The Stata Journal (2018)
18, Number 1, pp. 206–222

Standard-error correction in two-stage optimization models: A quasi–maximum likelihood estimation approach

Fernando Rios-Avila
Levy Economics Institute
Annandale-on-Hudson, NY
friosavi@levy.org

Gustavo Canavire-Bacarreza
School of Economics and Finance
Universidad EAFIT
Medellín, Colombia
gcanavir@eafit.edu.co

Abstract. Following Wooldridge (2014, *Journal of Econometrics* 182: 226–234), we discuss and implement in Stata an efficient maximum-likelihood approach to the estimation of corrected standard errors of two-stage optimization models. Specifically, we compare the robustness and efficiency of the proposed method with routines already implemented in Stata to deal with selection and endogeneity problems. This strategy is an alternative to the use of bootstrap methods and has the advantage that it can be easily applied for the estimation of two-stage optimization models for which already built-in programs are not yet available. It could be of particular use for addressing endogeneity in a nonlinear framework.

Keywords: st0520, maximum likelihood estimation, nonlinear models, endogeneity, two-step models, standard errors

1 Introduction

Selection and endogeneity are common problems in applied econometrics. One common approach to deal with these problems is the use of two-stage estimations, where results from one model, such as the predicted residuals, are used in a second model to obtain coefficient estimates that are unbiased and consistent in the presence of endogeneity or selection. However, to draw appropriate statistical inferences of the models for the estimation of the standard errors (SEs), one has to be careful given the probabilistic nature of the first-stage estimates that are introduced in the second model.

Although the properties and formulation of the correct SEs associated with two-stage optimization models have been available in the literature for decades (Murphy and Topel 1985; Newey and McFadden 1994; White 1982), outside of already packaged commands, practical implementation is not yet standard, particularly for the estimation of nonlinear models (Terza 2016). Despite the efforts of many authors to provide strategies and programming codes to facilitate the implementation of standard-error corrections in Stata (Hardin 2002; Hole 2006; Terza 2016), most applied researchers either implement bootstrap methods at best or ignore the problem by reporting the uncorrected errors at worst (Terza 2016).

In this article, we suggest a different, easier approach to implement the estimation of corrected SEs of two-stage optimization models, based on Wooldridge (2014). The method relies on the maximum likelihood (ML) estimation available in Stata, which is used to estimate a joint function that characterizes the data-generating process of the first-stage and second-stage systems under the assumption of conditional independent distributions. The main use of this strategy is to facilitate the estimation of endogenous nonlinear models using a control function approach such as the ones described in Terza, Basu, and Rathouz (2008) with corrected SEs.

The rest of the article is organized as follows: Section 2 reviews the framework of two-stage optimization models and characterization of the estimation of the correct SEs. Section 3 establishes the general framework of the estimation based on ML using the Stata command `ml` and the basic program structure that defines the objective function. Section 4 provides examples of the estimation compared with existing methods in Stata using Monte Carlo simulations. Section 5 concludes.

2 Two-stage estimations

Systems in which results from one estimation are used in a second model are common in the applied literature. The most prominent examples are the estimation of the two-stage Heckman selection models (Heckman 1979), two-stage least squares (2SLS) for the treatment of endogeneity in linear models, and the control-function approach for the robust estimation of nonlinear models with endogenous variables (Terza, Basu, and Rathouz 2008; Wooldridge 2014). Although these types of models can be fit jointly, two-step procedures are often easier to estimate because they require fewer restrictions on the joint distribution of the data-generating functions (Greene 2012).

While easy to implement, the main drawback of two-stage models has been that the estimation of SEs from the second stage alone are incorrect because they ignore the measurement error that carries over from using the predictions of one model in the next model. Hardin (2002) and Hole (2006) provide guidance for implementing the variance estimator suggested by Murphy and Topel (1985) when including the predictions of a first-stage model into a second model. More recently, Terza (2016) suggests an additional simplification of the estimation of SEs in two-stage models, emphasizing the application of the two-stage residual inclusion approach and the handling of endogeneity in nonlinear models.

The original model described in Hardin (2002) is characterized as follows,

$$E(\mathbf{y}_1|\mathbf{x}_1, \boldsymbol{\theta}_1) \tag{1}$$

$$E\left\{\mathbf{y}_2|\mathbf{x}_2, \boldsymbol{\theta}_2, E(\mathbf{y}_1|\mathbf{x}_1, \boldsymbol{\theta}_1)\right\} \tag{2}$$

where in (1) we model the conditional mean of the endogenous variable $\mathbf{y}_1$ as a function of exogenous variables $\mathbf{x}_1$, and in (2) we model the conditional mean of the variable $\mathbf{y}_2$ as a function of exogenous variables $\mathbf{x}_2$ and some form of the predicted values of the first model.

According to Hardin (2002), two approaches can be used to fit these models. The first approach is a full-information ML, where we start by specifying a joint distribution $f(\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, which is then maximized jointly. A second approach is the estimation of a limited-information ML, which is a two-step procedure. Because the first model depends only on the parameter $\boldsymbol{\theta}_1$, it can be consistently fit. The estimated parameters of the first model can be directly included in the second model, which can be fit using a conditional log-likelihood function:

$$\max_{\theta_2} L_2(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1) = \ln \Sigma \ln \left[f \left\{ \mathbf{y}_2|\mathbf{x}_2, \boldsymbol{\theta}_2, g\left(\mathbf{x}_1, \widehat{\boldsymbol{\theta}}_1\right) \right\} \right]$$

In this framework, Hardin (2002) and Hole (2006) indicate that while the raw variance obtained from the second-stage model is incorrect, it can be easily corrected using the Murphy–Topel approach, which states that

$$\widehat{V}(\boldsymbol{\theta}_2) = \widehat{V}^*(\boldsymbol{\theta}_2) + \widehat{V}^*(\boldsymbol{\theta}_2) \left(\widehat{\mathbf{C}}\widehat{\mathbf{V}}_1\widehat{\mathbf{C}}' + \widehat{\mathbf{R}}\widehat{\mathbf{V}}_1\widehat{\mathbf{C}}' - \widehat{\mathbf{C}}\widehat{\mathbf{V}}_1\widehat{\mathbf{R}}' \right) \widehat{V}^*(\boldsymbol{\theta}_2)$$

where $\widehat{V}^*(\boldsymbol{\theta}_2)$ is the incorrect estimation of the variance obtained from the second-stage maximization problem and $\widehat{\mathbf{V}}_1$ is the correct estimation of the variance of the first-stage estimate. $\widehat{\mathbf{C}}$ and $\widehat{\mathbf{R}}$ are the matrices products defined as

$$\widehat{\mathbf{C}} = E\left\{ \left(\frac{\partial L_2}{\partial \boldsymbol{\theta}_2} \right) \left(\frac{\partial L_2}{\partial \boldsymbol{\theta}_1'} \right) \right\} \quad \text{and} \quad \widehat{\mathbf{R}} = \left\{ \left(\frac{\partial L_2}{\partial \boldsymbol{\theta}_2'} \right) \left(\frac{\partial L_2}{\partial \boldsymbol{\theta}_1} \right) \right\}$$

which Hardin (2002) indicates can be traced back to represent the sandwich estimate of the variance given two ML functions $L_1(\boldsymbol{\theta}_1)$ and $L_2(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1)$ (see Hardin [2002] for details). Terza (2016) proposes a simplified method to address the correction of SEs in two-stage procedures that could be applied for the estimation of endogenous models in the framework of Terza, Basu, and Rathouz (2008) and the two-stage residual imputation (2SRI) approach, using as an example a model where the outcome and endogenous variables are nonlinear functions of the exogenous variables.

3 Quasi–ML approach

According to Wooldridge (2014), the model described above can also be fit using quasi–limited-information ML. As described in the article, for cases in which the models are linear (2SLS), and under the assumption that the errors of the models distribute normal and independent, the joint maximum log-likelihood function can be written as

$$L(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = L_1(\boldsymbol{\theta}_1) + L_2\left(\boldsymbol{\theta}_2|\widehat{\boldsymbol{\theta}}_1\right)$$

In other words, if we assume the distribution function of the parameter $\boldsymbol{\theta}_1$ and conditional distribution of $\boldsymbol{\theta}_2|\widehat{\boldsymbol{\theta}}_1$ are independent from each other, the full system of equations can be estimated using the ML approach. The estimation of this model would not require additional adjustments on the estimation of the SEs, because the measurement error from $\widehat{\boldsymbol{\theta}}_1$ is already accounted for in the model.

As the article argues, the estimation of this type of model can be done using readily available statistical software. Further, inferences can be drawn by reporting the White (1982) type of sandwich variance estimations to account for misspecified likelihood functions. Furthermore, Wooldridge (2014) indicates that the estimation of this joint quasi–maximum likelihood estimation (QMLE) can also be applied to a larger set of ML models that belong to the linear exponential family because they remain consistent even when the density function is partially misspecified (Cameron and Trivedi 2005). This suggests that a range of models such as the models described in Terza, Basu, and Rathouz (2008) and Wooldridge (2015) can also be fit consistently using quasi–ML functions.

3.1 Estimation of QMLE in Stata

The estimation of models using Stata ML functions is simple but requires some programing and knowledge of the appropriate density functions that the researcher assumes determines the data-generating process of the data in hand or some criteria of an objective function that needs to be optimized. In this section, we provide the general setup of how a program can be written to specify the objective function in the framework of a two-stage optimization model. For the specification of this program, we will use the `lf` estimation method within the ML environment in Stata. This is the simplest method for `ml` estimation commands, which require only specifying to an objective function that will be maximized.[1]

As described in Cameron and Trivedi (2010), the `lf` method can be used for the special case where the objective function is an m estimator—namely, where the objective function is the sum or average of some function $q(\cdot)$ over the sample of observations. As the authors indicate, this method can be applied for any function $q(\cdot)$, but robust SEs need to be reported if the objective function is not a likelihood-based function. Say that we are interested in jointly fitting a two-linear model, with normal and independent distributed errors:

$$y_{1i} = \beta_{10} + \beta_{11}x_{1i} + \beta_{12}x_{2i} + \beta_{13}x_{3i} + u_{1i} = X'_{1i}\boldsymbol{\beta}_1 + u_{1i} \quad \text{with } u_{1i} \sim N(0, \sigma_1) \tag{3}$$

$$y_{2i} = \beta_{20} + \beta_{21}x_{1i} + \beta_{23}x_{3i} + \beta_{24}x_{4i} + u_{2i} = X'_{2i}\boldsymbol{\beta}_2 + u_{2i} \quad \text{with } u_{2i} \sim N(0, \sigma_2) \tag{4}$$

If we were to estimate this function using ML, the log-likelihood function of the full system of equations would be written as

$$\begin{aligned} L_i &= L_i^1 + L_i^2 \\ L_i &= \ln\phi(y_{1i}, X'_{1i}\boldsymbol{\beta}_1, \sigma_1) + \ln\phi(y_{2i}, X'_{2i}\boldsymbol{\beta}_2, \sigma_2) \end{aligned} \tag{5}$$

where $\phi(y_{ki}, X'_{ki}\boldsymbol{\beta}_k, \sigma_k)$ is the normal density function, given the conditional mean $X'_{ki}\boldsymbol{\beta}_k$ and standard deviation σ_k.

To be able to fit this model using `ml`, we need to create a program that defines the objective function to be maximized as a function of parameters that need to be

1. Note that `ml` also allows for the specification of the analytical first and second derivatives of the objective function, which provides improvements in the speed of the estimations. This feature, however, is beyond the scope of this article.

estimated. In this simple model, we need to estimate four parameters, the conditional mean of $\mathbf{y}_1(\mathbf{X}'_{1i}\boldsymbol{\beta}_1)$, the conditional mean of $\mathbf{y}_2(\mathbf{X}'_{2i}\boldsymbol{\beta}_2)$, and their respective standard deviations σ_1 and σ_2. This can be done as follows:

```
program myols
args lnf xb1 lns1 xb2 lns2
quietly {
    tempvar lnf1 lnf2
    * Ordinary least squares (OLS) ML component L2
    generate double `lnf2´=ln(normalden($ML_y2,`xb2´,exp(`lns2´)))
    * OLS ML component L1
    generate double `lnf1´=ln(normalden($ML_y1,`xb1´,exp(`lns1´)))
    replace `lnf´=`lnf1´+`lnf2´
}
end
```

Thus, we begin by declaring the internal variable (`args`) that stores the values of the objective functions `lnf`, followed by the parameters that will be used to maximize the objective function. In this example, the four parameters that are estimated are the conditional means (`xb1` and `xb2`) and the standard deviations (`exp(lns1)` and `exp(lns2)`). Note that `xb1` and `xb2` can be linear combinations of explanatory variables. Also, because the standard deviations must be positive, they are not directly estimated, but their natural logs, `lns1` and `lns2`, are estimated instead. Two auxiliary variables are created as temporary variables `lnf1` and `lnf2` to store the log likelihood corresponding to the first and the second equation. They are functions of the parameters that need to be estimated and the dependent variables in the model that are called using the macros `$ML_y1` and `$ML_y2`, which represent the first and second dependent variables in the model. Finally, both likelihood functions are combined and stored in `lnf`, which resembles the expression in (5).

Once the program is stored in memory, the estimation of the model can be done using the following command:

```
ml model lf myols (xb1:y1=x1 x2 x3) (lns1:) (xb2:y2=x1 x3 x4) (lns2:), ///
    maximize vce(robust)
```

The first part, `ml model lf`, indicates that the model will be fit using the `lf` option of ML, which implies that the gradients and Hessians are estimated numerically. Right after, we specify the name of the program we just created, which declares the objective function, in this case the log likelihood associated with the two-equation system. Next, in the same order as they were presented in the program, we declare the information that will be used for the estimation of conditional means and standard deviations. In the order they are written, all variables that are to the left of `=` are considered dependent variables and are internally used in the program as `$ML_y1`, `$ML_y2`, etc. All variables to the right of `=` are used as explanatory variables that enter in the estimation of the parameters as linear combinations. In this example, the first and second dependent variables are `y1` and `y2`; after each variable, we specify the explanatory variables that are used for the conditional means. We assume homoskedastic errors, so no explanatory variables are written for `lns1` or `lns2`. After the comma, we indicate we want to maximize the objective function and ask to report robust SEs.

Imagine now that the models described in (3) and (4) are no longer independent. Let us assume that the outcome y_{1i} is also a function of y_{2i} and that both outcomes depend on some unobserved component v_i so that the system of equations specified before becomes

$$y_{1i} = \beta_{10} + \beta_{11}x_{1i} + \beta_{12}x_{2i} + \beta_{13}x_{3i} + \beta_{14}y_{2i} + u_{1i} + v_i \quad \text{with } u_{1i} \sim N(0, \sigma_1) \tag{6}$$
$$y_{2i} = \beta_{20} + \beta_{21}x_{1i} + \beta_{13}x_{3i} + \beta_{14}x_{4i} + u_{2i} + v_i \quad \text{with } u_{2i} \sim N(0, \sigma_2) \tag{7}$$

This is a common case of endogeneity between y_{1i} and y_{2i} caused by the presence of unobserved heterogeneity v_i. Under this structure, standard estimation methods will no longer provide consistent estimates for (6). One solution to deal with endogeneity is the use of 2SLS. This requires fitting a model where the endogenous covariate y_{2i} is modeled as a function of all exogenous variables in (6) and (7),[2] and the predicted outcome $\widehat{y}_{2i}$ is used in the main equation instead of the actual value y_{2i}. In other words, this implies the estimation of the following models:

$$y_{2i} = \alpha_{20} + \alpha_{11}x_{1i} + \alpha_{21}x_{2i} + \alpha_{13}x_{3i} + \beta_{14}x_{4i} + \epsilon_{2i}$$
$$y_{1i} = \alpha_{10} + \alpha_{11}x_{1i} + \alpha_{12}x_{2i} + \alpha_{13}x_{3i} + g\widehat{y}_{2i} + \epsilon_{1i}$$

Instead of using this two-step procedure manually, the program created above can be adapted to implement this process so that this model can be fit using the `ml` process. This requires specifying an additional coefficient that needs to be estimated. In this example, we call this link `g`. The program and the maximization command then are modified as follows:

```
program myols2
args lnf xb1 g lns1 xb2 lns2
quietly {
    tempvar lnf1 lnf2
    generate double `lnf2´=ln(normalden($ML_y2,`xb2´,exp(`lns2´)))
    generate double `lnf1´=ln(normalden($ML_y1,`xb1´+`g´*`xb2´,exp(`lns1´)))
    replace `lnf´=`lnf1´+`lnf2´
}
end
```

Notice that in this example, the log-likelihood function of the second equation, `lnf2`, is the same as before. The log-likelihood function of the first equation, `lnf1`, is modified so that, in addition to the linear combination of the exogenous variables `xb1`, it includes the conditional mean from the first equation, `xb2`, multiplied by the coefficient `g`. For the estimation of the model, we also need to modify the command so it specifies, in the same order as in the program, that an additional coefficient needs to be estimated.

```
ml model lf myols2 (xb1:y1=x1 x2 x3) (g:) (s1:) (xb2:y2=x1 x2 x3 x4) (s2:), ///
    maximize vce(robust)
```

Notice that the endogenous variable `y2` is not included in the list of explanatory variables in the main model, because we are using the predicted values instead.

2. Note that for the identification of the model, an exogenous covariate or instrumental variable that is correlated with the endogenous variable y_{2i} and uncorrelated with the final outcome y_{1i} is needed.

The above code can be easily extended to allow for a system of multiple equations. For example, if we were to have a model with two endogenous variables, their interaction on the main outcome equation can be accounted for by including an additional link parameter from the first-stage to the second-stage equations:

```
program myols3
args lnf xb1 g2 g3 s1 xb2 s2 xb3 s3
quietly {
    tempvar lnf1 lnf2 lnf3
    generate double `lnf2´=ln(normalden($ML_y2,`xb2´,exp(`lns2´)))
    generate double `lnf3´=ln(normalden($ML_y3,`xb3´,exp(`lns3´)))
    generate double `lnf1´=                                                  ///
        ln(normalden($ML_y1,`xb1´+`g2´*`xb2´+`g3´*`xb3´,exp(`lns1´)))
    replace `lnf´=`lnf1´+`lnf2´+`lnf3´
}
end
ml model lf myols_ols (xb1:y1=x1 x2) (g2:) (g3:) (lns1:) (xb2:y2=x1 x2 x3 x4) ///
    (lns2:) (xb3:y3=x1 x2 x3 x4) (lns3:), maximize vce(robust)
```

In this case, we have two endogenous variables, `y2` and `y3`, that are identified using instruments `x3` and `x4`. The predicted values of their corresponding models, `xb2` and `xb3`, are included in the outcome equation multiplied by the link parameters `g2` and `g3`.

This simple setup can be adapted to implement other strategies that involve two-step estimation modeling.[3] In the next section, we present three examples of how to use this setup to fit a Heckman selection model, a linear model with endogenous covariates, and a nonlinear model (probit) with endogenous covariates, comparing the results with Stata built-in procedures based on Monte Carlo simulations.

4 Applications

4.1 Simulated data structure

In this section, we apply the strategy described above to fit models with selection and endogeneity problems in the framework of linear and nonlinear outcomes and compare the results to already built-in Stata programs. For the applications that follow, we use Monte Carlo simulations to assess the consistency of the estimated SEs of the proposed method. Unless otherwise noted in each subsection, the simulated data are generated using the following data-generating process:

3. An additional advantage of this setup is that after the model is fit, one can obtain marginal effects by using the `margins` command with the option `vce(unconditional)`. We thank an anonymous reviewer who pointed out this feature.

The errors u_1 and u_2 are assumed to be distributed as a bivariate normal distribution, with mean zero, standard deviation 1, and a correlation of 0.7:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \sim B\left(\boldsymbol{\mu}_u = \begin{matrix} 0 \\ 0 \end{matrix}, \boldsymbol{\Sigma}_u = \begin{matrix} 1 & 0.7 \\ 0.7 & 1 \end{matrix}\right)$$

Four exogenous explanatory variables, $[x_1, x_2, x_3, x_4]$, are jointly distributed as a multivariate normal distribution, also with mean zero and a variance–covariance matrix $\boldsymbol{\Sigma}_x$:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \sim \text{MV}\left(\boldsymbol{\mu}_x = \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix}, \boldsymbol{\Sigma}_x = \begin{matrix} 1 & 0.3 & 0.3 & -0.1 \\ 0.3 & 1 & 0.35 & 0.1 \\ 0.3 & 0.35 & 1 & -0.15 \\ -0.1 & 0.1 & -0.15 & 1 \end{matrix}\right)$$

Finally, we consider three exogenous instrumental variables, z_1, z_2, and z_3, that are assumed to be distributed as jointly normal with mean zero and variance–covariance matrix $\boldsymbol{\Sigma}_z$:

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \sim \text{MV}\left(\boldsymbol{\mu}_z = \begin{matrix} 0 \\ 0 \\ 0 \end{matrix}, \boldsymbol{\Sigma}_z = \begin{matrix} 1 & 0.5 & -0.5 \\ 0.5 & 1 & -0.8 \\ -0.5 & -0.8 & 1 \end{matrix}\right)$$

Additional details on the data-generating process and the programs used for the estimation of the models can be found in the companion program to this article.

4.2 Heckman selection model. Two-stage procedure

As indicated before, a common problem in empirical analysis is the selection bias. This problem arises because the outcome of interest is not observed for everyone because of some unobserved nonrandom component. According to the standard Heckman (1979), if the probability of observing the outcome could be modeled as a probit model based on observed characteristics, including some variable or instrument that is strongly related to the probability of selection but not correlated to the outcome, then unbiased estimates of the parameters can be obtained using a two-step procedure, also known as a heckit procedure.

This procedure involves the estimation of a probit model in the first stage, using a set of explanatory variables ($\mathbf{Z}_i$) that may include variables not included in the main model ($\mathbf{X}_i$). The parameters of this first stage are used to estimate a selection term or inverse mills ratio, which are included in the outcome equation. Least-squares estimates from this equation would provide unbiased and consistent estimates of the parameters of interest

$$P(y_{2i} = 1) = \boldsymbol{\Phi}(\mathbf{Z}_i'\boldsymbol{\theta})$$

where Φ is the normal cumulative density function, and $y_{2i} = 1$ if $\mathbf{y}_2^* > 0$, and $= 0$ otherwise, and

$$y_{1i} = \mathbf{X}_i'\boldsymbol{\beta} + \lambda\frac{\phi(\mathbf{Z}_i'\boldsymbol{\theta})}{\Phi(\mathbf{Z}_i'\boldsymbol{\theta})} + e_i$$

where ϕ is the normal density function and y_{1i} is observed only if $y_{2i} = 1$.

This two-step procedure can be implemented in the framework of the QMLE method, where the corresponding log-likelihood function could be written as

$$L_i = L_{\text{selection}} + L_{\text{outcome}}$$

$$L_i = y_{2i}\ln\left\{\Phi(\mathbf{Z}_i'\boldsymbol{\theta})\right\} + (1-y_{2i})\ln\left\{1-\Phi(\mathbf{Z}_i'\boldsymbol{\theta})\right\} + y_{2i}\ln\phi\left(y_{1i}, \mathbf{X}_i'\boldsymbol{\beta} + \lambda\frac{\phi(\mathbf{Z}_i'\boldsymbol{\theta})}{\Phi(\mathbf{Z}_i'\boldsymbol{\theta})}, \sigma_e\right)$$

The program associated with the estimation of this model can then be written as

```
program myheckman
args lnf xb1 g lns1 zg1
quietly {
    * Selection equation
    tempvar lnf1 lnf2
    generate double `lnf1´=ln(normal(`zg1´)) if $ML_y2==1
    replace `lnf1´=ln(1-normal(`zg1´)) if $ML_y2==0
    * Outcome equation
    generate double `lnf2´=                          ///
        ln(normalden($ML_y1,`xb1´+`g´*normalden(`zg1´)/normal(`zg1´),exp(`lns1´)))
    replace `lnf2´=0 if $ML_y2==0
    * Adding both log likelihoods
    replace `lnf´=`lnf1´+`lnf2´
}
end
```

For the simulation exercise, we use the simulated data described above, with the data-generating process for the dependent variables $\mathbf{y}_1$ and $\mathbf{y}_2^*$ defined as follows:

Selection:

$$y_{2i}^* = 1 + x_{1i} - x_{2i} + x_{3i} + u_{2i} \rightarrow y_{2i} = 0 \quad \text{if} \quad y_{2i}^* \leq 0 \quad \text{and} \quad y_{2i} = 1 \quad \text{if} \quad y_{2i}^* > 0$$

Outcome:

$$y_{1i} = 2 + x_{1i} + 0.5 \times x_{2i} - x_{4i} + u_{1i} \quad \text{if} \quad y_{2i}^* > 0$$

In this case, the estimation of the model can be done using the following command:[4]

```
ml model lf myheckman (xb1:y1=x1 x2 x4) (g:) (lns1:) (xb2:y2=x1 x2 x3),  ///
    maximize missing
```

For the simulation, we draw 5,000 samples that are each of size 1,000. In table 1, we provide the summary statistics of the point estimates and the estimated SEs from the Stata built-in command `heckman` and the QMLE Heckman proposed here, as well as the standard deviations of the estimated coefficients from the Monte Carlo simulation exercise.

4. Notice that in the estimation command, we add the option `missing` to allow for missing observations (because of selection) to be kept in the estimation model.

Table 1. Simulation results: Selection problem

	Heckman ML		Heckman two-step		Heckman two-step QMLE	
	Coefficients	Simulated standard deviations	Coefficients	Simulated standard deviations	Coefficients	Simulated standard deviations
Beta `x1`	1.000	0.048	1.000	0.054	1.001	0.054
SE beta `x1`	0.048	0.003	0.054	0.003	0.054	0.004
Beta `x2`	0.501	0.042	0.501	0.044	0.501	0.045
SE beta `x2`	0.042	0.002	0.044	0.002	0.045	0.003
Beta `x4`	−1.000	0.036	−1.000	0.037	−1.000	0.037
SE beta `x4`	0.035	0.002	0.036	0.001	0.036	0.002
Constant	2.000	0.049	2.001	0.060	2.000	0.060
SE constant	0.049	0.005	0.061	0.003	0.061	0.004
Selection term/`g`	0.700	0.083	0.698	0.130	0.702	0.131
SE selection term/`g`	0.082	0.015	0.129	0.007	0.130	0.010

Note: The summary corresponds to 5,000 random draws of size 1,000 as described above. All commands were executed using robust SEs whenever possible. Only the outcome variables results are shown.

As expected from asymptotic theory, the point estimates for all three methods converge to the true parameters. Comparing the average estimated SE for the covariates `x1`, `x2`, and `x4` and the constant with the SEs from the simulation, we can conclude that they are estimated without bias in our proposed method. In regard to the selection term, `g`, we observe that the average estimated SEs converge to the ones obtained from the simulation.

While the method proposed here has a similar performance in terms of its asymptotic properties to the Stata `heckman twostep` estimation, it is worth noting that the `heckman ml` estimation is more efficient, showing smaller SEs of the equivalent selection term compared with the other two procedures[5] as well as for the SEs of the other models.

4.3 Linear regression model with endogeneity

A model that is typically used for addressing endogeneity problems when the outcome and the endogenous variables are continuous is the 2SLS process. This estimation involves obtaining linear predictions of the endogenous variables using all exogenous variables, which then are used instead of the original variables in the outcome equation (two-stage predictor substitution). Alternatively, one can obtain the predicted residuals of the first-stage models and add them to the main outcome model to control for the possible endogeneity. This strategy is also known as a control-function approach or two-

5. The Heckman ML model does not directly estimate a selection term as in the two-step procedure. However, an equivalent parameter is obtained using the product between the estimated correlation between the selection and outcome errors and the standard deviation of the error in the outcome model.

stage residual inclusion (2SRI) approach and is known for performing better when the outcome model is nonlinear (Terza, Basu, and Rathouz 2008; Wooldridge 2014, 2015). As indicated by Wooldridge (2014), using the 2SRI approach in an ML estimation framework is equivalent to using a limited-information ML approach.

Under the assumption that the errors in all the equations are normally and independent distributed, the ML function of the system of equations could be written as follows,

$$L_i = L_{iX} + L_{iY}$$
$$L_i = \ln\phi(X_i^z, \mathbf{Z}_i'\boldsymbol{\gamma}, \sigma_x) + \ln\phi\left\{y_i, \mathbf{X}_i^{x\prime}\boldsymbol{\beta}_x + (\mathbf{Z}_i'\boldsymbol{\gamma})\beta_z, \sigma_y\right\}$$

where $\mathbf{X}_i^x$ stands for the set of all exogenous variables that determine $\mathbf{y}$, X_i^z is an endogenous variable in the model, and $\mathbf{Z}_i$ is the set of all exogenous variables including the instrumental variables, which determine X_i^z.

If we assume a model with two endogenous variables, the program associated with the estimation of this model can then be written as

```
program myivreg2sls
args lnf xb1 g2 g3 lns1 zg1 lns2 zg2 lns3
quietly {
    tempvar lnf1 lnf2 lnf3
    generate double `lnf2´=ln(normalden($ML_y2,`zg1´,exp(`lns2´)))
    generate double `lnf3´=ln(normalden($ML_y3,`zg2´,exp(`lns3´)))
    generate double `lnf1´=                          ///
        ln(normalden($ML_y1,`xb1´+`g2´*`zg1´+`g3´*`zg2´,exp(`lns1´)))
    replace `lnf´=`lnf1´+`lnf2´+`lnf3´
}
end
```

If instead we use a 2SRI approach, the likelihood function and program would be modified as

$$L_i = \ln\phi(X_i^z, \mathbf{Z}_i'\boldsymbol{\gamma}, \sigma_x) + \ln\phi\left\{y_i, \mathbf{X}_i^{x\prime}\boldsymbol{\beta}_x + X_i^z\beta_z + (X_i^z - \mathbf{Z}_i'\boldsymbol{\gamma})\theta, \sigma_y\right\}$$

with the corresponding QMLE program:

```
program myivreg2sri
args lnf xb1 g2 g3 lns1 zg1 lns2 zg2 lns3
quietly {
    tempvar lnf1 lnf2 lnf3
    generate double `lnf2´=ln(normalden($ML_y2,`zg1´,exp(`lns2´)))
    generate double `lnf3´=ln(normalden($ML_y3,`zg2´,exp(`lns3´)))
    generate double `lnf1´=ln(normalden($ML_y1,`xb1´+`g2´*($ML_y2-`zg1´) ///
        +`g3´*($ML_y3-`zg2´),exp(`lns1´)))
    replace `lnf´=`lnf1´+`lnf2´+`lnf3´
}
end
```

In addition, because the outcome and endogenous models can be written assuming an additive error, we can drop the assumption of normality and specify the objective function in the program as if we were fitting the model using ordinary least squares

(OLS) or nonlinear least squares (NLS). This simplifies the model because the standard deviations do not need to be estimated (`lns1`, `lns2`, `lns3`). However, for appropriate statistical inferences, the model needs to be fit using the `vce(robust)` option for the estimation of the SEs:

```
program myivreg2sls2
args lnf xb1 g2 g3 zg1 zg2
quietly {
    tempvar lnf1 lnf2 lnf3
    generate double `lnf2´=-($ML_y2-`zg1´)^2
    generate double `lnf3´=-($ML_y3-`zg2´)^2
    generate double `lnf1´=-($ML_y1-(`xb1´+`g2´*`zg1´+`g3´*`zg2´))^2
    replace `lnf´=`lnf1´+`lnf2´+`lnf3´
}
end
```

For the simulation exercise, we assume a model where the outcome of interest $\mathbf{y}_1$ is a function of two exogenous variables, x_1 and x_2, and two endogenous variables, x_3^* and x_4^*, which are defined as follows:

$$x_{3i}^* = x_{3i} + 0.4 \times z_{i1} + 0.6 \times z_{2i} + u_{2i}$$
$$x_{4i}^* = x_{4i} - 0.5 \times z_{1i} + 0.5 \times z_{3i} + u_{2i}$$

At the same time, the outcome variable $\mathbf{y}_1$ is defined as

$$y_{1i} = 1 + 0.5 \times x_{1i} - 0.5 \times x_{2i} + 0.5 \times x_{3i}^* - 0.5 \times x_{4i}^* + u_{1i}$$

Using the programs presented above, we can fit the model using the following commands:

```
ml model lf myivreg2sls (xb1:y1=x1 x2) (g1:) (g2:) (lns1:)             ///
    (zg1:x3s=x1 x2 z1 z2 z3) (lns3:) (zg2:x4s=x1 x2 z1 z2 z3) (lns4:), ///
    maximize vce(robust)
ml model lf myivreg2sri (xb1:y1=x1 x2 x3s x4s) (g1:) (g2:) (lns3:)     ///
    (zg1:x3s=x1 x2 z1 z2 z3) (lns3:) (zg2:x4s=x1 x2 z1 z2 z3) (lns4:), ///
    maximize vce(robust)
ml model lf myivreg2sls2 (xb1:y1=x1 x2) (g1:) (g2:)                    ///
    (zg1:x3s=x1 x2 z1 z2 z3)  (zg2:x4s=x1 x2 z1 z2 z3), maximize vce(robust)
```

Regarding the estimation of the QMLE and quasi–OLS models, note that for the 2SRI, the endogenous variables are directly introduced in the set of explanatory variables in the outcome equation, which is not the case for the standard 2SLS. The summary of the results after the Monte Carlo simulations is presented in table 2.

Table 2. Linear model with endogenous covariates

	`ivregress 2sls`		`ivregress liml`		`ivregress gmm`	
	Coefficient	Simulated standard deviations	Coefficient	Simulated standard deviations	Coefficient	Simulated standard deviations
Beta `x1`	0.500	0.035	0.501	0.036	0.500	0.035
SE beta `x1`	0.036	0.006	0.037	0.009	0.036	0.006
Beta `x2`	−0.501	0.053	−0.495	0.057	−0.501	0.053
SE beta `x2`	0.052	0.016	0.056	0.030	0.052	0.016
Beta `x3`	0.502	0.096	0.489	0.107	0.502	0.096
SE beta `x3`	0.094	0.037	0.103	0.071	0.094	0.037
Beta `x4`	−0.499	0.097	−0.512	0.108	−0.499	0.097
SE beta `x4`	0.095	0.037	0.104	0.070	0.095	0.037
Constant	0.999	0.032	0.999	0.033	0.999	0.032
SE constant	0.033	0.005	0.034	0.007	0.033	0.005

	QMLE-2SLS		QMLE-2SRI		QNLS-2SLS	
	Coefficient	Simulated SE	Coefficient	Simulated SE	Coefficient	Simulated SE
Beta `x1`	0.500	0.035	0.500	0.035	0.500	0.035
SE beta `x1`	0.036	0.007	0.036	0.007	0.036	0.007
Beta `x2`	−0.500	0.054	−0.500	0.054	−0.500	0.054
SE beta `x2`	0.054	0.022	0.054	0.022	0.054	0.022
Beta `x3`	0.501	0.100	0.502	0.099	0.502	0.099
SE beta `x3`	0.099	0.053	0.098	0.051	0.098	0.051
Beta `x4`	−0.500	0.101	−0.500	0.100	−0.500	0.100
SE beta `x4`	0.099	0.051	0.099	0.049	0.099	0.049
Constant	0.999	0.033	0.999	0.032	0.999	0.032
SE constant	0.033	0.006	0.033	0.006	0.033	0.006

Note: The summary corresponds to 5,000 random draws of size 1,000 as described above. All commands were executed using robust SEs whenever possible. Only the coefficients of the main model are shown. QNLS-2SLS reports the results fitting the model using `ml` where the objective function minimizes the sum of squared errors of all the models.

In the upper section of table 2, we provide the results obtained from using the Stata command `ivregress` with the three methodologies that it allows (`2sls`, `liml`, and `gmm`). As expected, the estimated coefficients and SEs obtained from these strategies converge to the true values. Nevertheless, we observe that the results based on the limited-information ML approach provide somewhat larger SEs compared with the other two alternatives, with estimated coefficients that are the farthest from the true coefficients.

In the lower section, we provide the estimates from the QMLE estimation, using the standard 2SLS approach and the 2SRI approach. We also provide the summary for the estimations based on the 2SLS approach but drop the assumption of normality. Based on the simulation, the results from the three strategies are almost identical, with the coefficient estimates converging to the true values and SEs converging to the standard deviations obtained from the simulation. Compared with what we observe using the built-in commands, we observe only a small efficiency advantage from using either the `ivregress 2sls` or `ivregress gmm` approach.

4.4 Probit model with continuous endogenous covariates

According to Terza, Basu, and Rathouz (2008), a consistent estimation of models with endogeneity in the context of nonlinear outcome functions can be obtained using the two-stage residual approach, also known as the control-function approach (Wooldridge 2015). A special case of nonlinear outcomes is when the dependent variable can be specified using a probit model with continuous and endogenous covariates. The default approach for the estimation of such models in Stata is to use full-information ML. However, a control-function approach can also be used to obtain two-step estimates as described in Newey (1987) and Rivers and Vuong (1988).

According to this process, the first step is to estimate an OLS regression of the endogenous variable against all exogenous variables in the model and use this estimate to obtain the predicted residual. Next, the residuals are included in the main outcome model, which can be fit using a simple probit model. In this sense, the likelihood function corresponding to this model could be written as

$$L_i = L_{\text{OLS}} + L_{\text{Probit}}$$
$$L_i = \ln\phi(X_i^z, \mathbf{Z}_i'\boldsymbol{\gamma}, \sigma_x) + (y_i = 1) \times \ln P(y_i = 1) + (y = 0) \times \ln\{1 - P(y_i = 1)\}$$

where $P(y_i = 1) = \Phi\{\mathbf{X}_i^{x\prime}\boldsymbol{\beta}_t + (X_i^z - \mathbf{Z}_i'\boldsymbol{\gamma})\theta\}$, $\mathbf{X}_i^x$ is the set of exogenous variables, and X_i^z is the endogenous covariate.

Similar to the previous subsection, if we assume the model has two endogenous covariates, the program that defines the log-likelihood function can be written as follows:

```
program myivprobit1
args lnf xb1 g1 g2  zg1 lns2 zg2 lns3
quietly {
    tempvar lnf1 lnf2 lnf3 pr
    * First stage OLS
    generate double `lnf1´=ln(normalden($ML_y2,`zg1´,exp(`lns1´)))
    generate double `lnf2´=ln(normalden($ML_y3,`zg2´,exp(`lns2´)))
    * Estimated residual
    generate double `mresid´=`g1´ *($ML_y2-`zg1´)+`g2´*($ML_y3-`zg2´)
    * Probit model
    generate double `lnf3´=ln(1-normal(`xb1´+`mresid´)) if $ML_y1==0
    replace `lnf3´=ln(normal(`xb1´+`mresid´)) if $ML_y1==1
    replace `lnf´=`lnf1´+`lnf2´+`lnf3´
}
end
```

For the simulation exercise, we will use the data-generating process and definitions described in the previous section with one modification. The outcome variable will be defined as follows:

$$y_i^* = 1 + 0.5 \times x_{1i} - 0.5 \times x_{2i} + 0.5 \times x_{3i}^* - 0.5 \times x_{4i}^* + u_{1i}$$
$$y_i = \begin{cases} = 1 & \text{if} \quad y_i^* > 0 \\ = 0 & \text{if} \quad y_i^* \leq 0 \end{cases}$$

In this case, the model can be fit using the following command:

```
ml model lf myivprobit1 (xb1:y1=x1 x2 x3s x4s) (g1:) (g2:)    ///
    (zg1:x3s=x1 x2 z1 z2 z3) (lns1:) (zg2:x4s=x1 x2 z1 z2 z3) (lns2:),
    maximize vce(robust)
```

In table 3, we provide estimates from the Monte Carlo simulation using 5,000 draws with a sample size of 1,000 observations each, comparing the results from the Stata `ivprobit` built-in command using the two-step and the ML options, and compare them with the proposed method.

Table 3. Probit model with endogenous covariates

	`ivprobit mle`		`ivprobit twostep`	
	Coefficient	Simulated SE	Coefficient	Simulated SE
Beta `x1`	0.497	0.100	0.634	0.079
SE beta `x1`	0.105	0.043	0.081	0.009
Beta `x2`	−0.505	0.160	−0.635	0.111
SE beta `x2`	0.165	0.062	0.113	0.024
Beta `x3`	0.516	0.234	0.639	0.195
SE beta `x3`	0.238	0.088	0.195	0.057
Beta `x4`	−0.479	0.079	−0.629	0.200
SE beta `x4`	0.074	0.057	0.196	0.060
Constant	0.994	0.190	1.266	0.089
SE constant	0.198	0.103	0.089	0.008

	QMLE		QMLE with adjustment	
	Coefficient	Simulated SE	Coefficient	Simulated SE
Beta `x1`	0.636	0.079	0.499	0.099
SE beta `x1`	0.081	0.012	0.104	0.038
Beta `x2`	−0.631	0.115	−0.508	0.158
SE beta `x2`	0.117	0.048	0.163	0.044
Beta `x3`	0.628	0.205	0.521	0.232
SE beta `x3`	0.207	0.109	0.235	0.066
Beta `x4`	−0.642	0.210	−0.478	0.078
SE beta `x4`	0.209	0.104	0.074	0.084
Constant	1.268	0.090	0.998	0.188
SE constant	0.089	0.009	0.196	0.089

Note: The summary corresponds to 5,000 random draws of size 1,000 as described above. All commands were executed using robust SEs whenever possible. Only the coefficients of the main model are shown.

Based on the results from the simulation, we can draw some conclusions. On one hand, the correction obtained for the estimation of the SEs of the coefficients from the `ivprobit` command are almost identical to the ones obtained using our approach. On the other hand, while the `ivprobit twostep` and the simple implementation of our approach provide the same results, they cannot be directly compared with the coefficients

from the ML approach because the former are estimations of the true components only up to a scale. On the last two rows of table 3, we provide additional estimates using an adjustment in the spirit of Newey (1987). After the adjustment, the two-step–QMLE method produces results almost identical to the `ivprobit mle` approach. The main difference among the estimations is that the command `ivregress` produces SEs that seem to be estimated with more precision compared with our method.

5 Summary

In the past decade, a few articles have offered various strategies and codes to correct the estimation of SEs in the framework of two-stage optimization models, including Hardin (2002), Hole (2006), and, more recently, Terza (2016). While these strategies are potentially as simple to be implemented as bootstrap methods, bootstrap methods are usually preferred in the empirical literature because they do not require further manipulation to obtain the corrected estimations.

In this article, we provided an alternative strategy for the estimation of two-step models using a joint quasilikelihood function as described in Wooldridge (2014). While this method also requires some level of programming, the strategy described here provides an intuitive framework for the specification of the underlying likelihood function that needs to be estimated. The baseline programs provided can be easily modified to allow for a large set of two-stage optimization models, which would allow the consistent estimation of linear and nonlinear models in the presence of endogeneity or sample selection as proposed in Terza, Basu, and Rathouz (2008) and Terza (2016).

Using Monte Carlo simulations, we compared the results obtained from the proposed strategy with the results from built-in Stata commands in the presence of selection bias, linear models with endogeneity, and a nonlinear model (probit) with endogeneity, showing that performance on the estimation of point estimates and SEs is comparable.

With the latest release of Stata, a new set of commands, extended regressions, were introduced to address problems regarding endogeneity, sample selection, and endogenous treatments in linear, probit, and censored regression models, adding to the already substantial set of routines that handle these types of problems in empirical analysis. While the methodology proposed here has been used to fit models that are already "solved" in the literature, its usefulness goes beyond the standard models. The use of this estimation method may facilitate the estimation of two-stage nonlinear models in the spirit of Terza, Basu, and Rathouz (2008) and Wooldridge (2014, 2015), for which canned estimation procedures are not yet available.

6 References

Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.

———. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

Hardin, J. W. 2002. The robust variance estimator for two-stage models. *Stata Journal* 2: 253–266.

Heckman, J. J. 1979. Sample selection bias as a specification error. *Econometrica* 47: 153–161.

Hole, A. R. 2006. Calculating Murphy–Topel variance estimates in Stata: A simplified procedure. *Stata Journal* 6: 521–529.

Murphy, K. M., and R. H. Topel. 1985. Estimation and inference in two-step econometric models. *Journal of Business and Economic Statistics* 3: 370–379.

Newey, W. K. 1987. Efficient estimation of limited dependent variable models with endogenous explanatory variables. *Journal of Econometrics* 36: 231–250.

Newey, W. K., and D. McFadden. 1994. Large sample estimation and hypothesis testing. In *Handbook of Econometrics, vol. 4*, ed. R. F. Engle and D. L. McFadden, 2111–2245. Amsterdam: Elsevier.

Rivers, D., and Q. H. Vuong. 1988. Limited information estimators and exogeneity tests for simultaneous probit models. *Journal of Econometrics* 39: 347–366.

Terza, J. V. 2016. Simpler standard errors for two-stage optimization estimators. *Stata Journal* 16: 368–385.

Terza, J. V., A. Basu, and P. J. Rathouz. 2008. Two-stage residual inclusion estimation: Addressing endogeneity in health econometric modeling. *Journal of Health Economics* 27: 531–543.

White, H. 1982. Maximum likelihood estimation of misspecified models. *Econometrica* 50: 1–25.

Wooldridge, J. M. 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234.

———. 2015. Control function methods in applied econometrics. *Journal of Human Resources* 50: 420–445.

About the authors

Fernando Rios-Avila is a research scholar at Levy Economics Institute of Bard College under the Distribution of Income and Wealth program. His research interests include applied econometrics, labor economics, and poverty and inequality.

Gustavo Canavire-Bacarreza is the Director of the Center for Research on Economics and Finance (CIEF) and a professor in the Department of Economics at Universidad EAFIT in Medellín, Colombia. His main interests are applied econometrics, impact evaluation, labor economics, and development.