The Stata Journal (2016)
16, Number 1, pp. 37–51

Estimation of multivariate probit models via bivariate probit

John Mullahy
University of Wisconsin–Madison
National University of Ireland Galway
and National Bureau of Economic Research
Madison, WI
jmullahy@wisc.edu

Abstract. In this article, I suggest the utility of fitting multivariate probit models using a chain of bivariate probit estimators. This approach is based on Stata's `biprobit` and `suest` commands and is driven by a Mata function, `bvpmvp()`. I discuss two potential advantages of the approach over the `mvprobit` command (Cappellari and Jenkins, 2003, *Stata Journal* 3: 278–294): significant reductions in computation time and essentially unlimited dimensionality of the outcome set. Computation time is reduced because the approach does not rely on simulation methods; unlimited dimensionality arises because only pairs of outcomes are considered at each estimation stage. This approach provides a consistent estimator of all the multivariate probit model's parameters under the same assumptions required for consistent estimation via `mvprobit`, and simulation exercises I provide suggest no loss of estimator precision relative to `mvprobit`.

Keywords: st0423, bvpmvp(), bvopmvop(), multivariate probit models, bivariate probit

1 Introduction

In this article, I suggest the utility of fitting multivariate probit (MVP) models using a chain of bivariate probit estimators. I demonstrate how this approach, based on Stata's `biprobit` and `suest` commands and driven by the Mata function `bvpmvp()`, affords two potential advantages over the `mvprobit` command, that is, significant reductions in computation time and essentially unlimited dimensionality of the outcome set (`mvprobit`'s limit is $M = 20$ outcomes).[1] Computation time is reduced because, unlike `mvprobit`, `bvpmvp()` does not rely on simulation methods; unlimited dimensionality arises because only pairs of outcomes are considered at each estimation stage. Importantly, this `bvpmvp()` approach provides a consistent estimator of all the MVP model's parameters under the same assumptions required for consistent estimation via `mvprobit`, and the simulation exercises herein suggest no loss of estimator precision relative to `mvprobit`.

1. Stata/SE's restriction that `matsize` cannot exceed 11,000 ultimately places a limit on the size of the parameter vector that can be estimated. All references to Stata herein are to Stata/SE 13.1.

This approach was inspired by the goal of embedding MVP estimation in a large-replication bootstrap exercise. The simulation results that I present in section 5 suggest that the computation time saved by the `bvpmvp()` method relative to `mvprobit` can be significant, while numerical differences in the respective point estimates and estimated standard errors are trivial. Because the potential applicability of MVP models is broad, it is important in practice that such potential not be thwarted by computational challenges.

The remainder of the article is organized as follows. In section 2, I describe the MVP model and, in section 3, the `bvpmvp()` method. In section 4, I present the comparison empirical exercises and, in section 5, the comparative results. In section 6, I consider parallel issues involved in the estimation of multivariate ordered probit (MVOP) models, and in section 7, I finish with a summary.

2 The MVP model

The MVP model is typically specified as

$$y^*_{ij} = \mathbf{x}_i\boldsymbol{\beta}_j + u_{ij} \tag{1}$$

$$y_{ij} = 1(y^*_{ij} > 0) \tag{2}$$

$$\mathbf{u}_i = (u_{i1}, \ldots, u_{iM}) \sim \text{MVN}(\mathbf{O}, \mathbf{R}) \quad \text{or} \quad \mathbf{y}^*_i = (y_{i1}, \ldots, y_{iM}) \sim \text{MVN}(\mathbf{x}_i\mathbf{B}, \mathbf{R}) \tag{3}$$

where $i = 1, \ldots, N$ indexes observations, $j = 1, \ldots, M$ indexes outcomes, $\mathbf{x}_i$ is a K-vector of exogenous covariates, the $\mathbf{u}_i$ are assumed to be independent identically distributed across i but correlated across j for any i, and MVN denotes the multivariate normal distribution. (Henceforth, the i subscripts will be suppressed.) The standard normalization sets the diagonal elements of $\mathbf{R}$ equal to 1 so that $\mathbf{R}$ is a correlation matrix with off-diagonal elements $\rho_{pq}, \{p, q\} \in \{1, \ldots, M\}, p \neq q$.[2] With standard full-rank conditions on the $\mathbf{x}$'s and each $|\rho_{pq}| < 1$, $\mathbf{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_M)$ and $\mathbf{R}$ will be identified and estimable with sufficient sample variation in the $\mathbf{x}$'s.

3 Estimation and inference

Estimation of the M-outcome MVP model using `mvprobit` requires simulation of the MVN probabilities (Cappellari and Jenkins 2003), with `mvprobit` computation time increasing in M, K, N, and D (simulation draws).[3] However, all the parameters $(\mathbf{B}, \mathbf{R})$ can be estimated consistently using bivariate probit—implemented as Stata's `biprobit`

2. This normalization rules out cases like heteroskedastic errors (Wooldridge 2010, sec. 15.7.4). While this normalization is common—for instance, normalizing each univariate marginal to be a standard probit—it is not the only possible normalization of the covariance matrix.
3. Specifically, in the empirical exercises reported below as well as in some other simulations not reported here, `mvprobit` computation time increases—trivially in K, essentially proportionately in D, slightly more than proportionately in N, and at a rate between 2^M and 3^M in M. Greene and Hensher (2010) suggest that MVP computation time would increase with 2^M, but the results obtained in the simulations here suggest a somewhat greater rate of increase.

command—while consistent inferences about all of these parameters are afforded via Stata's `suest` command. Because the proposed approach proves significantly faster in terms of computation time with no obvious disadvantages, this strategy may merit consideration in applied work.

The key result for the proposed estimation strategy is that the multivariate normal distribution is fully characterized by the mean vector $\mathbf{xB}$ and correlation matrix $\mathbf{R}$. For present purposes, the key feature of the multivariate (conditional) normal distribution $F(y_1^*, \ldots, Y_M^*|\mathbf{x})$ is that all of its bivariate marginals—$F(y_j^*, y_m^*|\mathbf{x})$—are bivariate normal with mean vectors and correlation matrices corresponding to the respective submatrices of $\mathbf{xB}$ and $\mathbf{R}$ (Rao 1973, 8a.2.10).

Under the normalization that the diagonal elements of $\mathbf{R}$ are all one, the $\mathbf{B}$ parameters are identified using all M (conditional) univariate marginals $F(y_j^*|\mathbf{x})$; there is no need to appeal to the multivariate features of $F(y_1^*, \ldots, y_M^*|\mathbf{x})$ to identify $\mathbf{B}$. The $0.5M(M-1)$ bivariate marginals provide the additional information about the ρ_{pq} parameters. As such, identifying the parameters of all the bivariate marginals implies identification[4] of the parameters of the full multivariate joint distribution so that consistent estimation of all the bivariate marginal probit models $\Pr(y_p = t_p, y_q = t_q|\mathbf{x})$ provides consistent estimates of all the parameters $(\mathbf{B}, \mathbf{R})$ of the full MVP model for $\Pr(y_1 = t_1, \ldots, y_M = t_M|\mathbf{x})$ for $t_j \in \{0, 1\}, j = 1, \ldots, M$.

3.1 Estimation via bivariate probit

The proposed approach, which can be implemented using the Mata function `bvpmvp()`, is as follows. First, corresponding to each possible outcome pair, $0.5M(M-1)$ bivariate probit models are fit using `biprobit`, yielding one estimate[5] of each ρ_{pq} and $M-1$ estimate of $\boldsymbol{\beta}_j$, where $j = 1, \ldots, M$. Each $M-1$ estimate of $\boldsymbol{\beta}_j$ is consistent because each `biprobit` specification uses the same normalization on the relevant submatrices of $\mathbf{R}$. Each of these estimates $(\widehat{\boldsymbol{\beta}}_p, \widehat{\boldsymbol{\beta}}_q, \widehat{\rho}_{pq})_b$, where $b = 1, \ldots, 0.5M(M-1)$, is stored and then combined using Stata's `suest` command, which provides a consistent estimate of the joint variance–covariance matrix of all $M(M-1)(0.5+K)$ parameters estimated with the $0.5M(M-1)$ `biprobit` estimates. We denote this vector of parameter estimates and its estimated variance–covariance matrix as $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\Omega}}$, respectively.[6]

Second, we compute the simple averages $\widehat{\boldsymbol{\beta}}_{jA} = \{1/(M-1)\} \sum_{\substack{m=1 \\ m \neq j}}^{M} \widehat{\boldsymbol{\beta}}_{jm}$. This gives a $k \times M$ matrix of estimated averaged coefficients, denoted $\widehat{\mathbf{B}}_A = (\widehat{\boldsymbol{\beta}}_{1A}, \ldots, \widehat{\boldsymbol{\beta}}_{MA})$. Because a weighted average of consistent estimators is generally a consistent estimator, the resulting $\widehat{\mathbf{B}}_A$ will be consistent for $\mathbf{B}$. This averaging occurs because the $\mathbf{B}$ parameters in the proposed approach are overidentified; that is, there are $M-1$ consistent estimates of each $\boldsymbol{\beta}_j$, $j = 1, \ldots, M$. Some other rule could be used to compute one consistent estimate of each $\boldsymbol{\beta}_j$ from among the $M-1$ candidates, but unless alternative

4. As discussed below, identification of all the bivariate marginals implies overidentification of $\mathbf{B}$.
5. `biprobit` directly estimates the inverse hyperbolic tangent of ρ_{pq} or $0.5\ln\{(1+\rho_{pq})/(1-\rho_{pq})\}$.
6. $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\Omega}}$ are the `suest`-stored matrix results `e(b)` (a row vector) and `e(V)`, respectively.

strategies could boast significant precision gains, computational simplicity recommends the simple average as an obvious solution. See the appendix for further discussion.

Finally, we let $\mathbf{Q}$ denote the $0.5M(M-1)$ vector of the $\tanh^{-1}(\rho_{jk})$ estimated in each `biprobit` specification, and we define the $M\{0.5(M-1)+K\} \times 1$ vector $\widehat{\boldsymbol{\Theta}} = [\mathrm{vec}(\widehat{\mathbf{B}}_A)^T, \widehat{\mathbf{Q}}^T]^T$. We define $\mathbf{H}$ as the $M\{0.5(M-1)+K\} \times M(M-1)(0.5+K)$ averaging and selection matrix that maps $\widehat{\boldsymbol{\alpha}}$ to $\widehat{\boldsymbol{\Theta}}$; that is, $\widehat{\boldsymbol{\Theta}} = \mathbf{H}\widehat{\boldsymbol{\alpha}}^T$; the elements of $\mathbf{H}$ are $1/(M-1)$, 1, or 0.[7] The estimated variance–covariance matrix of $\widehat{\boldsymbol{\Theta}}$, useful for inference, is given by $\widehat{\mathrm{var}}(\widehat{\boldsymbol{\Theta}}) = \mathbf{H}\widehat{\boldsymbol{\Omega}}\mathbf{H}^T$.

3.2 bvpmvp(): A Mata function to implement the proposed estimation approach

The function `bvpmvp()` returns the $M\{k+0.5(M-1)\} \times [M\{k+0.5(M-1)\}+1]$ matrix, whose first column is $\widehat{\boldsymbol{\Theta}}^T$ and whose remaining elements are the elements of the $M\{k+0.5(M-1)\}$ dimension-symmetric square matrix $\widehat{\mathrm{var}}(\widehat{\boldsymbol{\Theta}})$. `bvpmvp()` takes six arguments: 1) a string containing the names of the M outcomes; 2) a string containing the names of the $K-1$ nonconstant covariates; 3) a (possibly null) string containing any "if" conditions for estimation; 4) a scalar indicating whether to display the interim estimation results; 5) a scalar indicating the rounding level of presented results; and 6) a scalar indicating whether to display the final results. For example,

```
bv1 = bvpmvp("y1 y2 y3 y4","x1 x2 x3 x4","if _n<=10000",0,.001,1)
bv2 = bvpmvp(yn,xn,ic,0,.001,1)
```

`bvpmvp()`'s summary report displays the $\widehat{\mathbf{B}}_A$ estimates, their estimated standard errors, and the estimated correlation matrix $\widehat{\mathbf{R}}$; an example is provided in exhibit 1. Of course, suppressing these results may be useful, for instance, in simulation or bootstrapping exercises. The do-file containing the Mata code for `bvpmvp()` is available with this article's supplementary materials.

7. A general form of the $\mathbf{H}$ matrix is complicated to express concisely. For example, for $M = 3$ and $K = 2$, the 9×15 $\mathbf{H}$ matrix, computed internally by `bvpmvp()`, is

$$\mathbf{H} = \begin{bmatrix}
0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

Exhibit 1: Sample output from `bvpmvp()` ($N = 10000$, $M = 4$, $K = 5$)

```
. mata
------------------------------------- mata (type end to exit) -------------------------------------
: yn="y1 y2 y3 y4"
: xn="x1 x2 x3 x4"
: ic="if _n<=10000"
: bv1=bvpmvp(yn,xn,ic,1,.001,1)

  ********************************************
  *                                          *
  *       Multivariate Probit: Results       *
  *                                          *
  ********************************************

  N. of Observations (from suest): 10000

  Estimation Sample: if _n<=10000


  Averaged Beta-Hat Point Estimates and Estimated Standard Errors

             1        2        3        4        5
  ┌──────────────────────────────────────────────────┐
 1│                 y1       y2       y3       y4
 2│
 3│       x1      .328    -.449     .315     .457
 4│             (.045)   (.046)   (.045)   (.046)
 5│
 6│       x2    -.331     .562     .388    -.441
 7│             (.045)   (.046)   (.045)   (.046)
 8│
 9│       x3      .32    -.398    -.321    -.452
10│             (.045)   (.046)   (.045)   (.046)
11│
12│       x4    -.392     .396     -.35     -.35
13│             (.045)   (.046)   (.045)   (.045)
14│
15│    _cons      .391    -.508     .321    -.452
16│             (.046)   (.047)   (.046)   (.047)
17│
  └──────────────────────────────────────────────────┘
```

```
Estimated Correlation (Rho) Matrix and Estimated Standard Errors

           1        2        3        4        5
 ┌──────────────────────────────────────────────────
 1 │                y1       y2       y3       y4
 2 │
 3 │       y1        1     .331     .507     .287
 4 │                      (.016)   (.013)   (.016)
 5 │
 6 │       y2     .331        1     .342     .203
 7 │            (.016)             (.016)   (.017)
 8 │
 9 │       y3     .507     .342        1     .309
10 │            (.013)   (.016)             (.016)
11 │
12 │       y4     .287     .203     .309        1
13 │            (.016)   (.017)   (.016)
14 │
 └──────────────────────────────────────────────────


Cut & Paste Matrix, Averaged Beta-Hat Point Estimates

(.328 , -.449 , .315 , .457) \
(-.331 , .562 , .388 , -.441) \
(.32 , -.398 , -.321 , -.452) \
(-.392 , .396 , -.35 , .45) \
(.391 , -.508 , .321 , -.452)


Cut & Paste Matrix, Estimated Correlation Matrix

(1 , .331 , .507 , .287) \
(.331 , 1 , .342 , .203) \
(.507 , .342 , 1 , .309) \
(.287 , .203 , .309 , 1)


: end
```

4 Simulation exercises

Here I present a simulation exercise to assess the relative performance of the proposed approach and the approach based on `mvprobit`. Three sample sizes ($N = 2000$, $N = 10000$, $N = 50000$) are considered. The data structure corresponding to (1)–(2) has either $K = 5$ or $K = 9$ covariates $\mathbf{x}$ (four or eight independently distributed uniform variates plus a constant) and $M = 8$ binary outcomes y_{ij} (only four of which are used in some specifications) corresponding to latent y^*_{ij} having cross-outcome correlations ρ_{ik} variously in $(0.2, 1/\sqrt{10}, 0.5)$ for all $j \neq k$; specifically, we have the following:

$$\mathbf{R} = \begin{bmatrix} 1 & & & & & & & \\ 10^{-0.5} & 1 & & & & & & \\ 0.5 & 10^{-0.5} & 1 & & & \text{(symm.)} & & \\ 10^{-0.5} & 0.2 & 10^{-0.5} & 1 & & & & \\ 0.5 & 10^{-0.5} & 0.5 & 10^{-0.5} & 1 & & & \\ 10^{-0.5} & 0.2 & 10^{-0.5} & 0.2 & 10^{-0.5} & 1 & & \\ 0.5 & 10^{-0.5} & 0.5 & 10^{-0.5} & 0.5 & 10^{-0.5} & 1 & \\ 10^{-0.5} & 0.2 & 10^{-0.5} & 0.2 & 10^{-0.5} & 0.2 & 10^{-0.5} & 1 \end{bmatrix}$$

For `mvprobit`, the `draws()` option was set both at 10 and 20. The simulations are performed using Stata/SE 13.1 on an iMac 3.4GHz Intel Core i7 processor and OS X v10.8.[8] The do-files containing the code used to generate the data and perform the simulations are available on request.

5 Simulation results

Key results of the simulations are summarized in tables 1–3. Table 1 displays the absolute and relative computation times for `mvprobit` and `bvpmvp()` estimation across the various combinations of the N, M, K, and D parameters. Enormous differences in computation time are seen between the two estimation methods across all the different parameter combinations (for reference, it may be useful to recall that there are 86,400 seconds in one day). Tables 2 and 3 present a side-by-side comparison of the point estimates of $\mathbf{B}$ and $\mathbf{R}$ obtained in one select specification ($N = 10000$, $M = 4$, $K = 5$). For both $\mathbf{B}$ and $\mathbf{R}$, the differences between the `mvprobit` and `bvpmvp()` point estimates and corresponding estimated standard errors are trivial.

8. The simulations set Stata's `matsize` parameter at 600 for all specifications. In some preliminary investigation, I observed that computation time for `bvpmvp()` increased significantly when `matsize` was set much larger than necessary; this was not the case for `mvprobit`.

Table 1. Estimation time comparisons (in seconds)

Parameters				Computation time		Relative difference
N	M	K	D	`mvprobit`	`bvpmvp()`	(ratio)
2,000	4	5	10	29	1	29
			20	53		53
		9	10	28	1	28
			20	54		54
	8	5	10	1,219	5	244
			20	2,041		408
		9	10	1,036	8	130
			20	2,044		256
10,000	4	5	10	142	2	71
			20	263		132
		9	10	137	3	46
			20	258		86
	8	5	10	4,628	14	331
			20	10,469		748
		9	10	4,669	19	246
			20	9,833		518
50,000	4	5	10	986	12	82
			20	1,937		161
		9	10	995	18	55
			20	1,970		109
	8	5	10	35,833	65	551
			20	72,406		1114
		9	10	36,647	86	426
			20	73,204		851

Legend

N: Number of sample observations

M: Number of outcomes

K: Number of covariates (including constant term)

D: Number of draws for `mvprobit`

Note: Stata's `matsize` parameter is set at 600 for all specifications.

Table 2. `mvprobit` and `bvpmvp()` comparison: point estimates, one example ($N = 10000$, $M = 4$, $K = 5$; estimated standard errors in parentheses)

Outcome	Covariate	`mvprobit` (draws = 20)	`bvpmvp()`
y_1	x_1	0.3265	0.3279
		(0.0448)	(0.0446)
	x_2	−0.3301	−0.3314
		(0.0447)	(0.0447)
	x_3	0.3184	0.3198
		(0.0447)	(0.0449)
	x_4	−0.3902	−0.3916
		(0.0448)	(0.0447)
	Constant	0.3901	0.3909
		(0.0466)	(0.0464)
y_2	x_1	−0.4487	−0.4487
		(0.0456)	(0.0455)
	x_2	0.5624	0.5620
		(0.0458)	(0.0456)
	x_3	−0.3998	−0.3977
		(0.0457)	(0.0457)
	x_4	0.4000	0.3961
		(0.0456)	(0.0457)
	Constant	−0.5086	−0.5079
		(0.0474)	(0.0474)
y_3	x_1	0.3102	0.3151
		(0.0445)	(0.0446)
	x_2	0.3846	0.3875
		(0.0445)	(0.0449)
	x_3	−0.3188	−0.3206
		(0.0446)	(0.0447)
	x_4	−0.3462	−0.3496
		(0.0446)	(0.0447)
	Constant	0.3230	0.3210
		(0.0463)	(0.0463)
y_4	x_1	0.4567	0.4573
		(0.0455)	(0.0457)
	x_2	−0.4438	−0.4408
		(0.0455)	(0.0457)
	x_3	−0.4489	−0.4516
		(0.0456)	(0.0457)
	x_4	0.4555	0.4499
		(0.0456)	(0.0453)
	Constant	−0.4552	−0.4524
		(0.0472)	(0.0472)

Table 3. `mvprobit` and `bvpmvp()` comparison: $\widehat{\mathbf{R}}$ point estimates, one example ($N = 10000$, $M = 4$, $K = 5$; estimated standard errors in parentheses)

$\mathbf{R}$	`mvprobit` (draws = 20)	`bvpmvp()`
ρ_{12}	0.3190	0.3308
	(0.0158)	(0.0159)
ρ_{13}	0.4942	0.5073
	(0.0134)	(0.0134)
ρ_{14}	0.2766	0.2872
	(0.0160)	(0.0161)
ρ_{23}	0.3356	0.3424
	(0.0156)	(0.0158)
ρ_{24}	0.2000	0.2034
	(0.0163)	(0.0167)
ρ_{34}	0.3059	0.3086
	(0.0157)	(0.0160)

We can see that using methods like `bvpmvp()` to fit MVP models merits consideration when reduced computation time is important.[9]

6 MVOP models

Analogous conceptual considerations arise in the context of MVOP models in which the observed ordered outcomes are $yo_j \in \{0, \ldots, G_j\}$ for finite integers $G_j \geq 1$. MVOP

9. Note that these simulations paint a somewhat "worst-case" picture for `mvprobit` estimation. The simulations use `mvprobit` "out of the box", that is, without specifying any options that might enhance estimation speed (see the help file for `mvprobit`; also see Cappellari and Jenkins [2003, 2006]). For instance, specifying a smaller number of draws (for example, `draws(3)` or `draws(5)`) would clearly result in faster estimation times; however, any diminished performance of the `mvprobit` estimator relative to the performance at a greater number of draws would be a potential consideration. Alternatively, using good starting values for $\mathbf{R}$ via `mvprobit`'s `atrho0()` option might also be expected to result in faster estimation times. One such approach would involve two stages: 1) to fit the full model using `mvprobit` with a small number of draws, for example, `draws(1)` or `draws(2)`; and 2) to use the estimate of $\mathbf{R}$ thus obtained to provide starting values for a second `mvprobit` estimation with a larger number of draws (for example, `draws(10)` or `draws(20)`) being specified. This approach—with `draws(1)` specified initially, followed by `draws(10)`—was examined in some simulations. It was observed in this instance that the two-stage approach resulted in roughly a 10% reduction in overall estimation time, due mainly to a smaller number of iterations (three versus four) required for convergence in the second stage. This article also has not considered how estimation using the `cmp` command (Roodman 2011) to fit the MVP model would compare with the `bvpmvp()` approach. I would like to thank Stephen Jenkins and an anonymous referee for their insights and suggestions on these matters.

modeling involves estimation of and inference about the parameters **B** and **R** as well as the vector of category cutpoints, **C** (for each outcome yo_j, there are G_j cutpoints that delineate the $G_j + 1$ categories).[10]

An estimation strategy fully analogous to `bvpmvp()` is not available because the `bioprobit` command (Sajaia 2008) does not permit postestimation prediction with the score option, as required by `suest`. However, an alternative, fully consistent, and computationally efficient approach is available, as follows. First, fit M univariate ordered probit models using Stata's `oprobit` command, and store these estimates using `estimates store`. This provides consistent estimates of the **B** and **C** parameters. Second, fit a chain of bivariate binary probit models using `biprobit`—as with `bvpmvp()`—and store these estimates using `estimates store`. This provides a consistent estimate of **R**.[11] Note that any thresholds used to map the ordered yo_{ij} to their corresponding coarsened binary outcomes should result in consistent estimates of **R**. `biprobit` uses the rule that a nonbinary outcome is treated as zero for zero values and one otherwise; this is a convenient mapping that minimizes programming burden. Third, combine all the estimates stored in these two steps by using `suest`. The estimates from `suest` can then be used for inference. The do-file containing the Mata code for the function `bvopmvop()` that implements this approach is available with this article's supplementary materials.[12] An example of `bvopmvop()` output is presented in exhibit 2.[13]

Exhibit 2: Sample output from `bvopmvop()` ($N = 10000$, $M = 4$, $K = 5$)

```
. mata
------------------------------------- mata (type end to exit) -------------------------------------
: yn="y1o y2o y3o y4o"
: xn="x1 x2 x3 x4"
: ic="if _n<=10000"
: bv2=bvopmvop(yn,xn,ic,1,.001,1)

  ****************************************************
  *                                                  *
  *       Multivariate Ordered Probit: Results       *
  *                                                  *
  ****************************************************

  N. of Observations (from suest): 10000

  Estimation Sample: if _n<=10000
```

10. For the MVOP model, **B** will not contain a parameter for the constant term because this is absorbed into the cutpoints **C**.
11. Note that this also provides consistent estimates of **B**, but these are unnecessary given those obtained in the first step.
12. `bvopmvop()` accommodates ordered outcomes having different numbers of cutpoints, including mixed ordered and binary outcomes. The single cutpoint estimated in `oprobit` for binary outcomes is -1 times the corresponding constant term that would be estimated using `probit`.
13. The outcomes in this example are ordered versions yo_j of the y_j used in the earlier simulations in which the outcome value 2 is assigned if $1 \leq y_j^* \leq 2$ and 3 is assigned if $y_j^* > 2$. Then, y_2 combines the top two categories, and y_3 combines the top three categories (that is, y_3 is the original binary measure). Thus the numbers of categories are $G_1 = 4$, $G_2 = 3$, $G_3 = 2$, and $G_4 = 4$.

Beta-Hat and Cutpoint Point Estimates and Estimated Standard Errors
(Note: SEs are from suest ests.)

	1	2	3	4	5
1		y1o	y2o	y3o	y4o
2					
3	x1	.379	-.457	.316	.464
4		(.038)	(.043)	(.045)	(.043)
5					
6	x2	-.325	.53	.388	-.44
7		(.038)	(.044)	(.045)	(.043)
8					
9	x3	.338	-.404	-.321	-.471
10		(.038)	(.043)	(.045)	(.043)
11					
12	x4	-.393	.397	-.348	.45
13		(.038)	(.043)	(.045)	(.043)
14					
15	cut1	-.354	.485	-.319	.447
16		(.04)	(.045)	(.046)	(.045)
17					
18	cut2	.356	1.379	--	1.305
19		(.04)	(.047)		(.047)
20					
21	cut3	1.079	--	--	2.18
22		(.041)			(.054)
23					

Estimated Correlation (Rho) Matrix and Estimated Standard Errors

	1	2	3	4	5
1		y1o	y2o	y3o	y4o
2					
3	y1o	1	.331	.507	.287
4			(.016)	(.013)	(.016)
5					
6	y2o	.331	1	.342	.203
7		(.016)		(.016)	(.017)
8					
9	y3o	.507	.342	1	.309
10		(.013)	(.016)		(.016)
11					
12	y4o	.287	.203	.309	1
13		(.016)	(.017)	(.016)	
14					

```
Cut & Paste Matrix, Beta-Hat and Cutpoint Point Estimates

(.379 , -.457 , .316 , .464) \
(-.325 , .53 , .388 , -.44) \
(.338 , -.404 , -.321 , -.471) \
(-.393 , .397 , -.348 , .45) \
(-.354 , .485 , -.319 , .447) \
(.356 , 1.379 , . , 1.305) \
(1.079 , . , . , 2.18)


Cut & Paste Matrix, Estimated Correlation Matrix

(1 , .331 , .507 , .287) \
(.331 , 1 , .342 , .203) \
(.507 , .342 , 1 , .309) \
(.287 , .203 , .309 , 1)


: end
```

7 Summary

In this article, I have presented a novel estimation strategy for consistent estimation of and inference about the parameters of MVP and MVOP models. The straightforward implementation of these approaches using available Mata programs recommends their consideration in applied work, particularly in situations involving large numbers of outcomes (M) and large sample sizes (N) or in situations requiring repeated MVP estimation (like bootstrapping exercises).

Note that the methods suggested here may prove useful in many but not all applications of MVP models. Ultimately, the methods proposed—as well as the `mvprobit` method—permit estimation of the joint conditional probability model $\Pr(\mathbf{y} = \mathbf{k}|\mathbf{x})$ for the M vectors of outcomes $\mathbf{y}$, all possible 2^M vectors $\mathbf{k} = (k_m)$, $k_m \in \{0, 1\}$, and exogenous covariates $\mathbf{x}$. As such, when these joint conditional probabilities are per se the estimands of interest, when they are instrumentally of interest in the estimation of other quantities (see Mullahy [2011] for discussion), or when reduced forms of structural models are of interest, the approach suggested here may prove useful. However, in other MVN contexts with binary outcomes—for example, where endogenous y_m are right-hand-side variables in the structural models for other latent y_j^*—consistent estimation of the structural parameters will typically demand attention to the full joint probability structure, not just its bivariate marginals.[14]

14. I thank an anonymous referee for emphasizing these points.

8 Acknowledgments

I thank Bill Greene, Stephen Jenkins, João Santos Silva, and an anonymous referee for helpful comments on earlier drafts. Support for this article was provided by the National Institute of Child Health and Human Development grant P2CHD047873 to the University of Wisconsin–Madison's Center for Demography and Ecology, by an Evidence for Action Grant from the Robert Wood Johnson Foundation, and by the Robert Wood Johnson Foundation Health and Society Scholars program at the University of Wisconsin–Madison.

9 References

Cappellari, L., and S. P. Jenkins. 2003. Multivariate probit regression using simulated maximum likelihood. *Stata Journal* 3: 278–294.

———. 2006. Calculation of multivariate normal probabilities by simulation, with applications to maximum simulated likelihood estimation. *Stata Journal* 6: 156–189.

Greene, W. H., and D. A. Hensher. 2010. *Modeling Ordered Choices: A Primer*. Cambridge: Cambridge University Press.

Mullahy, J. 2011. Marginal effects in multivariate probit and kindred discrete and count outcome models, with applications in health economics. NBER Working Paper No. 17588, The National Bureau of Economic Research. http://www.nber.org/papers/w17588.

Rao, C. R. 1973. *Linear Statistical Inference and Its Applications*. 2nd ed. New York: Wiley.

Roodman, D. 2011. Fitting fully observed recursive mixed-process models with cmp. *Stata Journal* 11: 159–206.

Sajaia, Z. 2008. bioprobit: Stata module for bivariate ordered probit regression. Statistical Software Components S456920, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s456920.html.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

About the author

John Mullahy is a professor of health economics at the University of Wisconsin–Madison.

Additional remarks on combining biprobit estimates

In general, the optimal approach to combining such multiple estimates in the overidentified case is to use a minimum-distance estimator with an optimal weight matrix

(Wooldridge 2010, sec. 14.5). In the present context, this would amount to computing a weighted average for each point estimate; that is, $\widehat{\boldsymbol{\beta}}_{jkw} = \sum_{\substack{m=1 \\ m\neq j}}^{M} w_{jkw}\widehat{\boldsymbol{\beta}}_{jkw}$, $j = 1,\ldots,M$, and $k = 1,\ldots,K$. However, implementing the minimum-distance approach can be computationally challenging. For example, consider the simplest case, $M = 3$. Even in this instance, the optimal (variance-minimizing) weights are complicated functions of the estimates' variances and covariances; suppressing the j, k subscripts, for $(p,q,r) \in (1,2,3)$, $p \neq q \neq r$, these optimal weights are

$$w_r = \frac{w_{rn}}{w_{rd}}$$

where

$$w_{rn} = \sigma_{pp}\sigma_{qq} - \sigma_{pq}^2 - \sigma_{qq}\sigma_{pr} - \sigma_{pp}\sigma_{rq} - \sigma_{pr}\sigma_{pq} - \sigma_{pq}\sigma_{rq}$$

and

$$\begin{aligned} w_{rd} = {} & \sigma_{pp}\sigma_{rr} + \sigma_{rr}\sigma_{qq} + \sigma_{pp}\sigma_{qq} - \sigma_{pr}^2 - \sigma_{pq}^2 - \sigma_{rq}^2 \\ & + 2(\sigma_{pr}\sigma_{pq} + \sigma_{pq}\sigma_{rq} + \sigma_{pr}\sigma_{rq} - \sigma_{pp}\sigma_{rq} - \sigma_{rr}\sigma_{pq} - \sigma_{qq}\sigma_{pr}) \end{aligned}$$

and where $\sigma_{\bullet\bullet}$ are variances and covariances of the parameter estimates (the empirical counterpart, $\widehat{w_r}$, would use $\widehat{\sigma_{\bullet\bullet}}$). The algebraic complexity of these weights increases rapidly as M increases.

The additional computational complexity involved in implementing such a minimum-distance approach is unlikely to be beneficial (in terms of precision) unless the optimal w_{jkm} were to diverge dramatically from $1/(M-1)$. The simulations here suggest that this is unlikely to be the case. Generally, the optimal weights will diverge from the equiweighted case of $1/(M-1)$ to the extent that the variances and covariances of and between the parameter point estimates differ substantively across the $(M-1)$ estimates.[15]

For illustration, arbitrarily selecting the $(M-1)$ point estimates corresponding to the parameter β_{11} (outcome y_1, covariate x_1) for the $N = 10000$, $M = 8$, and $K = 5$ specification, we find that the range of the 7 point estimates $\widehat{\beta_{11}}$ is $[0.3266, 0.3288]$, the range of the corresponding 7 estimated point-estimate variances is $[0.001983, 0.001995]$, and the range of the 28 estimated point-estimate covariances is $[0.001983, 0.001993]$. Therefore, it is unlikely that the optimal weights would diverge much from $1/(M-1)$.

The ultimately important result is that at least insofar as the simulations here are concerned, the differences between the `mvprobit` and `bvpmvp()` point estimates and estimated standard errors are inconsequentially small (see tables 2 and 3).

15. Bill Greene suggested that a computationally straightforward middle-ground weighting strategy would be, in essence, to ignore the cross-estimator covariances and compute the variance-matrix-weighted quantities, as follows:

$$\widehat{\boldsymbol{\beta}_{jv}} = \left[\sum_{\substack{m=1 \\ m\neq j}}^{M} \left\{\widehat{\text{var}}\left(\widehat{\boldsymbol{\beta}_m}\right)\right\}^{-1}\right]^{-1} \times \sum_{\substack{m=1 \\ m\neq j}}^{M} \left\{\widehat{\text{var}}\left(\widehat{\boldsymbol{\beta}_m}\right)\right\}^{-1} \widehat{\boldsymbol{\beta}_m}, \quad j = 1,\ldots,M$$