

# Introduction to Machine Learning

## Lab 4: Kernel Regression

Hongteng Xu

March 30, 2023

### 1 Motivation

- Implement some typical kernel functions, and check their feasibility in different models. You may find that some kernels work better for some models while perform poorly for the other models:)
- Implement classic kernel methods like Nadaraya-Watson estimator and kernel ridge regression (KRR). Observe their differences and connections.
- Take KRR as a good example. Try to sharpen your hyperparameter fine-tuning feelings.

### 2 Tasks

Please read Lecture 5 carefully before doing this lab work.

1. Implement the following four kinds of kernel functions and use them to achieve the Nadaraya-Watson estimator shown in the lecture.

$$\begin{aligned} \text{RBF: } K_h(\mathbf{x}, \mathbf{x}') &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{h}\right), & \text{Gate: } K_h(\mathbf{x}, \mathbf{x}') &= \begin{cases} \frac{1}{h}, & \|\mathbf{x} - \mathbf{x}'\|_1 \leq h \\ 0 & \text{Otherwise.} \end{cases} \\ \text{Triangle: } K_h(\mathbf{x}, \mathbf{x}') &= \begin{cases} \frac{2}{h}(1 - \frac{\|\mathbf{x} - \mathbf{x}'\|_1}{h}) & \|\mathbf{x} - \mathbf{x}'\|_1 \leq h \\ 0 & \text{Otherwise} \end{cases}, & \text{Linear: } K(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^T \mathbf{x}' \end{aligned} \quad (1)$$

where  $h$  is the hyperparameter. The NW estimator:  $y_{new} = \sum_{n=1}^N \frac{K(\mathbf{x}_{new}, \mathbf{x}_n)}{\sum_{i=1}^N K(\mathbf{x}_{new}, \mathbf{x}_i)} y_n$ .

2. Implement the closed-form solution of kernel ridge regression:

$$\min_{\mathbf{a}} \|\mathbf{y} - \mathbf{K}\mathbf{a}\|_2^2 + \tau \mathbf{a}^T \mathbf{K}\mathbf{a}. \quad (2)$$

(Hint: Use as few computations as possible.)

3. **Struggle with KRR's SGD:** Implement a stochastic gradient descent (SGD) algorithm to solve (2). (Hint: Do your best to make it work, and think about whether it actually “works” or not.)
4. **Struggle with KRR's BCD:** Leverage the knowledge you learned from Lasso. Design an iterative algorithm (block coordinate descent, BCD for short) to solve (2). (Hint: Update a subset of  $\mathbf{a}$ 's elements per step, and think about whether this BCD algorithm is possible to reduce complexity.)