

BOOT_SCM user guide

PsN 4.6.0

Revised 2016-04-13

1 Overview

The `boot_scm`, or bootstrap of the `scm`, program is an implementation of the method presented in [1]. The program depends heavily on the `scm` program, and all `scm` options except `base_ofv` apply also to `boot_scm`. Please refer to `scm_userguide.pdf` for help on `scm` options. Examples

```
boot_scm config_run1_nonlinear.scm -samples=100 -seed=12345
boot_scm config_run1_linearize.scm -samples=100 -seed=12345 -methodA
```

2 Input and options

2.1 Required input

A configuration file is required on the command-line. The format of the configuration file follows the format of the `scm` configuration file exactly. The input model must be set in the configuration file, it cannot be given on the `boot_scm` commandline. In addition to the configuration file, one command-line option is mandatory:

-samples = N

The number of bootstrapped datasets to run the `scm` on.

2.2 Optional input

These options are specific to `boot_scm`, and they can only be given on the command-line, not in the configuration file.

-methodA

Default not set. If the scm option `linearize=1` is not set in the scm config file, the bootstrap scm non-linear method will be used. If option `linearize=1` is set in the scm config file, by default the bootstrap scm linear method B (see algorithm description below) will be used. If option `linearize=1` is set together with option `-methodA` on the `boot_scm` command-line (no argument to `-methodA`) then the bootstrap scm linear method A will be used. If `linearize=1` is not set and option `-methodA` is set this will result in an error message. Setting `linearize=1` in the scm config file by default gives linearization using FOCE, for details see the scm userguide.

-run_final_models

Default not set. If set then `boot_scm` will run the final models from each scm on the original dataset and collect the ofv values in the output file `ofv_final.csv`

-dummy_covariates = *comma-separated list of covariates*

Default not used. If used, a new column for each listed covariate will be added to the dataset, containing a randomly permuted copy of the original covariate column and with header `X<name of original covariate>`. The dummy covariate will be tested for inclusion in the covariate model exactly like the original covariate. However, a known bug is that `boot_scm` will not correctly create a dummy covariate based on a time-varying covariate.

-stratify_on = *item in \$INPUT*

Default not used. It may be necessary to use stratification in the resampling procedure. For example, if the original data consists of two groups of patients - say 10 patients with full pharmacokinetic profiles and 90 patients with sparse steady state concentration measurements - it may be wise to restrict the resampling procedure to resample within the two groups,

producing bootstrap data sets that all contain 10 rich + 90 sparse data patients but with different compositions. Set `-stratify_on` to the column (the name in `$INPUT` in the model) that defines the two groups.

2.3 Some common PsN-options useful with `boot_scm`

For a complete list see `common_options.pdf`, or `psn_options -h` on the commandline.

-h or -?

Print the list of available options and exit.

-help

With `-help` all programs will print a longer help message. If an option name is given as argument, help will be printed for this option. If no option is specified, help text for all options will be printed.

-directory = 'string'

Default `boot_scm_dirN`, where `N` will start at 1 and be increased by one each time you run the script. The `directory` option sets the directory in which PsN will run `NONMEM` and where PsN-generated output files will be stored. You do not have to create the directory, it will be done for you. If you set `-directory` to a the name of a directory that already exists, PsN will run in the existing directory.

-seed = 'string'

You can set your own random seed to make PsN runs reproducible. The random seed is a string, so both `-seed=12345` and `-seed=JustinBieber` are valid. It is important to know that because of the way the Perl pseudo-random number generator works, for two similar string seeds the random sequences may be identical. This is the case e.g. with the two different seeds 123 and 122. Setting the same seed guarantees the same sequence, but setting two slightly different seeds does not guarantee two different random sequences, that must be verified.

-clean = *'integer'*

Default 1. The clean option can take four different values:

0 Nothing is removed

1 NONMEM binary and intermediate files except INTER are removed, and files specified with option `-extra_files`.

2 model and output files generated by PsN restarts are removed, and data files in the `NM_run` directory, and (if option `-nmqual` is used) the xml-formatted NONMEM output.

3 All `NM_run` directories are completely removed. If the PsN tool has created `modelfit_dir:s` inside the main run directory, these will also be removed.

-nm_version = *'string'*

Default is 'default'. If you have more than one NONMEM version installed you can use option `-nm_version` to choose which one to use, as long as it is defined in the `[nm_versions]` section in `psn.conf`, see `psn_configuration.pdf` for details. You can check which versions are defined, without opening `psn.conf`, using the command

```
psn -nm_versions
```

-threads = *'integer'*

Default 5 (if default PsN config file is used). Use the `threads` option to enable parallel execution of multiple models. This option decides how many models PsN will run at the same time, and it is completely independent of whether the individual models are run with serial NONMEM or parallel NONMEM. If you want to run a single model in parallel you must use options `-parafile` and `-nodes`. On a desktop computer it is recommended to not set `-threads` higher the number of CPUs in the system plus one. You can specify more threads, but it will probably not increase the performance. If you are running on a computer cluster, you should consult your system administrator to find out how many threads you can specify.

-version

Prints the PsN version number of the tool, and then exit.

-missing_data_token = 'string'

Default -99. This option sets the string that PsN accepts as missing data, and needs to be set correctly when PsN computes summary statistics for data set columns.

3 Algorithm overview

If IGNORE/ACCEPT is found in \$DATA (not counting single character IGNORE like e.g. IGNORE=@), the data will be filtered using a dummy model run in the preprocess_data_dir subdirectory of the boot_scm directory. The new dataset is called filtered.dta. A modified input model called orig_model_filtered_data.mod is created where the new dataset is used.

If -dummy_covariates is set, a modified input model (based on orig_model_filtered_data.mod or on the original input model if no filtering was done) called model_with_xcov.mod is created where the dummy covariates are added in \$INPUT and \$DATA specifies a new dataset called xcov_⟨old data name⟩ where the dummy covariates are added. The new model and dataset is created in preprocess_data_dir.

When using method A or Non-linear (i.e. if option linearize=1 is not set in the scm config file, or options linearize=1 and boot_scm option -methodA are both set): The program creates 'samples' bootstrapped datasets from the possibly pre-processed original dataset. Then a regular scm is run on each of these datasets, using the options set in the configuration file. Filtering on IGNORE/ACCEPT is skipped in these scm runs, since filtering was done during preprocessing if necessary.

When using method B (i.e. if option linearize=1 is set in the scm config file but not option -methodA on the boot_scm commandline): The program runs the possibly pre-processed input model with the possibly pre-processed dataset using the options set in the scm configuration file and terminates the run directly after the derivatives dataset has been generated. Then 'samples' bootstrapped datasets are created from the derivatives dataset. A regular scm is run on each bootstrapped dataset, using the options set in the scm configuration file.

In addition to the options in the scm configuration file, the bootstrapped derivatives data is used as input with option -derivatives_data (this is done

automatically, the user should not set this option), which makes the scm run faster since the derivatives generation step can be skipped. In these scm runs the filtering on IGNORE/ACCEPT is skipped, since filtering was done during pre-processing.

If there are time-varying covariates (option `time_varying` is set in the original configuration file) each scm run will include a run with the original, non-linear model on a bootstrapped version of the possibly pre-processed original dataset, using the same individuals in each sample as in the bootstrapping of the derivatives dataset. This extra run is needed to compute medians for the time-varying covariates.

If option `-run_final_models` is set: Run the final models from each scm on the original, possibly pre-processed, dataset.

4 Output

The file `bs_ids.csv` contains one row per bootstrapped dataset and one column per individual in the bootstrapped dataset. The value in each column gives the original data ID of that individual. The file `ofv_final.csv` is only created if option `run_final_models` is set. It contains one row per bootstrapped dataset plus one for the original, possibly pre-processed, model. It lists the ofvs of the final models from the scm, rerun on the original dataset. The file `covariate_inclusion.csv` has one row per bootstrapped dataset. There is one column per parameter-covariate-state combination possible given the `test_relations` and `valid_states` settings in the configuration file, excluding state 1 (which means 'not included'). For each bootstrapped dataset the value in the column is 1 if the relation is included in the final model, and 0 otherwise.

References

- [1] R. J. Keizer, A. Khandelwal, A. C Hooker, and M. O. Karlsson. “The bootstrap of Stepwise Covariate Modeling using linear approximations”. In: *PAGE 20 Abstr 2161* (2011).