MIMP user guide

PsN 4.6.0

Revised 2015-09-16

1 Overview

The mimp program implements the multiple imputation method [1], [2]. Example

```
mimp -base_model=base.mod -reg_model=reg.mod -mi_model=mi.mod
```

2 Input and options

2.1 Required input

Unlike other PsN scripts mimp does not take a bare model file name as input. All model file names are given as options. The dataset needs to contain a column that depicts for each row in the dataset if the covariate that is partly missing is missing or observed.

-base_model = *filename*

Base model. This is the model without the code snippet describing the effect of the covariate (that is partly missing) on the parameters. The individual parameter (empirical Bayes) estimates, of the parameter(s) on which the covariate has an effect, contains information about the individual response. The individual parameter estimates should be printed in $TABLE of the base model. The model must contain a complete $TABLE containing items exactly matching $INPUT of the next

model in the sequence. PsN does not check that $TABLE is correctly defined.

-reg_model = *filename*

Regression model. The regression model estimates the relationship between the partly missing covariate, the response (individual parameter estimates from the base model) and other (completely observed) covariates that carry information about the missing covariate. The partly missing covariate will be the dependent variable (DV) in this model and the column in the dataset that depicts if the covariate is missing or observed will be the missing dependent variable (MDV). The regression model must contain a complete $TABLE containing items exactly matching $INPUT of the multiple imputation model.

-mi_model = *filename*

Multiple imputation model. This model must contain one simulation part and one estimation part, followed by both a $SIM and a $EST record. The simulation part should contain the code for imputation of the missing covariate while the estimation part should be the final model including the code for the effect of the covariate on the parameters.

2.2 Optional input

-imputations = *M*

Default 6. The number of imputations to perform for each dataset.

-sim_model = *filename*

Simulation model. This model must contain a correct $SIM record and a complete $TABLE containing items exactly matching $INPUT of the base model. PsN does not check that $TABLE is correctly defined.

-samples = *N*

Number of datasets to simulate. Only relevant together with option -sim_model, required if sim_model is given.

-chain_models = *file1,file2,...*

An ordered list of chain models. These models can be used if there is a need for additional estimation models in between the base model and the regression model. The models must be listed in the order in which they are to be estimated. Each model must contain a complete $TABLE containing items exactly matching $INPUT of the next model in the sequence (the next chain model or the regression model if it is the last chain model). PsN does not check that $TABLE is correctly defined.

-alt_models = *model_1, model_2,...*

A comma-separated list of filenames of alternative models.

2.3 Some important common PsN options

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 mimp_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.

3 Output

The results are in the raw_results file of mi_dirJ, J=1:'imputations', (the multiple imputation subdirectories).

4 Algorithm overview

1. If option -sim_model is used: Create 'samples' copies of the simulation model and set a unique seed in each of them. Set FILE in $TABLE to a unique name (using order number of simulation file). Run the simulation models.

2. If option -sim_model is used: Create 'samples' copies of the base model. In each copy set filename in $DATA to a new simulated dataset.

3. In the base model set FILE in $TABLE to a unique name (repeat this 'samples' times, once for each copy of the base model, if -sim_model was used).

4. Run the base model (or 'samples' copies of the base model).

5. Repeat for each model given via option chain_models, if any: set filename in $DATA to the filename in $TABLE from the previous model (base model or previous chain model). Set FILE in $TABLE to a unique name. Run the chain model (or 'samples' copies of the chain model, each with different $DATA filename).

6. Set filename in $DATA of the regression model to the filename in $TABLE from the previous model in the sequence (either from the base model or the last chain model). Set FILE in $TABLE to a unique name. Run the regression model (or 'samples' copies of the regression model, each with different $DATA filename).

7. For each alternative model, if defined: Set filename in $DATA of the alternative model to the filename in $TABLE from the regression model. Set FILE in $TABLE to a unique name. Run the alternative model (or 'samples' copies of the alternative model, each with different $DATA filename).

8. Create 'imputations' copies of the multiple imputation model (or 'imputations' times 'samples' copies if -sim_model was used). Set filename in $DATA of the multiple imputation model to the filename in $TABLE from the regression model. Each filename from regression $TABLE is set in 'imputations' copies of the multiple imputation model. Set a unique seed in the $SIM of each multiple imputation model. Run the multiple imputation models.

References

[1] Å. Johansson and M. O. Karlsson. "Multiple Imputation of Missing Covariates in NONMEM and Evaluation of the Method's Sensitivity to η-Shrinkage". In: *AAPS J. 2013 Oct; 15(4): 1035–1042* (2013).

[2] Å. Johansson and M. O. Karlsson. "Comparison of methods for handling missing covariate data". In: *AAPS J. 2013 Oct; 15(4): 1232–1241* (2013).