STAT 542: Statistical Learning

RKHS and Kernel Ridge Regression

Ruoqing Zhu, Ph.D. <rqzhu@illinois.edu>

Course Website: https://teazrq.github.io/stat542/

March 31, 2022

Department of Statistics
University of Illinois at Urbana-Champaign

Outline

- RKHS Preliminaries
- The Representer Theorem
- Kernel Ridge Regression

Reproducing Kernel Hilbert Space

Space of Functions

- In many of our previous lectures, we considered estimating a function f within a certain space
- Example: Linear space

$$\{f : \exists w \in \mathbb{R}^p, f(x) = w^\mathsf{T} x, \forall x \in \mathbb{R}^p\}$$

- Example: Basis expansion

$$\{f : \exists w \in \mathbb{R}^m, f(x) = \sum_{j=1}^{m} w_j \phi_j(x), \forall x \in \mathbb{R}^p\}$$

where $(\phi_1(x), \phi_2(x), \ldots, \phi_m(x))$ is a pre-defined collection of m basis function, such as x^2, $\log(x)$, etc.

Space of Functions

- Oftentimes, these spaces of functions are either too simple or too difficult to work with
- Is it possible to have a space $\mathcal{H}$ of functions that is flexible enough, while the solution is also computationally simple
- We have seen such examples: smoothing spline — although we are solving the best function in a very large space (second order Sobolev), but the solution must be "represented" by the smoothing spline basis.

Reproducing Kernel Hilbert Space

- The Reproducing Kernel Hilbert Space (RKHS) is one such spaces that is flexible enough and computationally easy to work with
- One computational advantage of RKHS is that (see the kernel SVM example), when we want to calculate the inner product of two feature maps, it is the same as calculating the kernel:

$$k(x, z) = \langle \Phi(x), \Phi(z) \rangle$$

- Another computational advantage is that (smoothing spline example) if we solve for a penalized loss objective function by searching solutions in the RKHS, it solution has a finite sample representation.

Reproducing Kernel Hilbert Space

- Let's construct this RKHS.
- First, given a kernel function, we will view a data point $x \in \mathcal{X}$ as a real-valued function $k_x(\cdot) = k(x, \cdot) \in \mathbb{R}^{\mathcal{X}}$.
- Then, we will make a Hilbert space by first defining all finite linear combinations of $k_x(\cdot)$:

$$\mathcal{G} = \left\{ \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) \mid \alpha \in \mathbb{R}, n \in \mathbb{N}, x_i \in \mathcal{X} \right\}$$

Reproducing Kernel Hilbert Space

- For a Hilbert space, we need to equip it with an inner product ($\langle \cdot, \cdot \rangle$) and make it complete (all Cauchy sequence converges).
- For the inner product, if given any two functions k_x and k_z in $\mathcal{G}$, we define

$$\langle k_x, k_z \rangle = k(x, z)$$

- If we have $f = \sum_i \alpha_i k(x_i, \cdot)$ and $h = \sum_i \beta_i k(z_i, \cdot)$, then

$$\langle f, h \rangle = \sum_{i,j} \alpha_i \beta_j k(x_i, z_j)$$

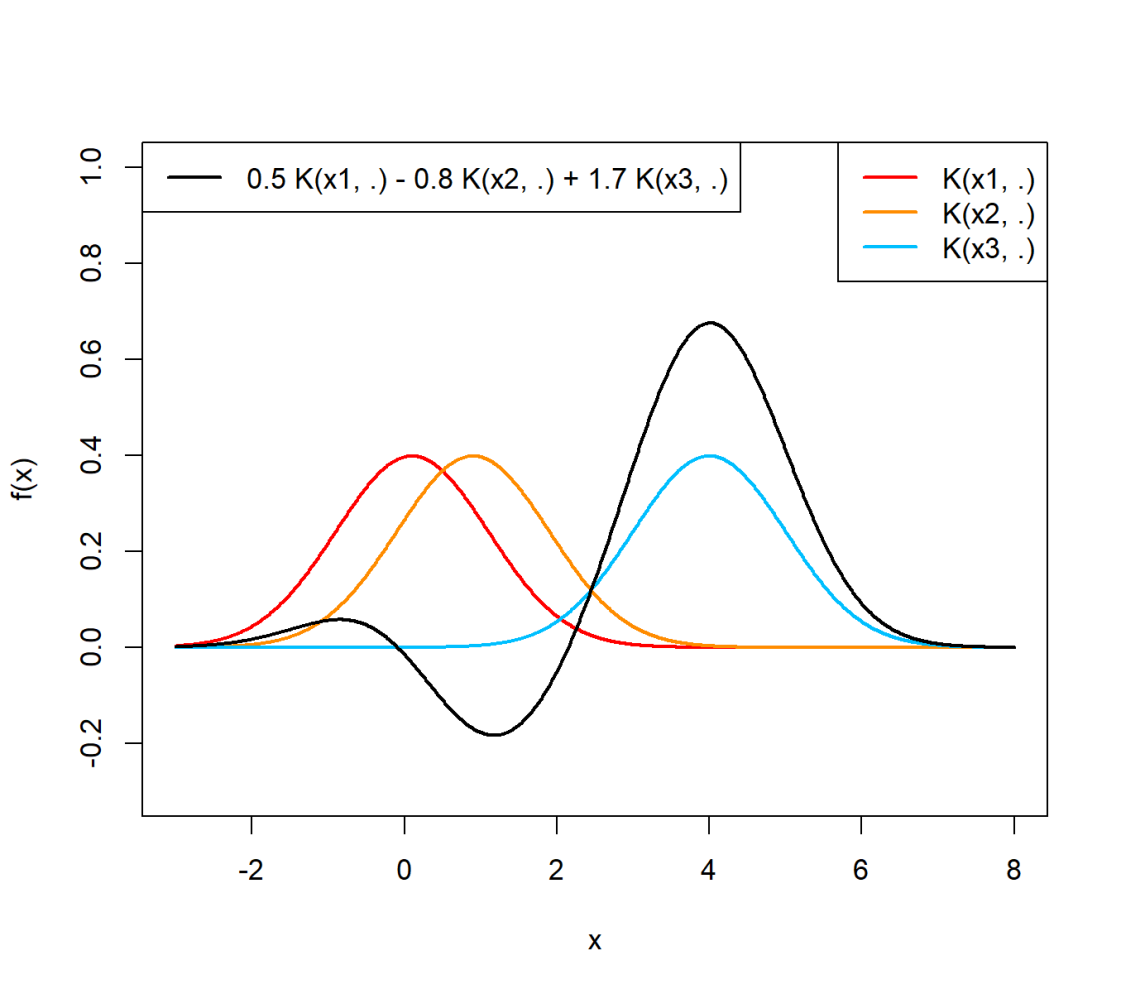
Reproducing Kernel Hilbert Space

- We then make $\mathcal{G}$ complete by including all limits of Cauchy sequences

$$\mathcal{H} = \bar{\mathcal{G}}$$

- $\mathcal{H}$ is our RKHS
- A (real) Hilbert space satisfies
  - symmetric: $\langle x, z \rangle = \langle z, x \rangle$
  - linear: $\langle ax_1 + bx_2, z \rangle = a\langle x_1, z \rangle + b\langle x_2, z \rangle$
  - positive definite: $\langle x, x \rangle \geq 0$ and $\langle x, x \rangle = 0$ iff $x = 0$

Reproducing Kernel Hilbert Space



The Reproducing Property

- Let's consider a function in $\mathcal{H}$, $f = \sum_i \alpha_i k(x_i, \cdot)$.
- We want to evaluate this function at x, i.e., $f(x)$.
- Instead, calculating its inner product with another function $k(x, \cdot)$ would "reproduce" this evaluation:

$$\begin{aligned}
\langle f, k(x, \cdot)\rangle &= \langle \sum_i \alpha_i k(x_i, \cdot), k(x, \cdot)\rangle \\
&= \sum_i \alpha_i \langle k(x_i, \cdot), k(x, \cdot)\rangle \\
&= \sum_i \alpha_i k(x_i, x) \\
&= f(x)
\end{aligned}$$

Kernel Functions

- By the Riesz representation theorem, any RKHS must be associated with a unique reproducing kernel K.
- As converse, by the Moore-Aronszajn theorem, any (symmetric and positive definite) kernel uniquely defines a RKHS
- The construction of kernels is very flexible, for example, we could take sums, transformations and products of existing kernel functions and make them a new kernel.

Examples

- The space $\mathcal{H}$ of all linear functions $f(x) = w^{\mathsf{T}}x$ is a RKHS
- In fact, we can also find the associated kernel using the reproducing property. First, take $f = K_z$ and by the reproducing property, we have

$$\langle K_z, K_x \rangle_{\mathcal{H}} = K_z(x) = K(z, x)$$

  Then, since f is a linear function, we must have

$$K(z, x) = z^{\mathsf{T}}x$$

- Hence, the linear kernel is the unique kernel associated with the space of linear functions.

Representor Theorem

- Hence, it is now safe to say that the RKHS and the kernel are two equivalent concepts.
- However, RKHS is still an infinite-dimensional vector space. How can we find the solution in this space? (think about the smoothing spline example).
- The Representor Theorem shows that the solution has to live in a finite-dimensional subspace
- Computationally tractable

The Representor Theorem

Representor Theorem

- Representer Theorem (Kimeldorf and Wahba, 1970)
  - Consider a positive-definite real-valued kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, associated with a RKHS $\mathcal{H}$.
  - If we are given a set of data $\{x_i, y_i\}_{i=1}^n$, then if we search for the best solutions in $\mathcal{H}$ of the optimization problem

$$\widehat{f} = \underset{f \in \mathcal{H}}{\arg\min}\ \mathcal{L}(\{y_i, f(x_i)\}_{i=1}^n) + p(\|f\|_{\mathcal{H}}^2)$$

  - The solution must have the form

$$\widehat{f} = \sum_{i=1}^{n} w_i K(\cdot, x_i)$$

- Here $\mathcal{L}$ is a loss function, p is a monotone penalty.
- The proof is similar to the smoothing spline example.

Connections

- In fact, these three things always go together:
  - $K(\cdot, \cdot)$: a symmetric and positive definite kernel
  - $\mathcal{H}$: a RKHS
  - $\Phi(x)$: a set of basis (by Mercer's theorem)
- Note: for the equivalence between K and Φ, we also showed examples of $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\mathsf{T}\mathbf{z})^2$ and $e^{-\gamma\|\mathbf{x}-\mathbf{z}\|}$ in the SVM lecture
- With these guarantees, we can first pick a kernel K, then the RKHS $\mathcal{H}$ is uniquely defined. However, to find the best function $\mathcal{H}$ to optimize our objective function, we only need to worry about its finite representation

$$f(\cdot) = \sum_{i=1}^{n} w_i K(\cdot, x_i).$$

Proof

- The proof of Representor Theorem consists of several steps
- Consider a penalized loss objective function

$$\widehat{f} = \arg\min_{f \in \mathcal{H}} \quad \mathcal{L}\big(\{y_i, f(x_i)\}_{i=1}^n\big) + p(\|f\|^2)$$

- Use the kernel K associated with $\mathcal{H}$ to define a set of functions

$$K(\cdot, x_i),\ K(\cdot, x_2),\ \cdots,\ K(\cdot, x_n)$$

- For any $f \in \mathcal{H}$, find its project on span$\{K(\cdot, x_i),\ \cdots,\ K(\cdot, x_n)\}$, and let the orthogonal complement as $h(\cdot)$.

Proof

- Hence we have

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, x_i) + h(\cdot),$$

where $\langle h(\cdot), K(\cdot, x_i) \rangle = 0$ for all i.

- For the loss function part, we only evaluate $f(\cdot)$ on the training data, which is (by the reproducing property)

$$\begin{aligned} f(x_j) &= \left\langle \sum_{i=1}^{n} \alpha_i K(\cdot, x_i) + h(\cdot), K(\cdot, x_j) \right\rangle \\ &= \sum_{i=1}^{n} \alpha_i \left\langle K(\cdot, x_i), K(\cdot, x_j) \right\rangle, \end{aligned}$$

which has nothing to do with $h(\cdot)$.

Proof

- For the penalty part,

$$\begin{aligned}\|f\|^2 =& \|\sum_{i=1}^{n} \alpha_i K(\cdot, x_i) + h(\cdot)\|^2 \\ &= \|\sum_{i=1}^{n} \alpha_i K(\cdot, x_i)\|^2 + \|h(\cdot)\|^2 \\ &\geq \|\sum_{i=1}^{n} \alpha_i K(\cdot, x_i)\|^2\end{aligned}$$

- As long as the penalty function $p(\cdot)$ is monotone increasing, the penalty on $\|f\|^2$ is larger than the penalty of its projection on the space spanned by n samples.
- This makes its projection a better solution, which must be represented by the observed sample.

Kernel Ridge Regression

Kernel Ridge Regression

- Lets go back to the classical ridge regression:

$$\frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2$$

- Recall that the exact solution can be obtained through the normal equation after taking the derivative

$$2\mathbf{X}^\mathsf{T}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 2n\lambda\boldsymbol{\beta}$$

- And

$$\boldsymbol{\beta} = (\mathbf{X}^\mathsf{T}\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$$

Kernel Ridge Regression

- Let's introduce an alternative view, and use the same technique in the SVM example to solve it

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2$$

- Introduce a new variable $z_i = y_i - x_i^\mathsf{T}\boldsymbol{\beta}$, for $i = 1, \ldots, n$. Then the original problem becomes (with a change of constant):

$$\begin{aligned} \underset{\mathbf{z},\boldsymbol{\beta}}{\text{minimize}} \quad & \frac{1}{2n\lambda}\|\mathbf{z}\|^2 + \frac{1}{2}\|\boldsymbol{\beta}\|^2 \\ \text{subject to} \quad & z_i = y_i - x_i^\mathsf{T}\boldsymbol{\beta}, \quad \text{for} \quad i = 1, \ldots, n \end{aligned}$$

- The form looks similar to the SVM primal problem

Kernel Ridge Regression

- Recall the SVM constrained optimization, we introduced Lagrangian multipliers α_i's (here $\alpha_i \in \mathbb{R}$):

$$\mathcal{L} = \frac{1}{2n\lambda}\mathbf{z}^\mathsf{T}\mathbf{z} + \frac{1}{2}\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\beta} + \sum_i \alpha_i(y_i - x_i^\mathsf{T}\boldsymbol{\beta} - z_i)$$

- Again, we have the primal:

$$\min_{\mathbf{z},\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \quad \mathcal{L}$$

- But we know the dual is probably easier:

$$\max_{\boldsymbol{\alpha}} \min_{\mathbf{z},\boldsymbol{\beta}} \quad \mathcal{L}$$

Kernel Ridge Regression

- First, we solve for the best $\mathbf{z}$ and $\boldsymbol{\beta}$ for any given $\boldsymbol{\alpha}$ (in the dual):

$$\frac{\partial \mathcal{L}}{\partial z_i} = \frac{1}{n\lambda} z_i - \alpha_i = 0 \quad \text{for} \quad i = 1, \ldots, n$$

$$\text{and} \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_i \alpha_i x_i = 0,$$

- This gives

$$z_i = n\lambda\alpha_i$$

$$\boldsymbol{\beta} = \sum_i \alpha_i x_i$$

- The best $\boldsymbol{\beta}$ is a linear function of x_i's. The solution must lie in the span of training data.

Kernel Ridge Regression

- How to predict a future point x? Since we have a linear model, the prediction for a new subject with covariate x is

$$f(x) = x^\mathsf{T}\beta = \frac{1}{\lambda}\sum_i \alpha_i x^\mathsf{T} x_i$$

- If we view $x^\mathsf{T}x_i$ as a linear kernel:

$$K(x, x_i) = x^\mathsf{T}x_i$$

- Then the prediction function is just

$$f(x) = \frac{1}{\lambda}\sum_i \alpha_i K(x, x_i)$$

- The reproducing property!

Kernel Ridge Regression

- To finish the dual form, plugin the optimizers of z_i and $\boldsymbol{\beta}$:

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \min_{\mathbf{z},\boldsymbol{\beta}} \mathcal{L} &= \max_{\boldsymbol{\alpha}} \frac{n\lambda}{2}\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\alpha} + \frac{1}{2}\sum_{i,j}\alpha_i x_i^\mathsf{T} x_j \alpha_j \\ &\quad + \sum_i \alpha_i \Big(y_i - x_i^\mathsf{T}\sum_j \alpha_j x_j - n\lambda\alpha_i\Big) \\ &= \max_{\boldsymbol{\alpha}} -\frac{n\lambda}{2}\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\alpha} - \frac{1}{2}\sum_{i,j}\alpha_i x_i^\mathsf{T} x_j \alpha_j + \boldsymbol{\alpha}^\mathsf{T}\mathbf{y}\end{aligned}$$

- We can apply the kernel trick

$$\begin{aligned}&\max_{\boldsymbol{\alpha}} -\frac{n\lambda}{2}\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\alpha} - \frac{1}{2}\sum_{i,j}\alpha_i K(x_i, x_j)\alpha_j + \boldsymbol{\alpha}^\mathsf{T}\mathbf{y} \\ = &\max_{\boldsymbol{\alpha}} -\frac{n\lambda}{2}\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\mathsf{T}\mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\alpha}^\mathsf{T}\mathbf{y}\end{aligned}$$

Kernel Ridge Regression

- To obtain the solution, we take a derivative w.r.t. $\boldsymbol{\alpha}$:

$$-n\lambda\mathbf{I}\boldsymbol{\alpha} - \mathbf{K}\boldsymbol{\alpha} + \mathbf{y} = \mathbf{0}$$

- The solution is given by

$$\boldsymbol{\alpha} = (n\lambda\mathbf{I} + \mathbf{K})^{-1}\mathbf{y}$$

- For linear kernel, $\mathbf{X}^\mathsf{T}\boldsymbol{\alpha} = \boldsymbol{\beta}$
- But for non-linear kernel functions this is easy to work with

Kernel Ridge Regression

- Back to the original Loss + Penalty form, a Kernel ridge regression is given by

$$
\begin{aligned}
&\arg\min_{\boldsymbol{\alpha}} && \frac{1}{n}\sum_i \left(y_i - f(x_i)\right)^2 + \lambda \|f\|_{\mathcal{H}}^2 \\
&= \arg\min_{\boldsymbol{\alpha}} && \frac{1}{n}\sum_i \left(y_i - \sum_{j=1}^{n} \alpha_j K(x_i, x_j)\right)^2 + \lambda \|f\|_{\mathcal{H}}^2 \\
&= \arg\min_{\boldsymbol{\alpha}} && \frac{1}{n}\|\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\alpha}^\mathsf{T}\mathbf{K}\boldsymbol{\alpha}
\end{aligned}
$$

Kernel Ridge Regression

- By taking the derivative with respect to $\boldsymbol{\alpha}$, we have (normal equation):
$$-\frac{1}{n}\mathbf{K}(\mathbf{y}-\mathbf{K}\boldsymbol{\alpha})+\lambda\mathbf{K}\boldsymbol{\alpha}=\mathbf{0}$$
- One solution is
$$\widehat{\alpha}=(\mathbf{K}+n\lambda\mathbf{I})^{-1}\mathbf{y}$$
- Then, for any new target point x_0, the prediction is
$$\widehat{f}(x_0)=\sum_{i=1}^{n}\widehat{\alpha}_i K(x_0,x_i)$$
- The tuning parameter α could be selected by cross-validation.

Kernel Ridge Regression

- Advantage: we can fit nonlinear functions $f(x)$.
- Computational cost:
  - Solving for $\boldsymbol{\alpha}$ involves inverting an $n \times n$ matrix.
  - All training samples x_i need to be saved for prediction.
- Inversion of a $n \times n$ matrix is $\mathcal{O}(n^3)$ computational complexity. This could be a disadvantage when n is extremal large.