Stationary Point Constrained Regression via Diffeomorphism Example

Contents

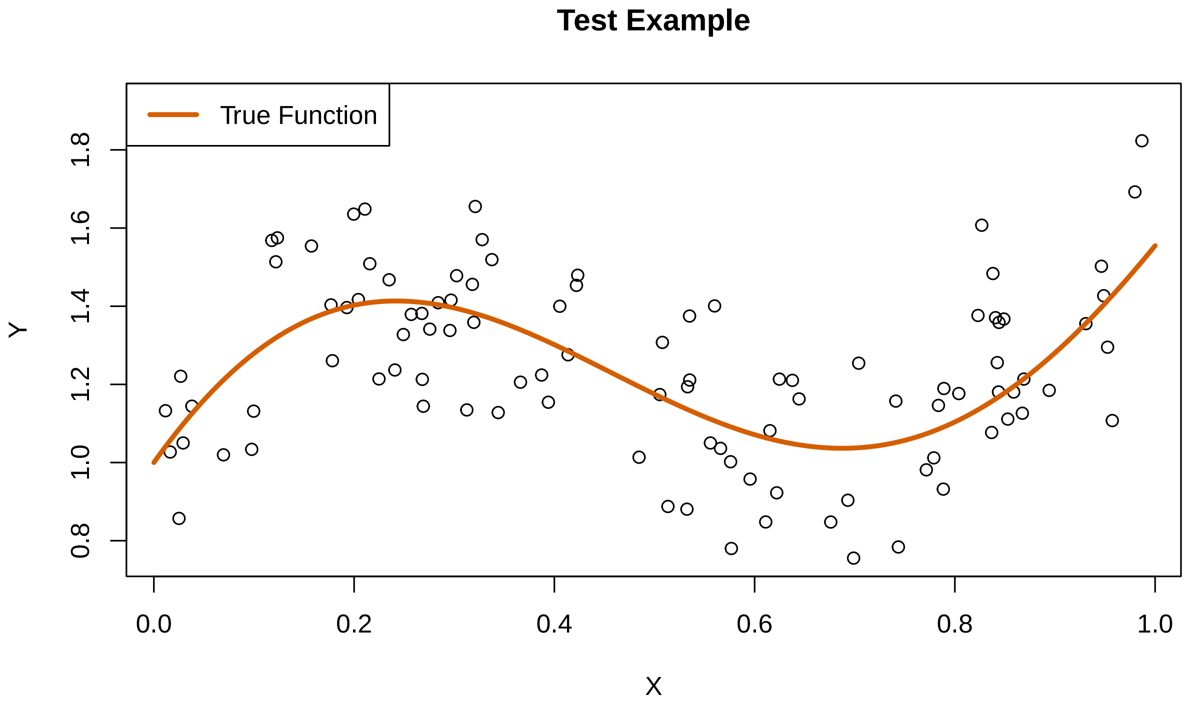
```
library(scrdiff)
```

Here we generate data, where the true function has two stationary points located at 0.2420 and 0.6873.

```
set.seed(49284)
X <- runif(100, 0, 1)
Y <- sin(1.1* 2 *X) + cos(4*X) + 0.7*2*X + rnorm(length(X), sd = 0.15)

X_grid <- seq(0, 1, length.out = 1000)
Y_true <- sin(1.1* 2 *X_grid) + cos(4*X_grid) + 0.7*2*X_grid


plot(X, Y, main = "Test Example", ylim = c(min(Y), max(Y) + 0.1))
lines(X_grid, Y_true, col = "#D55E00", lwd = 3)
legend("topleft", legend = c("True Function"), col = c("#D55E00"), lwd = 3)
```



For our diffeomorphism method, we have the following inputs:

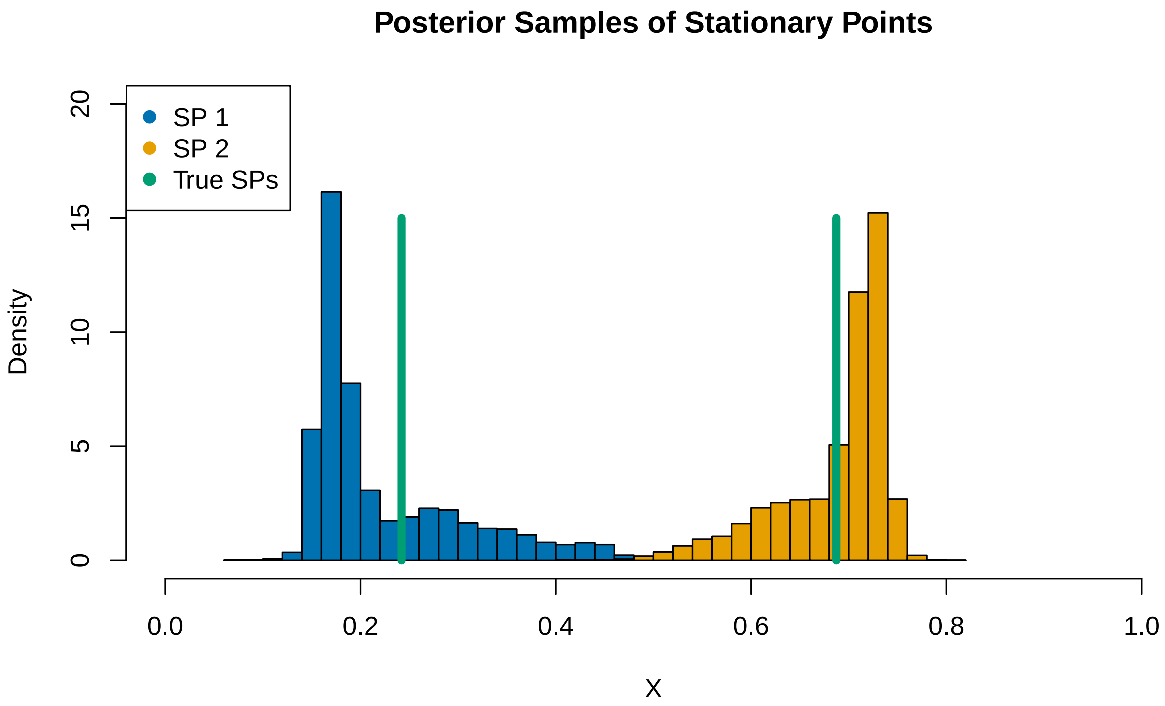
- **X** - length n vector, x points (input data)
- **Y** - length n vector, y points (output data)
- **num_betas** - positive integer > 1, number of weight parameters of the diffeomorphism (generally recommended around 4-15 based on the data size and number of stationary points)
- **num_stationary** - positive integer, number of stationary points
- **first_direction** - 1 or -1, 1 if the function is increasing between min(X) and the first stationary point, -1 if the function is decreasing
- **zero_is_zero** - boolean, TRUE if f(min(X)) = 0, FALSE otherwise and f(min(X)) needs to be estimated
- **interpolation** - 'cubic' or 'linear', referring to interpolation type of lambda height vector
- **b_vec** - NULL or vector of length num_stationary + 2, nodes of template function. If NULL, nodes will be equally spaced from each other
- **prior_mean** - (num_betas + num_stationary + 1 + 1(zero_is_zero) + 1 (2 height vectors at boundaries and for sigma)) dimensional vector of prior mean (Joint Independent Normal)
- **prior_sd** - (num_betas + num_stationary + 3 (2 height vectors at boundaries and for sigma)) dimension vector of prior standard deviations (Joint Independent Normal)
- **n_chains** - positive integer, number of MCMC chains
- **n_per_chain** - positive integer, number of samples per MCMC chain
- **cov_scale** - positive real number, scaling factor for proposal covariance matrices of MCMC chains (should be in between 1/6 and 1, higher dimensions should have a lower cov_scale)
- **apply_sm_correction** - boolean, whether to filter out low probability posterior modes via softmax correction
- **num_cores** - "ALL" or positive integer, number of cores to be used for parallelization, if "ALL", use all cores available

```r
# Parameters
num_betas <- 8
num_stationary <- 2 # Two stationary points
first_direction <- 1 # Function goes "up" first
zero_is_zero <- FALSE # f(min(X)) is not equal to 0
interpolation <- "cubic" # Template is Hermite Cubic Interpolation
b_vec <- NULL # Use default b_vec
n_chains <- 100 # 100 MCMC Chains
n_per_chain <- 2500 # 2500 samples per Chain
cov_scale <- 0.63
apply_sm_correction <- TRUE # Filter out low posterior modes
num_cores <- 2 # 2 for vignette, highly recommend to use ALL available (num_cores <- "ALL")

# Prior Mean and Standard Deviation
prior_mean <- c(rep(0, num_betas), mean(Y), #height at f(min(X))
                rep(0, num_stationary + 1), # cumulative sum parameters
                0)
prior_sd <- c(rep(1, num_betas), 2 * sd(Y),  # height at f(min(X))
              rep(max(Y) - min(Y), num_stationary + 1), # cumulative sum parameters
              0.5)
```

```r
# Apply Method, will take around 4-6 minutes
diff_method_results <- posterior_stat_points_diff_mcmc(X, Y, num_betas, num_stationary,
                                                       first_direction, zero_is_zero,
                                                       interpolation, b_vec, prior_mean,
                                                       prior_sd, n_chains, n_per_chain,
                                                       cov_scale, apply_sm_correction,
                                                       num_cores)
#> [1] "Finding Modes:"
#>   |                                                                              |
```

```
# Histogram of Posterior Stationary Points
hist(diff_method_results$diff_stat_points[[1]], xlim = c(0, 1),
     col = "#0072B2", main = "Posterior Samples of Stationary Points",
     xlab = "X", prob = TRUE, ylim = c(0, 20))
hist(diff_method_results$diff_stat_points[[2]], add = TRUE, breaks = 20,
     col = "#E69F00", prob = TRUE)
segments(0.2420, 0, 0.2420, 15, col = "#009E73", lwd = 5)
segments(0.6873, 0, 0.6873, 15, col = "#009E73", lwd = 5)
legend("topleft", legend = c("SP 1", "SP 2", "True SPs"),
       col = c("#0072B2", "#E69F00", "#009E73"), pch = 19)
```


Posterior Samples of Stationary Points
● SP 1
● SP 2
● True SPs
Density
X