Thin plate splines

한이정

January 12, 2020

1 *The elements of statistical learning* (Friedman et al.)

Multidimensional 스플라인 구현을 위해 가장 많이 참고한 자료

5.7 Multidimensional Splines

- Multi-dimensional spline models
  - Suppose $X \in \mathbb{R}^2$.
  - $h_{lk}(X_l),\ k = 1, ..., M_l$ is a basis functions for representing function of X_l.
  - Then the $M1 \times M2$ dimensional *tensor product basis* defined by

$$g_{jk} = h_{1j}(X_1)h_{2j}(X_2),\ j = 1, .., M_1,\ k = 1, .., M_2. \tag{1}$$

  - (1) can be used for representing a two-dimensional function:

$$g(X) = \sum_j \sum_k \beta_{jk} g_{jk}(X). \tag{2}$$

- Multi-dimensional smoothing splines
  - One-dimensional smoothing splines generalized to higher dimensions as well.
  - Suppose we have pairs y_i, x_i with $x_i \in \mathbb{R}^d$,

$$\min_f \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda J[f], \tag{3}$$

    where J is an appropriate penalty functional for stabilizing a function f in $\mathbb{R}^d$.

- Thin plate spline
  - For example, in $\mathbb{R}^2$

$$J[f] = \iint_{R^2} \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f(x)}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2} \right)^2 \right] dx_1 dx_2. \tag{4}$$

  - Optimizing (3) with this penalty leads to a smooth two-dimensional surface, known as a thin-plate spline.
  - It shares many properties with the one-dimensional cubic smoothing spline:
    * as $\lambda \to 0$, the solution approaches an interpolating function [the one with smallest penalty (4)];
    * as $\lambda \to \infty$, the solution approaches the least squares plane;
    * for intermediate values of λ, the solution can be represented as a linear expansion of basis functions, whose coefficients are obtained by a form of generalized ridge regression;
  - The solution has the form (공부 부족한 부분)

$$f(x) = \beta_0 + \beta^\top x + \sum_{j=1}^{N} \alpha_j h_j(x), \tag{5}$$

    where $h_j(x) = \|x - x_j\|^2 \log\|x - x_j\|$.
  - **These h_j are examples of radial basis functions, which are discussed in more detail in the next section.**
  - The coefficients are found by (5) into (3), which reduces to a finite-dimensional penalized least squares problem.
  - For the penalty to be finite, the coefficients α_j have to satisfy a set of linear constraints; see Exercise 5.14.

    (Ex. 5.14)

    Derive the constraints on the α_j in the thin-plate spline expansion (5) to guarantee that the penalty $J(f)$ is finite. How else could one ensure that the penalty was finite?

- The computational complexity for thin-plate splines is $O(N^3)$, since there is not in general any sparse structure that can be exploited. However, as with univariate smoothing splines, we can get away with substantially less than the N knots prescribed by the solution (5).
- In practice, it is usually sufficient to work with a lattice of knots covering the domain. The penalty is computed for the reduced expansion just as before. Using K knots reduces the computations to $O(NK^2 + K3)$.

Spatial GEV Model Example

- In this study the location parameter is parametrized by

$$\mu_s = \mu_0 + h(\mathbf{x}_s), \quad s = 1, ..., S,$$

- μ_0 is a global location parameter across the considered sites.
- $\mathbf{x}_s = (x_{1s}, x_{2s})^\top$ is the coordinate vector whose elements denote longitude and latitude of the site s.
- $h : \mathbb{R}^2 \mapsto \mathbb{R}^1$ is a spatial map consisting of TPS basis functions.
- Let the vector valued maps consisting of the B-splines basis functions by $\mathbf{B}_l(\cdot) = (B_{l1}(\cdot), \cdots, B_{lK_l}(\cdot))$ for $l = 1, 2$, which represent the features of a site. (식 (2) 참고)
- Then, the TPS basis functions are given by the row-wise Kronecker product of B-splines basis functions denoted by $\mathbf{B}(x_{1s}, x_{2s}) = \mathbf{B}_1(x_{1s}) \circ \mathbf{B}_2(x_{2s}) : \mathbb{R}^2 \mapsto \mathbb{R}^{K_1K_2}$.
- In our case for a given $\mathbf{x}_s = (x_{1s}, x_{2s})^\top$ the TPS basis functions return a $K_1K_2(= K)$ dimensional feature vector denoted by

$$\mathbf{z}_s = (B_{11}(x_{1s})B_{21}(x_{2s}), B_{11}(x_{1s})B_{22}(x_{2s}) \cdots, B_{1K_1}(x_{1s})B_{1K_2}(x_{2s}))^\top.$$

- Then the spatial function h is given by

$$h(\mathbf{x}_s) = \mathbb{B}(x_{1s}, x_{2s})^\top \boldsymbol{\beta} = \mathbf{z}_s^\top \boldsymbol{\beta}, \qquad \boldsymbol{\beta} \in \mathbb{R}^K$$

```
> library(fda)
> load("./kma_data/Pr_46.RData")
> x=unique(Pr_46$long) # 경도
> y=unique(Pr_46$lat)  # 위도
> x_bsobj=create.bspline.basis(range(x),breaks=quantile(x,prob=seq(0,1,length = 3)))
> y_bsobj=create.bspline.basis(range(y),breaks=quantile(y,prob=seq(0,1,length = 3)))
> xbs=eval.basis(x,x_bsobj)
> ybs=eval.basis(y,y_bsobj)
> # row-wise kronecker product (Z matrix)
> tensorbs=do.call('cbind', lapply(1:ncol(xbs), function(i) xbs[,i]*ybs))
```

- The TPS penalty function can be written as quadratic function of $\boldsymbol{\beta}$ (식 (5) 참고):

$$\{\boldsymbol{\Omega}_{\mathbf{B}_l}^{(2)}\}_{jk} = \int B_{lj}^{''}(t)B_{lk}^{''}(t)dt,$$
$$\{\boldsymbol{\Omega}_{\mathbf{B}_l}^{(1)}\}_{jk} = \int B_{lj}^{'}(t)B_{lk}^{'}(t)dt,$$
$$\{\boldsymbol{\Omega}_{\mathbf{B}_l}^{(0)}\}_{jk} = \int B_{lj}(t)B_{lk}(t)dt,$$

  where $\{A\}_{jk}$ denotes the element of A in the ith row and the jth column.

- Then, the TPS penalty function can be written as quadratic function of $\boldsymbol{\beta}$:

$$\begin{aligned} J[h(\cdot)] &= \boldsymbol{\beta}^\top \left(\boldsymbol{\Omega}_{\mathbf{B}_1}^{(2)} \otimes \boldsymbol{\Omega}_{\mathbf{B}_2}^{(0)} + 2\boldsymbol{\Omega}_{\mathbf{B}_1}^{(1)} \otimes \boldsymbol{\Omega}_{\mathbf{B}_2}^{(1)} + \boldsymbol{\Omega}_{\mathbf{B}_1}^{(0)} \otimes \boldsymbol{\Omega}_{\mathbf{B}_2}^{(2)} \right) \boldsymbol{\beta} \\ &\equiv \boldsymbol{\beta}^\top \boldsymbol{\Omega}_{\mathbf{B}} \boldsymbol{\beta}. \end{aligned}$$

  We use the R package *fda* to obtain $\boldsymbol{\Omega}_{\mathbf{B}}$.
  (cf. ?fda::bsplinepen) Each element is the inner product of two B-spline basis functions after applying the derivative or linear differential operator defined by Lfdobj.

```
> Fmat <- kronecker(bsplinepen(x_bsobj,Lfdobj=2),bsplinepen(y_bsobj,Lfdobj=0))
> Gmat <- kronecker(bsplinepen(x_bsobj,Lfdobj=0),bsplinepen(y_bsobj,Lfdobj=2))
> Hmat <- kronecker(bsplinepen(x_bsobj,Lfdobj=1),bsplinepen(y_bsobj,Lfdobj=1))
# Omega_B
> Om <- Fmat+Gmat+2*Hmat
```

- We estimate the spatial function by minimizing the penalized negative log-likelihood,

$$L_\lambda(\mu_0, \boldsymbol{\beta}, \boldsymbol{\theta}) = L(\mu_0, \boldsymbol{\beta}, \boldsymbol{\theta}) + \lambda J[h(\cdot; \beta)]. \tag{6}$$

5.8 Regularization and Reproducing Kernel Hilbert Spaces

- It is also known that a thin-plate splines is an expansion in radial basis functions, generated by the kernel

$$K(x, y) = \|x - y\|^2 \; log(\|x - y\|) \tag{7}$$

- **Radial basis functions are discussed in more detail in Section 6.7.**

- (cf. page 36)
  Radial basis functions are symmetric p-dimensional kernels located at particular centroids,

$$f_\theta(x) = \sum_{m=1}^{M} K_{\lambda_m}(\mu_m, x)\theta_m;$$

  for example, the Gaussian kernel $K_\lambda(\mu, x) = e^{-\|x-\mu\|^2/2}$ is popular.
  Radial basis functions have centroids μ_m and scales λ_m that have to be determined. The spline basis functions have knots. In general we would like the data to dictate them as well. Including these as parameters changes the regression problem from a straightforward linear problem to a combinatorially hard nonlinear problem. In practice, shortcuts such as greedy algorithms or two stage processes are used. Section 6.7 describes some such approaches.

  6.7 Radial Basis Functions and Kernels

2 Wood(2003) Thin plate regression splines

Package 'mgcv' in R (https://cran.r-project.org/web/packages/mgcv/mgcv.pdf)
mgcv::gam에 bs='tp' 또는 'ts'로 tps 적합 가능.
model.matrix 및 smooth.construct 함수로 penalty 함수 구할 수 있음.
그러나 model matrix **X**와 Penalty **S** 형태가 이해가 가지 않아 조사한 자료.

2.1 Introduction

- The first aim of this paper is to find optimal approximations to the thin plate splines which will remove the computational obstacles to their use, while minimizing the deterioration in model performance that is entailed by the approximation (i.e. to find optimal penalized regression splines).

- The second aim is to remove the knot placement problem from regression spline modelling in a way that will allow model selection by the hypothesis testing methods that are usually employed in regression modelling. Two immediate results of achieving these aims are to provide a good way of incorporating smooth function terms into non-linear models and to provide a way of incorporating thin plate spline like terms into generalized additive models (GAMs).

2.2 Low rank thin plate spline like smoothers

- Consider the problem of estimating the smooth function $f(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^d$.

- From $n(\geq d)$ observations $(y_i, \mathbf{x}_i)$ such that

$$y_i = f(\mathbf{x}_i) + \epsilon_i$$

- TPS can be used to estimate f by finding the function g minimizing

$$\|y - g\|^2 + \lambda J_{md}(g) \qquad (5)$$

- The wiggliness penalty is defined as

$$J_{md} = \int \cdots \int_{\mathbb{R}^d} \sum_{v_1 + \ldots v_d = m} \frac{m!}{v_1! \ldots v_d!} \left(\frac{\partial^m g}{\partial x_1^{v_1} \ldots \partial x_d^{v_d}} \right)^2 dx_1 \ldots dx_d. \qquad (6)$$

e.g) d=2, m=2

$$\iint \sum_{v_1 + v_2 = 2} \frac{2!}{v_1! v_2!} \left(\frac{\partial^2 g}{\partial x_1^{v_1} \partial x_2^{v_2}} \right)^2 dx_1 dx_2.$$

$$= \iint \frac{2!}{2!0!} \left(\frac{\partial^2 g}{\partial x_1^2 \partial x_2^0} \right)^2 + \frac{2!}{1!1!} \left(\frac{\partial^2 g}{\partial x_1^1 \partial x_2^1} \right)^2 + \frac{2!}{0!2!} \left(\frac{\partial^2 g}{\partial x_1^0 \partial x_2^2} \right)^2 dx_1 dx_2$$

e.g) d=2, m=1 (cf. https://en.wikipedia.org/wiki/Polyharmonic_spline)

$$\iint \sum_{v_1+v_2=1} \frac{1!}{v_1!v_2!} \left(\frac{\partial g}{\partial x_1^{v_1} \partial x_2^{v_2}} \right)^2 dx_1 dx_2.$$

$$= \iint \frac{1!}{1!0!} \left(\frac{\partial g}{\partial x_1^1 \partial x_2^0} \right)^2 + \frac{1!}{0!1!} \left(\frac{\partial g}{\partial x_1^0 \partial x_2^1} \right)^2 dx_1 dx_2$$

- Provided that we impose the technical restriction $2m > d$, it can be shown that the function minimizing expression (5) has the form

$$g(\mathbf{x}) = \sum_{i=1}^{n} \delta_i \eta_{md}(\|x - x_i\|) + \sum_{j=1}^{M} \alpha_j \phi_j(\mathbf{x}) \quad (7)$$

  where $\boldsymbol{\delta}$ and $\boldsymbol{\alpha}$ are unknown parameter vectors subject to the constraint that $\mathbf{T}'\boldsymbol{\delta} = 0$ and $T_{ij} = \phi_j(\mathbf{x}_i)$.

- The $M = \binom{m+d-1}{d}$ functions ϕ are linearly independent polynomials spanning the space of polynomials in $\mathbf{R}^d$ of degree less than m (i.e. the space of polynomials for which J_{md} is 0). Furthermore

$$\eta_{md}(r) = \begin{cases} ? & d \text{ even,} \\ ? & d \text{ odd.} \end{cases}$$

- Now, defining matrix $\mathbf{E}$ by $E_{ij} \equiv \nu_{md}(\|\mathbf{x}_i - \mathbf{x}_j\|)$.

- The spline fitting problem becomes

$$\min \|\mathbf{y} - \mathbf{E}\boldsymbol{\delta} - \mathbf{T}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\delta}'\mathbf{E}\boldsymbol{\delta} \quad \text{subject to } \mathbf{T}'\boldsymbol{\delta} = 0 \qquad (8)$$

2.3 Appendix A : Implementation by using standard software

Here are the steps required to construct a rank k basis

1. Form the $n \times n$ matrix $\mathbf{E}$ and the $n \times M$ matrix $\mathbf{T}$ defined in Section2.

2. Obtain the truncated spectral decomposition $\mathbf{E_k} = \mathbf{U}_k \mathbf{D}_k \mathbf{U'}_k$, by the use of any standard eigen-routines to finc the full spectral decomposition of $\mathbf{E}$.

3. Using standard routines, form the QR-decomposition $\mathbf{QR} = \mathbf{U_k'T}$ where the last $n - M$ rows of $\mathbf{R}$ are 0 and $\mathbf{Q}$ is orthonormal. Then the final $n - M$ colmuns of $\mathbf{Q}$ give $\mathbf{Z}_k$, the basis for the null space of the equality constraints. If efficiency matters then $\mathbf{Z}_k$ can be stored as M Householder rotations.

4. Writing the parameter $k-$vector of the thin plate regression spline as $\boldsymbol{\beta} = (\tilde{\boldsymbol{\delta}}', \boldsymbol{\alpha}')'$, then the $n \times k$ design matrix for the thin plate regression spline is $\mathbf{X} = (\mathbf{U_k D_k Z_k}, \mathbf{T})$. Similarly the penalty matrix woiuld be

$$\mathbf{S} = \begin{pmatrix} \mathbf{Z_k' D_k Z_k} & 0 \\ 0 & 0 \end{pmatrix}$$

where the padding with zero matrices is for notational convenience.

5. To fit penalized tprs

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \boldsymbol{\beta}' \mathbf{S} \boldsymbol{\beta}$$

```
> library(mgcv)
> load("~/GITHUB/gev/kma_data/Pr_46.RData")
> # stnlds로 dataframe 쪼개서 list에 분배
> ss = split.data.frame(Pr_46,Pr_46$stnlds)
> # 강수량(pr) 변수로만 이루어진 list 생성
> xlist = lapply(ss,"[[","pr")
> x=unique(Pr_46$long)
> y=unique(Pr_46$lat)
> pr_tmp=unlist(lapply(xlist,function(x) x[1]))
> knots=  data.frame(x=quantile(x,prob = seq(0, 1, length = 3)),
+                      y=quantile(y,prob = seq(0, 1, length = 3)))
> gam_object = s(x,y,bs="tp",m=2) # m=2 for normal cubic spline penalty
> gam_fit = gam(pr_tmp~s(x,y,bs="tp",m=2))
> gam_smooth = gam_fit$smooth[[1]]
> gam_smooth$bs.dim          # intercept + 29
[1] 30
> dim(model.matrix(gam_fit)) # X matrix
```

```
[1] 56 30
> dim(gam_smooth$S[[1]])     # penalty
[1] 29 29
```

3 Sangalli et al.(2013)Spatial spline regression models

TPS와 penalty가 다름.

Ramsay (2002), Wood et al. (2008)에 covariates 추가함.

References

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Ramsay, T. (2002). Spline smoothing over difficult regions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2):307–319.

Sangalli, L. M., Ramsay, J. O., and Ramsay, T. O. (2013). Spatial spline regression models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(4):681–703.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114.

Wood, S. N., Bravington, M. V., and Hedley, S. L. (2008). Soap film smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):931–955.