Extended grid user guide

2013-05-28 PsN-3.6.2

Important note: The algorithm description in this guide is complete, but other information is still missing.

Introduction

Non-parametric estimation on the extended grid. <reference>

Examples

extended_grid run1.mod

Input and options

extended_grid-specific input

A model file is required on the command-line. No other options are required.

-dv_column=<column_name>	The dependent variable column. Default DV.
-id_column=<column_name>	The header of the column marking individuals. Default ID.
-ids_to_simulate=N	The number of individuals to simulate. Default is the number of individuals in the original dataset.
-omega_multiplier=number	Default 1. The initial omegas in the simulation model and extended model will be multiplied by this number. This gives inflation (or deflation if the multiplier < 1) of the variance.

Some common PsN-options useful with extended_grid

For a complete list of common options see common_options_defaults_versions.pdf, or psn_options -h on the commandline.

-directory=extended_grid_dirN	The directory in which the script will run NONMEM can be named. The default name is “extended_grid_dirN” where N is increased by 1 each time you run the script. If the run is aborted or crashes, setting the directory to the one from which the script was running earlier can be done. PsN will then not run the model files that had finished, saving time.
-seed=N	A seed for the random number generator can be specified. This makes the run reproducible.
-help	With -help a longer help message will be printed.

Output

File final_nptab.csv

Known bugs and problems

If

Technical overview of algorithm

1) A run directory extended_grid_dirN is created (unless option -dir is set to something different).
2) The input model, without any changes, is run in the orig_modelfit_dir subdirectory. It is not checked that the input model does not contain NONP. **Improvement:** Add check that input model does not contain NONP.
3) The input model is copied to the simulation model called simulation.mod, along with the original dataset where the copy is called simulation.dta.
4) Modifications in the simulation model:

   a) The initial estimates are updated with the final estimates from the input model run.

   b) If option -omega_multiplier is set: For each OMEGA initial value, replace the initial value 'init' with 'init'*omega_multiplier. Diagonal and non-diagonal values are treated alike.

   c) Remove $ESTIMATION, $COVARIANCE, $TABLE.

   d) Set $SIMULATION (<seed>) ONLYSIM

   Improvement: Add (seed2 UNIFORM) if new option -uniform is set.

   e) Count the number of OMEGAs and set

   $TABLE ID ETA1 ETA2 ... ETA<nOMEGA> FIRSTONLY FILE=sim

   Note: 'ID' in this table is hard coded and is not affected by option id_column

   Improvement: Replace ID based on id_column. Also add NOAPPEND NOPRINT ONEHEADER

   f) If option -ids_to_simulate=N is set: If the number is smaller than or equal to the number of individuals in the dataset then all but the N first individuals are removed from simulation.dta. Else (N is larger than the number of individuals in the dataset) then the first individual in simulation.dta is copied N-<original number of ids> times and those individuals are appended so that simulation.dta contains N individuals in total. After modifications the IDs are renumbered ascending.

   Note: Handling of individuals is not affected by option id_column, so it is likely that option -ids_to_simulate will fail if id_column is not 'ID'.

   Improvement: Add support for -id_column and -ids_to_simulate in combination
5) The simulation model is run in the simulation_dir subdirectory.
6) The input model is copied to the extended grid model called extended.mod, along with the original dataset where the copy is called extended.dta.

7) Modifications in the extended grid model:

a) The initial estimates are updated with the final estimates from the input model run.

b) If option -omega_multiplier is set: For each OMEGA initial value, replace the initial value 'init' with 'init'*omega_multiplier. Diagonal and non-diagonal values are treated alike.

c) In extended.dta append a column of zeros and set header 'TYPE'. Also append rows for 'ids_to_simulate' (see help for this option) individuals, where each of these appended individuals have nOMEGA rows. On the *i*th row of extra individual *j* the DV column (either default DV or set with option dv_column) is set to the ETA*i* value for individual *j* read from simulated file sim (steps 4e and 5 above), the ID column (either default ID or set with option id_column) is set to *j* and the TYPE column to *i*.

d) Append TYPE to $INPUT

e) Edit $ERROR (no warning if $ERROR not present): Wherever EPS(n) is found replace it with ETA(n+n_etas_in_original_model). Enclose whole $ERROR block with 'IF (TYPE.LE.0) THEN' and 'ENDIF' unless a line starting with 'IF' was found in the $ERROR block. For each ETA*i* in the original model, append to $ERROR a line

IF(TYPE.EQ.*i*) Y = ETA(*i*)+ERR(2)*W/100000

f) For each ETA added in step 7e, add a $OMEGA with initial value 1

g) Add $SIGMA 1 FIX

h) set option MAXEVAL=0 POSTHOC in $ESTIMATION (no check that estimation method is classic)

i) remove $COVARIANCE

j) Add $NONPARAMETRIC MARGINALS MSFO=MSF01 UNCONDITIONAL

k) in $PRED/$PK code block (error message if neither is found) append line

JD = DEN_

and for each ETA*i* in the original model a line

DN*i*=CDEN_(*i*)

l) Add $TABLE ID ETA1 ... ETA<n_original_omega> JD DN1 ... DN<n_original_omega> NOPRINT NOAPPEND FIRSTONLY FILE=nptab

8) The extended grid model is run in the extended_dir subdirectory.

9) Analyze the output in file nptab.

a) First transform the JD column in two steps. In the first step leave the first <number of original individuals> rows unchanged, then transform the remaining rows according to new_value=old_value-(1/Ntotal) where Ntotal is the total number of rows in the JD column. In the second step transform all rows according to new_value =old_value*Ntotal/Noriginal where Noriginl is the number of original individuals.

b) For each eta column, sort the ETAs and the transformed JD values as a group in ascending order based on the ETA values. Then create a new column with the cumulative sum of the sorted JD values.

c) Print a final output table ‘final_nptab.csv’ with the transformed JD column followed by all pairs of sorted ETA and cumulative JD sum columns.