[ML20] Assignment 8

Danny Radosevich

Due: Mar 27 (before class)

[1] If $\kappa_1(a,b)$ and $\kappa_2(a,b)$ are two valid kernels, prove that $g(a,b) = \kappa_1(a,b) + \kappa_2(a,b)$ is also a valid kernel.

0.1 Proof

Assume at $\kappa_1(a,b)$ and $\kappa_2(a,b)$ are valid kernels. Then $\exists$ a kernel $g(a,b)$ such that:
$g(a,b) = \kappa_1(a,b) + \kappa_2(a,b)$
Because K_1 and K_2 are valid kernels the must both satisfy Mercer's Condition, so $\int \kappa_1 \geq 0$ and $\int \kappa_2 \geq 0$.
with K_1 and K_2 we can define an integral such that:
$g(a,b) = \int\int(\kappa_1(a,b) + \kappa_2(a,b))g(x)g(y)dxdy$
$= \int\int \kappa_1(a,b)g(x)g(y)dxdy + \int\int \kappa_2(a,b)g(x)g(y)dxdy$
Where $= \int\int \kappa_1(a,b)g(x)g(y)dxdy + \int\int \kappa_2(a,b)g(x)g(y)dxdy \geq 0$, $g(a,b)$ is also a valid kernel, again by Mercer's condition.

[2] Kernel ridge regression is a powerful nonparameteric model but suffers from the $O(n^3)$ computational complexity, where n is the size of training set. Please develop an efficient approximation KRR (AKRR).

Here are the requirements:

(i) Your model β still minimizes training error over the entire training set, i.e.,

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

(ii) For AKRR, assume the optimal model is made of k random training instances ($k < n$), i.e.,

$$\beta = \sum_{j=1}^{k}\alpha_j\phi(x_j). \tag{2}$$

(iii) Plug (2) back to (1), and derive the analytic solution of $\alpha = [\alpha_1, \ldots, \alpha_n]^T$. Importantly, show that the computational complexity of getting α is now $O(k^3)$.

$j(\alpha) = (\kappa\alpha - Y)^T(\kappa\alpha - y) + \lambda\alpha^T\kappa\alpha$
where $\kappa \in \mathbb{R}^{kxk}$ and is the Gram matrix
$J'(\alpha) = 0$
$\alpha = (\kappa + \lambda I)^{-1}Y$
Let $Y \in \mathbb{R}^{kx1}$
Previously to find α it was O(n^3), now both K and Y are bounded by k, rather than n so the order to find α is O(k^3)

Implement KRR and AKRR, using Gaussian kernel with hyper-parameter σ. Report your results below.

(a) Draw a figure of two curves for KRR. One is its training MSE versus σ and the other is its testing MSE versus σ. Properly choose 7 candidate values of σ so we may observe overfitting and underfitting.

(b) Properly choose a σ for AKRR and fix it. Draw a figure of two curves for AKRR. One is its training MSE versus k and the other is its testing MSE versus k. Choose 7 candidate values of k and a proper σ so that you can get as smooth and convergent curves as possible.

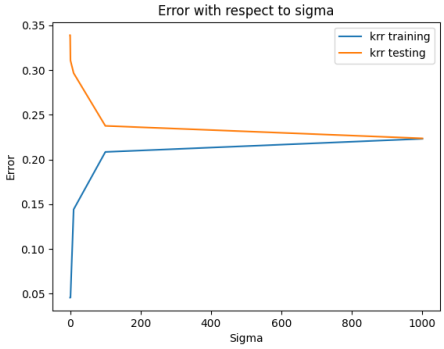

Fig. 1. KRR MSE versus σ.

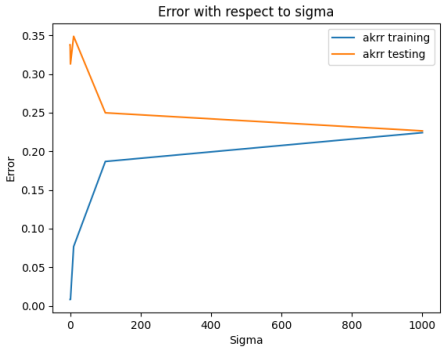

Fig. 2. AKRR MSE versus k. Here $\sigma = \ldots$

.

(c) [Bonus] ARKK+ is built on ARKK, but it selects k training instances in a non-random fashion. Please propose your own selection technique and briefly explain it here. You will get 30% bonus if you can show ARKK+ outperforms ARKK, i.e., under the same k and σ, ARKK+ has lower testing MSE – however, both ARKK and ARKK+ show have reasonable testing MSE, as compared with KRR. Report your results in the following table. (Search Python library that can record the running time of a segment of codes.)

Table 1. Performance of KRR, AKRR and AKRR+ ($k = ...$, $\sigma = ...$)

Method	**Training MSE**	**Testing MSE**	**Training Time**
KRR			
AKRR			
AKRR+			