

A Review of aML (Release 2)

Fiona Steele
Centre for Multilevel Modelling
Institute of Education, University of London

1. Introduction

1.1 Background

aML was developed by economists Lee Lillard and Stan Panis (Lillard and Panis, 2003), and first became commercially available in 2000. aML may be used to fit a wide range of multilevel models, and specializes in multilevel extensions of models which are commonly used in econometrics. The program was designed specifically for fitting multiprocess or simultaneous equation models to hierarchical data where one or more of the explanatory variables may be non-random or endogenous with respect to the outcome variable of interest.

1.2 Software/hardware requirements

aML is available for DOS, Unix and Linux operating systems. The PC version is a 32-bit application which requires Microsoft Windows 95/98/ME/XP/NT/2000.

1.3 Data input/output functionality

Data input

A potential source of confusion for some users is that the labeling of hierarchical levels is the reverse of that used in most of the multilevel literature and other multilevel software. Thus, level 1 refers to the highest, most aggregated level and subsequent levels are labeled 2, 3, etc. in order of increasing disaggregation. This convention will be used throughout this review.

aML will read data files in ASCII format only and the data must be structured in a particular way before they can be imported. To illustrate the data structure required by aML, suppose that we wish to fit a two-level model to data where students (level 2) are nested within schools (level 1). The input data file must have one record per level 2 unit (though a record may span more than one line). The first variable must contain the level 1 identifiers (called a *control variable* in aML). This is followed by the level 1 variables, and then by the level 2 variables.

For data with more than two levels, the user has a choice. One option is to structure the data so that there is one record for each unit of observation. For example, if we have a three-level structure with students (level 3) nested within classes (level 2) within schools (level 1) we could structure the data so that we have one record per level 3 unit. For situations where individuals are the units of observation, this is generally the most convenient data structure. The other option is to collapse levels so that there is one record per level 2 unit. This alternative is often convenient for repeated measures data. Consider a three-level structure where there are tests (level 3) nested within students (level 2) within schools (level 1). If the data are organized such that each (SAS, SPSS, Stata, ...) record corresponds to a student, with each test score stored as a separate variable, then it is easiest to write out an ASCII data set with one record per student; for each student the test scores, and any other level 3 variables, are written out as arrays with dimension equal to the number of tests for that student. In aML parlance, each test is a level 2 subbranch; the number of subbranches may vary across students. The number of level 2 subbranches is stored as a control variable which must follow the level 1 identifier in the ASCII data file to be read into aML. Full details on data preparation, including examples and sample SAS and Stata code for creating ASCII input files, are given in the User Guide.

aML also supports multiple data structures, a feature that is sometimes useful when fitting multiprocess models. In such cases the variables and/or levels may differ for each structure and it may be convenient to separate data into different data structures. The User's Guide gives the example of marriage formation where the model specification for the first marriage may differ from the specification for subsequent marriages, leading to separate data structures for each.

Once the data have been restructured into the required format, a program called `raw2aml` is run to convert the data into a binary aML data file.

Data output

Two output files are created. The first is the output from the `raw2aml` program and contains summary statistics; this is checked to verify that the data have been read in correctly. The second output file, obtained after running the main program, contains various results from the model estimation. A nice feature of the program is the facility to create an easy-to-read table of parameter estimates and standard errors using the `mktab` program; this program can also be used to tabulate output from several models for ease of comparison.

It is not possible to output data files from aML.

1.4 The interface

aML is a DOS program which may be run only in batch mode. The user creates two batch files: one containing instructions to read in the data, and the other containing the specification of the model to be fitted.

The program offers some limited facilities for carrying out data manipulations, e.g. creating dummy variables and interactions and carrying out some transformations, but the authors recommend that more extensive data manipulations be carried out beforehand in a general purpose statistical package. There is no graphical interface.

2. Standard modeling tools for multilevel analysis

2.1 Available models

It is possible to fit all of the most familiar multilevel models in aML. Hierarchical data with an arbitrary number of levels can be handled. It is also possible to fit some cross-classified models for non-nested structures, but only for continuous outcomes. Table 1 provides a summary of the models which can be fitted in aML. In addition to models for continuous and categorical responses, models for survival data and a number of econometric models are available. For survival data, a semi-parametric model is available in which the baseline log-hazard takes the form of a piecewise linear spline. All models may include random coefficients for the explanatory variables, and complex variation at any level is possible.

aML specializes in the estimation of multiprocess models, otherwise known as simultaneous equation models, of which the multivariate normal model and models for multiple mixed responses are special cases. Any mixture of continuous, categorical, or survival outcomes can be modeled jointly. Multiprocess models are generalizations of models for multivariate responses in which outcomes appear as explanatory variables in equations for other outcomes. These models can be fitted using methods for multivariate responses, provided that criteria for identifiability have been met. More general structural equation models, which include a measurement or factor model, cannot be fitted in aML.

All models are estimated using full information maximum likelihood. Where a closed form solution to the maximum likelihood function does not exist or where the residuals have a multivariate distribution of dimension greater than three, the residuals at each level are ‘integrated out’ numerically using Gauss-Hermite quadrature. The default number of quadrature points is 12, but the user may specify alternative values. In all examples considered in this review, the number of integration points was set to 16. Residuals are assumed to be normally distributed, although for univariate residuals this assumption may be relaxed by allowing residuals to have asymmetric finite mixture distributions.

2.2 Tools for statistical inference and model diagnostics

For each model, the log-likelihood value is output; this may be used to evaluate the overall goodness-of-fit of a model and to compare nested models. The `amltest` program, supplied with `aML`, can be used to perform a likelihood ratio test. T-ratios can also be output using the `mktab` program. By default approximate standard errors are computed, based on an approximation to the Hessian matrix. For small samples, standard errors based on the ‘numerical Hessian’ (the numerical first derivatives of the analytical first derivatives) or robust Huber-corrected standard errors are also available. Throughout this review standard errors based on the numerical Hessian are presented. As computation of numerical standard errors can be slow, users are recommended to first estimate their models with the default standard error option and then to perform one final iteration using the numerical option.

3. Model specifications: basic models

3.1 Two-level continuous response models

We begin by fitting a range of models to two-level continuous response data. The dataset consists of observations on 4059 students from 65 schools in the UK. The following variables were considered in the analyses: normalized exam score (y_{ij}), standardized London reading test score at intake (x_{1ij}), a binary indicator for student’s gender (x_{2ij}) which equals 1 for a girl and 0 for a boy, and a categorical variable for school gender (z_j) which equals 1 for a mixed school, 2 for a boys’ school, and 3 for a girls’ school. Taking the last category as the reference, dummy variables for z_j were created ($z_{1j} = 1$ for mixed school, 0 for other; $z_{2j} = 1$ for a boys’ school, 0 for other).

Notation

The models that were considered are described in Table 2. We adopt the notation of Goldstein (2003). Thus, for example, Model 1C in Table 2 is written:

$$y_{ij} = \beta_{0j} + \beta_{1j}x_{1ij} + \beta_2x_{2ij} + \beta_3z_{1j} + \beta_4z_{2j} + \beta_5x_{1ij}x_{2ij} + \varepsilon_{ij}$$

where

$$\beta_{0j} = \beta_0 + u_{0j}$$

$$\beta_{1j} = \beta_1 + u_{1j}$$

or, alternatively, $y_{ij} = \beta_0 + \beta_1x_{1ij} + \beta_2x_{2ij} + \beta_3z_{1j} + \beta_4z_{2j} + \beta_5x_{1ij}x_{2ij} + u_{0j} + x_{1ij}u_{1j} + \varepsilon_{ij}$.

The subscripts i and j index students (level 2) and schools (level 1) respectively. The student-level residuals, ε_{ij} , are assumed to follow a normal distribution. The school-level residuals (u_{0j}, u_{1j}) are assumed to follow a bivariate normal distribution with covariance matrix

$$\Omega_u = \begin{pmatrix} \sigma_{u0}^2 & \\ \sigma_{u01} & \sigma_{u1}^2 \end{pmatrix}.$$

In aML, standard deviations of the residuals and the correlations between residuals are estimated. The correlation between the school-level residuals in the model above is denoted by ρ_{u01} .

aML syntax

To specify a model in aML, the first step is to define the vector of variables forming the regression equation, called the *regressor set*. Next, the residual distributions are specified. The model statement specifies the type of model to be fitted and the fixed and random part of the regression equation. Finally, starting values for each parameter are supplied by the user. As an illustration, the syntax used to specify Model 1C above is presented here; a detailed explanation is given below. (The line numbers are provided for ease of reference only.)

```

1   define regressor set BetaX; var = 1 x1 x2 (z==1) (z==2) x1*x2;
2   define normal distribution; dim=2; name=u0; name=u1;
3   define normal distribution; dim=1; name=eps;
4   continuous model;
5   outcome=y;
6   model = regset BetaX
7         + res(draw=1,ref=u0)
8         + x1*res(draw=1,ref=u1)
9         + res(draw=_iid,ref=eps);
10  starting values;
11  Beta0      T      *
12  Beta1      T      *
13  Beta2      T      *
14  Beta3      T      *
15  Beta4      T      *
16  Beta5      T      *
17  Sig_u0     T      *
18  Sig_u1     T      *
19  Rho_u01    T      *
20  Sig_eps    T      *;

```

Note: * indicates a value to be supplied by the user.

Line 1 defines the regressor set, i.e. the fixed part of the model, $y_{ij} = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 z_{1j} + \beta_4 z_{2j} + \beta_5 x_{1ij} x_{2ij}$.

Lines 2 and 3 define the distributions of the residuals. The school-level residuals (u_{0j}, u_{1j}) follow a bivariate normal distribution, while ε_{ij} follows a univariate normal distribution. The level at which each set of residuals is defined is specified later.

The type of model to be fitted is specified in lines 4 to 9. Here, we have a continuous outcome variable y_{ij} which is a function of the regressor set BetaX (the fixed part of the model defined in line 1), some residuals, and an interaction between the variable x_{1ij} and residuals u_{1j} . Each residual is assigned to a level and given a label using `res(draw=id, ref=name)`. `Draw=1` indicates that a residual is defined at the highest level (level 1) (recall that the level 1 identifiers were read in as control variables at the data input stage). For residuals at lower levels, the name of the variable containing the identifying codes is substituted. Line 7 indicates that the right hand side of the regression equation will include level 1 residuals, called `u0`. The next line defines random coefficients at level 1, introducing the term $x_{1ij}u_{1j}$. Residuals `u0` and `u1` are defined to be at the same level by specifying the same draw for each. These residuals will be correlated since they were defined to come from a multivariate normal distribution in line 2. Finally, the equation includes level 2 residuals, called `eps`. The `draw=_iid` option indicates that `eps` is defined at the lowest level.

Starting values need to be supplied by the user. Values for the fixed parameters are specified first, followed by the random parameters in the order that they appear in the definition statements. A ‘T’ indicates that a parameter is to be estimated; an ‘F’ may be specified to constrain a parameter to the initial value. Several tips for obtaining good starting values are given in the User’s Guide. The general advice is to start with a simple model, releasing a small number of parameters for estimation at each stage. For example, in practice Model 1C was estimated in the following stages:

- i) A random intercepts model was estimated using OLS estimates as starting values for the fixed part parameters. To obtain starting values for the level 1 and 2 residual standard deviations, the residual variance from the OLS model was divided by two.
- ii) The coefficient of x_{1ij} was allowed to vary randomly across schools, but $\text{Corr}(u_{0j}, u_{1j})$ was assumed to be zero.
- iii) $\text{Corr}(u_{0j}, u_{1j})$ was estimated.

Results

Table 3 shows the parameter estimates and standard errors for the models described in Table 2. Also presented are details of the starting values used and the convergence times (for a Pentium III, 1000 MHz with 256 Mb RAM using Windows 2000 OS). The syntax used to fit the models is given in Appendix A.

3.2 Three-level continuous response models

Models were fitted to three-level data with students (level 3) nested within educational establishments (level 2) within Local Education Authorities (LEA). The outcome variable of interest (y_{ijk}) is the total A/AS point score for student i in establishment j in LEA k , which is treated as continuous. A three-level variance components model with no covariates was considered, i.e.

$$y_{ijk} = \beta_0 + v_k + u_{jk} + \varepsilon_{ijk}, \quad \text{(Model 2)}$$

where v_k , u_{jk} and ε_{ijk} are normally distributed residuals with standard deviations σ_v , σ_u and σ_ε respectively. The syntax for specifying Model 2 is given in Appendix A.

This is a large dataset with 31022 individuals in 2410 establishments in 131 LEAs. Using the standard version of aML, it was not possible to fit multilevel models to these data. An error message

is obtained, stating that the memory has been exceeded due to large cluster sizes; in these data, there are up to 969 students per LEA. It should be noted, however, that aML was designed with applications where there are few observations per hierarchical unit in mind (e.g. repeated measures within individuals). To partially address this problem extended versions of the program, called *bigaml* and *hugeaml*, are supplied free of charge with the standard version. Multilevel models were successfully fitted to the data described above using *bigaml*, but convergence times were very slow. The results for Model 2 and a 2-level model are shown in Table 4.

To obtain starting values for the three-level model, a two-level model (omitting the LEA level) was fitted first. In fitting the three-level model, the starting value for σ_ε was taken as its estimate from the two-level model; starting values for σ_u and σ_v were obtained using the estimate for σ_u obtained from the two-level model and assuming a ratio of establishment-level variance to LEA-level variance of 2:1.

3.3 Two-level models for binary data

Two-level models were fitted to binary data from the 1989 Bangladesh Fertility Survey. The data are a subsample of 1934 women grouped into 60 districts. The outcome variable is use of contraception at the survey (y_{ij}) which equals 1 for using contraception and 0 otherwise. Denote $\Pr(y_{ij}=1)$ by π_{ij} . Three covariates are considered: age at survey, centred at the sample mean (x_{1ij}); type of region of residence (x_{2ij}) which equals 1 for urban and 0 for rural; and number of living children (0=none, 1=one, 2=two, 3=three or more), represented by three dummy variables for the last three categories (x_{3ij} , x_{4ij} and x_{5ij} respectively). Logit/probit models with random coefficients were considered; the full specifications are given in Table 5. The syntax used to specify the models is given in Appendix B.

The parameter estimates, standard errors, details of starting values, and convergence times are shown in Table 6.

3.4 Two-level models for count data

Two types of model are available for analyzing count data in aML: Poisson and negative binomial models. We illustrate Poisson models here using two-level data on malignant melanoma mortality from 354 counties within 78 regions. The data are from 9 European countries, so there are potentially three hierarchical levels; we could allow for variation at the country level using fixed effects, but we do not pursue this here. The data consist of the number of observed deaths (y_{ij}) and expected deaths (E_{ij}) for county i in region j and a single explanatory variable, ultraviolet radiation exposure (x_{ij}). The expected number of deaths is proportional to the size of the population exposed to the risk of mortality in a region and is therefore our ‘exposure’ variable. The standardized mortality rate (SMR) is:

$$\lambda_{ij} = \frac{y_{ij}}{E_{ij}}.$$

The following two-level variance components Poisson model for the SMR was considered:

$$\log(\lambda_{ij}) = \beta_0 + \beta_1 x_{ij} + u_{0j}, \quad (\text{Model 4})$$

where $u_{0j} \sim N(0, \sigma_{u0}^2)$. The syntax used to specify this model is given in Appendix C, and the parameter estimates and standard errors are shown in Table 7.

It is possible to allow for overdispersion in aML by specifying a model of the form:

$$\log(y_{ij}) = \alpha \log(E_{ij}) + \beta_0 + \beta_1 x_{ij} + u_{0j},$$

where α is the overdispersion parameter.

3.5 Growth type models for repeated measures data

Two-level models were fitted to repeated measures data. Measurements on height (y_{it}) were taken on nine occasions (level 2) for 26 boys (level 1) between the ages of 11 and 13 years. The measurements are approximately 0.25 year apart. Details of the two models that were considered are given in Table 8. The syntax used to specify the models is given in Appendix D. In the first model, height is modeled as a cubic function of age (x_{it}); random coefficients for the linear and quadratic age terms allow for individual variation in growth curves. In the second model, AR(1) residuals are assumed at the occasion level. Other autoregressive and moving average models available in aML are AR(p), MA(q), and ARMA(p, q) for $p \leq 9$ and $q \leq 9$. Autoregressive and moving average residuals may be incorporated in any model that can be fitted in aML, although ARMA residuals may be used only in continuous outcome models. The results are shown in Table 9.

4. Model specifications: other multilevel models

4.1 Models for ordered categorical responses

aML can fit ordered logit and probit models to ordered categorical responses, and supports models where the values of the thresholds that delimit categories are known or unknown. In addition, outcomes that span more than one category are permitted. We consider only the ordered logit model with unknown thresholds. Two-level models are fitted to a subsample from the British Social Attitudes Survey. The data consist of responses to a set of dichotomous items on a woman's right to have an abortion under different circumstances. The outcome variable (y_{ij}) is a score constructed from these items ranging from 1 to 7, with a higher score corresponding to stronger support for abortion. Each of 410 respondents was asked the same set of questions on four occasions, 1983-1986, and y_{ij} denotes the response at year i for individual j . We consider one categorical covariate, religion (1=Roman catholic, 2=Protestant/Church of England, 3=Other, 4=None), represented by three dummy variables for the last three categories (x_{1j} , x_{2j} and x_{3j} respectively).

A random intercept ordered logit model is fitted:

$$\text{logit}[\Pr(y_{ij} \leq s)] = \tau_s + \beta_1 x_{1j} + \beta_2 x_{2j} + \beta_3 x_{3j} + u_j, \quad s = 1, \dots, 6 \quad (\text{Model 6})$$

where τ_s are the unknown thresholds, and u_j are individual-specific random effects which are assumed to be normally distributed with standard deviation σ_u . This model is commonly known as a proportional odds model or normal logistic model. The syntax used to specify Model 6 is given in Appendix E.

The parameter estimates and standard errors are shown in Table 10. A single-level model, with very crude starting values, was fitted first and the estimates from this model were used as starting values for the two-level model.

4.2 Models for unordered categorical responses

Multinomial logit and probit models can be fitted in aML. Here, we consider a random intercepts multinomial logit model fitted to data with a two-level structure. The data are a subsample from the Junior School Project with 1313 children nested within 48 primary schools (see Yang et al. 1999). The outcome variable is a teacher assessment of a child's behaviour, which is coded as: 1=top 25%, 2=middle 50% and 3=bottom 25%. We consider only one covariate, gender (1=boy, 0=girl) denoted by x_{ij} . Taking the second response category as the reference, a random intercepts multinomial logit model is written as:

$$\log \left[\frac{\Pr(y_{ij} = s)}{\Pr(y_{ij} = 2)} \right] = \beta_0^{(s)} + \beta_1^{(s)} x_{ij} + u_j^{(s)}, \quad s = 1, 3 \quad (\text{Model 7})$$

The school-level random effects $u_j^{(s)}$ are assumed to follow a bivariate normal distribution with standard deviation $\sigma_u^{(s)}$ ($s=1,3$) and correlation $\rho_u^{(13)}$. The syntax used to fit Model 7 is given in Appendix E.

The parameter estimates and standard errors are shown in Table 10. Starting values for the random effects model are from a single-level model.

4.3 Models for survival data

aML supports proportional hazards model where the baseline hazard is represented by a piecewise-linear spline (also known as a piecewise-linear Gompertz model). Time-varying covariates can be incorporated. Durations need to be input in the form of a lower and upper bound for each time interval. For example, if durations are measured in months then an event that occurred during the 10th month might be recorded with a lower bound of 9 and an upper bound of 10. When specifying the model, the user must specify the nodes for the piecewise-linear baseline hazard function. The slopes of the lines joining these nodes are then estimated. The User's Guide provides some advice for choosing an appropriate number of nodes.

We illustrate the specification of a multilevel survival model using data on contraceptive pill discontinuation in Morocco (Steele 2003). The data have a three-level structure with 1530 pill episodes nested within 1077 women in 105 sampling clusters. The survival time of interest is the duration in months of a continuous period of pill use until discontinuation. We consider two covariates: woman's education level (1=none, 2=primary, 3=secondary+) and age at the start of the episode (1=less than 25, 2=25-34, 3=older than 35). Dummy variables are created for the last two education categories (x_{1jk} and x_{2jk}) and for the first and last age categories (x_{3ijk} and x_{4ijk}). A three-level random intercepts model of the following form is fitted:

$$\log[h_{ijk}(t)] = f(t) + \beta_1 x_{1jk} + \beta_2 x_{2jk} + \beta_3 x_{3ijk} + \beta_4 x_{4ijk} + u_{jk} + v_k \quad (\text{Model 8})$$

where $f(t)$ is the baseline hazard which is represented by a piecewise-linear spline with nodes at 12, and 24 months, i.e.

$$f(t) = \alpha_0 + \alpha_1 \min[t, 12] + \alpha_2 \max[0, \min[t - 12, 12]] + \alpha_3 \max[0, t - 24].$$

The parameters α_1 , α_2 and α_3 are the slopes of the piecewise linear segments corresponding to 0-12, 12-24 and 24+ months respectively. The woman and cluster-level random effects, u_{jk} and v_k , are assumed to follow normal distributions with standard deviations σ_u and σ_v . The syntax used to specify Model 8 is given in Appendix E.

4.4 Models for mixed responses (multiprocess models)

aML was specifically designed for the joint modeling of two or more processes. The outcomes for these processes may be mixtures of any type of outcome variable supported by aML, e.g. continuous, binary, ordered/unordered categorical, survival, etc. As an illustration, we consider an example in which we jointly model a continuous response (y_{1ij} , an English reading test score) and a ordered categorical response (y_{2ij} , a behaviour rating). The data have a two-level structure with observations on 1119 children in 48 primary schools. We consider only one covariate, gender (x_{ij}).

The multiprocess model consists of two components: a continuous response model for y_{1ij} and an ordered probit model for y_{2ij} . We assume that underlying the ordered response y_{2ij} , there is a continuous latent variable y_{2ij}^* such that

$$y_{2ij} = \begin{cases} 1 & \text{if } y_{2ij}^* < \tau_1 \\ 2 & \text{if } \tau_1 \leq y_{2ij}^* < \tau_2 \\ 3 & \text{if } y_{2ij}^* \geq \tau_2 \end{cases}$$

where τ_1 and τ_2 are the thresholds separating the three categories.

The multiprocess model is then written as:

$$\begin{aligned} y_{1ij} &= \beta_0 + \beta_1 x_{ij} + u_{1j} + \varepsilon_{1ij} \\ y_{2ij}^* &= \gamma_1 x_{ij} + u_{2j} + \varepsilon_{2ij} \end{aligned} \quad \text{(Model 9)}$$

where the school-level random effects, u_{1j} and u_{2j} , are assumed to follow a bivariate normal distribution with standard deviations σ_{u1} and σ_{u2} , and correlation ρ_{u12} . The student-level random effects, ε_{1ij} and ε_{2ij} , are assumed to follow univariate normal distribution with standard deviations σ_{e1} and $\sigma_{e2}=1$.

The syntax used to specify Model 9 is given in Appendix E.

4.5 Models for cross-classified structures

Some types of cross-classified structures can be handled in aML. In the User's Guide an example is given where test scores are nested within students but, because students may have several teachers and teachers will teach more than one group of students, students and teachers are non-nested. Provided that there is clustering at the highest level, for example both students and teachers are nested within schools, it is straightforward to allow for student and teacher effects. It is not possible to fit cross-classified models to non-nested structures where there is no clustering at the highest level, for example if students are cross-classified by primary and secondary schools and no higher level of clustering can be identified.

4.6 Other models

Other types of multilevel model supported by aML, which have not been considered in this review, include:

- Multinomial probit models.
- Tobit models for censored normal outcomes.
- Truncated normal regression models.
- Heckman selection models.

5. Documentation and user support

A 446-page User's Guide is provided with the software, and some additional technical support is available on the aML website in the form of a short list of frequently asked questions. The User's Guide is clearly written, and includes numerous examples of syntax and output. In particular, data preparation is thoroughly explained and the inclusion of sample SAS and Stata code for creating data files in the required format is very useful.

The User's Guide assumes that users have a fairly high level of expertise in statistical modeling, and already have some familiarity with the models they wish to fit. For example, someone who is not familiar with multilevel modeling would need to consult at least an introductory text before turning to aML. For some more advanced topics, for example classical simultaneous equation models, it would be helpful if suggestions for further reading were provided.

Given the wide range of models that can be fitted in aML, there is scope for more extensive documentation. One example where more detailed documentation would be useful is the discussion of classical simultaneous models in the chapter on multiprocess modeling. Further discussion of these models, including details of how to specify models for different types of outcome variable, with examples of output would be a welcome addition.

There are two avenues for additional support. An excellent email support service is offered to registered aML users by Stan Panis. I have made extensive use of this service, and found the responses to be both quick and thorough. Users are also encouraged to post conceptual questions and modeling questions, not directly related to the technical operation of aML, to the aML users forum.

6. Final remarks

aML offers the widest range of models for hierarchical data among all multilevel software packages. A particular feature that is unique to aML is the potential to fit multiprocess (simultaneous equation) models to any mixture of outcome variables. aML is also recommended for fitting survival models and the econometric models mentioned in Section 4.6. For users with univariate continuous/categorical outcomes and multivariate continuous outcomes, however, there are easier-to-use packages on the market. Also, users with datasets that include very large cluster sizes would be advised to consider other packages.

The syntax is intuitive so that, once the user has grasped the fundamental concepts such as regressor sets and residual draws, more complex models can be specified fairly straightforwardly. The need to provide starting values in the syntax file is a little awkward, though these can be updated easily using the `update` program that comes with aML. Users should take care with the order in which they list starting values, particularly for some multiprocess models that are not documented in the User's Guide; it is important that users check that the order is correct by examining the start of the output file.

The flexibility of aML comes at some cost in terms of computational time. In general, computational time increases with cluster size and the dimension of the random effect covariance matrix. Also, convergence problems are likely to be encountered if the user does not build in model complexity gradually. Although some guidance is given in the User's Guide, the selection of suitable starting

values can be an art, and several attempts are sometimes necessary before convergence can be achieved.

Contact details and price

Full details on ordering aML can be found at www.applied-ml.com. A free demonstration version of the software can also be downloaded from this site. Current costs (all including a copy of the User Guide) are: \$1,199 for a standard license, \$599 for an academic license (including government and non-profit organizations), and reduced prices for student and course licenses.

References

Goldstein, H. (2003) *Multilevel Statistical Models*. 3rd edition. London: Arnold. (*in press*)

Lillard, Lee A. and Constantijn W.A. Panis (2003) *aML Multilevel Multiprocess Statistical Software, Version 2.0*. EconWare, Los Angeles, California.

Steele, F. (2003) Selection Effects of Source of Contraceptive Supply in an Analysis of Contraceptive Discontinuation: Multilevel Modelling when Random Effects are Correlated with an Explanatory Variable, *Journal of the Royal Statistical Society, Series A*, 166(3) (*to appear*).

Yang, M., Rasbash, J., Goldstein, H., and Barbosa, M. (1999) *MLwiN Macros for Advanced multilevel Modelling*. Institute of Education, London.

Table 1. Multilevel models which can be fitted in aML

Data/ model type	Covariates	Random slopes	Weighting	Variance function*	Comments
Continuous	Yes	Yes	Yes	Yes	
Binary/binomial	Yes	Yes	Yes	Yes	Logit and probit links
Poisson	Yes	Yes	Yes	Yes	
Negative binomial	Yes	Yes	Yes	Yes	
Repeated measures	Yes	Yes	Yes	Yes	
Nominal multinomial	Yes	Yes	Yes	Yes	Multinomial logit and multinomial probit
Ordered multinomial	Yes	Yes	Yes	Yes	Logit and probit links Known and unknown thresholds
Survival	Yes	Yes	Yes	Yes	
Time series	Yes	Yes	Yes	Yes	ARMA(p,q) and CAR(1)
Multivariate Normal	Yes	Yes	Yes	Yes	<i>Seemingly unrelated regression</i> is a special case where different covariates appear in each equation
Multiple mixed responses	Yes	Yes	Yes	Yes	Any mixture of continuous, binary, categorical, survival, time series responses, etc.
Tobit or censored normal	Yes	Yes	Yes	Yes	
Truncated normal	Yes	Yes	Yes	Yes	
Heckman selection	Yes	Yes	Yes	Yes	

*Heteroskedasticity as a function of covariates at any level.

Table 2. Specifications of models fitted to two-level continuous data

Model	Description	Equation
M1A	Variance components with covariates	$y_{ij} = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 z_{1j} + \beta_4 z_{2j} + u_{0j} + \varepsilon_{ij}$
M1B	Variance component with interaction between x_1 and x_2	$y_{ij} = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 z_{1j} + \beta_4 z_{2j} + \beta_5 x_{1ij} x_{2ij} + u_{0j} + \varepsilon_{ij}$
M1C	Random coefficient for x_{1ij}	$y_{ij} = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 z_{1j} + \beta_4 z_{2j} + \beta_5 x_{1ij} x_{2ij} + u_{0j} + x_{1ij} u_{1j} + \varepsilon_{ij}$
M1D	Student-level variance depends on x_{2ij}	$y_{ij} = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 z_{1j} + \beta_4 z_{2j} + \beta_5 x_{1ij} x_{2ij} + u_{0j} + x_{1ij} u_{1j} + x_{2ij} \varepsilon_{0ij} + (1 - x_{2ij}) \varepsilon_{1ij}$ <p>where $\text{Corr}(\varepsilon_{0ij}, \varepsilon_{1ij}) = 0$</p>

Note: See Section 3.1 for variable descriptions.

Table 3. Details of model fitting: models for two-level continuous data

Model	Parameter	Estimate (SE)	Starting values
M1A	β_0 β_1 β_2 β_3 β_4 σ_{u0} σ_ε	-0.010 (0.073) 0.560 (0.012) 0.167 (0.034) -0.159 (0.087) 0.013 (0.123) 0.285 (0.029) 0.750 (0.008)	OLS estimates for fixed part parameters. Starting values for σ_{u0} and σ_ε based on the OLS residual variance, assuming equal variance at each level.
M1B	β_0 β_1 β_2 β_3 β_4 β_5 σ_{u0} σ_ε	-0.004 (0.076) 0.562 (0.018) 0.167 (0.034) -0.164 (0.087) 0.015 (0.123) -0.003 (0.025) 0.285 (0.029) 0.750 (0.008)	Final estimates for M1A with $\beta_5=0$.
M1C	β_0 β_1 β_2 β_3 β_4 β_5 σ_{u0} σ_{u1} ρ_{u01} σ_ε	-0.010 (0.073) 0.551 (0.026) 0.169 (0.034) -0.179 (0.080) -0.002 (0.114) 0.008 (0.030) 0.282 (0.029) 0.121 (0.019) 0.588 (0.140) 0.742 (0.008)	Final estimates from M1B with $\sigma_{u1}=0.5$ and $\rho_{u01}=0$.
M1D	β_0 β_1 β_2 β_3 β_4 β_5 σ_{u0} σ_{u1} ρ_{u01} $\sigma_{\varepsilon0}$ $\sigma_{\varepsilon1}$	-0.012 (0.073) 0.550 (0.026) 0.169 (0.034) -0.179 (0.080) -0.000 (0.114) 0.007 (0.030) 0.282 (0.029) 0.121 (0.019) 0.585 (0.139) 0.766 (0.014) 0.724 (0.010)	Final estimates from M1C with $\sigma_{\varepsilon0}=\sigma_{\varepsilon1}=0.742$ (the residual variance estimate from M1C).

Note: All models converged in less than 20 seconds.

Table 4. Details of model fitting: models for three-level continuous data

Model	Parameter	Estimate (SE)	Starting values
2-level	β_0 σ_u σ_ε	5.331 (0.042) 1.697 (0.035) 2.919 (0.012)	OLS estimate for β_0 . Starting values for σ_u and σ_ε based on the OLS residual variance, assuming equal variance at each level.
3-level (Model 2)	β_0 σ_v σ_u σ_ε	5.317 (0.058) 0.387 (0.069) 1.660 (0.036) 2.917 (0.012)	Estimates from 2-level model for β_0 and σ_ε . Starting values for σ_v and σ_u based on the estimate of σ_u from the 2-level model, assuming a ratio of 2:1 for $\sigma_u^2 : \sigma_v^2$.

Note: The 2-level model converged in 6.9 hours, and the 3-level model in 4.8 hours.

Table 5. Specifications of models fitted to two-level binary response data

Model	Description	Equation
M3A	Variance components with covariates (logit link)	$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3ij} + \beta_4 x_{4ij} + \beta_5 x_{5ij} + u_{0j}$
M3B	Random coefficient on x_{2ij} (logit link)	$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3ij} + \beta_4 x_{4ij} + \beta_5 x_{5ij} + u_{0j} + x_{2ij} u_{1j}$
M3C	Random coefficient on x_{2ij} (probit link)	$\text{probit}(\pi_{ij}) = \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3ij} + \beta_4 x_{4ij} + \beta_5 x_{5ij} + u_{0j} + x_{2ij} u_{1j}$

Notes: See Section 3.3 for variable descriptions.

Table 6. Details of model fitting: models for two-level binary data

Model	Parameter	Estimate (SE)	Starting values
M3A	β_0	-1.686 (0.148)	Fixed part starting values from single-level logit model; $\sigma_{u0}=0.5$.
	β_1	-0.026 (0.008)	
	β_2	0.733 (0.119)	
	β_3	1.117 (0.158)	
	β_4	1.359 (0.174)	
	β_5	1.339 (0.180)	
	σ_{u0}	0.470 (0.083)	
M3B	β_0	-1.725 (0.161)	Starting values for fixed part and σ_{u0} from M2A. Starting values for σ_{u1} and ρ_{u01} are 0.5 and 0 respectively.
	β_1	-0.027 (0.008)	
	β_2	0.826 (0.170)	
	β_3	1.142 (0.160)	
	β_4	1.366 (0.177)	
	β_5	1.363 (0.183)	
	σ_{u0}	0.623 (0.102)	
	σ_{u1}	0.810 (0.199)	
	ρ_{u01}	-0.787 (0.122)	
M3C	β_0	-1.047 (0.095)	Model estimated in two stages: i) variance components model estimated (taking single-level estimates as starting values and $\sigma_{u0}=0.5$); ii) random coefficients model estimated taking variance components estimates as starting values for fixed part parameters and σ_{u0} , and $\sigma_{u1}=0.5$, $\rho_{u01}=0$.
	β_1	-0.016 (0.005)	
	β_2	0.507 (0.107)	
	β_3	0.689 (0.096)	
	β_4	0.825 (0.106)	
	β_5	0.825 (0.109)	
	σ_{u0}	0.379 (0.061)	
	σ_{u1}	0.496 (0.120)	
	ρ_{u01}	-0.797 (0.117)	

Note: M3A and M3B converged in less than 10 seconds; M3C converged in 45 seconds.

Table 7. Details of model fitting: Poisson model for two-level count data

Model	Parameter	Estimate (SE)	Starting values
M4	β_0	-0.138 (0.017)	Fixed part starting values from single-level Poisson model; $\sigma_{u0}=0.5$.
	β_1	-0.056 (0.004)	
	σ_{u0}	0.361 (0.014)	

Note: Model converged in 5 seconds.

Table 8. Specifications of growth type models for continuous repeated measures data

Model	Description	Equation
M5A	Polynomial growth curve (up to cubic) with random coefficient (up to quadratic)	$y_{it} = \beta_0 + \beta_1 x_{it} + \beta_2 x_{it}^2 + \beta_3 x_{it}^3 + u_{0i} + x_{it} u_{1i} + x_{it}^2 u_{2i} + \varepsilon_{it}$
M5B	M5A with AR(1) residuals at occasion level	$y_{it} = \beta_0 + \beta_1 x_{it} + \beta_2 x_{it}^2 + \beta_3 x_{it}^3 + u_{0i} + x_{it} u_{1i} + x_{it}^2 u_{2i} + \varepsilon_{it}$ where $\varepsilon_{it} = \phi \varepsilon_{i,t-1} + \delta_{it}$, $\text{Cov}(\varepsilon_{it}, \varepsilon_{js}) = 0$ for $i \neq j$

Note: See Section 3.5 for variable descriptions.

Table 9. Details of model fitting: growth type models for two-level continuous data

Model	Parameter	Estimate (SE)	Starting values
M5A	β_0	148.992 (1.548)	Estimated in 3 stages: i) variance components model using OLS estimates as starting values for fixed part parameters; ii) random coefficient for x_{it} ; iii) random coefficient for x_{it}^2 .
	β_1	6.177 (0.352)	
	β_2	0.748 (0.176)	
	β_3	0.461 (0.163)	
	σ_{u0}	7.890 (1.107)	
	σ_{u1}	1.666 (0.237)	
	σ_{u2}	0.789 (0.141)	
	ρ_{u01}	0.612 (0.127)	
	ρ_{u02}	0.221 (0.214)	
	ρ_{u12}	0.667 (0.139)	
	σ_ε	0.469 (0.027)	

Note: M5A converged in 20 seconds (taking total across 3 stages); M5B, using starting values from M5A with $\phi=0.5$, did not converge.

Table 10. Results from fitting multilevel models to ordered and unordered categorical outcomes

Model	Parameter	Estimate (SE)	Starting values
M6: Ordered logit	τ_1	-4.056 (0.640)	Starting values for fixed part parameters from single-level model; $\sigma_u = 0.5$.
	τ_2	-2.300 (0.604)	
	τ_3	0.510 (0.593)	
	τ_4	1.449 (0.595)	
	τ_5	2.341 (0.601)	
	τ_6	3.384 (0.608)	
	β_1	2.064 (0.634)	
	β_2	0.688 (0.707)	
	β_3	2.841 (0.651)	
	σ_u	2.283 (0.161)	
M7: Unordered logit	$\beta_0^{(1)}$	-0.587 (0.141)	Starting values for fixed part parameters from single-level model; $\sigma_u^{(1)} = \sigma_u^{(3)} = 0.5$, and $\rho_u^{(13)} = 0$.
	$\beta_1^{(1)}$	-0.550 (0.146)	
	$\beta_0^{(3)}$	-1.159 (0.127)	
	$\beta_1^{(3)}$	0.553 (0.145)	
	$\sigma_u^{(1)}$	0.688 (0.120)	
	$\sigma_u^{(3)}$	0.368 (0.105)	
	$\rho_u^{(13)}$	-0.220 (0.329)	

Notes:

1. See Sections 4.1 and 4.2 for details of model specifications and variable definitions.
2. Both models converged in 5 seconds.

APPENDIX A: Syntax for two- and three-level continuous response models

Two-level models are described in Table 2. Starting values must be supplied by the user, but are denoted by * here.

Model 1A

```
define regset BetaX;
  var = 1 x1 x2 (z==1) (z==2);

define normal distribution; dim=1; name=u0;
define normal distribution; dim=1; name=eps;

continuous model;
outcome=y;
model = regset BetaX + res(draw=1,ref=u0) + res(draw=_iid,ref=eps);

starting values;

Beta0      T      *
Beta1      T      *
Beta2      T      *
Beta3      T      *
Beta4      T      *
Sigu0     T      *
Sigeps    T      *
;
```

Model 1B

```
define regset BetaX;
  var = 1 x1 x2 (z==1) (z==2) x1*x2;

define normal distribution; dim=1; name=u0;
define normal distribution; dim=1; name=eps;

continuous model;
outcome=y;
model = regset BetaX + res(draw=1,ref=u0) + res(draw=_iid,ref=eps);

starting values;

Beta0      T      *
Beta1      T      *
Beta2      T      *
Beta3      T      *
Beta4      T      *
Beta5      T      *
Sigu0     T      *
Sigeps    T      *
;
```

Model 1C

```
define regset BetaX;
  var = 1 x1 x2 (z==1) (z==2) x1*x2;

define normal distribution; dim=2; name=u0; name=u1;
define normal distribution; dim=1; name=eps;

continuous model;
outcome=y;
model = regset BetaX + res(draw=1,ref=u0)
      + x1*res(draw=1,ref=u1)
      + res(draw=_iid,ref=eps);

starting values;

Beta0      T      *
Beta1      T      *
Beta2      T      *
Beta3      T      *
Beta4      T      *
Beta5      T      *
Sigu0      T      *
Sigul      T      *
Rho_u01    T      *
Sigeps     T      *
;
```

Model 1D

```
define regset BetaX;
  var = 1 x1 x2 (z==1) (z==2) x1*x2;

define normal distribution; dim=2; name=u0; name=u1;
define normal distribution; dim=1; name=eps0;
define normal distribution; dim=1; name=eps1;

continuous model; keep if (x2==0);
outcome=y;
model = regset BetaX + res(draw=1,ref=u0)
      + x1*res(draw=1,ref=u1)
      + res(draw=_iid,ref=eps0)
      ;

continuous model; keep if (x2==1);
outcome=y;
model = regset BetaX + res(draw=1,ref=u0)
      + x1*res(draw=1,ref=u1)
      + res(draw=_iid,ref=eps1)
      ;

starting values;

Beta0      T      *
```

```

Beta1      T      *
Beta2      T      *
Beta3      T      *
Beta4      T      *
Beta5      T      *
Sigu0      T      *
Sigu1      T      *
Rho_u01    T      *
Sigeps0    T      *
Sigeps1    T      *
;

```

Model 2

```

define regset BetaX;
  var = 1;

define normal distribution; dim=1; name=v;
define normal distribution; dim=1; name=u;
define normal distribution; dim=1; name=eps;

continuous model;
outcome=y;
model = regset BetaX + res(draw=1,ref=v) + res(draw=school,ref=u)
      + res(draw=_iid,ref=eps);

starting values;

Beta0      T      *
Sigv       T      *
Sigu       T      *
Sigeps     T      *
;

```


APPENDIX B: Syntax for two-level binary response models

Model 3A

```
define regset BetaX;
  var = 1 x1 x2 x3 x4 x5;

define normal distribution; dim=1; number of integration points=16;
  name=u0;

logit model;
  outcome=y;
  model = regset BetaX + intres(draw=1,ref=u0);

starting values;

Beta0      T      *
Beta1      T      *
Beta2      T      *
Beta3      T      *
Beta4      T      *
Beta5      T      *
Sigu0      T      *
;
```

Model 3B

```
define regset BetaX;
  var = 1 x1 x2 x3 x4 x5;

define normal distribution; dim=2; number of integration points=16;
  name=u0; name=u1;

logit model;
  outcome=y;
  model = regset BetaX + intres(draw=1,ref=u0)
    + x2*intres(draw=1,ref=u1);

starting values;

Beta0      T      *
Beta1      T      *
Beta2      T      *
Beta3      T      *
Beta4      T      *
Beta5      T      *
Sigu0      T      *
Sigu1      T      *
Rho_u01    T      *
;
```

Model 3C

Same as for Model 3B with logit model replaced by probit model.

APPENDIX C: Syntax for two-level Poisson model (M4)

Model specification given in Section 3.4.

```
define regset BetaX;  
  var = 1 x;  
  
define normal distribution; dim=1; number of integration points=16;  
  name=u0;  
  
poisson model;  
  outcome=y;  
  exposure=e;  
  incidence= exp(regset BetaX + intres(draw=region,ref=u0));  
  
starting values;  
  
Beta0      T      *  
Beta1      T      *  
Sigu0      T      *  
;
```

APPENDIX D: Syntax for two-level growth curve models to continuous data

Model 5A

```
define regset BetaX;
  var = 1 x x^2 x^3;

define normal distribution; dim=3; name=u0; name=u1; name=u2;
define normal distribution; dim=1; name=eps;

continuous model;
  outcome=y;
  model = regset BetaX + res(draw=1,ref=u0)
    + x*res(draw=1,ref=u1)
    + x*x*res(draw=1,ref=u2+ res(draw=_iid,ref=eps));

starting values;

Beta0      T      *
Beta1      T      *
Beta2      T      *
Beta3      T      *
sigu0      T      *
sigu1      T      *
sigu2      T      *
Rho_u01    T      *
Rho_u02    T      *
Rho_u12    T      *
Sigeps     T      *
;
```

Model 5B

```
define regset BetaX;
  var = 1 x x^2 x^3;

define normal distribution; dim=3; name=u0; name=u1; name=u2;
define ar(1) distribution;
  timevar (within level 1) =occasion;
  name=eps;

continuous model;
  outcome=y;
  model = regset BetaX + res(draw=1,ref=u0)
    + x*res(draw=1,ref=u1)
    + x*x*res(draw=1,ref=u2)+res(draw=_iid,ref=eps);

starting values;

Beta0      T      *
Beta1      T      *
Beta2      T      *
Beta3      T      *
sigu0      T      *
sigu1      T      *
```

sigu2	T	*
Rho_u01	T	*
Rho_u02	T	*
Rho_u12	T	*
Phi	T	*
Sigeps	T	*
;		

APPENDIX E: Syntax for other selected multilevel models

Model 6: Two-level ordered logit model, with random intercept and unknown thresholds

```
define vector Taus; dim=6;

define normal distribution; dim=1; number of integration points=32;
name=u;

define regset BetaX;
  var = x1 x2 x3;

ordered logit model;
  outcomes=y-1 y;
  thresholds=Taus;
  model = regset BetaX + intres(draw=1, ref=u);

starting values;

Tau1      T      *
Tau2      T      *
Tau3      T      *
Tau4      T      *
Tau5      T      *
Tau6      T      *
Sigu      T      *
Beta1     T      *
Beta2     T      *
Beta3     T      *
;
```

Model 7: Two-level unordered logit model, with random intercept and unknown thresholds

```
define regset BetaX1;
  var=1 x;
define regset BetaX3;
  var=1 x;

define normal distribution; dim=2; number of integration points=32;
name=u1;
name=u3;

multinomial logit model;
  outcome = y;
  model 1 = regset BetaX1 + intres(draw=1, ref=u1);
  model 3 = regset BetaX3 + intres(draw=1, ref=u3);

starting values;
Beta01    T      *
Beta11    T      *
Beta03    T      *
Beta13    T      *
```

```

Sigma1      T      *
Sigma3      T      *
Rho         T      *
;

```

Model 8: Three-level random intercept survival model

```

define spline Dur; nodes = 12 24;

define regressor set BetaX;
  var = 1 x1 x2 x3 x4;
define normal distribution; dim=1; number of integration points=16;
  name = v;
define normal distribution; dim=1; number of integration points=16;
  name = u;

hazard model;
  censor=censor; duration = lower upper;
  model = durspline(origin = 0, ref = Dur) + regset BetaX +
  intres(draw=1,ref=v)+
  intres(draw=womanid,ref=u);

starting values;

Alpha1      T      *
Alpha2      T      *
Alpha3      T      *
Alpha0      T      *
Beta1       T      *
Beta2       T      *
Beta3       T      *
Beta4       T      *
Sigv        T      *
Sigu        T      *
;

```

Model 9: Two-level multiprocess model for a continuous response and an ordered categorical response

```

define vector Taus; dim=2;

define regset BetaX;
  var=1 x;
define regset GammaX;
  var=x;

define normal distribution; dim=1; name=eps1;
define normal distribution; dim=1; name=eps2;
define normal distribution; dim=2;
  name=u1;
  name=u2;

continuous model;
  outcome=y1;
  model=regset BetaX+res(draw=1,ref=u1)+res(draw=id,ref=eps1);

```

```
ordered probit model;  
  outcomes=y2-1 y2;  
  thresholds=Taus;  
  model=regset GammaX+intres(draw=1,ref=u2)+res(draw=_iid,ref=eps2);
```

```
starting values;
```

```
starting values;
```

```
Tau1      T      *  
Tau2      T      *  
Beta0     T      *  
Beta1     T      *  
Gamma1    T      *  
Sigeps1   T      *  
Sigeps2   F      *  
Sigul     T      *  
Sigu2     T      *  
Rhou      T      *  
;
```