Maximum likelihood estimation of endogenous switching and sample selection models for binary, ordinal, and count variables

Alfonso Miranda
School of Economic and Management Studies
Keele University
Keele, UK
a.miranda@econ.keele.ac.uk

Sophia Rabe-Hesketh
Graduate School of Education
University of California
Berkeley, CA, and
Institute of Education
University of London
London, UK

Abstract. Studying behavior in economics, sociology, and statistics often involves fitting models in which the response variable depends on a dummy variable—also known as a regime-switch variable—or in which the response variable is observed only if a particular selection condition is met. In either case, standard regression techniques deliver inconsistent estimators if unobserved factors that affect the response are correlated with unobserved factors that affect the switching or selection variable. Consistent estimators can be obtained by maximum likelihood estimation of a joint model of the outcome and switching or selection variable. This article describes a “wrapper” program, ssm, that calls gllamm (Rabe-Hesketh, Skrondal, and Pickles, GLLAMM Manual [University of California–Berkeley, Division of Biostatistics, Working Paper Series, Paper No. 160]) to fit such models. The wrapper accepts data in a simple structure, has a straightforward syntax, and reports output that is easily interpretable. One important feature of ssm is that the log likelihood can be evaluated using adaptive quadrature (Rabe-Hesketh, Skrondal, and Pickles, Stata Journal 2: 1–21; Journal of Econometrics 128: 301–323).

Keywords: st0107, endogenous switching, sample selection, binary variable, count data, ordinal variable, probit, Poisson regression, adaptive quadrature, gllamm, wrapper, ssm

1 Introduction

Endogenous switching (ES) and sample selection (SS) are among the most common problems in economics, sociology, and statistics. ES is a concern whenever the dependent variable of a model is a function of a binary regime switch, whereas SS is a concern whenever the response variable is observed only if a selection condition is met. In either case, problems arise because standard regression techniques result in biased and inconsistent estimators if unobserved factors affecting the response are correlated with unobserved factors affecting the switch/selection process (Heckman 1978, 1979). Studies on smoking and drinking behavior, for instance, suggest that completing a higher-education degree may be endogenous because impatient individuals are both
more likely to engage in health-damaging behavior and less likely to invest in human
capital accumulation (Miranda and Bratti 2006). Similarly, given that women who do
not work have no wage information, estimating female wage equations is based on sam-
ples of women who actually do work. Therefore, unobserved factors affecting wage and
participation status may be correlated, and consistent estimation requires using an SS
model (Vella 1998).

For strictly continuous outcome variables, simple two-stage regression strategies have
been developed to address these problems (Heckman 1978, 1979). For binary, count,
and ordinal responses, however, accounting for SS or ES is essentially complicated by the
fact that a nonlinear model is used to fit the data. Then two-stage procedures analogous
to the Heckman (1979) method are only approximate and no appropriate distribution
results for the estimators are available. Hence, inference based on such procedures
may lead to wrong conclusions (Heckman 1978; de Ven and Praag 1981; Wooldridge
2002). Maximum likelihood (ML) techniques or two-stage method of moments is there-
fore needed.

Stata’s [R] heckman and [R] heckprob commands provide ML estimation for linear
and probit regression with SS, respectively. However, there are currently no analogous
commands for ordinal or count outcomes. Stata has several commands ([R] ivreg,
[R] ivprobit, [R] ivtobit) for ML estimation of models with continuous endogenous
regressors and has one command ([R] treatreg) for a continuous outcome with an en-
dogenous dummy variable (the ES problem). However, there are currently no commands
for noncontinuous outcomes with an endogenous dummy variable.

The Stata program gllamm (Rabe-Hesketh, Skrondal, and Pickles 2004a) can be
used to fit switching or selection models for all sorts of outcomes by ML (see also
Rabe-Hesketh, Skrondal, and Pickles 2002a). However, preparing the data and spec-
ifying the correct syntax require much expertise. Perhaps because of the lack of readily
available and easy-to-use software, ES and SS issues are often ignored whenever the
outcome variable is a count or an ordinal response.

This article describes a “wrapper” program, ssm, that calls gllamm to fit the model.
The wrapper accepts data in a simple structure, has a straightforward syntax, and
reports easily interpretable output. One important feature of gllamm is that the log
likelihood can be evaluated using adaptive quadrature (Rabe-Hesketh, Skrondal, and
Pickles 2002b, 2005). Another gllamm wrapper, cme for covariate measurement error
models, is described in Rabe-Hesketh, Skrondal, and Pickles (2003).

2 Binary variables

We start by discussing the case where the main outcome, \( y \), is a dichotomous variable.
By assumption, the switching/selection variable, \( S \), is also a binary variable.
2.1 The ES problem

In the ES problem, the response $y_i$ of the $i$th individual is always observed. Moreover, $y_i$ is assumed to depend on the endogenous dummy $S_i$ and a $K \times 1$ vector of explanatory variables (including the constant term), $x_i$. Similarly, the endogenous dummy $S_i$ depends on an $L \times 1$ vector of explanatory variables (including the constant term), $z_i$. No exclusion restrictions are needed to identify the model (Heckman 1978; Wilde 2000). As a consequence, vectors $x_i$ and $z_i$ may contain identical elements. It is good practice, however, to specify at least one exclusion restriction.

The model can be formulated as a system of equations for two latent (i.e., unobserved) responses. In particular, $y_i$ is assumed to be generated as

$$y_i^* = x_i' \beta + \theta S_i + u_i$$

where $y_i^*$ represents a latent continuous variable, $\beta$ represents a $K \times 1$ vector of parameters to be estimated, $\theta \in \mathbb{R}$ is the coefficient associated with the endogenous dummy, and $u_i$ is a residual term. A similar latent response model is specified for the switching dummy,

$$S_i^* = z_i' \gamma + v_i$$

where, as before, $S_i^*$ represents a latent continuous variable, $\gamma$ an $L \times 1$ vector of parameters, and $v_i$ a residual term.

Typically, a bivariate normal distribution is assumed for $u_i$ and $v_i$. To fit the model in `gllamm`, we must use a shared random effect, $\varepsilon_i$, to induce the dependence between $u_i$ and $v_i$,

$$u_i = \lambda \varepsilon_i + \tau_i$$

$$v_i = \varepsilon_i + \zeta_i$$

Here $\varepsilon_i$, $\tau_i$, and $\zeta_i$ are independently normally distributed with mean 0 and variance 1, and $\lambda$ is a free parameter (a factor loading). The covariance matrix of the residuals is given by

$$\text{Cov}\{(u_i, v_i)\} \equiv \Sigma = \begin{pmatrix} \lambda^2 + 1 & \lambda \\ \lambda & 2 \end{pmatrix}$$

so that the correlation is

$$\rho = \frac{\lambda}{\sqrt{2(\lambda^2 + 1)}}$$

There is only one free parameter, $\lambda$, which is identified because the data provide information on the correlation, $\rho$. There are, however, no free parameters for the variances since these are not identified for probit models.
The above parameterization differs from that usually used for bivariate probit models where the variances are set to 1. To convert the above model to the usual parameterization, we can think of dividing (1) for \( y^*_i \) by \( \sqrt{\lambda^2 + 1} \) and (2) for \( S^*_i \) by \( \sqrt{2} \), i.e., we must rescale all the estimated regression coefficients and use the delta method to obtain correct standard errors.

\( S_i \) is exogenous in (1) if \( \rho = 0 \). Consistent estimators of \( \beta \) and \( \theta \) are then obtained by fitting model (1) with ordinary probit regression; see [\( R \) probit]. If \( \rho \neq 0 \), however, this approach delivers inconsistent estimators because \( S_i \) is correlated with \( u_i \) via the unobserved heterogeneity term \( \varepsilon_i \). The presence of this bias is why one should use an ES model if \( S_i \) is suspected to be endogenous.

We may want to specify a logit model for \( y_i \) instead of a probit model. This task requires the error term \( u_i \) to be distributed as a logistic variate with variance \( \pi^2 / 3 \). However, in gllamm the shared random effect \( \varepsilon_i \) in (3) and (4) can be specified only as normal or discrete, so that even if \( \tau_i \) has a logistic distribution, \( u_i \) will not unless \( \lambda = 0 \). The main difference between probit and logit models is in the scaling of the parameters due to the difference in the residual variance (1 for probit and \( \pi^2 / 3 \) for logit). An approximate logit model can therefore be obtained by specifying a logistic distribution for \( \tau_i \) and changing the scale factor for \( \beta \) and \( \theta \) from \( 1/\sqrt{\lambda^2 + 1} \) to \( \sqrt{\pi^2 / 3 / \lambda^2 + \pi^2 / 3} \).

To fit the model in gllamm, all responses (\( y_i \) and \( S_i \)) must be stacked in one variable, \( q_{ji} \). Viewing the main response \( (j = 1) \) and the switching dummy \( (j = 2) \) as clustered within individuals, define the dummies \( d_{1ji} = 1 \) if \( j = 1 \) and \( d_{2ji} = 1 \) if \( j = 2 \). Then specify \( q_{ji} \) as having a Bernoulli (or binomial) distribution for both \( j = 1 \) and \( j = 2 \). Finally, specify a model for the conditional mean of \( q_{ji} \), \( E(q_{ji} | \varepsilon_i) \equiv \pi_{ji} \) as

\[
\eta_{ji} = g_j(\pi_{ji}) = d_{1ji}(x'_i\beta + \theta S_i + \lambda \varepsilon_i) + d_{2ji}(z'_i\gamma + \varepsilon_i)
\]

(5)

where \( g_j(\cdot) \) represents the link function for response \( q_{ji} \). As discussed before, \( g_2(\cdot) \) is restricted to be the probit link, whereas \( g_1(\cdot) \) can be either the probit link or the logit link.

The model is fitted by ML. To evaluate the likelihood, the unobserved heterogeneity term, \( \varepsilon_i \), must be integrated out. To do so, gllamm uses either ordinary Gauss–Hermite quadrature or adaptive quadrature (Rabe-Hesketh, Skrondal, and Pickles 2005). In each iteration of a Newton–Raphson algorithm, adaptive quadrature modifies the locations and weights of the Gauss–Hermite quadrature points by using the posterior distribution of \( \varepsilon_i \). The procedure delivers locations that are centered at the mean of the posterior distribution and spread out according to the posterior standard deviation. Adaptive quadrature has proven to achieve good accuracy with fewer points than Hermite–Gauss quadrature—possibly a major advantage whenever computing power is a relevant issue (Rabe-Hesketh, Skrondal, and Pickles 2002b). After estimation, a simple likelihood-ratio test can be used to test the null hypothesis that \( \rho = 0 \).
2.2 The SS problem

In the SS problem, the main outcome variable $y_i$ is observed only if a selection condition ($S_i = 1$) is met. The researcher always observes whether an individual has been selected to the sample ($S_i = 1$) or not ($S_i = 0$).

The SS model for dichotomous variables can easily be written as a system of equations for two latent variables, just like those presented in (1) and (2). Simply write

$$y_i^* = x_i'\beta + \lambda \varepsilon_i + \tau_i$$

$$S_i^* = z_i'\gamma + \varepsilon_i + \zeta_i$$

Equation (6) differs from (1) only by the absence of $S_i$ (the coefficient of $S_i$ would not be identified since $S_i = 1$ whenever $y_i$ is observed).

A mixed-response variable $q_{ji}$ may be created as in section 2.1. The model is as before except that $S_i$ is omitted:

$$\eta_{ji} = g_j(\pi_{ji}) = d_{1ji}(x_i'\beta + \lambda \varepsilon_i) + d_{2ji}(z_i'\gamma + \varepsilon_i)$$

If $\lambda = 0$ (so that $\rho = 0$), the individuals are randomly selected to the sample and consistent estimators of $\beta$ are obtained by estimating (6) using ordinary probit or logit regression.

3 Ordinal variables

In several contexts, the researcher must fit ES or SS models to ordinal variables. In those cases, the variable of interest, $y$, takes on $H$ response categories $y_h$, $h = 1, \ldots, H$. Moreover, categories are ordered,

$$y_1 < y_2 < \cdots < y_H$$

but the difference between any pair of categories has no cardinal interpretation. Examples of ordinal variables include responses to health status questions (excellent, good, bad), opinions of a candidate in an election (strongly support, neutral, strongly opposed), and answers to job satisfaction questions (highly satisfied, satisfied, nonsatisfied).

As for dichotomous variables, latent variable models can be used here. In particular, the latent response $y_i^*$ for the $i$th individual is assumed to be determined according to

$$y_i^* = x_i'\beta + \theta S_i + \lambda \varepsilon_i + \tau_i$$

in an ES problem, or according to

$$y_i^* = x_i'\beta + \lambda \varepsilon_i + \tau_i$$

in an SS framework. As before, $\lambda$, $\varepsilon_i$, and $\tau_i$ represent a factor loading, an unobserved heterogeneity term, and a random error, respectively. However, unlike (1) and (6), the
290  **ML estimation of endogenous switching and sample selection models**

vector of explanatory variables, \( x_i \), in (9) and (10) does not include the constant term. Instead, a threshold model determines the observed response,

\[
y_i = \begin{cases} 
  y_1 & \text{if } -\infty < y_i^* \leq \kappa_1 \\
  y_2 & \text{if } \kappa_1 < y_i^* \leq \kappa_2 \\
  \vdots & \vdots \\
  y_H & \text{if } \kappa_{H-1} < y_i^* \leq \infty 
\end{cases}
\]

where \( \kappa_s, s = 1, \ldots, H - 1 \) represent threshold parameters. The model for the switching or selection dummy remains as in (7).

The mixed-response model supposes that \( q_{ji} \) is distributed as a multinomial variate if \( j = 1 \) and as a Bernoulli variate if \( j = 2 \). For the dichotomous response \( S_i (j = 2) \), the linear predictor \( \eta_i \) determines the conditional probability of a ‘1’ response. In contrast, for the ordinal response \( y_i (j = 1) \), the category-specific linear predictors determine the cumulative probabilities

\[
\Pr(y_i > h|\varepsilon_i, x_i, S_i) \equiv \theta_{hi} = \sum_{s=h+1}^{H} \pi_{si}, \quad h = 1, \ldots, H - 1
\]

where \( \pi_{si} \) represents the conditional probability that \( y_i \) equals \( s \).

For ES, the linear predictor can then be written as

\[
\eta_{jhi} = d_{1ji}(x_i'\beta + \theta S_i - \kappa_h + \lambda \varepsilon_i) + d_{2ji}(z_i'\gamma + \varepsilon_i)
\]

where \( h = 1, \ldots, H - 1 \) when \( j = 1 \) and \( h = 0 \) when \( j = 2 \) [since \( \Pr(S_i = 1|\varepsilon_i, z_i) = \Pr(S_i > 0|\varepsilon_i, z_i) \) with \( \kappa_0 = 0 \). As before, a probit link is always used for \( g_2(\cdot) \), and the researcher can specify \( g_1(\cdot) \) either as the ordered probit link or as the ordered logit link.

The SS model excludes the selection dummy \( S_i \) from the list of conditioning variables in the equation of the expected value of \( q_{ji} \) when \( j = 1 \). That is,

\[
\eta_{jhi} = d_{1ji}(x_i'\beta - \kappa_h + \lambda \varepsilon_i) + d_{2ji}(z_i'\gamma + \varepsilon_i)
\]

In both SS and ES, thresholds \( \{\kappa_1, \ldots, \kappa_{H-1}\} \) are estimated along with the parameters \( \beta, \theta, \) and \( \gamma \). As in the binary case, the parameters \( \beta, \theta, \) and \( \gamma \) must be rescaled after estimation to account for the increased variance in (1) and (7) or (6) and (7), depending on whether an ES or an SS model is being fitted. A simple likelihood-ratio test can be used to test the null hypothesis that \( \rho = 0 \).

**4  Count variables**

We turn now to discuss how the model can be adapted to allow for a count variable. Unlike ordinal responses, count variables can in principle take an infinite number of discrete values, \( 0, 1, \ldots, \infty \), and there is a clear cardinal interpretation of the gap between any pair of such values. Examples of count variables include completed fertility, number
of car accidents, number of visits to a doctor in a month, and number of alcohol units consumed during a week.

Since the variable of interest is a count, a latent variable model is not suitable. We suppose instead that the count \( y_i \) follows a Poisson distribution,

\[
Pr(y_i; \mu_i) = \frac{\mu_i^{y_i} \exp(-\mu_i)}{y_i!}
\]

so that a log-linear model for the mean, \( \mu_i \), can be specified. In the ES model, we write

\[
\ln(\mu_i) = x_i'\beta + \theta S_i + \varepsilon_i
\]

(11)

whereas in SS the assumption is

\[
\ln(\mu_i) = x_i'\beta + \varepsilon_i
\]

(12)

The interpretation of \( \beta, \theta, \) and \( \varepsilon_i \) remain as in previous sections, and the vector of explanatory variables, \( x_i \), contains the constant term. For the switching/selection model, we write

\[
S_i^* = z_i'\gamma + \lambda \varepsilon_i + \zeta_i
\]

\[
S_i = \begin{cases} 1 & \text{if } S_i^* > 0 \\ 0 & \text{otherwise} \end{cases}
\]

(13)

As usual, \( \zeta_i \sim N(0,1) \) and independent of \( \varepsilon_i \). The model is identified by functional form (Kenkel and Terza 2001). Therefore, vectors \( x_i \) and \( z_i \) may contain the same elements.

Unlike models for binary and ordinal variables, here the researcher does not need to set the variance of the unobserved heterogeneity term \( \varepsilon_i \) to a constant. The variance of \( \varepsilon_i \) determines the amount of overdispersion in the counts (see below) and is hence identified. We therefore have another parameter, \( \sigma^2 = \text{Var}(\varepsilon_i) \). The total variance in the switching/selection model is then \( \lambda^2 \sigma^2 + 1 \). Clearly, this a reparameterization of the models described by Terza (1998), where this variance is set to 1. We can convert our estimates to that parameterization by dividing the regression coefficients in the switching/selection model by \( \sqrt{\lambda^2 \sigma^2 + 1} \).

Miranda (2004) describes how these models can be fitted using full information ML methods outside the glamm context—see help espoisson.

Although a Poisson distribution is used, the variance of the count \( y \), given the covariates, is not necessarily equal to the conditional mean. In fact, the model allows for overdispersion (Winkelman 2000). In general,

\[
\text{Var}(y_i|x_i, S_i) = E\{\text{Var}(y_i|\varepsilon_i, x_i, S_i)\} + \text{Var}\{E(y_i|\varepsilon_i, x_i, S_i)\}
\]

Hence, after some manipulation of (11) and using the normality assumption for \( \varepsilon_i \),

\[
\text{Var}(y_i|x_i, S_i) = E(y_i|\varepsilon_i, x_i, S_i) \left[ 1 + E(y_i|\varepsilon_i, x_i, S_i) \{\exp(\sigma^2) - 1\} \right]
\]
which implies that if $\sigma \neq 0$ the model exhibits overdispersion. Unlike models in sections 2 and 3, the factor loading $\lambda$ is introduced in the switching/selection equation here; see (13). This parameterization is convenient in the count data context because it allows the model to exhibit overdispersion ($\sigma \neq 0$) even when $S_i$ is found to be exogenous ($\rho = 0$).

The equivalent mixed-response model is obtained by assuming that $q_{ji}$ is distributed as a Poisson variate with a log link $g_1(\cdot)$ if $j = 1$ and as a binomial variate with a probit link $g_2(\cdot)$ if $j = 2$. As usual, in an ES model the linear predictor is

$$\eta_{ji} = d_{1ji} (x_i' \beta + \theta S_i + \varepsilon_i) + d_{2ji} (z_i' \gamma + \lambda \varepsilon_i)$$

whereas in an SS model $q_{ji}$ has conditional mean

$$\eta_{ji} = d_{1ji} (x_i' \beta + \varepsilon_i) + d_{2ji} (z_i' \gamma + \lambda \varepsilon_i)$$

## 5 The ssm command

### Syntax for ssm

```
ssm depvar [ indepvars ] [ if ] [ in ] [ weight ], switch(varname = varlist)
   family(familyname) link(linkname) [ quadrature(#) selection noconstant
       adapt robust commands nolog trace from(matrix) ]
```

fweights and pweights are allowed; see [u] 11.1.6 weight.

The outcome model is specified by

```
   depvar [ indepvars ], family(familyname) link(linkname)
```

This model is fitted in gllamm as a generalized linear model that contains an endogenous dummy among its observed covariates and an unobserved or latent random term. The switch model is a binary probit model that contains an unobserved random term that is correlated with the unobserved random term in the outcome model. The switching equation is specified by `switch(varname = varlist)`, where varname is the name of the endogenous dummy and varlist is a set of explanatory variables.

ES models are the default specification. SS models are obtained when the outcome is observed only if a selection condition is met and the selection dummy does not enter the outcome model. SS models are fitted when the `selection` option is used.
The following families and links are accepted:

<table>
<thead>
<tr>
<th>Family</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>poisson</td>
<td>log</td>
</tr>
<tr>
<td>binomial</td>
<td>logit</td>
</tr>
<tr>
<td></td>
<td>probit</td>
</tr>
<tr>
<td></td>
<td>ologit</td>
</tr>
<tr>
<td></td>
<td>oprobit</td>
</tr>
</tbody>
</table>

**Options**

- `switch(varname = varlist)` specifies the switching equation, where `varname` is the name of the endogenous dummy and `varlist` is a set of explanatory variables.
- `family(familyname)` specifies the distribution of `depvar`.
- `link(linkname)` specifies the link function.
- `quadrature(#)` specifies the number of quadrature points to be used.
- `selection` requests that an SS model be fitted instead of the default ES model.
- `noconstant` specifies that the linear predictor has no intercept term, thus forcing it through the origin on the scale defined by the link function.
- `adapt` specifies that adaptive quadrature is to be used instead of the default ordinary quadrature.
- `robust` specifies that the Huber/White/sandwich estimator of variance is to be used. If `pweights` are specified, `robust` is implied.
- `commands` displays the commands necessary to prepare the data and fits the model in `gllamm` instead of fitting the model with `ssm`. These commands can be copied into a do-file and should work without further editing. The do-file will change the data.
- `nolog` suppresses the iteration log.
- `trace` requests that the estimated coefficient vector be printed at each iteration. Also all the output produced by `gllamm` with the `trace` option is produced.
- `from(matrix)` specifies a matrix of starting values.
6 Estimation using ssm and glamm

6.1 Sample selection in a probit model

To exemplify using ssm in analyzing dichotomous responses, we discuss a probit model with sample selection here. We use simulated data. For each observation, three independent random draws from a standard normal distribution are taken to create an unobserved individual heterogeneity term, $\varepsilon_i$, and two uncorrelated random disturbances, $\tau_i$ and $\zeta_i$. Four more independent random draws from a standard normal distribution are obtained to generate a set of control variables, $x_1$–$x_4$. Finally, (1) and (6) are used together with a set of reasonable but arbitrary parameters to generate the binary response $y$ and the selection variable $sel$. We suppose that $y$ is a function of $x_1$ and $x_2$ and that the selection mechanism depends on $x_1$, $x_2$, $x_3$, and $x_4$.

To test the behavior of ssm, we generated several other datasets in which the values of the true parameters varied. ssm always produced satisfactory results. To illustrate the procedure, an example of a do-file is reproduced here.

```stata
begin do-file
set seed 12345678
set obs 3500
local lambda = 0.4
gen double ve = invnormal(uniform())
gen double zeta = invnormal(uniform())
gen double tau = invnormal(uniform())
gen double x1=invnormal(uniform())
gen double x2=invnormal(uniform())
gen double x3=invnormal(uniform())
gen double x4=invnormal(uniform())
replace x3 = (x3>0)
replace x4 = (x4>0)
gen double selstar = 0.58 + 0.93*x1 + 0.45*x2 - 0.64*x3 + 0.6*x4 + (ve + zeta)/sqrt(2)
gen sel = (selstar>0)
gen double ystar = 0.17 + 0.30*x1 + 0.11*x2 + ('lambda'*ve + tau)/sqrt(1+'lambda'^2)
gen y = (ystar>0)
replace y = . if sel==0
end do-file
```

The variance of the two composite errors $u_i$ and $v_i$—see (3) and (4)—has been set to unity to obtain the usual parameterization for probit models.

Fitting a probit model—see [R] probit—for the observed sample yields the following results:
. probit y x1 x2 if sel==1
Iteration 0:  log likelihood = -1395.5862
Iteration 1:  log likelihood = -1365.6029
Iteration 2:  log likelihood = -1365.5542
Iteration 3:  log likelihood = -1365.5542
Probit regression  Number of obs = 2210
LR chi2(2)     =  60.06
Prob > chi2    = 0.0000
Log likelihood = -1365.5542  Pseudo R2 = 0.0215

y  Coef.  Std. Err.  z  P>|z|  [95% Conf. Interval]
x1  .2442101  .0331065  7.38  0.000  .1793225  .3090978
x2  .0829626  .0289254  2.87  0.004  .0262699  .1396553
_cons  .351067  .0304893 11.51  0.000  .2913091  .4108249

Two inferences can be drawn from this output table. First, the estimated coefficient of x1 is considerably below its true value of 0.30. This result is in line with the fact that the true value of λ—and therefore ρ—is positive, so the econometrician should expect a negative bias. Second, the estimate for the constant is clearly positively biased.

To fit a probit sample selection model with ssm, the selection option is required together with a binomial family and a probit link.

. ssm y x1 x2, s(sel = x1 x2 x3 x4) q(16) family(binom) link(probit) sel adapt
(output omitted)
Sample Selection Probit Regression
(Adaptive quadrature -- 16 points)
Number of obs = 3500
Wald chi2(6)   = 1021.27
Log likelihood = -2915.0225

y  Coef.  Std. Err.  z  P>|z|  [95% Conf. Interval]
x1  .3770491  .0509866  7.40  0.000  .2771172  .4769811
x2  .1400234  .0331205  4.23  0.000  .0751083  .2049384
_cons  .1274961  .0735357  1.73  0.083  -.0166311  .2716234
selection  Coef.  Std. Err.  z  P>|z|  [95% Conf. Interval]
x1  .9681283  .0342559  28.26  0.000  .9009888  1.035269
x2  .4240503  .027396  15.48  0.000  .370355  .477755
x3  -.5845267  .0619791 -9.15  0.000  -.6664038  -.5026495
x4  .6702432  .0528234  12.69  0.000  .5667113  .7737751
_cons  .4499317  .0430671  10.45  0.000  .3655217  .5343416

Likelihood ratio test for rho=0: chi2(1) =  9.95 Prob > chi2 = 0.002

This model can also be fitted by using the official heckprob command (see [R] heckprob). heckprob reports the following:
ML estimation of endogenous switching and sample selection models

. heckprob y x1 x2, select(sel = x1 x2 x3 x4)
(output omitted)

Probit model with sample selection

<table>
<thead>
<tr>
<th></th>
<th>Number of obs = 3500</th>
<th>Censored obs = 1290</th>
<th>Uncensored obs = 2210</th>
<th>Wald chi2(2) = 55.48</th>
<th>Prob &gt; chi2 = 0.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log likelihood = -2915.022</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|                          | Coef.     | Std. Err. | z     | P>|z|  | [95% Conf. Interval] |
|--------------------------|-----------|-----------|-------|-----|----------------------|
| y                        |           |           |       |     |                      |
| x1                       | .3770487  | .0509867  | 7.40  | 0.000 | .2771166 .4769808    |
| x2                       | .1400232  | .0331206  | 4.23  | 0.000 | .0751081 .2049382    |
| _cons                    | .1274969  | .0735357  | 1.73  | 0.083 | -.0166303 .2716242   |
| sel                      |           |           |       |     |                      |
| x1                       | .9681283  | .0342559  | 28.26 | 0.000 | .9009881 1.035269    |
| x2                       | .4240503  | .027396   | 15.48 | 0.000 | .3703551 .4777456    |
| x3                       | -.584527  | .0519791  | -11.25| 0.000 | -.6864041 -.4826498  |
| x4                       | .6702436  | .0528234  | 12.69 | 0.000 | .5667116 .7737755    |
| _cons                    | .4499316  | .0430671  | 10.45 | 0.000 | .3655216 .5343416    |
| /athrho                  | .4153096  | .1348318  | 3.08  | 0.002 | .1510442 .679575     |
| rho                      | .3929717  | .1140101  | 3.46  | 0.001 | .1699059 .614238     |

LR test of indep. eqns. (rho = 0): chi2(1) = 9.95 Prob > chi2 = 0.0016

The estimates and standard errors from smm and heckprob are nearly identical. Both output tables show that after controlling for selection all coefficients are close to their true values. Further, a positive correlation coefficient $\rho$ is correctly detected.\(^1\)

### 6.2 Sample selection for an ordinal response

Following the strategy of the previous subsection, we use simulated data to illustrate how smm estimates sample selection models for ordinal variables—our previous work (Miranda and Rabe-Hesketh 2005) also discussed this example. Equations (2) and (10) are used to generate the ordinal response ordvar and the selection variable sel.

As before, x1–x4 are included in the selection equation, whereas the main response, ordvar, depends only on x1 and x2. To generate the data, section 6.1’s do-file needs minor changes. Namely, a set of thresholds for the ordinal model should be specified. Here a set of values for the true parameters were chosen such that a reasonable number of observations fall in each of the five categories of ordvar. The modified do-file is

---

1. heckprob reports a Wald test for the exclusion of all explanatory variables in the equation for the main response. In contrast, smm reports a Wald test for the exclusion of all explanatory variables in both main and selection/switch equations.
A. Miranda and S. Rabe-Hesketh

\begin{verbatim}
begin do-file
set seed 12345678
gen double ve = invnormal(uniform())
gen double zeta = invnormal(uniform())
gen double tau = invnormal(uniform())
gen double x1=invnormal(uniform())
gen double x2=invnormal(uniform())
gen double x3=invnormal(uniform())
gen double x4=invnormal(uniform())
gen double selstar = 0.58 + 0.93*x1 + 0.45*x2 - 0.64*x3 + 0.6*x4 + ///
(ve + zeta)/sqrt(2)
gen sel = (selstar>0)
gen double ystar = 0.30*x1 + 0.11*x2 + ///
('lambda'*ve + tau)/sqrt(1+'lambda'^2)
gen ordvar = 0
qui replace ordvar=1 if ystar>-0.40 & ystar<=0.17
qui replace ordvar=2 if ystar>0.17 & ystar<=0.45
qui replace ordvar=3 if ystar>0.45 & ystar<=0.80
qui replace ordvar=4 if ystar>0.80 & ystar<=1.25
qui replace ordvar=5 if ystar>1.25
replace ordvar=. if sel==0
end do-file
\end{verbatim}

Fitting an ordered probit model (see [R] oprobit) to the observed sample (i.e., individuals for which \texttt{sel} = 1), ignoring potential selection bias, produces the following results:

```
. oprobit ordvar x1 x2 if sel==1
(output omitted)
Ordered probit regression
Log likelihood = -3768.4917
```

| ordvar | Coef.  | Std. Err.  | z     | P>|z|  | [95% Conf. Interval] |
|--------|--------|------------|-------|------|----------------------|
| x1     | .2433844| .0255461   | 9.53  | 0.000 | .193315 -.2934538    |
| x2     | .1169096| .0230642   | 5.07  | 0.000 | .0717045 -.1621147   |

| /cut1  | -.5533054| .0302867   |       |      | -.6126662 -.4939446  |
| /cut2  | .0136767 | .0287603   |       |      | -.0426924 .0700458   |
| /cut3  | .2849928 | .0291054   |       |      | .2279473 .3420383    |
| /cut4  | .6002736 | .0306374   |       |      | .5902285 .7103217    |
| /cut5  | 1.096861 | .0345433   |       |      | 1.029157 1.164564    |
```

From this table, the reader can easily conclude that the estimated coefficient of \texttt{x1} is below its true value of 0.30, just like in the example of section 6.1. As before, this negative bias in the “slope” coefficients is what the econometrician should expect given a positive \texttt{rho}. In contrast with the results of section 6.1, however, estimates of the cutpoints are negatively rather than positively biased. Again this outcome is expected because cutpoints and constant are parameterized differently in ordered probit and probit.
ML estimation of endogenous switching and sample selection models

To fit a sample selection model for an ordinal response, one needs a binomial family and an oprobit link. The selection option should also be specified.

```
.ssm ordvar x1 x2, s(sel = x1 x2 x3 x4) q(16) adapt family(binom) link(oprobit)
> selection
```

Sample Selection Ordered Probit Regression
(Adaptive quadrature -- 16 points)

| Coef. | Std. Err. | z | P>|z| | [95% Conf. Interval] |
|-------|-----------|---|-----|------------------|
| ordvar |           |   |      |                  |
| x1     | .3154251  | .0292514 | 10.78 | 0.000 | .2580934 .3727568 |
| x2     | .1470225  | .0236525 | 6.22  | 0.000 | .1006644 .1933806 |
| selection |           |   |      |                  |
| x1     | .9573865  | .0356374 | 26.86 | 0.000 | .8875385 1.027234 |
| x2     | .4217439  | .0287655 | 14.71 | 0.000 | .365541  .4779468 |
| x3     | -.5968153 | .0303954 | -19.64| 0.000 | -.6563892 -.5372414 |
| x4     | .6372245  | .0308598 | 20.65 | 0.000 | .5767403 .6977087 |
| _cons  | .5448698  | .0288654 | 18.88 | 0.000 | .4882947 .6014449 |
| aux_ordvar |       |   |      |                  |
| _cut1  | -.4012285 | .0460499 | -8.71 | 0.000 | -.4914846 -.3109724 |
| _cut2  | .1853415  | .0419442 | 3.78  | 0.000 | .0761324 .2405506 |
| _cut3  | .4265045  | .0409127 | 10.42 | 0.000 | .3463171 .5068919 |
| _cut4  | .7873888  | .0404785 | 19.45 | 0.000 | .7080523 .8667253 |
| _cut5  | 1.229029  | .0420109 | 29.26 | 0.000 | 1.146689 1.311369 |
| rho    | .3181846  | .0688199 | 4.62  | 0.000 | .1624427 .4320025 |

After controlling for nonrandom selection, the coefficients on x1 and x2 in the ordinal probit are more in line with their true values. Similarly, ρ is estimated to be 0.32, a number reasonably close to the true parameter value of ρ = 0.26 = 0.4/√(0.4² + 1). As expected, a likelihood-ratio test for ρ = 0 rejects the null hypothesis at a significance level of 1%.

6.3 ES for a count

We use data from Kenkel and Terza (2001) to illustrate estimating an ES model with count data. In fact, we follow the simplified example discussed in chapter 14 of Skrondal and Rabe-Hesketh (2004). The data are a subsample of 2,467 individuals from the 1990 National Health Interview Survey core questionnaire and special supplements. All subjects are males who currently drink and have been told that they have hypertension. Kenkel and Terza are interested in studying the determinants of the number of alcoholic beverages consumed in the last 2 weeks, drinks. In particular, 687 individuals (28%) have been advised by a physician to reduce drinking, and the main objective is to estimate the causal effect of receiving such advice.
A. Miranda and S. Rabe-Hesketh

The main challenge of estimating the causal effect of advice on drinking is that advice may be endogenous: unobservables in the drinking equation may be correlated with unobservables in the advice equation. Kenkel and Terza point out, for example, that “health-minded individuals may have a higher than average propensity to seek advice, and a simultaneously higher than average propensity to avoid unhealthy behaviors like heavy drinking” (2001, 168). Neglecting the potential self-selection to treatment (advice) in the drinking equation may therefore result in biased and inconsistent estimators. The count drinks also appears to exhibit overdispersion since its unconditional mean is 15, whereas its unconditional variance is 23.

The ES Poisson model described in section 4 is used. The variables in the drinking equation are the following:

1. advice. Dummy variable for individual having been advised to reduce consumption of alcoholic beverages.
2. black. Dummy variable for individual being black.
3. hieduc. Dummy variable for individual having more than 12 years of education.

For the advice model, besides black and hieduc, the following controls are included:

1. hlthins. Dummy variable for individual having health insurance.
2. regmed. Dummy variable for individual having a registered source of medical care.
3. heart. Dummy variable for individual suffering from heart disease.

Now if advice is assumed to be exogenous in the drinking equation, one may use the standard poisson command to fit the model; see [R] poisson.

\[
\text{. poisson drinks advice black hieduc}
\]

|          | Coef.  | Std. Err. | z     | P>|z|   |  [95% Conf. Interval]    |
|----------|--------|-----------|-------|-------|-------------------------|
| drinks   |        |           |       |       |                         |
| advice   | .473367| .010918   | 43.36 | 0.000 | .4519682 – .4947659     |
| black    | -.3096865| .0168905| -18.33 | 0.000 | -.3427913 – -.2765817   |
| hieduc   | -.1826093| .0107983| -16.91 | 0.000 | -.2037736 – -.1614451   |
| _cons    | 2.650541| .0084928 | 312.09 | 0.000 | 2.633896 – 2.667187     |

Similarly, the advice equation would be fitted by a simple probit model; see [R] probit.
probit advice black hieduc hlthins regmed heart

Iteration 0:  log likelihood = -1459.2504
Iteration 1:  log likelihood = -1419.9530
Iteration 2:  log likelihood = -1419.9041
Iteration 3:  log likelihood = -1419.9041

Probit regression
Number of obs = 2467
LR chi2(5) = 78.69
Prob > chi2 = 0.0000
Log likelihood = -1419.9041
Pseudo R2 = 0.0270

|         | Coef. | Std. Err. | z     | P>|z|  | [95% Conf. Interval] |
|---------|-------|-----------|-------|------|---------------------|
| black   | 0.3031406 | 0.0780889 | 3.88  | 0.000 | 0.1500891 to 0.4561921 |
| hieduc  | -0.2520195 | 0.0560241 | -4.50 | 0.000 | -0.3618247 to -0.1422143 |
| hlthins | -0.2708712 | 0.0704249 | -3.85 | 0.000 | -0.4089013 to -0.132841 |
| regmed  | 0.1801329 | 0.0738763 | 2.44  | 0.015 | 0.0353279 to 0.3249278 |
| heart   | 0.1661613 | 0.0757854 | 2.19  | 0.028 | 0.0176246 to 0.314698 |
| _cons   | -0.4785404 | 0.0849039 | -5.64 | 0.000 | -0.649499 to -0.312138 |

If advice is endogenous in the drinking equation, ssm may be used to fit the required ES model.
A. Miranda and S. Rabe-Hesketh

. ssm drinks advice black hieduc, s(advice = black hieduc hlthins regmed heart)
> adapt q(16) family(poiss) link(log)
(output omitted)
Iteration 13: log likelihood = -10254.332
Adaptive quadrature has converged, running Newton-Raphson
Iteration 0: log likelihood = -10254.332
Iteration 1: log likelihood = -10254.332
Iteration 2: log likelihood = -10254.328
Iteration 3: log likelihood = -10254.328
Endogenous Switch Poisson Regression
(Adaptive quadrature -- 16 points)

Number of obs = 2467
Wald chi2(8) = 476.70
Log likelihood = -10254.328 Prob > chi2 = 0.0000

| Coef. Std. Err.  z    P>|z|    [95% Conf. Interval] |
|-----------------|-----------------|-----------------|-----------------|
| drinks          |                 |                 |                 |
| advice          | -2.413655       | .2324534        | -10.38          | 0.000          | -2.869255       | -1.958054        |
| black           | .1102961        | .1445246        | 0.76            | 0.445          | -.172967        | .3935592         |
| hieduc          | -.284437        | .0990839        | -2.87           | 0.004          | -.4786378       | -.0902362        |
| _cons           | 2.326277        | .0944165        | 24.64           | 0.000          | 2.141224        | 2.511329         |

| switch          |                 |                 |                 |
| black           | .3240436        | .0777796        | 4.17            | 0.000          | .1715984        | .4764888         |
| hieduc          | -.2139965       | .0555348        | -3.85           | 0.000          | -.3228428       | -.1051503        |
| hlthins         | -.1753961       | .0556584        | -3.15           | 0.002          | -.2844846       | -.0663076        |
| regmed          | .2066055        | .0566274        | 3.65            | 0.000          | .0956179        | .317532         |
| heart           | .2724067        | .0603824        | 4.48            | 0.000          | .1531774        | .3918361        |
| _cons           | -.5990564       | .0721062        | -8.31           | 0.000          | -.7403908       | -.4577299       |

| sigma           | 2.249861        | .0908619        | 24.76           | 0.000          | 2.071765        | 2.427937        |
| rho             | .8440512        | .0153186        | 55.10           | 0.000          | .8140272        | .8740752        |

Likelihood ratio test for rho=0: chi2(1)= 3099.76 Prob>chi2 = 0.0000

If $\rho = 0$ there is exogenous switching (EXS). Then the log likelihood of the ES model is simply the sum of the log likelihood of a Poisson model for drinks and a probit model for advice, which implies that EXS is nested within ES and a likelihood-ratio test can be used for testing $\rho = 0$. The likelihood-ratio test comparing the EXS with the ES model is highly significant ($\chi^2 = 3,099.76, p < 0.0001$).

The output table also reports an estimate for the standard deviation of $\varepsilon_i$. Though in a preliminary examination the econometrician may test for $\sigma = 0$ on the basis of a simple $t$ statistic, a boundary-value likelihood-ratio test should be used—such a procedure will properly account for the fact that under the null $\sigma$ lies on the boundary of the parameter space. Given that whenever $\sigma = 0$ the model collapses to an EXS framework, the $\chi^2$ reported by ssm can be used for these purposes. The test statistic is distributed as a 50:50 mixture of a $\chi^2_0$ and a $\chi^2_1$ variate (Chernoff 1954; Self and Liang 1987). Here $\sigma = 0$ is rejected at all conventional levels of significance.

Comparing results, the reader may learn that if the endogeneity of advice is neglected, advice appears to increase the consumption of alcoholic beverages by
ML estimation of endogenous switching and sample selection models

\[
\exp(0.47) - 1 \times 100 = 60\%.
\]

However, once potential self-selection is allowed, advice appears to reduce consumption by \(\exp(-2.4) - 1 \times 100 = -91\%\). A similar story applies to variable \texttt{black}, whose coefficient goes from negative and significant in the Poisson regression to positive and insignificant in the ES Poisson regression.

These results show that neglecting the potential endogeneity of a dummy variable may result in serious bias. In extreme cases like the one discussed here, the bias can be large enough to reverse the sign of one or more coefficients.

### 6.4 The commands option

For researchers who like to have full control and fit the model using \texttt{gllamm} directly, the \texttt{commands} option of \texttt{ssm} provides support by helping the user to prepare the data and issuing the appropriate \texttt{gllamm} command. The \texttt{commands} option causes \texttt{ssm} to return an output do-file that the researcher can then run to fit the corresponding model with \texttt{gllamm}. The user should save the data before running the output do-file, as the data will be changed irreversibly. Using the simulated data of section 6.2 with the \texttt{commands} option, \texttt{ssm} generates the following output:

```
* Select sample
mark touse
* Deal with frequency weights
gen one=1
collapse (sum) wt2=one, by(ordvar x1 x2 sel x1 x2 x3 x4 touse)
gen id=_n
#delimit ;
keep id ordvar x1 x2 sel x1 x2 x3 x4 wt2 touse;
#delimit cr
* Expand data
gen vartype1=1
gen vartype2=2
reshape long vartype, i(id)
gen cv=cond(vartype==1,1,0)
gen end=cond(vartype==2,1,0)
gen cons_c=cv
gen cons_d=end
gen cons=1
* Create new variables
gen x1_c=x1
gen x2_c=x2
gen x1_d=x1
gen x2_d=x2
gen x3_d=x3
gen x4_d=x4
```

* Replace zeros where needed
replace x1_c =0 if cv==0
replace x2_c =0 if cv==0
replace x1_d =0 if end==0
replace x2_d =0 if end==0
replace x3_d =0 if end==0
replace x4_d =0 if end==0
* Response
gen resp=ordvar
replace resp=sel if end==1
* Select relevant sample
#delimit ;
markout touse resp x1_c x2_c x1_d x2_d x3_d x4_d cons_d;
keep if touse;
#delimit cr
* Initial values
#delimit ;
matrix startv = (.2434, .1169, .9548, .4197, -.5959, .6386, .544, -.5533, .01368, .285, .6503, 1.097, .5, .5);
#delimit cr
* Estimation
eq fac: end cv
constraint def 1 [id1_1]end=1
* call gllamm:
#delimit ;
gllamm resp x1_c x2_c x1_d x2_d x3_d x4_d cons_d, i(id) weight(wt)
constraints(1) from(startv) long family(binom binom) nrf(1)
link(oprobit probit) fv(vartype) lv(vartype)
eq(fac) adapt nip(16) copy;
#delimit cr

The do-file starts by selecting the sample. Then frequency and probability weights are dealt with. To maximize speed, the data are collapsed and frequency weights are adjusted so that the total likelihood of the sample remains unchanged—this process does not reduce the number of observations in this example. Next the data are reshaped to long form and the variable vartype is generated. vartype indicates whether an observation contains information for the main response, vartype = 1, or for the selection/switch variable. After the data are reshaped, each individual in the sample will contribute 2 observations, one corresponding to the ordinal variable ordvar and the other corresponding to the SS variable. Two dummy variables, cv and end, are created to indicate the groups defined by vartype. Finally, a constant term for each equation and an overall constant are created.

Once the data have been reshaped, a set of new variables is generated to reflect that controls in the main equation (indexed by the suffix _c) may be different from the controls in the selection/switch equation (indexed by the suffix _d). Next zeros are replaced in _c variables if vartype = 2 to reflect that _c controls affect only the likelihood of the ordinal response. A similar approach is taken with the _d variables. This step concludes data preparation. ssm then provides the user with a set of starting
values in matrix startv. The starting values for the ordinal response are obtained by regressing ordv on x1 and x2, using the observed sample and an ordered probit model. Similarly, starting values for the selection equation are obtained on the basis of a probit model of sel on x1–x4.

The following lines in the do-file deal with the dependent variable resp, which is simply defined as ordvar if vartype = 1 and sel otherwise.

Next the gllamm model is set up. First, an equation, fac, specifies that a random intercept, $\epsilon_i$, at level 2 will interact with the dummies end and cv. By default, the factor loading for the first variable is set to 1. Second, the variance of $\epsilon_i$ is constrained to one by specifying constraint.

The user is now ready to call gllamm. After the estimation command, the response variable and all controls are listed—first controls for the ordinal response and then controls for the selection variable. Only the constant term for the selection model has been included because gllamm should be left free to fit the cutpoints in the ordered probit model. After listing the dependent and independent variables, the reader should indicate the name of the variable that indexes individuals by using the i() option. Then the weight(), constraints(), and from() options take care of specifying, respectively, the variable containing frequency weights, the name of any constraint imposed on the parameters (here constraint 1), and a matrix of initial values. The long option allows gllamm to accept a matrix of initial values with entries for parameters constrained by constraint 1.

Finally, the do-file defines the family and link functions to be used. Since a mixed-response model is being fitted, gllamm needs two entries in its family() option. The first entry will indicate how resp is distributed when vartype = 1 and the second entry will indicate how resp is distributed when vartype = 2. Since resp is distributed as a multinomial if vartype = 1 and as a binomial if vartype = 2, a binomial family is used in both cases. Next the link function is specified. As expected, an oprobit link is used when vartype = 1, and a probit link is used when vartype = 2. Options fv() and lv() specify that vartype defines what family and link is to be used with each observation. Option eqs() specifies the name of the equations that the user has previously defined to allow interactions between covariates and latent variables. Finally, option nrf() establishes the number of random effects at each level, here 1. The options nip(16) and adapt indicate that 16 quadrature points are to be used initially to evaluate the log likelihood and that an adaptive quadrature approach should be implemented. For more information on gllamm syntax, see Rabe-Hesketh, Skrondal, and Pickles (2002b, 2004a) and Rabe-Hesketh and Skrondal (2005).
Running the do-file generated by ssm produces the following gllamm output:

number of level 1 units = 5693
number of level 2 units = 3500

Condition Number = 18.02148

gllamm model with constraints:
  ( 1) [id1_1]end = 1

log likelihood = -5175.576549003807

|        | Coef. | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|--------|-------|-----------|-------|------|----------------------|
| resp   |       |           |       |      |                      |
| x1_c   | .3532042 | .0456895  | 7.73  | 0.000 | .2636545 .4427539   |
| x2_c   | .1646317 | .0301073  | 5.47  | 0.000 | .1056225 .223841    |
| x1_d   | 1.359349    | .0503939   | 26.86 | 0.000 | 1.255169 .452729    |
| x2_d   | .596436     | .0405533   | 14.71 | 0.000 | .516953 .6759189    |
| x3_d   | -.8440244  | .0429856   | -19.64 | 0.000 | -.928274 -.7597741  |
| x4_d   | .9011716   | .0436424   | 20.65 | 0.000 | .8156341 .9867092   |
| cons_d | .7705623 | .0408218  | 18.88 | 0.000 | .6905529 .8505716   |

| _cut11 |       |           |       |      |                      |
|        | cons  | -.4492848 | .0365023 | -12.31 | 0.000 | -.520828 -.377741 |

| _cut12 |       |           |       |      |                      |
|        | cons  | .1773062  | .0545316 | 3.25   | 0.001 | .0704263 .2841862  |

| _cut13 |       |           |       |      |                      |
|        | cons  | .4775877  | .0670557 | 7.12   | 0.000 | .346161 .6090145   |

| _cut14 |       |           |       |      |                      |
|        | cons  | .881696   | .085523  | 10.31  | 0.000 | .7140739 1.049318  |

| _cut15 |       |           |       |      |                      |
|        | cons  | 1.376232  | .1098664 | 12.53  | 0.000 | 1.160898 1.591567  |

Variances and covariances of random effects
------------------------------------------

***level 2 (id)

  var(1): 1 (0)

  loadings for random effect 1
  end: 1 (fixed)
  cv: .50387535 (.13665217)

Estimates for _c variables are coefficients on controls for the ordinal response, whereas estimates for _d variables are coefficients on controls for the selection dummy. Compared with usual probit estimates, the _c coefficients have increased by a factor of \( \sqrt{1 + \lambda^2} \) and the _d coefficients have increased by a factor of \( \sqrt{2} \). Similarly, all cutpoints increased by
a factor of $\sqrt{1 + \hat{\lambda}^2}$. The user must rescale the coefficients before interpreting results; otherwise, marginal effects will be mistakenly large. At the bottom of the output table, *gllamm* contains a panel with the estimates of variances and covariances of the random effects. There it is noted that variable *id* identifies level-2 units—in this case, individuals. Then the variance of the random intercept at level 2 is listed. Here the variance has been restricted to one. Finally, estimates for the factor loadings are reported. As discussed earlier, the factor loading for *end* has been set to unity to ensure that the model is properly identified. Finally, the factor loading for the dummy *cv* is reported. This coefficient is the estimate for $\lambda$ in (10). According to *gllamm* output, $\hat{\lambda} = 0.50$ with a standard error of 0.14. Clearly, the *gllamm* point estimate for $\lambda$ is fairly close to its true value of 0.4, and the true population parameter falls well inside the estimated 95% confidence interval.

### 7 Summary and discussion

The *ssm* wrapper fits a wide range of models that handle two common problems encountered in applied work: ES and SS. Simple two-stage regression strategies to fit these kinds of models are available whenever the outcome is a strictly continuous variate. For dichotomous, ordinal, and count variables, however, things are complicated by a nonlinear model’s being fitted to the data. Then two-stage Heckman-like procedures are only approximate and no valid asymptotic results are available to perform inference. For these reasons either ML estimation or two-stage method of moments is needed. In practice, however, ES and SS issues are commonly ignored whenever the outcome variable is a count or an ordinal response because commercial software does not provide a packaged solution that suits the average user. This article describes a wrapper program, *ssm*, that calls *gllamm* to fit the model by ML. The wrapper accepts data in the usual wide format, has a straightforward syntax, and reports output that is easily interpretable.

*gllamm* can fit many types of models (Rabe-Hesketh, Skrondal, and Pickles [2004b]; Skrondal and Rabe-Hesketh [2004]). The generality of the framework, however, means that users can find it difficult to specify models. *gllamm* wrappers like *ssm* offer a tailored alternative for relatively common but unsupported problems. Using *ssm* limits the user’s control and consequently the ability to introduce modifications. The loss of freedom, however, is compensated for by easier model implementation. To minimize restrictions imposed to the user, the *commands* option has been created. This option causes *ssm* to produce a do-file output that helps the user to prepare the data and specify the model for *gllamm*. The user can modify such a do-file to introduce variations in the fitted model—for instance, to define more parameter constraints.

In the past, the speed of *gllamm* has been a major concern. Since 2003, however, the speed of the program has improved significantly because large sections of *gllamm* were converted to internal Stata code. Though speed may remain a problem when fitting complicated models in *gllamm*, computation time was acceptable for all the examples discussed here.
8 References


ML estimation of endogenous switching and sample selection models


About the authors

Alfonso Miranda is a lecturer in economics in the School of Economic and Management Studies, Keele University, Keele, UK.

Sophia Rabe-Hesketh is 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, London, UK.