Estimating parameters of dichotomous and ordinal item response models with gllamm

Xiaohui Zheng and Sophia Rabe-Hesketh
Graduate School of Education
University of California, Berkeley
Berkeley, CA
zhengx@berkeley.edu

Abstract. Item response theory models are measurement models for categorical responses. Traditionally, the models are used in educational testing, where responses to test items can be viewed as indirect measures of latent ability. The test items are scored either dichotomously (correct–incorrect) or by using an ordinal scale (a grade from poor to excellent). Item response models also apply equally for measurement of other latent traits. Here we describe the one- and two-parameter logit models for dichotomous items, the partial-credit and rating scale models for ordinal items, and an extension of these models where the latent variable is regressed on explanatory variables. We show how these models can be expressed as generalized linear latent and mixed models and fitted by using the user-written command gllamm.

Keywords: st0129, glamm, glapred, latent variables, Rasch model, partial-credit model, rating scale model, latent regression, generalized linear latent and mixed model, adaptive quadrature, item response theory

1 Introduction

A latent variable is a characteristic that is not directly observable. Examples include intelligence, happiness, satisfaction, and attitudes. Latent variables can be measured indirectly through their effects on observable indicators, such as items in achievement tests or psychological questionnaires.

Item response theory (IRT) provides statistical models for the relationship between item responses and the latent variable. Unfortunately, Stata and other traditional statistical packages, such as SAS and SPSS, do not provide commands specifically for IRT model estimation. In Stata, one can use clogit for conditional maximum-likelihood estimation of the fixed-effect logistic model and xtlogit for marginal maximum-likelihood estimation of the one-parameter logistic model. The user-written command raschtest (Hardouin 2007) uses these commands as well as glamm for fitting IRT models and obtaining related fit statistics and graphs. However, this command cannot be used for ordinal response models. This article shows how the binary logit models for dichotomous items and the partial-credit and rating scale models for ordinal items can be placed within the generalized linear latent and mixed modeling (GLLAMM) framework and fitted by using the Stata program glamm (see Rabe-Hesketh, Skrondal, and Pickles...
[2004a] and Rabe-Hesketh and Skrondal [2005] for the graded response model). We also show how the models can be extended by regressing the latent variable on explanatory variables.

## 2 IRT models

### 2.1 One- and two-parameter logistic models

The Rasch model (Rasch 1960, 1961) is the most well-known IRT model for dichotomous responses. It was first proposed by Georg Rasch and further developed by Wright (1977) and Fischer (1995). In the Rasch model, the probability of a correct or positive response for item $i$ by person $n$ is modeled as a function of an item parameter, $\delta_i$, representing item difficulty, and a person parameter, $\theta_n$, representing the person’s magnitude of the latent trait:

$$
Pr(x_{in} = 1|\theta_n) = \frac{\exp(\theta_n - \delta_i)}{1 + \exp(\theta_n - \delta_i)}
$$

The model is referred to as a one-parameter logistic (1PL) model because there is one parameter, $\delta_i$, per item. For achievement tests, the latent trait is often referred to as person ability. An appealing property of the model is that persons and items are placed on a common scale. The probability of a correct response increases with person ability (for a given item) and decreases with item difficulty (for a given person) and equals 1/2 when the person ability equals the item difficulty.

Birnbaum (1968) introduced the two-parameter logistic (2PL) model, which includes a slope parameter, $\lambda_i$, in addition to the intercept parameter $\delta_i$:

$$
Pr(x_{in} = 1|\theta_n) = \frac{\exp\{\lambda_i(\theta_n - \delta_i)\}}{1 + \exp\{\lambda_i(\theta_n - \delta_i)\}}
$$

(1)

The slope parameter $\lambda_i$ is referred to as a discrimination parameter because it determines how well an item discriminates among different trait levels (at least, for $\theta_n$ near $\delta_i$). The terms in the curly braces are sometimes written as $(\lambda_i \theta_n - \beta_i)$, where $\beta_i$ is equivalent to $\lambda_i \delta_i$ in (1). In the alternative formulation, the item difficulty is represented by $\beta_i / \lambda_i$. In both the 1PL and 2PL models, it is usually assumed that $\theta_n \sim N(0, \psi)$. In the 2PL model, either $\psi$ or $\lambda_1$ is set to 1 for identification.

### 2.2 Partial-credit model

The partial-credit model (PCM; Masters 1982) is an extension of the Rasch model to polytomous items with ordered response categories $0, 1, \ldots, m_i$ for item $i$. 

\[ \textit{Estimating parameters with gllamm} \]
The PCM specifies the probability of responding in the $j$th category of item $i$ for person $n$ as a function of the person ability $\theta_n$ and step parameters $\delta_{ij}$ ($j > 0$)

$$\Pr(x_{in} = j|\theta_n) = \frac{\exp \sum_{l=0}^{j}(\theta_n - \delta_{il})}{\sum_{l=0}^{m_i} \exp \sum_{l=0}^{j}(\theta_n - \delta_{il})} \quad j = 0, 1, \ldots, m_i$$

where $\sum_{l=0}^{j}(\theta_n - \delta_{il}) = 0$. This is a special case of a multinomial logit model, namely, an adjacent category logit model (Agresti 2002) with

$$\ln \frac{\Pr(x_{in} = j|\theta_n)}{\Pr(x_{in} = j - 1|\theta_n)} = \theta_n - \delta_{ij}$$

The parameter $\delta_{ij}$ is known as the step difficulty associated with category $j$ of item $i$. It represents the added difficulty when moving the step from category $j - 1$ to category $j$ (Embretson and Reise 2000; Wilson 2004).

A 2PL PCM (Muraki 1992) can also be specified by including a slope parameter, $\lambda_i$, that allows each item to have a different discrimination.

### 2.3 Rating scale model

The rating scale model (RSM; Andrich 1978) is a special case of the PCM. It is appropriate if the $m_i = m$ response categories have the same meaning for all items and assumes that the differences in the step difficulties for different categories are the same for all items.

The RSM structures the step difficulties of main effects $\delta_i$ of items $i$ and $\tau_j$ of response categories $j$ ($j > 0$):

$$\Pr(x_{in} = j|\theta_n) = \frac{\exp \sum_{l=0}^{j}(\theta_n - (\delta_i + \tau_l))}{\sum_{k=0}^{m} \exp \sum_{l=0}^{j}(\theta_n - (\delta_i + \tau_l))} \quad j = 0, 1, \ldots, m$$

where $\sum_{l=0}^{j}(\theta_n - (\delta_i + \tau_l)) = 0$. Interpretation of the model parameters depends on the choice of constraints for $\tau_j$. Traditionally, the constraint $\sum_{l=0}^{\tau_l} = 0$ is used so that $\delta_i$ represents the scale value (Wright and Masters 1982) of item $i$, reflecting its overall difficulty relative to other items. Then $\tau_j$ ($j = 1, 2, \ldots, m$) is the threshold parameter (Wright and Masters 1982) of category $j$, representing the location of the $j$th step of each item relative to its scale value. An alternative constraint is $\tau_1 = 0$, so that $\delta_i$ represents the first step difficulty for item $i$ and $\tau_j$ ($j = 1, 2, \ldots, m$) represents the extra step difficulty of subsequent steps compared with the first step. The generalized RSM includes a slope parameter, $\lambda_i$.

(Continued on next page)
3 IRT models in the GLLAMM framework

3.1 GLLAMM framework for IRT models

GLLAMMs (Rabe-Hesketh, Skrondal, and Pickles 2004a; Skrondal and Rabe-Hesketh 2004) are a class of multilevel latent variable models. We will not describe the full framework here. For IRT models, we require only the response model, two levels of nesting, and a latent variable. Here the vector of linear predictors for person \(n\) can be written as

\[
\nu_n = X_n \beta + \theta_n Z_n \lambda
\]  

(2)

where \(X_n\) and \(Z_n\) are design matrices, \(\beta\) and \(\lambda\) are corresponding vectors of parameters, and \(\theta_n\) is a latent variable.

In the 1PL and 2PL models, \(\nu_n\) represents the vector of log odds for items \(i = 1, \ldots, I\) and person \(n\). In the PCM and RSM, \(\nu_n\) represents the vector of the logarithms of the numerators of the models for items \(i = 1, \ldots, I\) and response categories \(j = 1, \ldots, m_i\).

In the next section we show how specific IRT models are parameterized by giving the required form of the design matrices \(X_n\) and \(Z_n\). The columns of these design matrices correspond directly to the variables needed to fit the models with \texttt{gllamm}.

3.2 1PL and 2PL

In the 1PL and 2PL models, the vector of linear predictors \(\nu_n\) represents the log odds of a correct response. Under the framework in (2), the 1PL model for, say, four dichotomous items is written as

\[
\nu_n = \begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 \\
\end{bmatrix}
\begin{bmatrix}
\delta_1 \\
\delta_2 \\
\delta_3 \\
\delta_4 \\
\end{bmatrix}
+ \theta_n \begin{bmatrix}
1 \\
1 \\
1 \\
1 \\
\end{bmatrix}
\]

\[
\nu_n = \begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 \\
\end{bmatrix}
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\end{bmatrix}
+ \theta_n I_{4 \times 4}
\]

In the 2PL model, slope or discrimination parameters \(\lambda_i\) \((i = 1, 2, 3, 4)\) are introduced with \(\lambda_1\) set to 1 for identification:

\[
\begin{bmatrix}
\nu_{1n} \\
\nu_{2n} \\
\nu_{3n} \\
\nu_{4n} \\
\end{bmatrix}
= \begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 \\
\end{bmatrix}
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\end{bmatrix}
+ \theta_n I_{4 \times 4}
\]

\[
\begin{bmatrix}
\nu_{1n} \\
\nu_{2n} \\
\nu_{3n} \\
\nu_{4n} \\
\end{bmatrix}
= \begin{bmatrix}
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 \\
\end{bmatrix}
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\end{bmatrix}
+ \theta_n I_{4 \times 4}
\]

\[
\begin{bmatrix}
\lambda_1 \\
\lambda_2 \\
\lambda_3 \\
\lambda_4 \\
\end{bmatrix}
\]

3.3 Partial-credit model

In the PCM, the linear predictors \(\nu_{ijn}\) represent the logarithms of the numerators of the response probabilities:

\[
Pr(x_{in} = j|\theta_n) = \frac{\exp(\nu_{ijn})}{\sum_{j=0}^{m} \exp(\nu_{ikn})} \quad j = 0, 1, \ldots, m_i
\]
Consider first the numerator $\nu_{ijn}$:
when $j = 0, \nu_{i0n} = 0$
when $j = 1, \nu_{i1n} = 0 + (\theta_n - \delta_{i1})$
when $j = 2, \nu_{i2n} = 0 + (\theta_n - \delta_{i1}) + (\theta_n - \delta_{i2}) = -\delta_{i1} - \delta_{i2} + 2\theta_n$
when $j = 3, \nu_{i3n} = 0 + (\theta_n - \delta_{i1}) + (\theta_n - \delta_{i2}) + (\theta_n - \delta_{i3}) = -\delta_{i1} - \delta_{i2} - \delta_{i3} + 3\theta_n$

Since each response probability is a function of all $\nu_{ijn}$ ($j = 0, \ldots, m_i$) in the denominator, the data must be expanded so that each original response is represented by $m_i + 1$ rows in the expanded dataset. For two polytomous items, each with four response categories ($m_1 = m_2 = 3$), the PCM is parameterized as

$$
\begin{bmatrix}
\nu_{10n} \\
\nu_{11n} \\
\nu_{12n} \\
\nu_{13n} \\
\nu_{20n} \\
\nu_{21n} \\
\nu_{22n} \\
\nu_{23n} \\
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \begin{bmatrix}
\delta_{11} \\
\delta_{12} \\
\delta_{13} \\
\delta_{21} \\
\delta_{22} \\
\delta_{23} \\
\delta_{31} \\
\delta_{32} \\
\end{bmatrix} + \theta_n
$$

A 2PL PCM has a different $Z_n$ matrix followed by a loading vector:

$$
\begin{bmatrix}
\nu_{10n} \\
\nu_{11n} \\
\nu_{12n} \\
\nu_{13n} \\
\nu_{20n} \\
\nu_{21n} \\
\nu_{22n} \\
\nu_{23n} \\
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \begin{bmatrix}
\delta_{11} \\
\delta_{12} \\
\delta_{13} \\
\delta_{21} \\
\delta_{22} \\
\delta_{23} \\
\delta_{31} \\
\delta_{32} \\
\end{bmatrix} + \theta_n
$$

3.4 Rating scale model

For two polytomous items each with four response categories, the RSM has the following matrix form:

$$
\begin{bmatrix}
\nu_{10n} \\
\nu_{11n} \\
\nu_{12n} \\
\nu_{13n} \\
\nu_{20n} \\
\nu_{21n} \\
\nu_{22n} \\
\nu_{23n} \\
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \begin{bmatrix}
\delta_1 \\
\delta_2 \\
\tau_2 \\
\tau_3 \\
\end{bmatrix} + \theta_n
$$

In the 2PL RSM, the $Z_n$ matrix and the loading vector are the same as for the 2PL PCM.

4 gllamm

The gllamm command runs in Stata and performs maximum likelihood estimation for GLLAMMs by using adaptive quadrature (Rabe-Hesketh, Skrondal, and Pickles 2002; 2005). Here we introduce gllamm commands and options relevant to the estimation of
item response models. Users can refer to the gllamm manual (Rabe-Hesketh, Skrondal, and Pickles 2004a) for a full description of its commands and options.

### 4.1 Syntax

Below is the gllamm syntax with all options needed for fitting IRT models.

```plaintext
gllamm depname explnames, i(varname) [family(famname) link(linkname)
    noconstant eqs(eqname) geqs(eqnames)
    expanded(varname1 varname2 o) weightf(wtname) nip(#) adapt trace]
```

depname gives the name of the response variable. Item responses must be stacked into one response variable before estimation.

explnames gives the names of explanatory variables that form the columns of $X_n$.

### 4.2 Options

- **i(varname)** specifies the variable that defines the clusters (i.e., persons in the IRT models).
- **family(famname)** specifies the conditional distribution of the response given the linear predictor as one of the exponential family of distributions, such as binomial, poisson, and gamma. The default is family(gaussian).
- **link(linkname)** specifies the link function linking the linear predictor to the conditional expectation of the response. Available link functions include logit, probit, ologit, oprobit, and mlogit.
- **noconstant** omits the constant in the fixed part so that $X_n$ has as many columns as there are explanatory variables.
- **eqs(eqname)** specifies an equation that defines the columns of $Z_n$. The equation must be defined before running gllamm with an eq command (See appendix A of Rabe-Hesketh and Skrondal (2005) for more information about the eq command).
- **geqs(eqname)** specifies an equation for a regression of the latent variable on explanatory variables.
- **expanded(varname1 varname2 o)** indicates that the data have been expanded to have one row for each response category. varname1 labels each item–person combination identifying the groups of linear predictors that contribute to the same denominators. varname2 is an indicator for the chosen category identifying the linear predictor that should contribute to the numerator. o tells the program to estimate only one set of regression coefficients for the explanatory variables (not a separate set for each response category).
weightf(\text{wtname}) specifies the stub for variables (\text{wtname1}, \text{wtname2}, etc.) that contain frequency weights. The suffixes in the variable names determine at what level each weight applies. If only some of the weight variables exist, the other weights are assumed to be equal to 1. When many observations have the same response pattern, collapsing the data and using weights can speed up the estimation.

nip(\#) specifies the number of integration points to be used for evaluating the integral. The default is nip(8).

\texttt{adapt} requests adaptive rather than ordinary quadrature.

\texttt{trace} displays the parameter estimates in each iteration.

### 4.3 Examples

The data we use for dichotomous models are from an article (Thissen, Steinberg, and Wainer 1993, 71) that examined student spelling performance on four words: \textit{infidelity}, \textit{panoramic}, \textit{succumb}, and \textit{girder}. The sample includes 285 male and 374 female undergraduate students from the University of Kansas. Each item was scored as either correct or incorrect.

The data we use for ordinal models are from the 38th round of the State Survey conducted by Michigan State University’s Institute for Public Policy and Social Research (2005). The survey was administered to 949 Michigan citizens from May 28 to July 18, 2005, by telephone. The focus of the survey included charitable giving and volunteer activities of Michigan households. Five questions measured the public’s faith and trust in charity organizations. Respondents were asked to indicate to what degree they agree with the following five statements:

- “Charitable organizations are more effective now in providing services than they were 5 years ago.”
- “I place a low degree of trust in charitable organizations.”
- “Most charitable organizations are honest and ethical in their use of donated funds.”
- “Generally, charitable organizations play a major role in making our communities better places to live.”
- “On the whole, charitable organizations do not do a very good job in helping those who need help.”

The questions have four response categories corresponding to “strongly agree”, “somewhat agree”, “somewhat disagree”, and “strongly disagree”. For this article, we coded responses from 0 to 3, with larger scores indicating less favorable views of charities.
Estimating parameters with **gllamm**

**1PL and 2PL models**

We use the spelling data to illustrate the **gllamm** command for the binary logistic item response models. Below is a listing of the first six rows of data. i1 to i4 are the outcomes (1, correct; 0, incorrect) for the four spelling words and **male** is a dummy variable for being a male.

```
   . use spelling
   . list in 1/6, clean
       male  i1  i2  i3  i4  wt2
     1.      0  0  0  0  0  29
     2.      1  0  0  0  0  22
     3.      0  0  0  0  1  7
     4.      1  0  0  0  1 10
     5.      0  0  0  1  0  6
     6.      1  0  0  1  0  1
```

The data have been collapsed, with wt2 containing the frequency weights for each response–gender combination. For example, 29 females and 22 males spelled all four words incorrectly; seven females and 10 males could spell only the fourth word, *girder*, correctly.

To use **gllamm**, you must stack item responses into one response vector. First, we generate a new variable **pattern** as an identifier for each response–gender combination. Then variables i1 to i4 are stacked into a response variable, score, with **pattern** and **item** identifying the subject n and the item i, respectively. This layout corresponds to the vector $\nu_n$ in section 3.2.

```
   . gen pattern=_n
   . reshape long i, i(pattern) j(item)
   (output omitted)
   . rename i score
   . list in 1/8, clean
       pattern  item  male  score  wt2
     1.   1  1  0  0  29
     2.   1  2  0  0  29
     3.   1  3  0  0  29
     4.   1  4  0  0  29
     5.   2  1  1  0  22
     6.   2  2  1  0  22
     7.   2  3  1  0  22
     8.   2  4  1  0  22
```

Next four dummy variables, d1 to d4, are created for the items. These dummies are then changed to their negatives, negd1 to negd4, which constitute the columns of the design matrix $X_n$ in section 3.2.

```
   . tab item, gen(d)
   (output omitted)
   . forvalues i=1/4 {
      2.       generate negd'`i'=-d'`i'
      3.   }
```
The first four rows of `negd1` to `negd4` are below.

```
. list negd1-negd4 in 1/4, clean
     negd1  negd2  negd3  negd4
     1.   -1    0    0    0
     2.    0   -1    0    0
     3.    0    0   -1    0
     4.    0    0    0   -1
```

**IPL model.** We can now fit the one-parameter model with the command below. The `weight()` option specifies the stub `wt`; `gllamm` interprets `wt2` as level 2 weights, meaning that they apply to the entire level 2 cluster—here, person.

```
. gllamm score negd1-negd4, i(pattern) link(logit) family(binom) weight(wt)
>   nip(15) nocons adapt trace

(omit output)

```

**gllamm model**

```
log likelihood =  -1564.0028

```

| Coef. | Std. Err. | z    | P>|z| | [95% Conf. Interval] |
|-------|-----------|------|-------|---------------------|
| negd1 | -1.698033 | .1250384 | -13.58 | .000 | -1.943103 to -1.452962 |
| negd2 | -.6628738 | .1053714 | -6.29  | .000 | -.869377 to -.4563496 |
| negd3 | 1.080796  | .1111877 | 9.72   | .000 | .8628717 to 1.298719  |
| negd4 | -.1596644 | .1018936 | -1.57  | .117 | -.3593719 to .0400434 |

Variances and covariances of random effects

```
***level 2 (pattern)

```

```
var(1): 1.5424502 (.23150165)

```

. estimates store onepl

After fitting the model, we store the estimates for later use in likelihood-ratio tests. The coefficients of `negd1` to `negd4` in the output are the estimated item difficulties \( \hat{\delta}_i \). As indicated by the four estimates, the spelling of `infidelity` is the easiest and the spelling of `succumb` is the most difficult. The level 2 variance represents the variance of student abilities and is estimated as 1.54 with a standard error of 0.23.

Figure 1 shows item characteristic curves (ICCs) describing the relationship between ability levels and probabilities of passing each item. The values of \( \theta_n \) where the curves cross the 0.5 probability line are the estimated item difficulties. The figure is produced using the following command:

```
. twoway (function Infidelity=invlogit(x-[score]negd1), range(-6 6))
>   (function Panoramic =invlogit(x-[score]negd2), range(-6 6) lpatt(".") )
>   (function Succumb =invlogit(x-[score]negd3), range(-6 6) lpatt("-") )
>   (function Girder =invlogit(x-[score]negd4), range(-6 6) lpatt("_") )
```

where `[score]negd\_i` \((i = 1, 2, 3,\ and\ 4)\) accesses the estimate \( \hat{\delta}_i \).
Estimating parameters with gllamm

\[ 0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1 \]
\[ -5 \quad 0 \quad 5 \]

\[ x \]
Infidelity Panoramic Succumb Girder

Figure 1: ICCs of the four spelling items with the Rasch (1PL) model

Figure 2: ICCs of the four spelling items with the 2PL model

2PL model. The 2PL dichotomous model involves a vector, \( \lambda \), of item loadings. \texttt{eq} defines an equation for the columns of the corresponding design matrix \( Z_n \). The equation is then included in the \texttt{gllamm} command by using the \texttt{eqs()} option:

```bash
.gllamm score negd1-negd4, i(pattern) eqs(loading) link(logit) family(binom)
> weight(wt) nip(15) nocons adapt trace
> (output omitted)
gllamm model

log likelihood = -1563.2096
```

| score | Coef. | Std. Err. | z | P>|z| | [95% Conf. Interval] |
|-------|-------|-----------|---|-----|---------------------|
| negd1 | -1.580291 | .1411275 | -11.20 | 0.000 | -1.856896 -1.303686 |
| negd2 | -.6844128 | .1198053 | -5.71 | 0.000 | -.9192267 -.4495988 |
| negd3 | 1.101213 | .1368125 | 8.05 | 0.000 | .833065 1.36936 |
| negd4 | -.1629568 | .1049209 | -1.55 | 0.120 | -.3685981 .0426845 |

Variance and covariances of random effects

***level 2 (pattern)

var(1): .99850086 (.39317103)

loadings for random effect 1
d1: 1 (fixed)
d2: 1.3470427 (.38213122)
d3: 1.3026963 (.35903712)
d4: 1.316772 (.36213004)

. estimates store twopl

The coefficients of \( d1 \) to \( d4 \) under loadings for random effect 1 represent the estimated loadings of the four items. The estimates agree with those of previous studies that suggested that the four items have similar discrimination (Thissen, Steinberg, and Wainer 1993). The following likelihood-ratio test confirms this finding:
. lrtest onepl twopl  
Likelihood-ratio test  
(Assumption: onepl nested in twopl)  
LR chi2(3) =  1.59  
Prob > chi2 =  0.6625

The ICCs of the four items with the 2PL model are given in figure 2 and are plotted using the following command:

```
twoway > (function Infidelity=invlogit(x-[score]negd1), range(-6 6))
> (function Panoramic =invlogit([pat1_1l]d2*x-[score]negd2), range(-6 6) lpatt("."))
> (function Succumb =invlogit([pat1_1l]d3*x-[score]negd3), range(-6 6) lpatt("-"))
> (function Girder =invlogit([pat1_1l]d4*x-[score]negd4), range(-6 6) lpatt("_"))
```

Users can find out how to refer to parameters by displaying the matrix of the estimates:

```
. matrix list e(b)
e(b)[1,8]  
score: score: score: score: pat1_1l: pat1_1l:  
negd1 negd2 negd3 negd4 d2 d3  
y1  -1.5802912 -.68441276 1.1012126 -.16295682 1.3470427 1.3026963  
pat1_1l: pat1_1l:  
d4 d1  
y1  1.316772 .99925034
```

**PCM**

We use the charity data to illustrate the `gllamm` command for the PCM and RSM. The data are first collapsed so that there is one row per unique response pattern, with a weight variable, `wt2`, indicating the number of people for each response pattern.

```
. use charity, clear
. gen one=1
. collapse(sum) wt2=one, by(ta1-ta5)
. gen id=_n
. list in 1/2, clean
  ta1   ta2   ta3   ta4   ta5   wt2   id
  1.  0  0  0  0   0   27   1
  2.  0  0  0  0   1   5    2
```

Then the five columns of item responses are stacked into one response variable, `ta`. The `id` variable is the cluster identifier that labels each observation.

(Continued on next page)
Estimating parameters with gllamm

```
. reshape long ta, i(id) j(item)
(omitted)
. list in 1/10, clean
   id  item  ta  wt2
   1. 1   1   0   27
   2. 1   2   0   27
   3. 1   3   0   27
   4. 1   4   0   27
   5. 1   5   0   27
   6. 2   1   0   5
   7. 2   2   0   5
   8. 2   3   0   5
   9. 2   4   0   5
  10. 2   5   1   5

After item responses are stacked into one response variable, we create a new variable, obs, to identify each item–person combination for the PCM. The data are then expanded to have one row for each response category, as shown in section 3.3.

. drop if ta==.
(122 observations deleted)
. gen obs=_n
. expand 4
(5394 observations created)
. sort id item obs
. list in 1/8, clean
   id  item  ta  wt2  obs
   1. 1   1   0   27   1
   2. 1   1   0   27   1
   3. 1   1   0   27   1
   4. 1   1   0   27   1
   5. 1   2   0   27   2
   6. 1   2   0   27   2
   7. 1   2   0   27   2
   8. 1   2   0   27   2

Next we generate the variable x to contain all possible scores (0, 1, 2, 3) for each item–person combination. The variable chosen specifies the response category the people actually selected. The variable iti is a dummy for the ith item.

. by obs, sort: gen x = _n-1
. gen chosen = ta == x
. tab item, gen(iti)
(omitted)
```
The first eight rows of the resulting data are below.

```
. list id-it2 in 1/8, clean
      id item ta wt2 obs  x  chosen  it1 it2
  1.  1   1  0  27   1  1   1   0
  2.  1   1  0  27   1  1   0   1
  3.  1   1  0  27   1  2   0   1
  4.  1   1  0  27   1  3   0   1
  5.  1   2  0  27   2  0   1   0
  6.  1   2  0  27   2  1   0   1
  7.  1   2  0  27   2  2   0   1
  8.  1   2  0  27   2  3   0   1
```

The variables corresponding to the design matrix $X_n$ for the PCM given in section 3.3 are generated as follows:

```
. forvalues i=1/5 {
  2.   forvalues g=1/3 {
  3.     gen d`i’`g’ = -1*it`i’*(x>=`g’)
  4.   }
  5. }
```

```
. list d1_1-d2_3 in 1/8, clean
      d1_1 d1_2 d1_3 d2_1 d2_2 d2_3
  1.   0   0   0   0   0   0
  2.  -1   0   0   0   0   0
  3.  -1  -1   0   0   0   0
  4.  -1  -1  -1   0   0   0
  5.   0   0   0   0   0   0
  6.   0   0   0  -1   0   0
  7.   0   0   0  -1  -1   0
  8.   0   0   0  -1  -1  -1
```

The PCM is then fitted to the data by using the following commands. `eq` defines an equation corresponding to the columns of the design matrix $Z_n$. This equation is specified using the `eqs()` option. The `expand()` option tells the program that the data have been expanded to one row for each possible response category. The variable `obs` indicates which linear predictors need to be combined for the denominator of the PCM, and the dichotomous variable `chosen` picks out the linear predictor that goes into the numerator.

(Continued on next page)
Estimating parameters with gllamm

```
   . eq slope: x
   . gllamm x d1_1-d5_3, i(id) eqs(slope) link(mlogit) expand(obs chosen o)
>   weight(wt) adapt trace nocons
   (output omitted)
   gllamm model
   log likelihood = -5209.5824
```

| x     | Coef.   | Std. Err. | z      | P>|z|  | [95% Conf. Interval] |
|-------|---------|-----------|--------|-----|---------------------|
| d1_1  | -1.122041 | .0965448  | -11.62 | 0.000 | -1.311265 - .9328167 |
| d1_2  | 1.157892  | .0985121  | 11.75  | 0.000 | .964812  1.350972  |
| d1_3  | 1.887521  | .166362   | 11.35  | 0.000 | 1.561458  2.213584  |
| d2_1  | -.8315028 | .1088378  | -7.64 | 0.000 | -1.044821 -.6181846 |
| d2_2  | -.2690148 | .0899575  | -2.99 | 0.003 | -.4453284 -.0927013 |
| d2_3  | 1.835945  | .1250195  | 14.69  | 0.000 | 1.590911  2.080979  |
| d3_1  | -1.239309 | .0944551  | -13.12 | 0.000 | -1.424437 -1.05418  |
| d3_2  | 1.421748  | .1005407  | 14.14  | 0.000 | 1.224692  1.618804  |
| d3_3  | 1.853141  | .172      | 10.78  | 0.000 | 1.516201  2.190428  |
| d4_1  | -.3146175 | .0811645  | -3.88 | 0.000 | -.4736969 -.155538  |
| d4_2  | 2.013949  | .1264109  | 15.93  | 0.000 | 1.766189  2.261711  |
| d4_3  | 1.844851  | .2147572  | 8.59   | 0.000 | 1.423876  2.265826  |
| d5_1  | -.6076893 | .0928491  | -6.54 | 0.000 | -.7896701 -.4257084 |
| d5_2  | .6538114  | .0952031  | 6.87   | 0.000 | .4672168  .8404061  |
| d5_3  | 1.643881  | .1413931  | 11.63  | 0.000 | 1.366756  1.921007  |

```

Variance and covariances of random effects

***level 2 (id)

var(1): .78617553 (.07393701)

```

. estimates store pcm

The coefficient of $d_{ij}$ is the estimated step difficulty $\delta_{ij}$ for item $i$ and category $j$. To create category probability curves (CPCs) for each item, we first generate a latent scale variable, trait1, that increases in equal steps from $-4$ to 4. With the us() and mu options, the gllapred command calculates conditional probabilities given the latent variable trait1.

```
   quietly egen N=max(id)
   generate trait1 = (-4) + (id-1)*(4-(-4))/(N-1)
   gllapred prob1, mu us(trait)
```

The CPCs for item 4 under the PCM are plotted using the following command and are given in figure 3:
. twoway (line prob1 trait1 if x==0, sort) 
  > (line prob1 trait1 if x==1, sort lpatt(".")) 
  > (line prob1 trait1 if x==2, sort lpatt("-"))) 
  > (line prob1 trait1 if x==3, sort lpatt("_")) if item==4, 
  > legend(order(1 "strongly agree" 2 "somewhat agree" 
  > 3 "somewhat disagree" 4 "strongly disagree"))

Category 1, represented by the first curve from the left, is most likely to be observed 
among low-trait respondents, and category 4, represented by the last curve, is most 
likely to be observed among high-trait respondents. The value of the latent trait where 
the probability curves for adjacent categories \( j - 1 \) and \( j \) intersect is the estimated step 
parameter \( \delta_{4j} \), the coefficient of \( d_{4j} \) in the PCM output.

Figure 3: CPCs for item 4 under the PCM  Figure 4: CPCs for item 4 under the RSM

The 2PL PCM uses a different \( Z_n \) matrix, as shown in section 3.3, which can be 
generated as follows:

. forvalues i=1/5 {
  2. gen x_it'i'=x*it'i'
  3. }

. sort id item x
. list x_it1 x_it2 in 1/8, clean

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
</tbody>
</table>

The \texttt{gllamm} command for the 2PL PCM is

. eq load: x_it1-x_it5
. gllamm x d1_1-d5_3, i(id) eqs(load) link(mlogit) expand(obs chosen o)
  > weight(wt) adapt trace nocons
  (output omitted)
RSM

The design matrix $X_n$ for the RSM has fewer columns than the one for the PCM. We first generate the columns of the matrix that correspond to the common step parameters.

```stata
. gen step1 = -1*(x>=1)
. gen step2 = -1*(x>=2)
. gen step3 = -1*(x>=3)
```

The columns for the item scale parameters are generated within a loop:

```stata
. foreach var of varlist it* {

2. gen n`var' = -1*`var'*x

3. }
```

We now look at the design matrix $X_n$ for items 1 and 2, as given in section 3.4.

```stata
. sort id item x
. list nit1 nit2 step2 step3 in 1/8, clean

<table>
<thead>
<tr>
<th>nit1</th>
<th>nit2</th>
<th>step2</th>
<th>step3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3.</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>4.</td>
<td>-3</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>5.</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>7.</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
</tr>
<tr>
<td>8.</td>
<td>0</td>
<td>-3</td>
<td>-1</td>
</tr>
</tbody>
</table>
```

We then fit the RSM by using the following `gllamm` command:

```stata
. eq slope: x
. gllamm x nit1-nit5 step2 step3, i(id) eqs(slope) link(mlogit)
> expand(obs chosen o) weight(wt) adapt trace nocons
```

```
gllamm model

log likelihood = -5293.9307
```

|   | Coef. | Std. Err. |   z  | P>|z| | [95% Conf. Interval] |
|---|-------|-----------|------|-----|---------------------|
| nit1 | -.8765313 | .0671063 | -13.06 | .000 | -1.008057  -.745053 |
| nit2 | -1.447597 | .0723549 | -20.01 | .000 | -1.589410  -1.305784 |
| nit3 | -10.8179617 | .06858133 | -12.43 | .000 | -.9468534  -.68887 |
| nit4 | -20.76768 | .0632331 | -3.28 | .001 | -33.1615  -8.372 |
| nit5 | -95.118855 | .0699995 | -14.20 | .000 | -108.2502  -.819689 |
| step2 | 1.80703 | .0720486 | 25.08 | .000 | 1.665818  1.948243 |
| step3 | 2.801625 | .100877 | 27.77 | .000 | 2.603888  2.999361 |
Variances and covariances of random effects

***level 2 (id)

\[ \text{var}(1): 0.77909796 (0.07350611) \]

estimates store rsm

The 2PL RSM shares the same design matrix, \( Z_n \), as the 2PL PCM. The \texttt{gllamm} command for fitting the 2PL RSM is

\begin{verbatim}
. eq load: x_it1-x_it5
. gllamm x nit1-nit5 step2 step3, i(id) eqs(load) link(mlogit) 
> expand(obs chosen o) weight(wt) adapt trace nocons
(output omitted)
\end{verbatim}

In the RSM output, the coefficient of \( niti \) is the estimated step parameter \( \hat{\delta}_i \) for the first step of item \( i \). The coefficient of \( \text{step}_j \) is the estimated additional difficulty \( \hat{\tau}_j \) for the step from \( j - 1 \) to \( j \) (\( j = 2, 3 \)), whereas \( \tau_1 \) is constrained to 0 for all items. Table 1 shows the step difficulty estimates for items 1 and 2 on the basis of the PCM and the RSM.

### Table 1: Step difficulty estimates for items 1 and 2, using the PCM and the RSM

<table>
<thead>
<tr>
<th>Item</th>
<th>Step</th>
<th>PCM Step difficulty</th>
<th>Estimate</th>
<th>RSM Step difficulty</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>( \delta_{11} )</td>
<td>-1.06</td>
<td>( \delta_1 )</td>
<td>-0.90</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>( \delta_{12} )</td>
<td>1.09</td>
<td>( \delta_1 + \tau_2 )</td>
<td>-0.90 + 1.71</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>( \delta_{13} )</td>
<td>1.51</td>
<td>( \delta_1 + \tau_3 )</td>
<td>-0.90 + 2.61</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>( \delta_{21} )</td>
<td>-0.79</td>
<td>( \delta_2 )</td>
<td>-1.37</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>( \delta_{22} )</td>
<td>-0.29</td>
<td>( \delta_2 + \tau_2 )</td>
<td>-1.37 + 1.71</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>( \delta_{23} )</td>
<td>1.76</td>
<td>( \delta_2 + \tau_3 )</td>
<td>-1.37 + 2.61</td>
</tr>
</tbody>
</table>

Variance 0.66 0.70

The CPCs for item 4 under the RSM are given in figure 4. This graph is produced using the same commands as for the PCM.

Given that the RSM model is nested within the PCM, we use a likelihood-ratio chi-squared test via the \texttt{lrtest} command to compare the models. The parameter constraints imposed by the RSM model are clearly rejected.

\begin{verbatim}
. lrtest rsm pcm
Likelihood-ratio test LR chi2(8) = 168.70
(Assumption: rsm nested in pcm) Prob > chi2 = 0.0000
\end{verbatim}
4.4 Model extensions

The structural model of the GLLAMM framework (Rabe-Hesketh, Skrondal, and Pickles 2004a) allows latent variables to be regressed on each other and observed covariates. For a latent variable, the structural model becomes

\[ \theta_n = \gamma w_n + \zeta_n \]  

(3)

where \( w_n \) represents the vector of observed covariates with corresponding regression parameter vector \( \gamma \). The vector \( \zeta_n \) represents the disturbances.

Latent regression item response model

The latent regression Rasch model (Verhelst and Eggen 1989; Zwinderman 1991) is a 1PL model including person properties as predictors of the latent variable. Similar models have been presented by Mislevy (1987) for the 2PL model. For instance, the covariate \texttt{male} in the spelling data is dummy coded with a 1 for males and 0 for females. Under the structural model in (3), the latent variable \( \theta_n \) is modeled as

\[ \theta_n = \gamma \texttt{male}_n + \zeta_n \]

where \( \gamma \) is the regression coefficient of \texttt{male}, indicating the difference in spelling ability between male and female students.

We continue with our spelling data, using the following commands to fit the 1PL model combined with the structural model. The \texttt{eq} command defines the equation for the regression of the latent variable on \texttt{male}. The equation is then included in the \texttt{geqs()} option:

```
. eq f1: male
. gllamm score negd1-negd4, i(pattern) link(logit) family(binom) weight(wt)
>  geqs(f1) nip(15) nocons adapt trace
```

```
gllamm model  
log likelihood = -1562.4715
```

|      | Coef.  | Std. Err. |     z  |   P>|z|  |        [95% Conf. Interval]        |
|------|--------|-----------|-------|-------|-----------------------------------|
| negd1| -1.594569 | .1366747 | -11.67 | 0.000 | -1.862447 -1.326692             |
| negd2| -.5596258 | .1199634 | -4.66  | 0.000 | -.7947497 -.3245018            |
| negd3| 1.184559  | .1270645 |  9.32  | 0.000 | .935517  1.4336               |
| negd4| -.0563849 | .1174847 | -0.48  | 0.631 | -.2866506 .1738808            |
Variances and covariances of random effects

***level 2 (pattern)

\text{var}(1): 1.5297939 (.23035343)

Regressions of latent variables on covariates

random effect 1 has 1 covariates:

male: .24071446 (.13768983)

The output of the latent regression model is similar to that of the Rasch model. The coefficients of \(d_1\) to \(d_4\) are the four estimated item parameters. The level 2 variance is the variance of the disturbance or residual \(\zeta_n\) and is estimated as 1.53. The estimate of \(\gamma\), the coefficient of \text{male}, indicates that male students outperform female students by 0.24 logits, with a standard error of 0.14 logits. A latent regression can also be combined analogously with any of the other models described in this article.

**EAP scores**

After estimating the parameters of the IRT models with \texttt{gllamm}, we can run \texttt{gllapred} to obtain expected a posteriori (EAP) scores for each individual, also known as posterior means or empirical Bayes predictions.

For the IRT models in section 4.3 where no covariates are included, the EAP scores are given by

\[
E(\theta_n|Y_n) = \int_{-\infty}^{\infty} \theta_n \left\{ \prod_{i=1}^{J} \Pr(y_{in}|\theta_n) \right\} g(\theta_n) d\theta_n
\]

The following command with a \texttt{u} option produces posterior means and standard deviations of the latent variable, returned in the variables \texttt{thetam1} and \texttt{thetas1}, respectively.

\texttt{. gllapred theta, u}

For the extended models in section 4.4, the above command provides the posterior means and standard deviations of the disturbances \(\zeta_n\). To obtain the EAP estimates of the latent variable \(\theta_n\), we use the \texttt{fac} option.

\texttt{. gllapred theta, fac}

5 Conclusion

In this article, we expressed IRT models within the GLLAMM framework and fitted them with \texttt{gllamm}. GLLAMM also offers the flexibility to include extensions of the
Estimating parameters with gllamm

standard IRT models that fit within a nonlinear mixed-model framework (Rijmen et al. 2003; De Boeck and Wilson 2004). Besides IRT models, the GLLAMM framework encompasses a large variety of latent variable models, including generalized linear mixed models, structural equation models, latent class models, and multilevel versions of these models (Rabe-Hesketh, Skrondal, and Pickles 2004b). Moreover, GLLAMM can handle continuous responses, unordered categorical responses, counts, rankings (Skrondal and Rabe-Hesketh 2003), survival data, and mixed responses (Skrondal and Rabe-Hesketh 2004, chap. 14).

6 References


X. Zheng and S. Rabe-Hesketh


**About the author**

Xiaohui Zheng is a Ph.D. student in the Quantitative Methods and Evaluation Program at the Graduate School of Education, University of California, Berkeley.

Sophia Rabe-Hesketh is a professor at the Graduate School of Education and Graduate Group in Biostatistics, University of California, Berkeley. She is also Chair of Social Statistics at the Institute of Education, University of London, UK.