Machine Learning 2019-2020
Final Exam

EXAM ID 273

Contents

1 Logistic Regression

We are given that input spaces is $\mathbb{R}^d$ and label space is {0,1}.
and following is our model

$$f(x) = \sigma(w^T x + b) = P(Y = 1|X = x)$$

and

$$w^T x + b = ln\left(\frac{P(Y = 1|X = x)}{P(Y = 0|X = x)}\right)$$

we need to show that σ is a logistic function, that is

$$\sigma = \frac{1}{1 + e^{-(w^T x + b)}}$$

As label space is {0,1}, we know that $P(Y = 1|X = x) + P(Y = 0|X = x) = 1$.

$$\begin{aligned}
w^T x + b &= ln\left(\frac{P(Y = 1|X = x)}{P(Y = 0|X = x)}\right) \\
e^{w^T x + b} &= \left(\frac{P(Y = 1|X = x)}{P(Y = 0|X = x)}\right) \\
e^{-(w^T x + b)} &= \frac{P(Y = 0|X = x)}{P(Y = 1|X = x)} \\
1 + e^{-(w^T x + b)} &= 1 + \frac{P(Y = 0|X = x)}{P(Y = 1|X = x)} \\
1 + e^{-(w^T x + b)} &= \frac{P(Y = 0|X = x) + P(Y = 1|X = x)}{P(Y = 1|X = x)} \\
1 + e^{-(w^T x + b)} &= \frac{1}{P(Y = 1|X = x)} \\
\frac{1}{1 + e^{-(w^T x + b)}} &= P(Y = 1|X = x) \\
&= \sigma
\end{aligned}$$

Hence, Proved.

2 Linear Decision Boundaries from Logistic Regression and SVMs

2.1 Logistic regression model

We know that S_1 has same data points as $S_2\dot{S}$o, S_1 and S_1 both trains on same set of data points. So, we would have same loss function and gradient. Model for S_1 would also be optimal for model on $S_2\dot{T}$he decision boundary would also be optimal.

2.2 SVM with linear kernel and offset parameter

Not Attempted

3 Kernels

$$k(x,z) = (1-\gamma)e^{-||[x]_{1:d}-[z]_{1:d}||^2} + \gamma < [x]_{d+1:2d}, [z]_{d+1:2d} > \tag{1}$$

We need to show that k(x,z) is positive definite kernel.

We know from lecture notes that below functions are positive definite kernels when $K_1, K_2 : \chi \times \chi \to \mathbb{R}$ are positive definite kernels and $a \in \mathbb{R}^+, f : \chi \to \mathbb{R}$

$$k(x,z) = ak_1(x,z) \tag{2}$$
$$k(x,z) = k_1(x,z) + k_2(x,z) \tag{3}$$
$$k(x,z) = k_1(x,z)k_2(x,z) \tag{4}$$
$$k(x,z) = e^{k_1(x,z)} \tag{5}$$
$$k(x,z) = f(x)f(z) \tag{6}$$

Let $k_1(x,z) = e^{-||[x]_{1:d}-[z]_{1:d}||^2}$, $k_2(x,z) =< [x]_{d+1:2d}, [z]_{d+1:2d} >$ be positive definite kernels.

Then by using above equations (2) and (3), we can say that $(1-\gamma)(k_1(x,z) + \gamma(k_2(x,z)$ is also positive definite kernel.
So, in order to prove prove equation (1) we need to show that $k_1(x,z)$ and $k_2(x,z)$ are positive definite kernel.
First, let us prove that $k_2(x,z)$ is positive definite kernel.
For positive definite kernel k, $\forall c_1, ..., c_m \in \mathrm{R} : \sum_{i,j=1}^{m} c_i c_j K_{i,j} \geq 0$
$K_{i,j} = k(x_i, z_j)$

$k_2(x,z)$ is a dot product kernel ,so $\exists\Phi$ and a dot product such that $k_2(x,z) =< \Phi(x), \Phi(z) >$ then $\forall c \in \mathbb{R}$

$$
\begin{aligned}
\sum_{i,j=d+1}^{2d} c_i c_j K_2(x_i, x_j) &= \left\langle \sum_{i,j=d+1}^{2d} c_i c_j < \Phi(x_i), \Phi(x_j) > \right\rangle \\
&= \left\langle \sum_{i=d+1}^{2d} c_i \Phi(x_i), \sum_{j=d+1}^{2d} c_j \Phi(x_j) \right\rangle \\
&= || \sum_{i=d+1}^{2d} c_i \Phi(x_i) ||^2 \geq 0
\end{aligned}
$$

Hence, $k_2(x, z)$ is positive definite kernel.

Now, we need to show $k_1(x, z)$ is positive definite kernel.

$$
\begin{aligned}
k_1(x, z) &= e^{-||x-z||^2} \\
&= e^{-||x||^2} e^{-||z||^2} e^{2<x,z>}
\end{aligned}
$$

Let $k_3(x, z) = e^{-||x||^2} e^{-||z||^2}$
and $k_4(x, z) = e^{2<x,z>}$
then $k_1(x, z) = k_3(x, z) k_4(x, z)$
let $f : \chi \to \mathbb{R} = e^{-||x||^2}$, then according to equation (6), $k_3(x, z)$ will be a positive definite kernel.
We know that dot product kernel is positive definite kernel as shown above. And then by using equations (2) and (5), we can say that $k_4(x, z)$ is also positive definite kernel.
By using equation (4) we can say that $k_1(x, z)$ is positive definite kernel as $k_3(x, z)$ and $k_4(x, z)$ are positive definite kernel.
Hence, proved.

4 Sleep Staging

4.1 Data understanding and preprocessing

Find the python code in `code.zip`
Following are the Classes and their corresponding frequencies for training data.

Class	Frequencies
0	17566
1	3221
2	8523
3	1583
4	2831

Table 1: Training Data - Class and their frequencies

4.2 Principal component analysis

I have used skicit-learn python library for PCA. Find the code in `code.zip`

4.2.1 Plot of Eigenspectrum

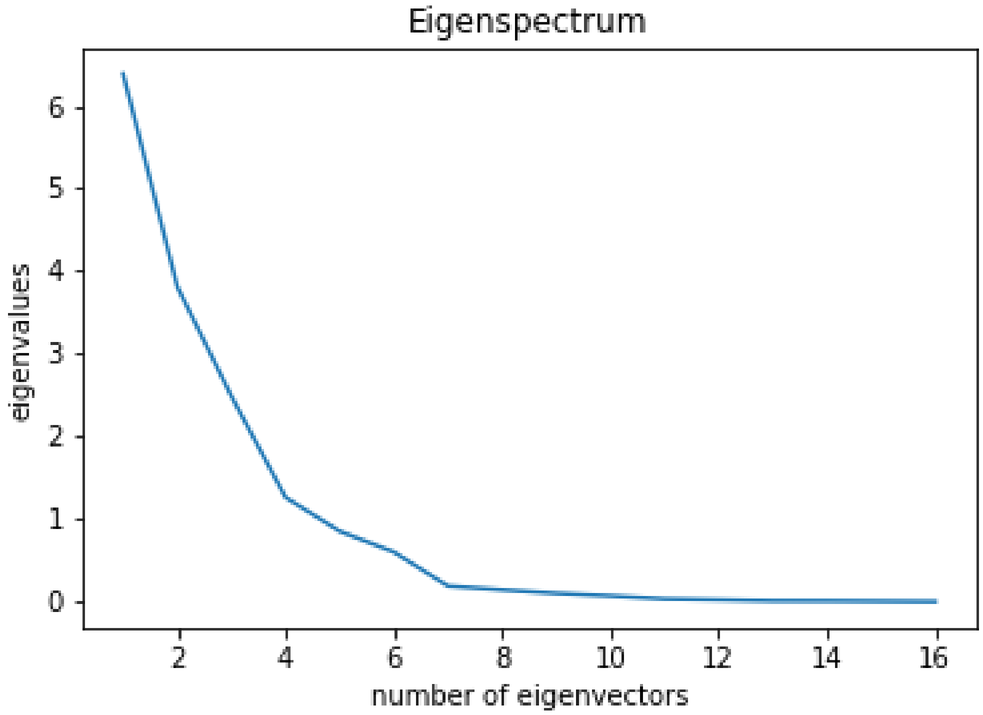

Figure 1: Plot of Eigenspectrum

4.2.2 Number of Components necessary to explain 90% variance

We can see in the following table that we need at 5 number of principal components to explain 90% variance.

Number of Principal Components	Cumulative Variances(%)
1	39.94
2	63.76
3	79.29
4	87.18
5	92.51
6	96.28
7	97.47
8	98.37
9	99.01
10	99.45
11	99.66
12	99.81
13	99.88
14	99.95
15	99.99
16	100.0

Table 2: Cumulative Variances

4.2.3 Scatter plot of the data projected on the first two principal components

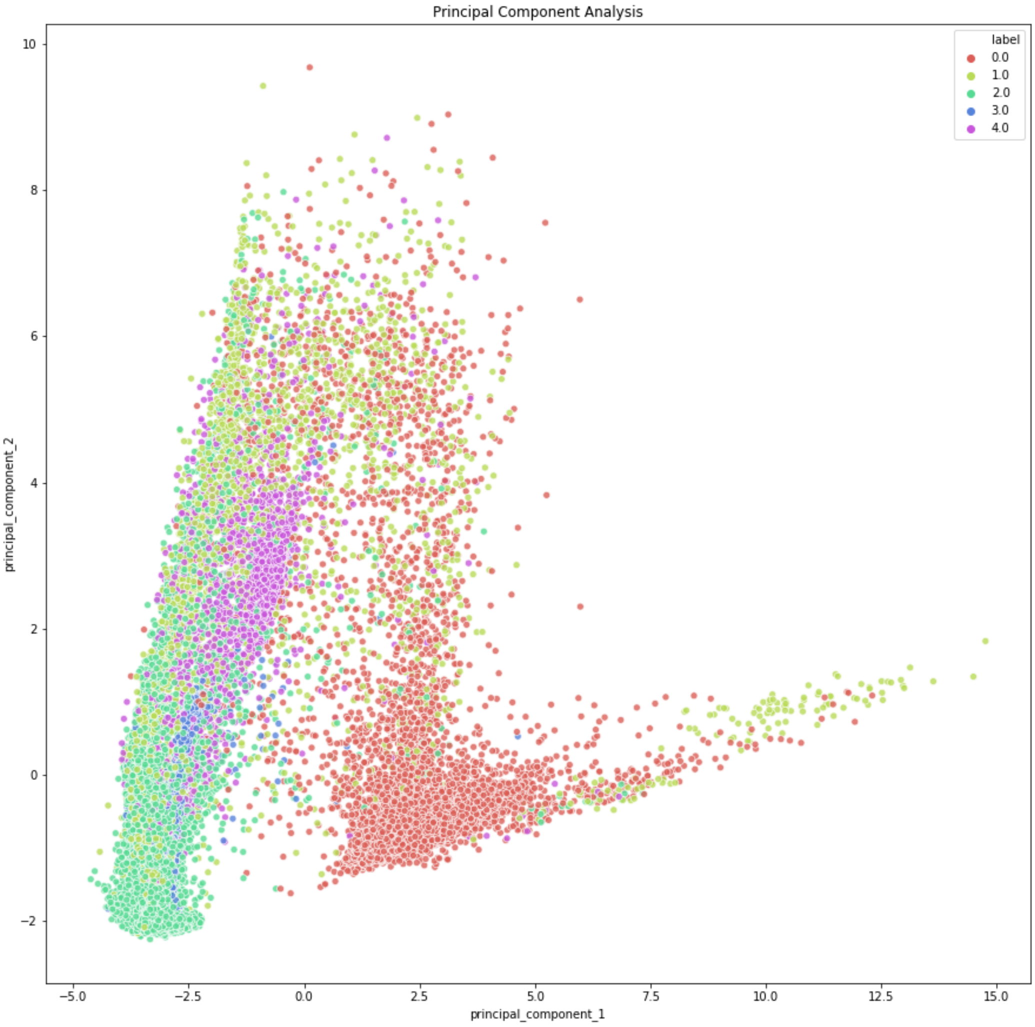

Figure 2: Scatter plot of the data projected on the first two principal components

4.3 Clustering

I have used skicit-learn python library for 5-means clustering. Find the code in `code.zip` We can see from the below plot, that some significant variation is explained by the principal components as there is some structure in the points projected along the two principal components. Points belonging to different classes are also overlapping.The cluster centers can be used to see the partition in the data.The similar data points are near to the cluster centers as much as possible and different as far as possible.

4.3.1 Plot of Cluster centers and data points

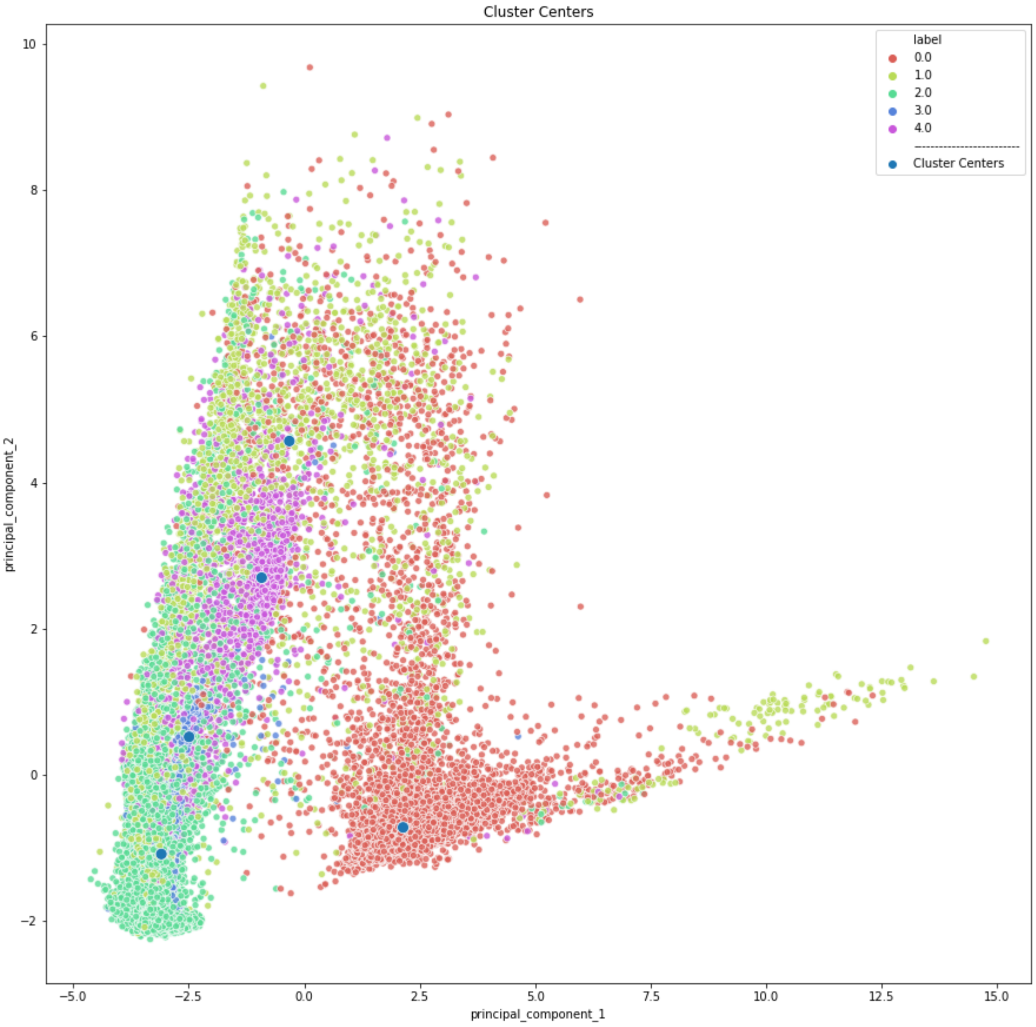

Figure 3: Plot of Cluster centers and data points

4.4 Classification

I have used skicit-learn python library for the following classifications. Find the code in `code.zip`

4.4.1 Multi-nominal Logistic Regression

It uses L2 regularization.

```
LogisticRegression(multi_class='multinomial', solver='lbfgs' ,max_iter
    =500)
```

Test Loss : 0.099
Training Loss : 0.149

4.4.2 Random Forests

```
RandomForestClassifier(n_estimators=i)
```

1. For 50 trees

   Test Loss : 0.113
   Training Loss : 0.0002

2. For 100 trees

   Test Loss : 0.111
   Training Loss : 2.96

3. For 200 trees

   Test Loss : 0.111
   Training Loss : 0.0

4.4.3 k-nearest-neighbor Classification

```
grid_params = {'n_neighbors': [i for i in range(3,60,2)]}
grid = GridSearchCV(KNeighborsClassifier(), grid_params, cv=5, n_jobs=-1)
```

I used cross validation to determine the number of neighbors. I used GridSearchCV and passed a list of odd values $[3, 5, ..., 59]$ for possible number of neighbors. As in case of odd number there would be no ties.
And then we could get the best parameters,i.e the number of neigbors from the our model generated by GridSearch.

Number of Neighbors : 51
Test Loss : 0.097
Training Loss : 0.145

5 Cross-Testing

5.1 Generalization bound for expected test loss of h^*

According to Theorem 3.2 in lecture notes,

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h,s) + \sqrt{\frac{\ln \frac{M}{\delta}}{2n}}\right) \leq \delta \tag{7}$$

where $\mathcal{H}$ is a finite set of hypothesis with$|\mathcal{H}| = M$.

Using the above equation (7), we can derive the generalization bound for expected test loss.

Let h^{1^*} be the model produced by the first student for 5-fold cross-testing and$\hat{L}(h^{1^*})$ its test loss and,
Let h^{2^*} be the model produced by the second student for 10-fold cross-testing and$\hat{L}(h^{2^*})$ its test loss

5.1.1 Case $h^* = h^{1^*}$

We have used 5-fold cross-testing, so we have 5 splits of data. here, M = 5 we would have 5 number of hypothesis model Using (7), we can say that

$$\mathbb{P}\left(\exists h^{1^*} \in \mathcal{H} : L(h^{1^*}) \geq \hat{L}(h^{1^*}, S_{test}) + \sqrt{\frac{\ln \frac{5}{\delta}}{2n_{test}}}\right) \leq \delta$$

we can also say that with probability at least $1 - \delta$ for all $h \in \mathcal{H}$

$$L(h^{1^*}) \leq \hat{L}(h^{1^*}, S_{test}) + \sqrt{\frac{\ln \frac{5}{\delta}}{2n_{test}}} \tag{8}$$

5.1.2 Case $h^* = h^{2^*}$

We have used 10-fold cross-testing, so we have 10 splits of data. here, M = 5 Using (7), we can say that

$$\mathbb{P}\left(\exists h^{2^*} \in \mathcal{H} : L(h^{2^*}) \geq \hat{L}(h^{2^*}, S_{test}) + \sqrt{\frac{\ln \frac{10}{\delta}}{2n_{test}}}\right) \leq \delta$$

we can also say that with probability at least $1 - \delta$ for all $h \in \mathcal{H}$

$$L(h^{2^*}) \leq \hat{L}(h^{2^*}, S_{test}) + \sqrt{\frac{\ln \frac{10}{\delta}}{2n_{test}}} \tag{9}$$

5.2 Select among h^{1^*} and h^{2^*}

$\hat{L}(h^{1^*}) = 0.1$
$\hat{L}(h^{2^*}) = 0.05$
$n = 10000$
$n_{test} = 10000/5 = 2000, for 5 - fold$
$n_{test} = 10000/10 = 1000, for 10 - fold$
$\delta = 0.01$

5.2.1 Case $h^* = h^{1^*}$

substituting values in equation (8), we get the following bound for probability atleast 0.99

$$\begin{aligned} L(h^{1^*}) &\leq 0.1 + \sqrt{\frac{\ln \frac{5}{0.01}}{2 * 2000}} \\ &\leq 0.139 \end{aligned}$$

5.2.2 Case $h^* = h^{2^*}$

substituting values in equation (9), we get the following bound for probability atleast 0.99

$$\begin{aligned} L(h^{2^*}) &\leq 0.05 + \sqrt{\frac{\ln \frac{10}{0.01}}{2 * 1000}} \\ &\leq 0.109 \end{aligned}$$

As Case $h^* = h^{2^*}$ has smaller test loss than Case $h^* = h^{1^*}$, so we will select h^{2^*} model.

6 Early Stopping

6.1 Is $\hat{L}(h_{t^*}, S_{val})$ an unbiased estimate of $L(h_{t^*})$

6.1.1 Predefined stopping

Yes, $\hat{L}(h_{t^*}, S_{val})$ is an unbiased estimate of $L(h_{t^*})$.
It is because new samples from the perspective of h are no way different from the samples in S. Any new sample could have happened to be in S instead of some other sample. They are exchangeable. Even if we exchange the samples, we would still get $h_{t^*} = h_{100}$.
$E[\hat{L}(h_{t^*}, S_{val})] = E[L(h_{t^*})]$.

6.1.2 Non-adaptive stopping

No, $\hat{L}(h_{t^*}, S_{val})$ is not an unbiased estimate of $L(h_{t^*})$.
The reason is that when we pick h_{t^*} that minimizes the error on S_{val}, from the perspective of h_{t^*} the samples in S_{val} no longer look identical to future samples. This is because h_{t^*} is selected in a very special way - it is selected to minimize the validation error on S_{val} and,

thus, it is tailored to S_{val} and most likely does better on S_{val} than on new random samples. $E[\hat{L}(h_{t^*}, S_{val})] \neq E[L(h_{t^*})]$.

6.1.3 Adaptive stopping

No, $\hat{L}(h_{t^*}, S_{val})$ is not an unbiased estimate of $L(h_{t^*})$.
The reason is that when we pick h_{t^*}, the best model observed during S_{val}, from the perspective of h_{t^*} the samples in S_{val} no longer look identical to future samples. This is because selection of h_{t^*} is based on S_{val} and, thus, it is tailored to S_{val} and most likely does better on S_{val} than on new random samples.
$E[\hat{L}(h_{t^*}, S_{val})] \neq E[L(h_{t^*})]$.

6.2 High probability bound on $L(h_{t^*})$

6.2.1 Predefined stopping

According to Theorem 3.1 in lecture notes, for a single hypothesis h

$$\mathbb{P}\left(L(h) \geq \hat{L}(h,s) + \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}\right) \leq \delta$$

we can also read the above theorem as, with probability $1-\delta$ we have

$$L(h) \leq \hat{L}(h,s) + \sqrt{\frac{\ln\frac{1}{\delta}}{2n}}$$

Using the above equation, we can derive the generalization bound on $L(h_{t^*})$ for Predefined Stopping in terms of $\hat{L}(h_{t^*}, S_{val}), \delta$ and the size n of the validation set S_{val}. Here, $h_{t^*} = h_{100}$

$$L(h_{t^*}) \leq \hat{L}(h_{t^*}, S_{val}) + \sqrt{\frac{\ln\frac{1}{\delta}}{2n}} \tag{10}$$

6.2.2 Non-adaptive stopping

According to Theorem 3.2 in lecture notes,

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h,s) + \sqrt{\frac{\ln\frac{M}{\delta}}{2n}}\right) \leq \delta$$

we can also read the above theorem as, with probability $1-\delta$ for all $h \in \mathcal{H}$ we have

$$L(h) \leq \hat{L}(h,s) + \sqrt{\frac{\ln\frac{M}{\delta}}{2n}}$$

where $\mathcal{H}$ is a finite set of hypothesis with$|\mathcal{H}| = M$.

Using the above equation , we can derive the generalization bound on $L(h_{t^*})$ for Predefined Stopping in terms of $\hat{L}(h_{t^*}, S_{val}), \delta$, the size n of the validation set S_{val} and the total number of epochs T. Here, M will be equal to T.

$$L(h_{t^*}) \leq \hat{L}(h_{t^*}, S_{val}) + \sqrt{\frac{\ln \frac{T}{\delta}}{2n}} \tag{11}$$

6.2.3 Adaptive stopping

According to Theorem 3.3 in lecture notes, where $\pi(h) \geq 0 \forall h$ and $\sum_{h \in \mathcal{H}} \pi(h) \leq 1$

$$\mathbb{P}\left(\exists h \in \mathcal{H} : L(h) \geq \hat{L}(h,s) + \sqrt{\frac{\ln \frac{1}{\pi(h)\delta}}{2n}}\right) \leq \delta$$

we can also read the above theorem as, with probability $1 - \delta$ for all $h \in \mathcal{H}$ we have

$$L(h) \leq \hat{L}(h,s) + \sqrt{\frac{\ln \frac{1}{\pi(h)\delta}}{2n}}$$

where $\mathcal{H}$ is a finite set of hypothesis.

Using the above equation , we can derive the generalization bound on $L(h_{t^*})$ for Predefined Stopping in terms of $\hat{L}(h_{t^*}, S_{val}), \delta$, the size n of the validation set S_{val} and using the series $\sum_{i=1}^{\infty} \frac{1}{i(i+1)} \leq 1$. We will use $\pi(h) = \frac{1}{t^*(t^*+1)}$

$$L(h_{t^*}) \leq \hat{L}(h_{t^*}, S_{val}) + \sqrt{\frac{\ln \frac{1}{\pi(h)\delta}}{2n}}$$

$$L(h_{t^*}) \leq \hat{L}(h_{t^*}, S_{val}) + \sqrt{\frac{\ln \frac{1}{\frac{1}{t^*(t^*+1)}\delta}}{2n}}$$

$$L(h_{t^*}) \leq \hat{L}(h_{t^*}, S_{val}) + \sqrt{\frac{\ln \frac{t^*(t^*+1)}{\delta}}{2n}} \tag{12}$$

6.3 Maximal numbers of epochs, T_{max}

Let maximal number of epochs be T_{max}.

we know that the loss is less than equal to 1. So, $\hat{L}(h_{t^*}, S_{val}) + \sqrt{\frac{\ln \frac{t^*(t^*+1)}{\delta}}{2n}} \leq 1$

$\hat{L}(h_{t^*}, S_{val})$ is bounded by 1. so, $\sqrt{\frac{\ln \frac{t^*(t^*+1)}{\delta}}{2n}}$ should also be bounded by 1. so, we can say that

$$\begin{aligned}
\sqrt{\frac{\ln\frac{t^*(t^*+1)}{\delta}}{2n}} &\leq 1 \\
\frac{\ln\frac{t^*(t^*+1)}{\delta}}{2n} &\leq 1 \\
\ln\frac{t^*(t^*+1)}{\delta} &\leq 2n \\
\frac{t^*(t^*+1)}{\delta} &\leq e^{2n} \\
t^*(t^*+1) &\leq \delta e^{2n} \\
(t^*)^2 &\leq \delta e^{2n}, (assuming, t^*+1 \approx t^*) \\
t^* &\leq \sqrt{\delta e^{2n}} \qquad (13)
\end{aligned}$$

we have T_{max} epochs, subsituting $t^* = T_{max}$, in above equation (13), we get

$$T_{max} \leq \sqrt{\delta e^{2n}}$$

6.4 Maximal numbers of epochs, T_{max}

As discussed above, we know $\sqrt{\frac{\ln\frac{1}{\delta}}{\pi(h)2n}}$ should also be bounded by 1.
Using the series $\sum_{i=1}^{\infty}\frac{1}{2^i} \leq 1$. We will use $\pi(h) = \frac{1}{2^{t^*}}$
so, we can say that

$$\begin{aligned}
\sqrt{\frac{\ln\frac{1}{\delta}}{\pi(h)2n}} &\leq 1 \\
\sqrt{\frac{\ln\frac{2^{t^*}}{\delta}}{2n}} &\leq 1 \\
\frac{\ln\frac{2^{t^*}}{\delta}}{2n} &\leq 1 \\
\ln\frac{2^{t^*}}{\delta} &\leq 2n \\
\frac{2^{t^*}}{\delta} &\leq e^{2n} \\
2^{t^*} &\leq \delta e^{2n} \\
t^* &\leq \log_2(\delta e^{2n}) \qquad (14)
\end{aligned}$$

we have T_{max} epochs, subsituting $t^* = T_{max}$, in above equation (14), we get

$$T_{max} \leq \log_2(\delta e^{2n})$$

6.5 Generalization bound for Adaptive stopping

Not Attempted