Penalized tensor B-spline neural network regression

Jeonggu Kim

Department of Statistics

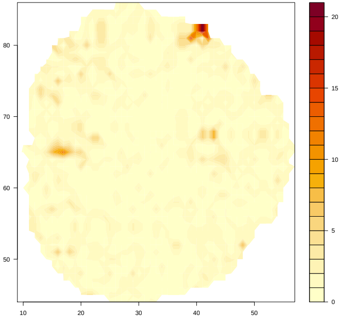
Korea University

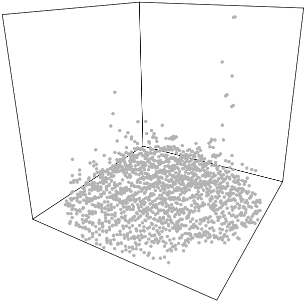
Contents

Introduction

FMRI

- Brain activity captured by local changes in oxygenation pattern
- Spatial smoothing prior to statistical analysis





Motivation

- Traditional non-adaptive regression estimators often fail to capture local trend such as sudden peak.
- We propose an adaptive regression estimator based on neural network with activation function having compact support.
- Application in neurophysiology, quality control, etc.

Model and estimator

Model

- Data : $\{(y_i, x_{1i}, x_{2i})\}_{i=1}^N$
- Model :

$$y_i = f(x_{1i}, x_{2i}) + \varepsilon_i$$

y_i : response

(x_{1i}, x_{2i}) : input

ε_i : random error with mean 0

Tensor B-spline neural network

- Tensor B-spline neural network (TBNN) function :

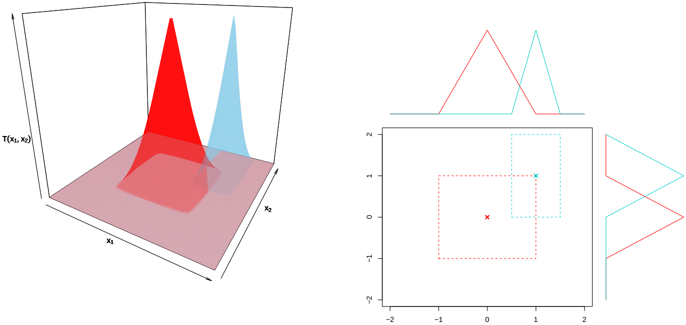
$$\mathsf{f}_\theta(x_1, x_2) = \beta_0 + \sum_{m=1}^{M} \beta_m \sigma(\alpha_{0m}^1 + \alpha_{1m}^1 x_1)\sigma(\alpha_{0m}^2 + \alpha_{1m}^2 x_2)$$

$\sigma(\cdot)$: B-spline activation function

$\theta = (\beta_0, \beta_1, \alpha_{01}^1, \alpha_{11}^1, \alpha_{01}^2, \alpha_{11}^2 \ldots, \alpha_{1M}^2) \in \mathbb{R}^{5M+1}$

- β : weight of tensor B-spline activated node
- $(\alpha_0^1, \alpha_1^1, \alpha_0^2, \alpha_1^2)$: parameter determining the center and support of tensor B-spline activation function

Tensor B-spline activation function



$\blacktriangle$: $(\alpha_0^1, \alpha_1^1, \alpha_0^2, \alpha_1^2) = (0, 1, 0, 1)$

$\blacktriangle$: $(\alpha_0^1, \alpha_1^1, \alpha_0^2, \alpha_1^2) = (-2, 2, -1, 1)$

Penalized tensor B-spline neural network estimator

- Penalized empirical risk with λ :

$$R^{\lambda}(\theta) = \frac{1}{2N}\sum_{i=1}^{N}(y_i - \mathsf{f}_{\theta}(x_{1i}, x_{2i}))^2 + \lambda \mathsf{pen}(\theta)$$

  $\lambda \geq 0$: complexity parameter

- Penalty function :

$$\mathsf{pen}(\theta) = \sum_{m=1}^{M} |\beta_m|$$

- Penalized TBNN estimator :

$$\hat{f} = \mathsf{f}_{\hat{\theta}^{\lambda}}$$

  where $\hat{\theta}^{\lambda} = \operatorname{argmin}_{\theta \in \mathbb{R}^{5M+1}} R^{\lambda}(\theta)$

Implementation

Coordinate descent algorithm

- Objective function : penalized empirical risk
- Coordinate descent algorithm :

$$\tilde{\theta}_j \leftarrow \underset{\theta_j \in \mathbb{R}}{\operatorname{argmin}}\, R^{\lambda}(\tilde{\theta}_1, \ldots, \tilde{\theta}_{j-1}, \theta_j, \tilde{\theta}_{j+1}, \ldots, \tilde{\theta}_J)$$

  where $\sim$ indicates the vector of current values and $J = 5M + 1$

  1. Update $\tilde{\beta}$ with ℓ_1-penalty
  2. Update $\tilde{\alpha}_0^1, \tilde{\alpha}_1^1, \tilde{\alpha}_0^2, \tilde{\alpha}_1^2$

- Initialization scheme specialized in TBNN

Complexity parameter selection

- Complexity parameter λ controls the number of nodes.
- Optimal complexity parameter is selected by

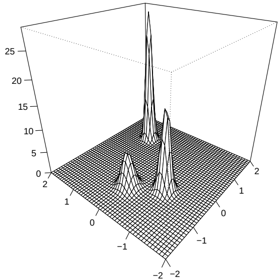
$$\mathsf{BIC} = N \log R(\hat{\theta}^{\lambda}) + \dim \log N$$

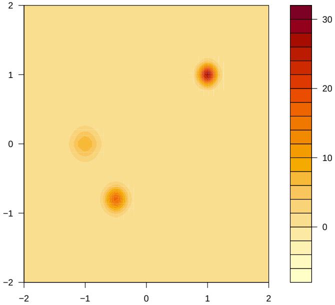
where dim = the number of nodes

Numerical studies

Simulation set-up

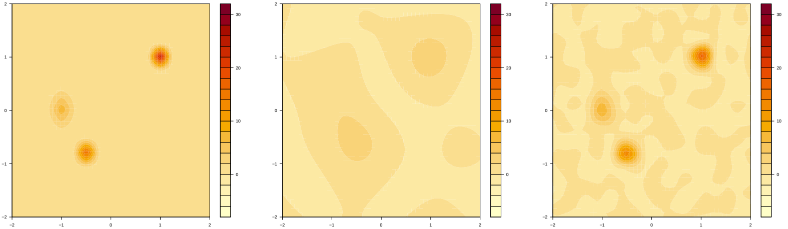
- True regression function with local peak





- Error : $\mathcal{N}(0, 2^2)$
- Competitor :
  1. Smoothing spline
  2. Adaptive P-spline

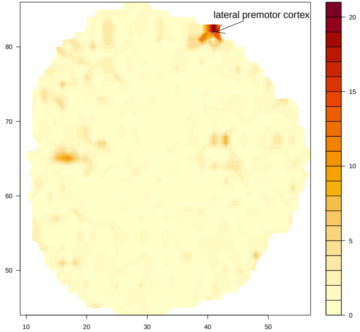
Simulation result



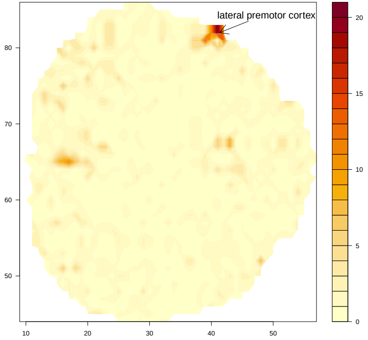
(a) TBNN (b) Smoothing spline (c) Adaptive P-spline

FMRI data

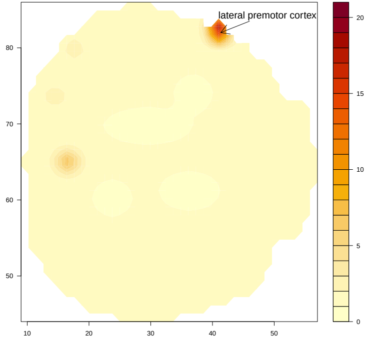
- Input : coordinate of cutting-plane of brain
- Response : FPQ
- 1567 points
- Subject conduct verbal fluency task
- Local peak on lateral premotor cortex



Penalized tensor B-spline neural network fit

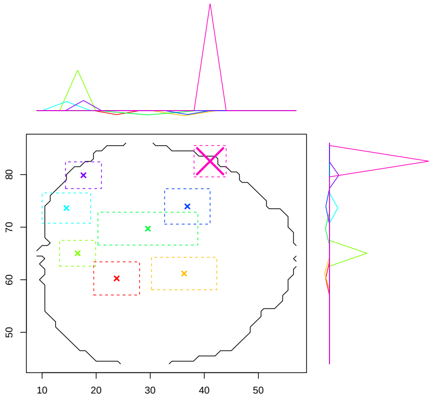


(a) FMRI data



(b) TBNN

Penalized tensor B-spline neural network nodes



(a) total 8 nodes

big marked node : capture the peak

Conclusion

Conclusion

- We develop an adaptive regression estimator based on TBNN.
- The proposed estimator effectively captures local peak with global pattern compared to the existing method.
- The node pruning process is done in the data-adaptive way by ℓ_1-penalization.
- Application to FMRI data exhibits a desired performance.

Appendix

Tensor B-spline activation function

- Tensor B-spline activation function :

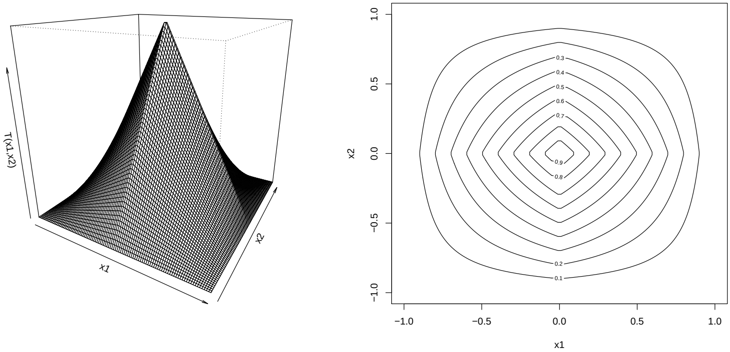
$$\mathsf{T}(x_1, x_2) = \sigma(a^1 + b^1 x_1)\sigma(a^2 + b^2 x_2)$$

where

$$\sigma(z) = \begin{cases} 1 + z & \text{if } \ z \in [-1, 0) \\ 1 - z & \text{if } \ z \in [0, 1) \\ 0 & \text{if } \ \text{otherwise} \end{cases}$$

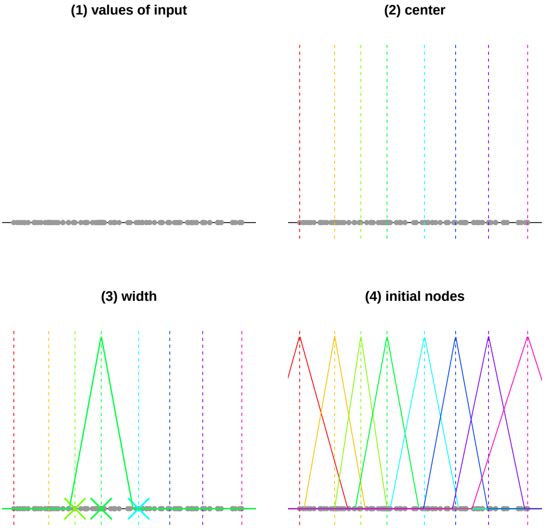
- Center : $\left(-\frac{a^1}{b^1},\ -\frac{a^2}{b^2}\right)$
- Support : $\left[\frac{-(1+a^1)}{b^1},\ \frac{1-a^1}{b^1}\right] \times \left[\frac{-(1+a^2)}{b^2},\ \frac{1-a^2}{b^2}\right]$

Tensor B-spline activation function plot



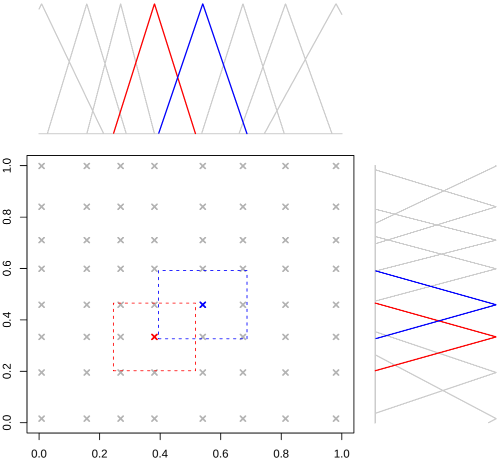
Initialization scheme on marginal line

- Initialization on x_1 axis



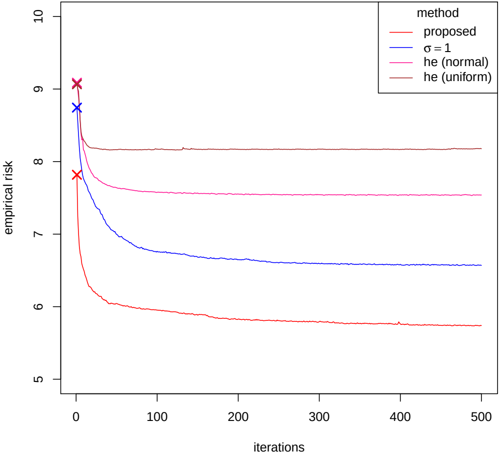
Initialization scheme on plane

- Tensor product of B-spline activation functions on x_1 and x_2 axis



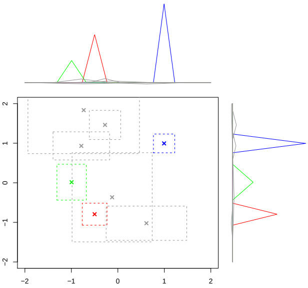
Initialization scheme comparison

- Comparison between the proposed scheme and others

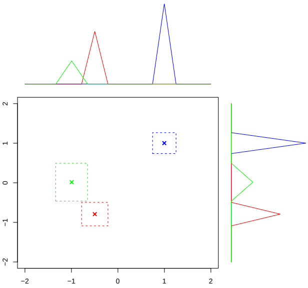


Role of complexity parameter

- Example for $\lambda_1 < \lambda_2$



(a) λ_1 : 8 nodes



(b) λ_2 : 3 nodes

Simulation result table

Error	N	TBNN	Smoothing spline	Adaptive P-spline
$\mathcal{N}(0, 2^2)$	1000	0.3646 (0.0694)	2.3400 (0.0109)	1.1021 (0.0207)
	2000	0.1648 (0.0280)	2.3180 (0.0050)	0.7627 (0.0117)
$\mathcal{N}(0, 4^2)$	1000	0.9442 (0.1167)	2.5940 (0.0347)	2.2457 (0.0705)
	2000	0.4210 (0.0604)	2.4545 (0.0173)	1.5790 (0.0346)

Averaged MSE (standard error) over 100 repetitions