CS771 Assignment 1

Aditi **Adithya** **Parv Goyal** **Siddhi Vora** **Vishal Kumar**

1 Cross Connection PUF

1.1 Time taken by upper signal to reach the finish line

We know that for a simple arbiter PUF:

$$t_i^u = (1 - c_i).(t_{i-1}^u + p_i) + c_i.(t_{i-1}^l + s_i)$$

We can rewrite this as:

$$t_i^u - t_{i-1}^u = c_i.(t_{i-1}^l - t_{i-1}^u - p_i + s_i) + p_i$$

$$t_2^u - t_1^u = c_2.(-\Delta_1 + s_2 - p_2) + p_2$$

$$\vdots$$

$$t_{32}^u - t_{31}^u = c_{32}.(-\Delta_{31} + s_{32} - p_{32}) + p_{32}$$

After adding all the terms:

$$t_{32}^u = -\sum_{i=2}^{32} c_i(\Delta_{i-1}) + \sum_{i=2}^{32} c_i.(s_i - p_i) + \sum_{i=2}^{32} p_i$$

$$2.t_{32}^u = -\sum_{i=2}^{32} 2.c_i(\Delta_{i-1}) + \sum_{i=2}^{32} 2.c_i.(s_i - p_i) + \sum_{i=2}^{32} 2.p_i$$

Using:

$$1 - 2.c_i = d_i$$

And Adding and Subtracting

$$\sum_{i=2}^{32} \Delta_{i-1}$$

$$2.t_{32}^u = -2.c_{32}.\Delta_{31} - ... - 2c_2.\Delta_1 + (\Delta_{31} - \Delta_{31}) + ... + (\Delta_1 - \Delta_1) + \sum_{i=2}^{32}(1 - d_i).(s_i - p_i) + \sum_{i=2}^{32} 2.p_i$$

$$2.t_{32}^u = d_{32}.\Delta_{31} + d_{31}.\Delta_{30} + ... + d_2.\Delta_1 - (\Delta_{31} + ... + \Delta_1) + \sum_{i=2}^{32}(1 - d_i).(s_i - p_i) + \sum_{i=2}^{32} 2.p_i$$

Preprint. Under review.

$$2.t^u_{32} = d_{32}.\Delta_{31} - (\Delta_{31} - d_{31}.\Delta_{30}) - ... - (\Delta_1 - d_2.\Delta_1) - (\Delta_{31} + ... + \Delta_1) + \sum_{i=2}^{32}(1-d_i)(s_i - p_i) + \sum_{i=2}^{32} 2.p_i$$

We know that:

$$\Delta_i - d_i.\Delta_{i-1} = \alpha_i.d_i + \beta_i$$

$$2.t^u_{32} = d_{32}.\Delta_{31} - (\alpha_{31}.d_{31} + \beta_{31}) - (\alpha_{30}.d_{30} + \beta_{30}) - ... - (\alpha_1.d_1 + \beta_1) + \sum_{i=2}^{32}(1-d_i).(s_i - p_i) + \sum_{i=2}^{32} 2.p_i$$

$$2.t^u_{32} = d_{32}.\Delta_{31} - \alpha_{31}.d_{31} - \alpha_{30}.d_{30} - ... - \alpha_1.d_1 - \sum_{i=2}^{32}(d_i).(s_i - p_i) + \lambda$$

Here:

$$\lambda = \sum_{i=2}^{32} s_i + p_i - \beta_{i-1}$$

Assuming:

$$y_i = s_i - p_i$$

the above equation can be written as:

$$2.t^u_{32} = [d_{32}.(\Delta_{31} - y_{32})] - [(\alpha_{31} + y_{31}).d_{31} - ... - (\alpha_2 + y_2).d_2 - \alpha_1.d_1] + \lambda$$

which can be further simplified to:

$$t^u_{32} = \gamma_{32}.d_{32} + \gamma_{31}.d_{31} + ... + \gamma_2.d_2 + \gamma_1.d_1 + \lambda$$

where:

$$\gamma_i = -(\alpha_i + y_i)/2 \text{ (for 1<i<32)}$$

$$\gamma_{32} = (\Delta_{31} - y_{32})/2$$

$$\gamma_1 = -\alpha_1/2$$

1.2 Dimensionality of Linear Model to predict arrival time

Here, $d_{32}.(\Delta_{31} - y_{32})$ provides 32 terms and $\gamma_{31}...\gamma_1$ are 31 terms which means our feature space contains total 63 terms. $[d_1, d_2, d_3, ..., d_{32}, d_{32}d_{31}, d_{32}d_{31}d_{30}, ..., d_{32}d_{31}d_{30}d_1]$ which is basically the feature space of $\Delta_{32} \cup d_1, d_2, d_3, ..., d_31$

So, **D = 63**.

1.3 COCO PUF

1.3.1 Preditcting Response0

By following a similar process as in *1.1* we get t^l_{32} (Time taken by lower signal of PUF0) as:

$$2.t^l_{32} = -[d_{32}.(\Delta_{31} + z_{32})] + (\alpha_{31} - z_{31}).d_{31} + ... + (\alpha_2 - z_2).d_2 + \alpha_1.d_1 + \mu$$

Where:

$$z_i = r_i - q_i$$

$$\mu = \sum_{i=2}^{32} r_i + q_i + \beta_{i-1}$$

which can be further simplified to:

$$t_{32}^l = A_{32}.d_{32} + A_{31}.d_{31} + ... + A_2.d_2 + A_1.d_1 + \mu$$

where:

$$A_i = (\alpha_i + z_i)/2 \text{ (for 1<i<32)}$$
$$A_{32} = -(\Delta_{31} - y_{32})/2$$
$$A_1 = \alpha_1/2$$

Similarly we can write $\bar{t}_{32}^l$ (Time taken by lower signal of PUF1) as:

$$2.\bar{t}_{32}^l = -[d_{32}.(\bar{\Delta}_{31} + \bar{z}_{32})] + (\bar{\alpha}_{31} - \bar{z}_{31}).d_{31} + ... + (\bar{\alpha}_2 - \bar{z}_2).d_2 + \bar{\alpha}_1.d_1 + \bar{\mu}$$

which can be further simplified to:

$$t_{32}^l = \bar{A}_{32}.d_{32} + \bar{A}_{31}.d_{31} + ... + \bar{A}_2.d_2 + \bar{A}_1.d_1 + \bar{\mu}$$

where:

$$\bar{A}_i = (\bar{\alpha}_i + \bar{z}_i)/2 \text{ (for 1<i<32)}$$
$$\bar{A}_{32} = -(\bar{\Delta}_{31} - \bar{z}_{32})/2$$
$$\bar{A}_1 = \bar{\alpha}_1/2$$

Response0 = 0 if $t_{32}^l - \bar{t}_{32}^l < 0$
Response0 = 1 if $t_{32}^l - \bar{t}_{32}^l > 0$

This can also be written as:

$$Response0 = \frac{1 + sign(t_{32}^l - \bar{t}_{32}^l)}{2}$$

Here:

$$t_{32}^l - \bar{t}_{32}^l = \frac{-[d_{32}.(\Delta_{31} - \bar{\Delta}_{31} + z_{32} - \bar{z}_{32})] + \sum_{i=2}^{31}(\alpha_i - \bar{\alpha}_i - z_i + \bar{z}_i).d_i + (\alpha_1 - \bar{\alpha}_1).d_1 + \mu - \bar{\mu}}{2}$$

$$t_{32}^l - \bar{t}_{32}^l = \frac{[d_{32}.(A_{32} - \bar{A}_{32}) + d_{31}.(A_{31} - \bar{A}_{31}) + ... + d_1.(A_1 - \bar{A}_1)] + (\mu - \bar{\mu})}{2}$$

We can assume :

$$B_i = \frac{A_i - \bar{A}_i}{2}$$

$$K = \frac{\mu - \bar{\mu}}{2}$$

1.3.2 Predicting Response1

By following similar process, we can write:

$Response1 = \dfrac{1 + sign(t^u_{32} - \bar{t}^u_{32})}{2}$ Here:

$$t^u_{32} - \bar{t}^u_{32} = \frac{[d_{32}.(\Delta_{31} - \bar{\Delta}_{31} + y_{32} - \bar{y}_{32})] - \sum_{i=2}^{31}(\alpha_i - \bar{\alpha}_i + y_i - \bar{y}_i).d_i - (\alpha_1 - \bar{\alpha}_1).d_1 + \lambda - \bar{\lambda}}{2}$$

$$t^l_{32} - \bar{t}^l_{32} = \frac{[d_{32}.(\gamma_{32} - \bar{\gamma}_{32}) + d_{31}.(\gamma_{31} - \bar{\gamma}_{31}) + ... + d_1.(\gamma_1 - \bar{\gamma}_1)] + (\lambda - \bar{\lambda})}{2}$$

1.4 Dimensionality of COCO PUF

Dimensionality for both **Response0** & **Response1** will be **63**, which is the same as that for arrival times for the upper signal in a simple arbiter PUF.

1.5 Code

Here is the link to the python file: sumit.py

1.6 Outcomes experiments with both LinearSVC and LogisticRegression methods

Performance Evaluation of LinearSVC with Various Loss Functions

Metric	Hinge	Squared Hinge
Total Features	478	478
Training Time (s)	1.5230	1.6543
Mapping Time (s)	0.07943	0.07156
Test Accuracy	0.9822	0.9944

Table 1: Performance Comparison of LinearSVC with Different Loss Functions

Analysis: The table illustrates that the model using squared hinge loss achieves slightly higher test accuracy compared to hinge loss. However, the difference is minimal. The squared hinge loss model requires slightly more training time, suggesting a potentially more complex optimization process.

Potential Causes for Observed Differences:

- Squared hinge loss may penalize outliers more, potentially enhancing generalization.
- Optimization with squared hinge loss might require more iterations, increasing training time.
- Dataset characteristics and challenges could favor one loss function over the other.

Performance Comparison Based on C Values

LinearSVC with Different C Values

Insights for LinearSVC:

- Medium C value achieves the highest test accuracy.
- Lower C value results in longer training time, possibly due to more complex optimization.
- Higher C value shows similar test accuracy with reduced training time.

C Value	Low	Medium	High
Total Features	478	478	478
Training Time (s)	1.8110	1.8451	1.7107
Mapping Time (s)	0.0738	0.0765	0.0769
Test Accuracy	0.9789	0.9934	0.9689

Table 2: Performance Comparison of LinearSVC with Different C Values

C Value	Low	Medium	High
Total Features	478	478	478
Training Time (s)	1.4897	1.5760	1.7872
Mapping Time (s)	0.0840	0.0837	0.0891
Test Accuracy	0.9761	0.9944	0.9954

Table 3: Performance Comparison of Logistic Regression with Different C Values

Logistic Regression with Different C Values

Insights for Logistic Regression:

- Medium C value achieves the highest test accuracy.
- Training times are significantly lower compared to LinearSVC across C values.
- High C value shows the highest test accuracy, indicating stronger regularization.

Effect of Tolerance Variation on Models

Model	Tolerance	Total Features	Training Time (s)	Mapping Time (s)	Test Accuracy
LinearSVC	Low	478	6.802	0.0770	0.9901
LinearSVC	Medium	478	5.640	0.0789	0.9805
LinearSVC	High	478	1.069	0.0951	0.9289
Logistic Regression	Low	478	1.089	0.1020	0.9908
Logistic Regression	Medium	478	1.260	0.1090	0.9906
Logistic Regression	High	478	1.050	0.1190	0.9878

Table 4: Effect of Variation in Tolerance on Models

Observations:

- Lower tolerance in LinearSVC leads to longer training times but better generalization.
- Tolerance variation has a smaller impact on Logistic Regression's training time.
- Both models show reduced test accuracy with higher tolerance, suggesting poorer generalization.

Conclusion

Based on the analysis, Logistic Regression with a medium C value (0.9944 accuracy, 1.746s training time, 0.0738s mapping time) balances accuracy and computational efficiency effectively.