Spatio-Temporal Expectile Regression Models
README

Elmar Spiegel[1,2], Thomas Kneib[1], Fabian Otto-Sobotka[3]
[1]University of Goettingen
[2]Helmholtz Zentrum München
[3]Carl von Ossietzky University Oldenburg

Contents

1 General remarks

Below you find a description of the R-code and the additional materials provided in the supplementary material. The files are separated between joint functions, application and simulation study.

To reproduce the content of the article apply the R-code as described below. For generating the figures of the paper also apply the LaTeX-files `Fig1.tex`, ..., `Fig5.tex` which aggregate and format the figures produced via R.

2 Joint functions

`function_expectreg_mgcv_bam11.R`

- *Description*: Functions to estimate spatial-temporal expectile regression.
- *Input*: No other files, just the `mgcv` package.
- *Output*: Estimated model
- *Process*: Combination of functions to estimate the model. The user should just apply the `expectreg_bam_smooth` function, where all options for the sub-functions, can be specified.
- *Runtime*: depends on input.

Main function for expectile regression

```
expectreg_bam_smooth(formula, data, expectiles, sm_par_vec, fixed = FALSE,
delta = 0.0001, step_max = 10, reltol = 1e-8, initial_w = TRUE,
opt_type=c("gcv","schall"), dev_schall = c("dev","gcv","non"),
quietly=FALSE, bounce_off=FALSE)
```

- Input
  - `formula`: is the formula of the model, similarly as in the `mgcv` package
  - `data`: are the data to be analyzed
  - `expectiles`: is the current asymmetry level. Just a scalar possible, no vector!
  - `sm_par_vec`: are the initial smoothing parameters
  - `fixed`: specifies whether the smoothing parameters are fixed or should be estimated
  - `delta`: is the convergence level of the Schall algorithm
  - `step_max`: is the maximal number of step halving iterations in the smoothing parameter estimation via the Schall algorithm
  - `reltol`: is the relative tolerance in the optimization of the smoothing parameter via `optim` (Nelder-Mead)
  - `initial_w`: is a binary variable to decide whether initial weights should be used in the estimation procedure. This will increase the speed.
  - `opt_type`: Whether the Schall algorithm, or the GCV optimization should be applied.
  - `dev_schall`: Which type of criterion should be applied to check, whether the new smoothing parameter results in better estimates. Either MWSE (dev), gcv, or none
  - `quietly`: Should the procedure run in silence or trace should be shown
  - `bounce_off`: If in the Nelder-Mead optimization of the GCV, a large fake GCV should be returned when the proposed smoothing parameter is not in the interval (1e-5,1e5). This is done since the original Nelder-Mead does not consider limits.
- Output: List containing
  - Output of the optimization algorithm
  - Estimated model based on the optimized smoothing parameters
    - `model`: Final `mgcv::bam`-model
    - `w1`: Final weights for the asymmetry
    - `gcv`: Final GCV value
    - `expectile`: asymmetry level
    - `it`: Number of iterations between coefficient estimation and calculation of weights
    - `fitted`: Fitted values
    - `coefficients`: Coefficients of the model
    - `sm_par_vec`: Smoothing parameters of the final model
- Details: Semiparametric regression consists of 2 optimizations: coefficients and smoothing parameters. Here we have an inner function that estimates the coefficients based on the iterative LAWS scheme. Therefore, we apply `mgcv::bam` to estimate the coefficients, calculate then the weights and estimate new weighted coefficients. This is iterated until the weights stay constant and the algorithm converges. The smoothing parameters are optimized outside this inner function of the coefficient estimation. Therefore, we have here two possibilities: direct GCV optimization via Nelder-Mead (`stats::optim`) or the self-implemented Fellner-Schall algorithm, which is described in detail in the main paper.

3 Simulation Study

- `Sim4_100_ti_T_....R`
  - *Description*: File to run the simulation study for one parameter combination:
    - `_homo` vs. `_hetero` for homoscedastic or heteroscedastic data
    - `_1` vs. `_2` for simulation design 1 or 2
    - `_2000` vs. `_5000` for the number of observations

  - *Input*:
    * Packages: `mgcv`, `MASS`, `Matrix`
    * `function_expectreg_mgcv_bam11.R` function to run semiparametric expectile regression
  - *Output*: Simulation results as one `.RData` file
  - *Process*:
    * Load libraries and source
    * Define functions to simulate the data and to evaluate the model output
    * Assign matrices to summarize the results
    * Simulate the data
    * Estimate the models in for loops for the different data sets, asymmetry levels and smoothing parameter selection methods; evaluate the models
    * Save output
  - *Runtime*: 10+ days

- `Simulation_Aggregate_Paper.R`
  - *Description*: File to aggregate the outputs of the different simulation runs and to plot the figures of the paper.
  - *Input*:
    * Packages: `mgcv`
    * `.RData` files of the simulation runs
  - *Output*: 1 pdf with several pages showing the results of the simulation study. Each picture shows the results for one asymmetry parameter ($\tau = 0.01, 0.02, 0.05, 0.10, 0.20, 0.50, 0.80, 0.90, 0.95, 0.98, 0.99$) as in the paper (where we did not show $0.2, 0.8$ for space reasons.). The number of observations is always 5000.
  - *Process*: Load the data, rearrange the matrices, rename the columns, plot.
    This is done in for loops based on the parameters PMWSE or MWSE and asymmetry level. However, MWSE is not displayed.
  - *Runtime*: Couple of minutes

- `Simulation_Aggregate_Supplement.R`
  - *Description*: File to aggregate the outputs of the different simulation runs and to plot the figures of the supplementary material.
  - *Input*:
    * Packages: `mgcv`
    * `.RData` files of the simulation runs
  - *Output*: 1 pdf with several pages showing the results of the simulation study. Each picture shows the results for one asymmetry parameter ($\tau = 0.01, 0.02, 0.05, 0.10, 0.20, 0.50, 0.80, 0.90, 0.95, 0.98, 0.99$) and either 2000 or 5000 observations as in the supplementary material
  - *Process*: Load the data, rearrange the matrices, rename the columns, plot.
    This is done in for loops based on the parameters PMWSE or MWSE and asymmetry level and number of observations. However, MWSE is not displayed.
  - *Runtime*: Couple of minutes

4 Application

To get the estimates as in the paper and the supplementary material the ordering is:

1. Estimate the models based on `Anisotrop_Model_....R` and `GMRF_Model_....R`. Take care these files only estimate the model for one asymmetry level and each file takes several days. The memory size should be 5GB RAM.

2. Build the plots with the files `Anisotrop_Aggregate_results_Paper.R` and `GMRF_Aggregate_results_Paper.R` for the paper version and `Anisotrop_Aggregate_results_Supplement.R` and `GMRF_Aggregate_results_Supplement.R` for the version of the supplement. Here the runtime are just several minutes but 15GB RAM are necessary.

4.1 R-files

- `plot_legend.R`
  - *Description*: File to build the legends of the maps. Based on code of the R-package `BayesX`
  - *Input*: No file, the function needs the limits of the effect and contains options for the display of the figure.
  - *Output*: Picture of the legend
  - *Process*: Assigns and plots legend as polygons and adds texts
  - *Runtime*: Milliseconds

- `Plot_ObservationStations_3d.R`
  - *Description*: Plots the location of the observation stations and the histogram of the temperatures
  - *Input*:
    - Packages: `mgcv`, `spatstat`, `sp`, `maptools`, `raster`, `rgeos`
    - `KL_Tageswerte_Beschreibung_Stationen.txt` meta-data of the observation stations
    - `Table_Temperatur_new3d.txt` data
    - `DEU_adm0.rds` borders of Germany
    - `GermanyTopo.tif` altitude of Germany
  - *Output*: 2 pdf:
    - 1 pdf with several pages showing the locations of the observation stations and the histogram of the temperatures.
    - 1 pdf showing the legend for Figure 2.
  - *Process*: Load the data, plot the figures
  - *Runtime*: Couple of minutes

- `Anisotrop_Model_....R`
  - *Description*: Estimates the model based on the coordinates for 1 asymmetry level ($\tau \in \{0.01, 0.02, 0.05, 0.10, 0.20, 0.50, 0.80, 0.90, 0.95, 0.98, 0.99\}$)
  - *Input*:
    - Packages: `mgcv`, `Matrix`, `MASS`, `BayesX`
    - `function_expectreg_mgcv_bam11.R` function to run semiparametric expectile regression
    - `Table_Temperatur_new3d.txt` data
    - `DEU_ROR.bnd` borders of Raumordnungsregionen Germany (Just for similarity with `GMRF_Model_...`)
    - `DEU_ROR.gra` neighborhood structure of Raumordnungsregionen Germany (Just for similarity with `GMRF_Model_...`)

- *Output*: 1 `.RData` file of the estimated model
- *Process*: Load the packages; set parameters of the models; load functions; define name of the model; load data; load Raumordungsregionen; Clean data; define formula; estimate initial model for initial smoothing parameters; estimate model; save output
- *Runtime*: 16+ days

- `GMRF_Model_....R`
  - *Description*: Estimates the model based on Raumordnungsregionen (regional aggregation) for 1 asymmetry level ($\tau \in \{0.01, 0.02, 0.05, 0.10, 0.20, 0.50, 0.80, 0.90, 0.95, 0.98, 0.99\}$)
  - *Input*:
    - Packages: `mgcv`, `Matrix`, `MASS`, `BayesX`
    - `function_expectreg_mgcv_bam11.R` function to run semiparametric expectile regression
    - `Table_Temperatur_new3d.txt` data
    - `DEU_ROR.bnd` borders of Raumordnungsregionen Germany
    - `DEU_ROR.gra` neighborhood structure of Raumordnungsregionen Germany
  - *Output*: 1 `.RData` file of the estimated model
  - *Process*: Load the packages; set parameters of the models; load functions; define name of the model; load data; load Raumordungsregionen; Clean data; define formula; estimate initial model for initial smoothing parameters; estimate model; save output
  - *Runtime*: 16+ days

- `Anisotrop_Aggregate_results_Paper.R`
  - *Description*: Summarizes the output of the models based on the coordinates and plots the figures for the paper
  - *Input*:
    - Packages: `mgcv`, `raster`, `colorspace`, `BayesX`
    - `plot_legend.R` function to plot the legend
    - `.RData` model outputs
    - `DEU_adm0.rds` borders of Germany
    - `GermanyTopo.tif` altitude of Germany
    - `KL_Tageswerte_Beschreibung_Stationen.txt` meta-data of the observation stations
  - *Output*: Several pdf containing the plots for the effects and the legends
  - *Process*: Load the packages; Load the outputs of the models; Load the Meta-Data; Build data sets for the predictions of univariate effects and the temporal effects; Build the predictions for these data sets; Load spatial meta data; Build data set for spatial predictions; Build spatial forecasts per specific day; Define characteristics of plots (colors, limits,...); Plot univariate effects and daily effect; Plot spatial effect; Plot temporal effect for specific cities.
  - *Runtime*: Couple of minutes

- `Anisotrop_Aggregate_results_Supplement.R`
  - *Description*: Summarizes the output of the models based on the coordinates and plots the figures for the supplementary material
  - *Input*:
    - Packages: `mgcv`, `raster`, `colorspace`, `BayesX`
    - `plot_legend.R` function to plot the legend
    - `.RData` model outputs

    - * `DEU_adm0.rds` borders of Germany
    - * `GermanyTopo.tif` altitude of Germany
    - * `KL_Tageswerte_Beschreibung_Stationen.txt` meta-data of the observation stations
  - *Output*: Several pdf containing the plots for the effects and the legends
  - *Process*: Load the packages; Load the outputs of the models; Load the Meta-Data; Build data sets for the predictions of univariate effects and the temporal effects; Build the predictions for these data sets; Load spatial meta data; Build data set for spatial predictions; Build spatial forecasts per specific day; Define characteristics of plots (colors, limits,...); Plot univariate effects and daily effect; Plot spatial effect; Plot temporal effect for specific cities.
  - *Runtime*: Couple of minutes

- `GMRF_Aggregate_results_Paper.R`
  - *Description*: Summarizes the output of the models based on the Raumordnungsregionen and plots the figures for the paper
  - *Input*:
    - * Packages: `mgcv`, `raster`, `colorspace`, `BayesX`
    - * `plot_legend.R` function to plot the legend
    - * `.RData` model outputs
    - * `DEU_adm0.rds` borders of Germany
    - * `GermanyTopo.tif` altitude of Germany
    - * `KL_Tageswerte_Beschreibung_Stationen.txt` meta-data of the observation stations
  - *Output*: Several pdf containing the plots for the effects and the legends
  - *Process*: Load the packages; Load the outputs of the models; Load the Meta-Data; Build data sets for the predictions of univariate effects and the temporal effects; Build the predictions for these data sets; Load spatial meta data; Build data set for spatial predictions; Build spatial forecasts per specific day; Define characteristics of plots (colors, limits,...); Plot univariate effects and daily effect; Plot spatial effect; Plot temporal effect for specific cities.
  - *Runtime*: Couple of minutes

- `GMRF_Aggregate_results_Supplement.R`
  - *Description*: Summarizes the output of the models based on the Raumordnungsregionen and plots the figures for the supplementary material
  - *Input*:
    - * Packages: `mgcv`, `raster`, `colorspace`, `BayesX`
    - * `plot_legend.R` function to plot the legend
    - * `.RData` model outputs
    - * `DEU_adm0.rds` borders of Germany
    - * `GermanyTopo.tif` altitude of Germany
    - * `KL_Tageswerte_Beschreibung_Stationen.txt` meta-data of the observation stations
  - *Output*: Several pdf containing the plots for the effects and the legends
  - *Process*: Load the packages; Load the outputs of the models; Load the Meta-Data; Build data sets for the predictions of univariate effects and the temporal effects; Build the predictions for these data sets; Load spatial meta data; Build data set for spatial predictions; Build spatial forecasts per specific day; Define characteristics of plots (colors, limits,...); Plot univariate effects and daily effect; Plot spatial effect; Plot temporal effect for specific cities.
  - *Runtime*: Couple of minutes

4.2 Other files

- `Table_Temperatur_new3d.txt` Data set for regional & anisotrop model based on data of the Deutscher Wetterdienst (DWD) (data set available at `https://www.uni-goettingen.de/de/zfs+working+papers/511092.html`)
- `KL_Tageswerte_Beschreibung_Stationen.txt` Meta data of the observation stations originally from the Deutscher Wetterdienst (DWD)
- `DEU_ROR.bnd` BND file of the borders of the Raumordnungsregionen based on data of the BBSR
- `DEU_ROR.gra` Neighborhood structure of the Raumordnungsregionen based on data of the BBSR
- `DEU_adm0.rds` Borders of Germany
- `GermanyTopo.tif` Topological profile of Germany

4.3 Notes

- The original data was downloaded at 2017 04 23 from Deutscher Wetterdienst DWD: `ftp://ftp-cdc.dwd.de/pub/CDC/observations_germany/climate/daily/kl/historical/`
- There it is provided one zip-folder per observation station.
  Note: The DWD changed their data structure at 2017 06 01
- The final data sets where built and saved as `Table_Temperatur_new3d.txt`.

5 SessionInfo

To get the configurations of the server we applied the files: `SessionInfo_Application.R` and `SessionInfo_Simulation.R`

The server was set up under `Ubuntu 17.10` and `R Version 3.4.4`. The used packages were

- `stats`, `graphics`, `grDevices`, `utils`, `datasets`, `methods`, `base`
- `BayesX_0.2-9`, `foreign_0.8-69`, `MASS_7.3-47`, `Matrix_1.2-14`, `mgcv_1.8-22`, `nlme_3.1-137`, `raster_2.6-7`, `rgeos_0.3-28`, `rpart_4.1-13`, `shapefiles_0.7`, `spatstat.data_1.2-0`
- `abind_1.4-5`, `coda_0.19-1`, `colorspace_1.3-2`, `compiler_3.4.4`, `deldir_0.1-15`, `goftest_1.1-1`, `grid_3.4.4`, `lattice_0.20-35`, `maptools_0.9-2`, `polyclip_1.8-7`, `Rcpp_0.12.14`, `rgdal_1.2-20`, `sp_1.2-5`, `spatstat_1.55-1`, `spatstat.utils_1.8-0`, `splines_3.4.4`, `tensor_1.5`