Bayesian Estimation of Monotone Single Index Models

Snigdha Das

Overview

The package `monotoneSIM` performs Bayesian estimation using B-Spline basis approximation of a Single Index Model where the unknown link function is assumed to be monotonically increasing. Consider a Single index models of the form:

$$y = g(x^\top \beta) + \epsilon$$

where y: response, x: p-dimensional predictors, β: p-dimensional coefficient vector, ϵ: independent N(0, σ_ϵ^2) random errors, g: unknown monotone link function. To construct a prior on g, we consider the basis expansion: $g(x) = \Sigma_{l=0}^{L} \xi_l \, \psi_l(x)$, where $\psi_l(x) = \int_{-1}^{x} h_l(t) \, \mathrm{d}t$, h_l is a B-Spline basis function of order 2. Then, g is monotonically increasing if and only if $\xi_l \geq 0 \; \forall \; l$, thereby enforcing the monotonicity constraint in an equivalent way.

Note: Here $g(x) = 0$, when $x \leq -1$ and $g(x) = g(1)$, when $x \geq 1$. The matrix of predictors is scaled so that every row has euclidean norm ≤ 1 and euclidean norm of β is taken to be 1 so that $|x^\top \beta| \leq 1$.

Functionality

The package provides two functions.

- The main function `monotoneSIM` performs a Markov Chain Monte Carlo algorithm to generate samples from the conditional posterior distribution of unknown parameters in a monotone Single Index Model - the unknown parameters being B-Spline basis coefficients that approximate the unknown link function, the single index parameter and the error variance of the model.
- The other function `monotoneFIT` calculates fitted values of the response using posterior mean of the single index parameter after generating an MCMC sample using the function `monotoneSIM`. It also returns the estimated link function of the model based on basis coefficients for a grid of supplied values.

Installation

To install this package from Github, run the following in your R console:

```r
devtools::install_github("das-snigdha/monotoneSIM")
```

An Example Demonstrating Usage

This package shall be implemented on the following simulated data.

We take two continuous variables and one dichotomous attribute as predictors.

```r
n = 100; p = 3;
X = matrix(rnorm(n*(p-1)), nrow = n, ncol = (p-1))
X = cbind(X, rbinom(n, 1, 0.5))
```

The following tasks shall be performed during the remaining of the semester:

- Perform Compatibility Checks on user supplied input.
- Select equispaced `knots`, $u_0, u_1, \ldots, u_L$ between -1 and 1, if not supplied.
- Select grid of x values, `grid.x` between -1 and 1 of length `size.grid.x`, if not supplied
- Perform scaling of covariate matrix `X` and supplied `beta.init`.
- Perform back-scaling on obtained `beta`, to get coefficients corresponding to unscaled X.
- Create a vignette to demonstrate the usage, if time permits.