CS 771 Major Assignment

Chadhurbala R V 251110022 **Nishu Kumar** 251110054 **Pranjal** 251110058

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, Kanpur, UP

Question 1: Derivation of the Kernel $\widetilde{K}$

Given Model

We are given a semi-parametric regression model of the form

$$y = p^\top \phi(z) \cdot x + b, \tag{1}$$

where $x \in \mathbb{R}_{\geq 0}$ is a scalar input, $z \in \mathbb{R}^2$, and ϕ is the feature map associated with the polynomial kernel

$$K(z_1, z_2) = \langle \phi(z_1), \phi(z_2) \rangle = (z_1^\top z_2 + c)^d. \tag{2}$$

The goal is to construct a feature map $\psi(x, z)$ such that the model becomes purely linear in that feature space:

$$\tilde{p}^\top \psi(x, z) = p^\top \phi(z) \cdot x + b,$$

and then compute the corresponding kernel

$$\widetilde{K}\big((x_1, z_1), (x_2, z_2)\big) = \langle \psi(x_1, z_1), \psi(x_2, z_2) \rangle.$$

Feature Map Construction

To incorporate both the multiplicative factor x and the bias b into a single linear model, we define the augmented feature map

$$\psi(x, z) = \begin{bmatrix} x\,\phi(z) \\ 1 \end{bmatrix}, \qquad \tilde{p} = \begin{bmatrix} p \\ b \end{bmatrix}. \tag{3}$$

Then

$$\tilde{p}^\top \psi(x, z) = p^\top \phi(z)\, x + b,$$

So, this choice of ψ satisfies the semi-parametric equivalence requirement.

Kernel Computation

Let (x_1, z_1) and (x_2, z_2) be two inputs. Their kernel value under the ψ-feature map is

$$\begin{aligned} \widetilde{K}\big((x_1, z_1), (x_2, z_2)\big) &= \left\langle \begin{bmatrix} x_1\phi(z_1) \\ 1 \end{bmatrix}, \begin{bmatrix} x_2\phi(z_2) \\ 1 \end{bmatrix} \right\rangle & (4) \\ &= (x_1\phi(z_1))^\top (x_2\phi(z_2)) + 1 \cdot 1 & (5) \\ &= x_1 x_2 \langle \phi(z_1), \phi(z_2) \rangle + 1. & (6) \end{aligned}$$

Substituting the polynomial kernel gives

$$\boxed{\widetilde{K}\big((x_1, z_1), (x_2, z_2)\big) = x_1 x_2 \left(z_1^\top z_2 + c\right)^d + 1.} \tag{7}$$

Preprint. Under review.

Question 2: Hyperparameter Selection

Search Space

We tuned the polynomial kernel parameters

$$d \in \{1, 2, 3, 4\}, \qquad c \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 2.0\},$$

using 5-fold cross-validation (CV) on the public training set.

Results

For each polynomial degree d, we evaluated all coefficient values c using 5-fold cross-validation. The tables below list the mean CV R2 score for every setting.

c	$d = 1$	$d = 2$	$d = 3$	$d = 4$
0.0	0.7861128168	0.5978894252	0.4765569466	0.4005928949
0.1	0.9709021807	0.9710487201	0.9694483280	0.9460725069
0.2	0.9709020782	0.9710801117	0.9709945688	0.9701950049
0.3	0.9709020291	0.9710810526	0.9710360521	0.9709715013
0.4	0.9709020017	0.9710810171	0.9710368423	0.9710130565
0.5	0.9709019844	0.9710809350	0.9710356107	0.9710145731
0.6	0.9709019725	0.9710808713	0.9710345816	0.9710130743
0.7	0.9709019638	0.9710808251	0.9710338184	0.9710116115
0.8	0.9709019572	0.9710807914	0.9710332439	0.9710104333
0.9	0.9709019520	0.9710807659	0.9710327983	0.9710094978
1.0	0.9709019478	0.9710807463	0.9710324430	0.9710087457
2.0	0.9709019285	0.9710806673	0.9710308569	0.9710054111

Table 1: Mean CV R2 scores for all (d, c) combinations

The top 5 settings ranked by mean CV R2 score are listed below:

Degree d	Coef0 c	Mean CV R2
2	0.3	0.9710810526
2	0.4	0.9710810171
2	0.5	0.9710809350
2	0.6	0.9710808712
2	0.7	0.9710808251

Table 2: Top 5 hyperparameter combinations (highest mean CV R2).

Following trend was observed from the above result:

- Degree $d = 2$ dominates all other choices.
- Values of c in the range 0.3–2.0 produce nearly identical performance.
- Degrees $d = 3$ and $d = 4$ approach, but never surpass, the best $d = 2$ results.

Test Performance

To verify generalization, we retrained models using the full public training set for each (d, c) combination and evaluated predictions on the unseen test set. The following table reports the corresponding Test R2 scores.

c	$d = 1$	$d = 2$	$d = 3$	$d = 4$
0.0	0.7898374853	0.5714201558	0.4404333752	0.3629864141
0.1	0.9696919782	0.9699487999	0.9695487244	0.9540010150
0.2	0.9696882486	0.9699273037	0.9700162202	0.9698363775
0.3	0.9696869951	0.9699203953	0.9699609773	0.9699987677
0.4	0.9696863664	0.9699177678	0.9699415181	0.9699532645
0.5	0.9696859886	0.9699165164	0.9699343274	0.9699333742
0.6	0.9696857364	0.9699158272	0.9699312909	0.9699247467
0.7	0.9696855562	0.9699154083	0.9699298667	0.9699204199
0.8	0.9696854210	0.9699151349	0.9699291461	0.9699179109
0.9	0.9696853157	0.9699149467	0.9699287616	0.9699162673
1.0	0.9696852315	0.9699148116	0.9699285492	0.9699150839
2.0	0.9696848523	0.9699143748	0.9699283223	0.9699101010

Table 3: Test R2 scores for all (d, c) configurations

Selected Hyperparameters

Since cross-validation reflects expected performance on unseen data, we anticipate that these settings should also transfer well to the secret evaluation set.

Based on the CV ranking , we select:

$$\boxed{d^\star = 2, \qquad c^\star = 0.3, \qquad \text{CV R2} = 0.9710810526.}$$

This configuration is used in `submit.py`.

Question 4:

Given

We are given the 1089-dimensional linear model $w \in \mathbb{R}^{1089}$ of an XOR arbiter PUF obtained from two 32-bit arbiter PUFs. Each arbiter PUF has 32 stages with 4 delays per stage, so there are

$$32 \times 4 \times 2 = 256$$

delays in total:

$$\{a_i, b_i, c_i, d_i\}_{i=0}^{31}, \qquad \{p_i, q_i, r_i, s_i\}_{i=0}^{31}.$$

For Single arbiter PUF (from delays to linear model):

For a k-bit arbiter PUF with $k = 32$. At stage i there are four path delays (p_i, q_i, r_i, s_i) as in the lecture notes. Following the standard derivation, we define

$$\alpha_i = \frac{p_i - q_i + r_i - s_i}{2}, \tag{8}$$

$$\beta_i = \frac{p_i - q_i - r_i + s_i}{2}, \qquad 0 \le i \le k-1. \tag{9}$$

The linear model $w \in \mathbb{R}^{k+1}$ (with bias) can be written in terms of these variables as

$$w_0 = \alpha_0, \tag{10}$$

$$w_i = \alpha_i + \beta_{i-1}, \qquad 1 \le i \le k-1, \tag{11}$$

$$w_k = \beta_{k-1}. \tag{12}$$

Thus, for a single arbiter PUF, the mapping

$$(p_i, q_i, r_i, s_i)_{i=0}^{k-1} \longmapsto w \in \mathbb{R}^{k+1}$$

is linear.

Kronecker-Product Decomposition of the XOR Arbiter PUF

Let $u, v \in \mathbb{R}^{k+1}$ be the linear models of the two component arbiter PUFs (with delays (a_i, b_i, c_i, d_i) and (p_i, q_i, r_i, s_i), respectively). The XOR arbiter PUF can be written as

$$w = u \otimes v \in \mathbb{R}^{(k+1)^2} = \mathbb{R}^{1089}, \tag{13}$$

where $\otimes$ denotes the Kronecker product. Therefore,

$$u \otimes v = (u_0 v_0, u_0 v_1, \ldots, u_0 v_k,\ u_1 v_0, \ldots, u_k v_k)^\top.$$

To recover vectors $\mathbf{u}$ and $\mathbf{v}$ from their Kronecker product $\mathbf{w} = \mathbf{u} \otimes \mathbf{v} \in \mathbb{R}^{(k+1)^2}$ with $k = 32$, we exploit the fact that the reshaped matrix is rank 1. We reshape $\mathbf{w}$ into a $(k+1) \times (k+1)$ matrix

$$Z = \mathrm{reshape}(\mathbf{w}, (k+1, k+1)).$$

For an ideal XOR arbiter PUF we have

$$Z_{ij} = u_i v_j \quad \Longleftrightarrow \quad Z = \mathbf{u}\mathbf{v}^\top,$$

so Z is a rank-1 matrix.

To recover $\mathbf{u}$ and $\mathbf{v}$ from $\mathbf{w}$ we proceed as follows.

1. Form $Z \in \mathbb{R}^{33 \times 33}$ by reshaping $\mathbf{w}$.
2. Compute the thin singular value decomposition

   $$Z = U \Sigma V^\top,$$

   and extract the leading singular triplet $(\sigma_1, \mathbf{u}_1, \mathbf{v}_1)$.
3. Set

   $$\hat{\mathbf{u}} = \sqrt{\sigma_1}\, \mathbf{u}_1, \qquad \hat{\mathbf{v}} = \sqrt{\sigma_1}\, \mathbf{v}_1.$$

By Eckart–Young theorem [1], this rank-1 truncated SVD provides the optimal approximation even in the presence of noise, minimizing $\|Z - \hat{\mathbf{u}}\hat{\mathbf{v}}^\top\|_F$.

Inverting an arbiter PUF model to non-negative delays

Once we have extracted $u, v \in \mathbb{R}^{33}$, we invert each of them independently to obtain the delays for the two PUFs.

Let $w \in \mathbb{R}^{k+1}$ ($k = 32$) denote the model of a single arbiter PUF (either u or v). We need to find (α_i, β_i) that satisfy

$$w_0 = \alpha_0, \tag{14}$$
$$w_i = \alpha_i + \beta_{i-1}, \quad 1 \le i \le k-1, \tag{15}$$
$$w_k = \beta_{k-1}. \tag{16}$$

This system has infinitely many solutions. To get one of the valid solution, we choose the following simple assignment:

$$\alpha_i = w_i, \qquad 0 \le i \le k-1, \tag{17}$$
$$\beta_i = 0, \qquad 0 \le i \le k-2, \tag{18}$$
$$\beta_{k-1} = w_k. \tag{19}$$

This choice satisfies all the above equations for any given w.

Since

$$\alpha_i = \frac{p_i - q_i + r_i - s_i}{2}, \tag{20}$$
$$\beta_i = \frac{p_i - q_i - r_i + s_i}{2}. \tag{21}$$

we can obtain:

$$p_i - q_i = \alpha_i + \beta_i \tag{22}$$
$$r_i - s_i = \alpha_i - \beta_i \tag{23}$$

Let

$$A_i = p_i - q_i, \qquad B_i = r_i - s_i. \tag{24}$$

For any real number D, we can represent it as a difference of two non-negative numbers in many ways. One way of doing it is

$$x = \max(D, 0), \qquad y = \max(-D, 0), \tag{25}$$

which guaranties $x, y \geq 0$ and $x - y = D$.

We apply this to each stage i:

$$A_i = p_i - q_i \qquad \Rightarrow \qquad p_i = \max(A_i, 0), \qquad q_i = \max(-A_i, 0), \tag{26}$$
$$B_i = r_i - s_i \qquad \Rightarrow \qquad r_i = \max(B_i, 0), \qquad s_i = \max(-B_i, 0). \tag{27}$$

This way all delays are non-negative and recomputing α_i and β_i from these delays yields exactly the same (α_i, β_i) (and hence the same model w). We apply this procedure to u to obtain (a_i, b_i, c_i, d_i) and to v to obtain (p_i, q_i, r_i, s_i).

References

[1] Carl Eckart and Gale Young. *The approximation of one matrix by another of lower rank*. Psychometrika, 1(3):211–218, 1936.

[2] CS771 Lecture Material. Department of Computer Science and Engineering, IIT Kanpur, 2025. Available: `https://www.cse.iitk.ac.in/users/purushot/courses/ml/2025-26-a/lectures.html`