CS771 - Introduction to Machine Learning

Instructor: Prof. Purushottam Kar

Assignment 1

July 17, 2024

Problem 1.1

By giving a detailed mathematical derivation (as given in the lecture slides), show how for a simple arbiter PUF, a linear model can predict the time it takes for the upper signal to reach the finish line. Specifically, give derivations for a map : 0,132 → RD mapping 32-bit 0/1-valued challenge vectors to D-dimensional feature vectors (for some D ¿ 0) so that for any arbiter PUF, there exists a D-dimensional linear model W RD and a bias term b R such that for all CRPs c 0,132, we have W(c)+b = tu(c) where tu(c) is the time it takes for the upper signal to reach the finish line when challenge c is input. Remember that tu(c) is, in general, a non-negative real number (say in milliseconds) and need not be a Boolean bit. W,b may depend on the PUF-specific constants such as delays in the multiplexers. However, the map (c) must depend only on c (and perhaps universal constants such as 2,2 etc). The map must not use PUF-specific constants such as delays.

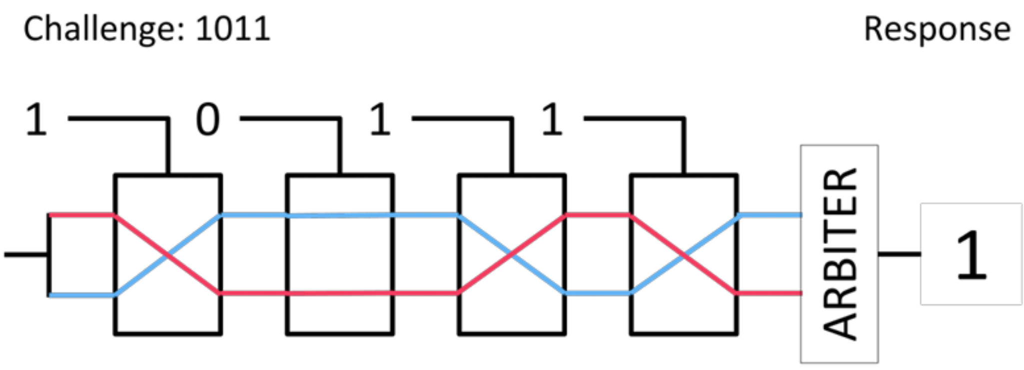

Figure 1: A simple arbiter PUF with 4 multiplexers

$$t_i^u = (t_{i-1}^u + P_i)(1 - C_i) + C_i(t_{i-1}^i + S_i)$$
$$t_i^l = (t_{i-1}^l + Q_i)(1 - C_i) + C_i(t_{i-1}^i + R_i)$$

$$\triangle_i^u = t_i^u - t_i^l$$
$$\triangle_i^u = (1 - C_i)(\triangle_{i-1} + p_i - q_i) + C_i(s_i - r_i - \triangle_{i-1})$$
$$\triangle_i^u = (1 - 2C_i)\triangle_{i-1} + (q_i - p_i + s_i - r_i)C_i + p_i - q_i$$

Assuming

$$\alpha_i = (p_i - q_i + r_i + s_i)/2$$
$$\beta_i = (p_i - q_i - r_i + s_i)/2$$

Solving the arbiter PUF gives the above solution as

$$\triangle_{31} = w_0.x_0 + w_1.x_1 + w_2.x_2 + ... + w_{31}.x_{31} + \beta_{31}$$

where

$$\mathrm{x}_i = d_i + d_{i+1} + ... + d_{31}$$
$$\mathrm{d}_i = (1 - 2C_i)$$

Adding t_i^u *and* t_i^l

$$\mathrm{S}_i = t_i^u + t_i^u$$
$$\mathrm{S}_i = (1 - C_i)(t_{i-1}^u + t_{i-1}^l + p_i + q_i) + C_i(t_{i-1}^u + t_{i-1}^l + s_i + r_i)$$
$$\mathrm{S}_i = (1 - C_i)(S_{i-1} + p_i + q_i) + C_i(S_{i-1} + s_i + r_i)$$
$$\mathrm{S}_i = S_{i-1} + (1 - C_i)(p_i + q_i) + C_i(s_i + r_i)$$
$$\mathrm{S}_i = S_{i-1} + C_i(s_i + r_i - p_i - q_i) + p_i + q_i$$

Solving this will give

$$\mathrm{S}_{31} = w'_0.x'_0 + w'_1.x'_1 + w'_2.x'_2 + ... + w'_{31}.x'_{31} + \gamma'_{31}$$

where

$$\mathrm{x'}_i = C_i$$
$$\mathrm{w'}_i = s_i + r_i - p_i - q_i$$
$$\gamma'_i = \sum(p_i + q_i)$$

Now we have solved both sum and difference for upper and lower time in an arbiter PUF so now

$$\mathrm{t}_{31}^u = (\triangle_{31} + S_{31})/2$$
$$\mathrm{t}_{31}^u = w_0.x_0/2 + w_i.x_1/2 + ... + w_{31}.x_{31}/2 + w'_0.x'_0/2 + w'_1.x'_1/2 + w'_2.x'_2/2 + ... + w'_{31}.x'_{31}/2 + \beta'_{31}/2 + \gamma'_{31}/2$$

Simplyfing

$$\mathrm{t}_{31}^u = w_0.x_0 + w_i.x_1 + ... + w_{31}.x_{31} + w'_0.x'_0 + w'_1.x'_1 + w'_2.x'_2 + ... + w'_{31}.x'_{31} + \beta'_{31}$$

Where

$$\mathrm{x}_i = d_i + d_{i+1} + ... + d_31$$
$$\mathrm{d}_i = (1 - 2C_i)$$
$$\mathrm{x'}_i = C_i$$

Problem 1.2

What dimensionality does the linear model need to have to predict the arrival time of the upper signal for an arbiter PUF? The dimensionality should be stated clearly and separately in your report, and not be implicit or hidden away in some calculations.

Solution : So in this above solution it is clear that we obtained 64 dimentions but if we observe closely then

$$\mathrm{x}_{31} = (1 - 2C_i) \;\; and \;\; x'_{31} = C_{31}$$

which can be easily combined to be used as one by adjusting other variables.
So the dimentionality would be 63 with x'$_{31}$

not existing getting absorbed in x$_{31}$

Problem 1.3

Similarly, show how a linear model can predict Response1 for a COCO-PUF. As before, your linear model may depend on the delay constants in PUF0 and PUF1 but your map must not use PUF-specific constants such as delays.

Solution : $\mathrm{A}_i = p_i - a_i$

$$B_i = s_i - d_i$$
$$Q_i = q_i - b_i$$
$$P_i = r_i - e_i$$

$$\mathrm{T}_2^u = (t_1^u + P_2)(1 - C_2) + C_2(t_1^i + S_2)$$
$$\mathrm{T}_i^u = (t_{i-1}^u + P_i)(1 - C_i) + C_i(t_{i-1}^i + S_i)$$
$$\mathrm{T}_i^l = (t_{i-1}^l + q_i)(1 - C_i) + C_i(t_{i-1}^u + r_i)$$
$$\mathrm{K}_i^u = (K_{i-1}^u + a_i)(1 - C_i) + C_i(K_{i-1}^l + d_i)$$
$$\mathrm{K}_i^l = (K_{i-1}^l + b_i)(1 - C_i) + C_i(K_{i-1}^u + e_i)$$

$$\triangle_i^u = (T_i^u - K_i^u) = (T_{i-1}^u - K_{i-1}^u + p_i - a_i)$$
$$= (\mathrm{T}_{i-1}^u - K_{i-1}^u + p_i - a_i) * (1 - C_i) + C_i(T_{i-1}^l - K_{i-1}^l + S_i - d_i)$$
$$=(\triangle_{i-1}^u + p_i - a_i)(1 - C_i) + C_i(\triangle_{i-1}^l + s_i - d_i)$$
$$=\mathrm{C}_i(\triangle_{i-1}^l - \triangle_{i-1}^u) + (B_i - A_i) * C_i + \triangle_{i-1}^u + A_i$$
$$\triangle_i^l = C_i(\triangle_{i-1}^l) + \triangle_{i-1}^u + (P_i - Q_i)C_i +_{i-1}^l + Q_i$$

$$\mathrm{X}_i = \triangle_i^u + \triangle_i^l = (B_i + P_i + Q_i - A_i)C_i + \triangle_{i-1}^u + \triangle_{i-1}^l + A_i + Q_i$$
$$\mathrm{Y}_i = \triangle_i^u - \triangle_i^l = (1 - 2C_i)(\triangle_{i-1}^u - \triangle_{i-1}^l) + (B_i - A_i + Q_i - P_i)C_i + A_i - Q_i$$

Y_i *is similar to arbiter PUF* X:

$$\alpha_i = (A_i - B_i + P_i - Q_i)/2$$
$$\beta_i = (A_i - B_i - P_i + Q_i)/2$$
$$\mathrm{d}_i = (1 - 2 \cdot c_i)$$

$$
\begin{aligned}
\mathrm{Y}_{31} &= w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots + w_{31} x_{31} + \beta_{31} \\
&= \mathrm{w}^T x + b
\end{aligned}
$$

$$\mathrm{x}_i = d_i \cdot d_{i+1} \cdot d_{i+2} \ldots. \cdot d_{31}$$

$$\mathrm{w}_0 = \alpha_0$$

$$\mathrm{w}_i = \alpha_0$$

Now for X:

$$\gamma_i = B_i - A_i + P_i - Q_i$$

$$\mathrm{X}_{31} = w'_0.x'_0 + w'_1.x'_1 + w'_2.x'_2 + ... + w'_{31}.x'_{31} + \mathrm{E}_{31}$$

$$\text{w'i} = \gamma i \qquad \mathrm{E}i = \sum(A_j + Q_j) \qquad \mathrm{E}i = \beta' i$$

$$\text{x'i} = \text{ci}$$

$$\triangle_{31}^{u} = (X_{31} + Y_{31})/2$$

$$\triangle_{31}^{l} = (X_{31} - Y_{31})/2$$

Problem 1.4

What dimensionality do your need the linear model to have to predict Response0 and Response1 for a COCO-PUF? This may be the same or different from the dimensionality you needed to predict the arrival times for the upper signal in a simple arbiter PUF. The dimensionality should be stated clearly and separately in your report, and not be implicit or hidden away in some calculations

Solution : The total dimensions are found to be 63.

Problem 1.6

Report outcomes of experiments with both the sklearn.svm.LinearSVC and sklearn.linear model.LogisticRegression methods when used to learn the linear model. In particular, report how various hyperparameters affected training time and test accuracy using tables and/or charts. Report these experiments with both LinearSVC and Logisti cRegression methods even if your own submission uses just one of these methods or some totally different linear model learning method (e.g. RidgeClassifier) In particular, you must report how at least 2 of the following affect training time and test accuracy:
(a) changing the loss hyperparameter in LinearSVC (hinge vs squared hinge)
(b) setting C in LinearSVC and LogisticRegression to high/low/medium values
(c) changing tol in LinearSVC and LogisticRegression to high/low/medium values
(d) changing the penalty (regularization) hyperparameter in LinearSVC and LogisticRegression (l2 vs l1)

Solution :

(1)

Linear SVC

for hinge

Test Accuracy for Response0 : 0.9836

Test Accuracy for Response1 : 0.9949

Time for train: 0.48372623440000095

Time for Map: 0.004954472199995052

For squared hinge Test Accuracy for Response0: 0.9802

Test Accuracy for Response1: 0.9972

Time for train: 0.9647416249999651

Time for Map: 0.004492467399973066

(2)

Linear SVC

C = low(0.01)

Test Accuracy for Response0: 0.9753

Test Accuracy for Response1: 0.9822

Time for train: 0.4078560581999966

Time for Map: 0.005099750599998743

C = medium(1)

Test Accuracy for Response0: 0.9836

Test Accuracy for Response1: 0.9949

Time for train: 1.1084746008000024

Time for Map: 0.005108231600013368

C = high(100)

Test Accuracy for Response0: 0.9549

Test Accuracy for Response1: 0.9941

Time for train: 1.1112683815999957

Time for Map: 0.00447765400000435

Logistic Regresssion

C = low(0.01)

Test Accuracy for Response0: 0.9769

Test Accuracy for Response1: 0.9911

Time for train: 0.4428860805999989

Time for Map: 0.0059085557999992485

C = medium(1)

Test Accuracy for Response0: 0.9805

Test Accuracy for Response1: 0.9966

Time for train: 0.67828792559998871

Time for Map: 0.007645803000013984

C = high(100)

Test Accuracy for Response0: 0.9808

Test Accuracy for Response1: 0.9984

Time for train: 0.5240781424000034

Time for Map: 0.007520615000004227