MIMP userguide

2013-05-28 PsN 3.6.2

Overview

The mimp program implements the multiple imputation method described in
Åsa Johansson “Comparison of methods for handling missing covariate data”
PAGE 20 (2011) Abstract 1982 [www.page-meeting.org/?abstract=1982]

Example

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

Input and options

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.

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.

Some common PsN-options useful with mimp

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

-directory=mimp_dirN	The run directory can be named.
-seed=X	Random seed.
-help	With -help mimp will print a longer help message.

Output

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

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.