Estimation of nested, zero-inflated and cross-nested ordered probit models in STATA

David Dale and Andrei Sirchenko

March 13, 2017

Abstract

TBA

JEL classification: .

Keywords: ordinal responses, zero-inflated outcomes, two- and three-part mixture model, endogenous regime switching.

1 Introduction

TBA...

2 Econometric framework

Left out

3 Stata commands

Syntax of the cnop, miop, and nop commands

```
cnop depvar indepvars [if] [in] [, zp(varlist) zn(varlist) infcat(integer 0) correlated
   cluster(varname) robust initial(string)]
```

This command fits a cross-nested ordered probit model with possibly different sets of covariates for each stage and possibly correlated errors by maximum likelihood.

```
miop depvar indepvars [if] [in] [, z (varlist) infcat(integer 0) correlated cluster(varname
   robust initial(string)]
```

This command fits a middle-inflated ordered probit model.

```
nop depvar indepvars [if] [in] [, zp(varlist) zn(varlist) infcat(integer 0) correlated
   cluster(varname) robust initial(string)]
```

This command fits a nested ordered probit model.

Options

options	Description
zp(*varlist*)	list of covariates for positive response in NOP and CNOP models; by default, it equals *indepvars*, the list of covariates for initial stage
zn(*varlist*)	list of covariates for negative response in NOP and CNOP models; by default, it equals *indepvars*, the list of covariates for initial stage
z(*varlist*)	list of covariates for non-zero response in ZIOP models; by default, it equals *indepvars*, the list of covariates for initial stage
infcat(*integer*)	value of the response variable that should be modeled as inflated; by default, it equals 0
correlated	flag that errors in the first and second stages may be correlated, forcing estimation of CNOPc, NOPc or ZIOPc model
robust	flag that variance-covariance estimator must be robust (based on “sandwich“) estimate
cluster(*varname*)	clustering variable for robust variance estimator
initial(*string*)	whitespace-delimited list of initial parameter values for estimation, in the following order: β, α, γ^+, μ^+, γ^-, μ^-, ρ^-, ρ^+

Examples

TBD

Stored results

`cnop`, `nop`, and `miop` store the following in `e()`:

e(N)	number of observations
e(cmd)	`cnop`, `nop`, or `miop`, respectively
e(depvar)	dependent variable of regression
e(b)	parameters vector
e(V)	variance-covariance matrix
e(sample)	marks estimation sample

CNOP postestimation commands

The predict command

The `predict` command after `cnop`, `nop`, and `miop` estimation commands produces either predicted probabilities or expected value of the response.

`predict` *varname* [*if*] [*in*] [, `zeroes regime output(`*string*`) at(`*string*`)`]

`name` is the name of predicted variable, if it is single, or prefix for names, if there are several predicted variables

`zeroes` indicates that different types of zeroes (i.e. “intrinsic zeroes“, or “positive zeroes“, or “negative zeroes“) must be predicted instead of different response values.

`regime` indicates that different groups of response (negative, positive or zero) must be predicted instead of different response values. This option is ignored if `zeroes` option is on.

`output(string)` specifies type of aggregating predicted probabilities of different response. Possible values are `mode` and `mean`, for predicting average or most probable outcome, and `cum` for predicting cumulative response probabilities (i.e. $p(y <= -2)$, $p(y <= -1)$,

$p(y <= 0)$ etc.). If not specified, raw response probabilities are predicted ($p(y = -2)$, $p(y = -1)$, $p(y = 0)$ etc.).

The cnopmargins command

`cnopmargins [, at(`*string*`) nominal(`*varlist*`) zeroes regime]`

This command prints marginal effects for the last estimated CNOP, MIOP or NOP model, calculated at the specified point, along with confidence intervals.

`at(string)` specifies at which point predictions must be calculated. If at is specified, (as a list of `varname=value` expressions, separated by comma), prediction is calculated at this point and posted on the screen without saving to the dataset. If some covariate names are not specified, their mean value is taken instead.

`nominal` is a space-separated list of covariates which should be considered as nominal; marginal effect is then calculated as difference between values at 0 and at 1.

`zeroes` and `regime` indicate that marginal effects should be calculated for different zeroes or for groups of response variable, as in `predict` command.

The cnopprobabilities command

`cnopprobabilities [, at(`*string*`) zeroes regime]`

This command prints predicted probabilities for the last estimated CNOP, MIOP or NOP model, calculated at the specified point, along with confidence intervals. The point `at` is specified like in `cnopmargins`.

The cnopcontrasts command

`cnopcontrasts [, at(`*string*`) to(`*string*`) zeroes regime]`

This command prints differences in predicted probabilities for the last estimated CNOP, MIOP or NOP model, calculated between the specified points, along with confidence intervals. The points `at` and `to` are specified like `at` in `cnopmargins`.

Examples

TBD

4 Finite sample performance

We conducted extensive Monte Carlo experiments to illustrate the finite sample performance of the ML estimators in the proposed models.

Monte Carlo design

We conducted simulations for six data-generating processes (*dgp):* NOP, NOPc, ZIOP (MIOP version), ZIOPc (MIOPc version), CNOP, and CNOPc. The data were generated and then estimated by the same model. For each dgp we generated samples with 200, 500 and 1000 observations. The number of replications was 10,000 in each experiment.

Three vectors of covariates $\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$ were drawn in each replication as $\mathbf{v_1} \overset{iid}{\sim} Normal(0,1) + 2$, $\mathbf{v_2} \overset{iid}{\sim} Normal(0,1)$, and $\mathbf{v_3} = -1$ if $\mathbf{w} \leq 0.3$, 0 if $0.3 < \mathbf{w} \leq 0.7$, or

1 if $\mathbf{w} > 0.7$, where $\mathbf{w} \overset{iid}{\sim} Uniform[0,1]$. The dependent variable was generated with five outcome categories: -2, -1, 0, 1 and 2. The values of the parameters were calibrated to yield on average the following frequencies of the above outcomes: 7%, 14%, 58%, 14% and 7%, respectively. To avoid the divergence of ML estimates due to the problem of complete separation (perfect prediction), which could happen if actual number of observations in some outcome category (specifically, -2 and 2) is very low, the samples with any category frequency lower than 6% were re-generated. At each iteration we checked that there is at least 6% of observations in each outcome category. The matrix of the PE, therefore, has $3 \times 5 = 15$ elements; their values, which depend on the values of the explanatory variables, are computed at the population medians of the covariates. The observations in repeated samples were drawn independently. The vectors of disturbance terms in the latent equations were repeatedly generated as iid $Normal(0,1)$ random variables in the case of the NOP, ZIOP and CNOP *dgp*. In the case of the NOPc and CNOPc models, the errors $\boldsymbol{\nu}$ in the inclination equation were generated as iid $Normal(0,1)$ random variables, but the errors $\boldsymbol{\varepsilon}^-$ and $\boldsymbol{\varepsilon}^+$ in the amount equations were drawn so that $(\boldsymbol{\nu}, \boldsymbol{\varepsilon}^-)$ and $(\boldsymbol{\nu}, \boldsymbol{\varepsilon}^+)$ are the standardized bivariate normal iid random variables with the correlation coefficients ρ^- and ρ^+, respectively. In the ZIOPc dgp, the errors $\boldsymbol{\nu}^0$ in the participation equation were generated as IID $Normal(0,1)$ random variables, but the errors $\boldsymbol{\varepsilon}^0$ in the amount equation were drawn so that $(\boldsymbol{\nu}^0, \boldsymbol{\varepsilon}^0)$ are the standardized bivariate normal iid random variables with the correlation coefficients ρ^0. The repeated samples were generated for the NOP, NOPc, CNOP and CNOPc *dgp* with $\mathbf{X} = (\mathbf{v_1}, \mathbf{v_2})$, $\mathbf{Z}^- = (\mathbf{v_1}, \mathbf{v_3})$, $\mathbf{Z}^+ = (\mathbf{v_2}, \mathbf{v_3})$, and for the ZIOP and ZIOPc dgp with $\mathbf{X}^0 = (\mathbf{v}_1, \mathbf{v}_3)$, $\mathbf{Z}^0 = (\mathbf{v}_2, \mathbf{v}_3)$. The true values of the simulation parameters are shown in Table 1.

Table 1. True values of parameters for simulation

	NOP	NOPc	ZIOP	ZIOPc	CNOP	CNOPc
$\boldsymbol{\beta}$	(0.6, 0.4)′	(0.6, 0.4)′	(0.6, 0.8)′	(0.6, 0.8)′	(0.6, 0.4)′	(0.6, 0.4)′
$\boldsymbol{\alpha}$	(0.21, 2.19)′	(0.21, 2.19)′	0.45	0.45	(0.9, 1.5)′	(0.9, 1.5)′
$\boldsymbol{\gamma}^-$	(0.2, 0.3)′	(0.2, 0.3)′			(0.2, 0.3)′	(0.2, 0.3)′
$\boldsymbol{\gamma}^+$	(0.3, 0.9)′	(0.3, 0.9)′			(0.3, 0.9)′	(0.3, 0.9)′
$\boldsymbol{\mu}^-$	-0.17	-0.5			(-0.67, 0.36)′	(-0.88, 0.12)′
$\boldsymbol{\mu}^+$	0.68	1.31			(0.02, 1.28)′	(0.49, 1.67)′
ρ^-		0.3				0.3
ρ^+		0.6				0.6
$\boldsymbol{\gamma}^0$			(0.5, 0.6)′	(0.5, 0.6)′		
$\boldsymbol{\mu}^0$			(-1.45, -0.55, 0.75, 1.65)′	(-1.18,-0.33, 0.9, 1.76)′		
ρ^0				0.5		

Table 2 reports the following measures of accuracy computed for the estimates of the parameters, probabilities and PE: *Bias* — the absolute difference between the estimated and true values, devided by the true value, averaged over all Monte Carlo runs, in percent; *RMSE* — the absolute root mean square error of the parameter estimates relative to their true values, averaged over all replications; *CP* — the empirical coverage probability, computed as the percentage of times the estimated asymptotic 95% confidence intervals cover the true values. To measure the accuracy of the estimates of the standard errors, we also computed the *s.e. bias* — the absolute difference between the average of the estimated asymptotic standard errors of the estimates and the standard deviation of the estimates in all replications, in percent. The above measures of accuracy computed for the estimates of the parameters are averaged across all parameters, for the estimates of the probabilities — averaged across five outcome categories, and for the estimates of the PEs — averaged across five outcome categories and across all covariates.

The simulations and estimations were performed using the MATA programming language. The starting values for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\beta}^0$, $\boldsymbol{\mu}^0$, $\boldsymbol{\mu}^+$, $\boldsymbol{\gamma}^+$, $\boldsymbol{\mu}^-$ and $\boldsymbol{\gamma}^-$ were obtained using the independent OP estimations of each latent equation. The starting values for each independent OP model were computed using the linear OLS estimations. The starting values for ρ^0, ρ^- and ρ^+ were obtained by maximizing the logarithms of the likelihood functions of the correlated models holding the other parameters fixed at their estimates in the corresponding uncorrelated model.

Results of simulations

For each model specification, bias and RMSE of parameter estimates decrease as sample size increases. RMSE decreases in most cases faster than asymptotic rate $\sqrt{n}$. This may be caused by a small number of large deviations in parameter estimation on small samples.

For most of model pairs and sample sizes, bias and RMSE is slightly higher for the correlated version. This is expected from a more complex model, estimated on the same sample size.

Standard error estimates for parameters on average correspond to the actual standard errors. Large deviations make standard errors estimates biased, especially on small samples, but this problem rapidly decreases with sample size. Anyway, rare large deviations do not prevent asymptotic coverage probabilities of 95% confidence intervals from being consistent. This means that confidence intervals for parameter estimates may be used safely.

For estimates of outcome probabilities and marginal effects the situation is qualitatively and quantitatively similar to estimates of parameters.

In general, results of Monte Carlo simulations show that estimators of CNOP family are consistent, but should be used carefully on small samples. As a rule of thumb, we would advise using at least 10 observations per variable in each outcome class, which corresponds to 1000 observations in our case.

5 The (correlated) nested ordered probit model

Left out

A special case when the CNOP model nests the MIOP model

Left out

Table 2. Monte Carlo simulations for different models and sample sizes

Sample size	DGP:	NOP	NOPc	ZIOP	ZIOPc	CNOP	CNOPc
		Parameters					
200	Bias, %	0.04	0.07	0.12	0.08	0.10	0.20
500		0.02	0.03	0.03	0.03	0.04	0.05
1000		0.01	0.01	0.01	0.01	0.02	0.02
200	RMSE	0.51	0.86	1.40	0.37	0.45	1.03
500		0.15	0.21	0.17	0.19	0.25	0.29
1000		0.10	0.14	0.10	0.12	0.16	0.18
200	Coverage probability (at 95% level), %	95.3	88.4	93.4	90.0	92.0	84.7
500		95.1	90.5	94.4	92.9	93.1	88.9
1000		95.3	92.1	95.1	94.8	93.8	91.7
200	Bias of standard error estimates, %	19.3	16.0	35.9	16.1	22.8	12.7
500		2.5	4.2	11.2	6.9	8.1	17.3
1000		1.6	3.4	4.8	3.6	3.4	5.1
		Probabilities					
200	Bias, %	0.0018	0.0019	0.0048	0.0061	0.0041	0.0049
500		0.0008	0.0010	0.0024	0.0034	0.0020	0.0028
1000		0.0004	0.0005	0.0014	0.0019	0.0009	0.0017
200	RMSE	0.02	0.03	0.03	0.03	0.03	0.03
500		0.01	0.02	0.02	0.02	0.02	0.02
1000		0.01	0.01	0.01	0.01	0.01	0.01
200	Coverage probability (at 95% level), %	94.4	94.4	95.3	95.3	95.1	94.8
500		95.4	95.2	95.6	95.6	95.9	95.7
1000		95.5	95.5	95.7	95.7	95.6	95.6
200	Bias of standard error estimates, %	4.2	4.2	6.9	6.4	5.5	15.1
500		3.9	4.6	6.9	6.1	5.3	16.6
1000		2.6	3.4	5.7	5.9	3.7	13.9
		Marginal effects on probabilities					
200	Bias, %	0.0018	0.0023	0.0052	0.0056	0.0050	0.0052
500		0.0007	0.0011	0.0024	0.0025	0.0023	0.0024
1000		0.0003	0.0006	0.0012	0.0013	0.0011	0.0014
200	RMSE	0.02	0.02	0.02	0.04	0.03	0.03
500		0.01	0.01	0.01	0.02	0.02	0.02
1000		0.01	0.01	0.01	0.02	0.01	0.01
200	Coverage probability (at 95% level), %	95.8	93.9	91.7	87.9	94.6	91.8
500		95.9	94.6	94.8	91.5	95.0	93.0
1000		95.6	95.0	95.3	93.9	95.1	93.9
200	Bias of standard error estimates, %	4.7	5.7	8.0	6.1	21.4	39.1
500		4.0	5.0	5.8	6.0	27.0	8.1
1000		2.4	3.4	4.2	5.7	11.6	7.4