Deep understanding of RKHS regression

Comparison with nonparametric smoothing, understanding of penalties

Sunwoo Lim

May 25, 2023

Review of nonparametric regression and project questions

Class of models for nonparametric regression : $y_i = f(x_i) + \epsilon_i, \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2), i = 1, ..., n$

A lot of situations in FDA required **nonparametric regression** for
1) making smooth functional object, 2) mean function estimation in sparse functional setting.

Models include:

1. Basis function model with model selection (either splines / FPCA)
2. Nonparametric smoothing with fixed basis (B-splines with second order derivative / Fourier with harmonic acceleration)
3. Reproducing Kernel Hilbert Space (RKHS) regression, also known as kernel ridge regression (KRR)

Observe the second and third model.

Nonparametric smoothing

$\min_{f \in \mathcal{S}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda J(f)$, where $\mathcal{S} := sp\{\boldsymbol{B}_1, ..., \boldsymbol{B}_p\}$, a p dimensional vector space of predetermined basis functions and $J(f)$, a penalty functional (e.g, $J(f) = \int f''^2$).

Letting $\boldsymbol{B} \in \mathbb{R}^{n \times p}$, a basis function evaluation matrix and $\boldsymbol{P} \in \mathbb{R}^{p \times p}$, a penalty matrix,

$\min_{\boldsymbol{\theta} \in \mathbb{R}^p} (\boldsymbol{y} - \boldsymbol{B\theta})^T (\boldsymbol{y} - \boldsymbol{B\theta}) + \lambda \boldsymbol{\theta}^T \boldsymbol{P\theta} \leftrightarrow \hat{\boldsymbol{\theta}} = (\boldsymbol{B}^T \boldsymbol{B} + \lambda \boldsymbol{P})^{-1} \boldsymbol{B}^T \boldsymbol{y}$

$\leftrightarrow \hat{f}(x) = \sum_{j=1}^{p} \hat{\theta}_j B_j(x)$.

RKHS regression

Letting $\mathcal{H}$ be the RKHS, and $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be the kernel, solve

$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda ||f||_{\mathcal{H}}^2$

$\leftrightarrow \min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{n} \alpha_j K(x_i, x_j)\right)^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j < K(\cdot, x_i), K(\cdot, x_j) >_{\mathcal{H}}$ (representator theorem)

$\leftrightarrow \min_{\boldsymbol{\alpha} \in \mathbb{R}^n} ||\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}||_2^2 + \lambda \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} \leftrightarrow \hat{\boldsymbol{\alpha}} = (\boldsymbol{K}^T \boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{K}^T \boldsymbol{y}$

$\leftrightarrow \hat{f}(x) = \sum_{j=1}^{n} \alpha_j K(x, x_j)$ (reproducing property).

Apparent similarities and differences

- Similarities
  1. Linear smoother
  2. solution in finite dimensional vector space
  3. Nonparametric smoothing can be viewed as finite truncation of KRR
- Differences
  1. Nonparametric smoothing: choose basis and penalty, KRR: choose kernel and the kernel will induce basis and penalty
  2. $p < n$ dimensional problem in nonparametric smoothing, n dimensional problem in KRR

Project questions

1. Understanding the basis and penalty $||f||_{\mathcal{H}}^2$ induced by different kernels
2. Understanding differences between KRR and nonparametric smoothing in depth

Understanding the basis & penalty by eigenvalue-eigenvector analysis of the smoother matrix

$\hat{f} = \boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{B} + \lambda\boldsymbol{P})^{-1}\boldsymbol{B}^T\boldsymbol{y} = \boldsymbol{S}_\lambda\boldsymbol{y}$: for smoothing splines.

$\hat{f} = \boldsymbol{K}(\boldsymbol{K}^T\boldsymbol{K} + \lambda\boldsymbol{I})^{-1}\boldsymbol{K}^T\boldsymbol{y} = \boldsymbol{S}_\lambda\boldsymbol{y}$: for KRR.

$\boldsymbol{S}_\lambda$ called **smoother matrix**.

Facts about smoother matrix

- $\boldsymbol{S}_\lambda^2 \preceq \boldsymbol{S}_\lambda$: shrinking nature
- Eigenvalues of $\boldsymbol{S}_\lambda$ are between 1 and 0, while the projection matrix has eigenvalues 1 or 0.
- $\boldsymbol{S}_\lambda = \sum_{i=1}^n \rho_i \boldsymbol{v}_i \boldsymbol{v}_i^T$, by eigendecomposition and $\boldsymbol{S}_\lambda\boldsymbol{y} = \sum_{i=1}^n \boldsymbol{v}_i\rho_i < \boldsymbol{v}_i, \boldsymbol{y} >$: decompose $\boldsymbol{y}$ by eigenbasis $\{\boldsymbol{v}_i\}_{i=1}^n$ with shrinkage by shrinking eigenvalues.
- $\{\boldsymbol{v}_i\}$ by decreasing eigenvalues tend to be more complex.
- When $\lambda > 0$, increasing λ, $\boldsymbol{S}$ has same eigenvectors, with smaller eigenvalues.
- Components with eigenvalues 1 are not penalized $\forall\lambda > 0$.
- Key : $\rho_i\boldsymbol{v}_i$ indicates which eigencomponent ($\boldsymbol{v_i}$) is used and how much shrinkage effect (ρ_i) is imposed on it.

Simulation settings

Data generated from $Y_i = f(x_i) + \epsilon_i$, $f : [0,1] \to \mathbb{R}$, $\epsilon_i \sim N(0, \sigma^2)$

1. $f_1(x) = 200 + sin(2\pi x)$: periodic sinusoid, smooth
2. $f_2(x) = 0.03\big(N(x; 0.02, 0.34^2) - N(x; 0.98, 0.34^2)\big) + 0.1504\big(N(x; 0.05, 0.06^2) - N(x; 0.95, 0.06^2)\big)$
3. $f_3(x) = 2.05\sqrt{x(1-x)}sin(\frac{2.1\pi}{x+0.05})$: nonlinear combination of sinusoid, not smooth (not of bounded variation)

- Train data of $n = 800$, uniformly sampled in $[0,1]$.
- Test data of $x = 0.001, 0.003, ..., 0.997, 0.999$: 500 points.
- $\sigma = 0.4$ for f_1 and f_2, $\sigma = 0.1$ for f_3.

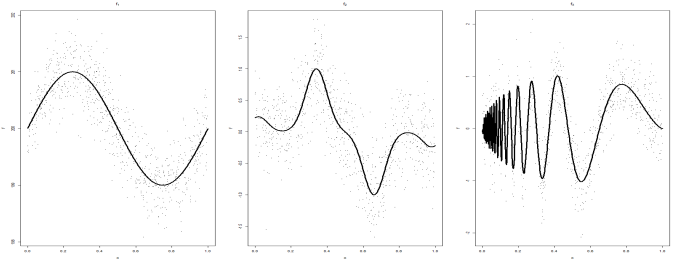
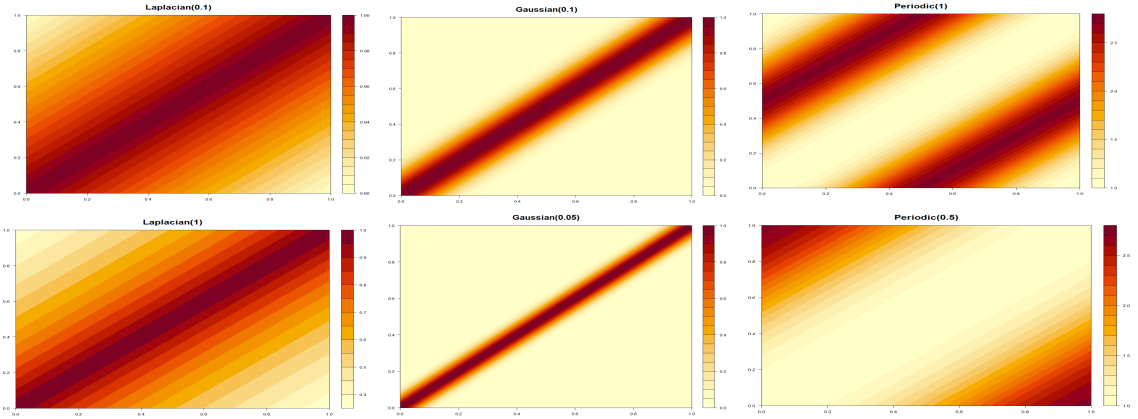


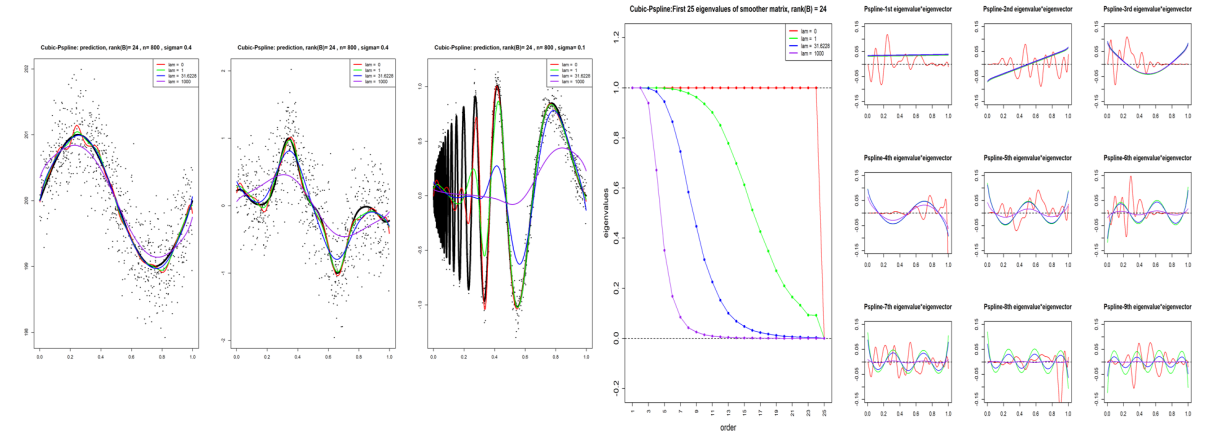
Figure: Simulation data (training data : dotted, true regression function : line)

Models to consider

- Cubic P-spline (very similar with cubic smoothing B-splines) with dimension 24 vs 400
- Laplacian kernel $K(x, y; \gamma) = exp(-\gamma||x - y||_1)$ with $\gamma = 0.1$ vs 1
- Gaussian kernel $K(x, y; \sigma) = exp(-0.5||x - y||_2^2/\sigma^2)$ with $\sigma = 0.1$ vs 0.05
- Periodic kernel $K(x, y; k) = exp(sin(k\pi|x - y|)^2))$ with $k = 1$ vs 0.5

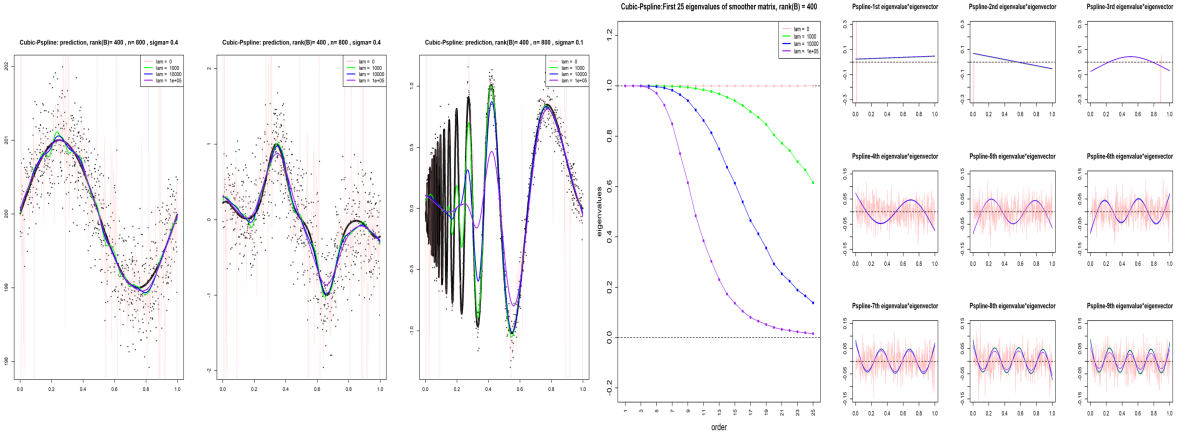


Cubic P-splines (with 20 interior knots $\leftrightarrow$ 24 basis functions)



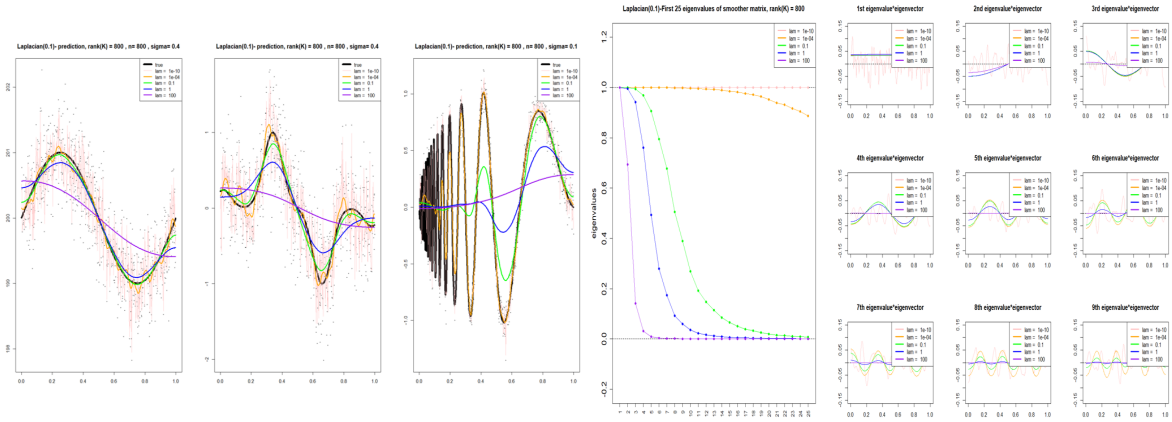
- When $\lambda = 0$, 24 components with eigenvalue 1 and rest are 0.
- When $\lambda > 0$, first two eigenvalues are 1 (not penalized) , and 3^{rd} to 24^{th} eigenvalues monotonically decrease.
- When $\lambda > 0$, all eigenvectors are identical and eigenvectors get more complex with decreasing eigenvalue.
- $\therefore$ Increasing λ, wiggly (high curvature) parts are first to be rectified.

Cubic P-splines (with 396 interior knots $\leftrightarrow$ 400 basis functions)



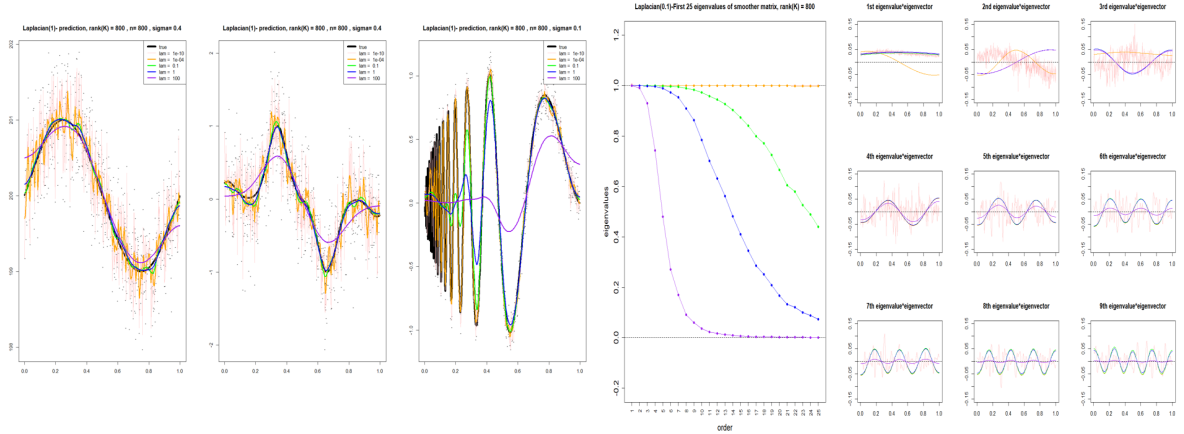
- When $\lambda = 0$, 400 components with eigenvalue 1 and rest are 0. Results in overly complicated function
- Compared to using 20 knots, early eigenvectors : slightly changed, but later eigenvectors : very similar.

Laplacian (with $\gamma = 0.1$)



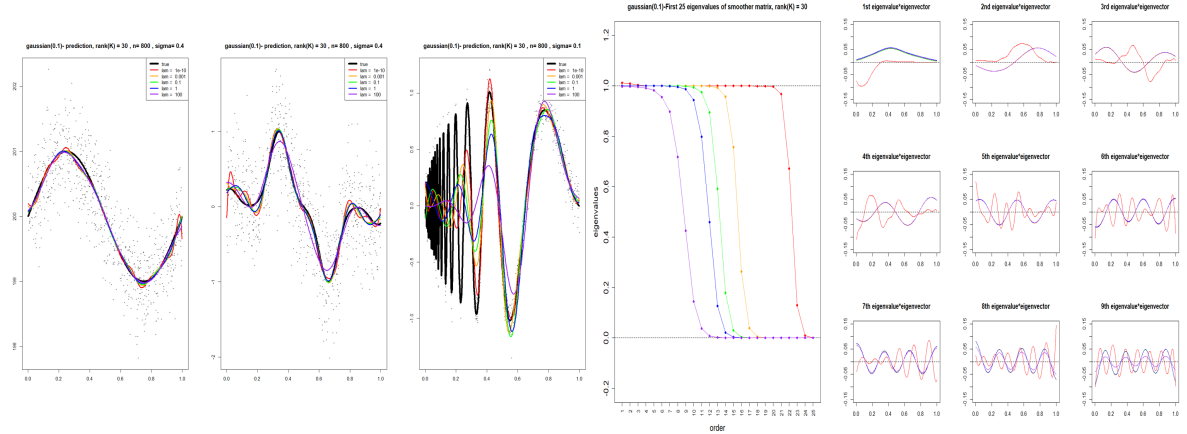
- $\lambda = 0 \rightarrow$ all eigenvalues 1 and eigenvectors are overly complicated.
- $\lambda > 0 \rightarrow$ first 1 eigenvalue is 1 and eigenvector is a constant. Eigenvalues monotonically decrease.
- $\lambda > 0 \rightarrow$ all eigenvectors are identical and eigenvectors get more complex with decreasing eigenvalue.
- Shape of eigenvectors look **sinusoidal**, pretty similar as P-splines :**polynomial**

Laplacian (with $\gamma = 1$)



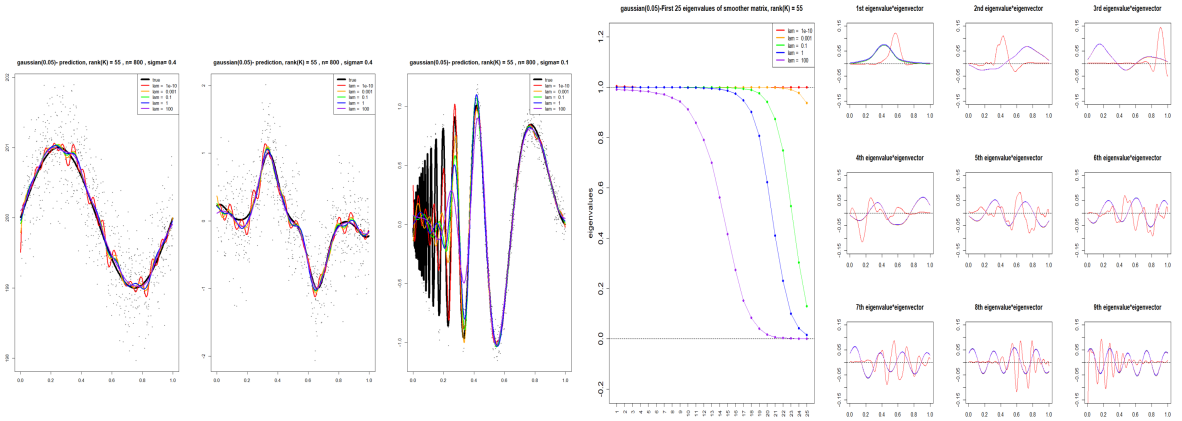
- Overall, wigglier behavior
- Same λ, but less penalty imposed (bigger eigenvalue)
- Similar eigenvectors as in previous case (sinusoidal)

Gaussian (with $\sigma = 0.1$)



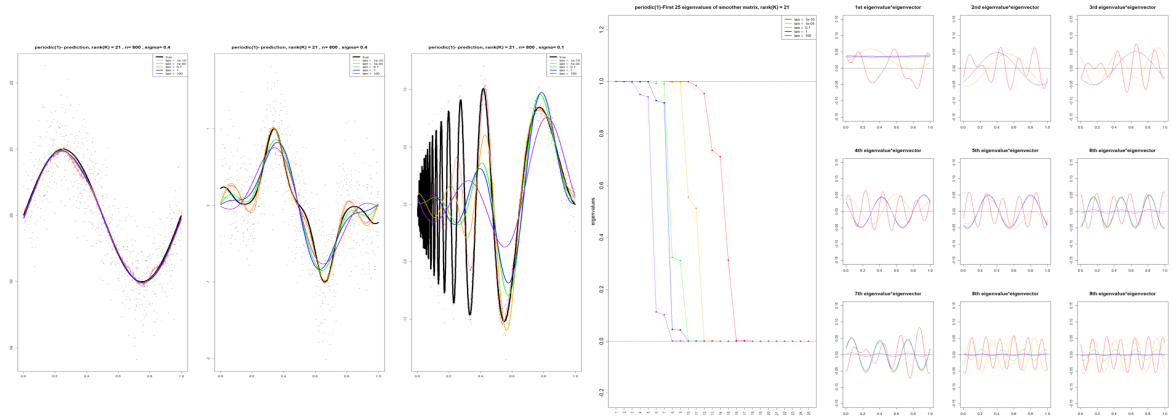
- First eigenvalue remains 1, but the first eigenvector is not a constant
- Shape of eigenvectors look **polynomial**, very similar as P-splines

Gaussian (with $\sigma = 0.05$)



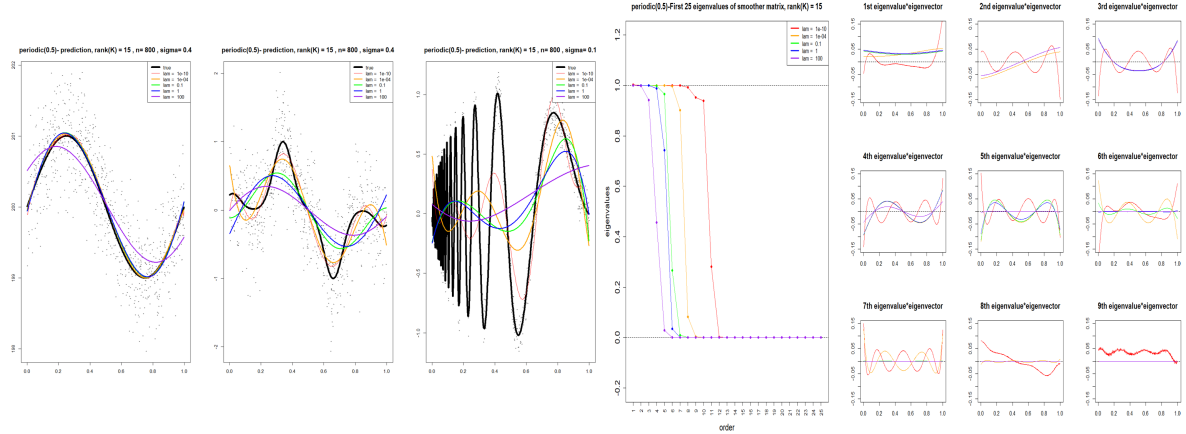
- Overall, wigglier behavior (same λ, but bigger eigenvalue $\leftrightarrow$ less penalty)
- early eigenvectors : slightly changed, but later eigenvectors : very similar

Periodic (with $k = 1$)



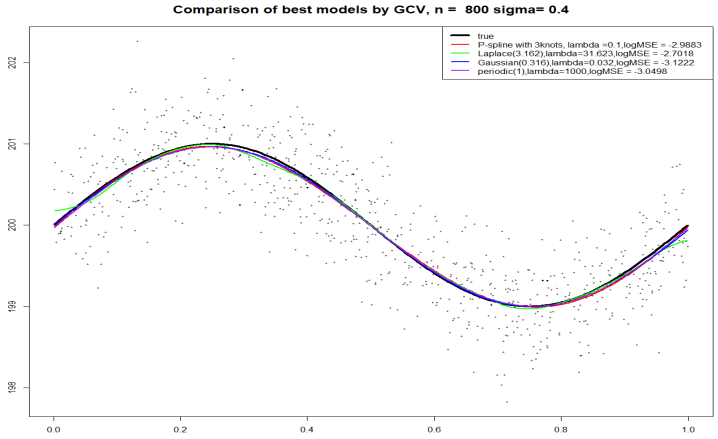
- First eigenvalue remains 1 and the first eigenvector is a constant function
- Shape of eigenvectors look **sinusoidal**, very similar as Laplacian
- Very well fit for f_1 as here, only simple eigenfunctions have big eigenvalues and eigenvalues shrink fastly.

Periodic (with $k = 0.5$)

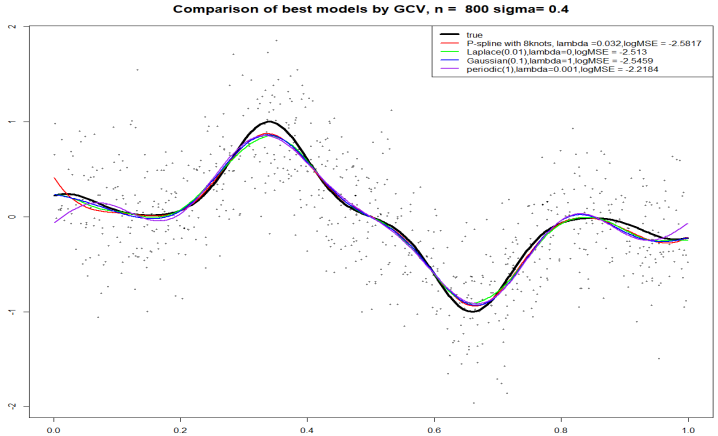


- The shape of the eigenvectors changed quite a lot compared to $k = 1$.
- eigenvectors shrink faster (resulting in simpler functions)

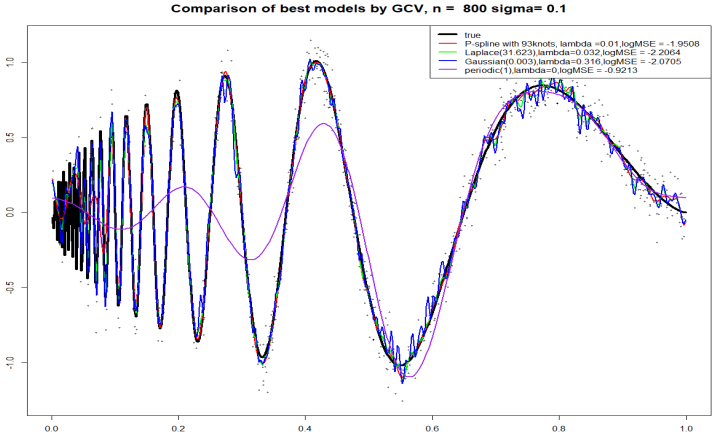
Best performer for each kernel selected by GCV



- Periodic kernel and Gaussian kernel performs well, while Laplacian kernel performs poorly (expected)
- Laplacian kernel tends to overfit



- Laplacian kernel and Gaussian kernel performs well, while periodic kernel performs poorly
- Periodic kernel results in sinusoidal fit



- Laplacian kernel and Gaussian kernel performs well, while periodic kernel performs poorly.
- Laplacian kernel well fits to a highly complex functions
- Overall, hard to adapt to varying smoothness (smooth in some location, highly oscillating in other locations)
- Periodic kernel results in sinusoidal fit

Summary

1. Understanding the basis functions and the penalty by eigenanalysis of the smoother matrix.
2. When $\lambda > 0$ (without numerical issue), the eigenvectors are identical with varying eigenvalues
3. When $\lambda > 0$ (without numerical issue), eigenvectors increase in complexity in the order of decreasing eigenvalues.
4. The shape of the kernel matrix of Laplacian and Gaussian are similar, but in most cases, Laplacian kernel matrix is positive definite, while PSD for Gaussian kernel.

   (Observe the difference between $||x-y||_1$ and $||x-y||_2^2$. L_2 norm square: rapid decay of kernel function value when two points get distant : banded matrix)
5. Periodic kernels lead to a sinusoidal fit
6. **Practically**, nonparametric smoothing with moderate number of knots is enough for smooth function estimation (Plus, computationally efficient and numerically stable ∵ inverting a big matrix is unstable)

References

1. https://www.stat.berkeley.edu/ryantibs/statlearn-s23/lectures/splines_rkhs.pdf
2. Hastie, T., Tibshirani, R., Friedman, J. H., Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.