{\gamma} = (Z'Z)^{-1}Z'\hat{\beta}$ | $f(\hat{\beta}_s, Z_s)$ $f(Z, \hat{b})$ |
| 1-step fixed effects (un-weighted) | $\hat{\gamma} = (U'U)^{-1}U'y = (Z'Z)^{-1}Z'\hat{\beta}$ $= (\sum_{j=1}^J Z_j'X_j'X_jZ_j)^{-1} \sum_{j=1}^J Z_j'X_j'y_j$ $= (\sum_{j=1}^J Z_j'Z_j)^{-1} \sum_{j=1}^J Z_j'\hat{\beta}_j$ | $f(X_jZ_j, Z_j'X_j'y_j)$ $f(Z_j'Z_j, Z_j'\hat{\beta}_j)$ |
| 1-step fixed effects (weighted) | $\hat{\gamma} = (U'V^{-1}U)^{-1}U'V^{-1}y = (Z'W^{-1}Z)^{-1}Z'W^{-1}\hat{\beta}$ $= (\sum_{j=1}^J Z_j'X_j'V_j^{-1}X_jZ_j)^{-1} \sum_{j=1}^J Z_j'X_j'V_j^{-1}y_j$ $= (\sum_{j=1}^J Z_j'W_j^{-1}Z_j)^{-1} \sum_{j=1}^J Z_j'W_j^{-1}\hat{\beta}_j$ | |
| level-2 dispersion | $\hat{\tau}_{sr} = (t_s't_r - trQ_s\hat{\Sigma}\Delta_{sr}Q_r)/trQ_sQ_r$ | $f(\hat{\beta}_s, \hat{\gamma}_s, Z_s, X_j'X_j)$ |
| shrinkage estimator | $\hat{\beta}_j^* = \Theta_j\hat{\beta}_j + (I - \Theta_j)Z_j'\hat{\gamma}$ | $f(\hat{\beta}_j, Z_j, \hat{\gamma}, \hat{\sigma}_j^2, \hat{\tau})$ |

they are neither best linear unbiased estimates (BLUEs) nor best linear unbiased predictors (BLUPs). The corresponding weighted estimate of the fixed effects estimates is fully efficient, non-iterative, and employs unbiased and consistent estimates of the variance components that can be computed directly from the OLS residuals. (Note that this may lead to negative variance estimates.) See Appendix A for further details.

In addition to the LS methods discussed above, maximum likelihood approaches may also be employed. Maximum likelihood estimates are defined as those estimates of γ , τ and σ that maximize the likelihood function. The full log-likelihood for the j th unit is

$$L_j(\sigma^2, \tau, \gamma) = -\frac{n_j}{2} \log(2\pi) - \frac{1}{2} \log |V_j| - \frac{1}{2} d_j' V_j^{-1} d_j, \quad (8)$$

where $d_j = Y_j - X_j W_j \gamma$ and $\text{Var}(y_j) = V_j = X_j \tau X_j' + \sigma_j^2 I$. Since the J units are independent, we write the log-likelihood for the entire model as a sum of unit log-likelihoods, *i.e.*,

$$L(\sigma^2, \tau, \gamma) = \sum_{j=1}^J L_j(\sigma^2, \tau, \gamma). \quad (9)$$

The calculation of the maximum likelihood estimates is not simple and may be performed via various methods such as the EM algorithm or Fisher Scoring (see Section 3.1).

Regardless of whether one-step, two-step, or maximum likelihood estimates are used in the multilevel model, we may also employ so-called shrinkage estimators for the level-1 regression

coefficients. These may be considered as a compromise between the separate OLS estimates and a prior grand mean estimate employing the data across all of the groups, similar to a classical James-Stein statistical estimator employed in the simple estimation of multiple population means problem. Formally, assuming that the variance components and fixed effects are known or good estimates exist, the multilevel model, i.e., shrinkage, estimate of β_j may be expressed as follows:

$$\hat{\beta}_j^* = \Theta_j \hat{\beta}_j + (I - \Theta_j) Z_j \gamma \quad (10)$$

where

$$\Theta_j = \tau(\tau + \sigma^2(X_j'X_j)^{-1})^{-1} \quad (11)$$

is the ratio of the parameter variance τ for β_j relative to the variance $\sigma^2(X_j'X_j)^{-1}$ for the OLS estimator for β_j plus this parameter variance matrix. Thus, if the OLS estimate is unreliable, $\hat{\beta}_j^*$ will pull $\hat{\beta}_j$ towards $Z_j \hat{\gamma}$, the prior estimate. The shrinkage estimator in equation (10) is often referred to as a Bayes or posterior estimator.

3 Distributed Multilevel Modeling

We now examine the various multilevel model estimators in order to develop the corresponding computational algorithms for a distributed setting. The groups of the multilevel hierarchy may be viewed as components of a distributed database. The goal is to avoid centralized computations, i.e., we would like each of the nodes or sites to perform their own computations and pass along the results to the other nodes. The passing of information may either be performed in a round-robin fashion or in a more sophisticated manner to obtain gains in efficiency.

First, consider the estimation of fixed effects γ , where estimates are obtained via the one-step approach of equation (28). Since the summations are over groups j , each group may calculate their own matrix products and pass them along to the next group. Regarding the first term that includes a matrix inverse, $(\sum_{j=1}^J Z_j' X_j' X_j Z_j)^{-1}$, only the final group needs to perform this inverse. The final group must also multiply this inverse by the second term final sum, $\sum_{j=1}^J Z_j' X_j' y_j$. Under the ‘usual’ notation for multilevel models, the fact that the different groups have fit different models at level-1 and level-2 would result in the matrices X_j and Z_j having different dimensions across groups, and hence preclude a simple summation of the matrix products in equation (28). However, the expanded matrix notation introduced earlier allows the seamless computation of these sums across groups.

Each group must compute matrix products $Z_j' X_j' X_j Z_j$ and $Z_j' X_j' y_j$ under the expanded

matrix format, where recall X_j is $n_j \times p$ and Z_j is $p \times pq$, with zero entries corresponding to the level-1 and level-2 model choices of group j . Now, let U_j represent a $1 \times p$ row vector with a 0 entry corresponding to a 0 column in X_j , meaning that this level-1 variable is not part of the level-1 model for group j . Similarly, let H_j represent a $p \times pq$ indicator matrix for the non-zero entries of the expanded matrix Z_j discussed earlier, reflecting the model choices at level-2 and level-1 for group j . Let $c_j = U_j \times H_j$, a $1 \times pq$ row vector, and let $C_j = c_j' c_j$, a $pq \times pq$ indicator matrix. Note that C_j is of the same dimension as $Z_j' X_j' X_j Z_j$, and represents an indicator matrix of the non-zero components of $Z_j' X_j' X_j Z_j$. Thus, this indicator matrix can be used to determine which rows and columns of the two matrices need to be multiplied. For instance, if the (i, j) th entry of C_j is equal to one, we know that we must compute the (i, j) th entry of $Z_j' X_j' X_j Z_j$. Otherwise, no computation is necessary. The (i, j) th entry of $A = Z_j' X_j' X_j Z_j$ can be computed via the formula below:

$$A_{ij} = \sum_{a=1}^p \left[\left(\sum_{b=1}^n \left[\left(\sum_{c=1}^p (Z_{ci} X_{bc}) \right) X_{ba} \right] \right) Z_{aj} \right].$$

A similar approach may be applied towards the computation of the (i, j) th entry of $Z_j' X_j' y_j$.

We may also obtain the fixed effects estimates in a distributed manner via the two-step estimator: $(\sum_{j=1}^J Z_j' Z_j)^{-1} \sum_{j=1}^J Z_j' \hat{\beta}_j$, where $\hat{\beta}_j$ is the OLS estimator obtained in the separate groups. As with the distributed one-step estimator above, the computations may be done in an analogous manner, where we employ the expanded forms of Z_j and $\hat{\beta}_j$. One advantage of this expanded form is that a particular group need not have to track the modeling choices of the other groups; it merely has to know the total available variables at level-1 and level-2 (and an agreement with respect to which variables correspond to which columns).

One option for the organization of the matrix sums is a round-robin format, i.e., each group computes their corresponding matrix product and passes the result to the next group in a line; See Figure 1. The only requirements are that each group knows where to send their information, and that the last group knows that it is the last group and that it must also compute a matrix inverse and one additional matrix product after this inverse. Thus, if there are J groups, we have essentially J basic steps to the process. Since we are dealing with simple sums, we may also add structure to the group hierarchy in order to obtain gains in efficiency. One possibility is to arrange the groups into a dyadic hierarchy or partition. For example, if there are eight groups, we first group them in pairs, then the pairs are grouped in sets of four, then finally the two sets of four are grouped into a single set of eight. Within each set, one of the groups is the leader and performs the computations for the group. As tasks may now be done in parallel, we obtain efficiency gains, requiring three steps instead of eight. In general, for J groups, a dyadic partition will require d steps, where $2^d = J$, i.e.,

$d = \log(J/2)$. Of course, a group must know the necessary information of the dyadic partition, i.e., it must know where to send its information, and the group that is the leader must know that it must perform the computation. As before, the final ‘group’ should know that it must compute the inverse and multiply the result by the second term final sum. Figure 2 illustrates the dyadic partition.

Thus far, only distributed estimation for the un-weighted estimators has been discussed. The corresponding weighted estimators require estimates of both the level-1 variance σ_j^2 and level-2 variance τ . Regarding the variance component estimation at level-1, one option is to employ the estimator $\hat{\sigma}_j^2 = e_j' e_j / (n_j - p)$ (See Appendix A). Since $e_j = y_j - X_j \hat{\beta}_j$, this information is available at the separate groups and no sharing of information is necessary. On the other hand, if one computes residuals via fitted values computed by a prior estimator $\hat{y}_j = X_j Z_j \hat{\gamma}$, each group would require the value of $\hat{\gamma}$ which was obtained via the method discussed above. This could be sent back to the individual groups after the distributed computation procedure. If groups require residuals via fitted values computed via the shrinkage estimator $\hat{y}_j = X_j \hat{\beta}_j^*$, the procedure is more involved due to the fact that the shrinkage estimator requires an estimate of the level-2 variance τ .

As discussed in Appendix A, an estimate of τ may be obtained via the unbiased estimator:

$$\hat{\tau}_{sr} = (t_s' t_r - \text{tr} Q_s \hat{\Sigma} \Delta_{sr} Q_r) / \text{tr} Q_s Q_r.$$

The problem with this formulation is that t_s and Q_s involve the Z_s matrix, and this represents level-2 information for a particular level-2 variable across all groups. Specifically, we require the result $Z_s' Z_s$. A brute force solution would be to send all the level-2 information to a centralized node and allow the centralized node to perform the computation. The centralized node would essentially receive information on the level-2 model choices of each group. However, a more distributed solution for computing to computing $Z_s' Z_s$ is as follows. The full structure of Z_s is given below for reference:

$$Z_s = \begin{pmatrix} 1 & Z_{11} & \dots & Z_{q1} \\ 1 & Z_{12} & \dots & Z_{q2} \\ 1 & \cdot & \dots & \cdot \\ 1 & \cdot & \dots & \cdot \\ 1 & \cdot & \dots & \cdot \\ 1 & Z_{1J} & \dots & Z_{qJ} \end{pmatrix}.$$

Each group must pass its row of Z_s along to the next group, along with the product of this row

against the other rows that it has received. A particular node does not have to compute the row products of other groups, as it will receive these results. The final group will thus have all the components of the matrix $Z'_s Z_s$. Moreover, each group does not receive information regarding the level-2 model choices of every other group, only the groups that precede it in the round-robin for example. The computations and information being passed are growing at each step: the r th node must pass forward r rows of Z_s , and also compute and pass $r(r - 1)$ row products of $Z'_s Z_s$. Indeed, the amount of information must be multiplied by a factor p , since we have Z_s for each of the p level-1 variables. Nevertheless, the final group will have all the information necessary to compute $Z'_s Z_s$ and hence Q_s . We also require the Δ_{sr} matrix, which recall is $J \times J$ diagonal with the (s, r) component of each $(X'_j X_j)^{-1}$. Thus, each group must calculate $(X'_j X_j)^{-1}$ to supply the components of each Δ_{sr} . If there exist p level-1 variables, each group will thus pass $p(p - 1)/2$ unique results since $(X'_j X_j)^{-1}$ is symmetric. Note that each group must once again perform calculations with expanded matrices to preserve variable-column correspondence. Moreover, each group does not have to manipulate the results of previous groups, they only have to place them in matrix Δ_{sr} . This matrix is ‘filled’ as the process evolves, and thus the final group will have Δ_{sr} for all s and r . Assuming that fixed effects from the previous distributed computation exist at the final node, along with level-1 variance estimate, the final group may compute $\hat{\tau}_{sr}$.

The distributed computation of $\hat{\tau}$ may also be performed via a dyadic partitioning of the nodes. Consider the computation of $Z'_s Z_s$. The necessary information (rows) and results (row products) must be sent ‘up’ the tree such that the subsequent node leaders can perform the remaining calculations. The final node would once again be responsible for forming the final matrices and estimate of τ . Although initially it may seem that a great deal of overhead would be necessary to account for the various pieces of information and which rows of Z_s need to be multiplied, we are dealing with simple products and simple combinations so the procedure is not as involved as one might suspect. For instance, in the case of a $J = 8$ dyadic partition, the leader of each pair initially calculates two row products (its own and against its pair/partner). Next, these results, along with the original row information of itself and partner, are sent up the tree. The leader of the four-sets must now calculate the row products for the groups that didn’t belong to the *same* pair at the first level; knowledge of which rows need to be multiplied saves the leader node from performing unnecessary computations. And so on. Note: if a node is not a node leader, it does not perform *any* computations related to $Z'_s Z_s$. However, each group must calculate $(X'_j X_j)^{-1}$ to supply the components of each Δ_{sr} ; the dyadic partition does not alter this .

Regarding the shrinkage estimates of the level-1 coefficients β_j , these also require estimates of the variance components. Thus, after variance component estimates have been obtained, they could

be distributed to the individual groups, thereby allowing them to calculate the necessary weights for a weighted average of the individual OLS estimate and the prior estimate. The prior estimate $\hat{y}_j = X_j Z_j \hat{\gamma}$ requires the result of the distributed estimate of the fixed effects γ .

3.1 Maximum Likelihood

In addition to the one-step and two-step distributed estimators developed above, we consider distributed estimation via maximum likelihood methods. For the simple estimators, the main hurdle to performing distributed estimation lies in the variance component estimation. For maximum likelihood, the variance components will once again present the main challenge. Below, we consider the computation of full information ML estimates via both the EM algorithm and Fisher scoring.

3.1.1 EM algorithm

First, assume that the level-1 variance is constant across groups. Then the EM algorithm leads to updates of γ , σ^2 and τ with

$$\hat{\gamma} = \left(\sum_{j=1}^J Z_j' X_j' X_j Z_j \right)^{-1} \sum_{j=1}^J Z_j' (X_j' Y_j - X_j' X_j u_j),$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{j=1}^J (d_j' M_j^2 d_j + \sigma^2 \text{tr}[X_j' X_j C_j^{-1}]),$$

and

$$\hat{\tau} = \frac{1}{J} \sum_{j=1}^J (u_j u_j' + \sigma^2 C_j^{-1}),$$

where $d_j = y_j - X_j Z_j \gamma$, $C_j^{-1} = (X_j' X_j + \sigma^2 \tau^{-1})^{-1}$, $M_j = I_j - X_j C_j^{-1} X_j'$, and $u_j = C_j^{-1} X_j' d_j$.

For instance, for the γ estimate, we need initial estimates of not only γ , but also initial estimates of the variance components. As each iteration of the EM algorithm requires a sum over the J groups, this summation could be accomplished in a similar fashion to a previous distributed estimation of fixed effects via the simpler methods. That is, each group computes its corresponding matrix product in the formulas above and passes the result to the next group. However, each of the nodes must possess both the initial and intermediate estimates of fixed effects and variance components as the EM algorithm cycles through its iterations. Regarding the EM expression for the level-2 variance τ , this is also a sum over J groups. This differs from our earlier ‘simple’ estimator, which involved the matrix Z_s containing information for all J groups. Thus, we can

avoid the distributed ‘filling’ procedure that was necessary to construct $Z_s'Z_s$ involving *inter-group* computations. As we now have a simple sum over groups, the calculations at each iteration can now be done in a separated and distributed manner. As with the fixed effects, both the initial and intermediate estimates of fixed effects and variance components are required at each node as the EM algorithm cycles through its iterations.

For all the EM algorithm update formulas, we still need methods with which to assess convergence of the algorithm. This convergence check could be done at the last node in the distributed algorithm or some designated centralized location. Regardless, the convergence assessment will be based on the deviance of the groups, where the group deviance is:

$$D_j = n_j \log(2\pi) + (n_j - r) \log(\sigma^2) + \log |\tau| - \log |C_j^{-1}| + \sigma^{-2} d_j' M_j d_j,$$

where

$$|V_j| = (\sigma^2)^{(n_j - r)} |\tau| / |C_j^{-1}|,$$

and r is the rank of τ . Given this decomposition of deviance per group, each group could pass D_j as it passes the components of the distributed formulas for the fixed effects and variance components. The last node could sum these deviances and perform the convergence check. Or, each of the nodes could keep track of its own unit deviance and perform a convergence check, and pass the result of the convergence check (binary) to the subsequent nodes in the distributed procedure. The last node could then perform an aggregate convergence check based on the results of the unit convergence results that it has received.

3.1.2 Fisher Scoring

The first derivative and expectation of the second derivative of the log-likelihood are needed for the Fisher Scoring algorithm. We will henceforth refer to these two items as the score vector and information matrix. Although expressions for these vectors may be found in Longford (1987) and Bryk & Raudenbush (1992), we follow the development of these terms provided by Hilden-Minton (1995), as these expressions are more amenable to a distributed framework. Below we closely follow the derivation provided by Hilden-Minton (1995).

Recall that the overall log-likelihood can be written as a sum of group log-likelihoods. Thus, we consider a given group log-likelihood and also drop the j subscript for purposes of illustration. To simplify the likelihood, let

$$\lambda = -\frac{1}{2} \{\log |V| + d' V^{-1} d\},$$

where $V = X\tau X' + \sigma^2 I$ and $d = Y - XW\gamma$. Suppose that ϕ and ϕ' are arbitrary elements of (σ^2, τ) . Then,

$$\frac{\partial(\log |V|)}{\partial\phi} = \text{tr} \left(V^{-1} \frac{\partial V}{\partial\phi} \right), \quad (12)$$

and

$$\frac{\partial(V^{-1})}{\partial\phi} = -V^{-1} \frac{\partial V}{\partial\phi} V^{-1}. \quad (13)$$

Thus, we have first and second derivatives

$$\frac{\partial\lambda}{\partial\phi} = -\frac{1}{2} \left\{ \text{tr} \left(V^{-1} \frac{\partial V}{\partial\phi} \right) - d' V^{-1} \frac{\partial V}{\partial\phi} V^{-1} d \right\} \quad (14)$$

and

$$\frac{\partial^2\lambda}{\partial\phi\partial\phi'} = -\frac{1}{2} \left\{ -\text{tr} \left(V^{-1} \frac{\partial V}{\partial\phi'} V^{-1} \frac{\partial V}{\partial\phi} \right) + 2d' V^{-1} \frac{\partial V}{\partial\phi'} V^{-1} \frac{\partial V}{\partial\phi} V^{-1} d \right\}. \quad (15)$$

Now taking the expectation of equation 15, we get

$$E \left(\frac{\partial^2\lambda}{\partial\phi\partial\phi'} \right) = -\frac{1}{2} \text{tr} \left(V^{-1} \frac{\partial V}{\partial\phi'} V^{-1} \frac{\partial V}{\partial\phi} \right). \quad (16)$$

Since $V = X\tau X' + \sigma^2 I$ and τ is symmetric,

$$\frac{\partial V}{\partial(\sigma^2)} = I$$

and

$$\frac{\partial V}{\partial\tau_{ij}} = \begin{cases} X u_i u_i' X' & \text{if } i = j \\ X u_i u_j' X' + X u_j u_i' X' & \text{if } i \neq j \end{cases},$$

where u_i is a column of the identity matrix of dimension P . After some algebra, one obtains

$$\begin{aligned} \frac{\partial\lambda}{\partial(\sigma^2)} &= -\frac{1}{2} \{ \text{tr}(V^{-1}) - d' V^{-1} d \} \\ &= -\frac{1}{2} \{ n\sigma^{-2} - \sigma^{-2} \text{tr}(X' X C^{-1}) - \sigma^{-4} d' M^2 d \} \end{aligned} \quad (17)$$

and

$$\begin{aligned} \frac{\partial\lambda}{\partial\tau_{ij}} &= -\frac{1}{2} \{ \text{tr}(V^{-1} X u_i u_j' X') + (1 - \delta_{ij}) \text{tr}(V^{-1} X u_j u_i' X') \\ &\quad - d' V^{-1} X u_i u_j' X' V^{-1} d - (1 - \delta_{ij}) d' V^{-1} X u_j u_i' X' V^{-1} d \} \\ &= -\frac{2 - \delta_{ij}}{2} \{ u_j' X' V^{-1} X u_i - (u_i' X' V^{-1} d)(u_j' X' V^{-1} d) \} \\ &= -\frac{2 - \delta_{ij}}{2} \{ \sigma^{-2} u_j' X' M X u_i - \sigma^{-4} (u_i' X' M d)(u_j' X' M d) \} \end{aligned} \quad (18)$$

Setting $\frac{\partial \lambda}{\partial \tau} = (\frac{\partial \lambda}{\partial \tau_{ij}})$, we may write equation 18 as

$$\frac{\partial \lambda}{\partial \tau} = -\sigma^{-4}(J - \frac{1}{2}I) * (\sigma^2 X' M X - X' M d(X' M d)'),$$

where J is a matrix of ones and $*$ indicates element-wise multiplication. Also for γ , we have

$$\frac{\partial \lambda}{\partial \gamma} = W' X' V^{-1} d = \sigma^{-2} W' X' M d. \quad (19)$$

Now we simplify the expected second derivatives.

$$\begin{aligned} E \left[\frac{\partial^2 \lambda}{(\partial(\sigma^2))^2} \right] &= -\frac{1}{2} \text{tr}(V^{-2}) \\ &= -\frac{1}{2} \sigma^{-4} \{n - 2\text{tr}(X' X C^{-1}) + \text{tr}(X' X C^{-1} X' X C^{-1})\}. \end{aligned} \quad (20)$$

$$\begin{aligned} E \left[\frac{\partial^2 \lambda}{\partial \tau_{ij} \partial \tau_{kl}} \right] &= -\frac{2 - \delta_{ij}}{2} u'_j X' V^{-1} X \left(\frac{2 - \delta_{kl}}{2} \right) \\ &\quad (u_k u'_l + u_l u'_k) X' V^{-1} X u_i \\ &= -\frac{(2 - \delta_{ij})(2 - \delta_{kl}) \sigma^{-4}}{4} \{ (u'_i X' M X u_k)(u'_j X' M X u_l) \\ &\quad + (u'_i X' M X u_l) u'_j X' M X u_k \} \end{aligned} \quad (21)$$

$$\begin{aligned} E \left[\frac{\partial^2 \lambda}{\partial \sigma^2 \partial \tau_{ij}} \right] &= -\frac{2 - \delta_{ij}}{2} u'_j X' V^{-2} X u_i \\ &= -\frac{2 - \delta_{ij}}{2} \sigma^{-4} u'_j X' M^2 X u_i. \end{aligned} \quad (22)$$

Also $E \left[\frac{\partial^2 \lambda}{\partial \phi \partial \gamma} \right]$ is zero, and

$$E \left[\frac{\partial^2 \lambda}{\partial \gamma \partial \gamma'} \right] = -W' X' V^{-1} X W = -\sigma^{-2} W' X' M X W. \quad (23)$$

Thus, the information matrix is block diagonal.

Consider the score vector and information matrix for the level-1 variance, provided in equations (17) and (20), respectively. We observe that given initial and intermediate estimates of fixed effects and variance components, each group can calculate their respective component of the score vector and information matrix. These may be passed along group to group, once again either in round robin or in a dyadic partition. As with the EM algorithm, either the last node or a centralized node can sum the respective components and perform both the Fisher updates and the convergence

check. The updates of the level-2 variance and fixed effects would proceed in a similar manner, since the respective score vector and information matrices may also be summed across the groups.

We see that the immediate advantage of both Fisher Scoring and the EM algorithm is that we may perform distributed computing for all estimators; i.e., only the results of group computations need to be passed along. The potential drawback is the overhead involved in communicating the intermediate results back to the groups, and then performing the distributed computing again across the groups. On the other hand, for large computations, where maximum likelihood may be very slow, this may be an advantage as computations are distributed across groups and moreover a dyadic partition may allow parallel computation.

Finally, as the shrinkage estimators are simply a weighted average of the OLS and prior estimate, these could be obtained after either EM or Fisher Scoring, once again passing the relevant information down to the groups if necessary.

4 Some additional issues

In this section we briefly examine some additional issues, such as the dynamic updating of the parameters of interest in the presence of new data, tolerance of the distributed algorithms in the event of a group's failure, and some security and privacy issues.

4.1 Recursive Updating

For both least squares and maximum likelihood estimators, we have considered the case of a fixed data set in the sense that the number of groups J is fixed and the number of observations per group n_j is also fixed. We briefly investigate some scenarios that extend the basic framework investigated in this paper. Specifically, we outline how inexpensive updates of the parameters of interest can be obtained when dynamic updates of the data occur. Such updates are required when (i) new observations become available for the i -th group/database, (ii) new groups emerge and (iii) a group expands the features (variables) it collects and wants to include them in the modeling process. This could arise from two scenarios. One, new variables may become available over time and hence be introduced into the current model. Two, different combinations of variables may be selected in response to model assessment criteria. Up until now we have not discussed model assessment, i.e., we have assumed that the given level-1 and level-2 models are acceptable. To be sure, this is an oversimplification, and this constitutes a topic of future research. Although a complete coverage of these topics is beyond the scope of the current paper, we provide some simple updating formulas.

To fix the notation, let $\hat{\gamma}_{(J)}$ represent the estimator with J groups, and $\hat{\gamma}_{(J+1)}$ represent the

estimator with the additional group. The question then becomes whether there exists a simple formula that allows us to obtain $\hat{\gamma}_{(J+1)}$ without running the full algorithm over $J + 1$ groups. This same question can also be applied to the case where the number of groups remains constant, but the number of observations per group n_j has increased. We focus on the estimation of the unweighted fixed effects $\hat{\gamma}_s = (Z'_s Z_s)^{-1} Z'_s \hat{\beta}_s$ when all groups use the same number of level-1 and level-2 variables. Thus, Z_s is a matrix of dimension $J \times q$ and β_s is a vector of size q .

We start by examining the effect of adding a new group, which results in a new row in the Z_s matrix and a new row in the β_s vector. Additionally, because of the way $\hat{\beta}_s$ is estimated, all the elements of this vector will change because the operation will cause the new row to affect all the other rows (a new row in a matrix inversion affects all other rows in the result). Because all elements of $\hat{\beta}_s$ will change, it follows that all the elements of $\hat{\gamma}_s$ will change also. Unfortunately, the inversion part of the formula $(Z'_s Z_s)^{-1}$ causes the new row to change all the results of all the rows. However, we can save some computing time by storing the old $(Z'_s Z)$ matrix product. This is because the following property was found to apply:

$$Z'_{\text{old}} Z_{\text{old}} = H' H + Z'_{\text{new}} Z_{\text{new}}, \quad (24)$$

where $Z'_{\text{old}} Z_{\text{old}}$ is the original product result before adding the new group (which needs to be stored), H is a vector containing all the second level variables of the new group, and $Z'_{\text{new}} Z_{\text{new}}$ is the result of the product that includes the new group.

Next, we examine the effect of adding a new level-2 variable for a total of $q + 1$ variables. This change only affects the Z_s matrix by adding a new column. Unfortunately, this new column causes the result of $(Z'_s Z_s)$ to change dimensions and scramble in such a way that no shortcut has been found to save calculation time. On the other hand, adding a new level-1 variable causes a new column addition in the X matrix. Because of the $(X'_j X_j)^{-1}$ operation, all values of $\hat{\beta}_s$ are affected. However, the Z_s matrix remains unaffected. Therefore, we would save a great deal of computing time by saving the $(Z'_s Z_s)^{-1} Z'_s$ result of the original calculation, because to account for the new level-1 variable we only have to recalculate the $\hat{\beta}_s$ vector and then multiply by this new vector. It's easy to see that the larger the Z_s , the more calculation time will be saved by storing the last $(Z'_s Z_s)^{-1} Z'_s$ obtained.

Finally, although adding new observations to the model affects the variables of the level-1 equation and thus also affects the β_s values in the level-2 equation, the Z_s remains unchanged. Therefore, the same logic as above for adding a new level-1 variable applies: by storing the last result calculated, we could save considerable computing time (in exactly the same way as in Sce-

nario 3). Additionally, notice that adding a new observation adds a new row to the X matrix, and therefore we can also save computing time by applying the same logic as in adding a new group: By storing the original $X_j'X_j$, we can use it to calculate the new one, because:

$$X_{\text{old}}'X_{\text{old}} = H'H + X_{\text{new}}'Z_{\text{new}}, \quad (25)$$

Where H is the vector of the new observations, and X_{old} and X_{new} are defined in a similar fashion of Z_{old} and Z_{old} in the first scenario.

Overall, it can be seen that with a judicious storage of some intermediate products of matrices, significant computational gains can be obtained in most scenarios examined, the sole exception being the addition of new level-2 variables. An interesting question to be considered in future work is whether one can obtain an approximate update by performing an inexpensive calculation and assessing it quantitatively.

4.2 Computational Issues

We briefly discussed some computational complexity issues pertaining to the distributed algorithms presented in Section 3. It can be seen that many expressions in both the least squares and the maximum likelihood algorithms involve taking sums of matrices. In order to illustrate the savings of the algorithm we consider the estimation of the fixed effects γ as a working example. This calculation involves $(\sum_{j=1}^J Z_j'X_j'X_jZ_j)^{-1}$ and $\sum_{j=1}^J Z_j'X_j'y_j$, where Z_j is a $p \times pq$ matrix, X_j an $n_j \times p$ matrix and y_j a n_j -column vector. The product $Z_j'X_j'X_jZ_j$ under the standard outer product version (see Golub and Van Loan (1996) requires $2(qp^2 + p^2n_j + p^2qn_j)$ flops; since, the dominant term is p^2qn_j , this implies that the time complexity of all the terms in the sum is $\mathcal{O}(Jp^2qn_j)$. Further, the addition of those matrices requires $\mathcal{O}(J(pq)^2)$ flops. Analogous calculations apply to the other sum. In many applications, the number of groups is significantly larger than p and/or q and occasionally of the order of n_j , which shows that the distributed algorithms exhibit an order of magnitude of computational savings. Such savings are even more important in the iterative EM algorithm used for obtaining maximum likelihood estimates.

Regarding advice to the user, the upshot of this discussion is that the realized computational savings obtained from the proposed multilevel framework are due to the distribution of such calculations at the individual sites. Thus, if all the data are available in one place, the standard non-distributed algorithms would be advisable. Specifically, in terms of memory, there are no direct savings using the distributed algorithm, since even under a central repository model the groups would be processed sequentially. However, if all the data were stored in a central location, the

input/output operations would have to be scaled by a factor of J . Notice that the extended representation of the multilevel model introduced in Section 2 does not burden any of the calculations due to the availability of efficient sparse matrix representations (Golub and Van Loan, 1996).

4.3 Fault Tolerance

Under distributed estimation, computations must be passed from group to group, either in say round-robin format or in some form of dyadic partition. Regardless, there exists the possibility where one of the groups is unable to send its computation to the subsequent group in the procedure. Thus, there must exist a procedure to bypass this group such that the overall procedure can be completed. In addition, the final group that is responsible for computing the parameter estimates must be aware that the number of groups has changed for any parameter estimates that involve J , the total number of groups. A quick inspection of our table of estimators reveals that this should not be a problem. However, care must also be taken with respect to matrix dimensions, as the final group now has to construct Z_s of dimension $(J - 1) \times q$ instead of $J \times q$. In addition, the structure of the distributed computing has an effect on the amount of information that is lost due to the failure of one group. For instance, assuming that the failed group can be bypassed accordingly under a robust round-robin procedure, only the information from one group is lost. The same happens in a hierarchical tree, if there exists a signaling mechanism that alerts nodes in the hierarchy of lost nodes. In that case, one of the child nodes of the lost one assumes the role of the leader and performs the necessary computations, subject to an adjustment of the dimensions of the matrices involved. The above discussion shows that as long as fault tolerance procedures and the corresponding signaling mechanisms have been incorporated into the procedure, the distributed algorithms discussed are not affected.

4.4 Security/Privacy Issues

Thus far we have operated under the assumption that the distributed databases or groups are owned or managed by a single operational entity. However, in practice, these databases may be owned and managed by distinct governing agencies/bodies. Consider the case of educational data associated with school performance that is managed by state or private educational agencies. Under such a scenario, distributed computing may be the only available option as the formation of an integrated database is often problematic due to security/confidentiality concerns, i.e., access to student and school identities and information is often restricted. However, thus far, we have not considered distributed multilevel modeling under security concerns. Thus, in the various distributed computing

algorithms developed above, we must account for the variety of security/confidentiality concerns that may exist.

These issues lie in the general area known as data confidentiality, or as statistical disclosure limitation in the context of official statistics (Duncan et al., 1993; Willenborg and de Wal, 2001). The essential problem is that many statistical agencies face the conflicting objectives of both protecting the confidentiality of their data subjects and disseminating useful information derived from their data. Karr et al. (2005) describe various levels of security concerns, ranging from the need to protect the group or database identity/origin of the subjects, to the need to protect the associated covariate values for these subjects. Under such confidentiality concerns, they present methods for secure regression on distributed databases. Specifically, they show how to perform secure linear regression for the case of horizontally partitioned data, i.e., the participating agencies possess databases that contain the same numerical information for disjoint sets of cases or subjects. They present a range of solutions for the standard regression problem corresponding to the various levels of data confidentiality.

Karr et al. (2004) consider a data integration approach to integrate the databases while preserving privacy. The goal of secure data integration is to protect the origin of any record. Karr et al. (2004) present a secure data integration algorithm that employs synthetic data and permutation to protect the group/agency identity of records; the synthetic data plays a role similar to that of the random number used in their secure summation (see below). Although their secure data integration approach may be employed to conceal the origins of the records, doing so we will also lose the group structure of the data and thereby preclude the use of multilevel model methods. A compromise approach would be to retain the group structure while still protecting the group identities. There exist at least two ways to accomplish this: 1) The group indices could be permuted, such that a record y_{ij} does not necessarily correspond to the j th group, or 2) The correspondence of the group j indices to the actual nominal group names could be withheld, or 3) synthetic group names could be used to correspond to the group indices, etc. This approach would work so long as the identities could not be inferred from the values of level-1 or level-2 information, e.g., if there was only one private school, the group with this label would be automatically revealed.

In some cases, data integration will not be an option, say due to storage reasons or increased confidentiality concerns (note that the data integration method of Karr et al. (2004) does not provide 100% data confidentiality). Secure multi-party computation represents a viable alternative, i.e., methods for performing computations in which multiple parties hold “pieces” of the computation. The objective is to obtain some final result, and disclose as little information as possible en route to computing this final result. For the case of secure linear regression, Karr et al. (2004)

present a secure summation algorithm for obtaining the estimated regression coefficients. The algorithm begins with the first group adding a random number to its contribution to the desired sum; subsequent groups observe the current cumulative sum and contribute their component to the sum without knowing the contributions of the other groups. The final group passes the result back to the first group, which subtracts the random number to yield the desired sum. The integrity of this procedure depends upon certain assumptions, e.g., whether or not groups collude, etc.

The secure summation protocol can be directly applied to many of our distributed algorithms (for both simple two-step and iterative maximum likelihood) which involve summations of local database computations. In some of the cases the secure summation algorithm might not even be necessary given that the subsidiary components provided by the groups may not reveal relevant information. The secure summation protocol would not directly apply to the weighted two-step approach when estimating the level-2 variance (see Section 3) since this does not involve simple summations of local database computations. Alternatively, one may consider our proposed algorithms for the components of the weighted two-step approach (e.g., $Z'_s Z_s$) as a data integration task, and thus the secure data integration of Karr et al. (2004) may be invoked accordingly.

5 Concluding Remarks

In this paper, we develop algorithms for obtaining both least squares and maximum likelihood estimates of multilevel models for data distributed across databases. In particular, the required data structures at each site, together with the information that needs to be communicated between sites are determined. Some additional issues involving dynamic data updates, computational complexity, tolerance of the algorithms to database failures, and security are also briefly discussed.

In the current presentation the model was considered to be known (agreed upon by the sites beforehand) and fixed. This is perfectly appropriate in the current context, where the emphasis is placed on distributed computations for multilevel models. However, in a privacy/security context, where the participating entities do not want to merge (share) the raw data for such concerns, the model to be estimated may not be agreed upon in advance. In such a setting, it then becomes important to examine model fitting and assessment issues; namely, how the groups can obtain model diagnostics in a distributed fashion, so that at a subsequent stage they can decide on the best multilevel model at the local and global level for both inference and prediction purposes, by only sharing a limited amount of information.

Acknowledgments: The authors would like to thank the Editor Luke Tierney, the AE and an anonymous referee for helpful comments and suggestions. The work of GM was supported in part

by NIH grant NCRR-5P41RR018627-04.

A Estimation Methods

First, consider estimates of the β_j . Although we are estimating random variables, the usual criteria of bias and variance are still relevant (Rao, 1965a; Rao, 1965b; Swamy, 1970, 1971; Pfefferman, 1984). The minimum variance unbiased linear estimate for β_j is $\hat{\beta}_j = (X_j'X_j)^{-1}X_j'y_j$. Thus, we compute regression coefficients separately in the different groups via the standard estimator. In matrix form, we have $\hat{\beta} = (X'X)^{-1}X'y$.

The expectation of $\hat{\beta}_j$ is $E(\hat{\beta}_j) = Z_j\gamma$, and its variance is $W_j = \tau + \sigma_j^2(X_j'X_j)^{-1}$. Defining $W = W_1 + \dots + W_J$, the analogous matrix results are that $\hat{\beta}$ has expectation $Z\gamma$ and dispersion W . In addition, consider the level-1 residuals $e_j = y_j - X_j\hat{\beta}_j$. These residuals have zero expectation and

$$E(e_j e_j') = \sigma_j^2(I - X_j(X_j'X_j)^{-1}X_j').$$

Thus, $E(e_j' e_j) = \sigma_j^2(n_j - p)$, and hence an unbiased estimator is $\hat{\sigma}_j^2 = e_j' e_j / (n_j - p)$. OLS within groups provides unbiased estimates for the level-1 parameters β_j and σ_j^2 . Next, the level-1 regression coefficients may be used to obtain estimates for the fixed effects γ , where we may view this step as the regression of the fixed effects γ on the regression coefficients β_j . Thus, $\hat{\gamma} = (Z'Z)^{-1}Z'\hat{\beta}$. With respect to the J regression coefficients for variable s in the J groups, we may write $\hat{\gamma}_s = (Z_s'Z_s)^{-1}Z_s'\hat{\beta}_s$.

An estimate for the τ matrix, i.e., the variance-covariance matrix of the level-1 regression coefficients, is also required. De Leeuw & Kreft (1986) present a generalization of a method of Rao (1965a) and Swamy (1970) to estimate the level-2 variance matrix. Specifically, define level-2 residuals $t_s = \hat{\beta}_s - Z_s\hat{\gamma}_s$. Writing $t_s = Q_s\hat{\beta}_s$, where $Q_s = I - Z_s(Z_s'Z_s)^{-1}Z_s'$, then $E(t_s) = 0$ and we have:

$$E(t_s t_r') = Q_s(\tau_{sr}I + \Sigma\Delta_{sr})Q_r. \quad (26)$$

Above, Σ is used for the diagonal matrix with the σ_j^2 , and Δ_{sr} for the diagonal matrix with all (s, r) elements of the J matrices $(X_j'X_j)^{-1}$ on the diagonal. From the above equation, we obtain the unbiased estimate

$$\hat{\tau}_{sr} = (t_s' t_r - tr Q_s \hat{\Sigma} \Delta_{sr} Q_r) / tr Q_s Q_r, \quad (27)$$

where $\hat{\Sigma}$ has the $\hat{\sigma}_j^2$ on the diagonal. Thus, after two ordinary least squares steps, we have unbiased estimates of all parameters of interest.

The above 'two-step' estimates may be contrasted with the corresponding estimates that are obtained via the single equation specification (see de Leeuw & Kreft, 1986, for more details). The main result is that there often isn't a huge difference in these approaches. Based on the single

equation format in Equation (7), we may directly write:

$$\hat{\gamma} = (U'U)^{-1}U'y = \left(\sum_{j=1}^J Z_j'X_j'X_jZ_j\right)^{-1} \sum_{j=1}^J Z_j'X_j'y_j. \quad (28)$$

One may also attempt a weighted least squares approach in order to develop procedures that are more satisfactory from a statistical point of view. Specifically, we alter the above solution as follows:

$$\hat{\gamma}_{weighted} = (U'V^{-1}U)^{-1}U'V^{-1}y = \left(\sum_{j=1}^J Z_j'X_j'V_j^{-1}X_jZ_j\right)^{-1} \sum_{j=1}^J Z_j'X_j'V_j^{-1}y_j. \quad (29)$$

Since $\text{Var}(y_j) = V_j = X_j\tau X_j' + \sigma_j^2I$ is generally unknown, we may substitute the estimates of σ_j^2 and τ shown above, as suggested originally by Swamy (1970, 1971). This provides an unbiased estimate of \hat{V} , and γ may then be estimated by substituting \hat{V} for V above. As estimates are no longer linear in the observations, the simple calculus of bias no longer applies and one must resort to asymptotic methods to evaluate the estimates.

De Leeuw & Kreft (1986) utilize a formula from Swamy (1971, p.101) for the inverse of V_j in order to illustrate a dramatic simplification of the estimate above. Specifically, the formula is:

$$V_j^{-1} = \sigma^{-2}[I - X_j(X_j'X_j)^{-1}X_j'] + X_j(X_j'X_j)^{-1}W_j^{-1}(X_j'X_j)^{-1}X_j', \quad (30)$$

where $W_j = \tau + \sigma_j^2(X_j'X_j)^{-1}$. This implies that $X_j'V_j^{-1}X_j = W_j^{-1}$ and that $X_j'V_j^{-1}y_j = W_j^{-1}\hat{\beta}_j$. Thus,

$$\hat{\gamma} = (U'V^{-1}U)^{-1}U'V^{-1}y = (Z'W^{-1}Z)^{-1}Z'W^{-1}\hat{\beta}. \quad (31)$$

Thus, we observe that 1) we have replaced inversion of matrices V_j , of order n_j , by inversion of matrices W_j , of order p , and 2) it is clear from this equation that the Gauss-Markov estimate can be interpreted as a two-step estimate, where step one is the estimation of the level-1 coefficients and step two is the estimation of the level-2 coefficients. Regarding variance components, they are estimated the same as previously. Thus, two-step estimates may be produced via either a weighted or un-weighted approach, and we note that they have the same asymptotic distribution.

In addition to the one-step and two-step methods discussed above, maximum likelihood approaches may also be employed. Maximum likelihood estimates are defined as those estimates of γ , τ and Σ that maximize the likelihood function. The full log-likelihood for the j th unit is

$$L_j(\sigma^2, \tau, \gamma) = -\frac{n_j}{2} \log(2\pi) - \frac{1}{2} \log |V_j| - \frac{1}{2} d_j' V_j^{-1} d_j, \quad (32)$$

where $d_j = Y_j - X_j W_j \gamma$. Since the J units are independent, we write the log-likelihood for the entire model as a sum of unit log-likelihoods, *i.e.*,

$$L(\sigma^2, \tau, \gamma) = \sum_{j=1}^J L_j(\sigma^2, \tau, \gamma). \quad (33)$$

The calculation of the maximum likelihood estimates may be performed via various methods such as the EM algorithm or Fisher Scoring (see Section 3.1).

References

- Bryk, A.S., & Raudenbush, S.W. (1992). *Hierarchical Linear Models: applications and data analysis methods*, Sage: Newbury Park, California.
- de Leeuw, J.n & Kreft, I., eds. (2002). *Handbook of Multilevel Quantitative Analysis*. Boston, Dordrecht, London: Kluwer Academic Publishers. *In Press*.
- de Leeuw, J. & Kreft, I. (1995). "Questioning Multilevel Models," *Journal of the Educational and Behavioral Statistics*, 20, pp.171-189.
- de Leeuw, J. & Kreft, I. (1986). "Random Coefficient Models for Multilevel Analysis," *Journal of Educational Statistics*, 11, pp.57-86.
- Duncan, G.T., Jabine, T.B., and de Wolf, V.A., editors (1993). *Private Lives and Public Policies: Confidentiality and Accessibility of Government Statistics*. National Academy Press, Washington. Report of a Panel on Confidentiality and Data Access, Committee on National Statistics.
- Golub, G.H. and Van Loan, C.E. (1996) *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD
- Hilden-Minton, J. (1995). "Multilevel diagnostics for mixed and hierarchical linear models," unpublished Ph.D. dissertation, department of mathematics, University of California, Los Angeles.
- Judge, G.G., et al. (1988). *Introduction to the theory and practice of econometrics, (2nd ed)*. New York: Wiley.
- Karr, A., Lin X., and Sanil, A. (2005). "Secure Regression on Distributed Databases," *Journal of Computational and Graphical Statistics*, 14, 263-279.
- Longford, N.T. (1987), "A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects," *Biometrika*, 74, 817 - 827.
- Rao, C.R. (1987). "Prediction of Future Observations in Growth Curve Models," *Statistical Science*, 2, 434-471.
- Raudenbush, S.W. and Bryk, A.S. (2002). *Hierarchical Linear Models: applications and data analysis methods*, Sage: Thousand Oaks, California.
- Robinson, G.K. (1991). "That BLUP is a Good Thing," *Statistical Science*, 6, 15-51.
- Willenborg, L.C.R.J. and de Waal, T. (2001). *Elements of Statistical Disclosure Control*. Springer-Verlag, New York.

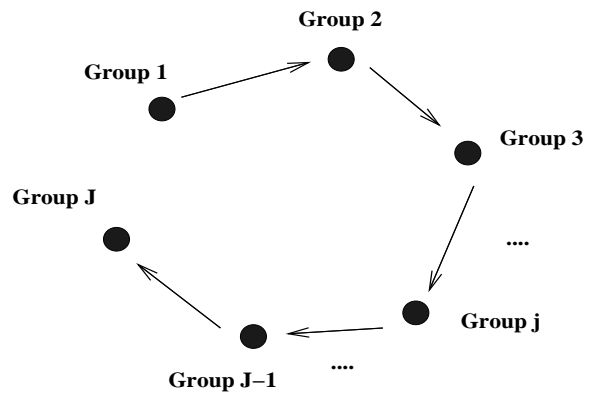


Figure 1: Example of round robin partition of groups.

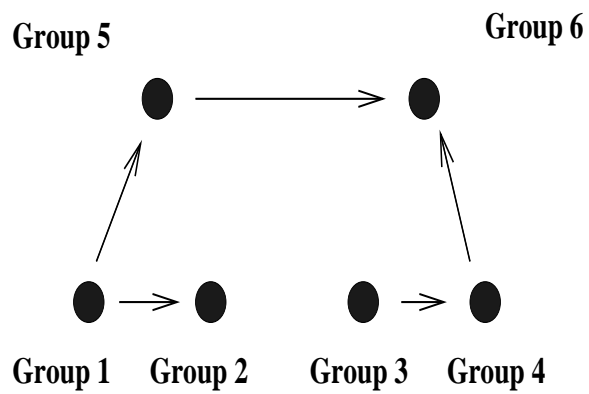


Figure 2: Example of dyadic partition of groups.