MATLAB code instruction for the Generalized Sparse Precision matrix Selection (GSPS) algorithm for fitting univariate/multivariate, isotropic/anisotropic Gaussian Random Field (GRF) models

`http://arxiv.org/abs/1405.5576`

Sam Davanloo Tajbakhsh

`https://u.osu.edu/davanloo/`

System Requirement

The code works on Macintosh and Windows operating systems. On Macintosh systems, the code is tested on OS X Mavericks, Yosemite, and El Capitan. On Windows systems, it is tested on Windows 7 and 8. These tests are done on MATLAB 2014 and 2015b.

The SPS Package

The GSPS package includes a script driver file *driverGeneralSPS.m*. All of the functions are available in the *SPS_V05* folder which should be added to the MATLAB path.

The Training Data

Let $X \in \mathbb{R}^{n\times d}$ be the matrix of input/independent variables with d features and $Y \in \mathbb{R}^{n\times N\times 1}$ (for *univariate* response) or $Y \in \mathbb{R}^{n\times r\times N}$ (for r-*variate* response) be N realizations of the Gaussian Random Field (GRF) observed over n distinct points. X and Y are considered as the training data; so if cross-validation is required, it should be performed before this stage.

Main Variables

blk: A cell array containing the segmentation (blocking) information.
op1: A cell array containing the setting for the STAGE-I problem of the SPS algorithm.
hyper: A cell array containing the setting for the covariance function.
process: A cell array containing the setting for computational capabilities.

How to Run the GSPS Code

In this section, we briefly discuss how to run the GSPS code. First, open the *driverGeneralSPS.m* file. The lines with left arrow sign (←) at the end are those that need to be tuned by the user. Below, we will go over these lines starting from the top:

- *blk.scheme* determines the segmentation scheme. User may chose either *'SS'* (Spatial Segmentation) or *'RS'* (Random Segmentation). Please refer to the paper for more information on these two blocking schemes.
- *uX* variable in the 'SS' segmentation scheme is a $d \times 1$ vector containing the number blocks along each coordinate. A vector of all ones basically impose no segmentation scheme which is used when n is not big.
- *blk.K* determines the number of blocks in the 'RS' segmentation scheme. *blk.K=1* defines only one block.
- *opt1.monitor* allows visual monitoring of the STAGE-I problem. If set to *'on'* shows information on iterations of the ADMM algorithm; otherwise, set it to *'off'*.
- *opt1.tol.primal* is the primal feasibility threshold for the ADMM algorithm. If *both* primal and dual feasibilities (see the next item) go below their thresholds, the algorithm will terminate.
- *opt1.tol.dual* is the dual feasibility threshold for the ADMM algorithm. If *both* primal and dual feasibilities go below their thresholds, the algorithm will terminate.
- *opt1.maxItr* is the maximum number of iterations of the ADMM algorithm and is another stopping criterion.
- *hyper.covFunc* determines the parametric covariance function of interest. User may select from the list below:
  - *SEiso*: *Isotropic* Squared-Exponential covariance function.
  - *PEiso*: *Isotropic* Powered-Exponential covariance function - if selected, then *hyper.p* determines the power where it should be an integer number. Otherwise, set to [ ].
  - *Materniso*: *Isotropic* Matern covariance function - if selected, then *hyper.nu* determines the smoothness parameter which can be equal to 1/2, 3/2, or 5/2. Otherwise, set it to [ ].
  - *Expiso*: *Isotropic* Exponential covariance function.
  - *AnisDiag*: *Anisotropic* covariance function with a diagonal M matrix of the form
  $$c(\mathbf{x}, \mathbf{x}'; M, \sigma^2) = \sigma^2 \exp\Big(-(\mathbf{x}-\mathbf{x}')^\top M (\mathbf{x}-\mathbf{x}')\Big) \qquad (1)$$
  where $M = \text{diag}(\ell)^{-1}$, $\ell \in \mathbb{R}^d$. For more information refer to *Gaussian Processes for Machine Learning, by Rasmussen and Williams (2006), page 106. If selected,* the lower bound (LB) and upper bound (UB) for elements of ℓ should be selected in the following lines.

  - *AnisGeneral*: *Anisotropic* covariance function as (1) with the matrix M constructed as

    $$M = \Delta\Delta^\top + \mathrm{diag}(\ell)^{-1}$$

    where $\Delta \in \mathbb{R}^{d \times g}$ with $g < d$. *If selected*, the lower bound (LB) and upper bound (UB) for elements of ℓ and Δ should be selected in the following lines. Note that a too large UB for Δ elements may result in an invalid covariance matrix.

- *hyper.nugget* If set to *'true'*, then it includes the nugget into the model; otherwise, user may set it to *'false'*.

- *process.type* determines if parallel processing capabilities are available or not. If so, set it to *'parallel'*; otherwise, set it to *'single'*.

- *process.nCores* determines the number of accessible processing cores and is considered only if *process.type* is *'parallel'*.

Outputs

In the *univariate* case, the *spsEstimatorHyper* function begins the GSPS algorithm. It outputs the GSPS estimates of the covariance function parameter available in *hyper.param.val*.

In the *multivariate* case, the *spsEstimatorMultVar* function begins the GSPS algorithm. It outputs the GSPS estimates of the correlation function parameters available in *hyper.param.val*. Furthermore, the between-response covariance matrix estimate $\hat{\Gamma}$ is available in *GammaHat* variable.