NONPB user guide, PIND user guide

2013-05-28 PsN 3.6.2

Reference and introduction

Paul G. Baverel, Radojka M. Savic, Mats O. Karlsson. Two bootstrapping routines for obtaining imprecision estimates for nonparametric parameter distributions in nonlinear mixed effects models. *J Pharmacokinet Pharmacodyn* 2010.

When parameter estimates are used in predictions or decisions, it is important to consider the magnitude of imprecision associated with the estimation. The nonpb is a resampling-based method for estimating imprecision in nonparametric distribution (NPD) estimates obtained in NONMEM. Imprecision in the NPD can be estimated by means of two different resampling procedures. The full method, -nonpb_version=2, relies on bootstrap sampling from the raw data and a re-estimation of both the preceding parametric (FOCE) and the nonparametric step. The simplified method, -nonpb_version=1, relies on bootstrap sampling of individual nonparametric probability distributions. Nonparametric confidence intervals are computed. In addition of providing information about the precision of nonparametric parameter estimates, the nonpb methods can serve as diagnostic tools for the detection of misspecified parameter distributions.

The P individuals method is availabe as a separate PsN script pind.

Example

nonpb run12.mod -samples=500

pind run12.mod ….

Input and options

Options to nonpb

A model file is required on the command-line.

-samples=N	The samples option is required. The number of samples in the bootstrap.
-lst_file=file.lst	Name of the lst-file with estimates for the input model. Default is the name of the input model with a .mod extension replaced with .lst.
-nonpb_version=X	The version of the nonpb method. Default is 1, the simplified version. Version can also be set to 2, the full version.

Options to pind

A model file is required on the command-line.

-lst_file=file.lst	Name of of the lst-file with estimates for the input model. Default is the name of the input model with a .mod extension replaced with .lst.

-ind_param=eta *or* theta	Optional, default eta. The parameter to set in individual ofv model files. Possibilities are eta or theta.

Some general PsN-options which are useful in combination with nonpb or pind

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

-directory=run1_nonpb	The directory in which the script will run NONMEM can be named. The default name is “nonpb_dirX” where X is increased by 1 each time you run the script.
-seed=X	A seed for the random number generator can be specified. This makes the run reproducible. It acts as a starting point for the random number generator when creating the bootstrapped data sets.
-threads=X	The number of parallel processes to start for the model runs on a parallel computer.
-help	With -help PsN will print a longer help message.

Output

Intermediate files written during the procedure are found in the intermediate_files subdirectory of the nonpb run directory. In the result_files subdirectory the files CI_results_ETAk.csv are found, where k=1,...,(number of ETAs). CI_results_ETAj.csv contains the mean, median, 25^{th}, 50^{th}, 90^{th} and 95^{th} confidence intervals.

Recovering a crashed/stopped nonpb

No special methods implemented. Try starting with the same command, with -directory set to existing run directory. Do not set option rerun.

Technical overview of algorithm

1. Only if version 2 – full (version 1 simplified starts at step 3) : Create S bootstrapped datasets and estimate the input model with each of the S bootstrap datasets.
2. For input model (using original data, original model file and original final parameter estimates) and for *each* bootstrapped model, S+1 in total: Update initial estimates from the final parameter estimates of the bootstrap output run.
3. If version 2 (full): For S models from step 2 (excluding original model file), compute individual probabilities using procedure described in P-individuals section – this procedure ends up with S files P_values_j.csv, j=1,...,S and S table files bs_model_j.patab. For the original model file from step 2, modify and run model according to steps 1a-1h of P-individuals section, giving table file original.patab.

If version 1 (simplified): Compute individual probabilities for single input model using procedure described in P-individuals section – this procedure gives the file P_values.csv and the table file original.patab.

4. If version 2 (full): For each P_values_j.csv file, bootstrap individual vectors (columns) according to the bootstrap scheme in step 1; e.g. pick up individual vectors corresponding to the individuals contained in the bs_pr1_j.dta – name this new file P_values_bootstrapped_j.csv

If version 1 (simplified): bootstrap individual vectors (columns) of P_values.csv according to a newly generated bootstrap scheme S times. Name the S new files P_values_bootstrapped_j.csv, j=1,...,S

5. For each P_values_bootstrapped_j.csv file, sum up rows.
6. Create a new file called bootstrapped_np_probabilities_j.csv which will contain ID column, all ETA values (ETA1-ETAX) from table file from step 3, if version 2 from bs_model_j.patab or if version 1 from original.patab S times, and BOTP column containing sum of the rows from step 5 (in total 2+X columns, X being number of ETAs). Note: Sum column represents probabilities.
7. For each bootstrapped_np_probabilities_j.csv, for each ETA_k in file bootstrapped_np_probabilities_j.csv, transform BOTP column into cumulative probability:
   For each ETA_k:
   - Sort the ETA column and BOTP column ascending (according the ETA value)
   - Transform BOTP column into cumulative probability column (add them up cumulatively)
   - Output file(s) (X files for each j) called bootstrap_CBJD_ETAk_j.csv which will contain 3 columns: ID, ETA_k, cumulative bootstrapped JD (CBJD))
8. For each ETA_k, create a final file row1_results_ETA_k.csv. From the file bootstrap_CBJD_ETAk_j.csv, copy columns ETA_k and CBJD and rename them ETAk_j and CBJD_j (will refer to these two columns as j*th column pair* in following text)
9. From a file row1_results_ETA_k.csv, create a new file, bootstrap_CBJD_ETAk_adjusted.csv in which each ETAk_j vector will be replaced with the ETA_k vector from the original.patab from step 3 (sorted ascending) and each CBJD_j vector will be adjusted so (CBJD_j_adjusted), that for each element of the ETA_k vector in the Xth column, CBJD_j value corresponding for the first lower element (first ETAk_j value lower than ETAk value) of the ETAk_j column will be chosen.
10. Compute confidence intervals (3*X files): a) Sort ascending ETAk values from original.patab from step 3 b) For each ETAk create a file called row2_results_ETAk.csv which will contain S+1 columns and i rows. First column contains ascending original ETAi values from step a). S columns contain X values of and CBJD_j_adjusted found in step 9, each row represents results for each ETAk_i.
11. For each ETAk: For each ETAi (row) in file row2_results_ETAk.csv, read out the mean, median, 25^{th}, 50^{th}, 90^{th} and 95^{th} confidence intervals (based on N values). Store results in CI_results_ETAk.csv which will contain 11 columns and I rows

P individuals procedure

1. a) Update initial estimates from the final parameter estimates of the original output (pind.lst) and FIX them (THETA/OMEGA and SIGMA) b)Skip estimation step by setting MAXEVALS=0 c) Add $NONPARAMETRIC UNCONDITIONAL d)Add in $PK if it exists, otherwise in $PRED:
   JD = DEN_
   DNj = CDEN_(j)
   where j= 1...X (which gives X rows) and where X is the number of ETAs in original model.
   e) Add a nptab table to print ID, JD, ETAX and DNj. Set FIRSTONLY. f) Remove $COVARIANCE g) Remove all other existing tables of original model file. h) Run the model (lst file named “np.lst”)

2. Read the nptab file produced in step 2 and create matrix that contains the Joint Density (JD) in a column (output the JD column of nptab into a separate file called jdtab). The sum of values in jdtab should equal 1 (within rounding error).
3. For each row of the jdtab, denoted “i”: a) create a copy of the modified model from step 1 and replace each ETAj in the $PK, $PRED and $ERROR records in the model file with the value (number) for that ETAj obtained from the nptab file. b) Add DATA=(ID) to $CONTR. c) Add $SUBROUTINE record if not already present in modelfile. d) Add CONTR=iofvcont.f to $SUBROUTINE, where iofvcont.f is custom CONTR routine that prints individual objective function values to file "fort.80" e) Fix OMEGAs to zero f) Remove $TABLE and $NONPARAMETRIC g) Run the model with MAXEVAL=0 (lst file named “ofv_i.lst”) h) Extract the last N lines from fort.80 file into a new file called "ofv_model_i", where N is the number of individuals.
4. From the N files called "ofv_model_i" create a new file called “ind_ofv”. This file will have N columns and N rows and should be created using the following principle: starting with i=1, the values of "ofv_model_i" should be transposed to generate the first row of the file “ind_ofv”. Thereafter the same procedure with i=2, 3, …N.
5. Computation of Likelihood and individual probabilities: For each value (iOFV) in the matrix ind_ofv, calculate: iL = exp(-iOFV/2). Then do a scalar multiplication between this matrix (iL) and the matrix with JD values (from jdtab such as

[row1 (iL) (N values)*row1 (JD) (single value), row2 (iL)*row2 (JD)….]

(output into file “L_times_P.csv” which thus will have dimension = N * N)

6. Sum up each column in the resulting L_times_P.csv matrix. Output it as sum_LP.csv file (dimensions: 1(row)*N (columns)
7. Divide each column vector (each value in the column) of L_times_P.csv matrix, with the value from the corresponding column of sum_LP.csv (single value) and divide it with N. (output it as P_values.csv file)