Home Assignment 3
Machine Learning A

FEDERICO FIORIO

September 26, 2022

1 How to Split a Sample into Training and Test Sets (20 points)

1.1
For simplicity, $\hat{h}^*_{S^{train}} = h$
Since S^{train} and S^{test} come from the same dataset S, we can say that the samples are i.i.d.
Consider X and Y taken from the S^{test}:
$l(h(X_i), Y_i)$ for each i in n^{test}, has the same distribution as $l(h(X), Y)$ for any sample in S^{test}.
$\hat{L}(h, S^{test}) = \frac{1}{n^{test}} \sum_{i=1}^{n^{test}} Z_i$
where $Z_i = l(h(X_i), Y_i)$
so $E[Z_i] = E[l(h(X), Y)] = L(h)$
Since $\frac{1}{n^{test}} \sum_{i=1}^{n^{test}} Z_i$ is the average of n^{test} i.i.d. random variables with $E[Z_i] = L(h)$ we can apply Hoeffding's bound, but we have to assume that l has values between [0,1] for all predictions Y' and all test values Y:

$$P(L(h) - \frac{1}{n} \sum_{i=1}^{n^{test}} Z_i \geq \epsilon) \leq e^{-2n\epsilon^2}$$

if we take $\epsilon = \sqrt{\frac{\ln \frac{1}{\delta}}{2n^{test}}}$

$$P(L(h) \geq \hat{L}(h, S^{test}) + \sqrt{\frac{\ln \frac{1}{\delta}}{2n^{test}}}) \leq \delta$$

which becomes :

$$P(L(h) \leq \hat{L}(h, S^{test}) + \sqrt{\frac{\ln \frac{1}{\delta}}{2n^{test}}}) \geq 1 - \delta$$

2 Confidence Intervals for Bernoulli Distribution (15 points)

2.1
$(\mu - \sqrt{\frac{\ln \frac{2}{\delta}}{2n}}, \mu + \sqrt{\frac{\ln \frac{2}{\delta}}{2n}})$
=
(0.5594615006515282, 0.6820411827116202)

procedure:
Based on the definition of CI:
I considered $X_1, ..., X_n$ samples from a particular distribution, in this case the Bernoulli distribution.
I then used as parameter of the distribution the mean μ
A C.I. for μ, in my case, is a function:
$CI(X_1, ..., X_n, \delta)$
$P(\mu \in CI) \geq 1 - \delta$

I know that the samples are bounded by [0,1] because they belong to a Bernoulli distribution, so I know that I can apply Hoeffding's inequality to be able to derive the CI for this Bernoulli distribution based on the parameter μ

$$C.I. = (\mu - \sqrt{\frac{\ln \frac{2}{\delta}}{2n}}, \mu + \sqrt{\frac{\ln \frac{2}{\delta}}{2n}})$$

and since μ belongs to [0,1] we can say:

$$C.I. = (max(\mu - \sqrt{\frac{\ln \frac{2}{\delta}}{2n}}, 0), min(\mu + \sqrt{\frac{\ln \frac{2}{\delta}}{2n}}, 1))$$

2.2

We know that with confidence at least 0.98 the C.I. for λ is (0.7374445305772481, 0.8897594996494522)
At the same time, I know that μ will be in the C.I. (0.5266332358430408, 0.7148694475201076) with confidence 0.9999.
Since the max value of μ with confidence $\geq$ 98 is 0.714 and the min value for λ is 0.737 with confidence $\geq$ 98 we can say that overall, we are confident with confidence $\geq$ 98 that $\lambda > \mu$

I can still apply Hoeffding because in U, samples are independent, same for S, they have the same bounds [0,1] so I can compare the C.I.s even without the independence of U and S.

3 Big and Small (25 points)

1
I'm trying to figure out which is the best h to choose, I have to consider 2 different training sets and 2 different test set, I cannot choose the h based on the error, because this won't tell me which h generalize better over the expected loss $\hat{L}(h)$
Since both models are trained over the same distribution of data, their expected loss should be the same.
In our case $\hat{L}(h_a, s_b) = 0.03$ has been trained on 9000 samples and tested on 1000, and $\hat{L}(h_b, s_a) = 0.06$ has been trained on 1000 samples and tested on 9000.

$$P(L(h_a) \geq \hat{L}(h_a, s_b) + \sqrt{\frac{\ln \frac{1}{\delta}}{2n_b}}) \leq \delta$$

$$P(L(h_b) \geq \hat{L}(h_b, s_a) + \sqrt{\frac{\ln \frac{1}{\delta}}{2n_a}}) \leq \delta$$

using a union bound, P(A U B) $\leq P(A) + P(B)$, $\delta' = \frac{\delta}{2}$, because we have 2 hypothesis:

$H = \{h_a, h_b\}$
$P(\exists h \in H : L(h) \geq \tilde{L}(h, s) + \sqrt{\frac{\ln \frac{M}{\delta}}{2n}}) \leq \delta$
which becomes:
$P(L(h_a) \geq \tilde{L}(h_a, s_b) + \sqrt{\frac{\ln \frac{2}{\delta'}}{2n_b}}) + P(L(h_b) \geq \tilde{L}(h_b, s_a) + \sqrt{\frac{\ln \frac{2}{\delta'}}{2n_a}}) \leq \delta'$
$P(L(h_a) \leq \tilde{L}(h_a, s_b) + \sqrt{\frac{\ln \frac{2}{\delta'}}{2n_b}}) + P(L(h_b) \leq \tilde{L}(h_b, s_a) + \sqrt{\frac{\ln \frac{2}{\delta'}}{2n_a}}) \geq 1 - \delta'$
if I pick $\delta' = 0.05$ we get the bound for prob 95%.

For which classifier to pick, I have to derive a lower and upper bound for both hypothesis in order to understand which hyp. it's the worst.

given $\delta = 0.05$, n = 1000 and $\hat{L}(h_a, s_b) = 0.03$:
$P(L(h_a) \leq 0.07) \geq 0.95$
$P(L(h_a) \geq 0.00) \geq 0.95$

given $\delta = 0.05$, n = 9000 and $\hat{L}(h_b, s_a) = 0.06$:
upper bound:
$P(L(h_b) \leq 0.074) \geq 0.95$
lower bound:

$P(L(h_b) \geq 0.045) \geq 0.95$

So as final answer, I would take the classifier A, trained on 9000 samples and tested on 1000, because its interval of errors it's more promising

2.
I would keep the same reasoning as before, and this time, the choice is harder, if we want to greed we can go for classifier A which oscillates more, and for B if we want to stay more safe.

given $\delta = 0.01$, n = 1000 and $\hat{L}(h_a, s_b) = 0.03$:
$P(L(h_a) \leq 0.081) \geq 0.99$
$P(L(h_a) \geq .0.00) \geq 0.99$

given $\delta = 0.01$, n = 9000 and $\hat{L}(h_b, s_a) = 0.06$:
$P(L(h_b) \leq 0.077) \geq 0.99$
$P(L(h_b) \geq 0.042) \geq 0.99$

4 Preprocessing (20 points)

4.1 9.1 (4 points)

Mr. Good = (37, 45000), Mr. bad = (22, 40000), Mr. unknown = (21, 36000).
In order to compute K-nn algorithm we need to defien the distance measure to use and the number of k nearest neighbours.
Since we have only 2 examples I will assume that k = 1 , the distance measure that I will use it's the common Euclidean distance: $\sqrt{(37-21)^2 + (45000-36000)^2} = 57611.73$ and $\sqrt{(22-21)^2 + (40000-36000)^2} = 53798.892$
This means that based on K-nn Mr. unknown should be classified as bad.

4.2 9.2 (4 points)

$Z = (I - \frac{1}{N}11^T)X =$
$X - \frac{1}{N}11^T X$
$\frac{1}{N}11^T X$ = matrix where each row is a vector and each component of this vector is the mean of the columns X.
to clarify, given matrix X:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$\frac{1}{N}11^T X =$

$$\begin{bmatrix} (1+4+7)/N & (2+5+8)/N & (3+6+9)/N \\ (1+4+7)/N & (2+5+8)/N & (3+6+9)/N \\ (1+4+7)/N & (2+5+8)/N & (3+6+9)/N \end{bmatrix}$$

so $Z = X - \frac{1}{N}\mathbf{1}\mathbf{1}^T X$ is the same as $Z = X - \mathbf{1}\tilde{x}^T$ because $\frac{1}{N}\mathbf{1}\mathbf{1}^T X = \mathbf{1}\tilde{x}^T$
$\mathbf{1}[\frac{1}{N}X^T\mathbf{1}]^T =$
$\mathbf{1}[\frac{1}{N}\mathbf{1}^T X] =$
$\frac{1}{N}\mathbf{1}\mathbf{1}^T X$

4.3 9.4 (12 points)

a)
variance$(\tilde{x_1}) = 1$
variance$(\tilde{x_2}) = var(\sqrt{(1-\epsilon)}\tilde{x_1} + \epsilon\tilde{x_2}) =$
$var(\sqrt{1-\epsilon^2}\tilde{x_1}) + var(\epsilon\tilde{x_2}) =$
$1 - \epsilon^2 + \epsilon^2 =$

covariance$(x_1, x_2) =$
A $= \begin{bmatrix} 1 & 0 \\ \sqrt{1-\epsilon^2} & \epsilon \end{bmatrix}$
$A^T = \begin{bmatrix} 1 & \sqrt{1-\epsilon^2} \\ 0 & \epsilon \end{bmatrix}$
$cov(x_1, x_2) = \begin{bmatrix} 1 & \sqrt{1-\epsilon^2} \\ \sqrt{1-\epsilon^2} & 1 \end{bmatrix}$
b)

f(x) $= w_1\tilde{x_1} + w_2(\sqrt{1-\epsilon^2}\tilde{x_1} + \epsilon\tilde{x_2}) =$

$\tilde{x_1}(w_1 + w_2\sqrt{1-\epsilon^2}) + w_2\epsilon\tilde{x_2}$ on which I get:

$w_2 = \frac{\tilde{w_2}}{\epsilon}$

$w_1 = \tilde{w_1} - \frac{\tilde{w_2}}{\epsilon}\sqrt{1-\epsilon^2}$
c)
I should be able to solve this optimization constrained problem with the Lagrangian method, so that we obtain the minimized function $w_1^2 + w_2^2$

d) C increases

5 Variable importance (20 points)

How many solutions (i.e., optimal values for the coefficients) would the linear regression optimization problem (without regularization) have if the one-hot encoding was used? Why? Why would it be difficult to interpret the variable importance if the one-hot encoding was used?

In case we want to use one-hot encoding for categorical data for our regression model, we would obtain k dummy variables where k is the number of categorical features.
For example, we have $apple, banana$ and we use one-hot encoding to get a usable representation for our model, we would need 2 dummy variables for the one-hot encoding representation.
The problem with using k as number of coefficients is that we will end up in the so called: "dummy variable trap".
The dummy variable trap happens when we have created dummy variables for our one-hot encoding representation but some of these variables are actually correlated/dependent one another.
If the variables are highly correlated we end up in a situation called multicollinearity.
In a linear model we assume to be able to change the values of a given coefficient without changing the values of the other variables.
However, when two or more variables are highly correlated, it becomes difficult to change one variable without changing another.
This makes it difficult for the regression model variables to be estimated correctly because of the correlation that the coefficients have.
So we prefer to use k-1 values for the one-hot encoding representation of the labeled data, in this way we avoid the correlation problem and we can understand the weights of the model correctly. To recap: When using one-hot encoding to handle categorical data then one dummy variable (attribute) can be predicted with the help of other dummy variables. This means that the variables are highly-correlated, this lead us to the dummy variable trap where the coefficients of our regression model are correlated and not really interpretable on the scale of how important they are because the z-statistic would be uninterpretable.

6 [Optional, Not for Hand-in] Distribution of Student's Grades