Maximum likelihood estimation of generalized linear models with covariate measurement error

Sophia Rabe-Hesketh
Graduate School of Education
University of California
Berkeley, CA

Anders Skrondal
Division of Epidemiology
Norwegian Institute of Public Health, Oslo

Andrew Pickles
School of Epidemiology
and Health Science & CCSR
The University of Manchester

Abstract. Generalized linear models with covariate measurement error can be estimated by maximum likelihood using \texttt{gllamm}, a program that fits a large class of multilevel latent variable models (Rabe-Hesketh, Skrondal, and Pickles 2004b). The program uses adaptive quadrature to evaluate the log-likelihood, producing more reliable results than many other methods (Rabe-Hesketh, Skrondal, and Pickles 2002). For a single covariate measured with error (assuming a classical measurement model), we describe a `wrapper' command \texttt{cme} that calls \texttt{gllamm} to estimate the model. The wrapper makes life easy for the user by accepting a simple syntax and data structure and producing extended and easily interpretable output. The commands for preparing the data and running \texttt{gllamm} can also be obtained from \texttt{cme} and run in a do-file. We first discuss the case where replicate measurements are available and subsequently consider estimation when the measurement error variance is instead assumed known. The latter approach is useful for sensitivity analysis assessing the impact of assuming perfectly measured covariates in generalized linear models. An advantage of using \texttt{gllamm} directly is that the classical covariate measurement error model can be extended in various ways. For instance, we can use nonparametric maximum likelihood estimation (NPMLE) to relax the normality assumption for the true covariate. We can also specify a congeneric measurement model which relaxes the assumption that the repeated measurements are exchangeable replicates by allowing for different measurement scales and error variances.

Keywords: covariate measurement error, measurement model, factor model, congeneric measurement model, adaptive quadrature, nonparametric maximum likelihood, NPMLE, latent class model, empirical Bayes, simulation, wrapper, sensitivity analysis, \texttt{gllamm}, \texttt{cme}

1 Introduction

Covariates are often measured with error, their true values being unobservable or ‘latent’. For simplicity, we assume that this is only the case for one covariate $u_i$, whereas the other covariates $z_i$ are perfectly measured or ‘observed’. The problem then is to estimate a generalized linear model for a response or ‘outcome’ variable $y_i$ with link
function $g(\cdot)$, 
\[ g(\mu_i) = z_i^{(0)} \beta + u_i \beta_u. \] (1)
We will call this model the outcome model (‘disease model’ in epidemiology).

It is well-known that substituting an error-prone measured covariate $w_i$ for the true covariate $u_i$ will generally lead to biased estimates of both $\beta_u$ and $\beta$. We can attempt to correct the bias if further information is available, such as the true covariate values in a ‘validation’ (sub)sample, instrumental variables, replicate measurements (at least in subsample), or knowledge of the measurement properties (e.g. measurement error variance).

In this paper we describe maximum likelihood estimation of generalized linear models with covariate measurement error, making use of either replicate measurements or known measurement error variance. Remarkably, the maximum likelihood approach has received relatively scant attention in the literature, probably because it has not been available in standard software. We hope that the \texttt{cme} (Covariate Measurement Error) command introduced here will remedy this. This ‘wrapper’ command uses the reliable adaptive quadrature method (Rabe-Hesketh et al. 2002) implemented in the general \texttt{gllamm} command but is easier to use than \texttt{gllamm} because of its tailor-made syntax and output. Alternative methods such as regression calibration, instrumental variables, and simulation extrapolation are discussed in the companion papers by Hardin and Carroll (2003) and Hardin et al. (2003a,b,c).

The covariate measurement error model consists of three submodels (e.g. Clayton 1992), the outcome model in (1), a measurement model, specifying the relationship between the true covariate and its measurements, and a true covariate model, specifying a regression of the true covariate on observed covariates. Section 2.1 describes the classical covariate measurement error model. Although the outcome submodel of this classical model is very general, the measurement and true covariate submodels are rather restrictive. One limitation is that a normal distribution is assumed for the true covariate (given the observed covariates), raising concerns about robustness. In Section 2.2 we therefore relax this assumption by using nonparametric maximum likelihood estimation. Another limitation is the implicit assumption of identical measurement properties for the fallible measures of the true covariate. We relax this assumption by introducing the general congeneric measurement model in Section 2.3. Maximum likelihood estimation of the classical model and its extensions using the \texttt{cme} and \texttt{gllamm} commands is described in Section 3.

2 Covariate measurement error models

2.1 Classical model

We first consider the situation where the true covariate has been measured $n_i$ times for unit $i$, giving fallible measures $w_{ij}$, $j = 1, \ldots, n_i$, $n_i \leq k$. Sometimes replicate measurements are available only on a subsample, all other units being measured once. If no replicate measurements are available (on any of the units), at least one model
parameter must be assumed known, a situation we will return to later.

The classical measurement model can be written as

\[ w_{ij} = u_i + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma^2). \] (2)

Here the repeated measurements for a unit have mean equal to the true covariate. The measurement errors \( \epsilon_{ij} \) are independently normally distributed with zero mean and constant measurement error variance \( \sigma^2 \) and are independent of the true covariate \( u_i \). This implies that the measurements are conditionally independent given the true covariate.

To complete specification of the covariate measurement error model, we need to define a true covariate model (‘exposure model’ in epidemiology),

\[ u_i = z_i' \gamma + \zeta_i, \quad \zeta_i \sim N(0, \tau^2), \] (3)

where \( \gamma \) are regression parameters and \( \zeta_i \) disturbances or residuals. The model includes \( z_i \) because it would be unrealistic to assume that the true covariate is independent of the other (observed) covariates in the outcome model.

The joint model, comprising the three submodels in (1) to (3), is illustrated in Figure 1 for the case of a single observed covariate \( z_i \) and up to \( k = 2 \) measurements of the true covariate. Here circles represent latent variables and rectangles observed variables. Long arrows represent linear relations in the linear predictor and short arrows represent residual variability. For the true covariate and measurement models, this residual variability is an additive error term, but for the outcome \( y_i \) it could be, for instance, binomial or Poisson variability depending on the specified distribution. It is apparent from the diagram that the measurements are conditionally independent of the outcome \( y_i \) given the true covariate, a property known as nondifferential measurement error.

We see that the covariate \( z_i \) has an indirect effect \( \gamma_1 \beta_u \) on the outcome mediated by the true covariate in addition to the direct effect \( \beta_u \). The total effect is simply the sum of these effects. This can also be seen by substituting the true covariate model (3) into the outcome model (1), giving the reduced form outcome model

\[
g(\mu_i) = z_i' \beta + (z_i' \gamma + \zeta_i) \beta_u = z_i' (\beta + \gamma \beta_u) + \zeta_i \beta_u.
\]

The reduced form measurement model becomes

\[ w_{ij} = z_i' \gamma + \zeta_i + \epsilon_{ij}. \]

Incidentally, when all direct and indirect effects of covariates are included in the model, it is easier to estimate the reduced form parameters \( \alpha = \beta + \gamma \beta_u \), \( \beta_u \) and \( \gamma \) (e.g. Aitkin and Rocci 2002) from which the parameters of primary interest, the structural parameters \( \beta, \gamma \) and \( \beta_u \), can be derived. However, removing (at least two of) the direct effects
Maximum likelihood

Figure 1: Path diagram with direct and indirect effects of $z$ on $y$

requires nonlinear constraints of the form $\alpha_p = \gamma_p \beta_u$ for the reduced form parameters. Our programs therefore estimate the structural parameters directly.

The model is not identified if there are no replicate measurements, i.e. $k=1$. In this case a parameter constraint is required to estimate the model, for instance setting the measurement error variance $\sigma^2$ to a known non-negative constant, typically an estimate from another study. ‘Transporting’ a parameter estimate this way requires that the parameter does not vary between populations, an assumption that is reasonable for the measurement error variance. By contrast, the reliability depends on the true covariate variance, a characteristic of the population. A major problem with treating unknown parameters as known is that estimation uncertainly is not taken into account. We therefore consider this ‘plugging in’ approach most useful as a sensitivity analysis to investigate how the parameter estimates change for different values of the assumed measurement error variance. Importantly, this allows us to assess the impact of the implicit assumption of perfectly measured covariates in generalized linear models.

2.2 Nonparametric distribution of the true covariate

Instead of assuming a normal distribution for the true covariate $u_i$, or for the residual $\zeta_i$ when there are observed covariates, we can leave the distribution unspecified. The nonparametric maximum likelihood estimator (NPMLE) of the distribution is discrete (Laird 1978; Heckman and Singer 1984) with masses $\pi_c$ at a finite number of locations $\zeta_i = e_c$, $c = 1, \ldots, C$. The number of masses is determined to achieve the largest possible likelihood. Attempts to estimate the model with $C + 1$ masses would result either in one mass having an estimated probability approaching zero or two masses nearly sharing the same estimated location. This model is also sometimes referred to as a semiparametric mixture model.
Our approach to finding the NPMLE (Rabe-Hesketh, Pickles, and Skrondal 2003) is to start with a small number of masses and estimate both locations and masses by maximum likelihood, along with all the other parameters. We then use the idea of directional derivative, referred to as gateaux derivative in Heckman and Singer (1984), to decide if another mass should be added. Here, a very small new mass is moved across a wide range of locations keeping all parameters at their ‘current’ estimates. If the log-likelihood increases at any location, a new mass is introduced. It is helpful to use the location resulting in the greatest increase in log-likelihood as a starting value for the new masspoint. NPMLE for covariate measurement error models has been discussed by Roeder et al. (1996), Schafer (2001), Aitkin and Rocci (2002), and Rabe-Hesketh et al. (2003). General references include Lindsay (1995) and Böhning (2000).

2.3 Congeneric measurement model

The classical measurement error model in (2) assumes that the fallible measures have the same mean (no relative bias) and measurement error variance. This is reasonable if the measures are essentially exchangeable replicates. However, if the measurements are separated in time there may be a ‘drift’ in the mean measurement (e.g. Carroll, Ruppert, and Stefanski 1995). More importantly, if the fallible measures are obtained by different methods such as different instruments or different raters, we should allow the measures to have different means, scales, and measurement error variances. We can therefore extend the classical measurement error model to a congeneric measurement model (Jöreskog 1971) or one-factor model of the form

$$w_{ij} = \delta_j + \lambda_j u_i + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma_j^2), \quad (4)$$

where $\delta_1 = 0$ and $\lambda_1 = 1$ for identification. Here $\delta_j$ represents the bias of measure $j$, $\lambda_j$ the scale or factor loading and $\sigma_j^2$ the measurement error variance.

The following hierarchy of increasingly restrictive versions of the congeneric measurement models is often considered (Jöreskog 1971; Lord and Novick 1968):

1. **Essentially tau-equivalent model.**
   The true scores have identical scales, i.e. all factor loadings are set equal across raters or instruments $j$, $\lambda_j = \lambda = 1$.

2. **Tau-equivalent model.**
   The true scores have identical locations and scales, i.e. the intercepts are also set equal, $\delta_j = \delta = 0$.

3. **Parallel or classical measurement model in (2).**
   The measurement properties of raters or instruments are identical, i.e. the measurement error variances are also set equal, $\sigma_j^2 = \sigma^2$.

Returning to the congeneric measurement model, we interpret $\lambda_j u_i$ as the true score for unit $i$ expressed in the scale of instrument or rater $j$. The measures can have different
Maximum likelihood

Reliabilities defined as the proportion of the total variance that is due to variability between the units’ true scores (true covariate values)

\[ R_j = \frac{\lambda_j^2 \tau^2}{\lambda_j^2 \tau^2 + \sigma_j^2} \]  

If the true covariate model includes covariates, the reliability is interpreted as conditional on these covariates. This conditional reliability could be substantially lower than the unconditional one if the covariates explain a substantial portion of the between-subject variance in the true covariate. We refer to Dunn (1989, 1992) and Dunn and Roberts (1999) for useful treatments of reliability and measurement models.

The congeneric measurement model can be further generalized by specifying a generalized linear factor model as

\[ g(\mu_{ij}) = \delta_j + \lambda_j u_i. \]

If the responses are dichotomous, this is a two-parameter item-response model (Birnbaum 1968). If the true covariate is categorical, a latent class model (e.g. Clogg 1995) can be specified where \( u_i \) is categorical and the true covariate model becomes a multinomial logit model. We refer to Bartholomew and Knott (1999) and Skrondal and Rabe-Hesketh (2004) for theory and applications of generalized linear models with latent variables.

3 Estimation using cme and gllamm

3.1 Classical covariate measurement error model

We will illustrate covariate measurement error modeling using data from Morris et al. (1977). This study investigated the relationship between diet and coronary heart disease (CHD). At the time of recruitment, 337 middle-aged men weighed their food intake over a 7-day period, allowing food constituents to be derived. A subsample of 76 of the men repeated this 6 months later, and all the men were then followed up for CHD. We will estimate the effect of dietary fibre intake on CHD, controlling for occupation. The relevant variables are:

- \text{chd}: dummy for coronary heart disease, CHD (1: present, 0: absent)
- \text{fibre1} & \text{fibre2}: dietary fibre intake (grams/day) at first and second occasions
- \text{bus}: dummy for man working for London Transport (1: London Transport, 0: bank staff)
- \text{id}: subject identifier

Since fibre intake has a skewed distribution, we will analyze log-fibre, \text{l fib}re1 and \text{l fib}re2, created using \text{gen l fib}re1 = \text{ln(fibre1)} and similarly for \text{l fib}re2.
We specify a classical measurement error model for log-fibre of the form of (2) and a logistic regression model for CHD with observed covariate bus (corresponding to $z_i$) and true log-fibre intake, $u_i$ as in (1), where $g(\cdot)$ is the logit link. True log-fibre intake is regressed on bus as in (3).

**cme command**

The classical covariate measurement error model can be estimated using the command **cme** which is a wrapper for **gllamm**. The syntax is as for **glm** except that the measurement model for the true covariate is specified as **(label: measure1 measure2 ... measurek)**. Note that the variables **measure1 measure2 ... measurek** can contain missing values if not all units have been measured on all occasions as is the case here. All subjects with at least one nonmissing value will contribute to the analysis. For the present example, we obtain:

```
.cme chd bus (lfib: lfibre1 lfibre2), l(logit) f(binom) nolog
```

```
gllamm covariate measurement error model
   No. of obs = 333
   log likelihood = -186.93044

  OUTCOME MODEL

                          chd    Coef.  Std. Err.    z   P>|z|   [95% Conf. Interval]
  chd                bus   -.1898458   .3396767   -0.56  0.576   -.8555999   .4759083
                        lfib   -1.962491   .7266261   -2.70  0.007   -3.386652   -.5383299
                        _cons  3.642282  2.026281    1.80  0.072   -.3291551  7.613719

  TRUE COVARIATE MODEL

                          lfib    Coef.  Std. Err.    z   P>|z|   [95% Conf. Interval]
  lfib                bus   -.1208521  .0327976   -3.68  0.000   -.1851343   -.05657
                        _cons  2.863706  .0238865  119.89  0.000  2.819551  2.907862
  res. var.          .0701939  .0075809   0.00    0.99    .0561219  .0858385

  MEASUREMENT MODEL

                          error var.  Coef.  Std. Err.  [95% Conf. Interval]
  error var.        .0217377  .0035285   .0158142  .0298801
   reliability       .7635446  .0413050   .6755186   .8366307
```

There are three tables in the output, the generalized linear outcome model of primary interest, followed by the true covariate model and the measurement model. In the outcome model, the coefficient $-1.96$ of **lfib** represents the estimated effect $\tilde{\beta}_u$ of true log-fibre. The corresponding odds ratio, $\exp(-1.956183) = 0.14$, can be obtained along
with the odds ratio for bus using the eform option. This extremely large estimated protective effect of log-fibre is probably due to omitting important confounding variables such as exercise which is protective of heart disease and increases food intake, including fibre. Occupation does not appear to have an important direct effect on CHD. In the true covariate model, we see that London transport staff consume less fibre than bank staff. The residual variance (i.e. the within-occupation variance) of true log-fibre is estimated as $\hat{\tau}^2 = 0.07$. In the measurement model, the measurement error variance is estimated as $\hat{\sigma}^2 = 0.02$. The estimated reliability, conditional on occupation, is $R = \hat{\tau}^2 / (\hat{\tau}^2 + \hat{\sigma}^2) = 0.76$. These estimates are shown under ‘Classical model’ in Table 1. Note that by default 8-point adaptive quadrature was used to obtain these estimates. To ensure that integration is accurate (see Rabe-Hesketh, Skrondal, and Pickles 2002), these estimates should be compared with estimates using a larger number of points, for instance 12 using the nip(12) option.

We can use the total and indirect options to obtain total and indirect effects of bus, in terms of odds ratios if the eform option is used:

```
cme, eform indirect total
gllamm covariate measurement error model
log likelihood = -186.93044
```

<table>
<thead>
<tr>
<th>OUTCOME MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>chd</td>
</tr>
<tr>
<td>chd</td>
</tr>
<tr>
<td>chd</td>
</tr>
</tbody>
</table>

Table 1: Estimates of different models for diet and CHD data

<table>
<thead>
<tr>
<th></th>
<th>Classical model</th>
<th>Only indirect effect</th>
<th>NPMLE (dir. &amp; indir.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Est (SE)</td>
<td>Est (SE)</td>
<td>Est (SE)</td>
</tr>
<tr>
<td>Outcome model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta_0$ [Cons]</td>
<td>-1.977 (0.257)</td>
<td>-2.069 (0.202)</td>
<td>-1.972 (0.257)</td>
</tr>
<tr>
<td>$\beta_1$ [Bus]</td>
<td>-0.190 (0.340)</td>
<td>-0.183 (0.338)</td>
<td></td>
</tr>
<tr>
<td>$\beta_u$</td>
<td>-1.956 (0.726)</td>
<td>-1.862 (0.702)</td>
<td>-1.928 (0.702)</td>
</tr>
<tr>
<td>True covariate model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu$ [Cons]</td>
<td>2.864 (0.024)</td>
<td>2.863 (0.024)</td>
<td>2.861 (0.023)</td>
</tr>
<tr>
<td>$\gamma_1$ [Bus]</td>
<td>-0.120 (0.033)</td>
<td>-0.120 (0.033)</td>
<td>-0.115 (0.033)</td>
</tr>
<tr>
<td>$\tau^2$</td>
<td>0.070 (0.008)</td>
<td>0.070 (0.008)</td>
<td>0.073 (–)</td>
</tr>
<tr>
<td>Measurement model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>0.022 (0.004)</td>
<td>0.022 (0.004)</td>
<td>0.019 (0.003)</td>
</tr>
<tr>
<td>Log-likelihood</td>
<td>-186.93</td>
<td>-187.09</td>
<td>-177.87</td>
</tr>
</tbody>
</table>
The indirect effect of reduced fibre intake among transport staff is to increase the odds of CHD (estimated odds ratio of 1.27). The protective direct effect of being transport staff (estimated odds ratio of 0.83) counteracts this, giving a negligible total effect (estimated odds ratio of $1.27 \times 0.83 = 1.05$). However, there is not much evidence for a direct effect and we may therefore wish to estimate a more parsimonious model omitting this effect (possibly to perform a likelihood ratio test):

```
cme chd (lfib: lfibre1 lfibre2), l(logit) f(binom) tcovmod(bus)
```

Here `bus` has been omitted after `chd` whereas the `tcovmod()` option specifies `bus` as a covariate for the true covariate model. The estimates are shown under ‘Only indirect effect’ in Table 1. The total effect of `bus` is now an odds ratio of 1.25 with a 95% confidence interval from 1.02 to 1.53.

We can estimate these models with the measurement error variance $\sigma^2$ constrained at a particular value using the `mevar(#)` option. As discussed in Section 2.1, such a constraint is necessary for model identification if there are no replicate measurements. We will pretend that we only had the first log-fibre measurement `lfibre1` and explore how the estimate of $\beta_u$ changes for a range of values of $\sigma^2$ from 0 to 0.05:

```
.gen beta_u = .
gen variance = (_n-1)/200 in 1/11
.glm chd bus lfibre1, l(logit) f(binom)
.replace beta_u = [chd]lfibre1 in 1
.forvalues s=2/11{
   local var=(s-1)/200
   cme chd bus (lfib: lfibre1), l(logit) f(binom) mevar(`var')
   replace beta_u = [chd]lfib in `s'
}
twoway connect beta_u variance, ytitle(Log odds ratio for true log-fiber) /*
   xtitle(Measurement error variance)
```

Here the estimate for $\sigma^2 = 0$ is obtained using `glm`, whereas `cme` is used within a `forvalues` loop for variances $\sigma^2(s) = (s-1)/200$, $s = 2, \ldots, 11$. The corresponding estimates $\hat{\beta}_u(s)$ are accessed using the ‘equation name’ `chd` (outcome variable) and the ‘column name’ `lfib` (the label for the true covariate specified in the `cme` command). The resulting graph of $\hat{\beta}_u(s)$ versus $\sigma^2(s) = (s-1)/200$ is shown in Figure 2. Such a graph is useful for assessing the sensitivity of the parameter estimates to the implicit
assumption in generalized linear models that there is no covariate measurement error.

The \texttt{cme} command allows a wide range of generalized linear models for the outcome as well as ordinal logit and probit models (see Appendix A for all the options), but the classical measurement error model and a normally distributed true covariate are always assumed. To relax these assumptions, \texttt{gllamm} would have to be used directly. As we will see, the \texttt{commands} option of \texttt{cme} is very useful for this purpose since it creates commands for estimating the model in \texttt{gllamm}, including all the necessary data manipulation.

**GLLAMM framework**

Here we briefly describe the GLLAMM (Generalized linear latent and mixed model) framework confining ourselves to the special case of two-level models with a single latent variable (the general framework is described in Rabe-Hesketh et al. (2004b) and Skrondal and Rabe-Hesketh (2004)). All response variables, the outcome $y_i$ (CHD) and the repeated measurements $w_{ij}$ of the true covariate are stacked in a single response vector $y$ with elements indexed $ij$, where $j = 1$ if the element corresponds to the outcome for unit $i$ and $j = 2, \cdots, k + 1$ for the fallible measurements. The linear predictor then has the form

$$g_{ij}(\mu_{ij}) = x_{ij}' \beta + u_i v_{ij}' \lambda, \quad \lambda_1 = 1,$$

where $x_{ij}$ and $v_{ij}$ are variables, $\beta$ and $\lambda$ parameters and $u_i$ is a latent variable. The $ij$ subscript for the link implies that different link functions can apply to different responses, for example, the logit link for the outcome and the identity link for the fallible measurements. The fixed part $x_{ij}' \beta$ allows inclusion of observed covariates in the model. To include the observed covariates $z_i$ in the outcome model only, we must multiply these covariates by a dummy variable for CHD

$$d_{ij} = \begin{cases} 1 & \text{if } j = 1 \\ 0 & \text{if } j > 1 \end{cases}.$$
The variables $v_{ij}$ are typically dummy variables, allowing the regressions for different response variables on the latent variable $u_i$ to have different regression coefficients $\lambda_p$. In the present example, using the dummy variables $v_{1ij} = d_{mij}$,

$$d_{mij} = 1 - d_{1ij} = \begin{cases} 
0 & \text{if } j = 1 \\
1 & \text{if } j > 1
\end{cases},$$

and $v_{2ij} = d_{1ij}$ results in the following model

$$g_{ij}(\mu_{ij}) = d_{1ij}z'_{ij}\beta + u_i(\lambda_1d_{mij} + \lambda_2d_{1ij}), \quad \lambda_1 = 1$$

$$\begin{cases} 
z'_{ij}\beta + u_i\lambda_2 & \text{if } j = 1 \\
u_i & \text{if } j > 1
\end{cases}.$$  \hspace{1cm} (6)

Here $\lambda_2$ corresponds to $\beta_2$ in (1). Note that we only required a single dummy variable for both log-fibre measurements since the regressions on $u_i$ share the same regression coefficient $\lambda_1 = 1$.

Finally we need to specify the true covariate model, called the structural model in the GLLAMM framework, as

$$u_i = z'\gamma + \zeta_i, \quad \zeta_i \sim N(0, \tau^2).$$

Note that we assume here that the first observed covariate $z_{1i}$ is a constant so that the true covariate model and outcome model both contain intercepts, $\gamma_0$ and $\beta_0$, respectively. In Section 3.3 we will omit $\gamma_0$ from the true covariate model and include a constant in the measurement model instead, giving an equivalent model but with a different estimate of $\beta_0$.

Note that GLLAMMs allow more general structural models, regressing latent variables at different levels on same or higher-level latent and observed variables.

**gllamm command**

We will now show how all commands necessary for estimating the classical covariate measurement error model for the diet and CHD example can be generated using the commands option in cme (we have added the line numbers for ease of reference):

```
cme chd bus (lfib: lfibre1 lfibre2), l(logit) f(binom) nolog commands
```

```
1 * starting values
2 matrix startv = (-.1474, 2.458, -1.458, -1.54, .3555, -.1237, 2.867)
3 gen _id = _n
4 * collapse data to make gllamm faster
5 gen _one = 1
6 collapse (sum) _wt2 = _one, by(chd lfibre1 lfibre2 _id bus bus)
7 * give response variable and replicate measurements same prefix
8 rename chd _r1
9 rename lfibre1 _r2
10 rename lfibre2 _r3
```
11 * reshape data to long
12 reshape long _r, i(_id) j(_var)
13 * create dummy variables and interactions
14 gen byte cons = 1
15 gen byte _d1 = _var == 1
16 gen byte _dmeas = 1-_d1
17 gen _type = _d1 + 2*_dmeas /* response type */
18 gen _bus_d1 = bus*_d1
19 * define equations
20 eq load: _dmeas _d1
21 eq f1: bus cons
22 * call gllamm
23 gllamm _r _bus_d1 _d1, /*
24 */ i(_id) nocons eqs(load) link(logit ident) family(binom gauss) /*
25 */ lv(_type) fv(_type) geqs(f1) from(startv) copy adapt /*
26 */ weightf(_wt) nolog

Note that running this do-file will change the data irreversibly so users may wish to save their data first. The variables generated by the do-file start with `1 to distinguish them from other variables in the dataset. If the variables _id or _one already exist, the do-file will stop running and the user has to make changes to the do-file or data. In line 2, the starting values generated by cme are placed into a matrix startv for later use. In lines 5 and 6, the data are collapsed to reduce the number of observations if several units share the same values on all variables contributing to the analysis (not the case in this example). Since the two repeated log-fibre measurements and CHD are all response variables, these variables need to be stacked into a single variable _r. In lines 8 to 10, the variables are therefore first renamed to _r1, _r2 and _r3 and in line 12 the reshape command is used to stack these responses into _r and create a variable _var taking on the values 1,2,3 for responses originally in _r1, _r2 and _r3, respectively.

A constant is generated in line 14. Two response models will need to be specified, one for CHD and one for log-fibre. Lines 15 to 17 therefore create dummy variables _d1 (d_{1ij}) for CHD and _dmeas (d_{mij} = 1 - d_{1ij}) for log-fibre and a variable _type taking on values 1 and 2 for these two response-types, respectively. In line 18, the explanatory variable bus in the CHD model is multiplied by _d1 to pick out the values of bus for responses corresponding to CHD.

The data manipulation is now complete and the model can be specified. In line 20, the linear combination of variables v_{ij}'\lambda multiplying the latent variable is specified using the equation command eq load: _dmeas _d1. To include the effect of bus on true log-fibre, line 21 defines an equation for the structural model. Since there could be several latent variables, the second character of the equation name must be a number, here 1 for the first (and only) latent variable. In the gllamm command, the response variable and the variables for the fixed part x_{ij}'\beta are first listed in line 23. In line 24, the I() option declares _id as the unit-identifier i and the nocons option switches off the overall constant. The eqs() option passes the equation for v_{ij}'\lambda to gllamm. The link() and family() options list the links and families needed. In line 25, lv() and fv() specify that the first link and family should apply when _type equals 1 and the
second when it equals 2. The equation for the structural model for the true covariate is passed to \texttt{gllamm} using the \texttt{geqs()} option. The \texttt{from(startv)} and \texttt{copy} options specify that starting values are in the matrix \texttt{startv} and should be copied in the order in which they appear (not according to column and equation names). The \texttt{adapt} option causes \texttt{gllamm} to use adaptive quadrature. In line 16 \texttt{weightf(wt)} declares that unit-specific frequency weights can be found in \texttt{wt2} whereas any response-specific weights would be in \texttt{wt1}. Since \texttt{wt1} does not exist, the latter are assumed to be 1. For further explanations of these options see \texttt{help gllamm}, the \texttt{gllamm} manual (Rabe-Hesketh, Pickles, and Skrondal 2001a) and \texttt{gllamm} book (Rabe-Hesketh, Pickles, and Skrondal 2004a).

Running this do-file gives the following \texttt{gllamm} output:

\begin{verbatim}
number of level 1 units = 742
number of level 2 units = 333
Condition Number = 165.57248
gllamm model
log likelihood = -186.93043

. \_R | Coef.  Std. Err.  z  P>|z|  [95% Conf. Interval]
-----------------|-----------------------------|-------------|-----------------------------|
  _bus_d1 | -.1891339  .3396477  -0.56  0.578  -.8548312 .4765634
  _d1    |  3.631528   2.025339   1.79  0.073  -.3380629 7.601119

Variance at level 1
---------------------------------------------------------------------
.02172779 (.00352547)

Variances and covariances of random effects
---------------------------------------------------------------------
***level 2 (_id)

var(1):  .07022409 (.007582)

loadings for random effect 1
  _dmeas: 1 (fixed)
  _d1:  -.1986744 (.72624942)

Regressions of latent variables on covariates
---------------------------------------------------------------------

random effect 1 has 2 covariates:
  bus:  -.12085336 (.03280164)
  cons:  2.8637094 (.0238897)
---------------------------------------------------------------------
\end{verbatim}

Here $\zeta_i$ (and $u_i$) are referred to as ‘random effects’. The estimated residual variance $\tau^2$ and ‘loadings’ $\lambda$ are listed under ‘Variances and covariances of random effects’.
The coefficient $\beta_0 \equiv \lambda_2$ is therefore somewhat hidden, whereas the mean of log-fibre $\mu$ is presented in the initial regression table as the regression coefficient of _dmeas_. The measurement error variance is referred to as ‘Variance at level 1’ since it represents the variance between different ‘level-1’ units $ij$ within the same ‘level-2’ unit $i$. The estimated effect $\gamma_1$ of bus on true log-fibre is listed under ‘Regressions of latent variables on covariates’.

An important advantage of understanding how **gllamm** works is that many of the assumptions of the classical covariate measurement error model can be relaxed and the model extended in many ways, some of which will be discussed in the following sections. Another advantage is that we can make use of **gllamm**’s post-estimation commands to obtain predictions and simulate from the model. For example, to predict true exposure by empirical Bayes, use

```
gllapred truefib, u /* predict $\zeta_i$ */
.replace truefib = truefib + _r_dmeas /* predict $u_i = \mu + \zeta_i$ */
```

To simulate responses from the model (including measurements for the second occasion that were missing), use

```
gllasim resp, fsample
```

### 3.2 Nonparametric true covariate distribution

We now relax the normality assumption for true log-fibre intake on which the previous analyses have relied. Assuming the data are still as we left them in the previous section, we can estimate a discrete true covariate distribution with three masses using **gllamm** with the `ip(f)` and `nip(3)` options:

```
.matrix a=e(b)
.local k = e(k)
.local ll = e(ll)
gllamm _r _bus_d1 _d1, /*
i(_id) nocons eqs(load) link(logit ident) family(binom gauss) /*
   i(_id) nocons eqs(load) link(logit ident) family(binom gauss) /*
   eq load: _dmeas _d1 
   eq f1: _bus cons 
gllamm _r _bus_d1 _d1, /*
   */ i(_id) nocons eqs(load) link(logit ident) family(binom gauss) /*
   */ f(v_type) l(v_type) geqs(f1) ip(f) nip(3)
```

(output not shown).

To see if a fourth mass can be added, we next specify `nip(4)` and use the `gateaux()` and `lf0()` options. In the `gateaux()` option we specify the minimum, maximum and number of steps for the search for a new mass location. In the `lf0()` option, we specify the number of parameters and log-likelihood of the three-mass model. The latter is necessary so that **gllamm** can check whether the log-likelihood increases when a tiny fourth mass is placed at any of the locations tried in the search. The necessary commands are:

```
.matrix a=e(b)
.local k = e(k)
.local ll = e(ll)
gllamm _r _bus_d1 _d1, /*
   */ i(_id) nocons eqs(load) link(logit ident) family(binom gauss) /*
   */ i(_id) nocons eqs(load) link(logit ident) family(binom gauss) /*
   /*
```

(output not shown).
(output not shown). At the end of the search, *gllamm* prints

\[
\text{maximum gateaux derivative is } 0.59762692
\]

and starts estimating the four-mass model using the location with the greatest increase in log-likelihood of 0.60 as the starting value for the new location. Increasing the number of masses this way one by one, the maximum gateaux derivative remains positive until the 8-mass solution is reached, giving the following output:

- number of level 1 units = 742
- number of level 2 units = 333
- Condition Number = 180.2991

<table>
<thead>
<tr>
<th>gllamm model</th>
<th>Number of obs = 333</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log likelihood = -177.87417</td>
<td>LR chi2(2) = 0.79</td>
</tr>
<tr>
<td>Prob &gt; chi2 = 0.6751</td>
<td></td>
</tr>
</tbody>
</table>

|    | Coef. | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|----|-------|-----------|-------|------|----------------------|
| _bus_d1 | -0.1832444 | 0.3382888 | -0.54 | 0.588 | -0.8462783 | 0.4797894 |
| _d1   | 3.543075  | 1.958057  | 1.81  | 0.070 | -0.2946467 | 7.380796  |

Variance at level 1

- .01902727 (.00306925)

Probabilities and locations of random effects

---

***level 2 (_id)***

loc1: -.74544, -.2587, .68476, 1.0856, -.53297, .20197, .01391

var(1): .07299486

loadings for random effect 1

_dmeas: 1 (fixed)

_prob: 0.0216, 0.2264, 0.0982, 0.0146, 0.0062, 0.0341, 0.1515, 0.4474

Regressions of latent variables on covariates

---

random effect 1 has 2 covariates:

_bus: -.11507198 (.03271307)

_cons: 2.8606632 (.2598858)

---

Up to ‘Probabilities and locations of random effects’ and from ‘Regressions of latent variables on covariates’, the output is the same as for normally dis-
tributed true covariate (see previous section). In between, the estimated locations $\hat{c}_c$, $c = 1, \ldots, 8$ of the eight masses are listed to the right of loc1:, followed by the variance of the estimated discrete distribution to the right of var(1):. This variance is not a model parameter but is derived from the estimated locations and probabilities using

$$\text{var}(1) = \sum_{c=1}^{8} \hat{\pi}_c \hat{c}_c^2,$$

since the mean of the discrete distribution is set to 0, $\sum_{c=1}^{8} \hat{\pi}_c \hat{c}_c = 0$ (only 7 locations were independently estimated). The estimated factor loading $\hat{\beta}_a$ is then given as for a continuous true covariate, followed by the estimated probabilities $\hat{\pi}_c$, $c = 1, \ldots, 8$ to the right of prob: (to obtain the corresponding log-odds parameters with their standard errors, as well as standard errors for the location parameters, use \texttt{gllamm}, allc.) The parameter estimates (except for the locations and probabilities) are given under ‘NPMLE’ in Table 1. It is remarkable how close the estimates are to those obtained by assuming a normally distributed true covariate (see ‘Direct and indirect effects’ in Table 1).

The discrete distribution can be displayed graphically using the following sequence of commands:

```
. * save locations and log probabilities as variables
  . matrix locs = e(zlcs2)'
  . matrix lp = e(zps2)'
  . svmat locs
  . svmat lp
  . * calculate probabilities
  . gen p = exp(lp1)
  . * plot masses
  . twoway (dropline p locs1), xtitle(Location) ytitle(Probability)
```

giving the graph in the left panel of Figure 3. Here we have used matrices of locations and log-probabilities stored in the first (and only) rows of \texttt{zlcs2} and \texttt{zps2}, respectively. After transposing these matrices so that the estimates are in the first (and only) columns, we place the values into new variables \texttt{locs1} and \texttt{lp1} using the \texttt{svmat} command. We then convert the log-probabilities to probabilities \texttt{p} and plot the masses using the \texttt{twoway dropline} command.

It is also useful to plot the corresponding cumulative distribution together with that of the normal distribution with zero mean and variance $\hat{\sigma}^2 = 0.07019772$ estimated in Section 3.1:

```
. * set end-points for cumulative probabilities
  . replace locs1 = -1 in 9
  . replace p = 0 in 9
  . replace locs1 = 1.1 in 10
  . replace p = 0 in 10
  . * calculate cumulative probabilities
  . sort locs1
```

. gen cump = sum(p)
. * plot cumulative probabilities
. twoway (line cump locs1, connect(stairstep)) /*
>     */ (function y=norm(x/sqrt(.07019772)), range(-1 1)), /*
>     */ xtitle(Location) ytitle(Cumulative probability) /*
>     */ legend(order(1 "NPMLE" 2 "Normal"))

Figure 3: Nonparametric true covariate distribution

giving the graph in the right panel of Figure 3. Here we first create two artificial masses at -1 and 1.1 with zero probabilities so that the cumulative distribution will start at 0 and finish at 1. We then sort the locations and calculate the cumulative probabilities using the sum() function. The twoway line command with the connect(stairstep) option creates the stair-shaped plot for the discrete cumulative distribution and the twoway function command combined with the norm() function produces a smooth curve for the normal cumulative distribution function with the required variance.

The estimated discrete distribution resembles the normal distribution. This may be the reason why the NPML estimates so closely resemble those assuming normality. In Rabe-Hesketh et al. (2003), raw fibre was also analyzed, giving a skewed nonparametric distribution for true fibre and greater differences between the NPML estimates and those assuming normality. Rabe-Hesketh et al. (2003) also carried out simulations to investigate the performance of parameter estimates and empirical Bayes predictions for both NPMLE and assuming normality when the true covariate distribution is either normal or skewed.

It is worth noting that a syntax very similar to that used in this section would be used to specify a latent class model with two classes when the true covariate as well as the measurements are dichotomous:

. gllamm _r _bus_d1 _d1, i(_id) nocons eqs(load) /*
>     */ link(logit) family(binom) peqs(f1) ip(f) nip(2)
3.3 Congeneric measurement model

Returning to a normally distributed true covariate, we now extend the measurement model to the congeneric measurement model in (4) and estimate this model using *gllamm*. However, the congeneric model is not identified with only two measurements of log-fibre and we will therefore simulate data resembling the CHD data but with three measures of log-fibre. Defining dummy variables $d_{2ij}$ to $d_{4ij}$ for the measures of log-fibre, the model can be written as

$$
g(\mu_{ij}) = d_{1ij}\beta_0 + d_{mij}\mu + d_{3ij}\alpha_2 + d_{4ij}\alpha_3 + u_i(\lambda_1d_{2ij} + \lambda_2d_{3ij} + \lambda_3d_{4ij} + \lambda_4d_{1ij}), \quad \lambda_1 = 1
$$

$$
= \begin{cases} 
\beta_0 + u_i\lambda_4 & \text{if } j = 1 \text{ (CHD)} \\
\mu + u_i & \text{if } j = 2 \text{ (measure 1)} \\
\mu + \alpha_2 + u_i\lambda_2 & \text{if } j = 3 \text{ (measure 2)} \\
\mu + \alpha_3 + u_i\lambda_3 & \text{if } j = 4 \text{ (measure 3)} 
\end{cases}
$$

Note that a constant $\mu$ is now included in the measurement model instead of using $\gamma_0$ in the true covariate model. (When no starting values are provided to *gllamm*, it is advisable to include constants in the response model instead of the structural model.)

To allow for different measurement error variances $\sigma_j^2$, we model the log of the ‘level 1’ standard deviation $s_{ij}$ as

$$
\ln(s_{ij}) = \delta_1d_{2ij} + \delta_2d_{3ij} + \delta_3d_{4ij}, \quad \sigma_j^2 = \exp(2\omega_j).
$$

To simulate the data, we first create new observations for the third measurement of log-fibre by replicating the records representing the second measurement (_var=3)

```
. expand 2 if _var==3
```

We need separate dummy variables d2, d3 and d4 for the three measures:

```
. qui tab _var, gen(d) 
. bysort _id _var: gen d4 = _n==2 
. replace d3=0 if d4==1
```

The first command only generates three dummy variables d1 to d3 since _var equals 3 for both the second and third measures. Each subject therefore has two observations with d3=1. In the second and third commands we set d4 to 1 and d3 to 0 for one of these observations. We can now specify the model we wish to simulate from using the *gllamm* command, preventing *gllamm* from estimating the model using the *eval* option which instructs *gllamm* to merely evaluate the log-likelihood.

First specify the linear predictor $d_{2ij} + \lambda_2d_{3ij} + \lambda_3d_{4ij} + \lambda_4d_{1ij}$ multiplying $u_i$ using

```
. eq load: d2 d3 d4 d1
```

where d2 is specified first since the first factor loading $\lambda_1$ equals 1. Then specify the linear predictor $\delta_1d_{2ij} + \delta_2d_{3ij} + \delta_3d_{4ij}$ for the log standard deviation using
The latter is passed to `gllamm` using the `s()` option:

```stata
. matrix a=J(1,12,0) /* matrix with 12 columns, all elements equal to 0 */
. qui gllamm r d1 _dmeas d3 d4, /*
> */ i(_id) nocons eqs(load) link(logit ident) family(binom gauss) /*
> */ fv(_type) lv(_type) geqs(u1) s(het) from(a) copy eval
```

Although no model has not been estimated, ‘estimates’ have been stored and the parameter matrix with values as specified in the `from()` option can be obtained using

```stata
. matrix a=e(b)
. matrix list a
```

We can now specify the parameter values of the model from which we wish to simulate in the order required by the simulation command `gllasim` (same order as for `gllamm`):

```stata
. * Intercept for CHD: \( \beta_0 \)
. matrix a[1,1] = -2
. * Mean of measure 1: \( \mu \)
. matrix a[1,2] = 3
. * Bias parameters: \( \alpha_j \)
. matrix a[1,3] = 1 /* measure 2 */
. matrix a[1,4] = -1 /* measure 3 */
. * Log(sd) parameters: \( \delta_j \)
. matrix a[1,5] = -2 /* measure 1 */
. matrix a[1,6] = -2 /* measure 2 */
. matrix a[1,7] = -1 /* measure 3 */
. * Factor loadings: \( \lambda_j \)
. matrix a[1,8] = 1.5 /* measure 2 */
. matrix a[1,9] = 2 /* measure 3 */
. matrix a[1,10] = -2 /* \( \beta_u \) */
. * True covariate sd: \( \tau \)
. matrix a[1,11] = 0.3
. * Effect of bus on true covariate: \( \gamma_1 \)
. matrix a[1,12] = -0.2
```

We now set the random number seed (to allow replication) and simulate responses from the model specified in the previous `gllamm` command but with parameters in `a` and using the full sample instead of the previous estimation sample:

```stata
. set seed 131123
. gllasim r, from(a) fsample
```

The parameters can then be estimated using the same `gllamm` command as before:
. gllamm r_d1 _dmeas d3 d4, /*
> */ i(_id) nocons eqs(load) link(logit ident) family(binom gauss) /*
> */ fv(_type) lv(_type) geqs(u1) s(het) adapt

Running adaptive quadrature
Iteration 1:  log likelihood = -677.30613
Iteration 2:  log likelihood = -458.65378
Iteration 3:  log likelihood = -419.96371
Iteration 4:  log likelihood = -410.83011
Iteration 5:  log likelihood = -410.55348
Iteration 6:  log likelihood = -410.55324

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:  log likelihood = -410.55324
Iteration 1:  log likelihood = -410.55324

number of level 1 units = 1332
number of level 2 units = 333

Condition Number = 46.814881

gllamm model
log likelihood = -410.55324

|       | Coef.  | Std. Err. | z    | P>|z|   | [95% Conf. Interval] |
|-------|--------|-----------|------|-------|----------------------|
|  _d1  | -1.997545 | .2007127  | -9.95 | 0.000 | -2.390935 -1.604155  |
|  _dmeas | 3.043896 | .026312  | 115.68 | 0.000 | 2.992325 3.095467   |
|   d3  | 1.0229   | .0176837  | 57.84 | 0.000 | .9882403 1.057559   |
|   d4  | -.9608751 | .0352518 | -27.26 | 0.000 | -1.029967 -.8917829 |

Variance at level 1

---

equation for log standard deviaton:

d2: -1.9827443 (.06099411)
d3: -1.8834334 (.09502256)
d4: -1.0093654 (.04794215)

Variances and covariances of random effects

***level 2 (_id)

<table>
<thead>
<tr>
<th></th>
<th>var(1): .09123715 (.00840143)</th>
</tr>
</thead>
<tbody>
<tr>
<td>d2: 1 (fixed)</td>
<td>1.5103875 (.04910599)</td>
</tr>
<tr>
<td>d3: 2.0653678 (.08170733)</td>
<td>-1.0363324 (.51209769)</td>
</tr>
</tbody>
</table>

Regressions of latent variables on covariates

---
random effect 1 has 1 covariates:
bus: -.24193129 (.03478306)

The estimates are quite close to the true values and stay the same (to about four decimal places) when 12-point quadrature is used. Assuming equal measurement error variances and factor loadings but retaining the bias parameters \( \alpha_j \), the log-likelihood decreases substantially from -410.55 to -686.85 (12-pt adaptive quadrature) and the estimate \( \hat{\beta}_u \) of the effect of the true covariate is attenuated from \(-1.54 (0.51)\) to \(-1.04 (0.35)\), the true value being \(-2.\)

4 Discussion

We have developed the \texttt{gllamm} wrapper \texttt{cme} which makes estimation of an important class of generalized linear covariate measurement error models extremely easy while making use of the reliable adaptive quadrature method provided by \texttt{gllamm}. The wrapper will also produce the commands necessary to manipulate the data and estimate the model in \texttt{gllamm}, which should be useful for learning to use \texttt{gllamm}. The generated commands can subsequently be modified to extend the model or to use \texttt{gllamm}'s post-estimation commands \texttt{gllapred} and \texttt{gllasim} for prediction and simulation.

We have also extended the classical covariate measurement error model to relax the assumptions that the true covariate is normally distributed and that the fallible measurements of the true covariate are exchangeable replicates with the same measurement properties.

In addition to dichotomous outcomes and continuous measurements as discussed here, \texttt{gllamm} can handle continuous, censored, ordinal and nominal responses and rankings (Skrondal and Rabe-Hesketh 2003), as well as continuous and discrete-time durations (Rabe-Hesketh et al. 2001b). Moreover, the measurements may be of mixed type.

We have discussed only the most common case of a single covariate measured with error. Models with several imperfectly measured covariates can easily be handled by \texttt{gllamm}, see also Rabe-Hesketh et al. (2004b). The NPMLE example showed how dichotomous true covariates and latent class modeling can be handled in \texttt{gllamm}. However, at present it is not possible to have both continuous and categorical true covariates in the same model. When the true covariate is available in a validation sample, the model can also be estimated in \texttt{gllamm}, see Skrondal and Rabe-Hesketh (2004), Chapter 14, for an example. \texttt{gllamm} can be used to estimate multilevel generalized linear mixed models with covariate measurement error. When the outcome has been measured several times, it can also be useful to construct models for a latent true outcome, see Rabe-Hesketh et al. (2004b), Skrondal and Rabe-Hesketh (2004) and Rabe-Hesketh et al. (2004a). Finally, we can allow measurement errors to be correlated (if the model is identified) using further latent variables.
The \texttt{gllamm} command is also useful for other problems involving latent variables, such as multilevel random effects models, multilevel structural equation models (including factor and item response models), latent class models and multilevel selection and endogenous treatment models, see Skrondal and Rabe-Hesketh (2004).

Bobby Gutierrez at Stata Corporation has achieved a considerable increase in the speed of \texttt{gllamm} in January (2003) by converting an important part of \texttt{gllamm} to internal code. As a result, \texttt{gllamm} is fairly quick for the models estimated in this paper. For models including many latent variables, \texttt{gllamm} can however still be slow and we are working on further speed improvements.

\section*{Acknowledgements}

We would like to thank David Clayton for kindly providing us with the data.

\section*{References}


—. 2003b. The regression calibration method for fitting generalized linear models with additive measurement error. The Stata Journal ?: ??

—. 2003c. The simulation extrapolation method for fitting generalized linear models with additive measurement error. The Stata Journal ?: ??


—. 2004a. Multilevel and Structural Equation Modeling of Continuous, Categorical and Event Data. College Station, TX: Stata Press.


Appendix: Syntax for cme

cme depvar [varlist] (label: varlist) [weight] [if exp] [in range] [, mevar(#) family(familyname) link(linkname) denom(varname) noconstant offset(varname) tcovmod(varlist) simple nip(#) noadapt robust cluster(varname) indirect total eform level(#) noglog trace from(matrix) copy skip ]

The outcome model is specified by depvar [varlist], family(familyname), etc.

The classical measurement model for the true covariate is specified by (label: varlist), where label is the name of the true covariate (cannot be the same as an existing variable in the data set) and varlist are the fallible (continuous) measurements of the true covariate. At least two variables are required unless the mevar(#) option is used.

The true covariate model is a linear regression with explanatory variables (observed covariates) [varlist] unless the tcovmod(varlist) option is used to specify different explanatory variables.

families links
  gaussian identity
  poisson log
  gamma reciprocal
  binomial logit
  probit
  cll (complimentary log-log)
  ologit (o stands for ordinal)
  oprobit
  ocll
Options

mevar(#) specifies the measurement error variance. This option is required if there are no replicate measurements.

family(familyname) specifies the distribution of depvar; family(gaussian) is the default.

denom(varname) specifies the binomial denominator for the binomial link when depvar is the number of successes out of a fixed number of trials.

link(linkname) specifies the link function; the default is the canonical link for the family() specified.

noconstant specifies that the linear predictor has no intercept term, thus forcing it through the origin on the scale defined by the link function.

offset(varname) specifies an offset to be added to the linear predictor.

tcovmod(varlist) specifies the observed covariates to be used in the true covariate model; a constant will automatically be estimated.

simple specifies that there are no observed covariates in the true covariate model.

nip(#) the number of quadrature points to be used.

noadapt use ordinary quadrature instead of the default adaptive quadrature.

robust specifies that the Huber/White/sandwich estimator of variance is to be used.

cluster(varname) specifies that the observations are independent across groups (clusters), but not necessarily within groups.

commands displays the commands necessary to prepare the data and estimate the model in gllamm instead of estimating the model. Note that these commands change the data!

indirect displays the indirect effects of observed covariates on the outcome via the true covariate - this is shown for all covariates in the true covariate model.

total displays the total effects (indirect effects plus direct effects) of observed covariates on the outcome via the true covariate - this is shown for all covariates in the true covariate model.

eform displays the exponentiated coefficients and corresponding standard errors and confidence intervals.

level(#) specifies the confidence level, in percent, for confidence intervals (default 95).

nolog suppresses the iteration log.

trace requests that the estimated coefficient vector be printed at each iteration. In addition, all the output produced by gllamm with the trace option is also produced.

from(matrix) specifies a matrix of starting values.
Maximum likelihood

copy combined with the `from(matrix)` option, specifies that the matrix of starting values is in the order required by `cme` so that equation and column names will be ignored.

skip combined with the `from(matrix)` option, allows the matrix of starting values to contain extra parameters.

About the Authors

Sophia Rabe-Hesketh is Professor in the Graduate School of Education, University of California, Berkeley.

Anders Skrondal is Head of Biostatistics, Division of Epidemiology, Norwegian Institute of Public Health, Oslo.

Andrew Pickles is Professor of Epidemiological and Social Statistics at the School of Epidemiology and Health Science & CCSR, The University of Manchester.