STooDs configuration files

July 5, 2021

1 Introduction

The configuration files aim at specifying the key ingredients of a STooDs case study:

1. A **dataset** containing values taken by one or several predictand variables $\boldsymbol{Y}$ varying in space, time or other dimensions, along with the values taken by potential covariates $\boldsymbol{X}$ (aka predictors).
2. A **model** making assumptions on the probabilistic mechanism that generated the data. Typically, a STooDs model can be schematized as follows:
   (a) each variable in the dataset follows a distribution.
   (b) the parameters of this distribution may vary in space, time or other dimensions.
   (c) this variability is specified by formulas that combine parameters, covariates and processes.
3. The **prior distribution** specified for each parameter (if any).
4. The **hyperdistribution** specified for each process (if any).
5. the properties of the **MCMC sampler** used to explore the posterior distribution associated with the model and the dataset.

Typical configuration files can be found in the folder 'Examples'. For a given case study, **all configuration files** should be located in the same **workspace** folder. **Data files**, however, can be stored anywhere on the computer. Such data files include the file containing the $\boldsymbol{X}/\boldsymbol{Y}$ dataset (point 1. above) and additional files describing the used dimensions (e.g. location of stations for space, years for time, etc.).

2 Data file and dataset configuration file

The data file should be structured as follows:

1. (compulsory) A single column containing all data values for the predictand $\boldsymbol{Y}$.
2. (compulsory) A single column containing the variable index. If there is a single variable, the whole column should be equal to 1. If there are K variables, integers 1...K should be used to indicate the variable associated with each row.
3. (optional) One or several columns containing the covariates values $\boldsymbol{X}$.
4. (optional) One or several columns containing the dimension indices. For instance, if 'time' and 'space' dimensions are used, a first column should contain the time step and a second column the site number associated with each row. The names of these columns should correspond to (or at least contain) the names of the dimensions.
5. (optional) A single column containing the censoring type of the data. 0 corresponds to interval censoring, any negative number corresponds to a 'less than' censoring (true value is smaller than Y[i]), any positive number corresponds to 'more than' censoring (true value is larger than Y[i]).
6. (optional) A single column containing the width of the censoring interval for each data. True value is supposed to be in Y[i]+/- width[i] (0 thus leads to no censoring)

The dataset configuration file specifies how the data file should be interpreted and it contains the following lines:

1. (string) A descriptive name for the dataset.
2. (string) The file where the dataset is stored. It is recommended to use quotes and to write the full path to the data file.
3. (integer) The number of header lines in the data file.
4. (integer) The number of rows in the data file (excluding header lines).
5. (integer) The number of columns in the data file
6. (integer) The column containing the values of the predictand $\boldsymbol{Y}$.
7. (integer) The column containing the variable index.
8. (integer) The columns containing the values of the covariates $\boldsymbol{X}$ (comma-separated if several covariates, 0 if no covariate).
9. (integer) The columns containing the dimension indices (comma-separated if several dimensions, 0 if no dimension is used).
10. (integer) The column containing the censoring type (0 for no censoring).
11. (integer) The column containing the censoring interval width when interval censoring is used (0 if not used).

3 Model configuration file

The model configuration file should be named 'model.config' and it contains the following lines:

1. (string) A descriptive name for the model.
2. (integer) The number of variables nVar.
3. (string) The name of each variable (size nVar, comma-separated).
4. (string) The parent distribution of each variable (size nVar, comma-separated). See section 7.1 for a list of available distributions.
5. (integer) The number of parameters for each parent distribution (size nVar, comma-separated).
6. (string) The name of each parent parameter (size sum(nParentPar), comma-separated).
7. (integer) The number of covariates nCov.
8. (string) The name of each covariate (size nCov, comma-separated).
9. (integer) The number of model parameters nPar.
10. (string) The name of each model parameter (size nPar, comma-separated).
11. (string) The configuration file for model parameters (priors).
12. (integer) The number of dimensions nDim.
13. (string) The configuration file for each dimension (size nDim, comma-separated).
14. (integer) The number of processes nPro.
15. (string) The name of each process (size nPro, comma-separated).
16. (string) The configuration file for each process (size nPro, comma-separated).
17. (string) The formula for deriving the first parent parameter from model parameters, processes and covariates.
18. ...
19. (string) The formula for deriving the last parent parameter from model parameters, processes and covariates.
20. (string) The dataset configuration file.

4 Parameter configuration file

The parameter configuration file contains the specifications for the model parameters declared in lines 9-11 of the previous model configuration file. For each parameter, a block of 4 lines needs to be specified:

1. (string) parameter name. Make sure that the name is consistent with the ones declared in line 10 of the model configuration file.
2. (real) initial guess (will be used as a starting point for MCMC sampling). Make sure this starting point does not lead to a zero prior or likelihood, otherwise MCMC sampling won't start.
3. (string) prior distribution. See section 7.1 for a list of available distributions.
4. (real) prior parameters, comma-separated.

Make sure that the parameters appear in the same order as declared in line 10 of the model configuration file.

5 Dimension and process configuration files

A process can be viewed as a parameter varying along a given dimension. Note that several processes can be defined along the same dimension (e.g. several spatial processes). Dimension and process configuration files are therefore both needed as soon as at least one process is used in the model.

5.1 Dimension data and configuration files

There should be as many dimension data and configuration files as there are dimensions. A dimension data file should be structured as follows:

1. (compulsory) One or several columns containing the coordinates of points in the dimension. For space this could be 2 columns containing the geographical coordinates of stations; for time this could be one column containing the years associated with time steps.
2. (optional) One or several columns containing covariates that vary in this dimension only. For space this could be elevation, distance to sea, etc. For time this could be a climate index.

The dimension configuration file then specifies how this data file should be interpreted along with other properties of the dimension:

1. (string) A descriptive name for the dimension.
2. (string) The file where the dimension dataset is stored. It is recommended to use quotes and to write the full path to the data file.
3. (integer) The number of header lines in the dimension data file.
4. (integer) The number of rows in the dimension data file (excluding header lines).
5. (integer) The number of columns in the dimension data file.
6. (integer) The number of coordinates associated with the dimension (typically 1 for time, 2 for space, etc.).
7. (integer) The columns containing the coordinates, comma-separated.
8. (integer) The number of covariates associated with the dimension.
9. (string) The names of the covariates (comma-separated if several).
10. (integer) The columns containing the values of the covariates (comma-separated if several, 0 if no covariate).
11. (string) The function used to compute distances in the dimension. Only available at the moment: "Euclidean" or "Haversine" (for lon-lat coordinates).
12. (integer) The number of parameters of the distance function. Not properly implemented yet, use 0.

5.2 Process configuration file

There should be as many process configuration files as there are processes used in the model. A process configuration file contains the following lines:

1. (string) A descriptive name for the process. Make sure that the name is consistent with the ones declared in line 15 of the model configuration file.
2. (string) The name of the dimension the process is defined on. Make sure it is consistent with the names declared in the dimension configuration files.
3. (integer) The constraint index. 0 means that the process is unconstrained; 1 means that the process is forced to have mean zero; 2 means that the process is forced to have mean zero and standard deviation one. Such constraints are necessary in some models to avoid non-identifiability.
4. (real or string) Initial guess value for the process (will be used as a starting point for MCMC sampling). If a real number is provided, it is used as the initial guess for all values of the process (i.e. at every point of the corresponding dimension, i.e. at every site for a spatial process, every time step for a temporal process, etc.). If a string is provided, it should correspond to the name of a dimension covariate (as declared in line 9 of the dimension configuration file) which is then used as initial guess.
5. (string) The (hyper)-distribution of the process. Should be a multivariate distribution as listed in section 7.2.
6. (integer) The number of parameters used to specify the hyper-distribution.
7. (string) The name of each hyper-parameter (comma-separated if several). Make sure that the names are not already used elsewhere, otherwise an error will be thrown.
8. (integer) The number of scalar parameters nSpar: see explanations in section 7.2.
9. (integer) The indices of scalar parameters in the full parameter list of line 7 (size nSpar, comma-separated).
10. (string) The formula used to compute the vector parameter of the hyper-distribution (typically, the mean vector). It may depend on any of the hyper-parameters declared in line 7 and any covariate of the dimension. See section 7.2 for more explanation.
11. (string) The formula used to compute the matrix parameter of the hyper-distribution (typically, the covariance matrix). It may depend on any of the hyper-parameters declared in line 7, on any covariate of the dimension and on pairwise distances of the dimension, noted Ddimname (e.g. Dspace, Dtime). See section 7.2 for more explanation.
12. For each hyper-parameter declared in line 7, a 4-line parameter block giving the name, initial guess, prior distribution and prior parameters as described in section 4.

6 MCMC configuration file

The MCMC configuration file should be named 'mcmc.config' and it contains the following lines:

1. (integer) Nsim, the total number of MCMC iterations.
2. (integer) Nadapt, the adaptation period: jump sizes are increased/decreased every Nadapt iterations to comply with the desired moving rates.
3. (integer) stopAdapt: no more adaptation is performed after iteration number stopAdapt.
4. (integer) Nburn, the number of 'burnt' iterations that will not be written to file.
5. (integer) Nslim, the slimming factor: only one iteration every Nslim will be written to file.
6. (real) minMoveRate, the lower bound for the desired move rate interval.
7. (real) maxMoveRate, the upper bound for the desired move rate interval.
8. (real smaller than 1) downMult, the multiplication factor used to decrease jump size when move rate is too low.

9. (real larger than 1) upMult, the multiplication factor used to increase jump size when move rate is too high. Avoid using 1/dowMult otherwise the jump size will just oscillate between 2 values.

10. (string) The name of the MCMC result file (name of the file only, not full path).

11. (logical .true. or .false.) useSpeedUp, use computational tricks to optimize speed? Should always be .true., except if one wishes to verify the equivalence speedUp/no speedUp.

12. (string) The name of the file reporting MCMC jump sizes at each adaptation (name of the file only, not full path). No reporting if empty.

13. (string) The name of the file reporting MCMC move rates at each adaptation (name of the file only, not full path). No reporting if empty.

7 Appendix

7.1 Available distributions

The table below gives the univariate distributions available in STooDs and their parameterization. Unless mentioned otherwise, Wikipedia's parameterizations are used.

ID	**Parameters**	**Comments**
Gaussian	2: mean and standard deviation	
Uniform	2: lower and upper bound	
Triangle	3: peak, lower bound and upper bound	
LogNormal	2: mean and standard deviation of log-variable	
LogNormal3	3: threshold, mean of log-excesses and standard deviation of log-excesses	
Exponential	2: threshold and scale	The scale is used rather than the rate
GPD	3: threshold, scale and shape	Negative shape = left-bounded and heavy right tail
Gumbel	2: location and scale	
GEV	3: location, scale and shape	Negative shape = left-bounded and heavy right tail
GEV_min	3: location, scale and shape	Positive shape = left-bounded and heavy right tail
Inverse_Chi2	2: degrees of freedom and scale	It is a SCALED inverse chi-squared distribution
PearsonIII	3: location, scale and shape	
Beta	2: shape1, shape2	
Kumaraswamy	2: shape1, shape2	
Geometric	1: success probability	Support starts at one, not zero
Poisson	1: rate	
Bernoulli	1: success probability	
Binomial	2: success probability and number of trials	
NegBinomial	2: success probability and number of failures	
FlatPrior	0	Improper distribution: U[-∞;+∞]
FlatPrior+	0	Improper distribution: U[0;+∞]
FlatPrior-	0	Improper distribution: U[-∞;0]
FIX	0	Pseudo-distribution used to fix the value of a parameter at its initial guess

7.2 Available multivariate distributions

Parameterization of multivariate distributions is more difficult than that of univariate distributions: it is not practical to stack all parameters into a vector. Consider for instance the symmetric covariance matrix of a multivariate Gaussian distribution and the many ways it can be flatten into a vector: this leaves too much room for confusion and error.

Consequently, p-variate distributions in STooDs can use 3 types of parameters:

1. scalar parameters such as the mean and the standard deviation of an IID multivariate Gaussian distribution, or the lag-1 autocorrelation coefficient of an AR(1) process.

2. a vector parameter of size p: typically, the mean vector.

3. a matrix parameter of size p*p: typically, the covariance matrix.

The table below gives the univariate distributions available in STooDs and their parameterization.

ID	**scalar parameters**	**vector par.**	**matrix par.**
Gaussian_IID	2: mean and standard deviation	0	0
Gaussian	0	1: mean	1: covariance
Gaussian_AR1	3: constant, innovation st. dev. and lag-1 autocorrelation	0	0
Gaussian_AR1_vmean	2: innovation st. dev. and lag-1 autocorrelation	1: mean	0
NNGP (nearest-neighbour Gaussian process)	1: number of neighbours	1: mean	1: covariance
Flat	0	0	0