Calculating Murphy–Topel variance estimates in Stata: A simplified procedure

Arne Risa Hole
National Primary Care Research and Development Centre
Centre for Health Economics
University of York
York, UK
ah522@york.ac.uk

Abstract. Building on the work by Hardin (Stata Journal 2: 253–266), this note shows how the calculation of the Murphy–Topel variance estimator for two-step models can be simplified in Stata by using the scores option of predict. A new benefit is that this new approach simplifies changes to the model specification.

Keywords: st0114, two-step estimation, Murphy–Topel estimator

1 Introduction

In a previous issue of the Stata Journal, Hardin (2002) describes the relationship between the sandwich variance estimator for two-step models and the variance estimator suggested by Murphy and Topel (1985). He also illustrates how both variance estimators can be calculated in Stata. This note shows that the calculation procedure suggested by Hardin can be simplified by using Stata’s scores option of predict. An added benefit is that this new approach simplifies changes to the model specification.

2 The Murphy–Topel estimator

Model systems in which one model is embedded in another appear often in the applied literature. A common case is where the prediction from one model is used as a regressor in a second model,

Model 1: \[ E[y_1 | X_1, \theta_1] \]
Model 2: \[ E[y_2 | X_2, \theta_2] \]

where \( X_1(n \times q) \) and \( X_2(n \times p) \) are data matrices and one of the columns in \( X_2 \) contains the predicted values from model 1. \( \theta_1 \) and \( \theta_2 \) are vectors of parameters that contain the regression coefficients \( \beta_1 \) and \( \beta_2 \), as well as any auxiliary parameters in the models.\(^1\) Since the predicted values from model 1 are included in \( X_2 \), the first parameter vector

---

\(^1\) We restrict our attention to two-step models in which each model has one index function/regression equation.
Calculating Murphy–Topel variance estimates in Stata

θ₁ appears in both models, whereas the second parameter vector θ₂ appears only in the second model. Although θ₁ and θ₂ could be estimated jointly, fitting the models by using a two-step procedure is often easier (see Greene 2003, 508, for a discussion). With this approach, model 1 is fitted first, since it does not involve the second parameter vector. Then model 2 is fitted conditional on the results from the first step. Although this approach leads to a consistent estimate of θ₂, the estimated covariance matrix for model 2 needs to be adjusted to take into account the variability in \( \hat{\theta}_1 \) (since \( \hat{\theta}_1 \) is an estimate of \( \theta_1 \) rather than its true value).

From Hardin (2002) and Greene (2003), the Murphy–Topel estimate of variance for a two-step model is given by

\[
\hat{V}_2 + \hat{V}_2 (\hat{C} \hat{V}_1 \hat{C}' - \hat{R} \hat{V}_1 \hat{C}' - \hat{C} \hat{V}_1 \hat{R}') \hat{V}_2
\]

where \( \hat{V}_1 (q \times q) \) and \( \hat{V}_2 (p \times p) \) are the estimated covariance matrices for model 1 and model 2, respectively, where each is the model-based estimate not taking into account that the estimate of the parameter vector in model 1 is embedded in model 2.

Further,

\[
\hat{C} = (p \times q) \text{ matrix given by } \left\{ \sum_{i=1}^{n} \left( \frac{\partial \ln f_{i2}}{\partial \theta_2} \right) \left( \frac{\partial \ln f_{i2}}{\partial \theta_1'} \right) \right\}
\]

\[
\hat{R} = (p \times q) \text{ matrix given by } \left\{ \sum_{i=1}^{n} \left( \frac{\partial \ln f_{i2}}{\partial \theta_2} \right) \left( \frac{\partial \ln f_{i1}}{\partial \theta_1'} \right) \right\}
\]

where \( f_{i1} \) and \( f_{i2} \) are observation \( i \)'s contribution to the likelihood function of models 1 and 2, respectively. These expressions are conveniently generated using Stata’s scores option of predict. I give examples in the next section.

3 Examples

We begin by replicating the example in Hardin (2002), in which the predicted probabilities from a logit model are used as an explanatory variable in a Poisson model. In this case, neither model has any auxiliary parameters, so \( \theta_1 = \beta_1 \) and \( \theta_2 = \beta_2 \). The first step is to fit the models, saving the scores from both models, the predicted values from the first-stage model, the naïve variance estimates, and the estimated coefficient in the second model for the covariate that was predicted in the initial model:

```
/* First stage: logit, save score as s1 */
. logit z age income ownrent selfemp
. predict double s1, scores
```

2. Only the covariance matrix for model 2 needs to be adjusted; the estimated covariance matrix for model 1 is correct.
For the logit model the derivative of \( \gamma \) in the first- and second-stage models, respectively; the partial derivatives in \( \gamma \) and \( x \) from each model, and a partial derivative of \( \tilde{z}_i \),

\[
\frac{\partial \ln f_{i1}}{\partial \beta_1} = \frac{\partial \ln f_{i1}}{\partial (x_{i1} \beta_1)} \frac{\partial (x_{i1} \beta_1)}{\partial \beta_1} = \frac{\partial \ln f_{i1}}{\partial (x_{i1} \beta_1)} \frac{x_{i1}}{\partial \beta_1}
\]

\[
\frac{\partial \ln f_{i2}}{\partial \beta_2} = \frac{\partial \ln f_{i2}}{\partial (x_{i2} \beta_2)} x_{i2}
\]

\[
\frac{\partial \ln f_{i2}}{\partial \beta_1} = \frac{\partial \ln f_{i2}}{\partial (x_{i1} \beta_1)} x_{i1} = \frac{\partial \ln f_{i2}}{\partial (x_{i2} \beta_2)} \frac{\partial (x_{i2} \beta_2)}{\partial (x_{i1} \beta_1)} x_{i1} \frac{x_{i1}}{\partial \beta_1} = \frac{\partial \ln f_{i2}}{\partial (x_{i2} \beta_2)} \frac{\partial \tilde{z}_i}{\partial x_{i1}} \frac{\partial \tilde{z}_i}{\partial \beta_1} \frac{x_{i1}}{\partial \beta_1} \frac{x_{i1}}{\partial \beta_1}
\]

For the logit model the derivative of \( \tilde{z}_i \) with respect to model 1’s index function equals \( \tilde{z}_i (1 - \tilde{z}_i) \) since \( \tilde{z}_i = \exp(x_{i1} \beta_1) / (1 + \exp(x_{i1} \beta_1)) \). With these results, we can rewrite \( \hat{C} \) and \( \hat{R} \) as follows,

\[
\hat{C} = \sum_{i=1}^{n} x_{i2} \left( s_{i2}^2 \frac{\partial \tilde{z}_i}{\partial (x_{i1} \beta_1)} \right) x_{i1} = X_{i2} \text{Diag} \left( s_{i2}^2 \frac{\partial \tilde{z}_i}{\partial (x_{i1} \beta_1)} \right) \hat{X}_1
\]

\[
\hat{R} = \sum_{i=1}^{n} x_{i2} (s_{i2} s_{i1}) x_{i1} = X_{i2} \text{Diag} \left( s_{i2} s_{i1} \right) \hat{X}_1
\]

where

\[
s_{i1} = \frac{\partial \ln f_{i1}}{\partial (x_{i1} \beta_1)} \quad \text{and} \quad s_{i2} = \frac{\partial \ln f_{i2}}{\partial (x_{i2} \beta_2)}
\]

This structure is common for all two-step models in which each model has one index function and no auxiliary parameters (when the models have auxiliary parameters, the
Calculating Murphy–Topel variance estimates in Stata

calculations are somewhat more complicated as we will see below). With this informa-
tion it is straightforward to compute the $\hat{C}$ and $\hat{R}$ matrices by using Stata’s matrix
accum command as suggested by Hardin:

```
// Calculate C using scores
.matrix accum C = age income ownrent selfemp const age income avgexp zhat
> const [iw=s2*s2*zhat*(1-zhat)*zz], nocons
// Calculate R using scores
.matrix accum R = age income ownrent selfemp const age income avgexp zhat
> const [iw=s2*s1], nocons
// Get only the desired partition
.matrix C = C[6..10,1..5]
.matrix R = R[6..10,1..5]
.matrix M = V2 + (V2 * (C*V1*C' - R*V1*C' - C*V1*R') * V2)
capture program drop doit
.matrix b = e(b)
.program define doit, eclass
    ereturn post b M
    ereturn local vcetype "Mtopel"
    ereturn display
.end
.doit
```

For comparison, the Poisson output with unadjusted standard errors is given below:

```
Poisson regression
Number of obs = 100
LR chi2(4) = 27.21
Prob > chi2 = 0.0000
Log likelihood = -78.330992
Pseudo R2 = 0.1480
```

```
                  Mtopel
                      Coef.  Std. Err.      z    P>|z|     [95% Conf. Interval]
                      y
          age       0.0731059   .1096293    0.67   0.505    -.1417636   .2879755
         income    .0452336   .4375397    0.10   0.918   -.8123285   .9027957
         avgexp    -.0068969   .0042657   -1.62   0.106   -.0152561   .0014623
           zhat     4.632355    10.82669    0.43   0.669  -16.58757   25.85228
            _cons  -6.319947   9.661564   -0.65   0.513  -25.25626   12.61637

```

The manual calculation (that which must be derived and then specified by the user) for
calculation of the estimates involves the evaluation of $\partial \hat{z}_i / \partial (x_{i1} \hat{\beta}_1)$; the scores option of predict makes the remaining calculation in Hardin’s procedure redundant.
The main advantage of deriving the Murphy–Topel variance estimate in this way is that modifying the code to use a different model in one of the two steps is easy. If, for example, one were interested in using a probit model in the second step instead of a Poisson model, one need only replace `poisson` with `probit` in the code above. This change produces the following results:

| Mtopel | Coef.  | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|--------|--------|-----------|-------|------|---------------------|
| age    | 0.040167 | 0.0375665 | 1.07  | 0.285 | -0.0334619 - 0.1137959 |
| income | 0.1221488 | 0.1441061 | 0.85  | 0.397 | -0.1602941 - 0.4045916 |
| avgexp | -0.0023966 | 0.0010854 | -2.16 | 0.031 | -0.0044739 - 0.0002192 |
| zhat   | 2.15221 | 2.385346 | 0.90  | 0.367 | -2.522371 - 6.828014 |
| _cons  | -3.8865 | 2.604024 | -1.49 | 0.136 | -8.990293 - 1.217293 |

Changing the first-stage model to a probit instead of a logit takes a little more work since we have to take into account that the derivative of \( \hat{z}_i \) with respect to model 1's index function now equals \( \phi(x_i \hat{\beta}_1) \) since \( \hat{z}_i = \Phi(x_i \hat{\beta}_1) \). In addition to changing `logit` to `probit` in the code and saving the linear prediction from the probit model as variable `xb`, we must change the line calculating \( \hat{C} \) as follows,

```
// Calculate C using scores
. matrix accum C = age income ownrent selfemp const age income avgexp zhat
> const [iw=s2*s2*normalden(xb)*zz], nocons
```

which produces the following results:

| Mtopel | Coef.  | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|--------|--------|-----------|-------|------|---------------------|
| y      | 0.0803012 | 0.1509582 | 0.53  | 0.595 | -0.2155714 - 0.3761738 |
| age    | 0.0397158 | 0.0221716 | 0.08  | 0.939 | -0.9837218 - 1.063153 |
| income | -0.0068861 | 0.0047102 | -1.46 | 0.144 | -0.0161178 - 0.0023457 |
| avgexp | 5.393431 | 14.91054 | 0.36  | 0.718 | -23.83068 - 34.61765 |
| _cons  | -7.094363 | 13.68211 | -0.52 | 0.604 | -33.91080 - 19.72207 |

To use a linear regression model in the first step takes a little more work since we have to modify the results from the `regress` command to get the maximum likelihood estimates of the covariance matrix and the mean squared error of the regression:

```
. reg z age income ownrent selfemp // First stage: regression
. predict double zhat // Generated variable for second stage
. predict double res, res // Get residuals + squared residuals
. gen double res2 = res^2
. quietly sum res2
. scalar mse = r(mean) // ML estimate of sigma^2
. matrix V1 = (e(df_r)/e(N))*e(V) // ML estimate of covariance matrix
. gen double s1 = res*(1/mse) // Generate score
```
Further, we must amend the code to take into account that the derivative of $\hat{z}_i$ with respect to model 1’s index function now equals 1 since $\hat{z}_i = x_{1i} \hat{\beta}_1$.

```stata
// Calculate C using scores
.matrix accum C = age income ownrent selfemp const age income avgexp zhat
> const [iw=s2*s2*zz], nocons
```

which produces the following results:

```
.doit

Mtopel
Coef. Std. Err. z P>|z| [95% Conf. Interval]

y
age .1097948 .4069624 0.27 0.787 -.6878369 .9074264
income -.0550747 1.280603 -0.04 0.966 -2.565009 2.454860
avgexp -.0068635 .0061429 -1.12 0.264 -.0189034 .0051765
zhat 7.46005 34.49451 0.22 0.829 -60.14795 75.06805
_cons -9.27511 33.76454 -0.27 0.784 -75.45239 56.90217
```

If we want to use a negative binomial model instead of a Poisson model in the second stage, we have to take into account the auxiliary (dispersion) parameter in the negative binomial model when deriving the Murphy–Topel variance estimate. Now $\theta_2$ has two segments: the regression coefficients $\beta_2$ and the auxiliary parameter $\alpha$.\(^3\) Here we have the following,

$$
\hat{C} = \sum_{i=1}^n x_{1i}^' \left\{ s_{i2}^2 \frac{\partial \hat{z}_i}{\partial (x_{1i} \hat{\beta}_1)} \right\} x_{1i} = \tilde{X}_2^' \text{Diag} \left\{ s_{i2}^2 \frac{\partial \hat{z}_i}{\partial (x_{1i} \hat{\beta}_1)} \right\} X_1
$$

$$
\hat{R} = \sum_{i=1}^n x_{1i}^' s_{i2} s_{i1} x_{1i} = \tilde{X}_2^' \text{Diag} \{ s_{i2} s_{i1} \} X_1
$$

where

$$
\tilde{X}_2 = \left( X_2, \frac{a_i}{s_{i2}} \right) \quad \text{and} \quad a_i = \frac{\partial \ln f_i}{\partial \alpha}
$$

The only correction necessary to allow for the presence of the auxiliary parameter is to replace $X_2$ with $\tilde{X}_2$ in the previous equations. $\tilde{X}_2$ is simply $X_2$ with an additional column appended that contains the derivative of the log-likelihood function with respect to the auxiliary parameter divided by the derivative of the log-likelihood function with respect to the index function.\(^4\) Both these derivatives/scores can be calculated in Stata by specifying the `scores` option with `predict` after running `nbreg`. The code and results are given below:

---

\(^3\) $\alpha$ is used here generally to denote an auxiliary parameter; it should not be confused with the $\alpha$ in the description of the `nbreg` command in the Stata manual (our $\alpha$ actually equals $\ln \alpha$ in the manual).

\(^4\) We divide by $s_{i2}$ to undo the weighting by $s_{i2}$ in the square brackets.
. logit z age income ownrent selfemp
. predict double s1, scores
. matrix V1 = e(V)
. predict double zhat
. nbreg y age income avgexp zhat
. predict double s2 a, scores
. matrix V2 = e(V)
. scalar zz = _b[zhata]/s2 // Divide a by s2 to undo weighting below
. gen a_s = a / s2
. matrix accum C = age income ownrent selfemp const age income avgexp zhat
> const a_s [iw=s2*s2*zhat*(1-zhat)*zz], nocons
. matrix accum R = age income ownrent selfemp const age income avgexp zhat
> const a_s [iw=s2*s1], nocons
. matrix C = C[6..11,1..5]
. matrix R = R[6..11,1..5]
. matrix M = V2 + (V2 * (C*V1*C\' - R*V1*C\' - C*V1*R\') * V2)
. matrix b = e(b)
. doit

|        | Mtopel Coef. | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|--------|--------------|-----------|-------|-----|---------------------|
| y      |              |           |       |     |                     |
| age    | .107667      | .1097165  | 0.98  | 0.326 | -.1073833 .3226973 |
| income | .0209116     | .3621894  | 0.06  | 0.954 | -.6889665 .7307897 |
| avgexp | -.005743     | .0023503  | -2.44 | 0.015 | -.0103495 -.0011365 |
| zhat   | 6.469631     | 7.848509  | 0.82  | 0.410 | -8.913164 21.85243 |
| _cons  | -8.807249    | 8.353285  | -1.05 | 0.292 | -25.17939 7.564889 |
| lalpha |              |           |       |     |                     |
| _cons  | 1.15111      | .5468807  | 2.10  | 0.035 | .0792434 2.222976  |

What if there is more than one auxiliary parameter in the model? This situation can easily be accommodated using the above setup. Say that we wanted instead to use an ordered probit in the second stage and a probit in the first stage. Collapsing the dependent variable to three categories—0, 1, and 2, where 2 is “2 or higher”—produces a model with two auxiliary parameters or cutoff points. Here

\[ \tilde{\mathbf{X}_2} = \left( \mathbf{X}_2, \frac{\alpha_1}{s_{21}}, \frac{\alpha_2}{s_{22}} \right) \]

where \[ a_1 = \frac{\partial \ln f_{12}}{\partial \alpha_1} \] and \[ a_2 = \frac{\partial \ln f_{22}}{\partial \alpha_2} \]

where \( \alpha_1 \) and \( \alpha_2 \) are the two auxiliary parameters in the model. The following code estimates the models and produces the Murphy–Topel variance estimate:

. probit z age income ownrent selfemp
. predict double s1, scores
. matrix V1 = e(V)
Calculating Murphy–Topel variance estimates in Stata

```
. predict double zhat
. predict xb, xb // Generate linear prediction
. gen y_ordered = y // Generate depvar for ordered probit
. recode y_ordered (3=2) (4=2) (7=2)
. oprobit y_ordered age income avgexp zhat
. predict double s2 a1 a2, scores
. matrix V2 = e(V)
. scalar zz = _b[zhat]
. gen a1_s = a1 / s2 // Divide by s2 to undo weighting below
. gen a2_s = a2 / s2
// Calculate C using scores
. matrix accum C = age income ownrent selfemp const age income avgexp zhat > a1_s a2_s [iw=s2*s2*normalden(xb)*zz], nocons
// Calculate R using scores
. matrix accum R = age income ownrent selfemp const age income avgexp zhat > a1_s a2_s [iw=s2*s1], nocons
// Get only the desired partition
. matrix C = C[6..11,1..5]
. matrix R = R[6..11,1..5]
. matrix M = V2 + (V2 * (C*V1*C' - R*V1*C' - C*V1*R') * V2)
. matrix b = e(b)
. doit
```

| Coef.  | Std. Err. | z      | P>|z|   | [95% Conf. Interval] |
|--------|-----------|--------|-------|----------------------|
| y_ordered |          |        |       |                      |
| age    | .0415961  | .0383581| 1.08  | 0.278                |
| income | .1451392  | .1519067| 0.96  | 0.339                |
| avgexp | -.0028311 | .0011394| -2.48 | 0.013                |
| zhat   | 2.551639  | 2.640499| .97   | 0.334                |

| cut1   |          |        |       |                      |
| _cons  | 4.237672 | 2.859636| 1.48  | 0.138                |

| cut2   |          |        |       |                      |
| _cons  | 4.799178 | 2.871063| 1.67  | 0.095                |

4 Conclusion

This note demonstrates how the Murphy–Topel variance estimator for two-step models can be calculated in Stata by using the `scores` option of `predict`. This approach reduces the amount of calculation needed to obtain the variance estimate and makes changing from one model specification to another straightforward.
5 Acknowledgments

I thank the anonymous referee and the production editor whose comments and suggestions have greatly improved the paper. NPCRDC receives funding from the Department of Health. The views expressed here are not necessarily those of the funders.

6 References


About the author

Arne Risa Hole is a research fellow at the Centre for Health Economics, York, UK.