Conditional mixed-process models

Christopher F Baum

ECON 8823: Applied Econometrics

Boston College, Spring 2016

The CMP framework

We present the conditional mixed-process (CMP) framework implemented by David Roodman's `cmp` command. It is a user-written addition to Stata. In order to use it, you must give the commands `ssc install cmp` and `ssc install ghk2` when connected to the Internet.

This will install the latest version of the program, which has been updated since its description in a *Stata Journal* article, "Fitting fully observed recursive mixed-process models with cmp," 11:2, 159–206.

The do-files referred to below can be downloaded in a zip file.

Concept of CMP modeling

The underlying concept of modeling in the CMP framework is that we may often want to jointly estimate two or more equations with linkages among their error processes. There may or may not be relationships among their dependent variables. In the simplest case, these are independent equations with correlated errors.

This is indeed the concept of Zellner's Seemingly Unrelated Regression estimator, implemented in Stata as `sureg`. Using this command, we specify equations for dependent variables $y_1, y_2, \ldots, y_M$, each of could be consistently estimated by ordinary least squares. That is, each equation satisfies the crucial zero conditional mean assumption, $E[u_j|X_j] = 0$, ruling out simultaneity, or the presence of endogenous variables in the X_j.

Why might we use the SUR estimator? Because if there are meaningful correlations between the error processes u_j, the SUR estimates, taking account of those correlations, will be more efficient than those derived from single-equation OLS regressions. We also gain the ability to test for (and impose) cross-equation constraints in this framework. But the key issue here is the importance of estimating the equations jointly, using a systems approach.

SUR is a generalized least-squares estimator. However, with the `isure` option, the estimates are iterated until convergence. Those estimates are then equivalent to those derived from Full Information Maximum Likelihod (FIML) of the same model, assuming multivariate Normal error processes.

The CMP modeling framework is essentially that of seemingly unrelated regressions, but in a much broader sense. The individual equations need not be classical regressions with a continuous dependent variable. They may be binary, estimated by binomial probit; ordered, estimated by ordered probit; categorical, estimated by multinomial probit; censored, estimated by tobit; or based on interval measures, estimated by intreg.

A single invocation of `cmp` may specify several equations, each of which may use a different estimation technique.

Furthermore, `cmp` allows each equation's model to vary by observations. In the familiar Heckman selection model (e.g., Stata's `heckman`), we observe the entire sample, of whom only a subsample are selected (for instance, only some individuals work outside the home). In that context, a probit is used to estimate the probability of selection (employment), and a regression is then estimated for only those who are workers.

The maximum likelihood approach to estimating these two equations as a system, rather than as a two-step estimator, has clear benefits and potential efficiency gains. The `cmp` framework implements the systems approach, not only for traditional Heckman selection models, but for any combination of its supported components.

New features

Major features have been added to cmp since Roodman (*Stata Journal*, 2011), and are only documented in its help file. They include:

- The rank-ordered probit model is available. It generalizes the multinomial probit model to fit ranking data. See `asroprobit`.
- Truncation is now a general modeling feature rather than a regression type. This allows modeling of a pre-censoring truncation process in all models except multinomial and rank-ordered probit.
- Each equation's linear functional (XB) can appear on the right side of any equation, even when it is modeled as latent (not fully observed), and even if the resulting equation system is simultaneous rather than recursive.
- Multilevel random effects and coefficients can now be modelled, using simulation or (adaptive) quadrature. These can be correlated within and across equations.

Overview of cmp

cmp fits a large family of multi-equation, multi-level, conditional mixed-process estimators. Right-side references to left-side variables must together have a recursive structure when those references are to the observed, censored variables, but references to the (latent) linear functionals may be collectively simultaneous.

The various terms in that description can be defined as follows:

- "Multi-equation" means that cmp can fit Seemingly Unrelated (SUR) systems, instrumental variables (IV) systems, and some simultaneous-equation systems. As a special case, single-equation models can be fit too.

- "Multi-level" means that random coefficients and effects (intercepts) can be modelled at various levels in hierarchical fashion, the classic example being a model of education outcomes with unobserved school and class effects. Since the models can also be multi-equation, random effects at a given level are allowed by default to be correlated across equations. E.g., school and class effects may be correlated across outcomes such as math and readings scores. Effects at different levels, however, are assumed uncorrelated with each other, with the observation-level errors, and with the regressors.

Note that we will not further discuss the implementation of multi-level modeling in this workshop.

- "Mixed process" means that different equations can have different kinds of dependent variables (response types). The choices, all generalized linear models with a Gaussian error distribution, are: continuous and unbounded (the classical linear regression model), tobit (left-, right-, or bi-censored), interval-censored, probit, ordered probit, multinomial probit, and rank-ordered probit. Pre-censoring truncation can be modeled for most response types. A dependent variable in one equation can appear on the right side of another equation.

- "Conditional" means that the model can vary by observation. An equation can be dropped for observations for which it is not relevant–if, say, a worker retraining program is not offered in a city then the determinants of uptake cannot be modeled there. The type of a dependent variable can even vary by observation. In this sense, the model is conditional on the data.
- "Recursive" means, however, that when censored dependent variables appear in each others' equations, these references must break the equations into stages. If A, B, C, and D are all binary dependent variables, modeled as probits, then A and B could be modeled determinants of C and C as a determinant of D, but D could not then be a modeled determinant of A, B, or C.

- "Simultaneous" means that that recursivity is not required in the references to linear (latent) dependent variables. If A*, B*, C*, and D* are the hypothesized, unobserved linear functionals behind the observed A, B, C, and D–i.e., if A=0 when A*<0 and A=1 when A*>=0, etc.–then D* can appear in any of the equations even though D cannot.

Broadly, cmp is appropriate for two classes of models: 1) those in which the posited data-generating process is fully modeled; and 2) those in which some equations are structural, while others are reduced form, providing instruments for identification of the parameters in the structural equations, as in two-stage least squares.

In the first case, cmp is a full-information maximum likelihood (FIML) estimator, and all estimated parameters are structural. In the latter, it is a limited-information (LIML) estimator, and only the final stage's or stages' coefficients are structural.

Thanks to the flexibility of Stata's `ml`, on which it is built, `cmp` accepts linear coefficient constraints as well as all weight types, vce types (robust, cluster, linearized, etc.), and `svy` settings.

cmp can reproduce the results of many Stata estimation commands, but goes beyond them in many ways. It offers more flexibility in model construction. For example, one can regress a continuous variable on two endogenous variables, one binary and the other sometimes left-censored, instrumenting each with additional variables. And cmp usually allows the model to vary by observations. Equations can have different samples, overlapping or non-overlapping. Heckman selection modeling can be incorporated into a wide variety of models through auxilliary probit equations.

In some cases, the gain is consistent estimation where it was difficult before. Sometimes the gain is in efficiency. For example, if C is continuous, B is a sometimes-left-censored determinant of C, and A is an instrument, then the effect of B on C can be consistently estimated with 2SLS. However, a `cmp` estimate that uses the information that B is censored will be more efficient, based as it is on a more accurate model.

As cmp is a Maximum Likelihood estimator built on Stata's `ml`, equations are specified according to the `ml model` syntax. This means that for instrumented regressions, `cmp` differs from `ivregress, ivprobit, ivtobit` and similar commands in not automatically including exogenous regressors (included instruments) from the second stage in the first stage. So you must arrange for this yourself.

For example, `ivregress 2sls y x1 (x2=z)` corresponds to `cmp (y=x1 x2) (x2=x1 z), ind($cmp_cont $cmp_cont)`. The indicators option specifies the model to be used for each equation: in this case, `$cmp_cont`, implying linear regression with a continuous dependent variable.

At its heart cmp is an SUR (seemingly unrelated regressions) estimator. With major exceptions we will mention, it treats the equations as related to each other only in having errors that jointly normally distributed. Mathematically, the likelihood it computes is conditioned on observing all right-side variables, including those that also appear on the left side of equations.

Maximum likelihood (ML) SUR estimators are appropriate for many multi-equation models that are not SUR, meaning ones in which endogenous variables appear on the right side of other equations. Models of this kind for which ML SUR is consistent must satisfy two criteria:

1. They are recursive. In other words, the equations can be arranged so that the matrix of coefficients of the dependent variables in each others' equations is triangular. This means the models have clearly defined stages, though there can be more than one equation per stage.
2. They are "fully observed." Dependent variables in one stage enter subsequent stages only as observed. So for instance in a model containing dependent variables C and D, if C is a categorical variable modeled as ordered probit, then C, not the latent variable underlying it (C*), must enter the model for D.

The most recent versions of cmp can handle violations of both conditions by the usual method of transforming a simultaneous system into an SUR system with "reduced form" coefficients. Condition 2 is no longer required: models may refer to latent variables, using a # suffix. `cmp (y1 = y2# x1) (y2 = x2), ind($cmp_probit $cmp_probit)` models y1 and y2 as probit and y1 as depending on the unobserved linear functional behind y2. The #-suffixed references should be to names of equations rather than dependent variables, which are the same by default.

So, equivalent to the previous example is `cmp (eq1:y1 = eq2# x1) (eq2:y2 = x2), ind($cmp_probit $cmp_probit)`. In addition, references to (latent) linear dependent variables need not satisfy condition 1. So `cmp (y1 = y2# x1) (y2 = y1# x2), ind($cmp_probit $cmp_probit)` is acceptable.

Examples of cmp reproducing existing commands

We now strengthen your understanding of the CMP framework by showing how it relates to existing Stata commands, which will help you see how its syntax works. To define the cmp macros needed for the `indicators` option, you should always include the command `cmp setup` in your do-file before invoking `cmp`.

```
cmp setup
webuse laborsup, clear

reg kids fem_inc male_educ

cmp (kids = fem_inc male_educ), ind($cmp_cont) qui
```

You can execute this from do-file `cmp1.do`. The standard errors will differ, as will the RMS error, due to the large-sample statistics produced by a maximum likelihood estimator.

sureg

Next, let's consider a seemingly unrelated regression model, using the `isure` option:

```
sureg (kids = fem_inc male_educ) (fem_work = male_educ), isure

cmp (kids = fem_inc male_educ) (fem_work = male_educ), ///
ind($cmp_cont $cmp_cont) qui
```

You can execute this from do-file `cmp2.do`. Notice that the cmp output includes an estimate of the correlation between the error terms, used to gain efficiency in the SUR framework.

ivregress (2SLS)

We illustrate the equivalence of two-stage least squares via Stata's `ivregress`:

```
ivregress 2sls fem_work fem_inc (kids = male_educ), first

cmp (kids = fem_inc male_educ) (fem_work = kids fem_inc), ///
ind($cmp_cont $cmp_cont) qui
```

You can execute this from do-file `cmp3.do`. Notice that `cmp` produces both the first-stage and second-stage regression output that is shown with `ivregress, first`.

As `cmp` is a maximum-likelihood estimator, we can also draw parallels to the `ivregress` implementation of Limited-Information Maximum Likelihood (LIML):

```
ivregress liml fem_work (kids = male_educ other_inc fem_inc)

cmp (kids = fem_inc male_educ other_inc) (fem_work = kids), ///
ind($cmp_cont $cmp_cont) qui
```

You can execute this from do-file `cmp4.do`.

probit predictions and marginal effects

We illustrate how the predictions and average marginal effects of `probit` can be reproduced by `cmp`:

```
probit kids fem_inc male_educ
predict p
margins, dydx(*)

cmp (kids = fem_inc male_educ), ind($cmp_probit) qui
predict p2, pr
margins, dydx(*) predict(pr)
```

You can execute this from do-file `cmp5.do`. After `cmp`, you must use the `predict(pr)` option to specify the basis for the computation of marginal effects.

ordered probit predictions and marginal effects

Likewise for ordered probit:

```
oprobit kids fem_inc male_educ
margins, dydx(*) predict(outcome(#2))

cmp (kids = fem_inc male_educ), ind($cmp_oprobit) qui
margins, dydx(*) predict(eq(#1) outcome(#2) pr)
```

You can execute this from do-file `cmp6.do`.

Seemingly unrelated bivariate probit

```
gen byte anykids = kids > 0
biprobit (anykids = fem_inc male_educ) (fem_work = male_educ)

cmp (anykids = fem_inc male_educ) (fem_work = male_educ), ///
ind($cmp_probit $cmp_probit) qui
```

You can execute this from do-file `cmp7.do`. In this form of bivariate probit, we observe both binary outcomes for all individuals in the sample.

Instrumental variables probit

```
ivprobit fem_work fem_educ kids (other_inc = male_educ), first
margins, predict(pr) dydx(*)

cmp (fem_work = other_inc fem_educ kids) (other_inc = fem_educ kids male_educ), ///
ind($cmp_probit $cmp_cont) qui
margins, predict(pr eq(#1)) dydx(*) force
```

You can execute this from do-file `cmp8.do`.

Treatment regression

```
treatreg other_inc fem_educ kids, treat(fem_work = male_educ)

cmp (other_inc = fem_educ kids fem_work) (fem_work = male_educ), ///
ind($cmp_cont $cmp_probit) qui
```

You can execute this from do-file `cmp9.do`.

Tobit estimation

```
tobit fem_inc kids male_educ, ll(0)

cmp (fem_inc = kids male_educ), ind("cond(fem_inc, $cmp_cont, $cmp_left)") qui
```

You can execute this from do-file `cmp10.do`.

instrumental variables Tobit

```
ivtobit fem_inc kids (male_educ = other_inc fem_work), ll(0) first

cmp (fem_inc=kids male_educ) (male_educ=kids other_inc fem_work), ///
ind("cond(fem_inc,$cmp_cont,$cmp_left)" $cmp_cont) qui
```

You can execute this from do-file `cmp11.do`.

Interval regression

```
webuse intregxmpl, clear
intreg wage1 wage2 age age2 nev_mar rural school tenure

cmp (wage1 wage2 = age age2 nev_mar rural school tenure), ind($cmp_int) qui
```

You can execute this from do-file `cmp12.do`.

Truncated regression

```
webuse laborsub, clear
truncreg whrs kl6 k618 wa we, ll(0)

cmp (whrs = kl6 k618 wa we, trunc(0 .)), ind($cmp_cont) qui
```

You can execute this from do-file `cmp13.do`.

Heckman selection regression

```
webuse womenwk, clear
heckman wage education age, select(married children education age) ///
mills(heckman_mills)
gen selectvar = wage <.
cmp (wage = education age) (selectvar = married children education age), ///
ind(selectvar $cmp_probit) nolr qui
predict cmp_mills, eq(selectvar)
replace cmp_mills = normalden(cmp_mills) / normal(cmp_mills)
su heckman_mills cmp_mills
```

You can execute this from do-file `cmp14.do`. Notice the transformation needed to the `cmp` prediction to produce the Mills ratio.

Heckman probit model with sample selection

```
gen wage2 = wage > 20 if wage < .
heckprob wage2 education age, select(married children education age)

cmp (wage2 = education age) (selectvar = married children education age), ///
ind(selectvar*$cmp_probit $cmp_probit) qui
```

You can execute this from do-file `cmp15.do`.

Simultaneous equations

```
webuse klein, clear
reg3 (consump wagepriv wagegovt) (wagepriv consump govt capital1), ireg3

cmp (consump = wagepriv# wagegovt) (wagepriv = consump# govt capital1), ///
ind($cmp_cont $cmp_cont) nolr tech(dfp) qui
```

These are two of the equations of Klein's Model I, which you may execute from do-file `cmp16.do`. We use `reg3, isure` to iterate the three-stage least squares solution.

In the `cmp` specification, notice that each dependent variable appears in latent form on the RHS of the other equation, suffixed with #.

Examples of cmp's additional capabilities

The following examples illustrate models that cannot be estimated with any of Stata's official commands.

Regress an unbounded, continuous variable on an instrumented binary variable. 2SLS is consistent but is less efficient.

```
webuse laborsup, clear

cmp (other_inc = fem_work) (fem_work = kids), ind($cmp_cont $cmp_probit) qui robust


ivregress 2sls other_inc (fem_work = kids), robust
```

You can execute this from do-file `cmp17.do`.

Regress a continuous variable on a left-censored variable (female income), which is only modeled for observations in which the woman works.

```
gen byte ind2 = cond(fem_work, cond(fem_inc, $cmp_cont, $cmp_left), $cmp_out)

cmp (other_inc=fem_inc kids) (fem_inc=fem_educ), ind($cmp_cont ind2) qui robust
```

You can execute this from do-file `cmp18.do`.

Instrumental variables ordered probit

```
cmp (kids = fem_educ) (fem_educ = fem_work), ind($cmp_oprobit $cmp_cont) ///
tech(dfp) nolr qui
margins, dydx(*) predict(eq(#1) pr outcome(#1)) force
margins, dydx(*) predict(eq(#1) pr outcome(#2)) force
```

You can execute this from do-file `cmp19.do`.

Ordered probit with Heckman selection modeling

```
webuse womenwk, clear
gen selectvar = wage<.
gen wage3 = (wage > 10)+(wage > 30) if wage < .
tab wage3 selectvar, mi
cmp (wage3 = education age) (selectvar = married children education age), ///
ind(selectvar*$cmp_oprobit $cmp_probit) qui
```

You can execute this from do-file `cmp20.do`.

Bivariate seemingly unrelated ordered probit

```
webuse laborsup, clear
gen byte kids2 = kids + int(uniform()*3)
tab kids2 kids
cmp (kids=fem_educ) (kids2=fem_educ), ind($cmp_oprobit $cmp_oprobit) ///
nolr tech(dfp) qui
predict prNK*, pr
su prNK*
```

You can execute this from do-file `cmp21.do`.

Summary

In summary, the CMP modeling framework provides some significant extensions to official Stata's estimation capabilities, and provides you with the opportunity to estimate a broader set of models. This may be particularly important in a micro data or panel micro data context, where many interesting questions require that you take issues such as selection, categorical measurements, and the like into account.

We have not discussed the applicability of the CMP framework to multilevel mixed (hierarchical) models. There are a number of examples in the `cmp` help file that discuss those capabilities. Likewise, `cmp` may be used in the context of a complex survey sampling design, in which you would use the `svy:` prefix or, in `cmp`, the `svy` option.