Review of A Simple Spline-based Multilevel Modeling Approach

Taeyoung Choi

Statistic

December 9, 2019

Overview

Introduce

- This paper introduce a spline-based multilevel approach for effectively analyzing complex data collected from multiple sourse.
- Developed method is estimation procedure combining Expectation-Maximization algorithm and nonparametric regression approach

Keyword

- Expectaion-Maximization(EM)
- Hurdle Model
- Nonparametric Regression
- Latent Variable
- Large-scaled Data
- Linear Smoother
- Smoothing Spline

Using data collected from various sourse

However, pose a few challenges:

- Data structure is complex;
- Traditional assumptions on the sampling process are difficult to justify;
- The volume of data is larger than the data collected by traditional method;

Solution from challenges

In this work:

- We propose a statistical framework to analyze the data merged from various sourse;
- The proposed model uses a **multilevel model** under nonparametric functional structural model assumption;

Our aim

It is to develop a statistical procedure for nonparametric modeling approach on the underlying regression function under **the multilevel model structure**, as well as deriving the theoretical properties of such procedure.

Parameter and observation

The location and date of each event of a group of musicians are observed over the entire USA for a given time period.

$$\text{number of event } i = 1, \ldots, n$$

$$\text{month } j = 1, \ldots, J$$

There is number of concert events in the j-th month:

$$\mathbf{y_j} = (y_{1j}, \ldots, y_{nj})^T$$

with expactation vector :

$$\begin{aligned}\eta_j &= (\eta(X_1; \beta_j), \ldots, \eta(X_n; \beta_j))^T \\ &= (\eta_{1j}, \ldots, \eta_{nj})^T\end{aligned}$$

η is known link function.

Parameter and observation

The β is unknown time-varying parameter to be estimate

$$\beta_j = (\beta_{j1}, \ldots, \beta_{jp})^T$$

The $\mathbf{X}$ is CBSA specific regional covariates.

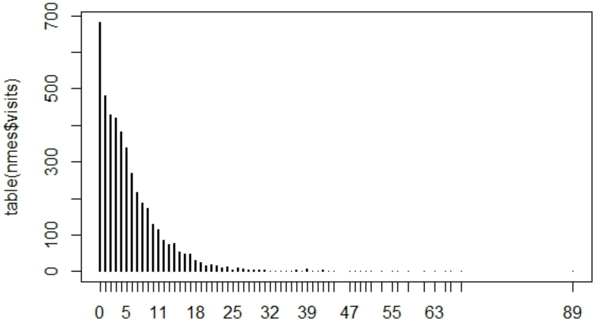
$$\mathbf{X}_i = (x_{j1}, \ldots, x_{jp})^T$$

Hurdle Model

Assume :

$$y_{ij} \sim f(y_{ij}; \mathbf{X}_i, \beta_j) \tag{1}$$

where f is a parametric model(Hurdle model)



Assume

The time varying parameter β_j's are assume :

$$\beta_{jk} = \mu_k(z_j) + \epsilon_{jk},\ k = 1, \ldots, p$$

To simplify, let $p = 1$

$$\beta_j = \mu(z_j) + \epsilon_j \tag{2}$$

where months are re-scaled to $z_1, \ldots, z_J \in [0, 1]$, $\mu(z_j)$ is mean function of β_j that belong to a class of smooth function with respect to z_j;
ϵ_j is random error;

$$\epsilon_j \sim N(0, \sigma^2),\ \sigma > 0$$

Main interest and challenge

Interest

- To investigate how the relationship between y_{ij} and $\mathbf{X}_i$ changes over time;
- This can be achived by finding the 2nd level model which capture the temporal change of β_j;

Challenge

- There is no direct measurement of β_j;
- The smoothing splines approach and EM algorithm;

Estimation

(2) is consider the smoothing spline;

$$\sum_{j=1}^{J}(\beta_j - \mu(z_j))^2 + \lambda \int_0^1 (\mu^{''})^2 \tag{3}$$

$\mu \in \mathcal{W}^{(2,2)}$, where $\quad \mathcal{W}^{(2,2)} = \{m \in L_2 : \int (m^{''})^2\}$
Sobolev class of twice differentiable function and λ is panalty constant.

Estimation(2)

(3) rewritten as

$$\min_{\boldsymbol{\theta}\in\mathbf{R}^J} g(\theta;\beta) = (\boldsymbol{\beta} - N\boldsymbol{\theta})^T(\boldsymbol{\beta} - N\boldsymbol{\theta}) + \lambda\boldsymbol{\theta}^T W\boldsymbol{\theta} \tag{4}$$

where

$$\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)$$

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_J)$$

$$N^T_{J\times J} = (N_{ij}),\ N_{ij} = N_i(z_j)$$

$$W = (w_{ij}), w_{ij} = \int N_i^{''} N_j^{''}$$

with natural cubic spline N_j having $z_1, \ldots, z_J$ Knots

Information related to β_j

We can be obtained from the observation $(\mathbf{y_j}, \mathbf{X})$ from **Two-step**
1st-step

- Obtain $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_J)^T$ by individually fitting the model in $f(y_{ij}; x_j, \beta_j)$ for each j, $j = 1, \ldots, J$, since β is not observable.
- It is treated as a latent variable, where $\hat{\beta}$is treated as an observation for $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^T$
- Once all $\hat{\beta}_1, \ldots, \hat{\beta}_J$ are obtained, define $h(\hat{\beta}_j|\beta_j)$, where the probability density of the sampling distribution of $\hat{\beta}_j$
- Maximum likelihood is used to obtain $\hat{\beta}_j$
- $h(\hat{\beta}_j|\beta_j)$ can assume $AN(\beta_j, V_j)$, where V_j is inverse of Fisher information matrix.

Information related to $\beta_j(2)$

2st-step

We estimate θ using EM, by minimizing (3) with a consideration of the $\hat{\boldsymbol{\beta}}$

- Obtain $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_J)^T$ by $h(\hat{\beta}_j|\beta_j)$
- Specifically, the smoothing spline criterian in (4) can be viewed as the negative complete loglikelihood with

$$(\beta|\theta, \sigma^2) \sim N(N\theta, \Sigma)$$

$$(\theta|\sigma^2) \sim N(0, \sigma^2\lambda^{-1}W^{-1})$$ Is it related to Fisher information??

where $\Sigma = \sigma^2 I_J$

$$\max_{\boldsymbol{\theta}\in\mathbf{R}^{\mathbf{J}}} \int h(\hat{\beta}_j|\beta_j)\exp\left(-\frac{g(\theta;\beta)}{2}\right) d\beta \tag{5}$$

Proof

$$\mathsf{E}_\beta(\text{likelihood of } \theta|\hat\beta;\hat\theta,\sigma) = \int \exp\left(-\frac{g(\theta;\beta)}{2}\right)\cdot h(\beta|\hat\beta;\hat\theta,\sigma))d\beta$$

$$= \int \exp\left(-\frac{g(\theta;\beta)}{2}\right)\cdot\left(\frac{h(\hat\beta,\beta|\hat\theta^{(m)},\sigma)}{\int h(\hat\beta,\beta|\hat\theta^{(m)},\sigma)d\beta}\right)d\beta$$

$$= \frac{1}{\int h(\hat\beta,\beta|\hat\theta^{(m)},\sigma)d\beta}\int \exp\left(-\frac{g(\theta;\beta)}{2}\right)\cdot h(\hat\beta,\beta|\hat\theta^{(m)},\sigma)$$

$$= \frac{1}{\int h(\hat\beta|\beta)h(\hat\beta,\beta|\hat\theta^{(m)},\sigma)d\beta}\int \exp\left(-\frac{g(\theta;\beta)}{2}\right)\cdot h(\hat\beta|\beta)\cdot h(\hat\beta,\beta|\hat\theta^{(m)},\sigma)$$

$$= \frac{1}{\mathsf{E}\left(h(\beta\hat|\beta)\right)}\cdot\mathsf{E}\left(\exp\left(-\frac{g(\theta;\beta)}{2}\right)\cdot h(\hat\beta|\beta)\right)$$

$$= \max\mathsf{E}\left(h(\hat\beta|\beta)\cdot\exp\left(-\frac{g(\theta;\beta)}{2}\right)\right)$$

Using Estimation

$$\hat{\boldsymbol{\theta}} = \text{argmin E}[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^{\mathsf{T}}\mathbf{V}^{-\mathbf{1}}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{N\theta})^{\mathsf{T}}(\boldsymbol{\beta} - \boldsymbol{N\theta})$$
$$+\boldsymbol{\lambda}\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{\theta}|\hat{\beta}; \hat{\boldsymbol{\theta}}^{\boldsymbol{(m)}}, \sigma] \qquad (6)$$

It is straightforward to derive. where

$$(\beta|\hat{\beta}; \theta, \sigma) \sim N\left((\mathbf{V}^{-1} + \Sigma^{-1})^{-1}(\mathbf{V}^{-1}\hat{\boldsymbol{\beta}} + \Sigma^{-1}\boldsymbol{N\theta}),\ (\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\right)$$
$$= N(\boldsymbol{\beta}^{\boldsymbol{*}},\ \mathbf{V}^{*})$$

Using Estimation(2)

$$\begin{aligned}\frac{\partial}{\partial\theta}(6) &= \mathsf{E}[\frac{\partial}{\partial\theta}(\boldsymbol{\beta}-\boldsymbol{N\theta})^{\mathsf{T}}(\boldsymbol{\beta}-\boldsymbol{N\theta})+\lambda\frac{\partial}{\partial\theta}\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{W\theta}|\hat{\boldsymbol{\beta}};\hat{\boldsymbol{\theta}}^{\boldsymbol{(m)}},\sigma] \\ &= \mathsf{E}[-2\boldsymbol{N^T}\beta+2\boldsymbol{N^T}\boldsymbol{N}\hat{\boldsymbol{\theta}}+2\lambda\boldsymbol{W}\hat{\boldsymbol{\theta}}|\hat{\beta};\hat{\boldsymbol{\theta}}^{\boldsymbol{(m)}},\sigma] \overset{Let}{=} 0\end{aligned}$$

From upper equation,

$$\begin{aligned}(\boldsymbol{N}^T\boldsymbol{N}+\lambda\boldsymbol{W})\cdot\hat{\boldsymbol{\theta}}^{(m+1)} &= \boldsymbol{N}^T\boldsymbol{N}\hat{\boldsymbol{\theta}}^{(m+1)}+\lambda\boldsymbol{W}\hat{\boldsymbol{\theta}}^{(m+1)} \\ &= \boldsymbol{N}^T\mathsf{E}[\beta|\hat{\beta};\hat{\theta}^{(m)},\sigma] = \boldsymbol{N}^T\boldsymbol{\beta^*} \\ &= \boldsymbol{N}^T\cdot(\mathbf{V}^{-1}+\Sigma^{-1})^{-1}(\mathbf{V}^{-1}\hat{\boldsymbol{\beta}}+\Sigma^{-1}\boldsymbol{N\theta}) \qquad (7)\end{aligned}$$

Using Estimation(3)

$$\therefore\ \hat{\boldsymbol{\theta}}^{(m+1)} = (\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{W})^{-1}\boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}(\mathbf{V}^{-1}\hat{\boldsymbol{\beta}} + \Sigma^{-1}\boldsymbol{N}\hat{\boldsymbol{\theta}}^{(m)})$$

If $m \to \infty$, then $\hat{\theta}$

$$\begin{aligned}\hat{\boldsymbol{\theta}} = {} & (\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{W})^{-1}\boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\mathbf{V}^{-1}\hat{\boldsymbol{\beta}} \\ & + (\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{W})^{-1}\boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\Sigma^{-1}\boldsymbol{N}\hat{\boldsymbol{\theta}}\end{aligned}$$

$$\begin{aligned}(\mathbf{I} - (\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{W})^{-1}\boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\Sigma^{-1}\boldsymbol{N})\hat{\boldsymbol{\theta}} \\ = (\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{W})^{-1}\boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\mathbf{V}^{-1}\hat{\boldsymbol{\beta}}\end{aligned}$$

$$((\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{W}) - \boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\Sigma^{-1}\boldsymbol{N})\hat{\boldsymbol{\theta}} = \boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\mathbf{V}^{-1}\hat{\boldsymbol{\beta}}$$

Using Estimation(4)

$$
\begin{aligned}
\hat{\boldsymbol{\theta}} &= ((\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{W}) - \boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\Sigma^{-1}\boldsymbol{N})^{-1} \times \\
&\quad \boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\mathbf{V}^{-1}\hat{\boldsymbol{\beta}} \\
&= [\boldsymbol{N}^T(\mathbf{I} - \mathbf{V}^{-1} + \Sigma^{-1})^{-1}\Sigma^{-1})\boldsymbol{N} + \lambda\boldsymbol{W}]^{-1}\boldsymbol{N}^T(\mathbf{V}^{-1} + \Sigma^{-1})^{-1}\mathbf{V}^{-1}\hat{\boldsymbol{\beta}} \\
(\because &\boldsymbol{D} = \mathbf{I} - \boldsymbol{V}^*\Sigma^{-1} = \boldsymbol{V}^*\boldsymbol{V}^{-1}) \\
&= [\boldsymbol{N}^T\boldsymbol{D}\boldsymbol{N} + \lambda\boldsymbol{W}]^{-1}\boldsymbol{N}^T\boldsymbol{D}\hat{\boldsymbol{\beta}} \qquad (8)
\end{aligned}
$$

Using Estimation(5)

$$\hat{\mu}(z) = \sum_{j=1}^{J} N_j(z)\hat{\theta}_j = \sum_{j=1}^{J} l_{\lambda,j}(z)\hat{\beta}_j \tag{9}$$

$\hat{\mu}(\cdot)$ is a linear smoother for $\hat{\boldsymbol{\beta}}$.

$$\boldsymbol{l}_\lambda(z) = (l_{\lambda,1}(z), \cdots, l_{\lambda,J}(z))$$

Thus, the fitted response is

$$\hat{\boldsymbol{\mu}} \equiv (\hat{\mu}(z_1), \cdots, \hat{\mu}(z_J))^T = S_\lambda \hat{\beta}$$

with smoothing matrix $S_\lambda = (\boldsymbol{l}_\lambda(z_1), \cdots, \boldsymbol{l}_\lambda(z_J))$

Searching λ by GCV

This paper calculate Genealized Cross Validation(GCV).

$$\mathrm{GCV}(\lambda) = J^{-1} \sum_{j=1}^{J} \left(\frac{\hat{\beta}_j - \hat{\mu}(z_j)}{1 - \mathrm{S}_\lambda / J} \right)^2 \tag{10}$$

effective degree of freedom is $\mathrm{df}_\lambda = \mathrm{tr}(\mathrm{S}_\lambda)$ for linear smoother.
where $h(\hat{\beta}_j|\beta_j) \sim AN(\beta_j,\ V_j)$ or $\mathrm{E}[\beta|\hat{\beta};\hat{\theta}^{(m)},\sigma]$
$\beta_1^{(m)}, \cdots, \beta_B^{(m)}$ generated from dist

$$(\beta|\hat{\beta};\theta,\sigma) \sim N(\boldsymbol{\beta}^*,\ \mathbf{V}^*)$$

can be estimated by

$$\frac{\sum_{b=1}^{B} w_b \beta_b^{(m)}}{\sum_{b=1}^{B} w_b}$$

where w_b is weight for the $\beta_b^{(m)}$, $w_b = h(\hat{\beta}_j|\beta_j)$, straightforward choice.

Expactation

$$
\begin{aligned}
\mathsf{E}[\beta|\hat{\beta};\hat{\theta}^{(m)},\sigma] &= \int \beta h(\beta|\hat{\beta};\hat{\theta}^{(m)},\sigma)d\beta \\
&= \int \beta \frac{h(\hat{\beta},\beta|\hat{\theta}^{(m)},\sigma)}{\int h(\hat{\beta},\beta|\hat{\theta}^{(m)},\sigma)d\beta} \\
&= \frac{\int \beta h(\hat{\beta}|\beta)h(\beta|\hat{\theta}^{(m)},\sigma)d\beta}{\int h(\hat{\beta},\beta|\hat{\theta}^{(m)},\sigma)d\beta} \\
&= \frac{\mathsf{E}_{\beta|\hat{\theta}^{(m)},\sigma}[\beta h(\hat{\beta}|\beta)]}{\mathsf{E}_{\beta|\hat{\theta}^{(m)},\sigma}[h(\hat{\beta}|\beta)]}
\end{aligned}
\tag{11}
$$

Thus, once a form of $h(\hat{\beta}|\beta)$ is given, generate MC sample $\beta_b \sim (\beta_b;\hat{\theta}^{(m)},\sigma)$

$$
\mathsf{E}[\beta|\hat{\beta};\hat{\theta}^{(m)},\sigma] \approx \frac{\sum_{b=1}^{B}\beta h(\hat{\beta}|\beta_b)}{\sum_{b=1}^{B}h(\hat{\beta}|\beta_b)}.
$$
(by WLLN, B is larger and larger)

Simulation

Naive: A two-step approach that is simpler than ours.

- Fit data model (1) for each partion in order to obtain $\hat{\beta}_j$ for $j = 1, \cdots, J$ as observations to fit nonparametric regression (2) by minimizing (3) with $\boldsymbol{\beta}$ substituted by $\hat{\boldsymbol{\beta}}$.

Simulation methodology:

- For the simulation, we assume an interval is divided into equally spaced J partitions.
- Let $\mu(z_1), \cdots, \mu(z_J)$ be the true structure model values at the center point J partition.
- We estimeate $\hat{\mu}(z_1), \cdots, \hat{\mu}(z_J)$ and calcuate the Root Mean Squared Error **(RMSE)** of $\hat{\mu}_j$, $\left[J^{-1} \sum_{j=1}^{J} (\mu(z_j) - \hat{\mu}(z_j))^2\right]^{1/2}$ for Naive and Multilevel method.

Simulation Example

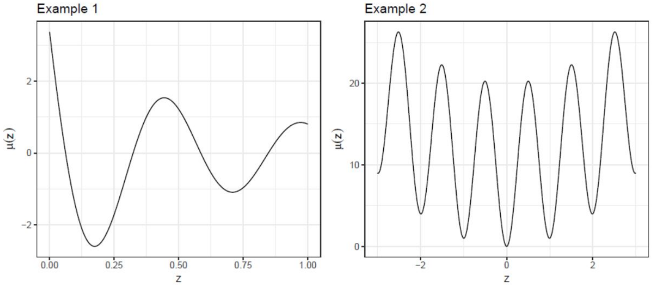


Figure 1: Two test functions used in the simulation study.

The true model of Example1;

$$\mu(z) = \frac{sin(12(z+0.2))}{z+0.2}$$

The true model of Example2;

$$\mu(z) = 10 + z^2 - 10cos(2\pi z)$$

Simulation Assume

When our model is LM;

- $y_{jk} \sim N(X_k\beta_j, \tau^2)$ with $\tau = 2, 4, 8$, for $k = 1, \cdots, n_j$
- $\beta_j \sim$N$(\mu(z_j), \sigma^2)$ with $\sigma = 0.5$

When our model is GLM;

- $y_{jk} \sim$ Bern(p_{jk}),,where $p_{jk} = (1 + \exp(-\eta_{jk}))^{-1}$ and $\eta_{jk} \sim N(X_k\beta_j, \tau^2)$ with $\tau = 2, 4, 8$, for $k = 1, \cdots, n_j$
- $\beta_j \sim$N$(\mu(z_j), \sigma^2)$ with $\sigma = 0.5$

The design matrix X_k is generated from N(0,1).
The sample size for each partition $n_j$is chosen between N_{min} and 200 with the equal probability, where $N_{min} = 50$.

Simulation example1

Model	τ	N_{min}	Naive	Naive λ	Multilevel	Multilevel λ
LM	2	50	0.293	0.000106	0.284	0.000094
	4	50	0.274	0.00026	0.267	0.000215
	8	50	0.393	0.00015	0.406	0.00006
GLM	2	50	0.526	0.000168	0.55	0.0001
	4	50	0.863	0.000187	0.878	0.000096
	8	50	1.07	0.00024	1.08	0.00011

Simulation example2

Model	τ	N_{min}	Naive	Naive λ	Multilevel	Multilevel λ
LM	2	50	0.455	0.000035	*0.428*	0.000036
	4	50	0.481	0.00005	0.469	0.000034
	8	50	0.667	0.000115	0.662	0.0000325

Issue example2(2)

Setting : Model GLM, $\tau = 2$, $N_{min} = 50$;
We can get $RMSE$s multilevel approach under GLM.;
$RMSE_1 = 3.21(\lambda = 0.00023)$
$RMSE_2 = 3.05(\lambda = 0.000395)$
$RMSE_3 = 3.43(\lambda = 0.00062)$
$RMSE_4 = 2.98(\lambda = 0.000508)$
$RMSE_5 = 3.04(\lambda = 0.000197)$
We can get $RMSE$s naive approach under GLM.
$RMSE_1 = 12.9(\lambda = 46.4)$
$RMSE_2 = 16.3(\lambda = 54.3)$
$RMSE_3 = 76.8(\lambda = 93.465)$
$RMSE_4 = 7.2(\lambda = 11.2)$
$RMSE_5 = 22.6(\lambda = 394.54)$

Future Study

- I need to find out why the simulation results are different from paper.
- This simulation needs to the 200 replicates for RMSE.
- The natural spline will change to The penalized B-spline.
- In paper's code, I will change to penalized B-spline's code, too.
- Check that there are no deficiencies in the theoretical part.

$$C(x) = \sum_{i=1}^{I} a_i B_{i,d}(x), \quad \text{where } I = K+d+1,$$

K is number of knots

d is degree of piecewise poly

a_i is control point.

No penalty

B-spline power basis $B_{i,d}(x)$

$$1 \quad x \quad \cdots \quad x^d, \quad (x-\varepsilon_1)^d, \quad \cdots, \quad (x-\varepsilon_K)^d$$

$$\hat{a}_i = \underset{a_i}{\arg\min} \sum_{j=1}^{J} \{ y_j - \sum_{i=1}^{I} a_i B_{i,d}(x_j) \}^2 = \arg\min \sum_{j=1}^{J} \{ y_j - C(x_j) \}^2$$

$$\hat{C}(z) = \sum_{i=1}^{I} \hat{a}_i B_{i,d}(z)$$

$y_j = \mu(z_j) + \varepsilon_j$, z_j is sparse time.

$$C(x) = \sum_{i=1}^{I} a_i B_{i,d}(x) = a_1 + a_2 x + a_3 x^2 + \cdots + a_{2+d}(x-\varepsilon_1)^d + \cdots + a_{K+d+1}(x-\varepsilon_K)^d$$

$$\mu(z) \approx \sum_{j=1}^{I} a_j B_j(z)$$

$\underset{n\times 1}{y} = (y_1, \cdots, y_n)^T$, $n \times I$ matrix $\mathbb{B}$, $\alpha = (a_1 \ \cdots \ a_I)^T$

$$SSE(\alpha^*) = (y - \mathbb{B}\alpha)^T (y - \mathbb{B}\alpha)$$

$$\hat{\alpha}^* = (\mathbb{B}^T \mathbb{B})^{-1} \mathbb{B}^T y$$

$$\hat{\mu}(z) = B(z)\hat{\alpha}^* = B(z)(\mathbb{B}^T\mathbb{B})^{-1}\mathbb{B}^T y$$

P-spline $= \sum (y_i - \mu(z_i))^2 + \lambda \int [D^2 \mu(z)]^2 dz$

$$\int (D^2\mu(z))^2 dz = \int \alpha^T [D^2 B(z)][D^2 B(z)]^T \alpha \, dz = \alpha^T R_2 \alpha$$

$[R_2]_{jk} = \int [D^2 B(z)][D^2 B(z)] dz$ is penalty matrix.

$$\hat{\alpha} = [\mathbb{B}^T\mathbb{B} + \lambda R_2]^{-1} \mathbb{B}^T y$$

$$\hat{y} = \mathbb{B}[\mathbb{B}^T\mathbb{B} + \lambda R_2]^{-1}\mathbb{B}^T y = \ell(\lambda) y \longrightarrow \text{Linear Smoother.}$$

$$\hat{a}_i = \underset{a_i}{\arg\min} \sum_{j=1}^{n} \left\{ y_j - \sum_{i=1}^{I} a_i B_{i,d}(x_j) \right\}^2 + \lambda \int_{x_{min}}^{x_{max}} \left\{ \sum_{i=1}^{I} a_i B''_{i,d}(x) \right\}^2 dx$$

EM Algorithm

$h(\hat{\beta} | \beta) \sim AN(\beta_j, V_j) \longrightarrow V_j$ is fisher information.

$(\beta | \alpha, \sigma) \sim N(N\alpha, \Sigma)$

$(\alpha | \sigma^2) \sim N(0, \sigma^2 \lambda^{-1} W^{-1})$

$$(\boldsymbol{\beta} \mid \hat{\boldsymbol{\beta}}; \boldsymbol{\theta}, \sigma) \sim N(\boldsymbol{\beta}^*, \boldsymbol{V}^*),$$

$$\boldsymbol{\beta}^* = (\boldsymbol{V}^{-1} + \boldsymbol{\Sigma}^{-1})^{-1} \left(\boldsymbol{V}^{-1}\hat{\boldsymbol{\beta}} + \boldsymbol{\Sigma}^{-1}\boldsymbol{N}\boldsymbol{\theta} \right) \text{ and } \boldsymbol{V}^* = (\boldsymbol{V}^{-1} + \boldsymbol{\Sigma}^{-1})^{-1}$$

$$\underset{n \times I}{B_{i,d}(x)} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^d & (x_1 - \varepsilon_1)^d & \cdots & (x_1 - \varepsilon_K)^d \\ \vdots & \vdots & \vdots & & & \ddots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^d & (x_n - \varepsilon_1)^d & \cdots & (x_n - \varepsilon_K)^d \end{pmatrix}$$

$\hat{\alpha}$ is coefficient estimators matrix of penalized B-spline B

$$\hat{\alpha}^{(m+1)} = (N^T N + \lambda W)^{-1} N^T (V^{-1} + \Sigma^{-1})^{-1} (V^{-1} \hat{y} + \Sigma^{-1} N \hat{\theta}^{(m)})$$

$$\hat{\alpha} = [N^T N + \lambda W - N^T (V^{-1} + \Sigma^{-1})^{-1} \Sigma^{-1} N]^{-1} N^T \underbrace{(V^{-1} + \Sigma^{-1})^{-1}}_{V^*} V^{-1} \hat{y}$$

$$= [N^T \underline{(I - (V^{-1} + \Sigma^{-1})^{-1} \Sigma^{-1})} N + \lambda W]^{-1} N^T \underline{(V^{-1} + \Sigma^{-1})^{-1} V^{-1}} \hat{y}$$

$$\because D = I - V^* \Sigma^{-1} = V^* (V^{*-1} - \Sigma^{-1}) = V^* V^{-1}$$

$$= [N^T D N + \lambda W]^{-1} N^T D \hat{y}$$

$$\hat{\mu}(z) = \sum_{i=1}^{I} l_{\lambda,i}(z) \cdot \hat{y}_i$$

#' Builds `granular'' data

#'　　　　　　　　　　기울기, 기울기의 분산 구하기

#' obtains the regression slope and its variance

#' certainly not optimal but this step shouldn't take long regardless

#' @param x_k design matrix

#' @param y_k response vector

#' @param mod underlying model; eitherlmorglm`

#' @export

```
granular <- function(x_k, y_k, mod) {
  # summarizing the regression part
  if (mod == "glm")
    fit_lm <- glm(y_k ~ x_k, family = "binomial")
  if (mod == "lm")
    fit_lm <- lm(y_k ~ x_k)

  kth_beta_hat <- coef(fit_lm)[2]
  kth_var <- diag(vcov(fit_lm))[2]
  grain_out <- list(kth_beta_hat, kth_var)
  grain_out
}
```

#' Generates kerel matrix

#'

#' Generates kernel matrix of J by J, where J = length(z) for multilevel splines

#' certainly not optimal but this step shouldn't take long regardless.

#' Used the formulation from Reinsch (1967).

#' @author YD Hwang and ER Lee

#' @param z Mid-interval value vector, it is safe to assume this to be equi-distant, but in principle it doesn't have to be. it's not tested though.

#' @export

```
make_K <- function(z) {
  J <- length(z)
  Del <- matrix(0, nrow = J - 2, ncol = J)
  W <- matrix(0, nrow = J - 2, ncol = J - 2)
  h <- diff(z)
  for (l in 1:(J - 2)) {
    Del[l, l] <- 1/h[l]
    Del[l, (l + 1)] <- -1/h[l] - 1/h[(l + 1)]
    Del[l, (l + 2)] <- 1/h[(l + 1)]
    W[(l - 1), l] <- W[l, (l - 1)] <- h[l]/6
    W[l, l] <- (h[l] + h[l + 1])/3
  }
  K <- t(Del) %*% solve(W) %*% Del
  K
}
```

$D^T W^{-1} D$

커널 fnc 생성