CS771 - Assignment 1

Authors
Neerav Sethi
Kunaal Gautam
Aryan Agarwal
Saqib
Shrish Shete
Mutasim Khan

1 Part I

Objective: To demonstrate the existence of a linear model capable of accurately predicting responses for CAR PUFs.

Let Δw and Δr be the time delays between the output signals for the 1^{st} PUF and the 2^{nd} PUF respectively. Then we know that

$$\Delta w = w_w^T X + b_w$$
$$\Delta r = w_r^T X + b_r$$

Where X is a 32-dimensional vector with $x_i = d_i \cdot d_{i+1} \cdot ... \cdot d_{31}$ and $d_i = 1 - 2c_i$ (c_i being the $(i+1)^{th}$ bit).

The Problem tells us:

$$[\Delta w - \Delta \tau] > \tau$$

for the response to be 1. ($\tau > 0$ is the secret threshold value of the CAR PUF).

Therefore:

$$[(w_w - w_r)^T X - (b_w - b_r)]^2 - \tau^2 > 0$$

Which, on simplificaton yeilds:

$$[(W^T X + B)^2 - \tau^2] > 0$$

Where:

$$W = w_w - w_r$$
$$B = b_w - b_r$$

Now,

$$\begin{aligned}(W^T X + B)^2 &= (W^T X + B)(W^T X + B) \\ &= (W^T X)^2 + 2(W^T X)B + B^2\end{aligned}$$

Therefore,

$$W = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{31} \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{31} \end{bmatrix}$$

Where, $W^T X = \sum_{i=0}^{31} \alpha_i x_i$.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

So,

$$\left(\sum_{i=0}^{31}\alpha_i x_i\right)^2 = \sum_{i=0}^{31}\alpha_i^2 x_i^2 + 2\sum_{i=0}^{31}\sum_{\substack{j=0\\j<i}}^{31}\alpha_i\alpha_j x_i x_j$$

The equation then takes the form:

$$\sum_{i=0}^{31}\alpha_i^2 + 2\sum_{i=0}^{31}\sum_{\substack{j=0\\j<i}}^{31}\alpha_i\alpha_j x_i x_j + 2B\sum_{i=0}^{31}\alpha_i x_i + (B^2-\tau^2) > 0$$

Since $x_i \epsilon \{-1,1\}$ the value of x_i^2 will be 1 for all values of i

Therefore, the response condition becomes:

$$[\mathbf{W^T X} + \mathbf{B}] > 0$$

Where:

$$\mathbf{B} = B^2 - \tau^2 + \sum_{i=0}^{31}\alpha_i^2 \quad , \quad \mathbf{X} = \begin{bmatrix} x_0x_1 \\ x_0x_2 \\ \vdots \\ x_{30}x_{31} \\ x_0 \\ x_1 \\ \vdots \\ x_{31} \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} \alpha_0\alpha_1 \\ \alpha_0\alpha_2 \\ \vdots \\ \alpha_{30}\alpha_{31} \\ 2B\alpha_0 \\ 2B\alpha_1 \\ \vdots \\ 2B\alpha_{31} \end{bmatrix}$$

$\mathbf{X}$ has dimension:

$$32 + \binom{32}{2} = 528$$

The map $\omega(X)$ will be a map from R^{32} to R^{528} (will map X to $\mathbf{X}$)

We can represent X as $G(c)$ where G is a map from R^{32} to R^{32} (as $x_i = d_i \cdot d_{i+1} \cdot ... \cdot d_{31}$ where $d_i = 1 - 2c_i$)

Therefore we have a map $F(c)$ which can be defined to be $\omega \circ G(c)$ which is contingent on the challenges but not on any constant terms

Hence, the linear model to break this CAR-PUF can be given using the linear model $\mathbf{W}^\top \boldsymbol{X} + \mathbf{B}$, where given a challenge c, we can map it to $F(c)$, and the response to this challenge can be given by:

$$\frac{1 + \text{sign}(\mathbf{W}^\top \boldsymbol{\phi}(c) + B)}{2}$$

Hence Proved

2 Part III

Report outcomes of experiments with both the sklearn.svm.LinearSVC and sklearn.linear model.Logistic regression 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 Logistic Regression 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 the following affect training time and test accuracy:

Effect of Tolerance - Based on our experimentation with these different hyperparameters, a very definite inference that can be made is that there is an extremely marginal effect on accuracy by

Table 1: Setting C in LinearSVC and LogisticRegression to high/low/medium values (tol = 0.0001)

Model	Training Time (s)	Mapping Time (s)	Accuracy
Linear SVC (small C, C = 0.01)	6.122	0.403	97.3
Linear SVC (Medium C, C = 1)	9.820	0.428	98.87
Linear SVC (Optimal C, C = 10)	12.820	0.432	99.21
Linear SVC (High C, C = 100)	60.192	0.401	96.2
LogisticRegression (small C, C = 0.01)	2.206	0.410	98.1
LogisticRegression (Medium C, C = 1)	2.919	0.438	99.22
LogisticRegression (High C/ Optimal C, C = 100)	5.086	0.393	99.30

Table 2: Changing tol in LinearSVC and LogisticRegression to high/low/medium values (C=10 for LinearSVC and C =100 for LogisticRegression

Model	Training Time (s)	Mapping Time (s)	Accuracy
Linear SVC (high tol, tol = 1e-2)	9.89	0.45	99.29
Linear SVC (Medium tol, tol = 1e-4)	15.62	0.40	99.28
Linear SVC (small tol, tol = 1e-5)	55.99	0.51	99.28
LogisticRegression (high tol, tol = 1e-2)	3.74	0.42	99.32
LogisticRegression (Medium tol, tol = 1e-4)	4.57	0.43	99.30
LogisticRegression (small tol, tol = 1e-5)	5.24	0.49	99.30

increasing tolerance. The reason for that can be attributed to Logistic Regression itself, which due to its simple implementation of binary classification on data which is linearly separable, results in a quick convergence in most cases. This of course is relative and the marginal effect referred above, is observed only in very high values of tolerance. The same is not observed in LinearSVC since in that case, the tolerance value acts as a influential constraint for the optimisation process. So, increasing its value would results in the said process to terminate early, potentially causing it to stop before it has converged optimally. Hence, the model may not yield the best possible decision boundary, resulting in lower accuracy. There's also an added effect of weakening the regularisation effect, which may cause overfitting on the unseen data, leading to lower accuracy.

Effect of C - In evaluating both the models, we observed that there is a direct relation between the regularization parameter, C, and training time - increasing one leads to increase in the other.Test accuracy initially increased with rising C values, reaching a peak at C = 10 for LinearSVC and C = 100 for Logistic Regression. Once we hit these saturation points, any further increment in the parameter had very marginal changes in the accuracy, which implies that there exists an optimal balance between the test accuracy and training time, and the above two represents the peak performance for the respective models.