This form documents the artifacts associated with the article (i.e., the data and code supporting the computational findings) and describes how to reproduce the findings.

Part 1: Data

☐ This paper does not involve analysis of external data (i.e., no data are used or the only data are generated by the authors via simulation in their code).

☒ I certify that the author(s) of the manuscript have legitimate access to and permission to use the data used in this manuscript.

Abstract

The data analyzed in the paper consist of simulation data and precipitation data over the contiguous United States. The simulation data in Section 4 is generated via `R/generate_sim_data.R`. In Section 5, the data we used are average precipitation data aggregated from the daily data over October 1, 2017 to September 30, 2018. The daily data are originally from the Global Historical Climate Network-Daily database, preprocessed by Dr. Mark Risser, and publicly available at his personal website. We focus on observations at 1939 stations located to the west of 90°W, which are available in `data/CONUS_precip_west_18.csv`.

Availability

☒ Data **are** publicly available.
☐ Data **cannot be made** publicly available.

If the data are publicly available, see the *Publicly available data* section. Otherwise, see the *Non-publicly available data* section, below.

Publicly available data

☐ Data are available online at:

☒ Data are available as part of the paper's supplementary material.

☐ Data are publicly available by request, following the process described here:

☐ Data are or will be made available through some other mechanism, described here:

Non-publicly available data

Description

File format(s)

☒ CSV or other plain text.
☒ Software-specific binary format (.Rda, Python pickle, etc.): .RData
☐ Standardized binary format (e.g., netCDF, HDF5, etc.):
☐ Other (please specify):

Data dictionary

☒ Provided by authors in the following file(s): `data/README.md`
☐ Data file(s) is(are) self-describing (e.g., netCDF files)
☐ Available at the following URL:

Additional Information (optional)

Part 2: Code

Abstract

The model implementation and data analysis in the paper were done in R. The `R/` folder provides code for 1) generating simulation inputs, 2) fitting SPGP to simulation and precipitation data via Markov chain Monte Carlo (MCMC), 3) Bayesian predictive inference of SPGP, 4) obtaining results from the competing method TGP, and 5) reproducing key figures and tables in the paper.

Description

Code format(s)

☒ Script files

- ☒ R
- ☐ Python
- ☐ Matlab
- ☐ Other:

☐ Package

- ☐ R
- ☐ Python
- ☐ MATLAB toolbox
- ☐ Other:

☒ Reproducible report

- ☒ R Markdown
- ☐ Jupyter notebook
- ☐ Other:

☐ Shell script
☐ Other (please specify):

Supporting software requirements

Version of primary software used R version 3.6.3.

Libraries and dependencies used by the code Required R packages and their version numbers are listed in `dependencies.txt`.

The implementation of nearest neighbor Gaussian process (NNGP) approximation used in SPGP is based on the BRISC C/C++ library, which is included in `BRISC/` and publicly available at https://github.com/ArkajyotiSaha/BRISC.

Supporting system/hardware requirements (optional)

Parallelization used

☐ No parallel code used
☒ Multi-core parallelization on a single machine/node

- Number of cores used: 5

☐ Multi-machine/multi-node parallelization

- Number of nodes and cores used:

License

☒ MIT License (default)
☐ BSD
☐ GPL v3.0
☐ Creative Commons
☐ Other: (please specify)

Additional information (optional)

The code provided in this repository consists of a main R Markdown file `reproduce.Rmd`, which calls R scripts in `R/` to reproduce key results in the paper.

The R code in `R/` is organized as follows:

- Code for simulation studies:
  - `generate_sim_data.R`: Code to generate simulation data inputs.
  - `SPGP_func_iso.R`: Code to implement isotropic SPGP model.
  - `SPGP_sim_select_L.R`: Code to select hyperparameter L via DIC.
  - `SPGP_sim_fit_model.R`: Code to fit SPGP model via MCMC and make prediction at hold-out locations.
  - `SPGP_sim_predictive_field.R`: Code to generate predictive fields from SPGP.
  - `TGP_sim_fit_model.R`: Code to fit competing method TGP and make prediction at hold-out locations.
  - `TGP_sim_predictive_field.R`: Code to generate predictive fields from competing method TGP.
  - `SPGP_sim_results.R`: Code to reproduce key figures and tables in simulation studies.
- Code for precipitation data analysis:
  - `SPGP_func_aniso.R`: Code to implement anisotropic SPGP model.
  - `SPGP_application_fit_model.R`: Code to fit SPGP model via MCMC and make prediction at hold-out locations.
  - `SPGP_application_predictive_field.R`: Code to generate predictive fields from SPGP.
  - `TGP_application_fit_model.R`: Code to fit competing method TGP and make prediction at hold-out locations.
  - `TGP_application_predictive_field.R`: Code to generate predictive fields from competing method TGP.
  - `SPGP_application_results.R`: Code to reproduce key figures and tables in precipitation data analysis.
- `utils.R`: Utility functions.

Part 3: Reproducibility workflow

Scope

The provided workflow reproduces:

- ☐ Any numbers provided in text in the paper
- ☐ The computational method(s) presented in the paper (i.e., code is provided that implements the method(s))
- ☐ All tables and figures in the paper
- ☒ Selected tables and figures in the paper, as explained and justified below: This workflow reproduces all tables and figures related to SPGP and its main competitor TGP in the key sections of the paper, including 1) Section 4 (Simulation Studies), 2) Section 5 (Real Data Analysis), and 3) Sections S3.2 and S4 of the Supplementary Material, which provide supplementary results to Sections 4 and 5, respectively.

Workflow

Location

The workflow is available:

- ☒ As part of the paper's supplementary material.
- ☐ In this Git repository:
- ☐ Other (please specify):

Format(s)

- ☐ Single master code file
- ☐ Wrapper (shell) script(s)
- ☒ Self-contained R Markdown file, Jupyter notebook, or other literate programming approach
- ☐ Text file (e.g., a readme-style file) that documents workflow
- ☐ Makefile
- ☐ Other (more detail in *Instructions* below)

Instructions

All workflow information to reproduce key results of the paper is contained in `reproduce.Rmd`. The steps for running the workflow are:

Step 1: Install all required R packages in `dependencies.txt`.

Step 2 (Optional): Compile the source code for the BRISC library by running the following command in your terminal:

```
R CMD SHLIB BRISC/BRISC.cpp BRISC/BRISC_Predict.cpp BRISC/util.cpp BRISC/init.c BRISC/lbfgs.c
```

This step is optional and only required if you want to re-run SPGP model fitting and prediction.

Step 3 (Optional): Modify the options in Line 19-23 of `reproduce.Rmd` accordingly. By default, key figures and tables are reproduced from pre-computed model outputs in `data/`. To re-generate model outputs, make the following change(s):

- To re-generate simulation input data, set `gen_sim_input = TRUE`.
- To re-run SPGP model in simulation, set `fit_SPGP_sim = TRUE`.
- To re-run TGP model in simulation, set `fit_TGP_sim = TRUE`.
- To re-run SPGP model in precipitation data application, set `fit_SPGP_app = TRUE`.
- To re-run TGP model in precipitation data application, set `fit_TGP_app = TRUE`.

Please note that re-running these models can take several hours.

Step 4: Run the workflow in `reproduce.Rmd` by compiling the R Markdown file. For example, this can be done by running the following command in your terminal:

```
Rscript -e "rmarkdown::render('reproduce.Rmd')"
```

Reproducible results can be found in the generated `reproduce.html` file.

Expected run-time

Approximate time needed to reproduce the analyses on a standard desktop machine:

☐ < 1 minute
☐ 1-10 minutes
☐ 10-60 minutes
☐ 1-8 hours
☒ > 8 hours
☐ Not feasible to run on a desktop machine, as described here:

Additional information (optional)

Notes (optional)