4033/5033 Assignment 2

Antonio Natusch Zarco

Due on Sep 21, 2025

In this assignment, we will implement an efficient variant of KRR called accelerated kernel ridge regression (AKRR). Recall the optimal KRR model is expressed by all training points i.e., $\beta = \sum_{i=1}^{n} \alpha_i \phi(x_i)$. AKRR approximates this model using a (random) subset of m training points i.e.,

$$\tilde{\beta} = \sum_{i=1}^{m} \alpha_i \phi(x_i), \tag{1}$$

where integer m is a hyperparameter and often $m < n$.

Other parts of AKRR are the same as KRR i.e., we plug (1) back to the following objective function

$$J(\tilde{\beta}) = \sum_{i=1}^{n} (\phi(x_i)^T \tilde{\beta} - y_i)^2 + \lambda \tilde{\beta}^T \tilde{\beta}, \tag{2}$$

to get a dual objective $J(\tilde{\alpha})$ of $\tilde{\alpha} = [\alpha_1, \ldots, \alpha_m]^T$.

Then, we solve $\min_{\tilde{\alpha}} J(\tilde{\alpha})$ to get the optimal $\tilde{\alpha}$ and predict the label of any point z as

$$\tilde{\beta}^T \phi(z) = \sum_{j=1}^{m} \alpha_j \phi(x_j)^T \phi(z) = \sum_{j=1}^{m} \alpha_j k(x_j, z). \tag{3}$$

Part I: Theory

[Task 1] Derive the optimal $\tilde{\alpha}$ for problem $\min_{\tilde{\alpha}} J(\tilde{\alpha})$. (tip: you may first derive a matrix form of $J(\tilde{\alpha})$ using proper kernel matrices e.g., $\tilde{K} \in \mathbb{R}^{n \times m}$ where the m columns correspond to the m instances in (1)).

[Task 1 - Solution]

Defining the proper kernel matrices:

$\tilde{K} \in \mathbb{R}^{n \times m}, (\tilde{K}_{ij}) = k(x_i, x_j)$,

and the basis-basis kernel

$K_m \in \mathbb{R}^{m \times m}, (K_m)_{j\ell} = k(x_j, x_\ell)$

Let $\tilde{\alpha} = (\tilde{\alpha_1}, ..., \tilde{\alpha_m})^T$ and $y = (y_1, ..., y_n)^T$.

Predictions on the training set are

$\hat{y} = \tilde{K\alpha}$

Now, for the optimality, differentiate and set to zero:

$\nabla_{\tilde{\alpha}} J = 2\tilde{K}^T(\tilde{K\alpha} - y) + 2\lambda K_m \tilde{\alpha} = 0$ Thus, the normal equations are:

$(\tilde{K}^T \tilde{K} + \lambda K_m)\tilde{\alpha} = \tilde{K}^T y$

and the closed form solution is

$\tilde{\alpha} = (\tilde{K}^T\tilde{K} + \lambda K_m)^{-1}\tilde{K}^T y$

[Task 2] Justify the computational complexity for computing the optimal $\tilde{\alpha}$ is $O(mnp + m^2n)$. (tip: you just need to explain which part of the $\tilde{\alpha}$ computation contributes to which term in the big O notation.)

[Task 2 - Solution]

1. Build the rectangular Kernel $\tilde{K}$ of size $n \times m$

- Evaluations of the $n \cdot m$ kernel are needed.
- For common kernels that use an inner product or distance in R^p (whether linear, radial basis function, or polynomial), each evaluation is $O(p)$.
- Thus, the cost is $\boxed{O(mnp)}$.

2. Form the normal-equation pieces and right-hand side

- Compute $\tilde{K}^T\tilde{K}$ (an $m \times m$ Gram matrix of $\tilde{K}$): each of the m^2 entires is a dot product of two n-length columns $\rightarrow O(n)$ each $\rightarrow \boxed{O(m^2n)}$.
- Compute $\tilde{K}^T y : O(nm)$ (lower order than m^2n).

3. Solve the $m \times m$ linear system

- Using Cholesky on $\tilde{K}^T\tilde{K} + \lambda K_m : O(m^3)$.
- In the intended regime $n \geq m$, the $O(m^2n)$ term dominates $O(m^3)$, so it's absorbed by m^2n.

Putting the dominant terms together: $\boxed{O(mnp) + O(m^2n)}$.

- $O(mnp) \leftarrow$ building $\tilde{K}$ (kernel evaluations).
- $O(m^2n) \leftarrow$ forming $\tilde{K}^T\tilde{K}$ (and it dominates the linear-solve in the common $n \geq m$) setting.

Part II: Implementation

Implement AKRR based on 'hw3_akrr.py'. Below are some instructions.

– Given a training set, simply choose the first m instances to express $\tilde{\beta}$ for AKRR.

– Use RBF kernel and pick a proper γ by yourself for the kernel. Also pick a proper λ by yourself.

– Report testing MSE versus m in Figure 1. Pick at least five values of m by yourself which can provide a comprehensive picture of its impact on model performance. A KRR with RBF kernel from scikit-learn is given in the template. Do not remove it. Give it the same γ and λ you chose for AKRR. Figure 1 should contain a curve of AKRR error and a line of KRR error (which should not change as m increases).

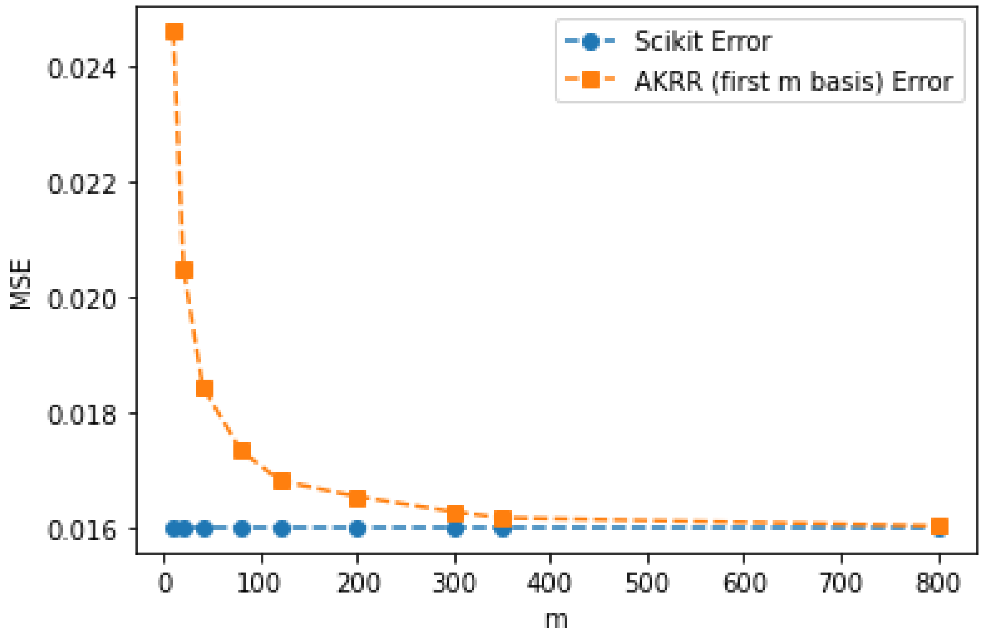

Fig. 1. Testing Error versus m for AKRR and KRR.

<u>Submission Instruction</u>

Please submit two files to Canvas.

(i) Submit 'hw3.pdf'. It should contain your answers to all questions in this document. A latex template 'hw2_Latex.txt' is provided.

(ii) Submit 'hw2_akrr.py'. It should be the code that generates Figure 1.