Introduction to Machine Learning **Liad Iluz**
Assignment 5 **208427435**

Theoretical Assignment

(1) (Suboptimality of ID3).

Consider the following training set S, where $X = \{0,1\}^3$ and $Y = \{0,1\}$:

$$\big((1,1,1),1\big)$$
$$\big((1,0,0),1\big)$$
$$\big((1,1,0),0\big)$$
$$\big((0,0,1),0\big)$$

Suppose we wish to use this training set in order to build a decision tree of depth 2 (i.e., for each input we are allowed to ask two questions of the form ($x_i = 0$?) before deciding on the label).

a. Suppose we run the $ID3$ algorithm up to depth 2 (namely, we pick the root node and its children according to the algorithm, but instead of keeping on with the recursion, we stop and pick leaves according to the majority label in each subtree). Assume that the subroutine used to measure the quality of each feature is based on the entropy function (so we measure the information gain), and that if two features get the same score, one of them is picked arbitrarily. Show that the training error of the resulting decision tree is at least $1/4$.

b. Find a decision tree of depth 2 that attains zero training error.

Solution:

a. Probabilities table:

	X_1	X_2	X_3	Y
0	0.25	0.5	0.5	0.5
1	0.75	0.5	0.5	0.5

Conditional probabilities: $\mathbb{P}_S[Y = 1|X_i = j]$

	$X_1 = 1$	$X_1 = 0$	$X_2 = 1$	$X_2 = 0$	$X_3 = 1$	$X_3 = 0$
$Y = 1$	2/3	0	0.5	0.5	0.5	0.5

Choosing the root: $\arg\max_{i\in\{1,2,3\}} G(S,i)$ where $G(S,i) = I(Y;X_i) = H[Y] - H[Y|X_i]$.

- $H[Y] = \mathbb{P}_S[Y=1]\log\frac{1}{\mathbb{P}_S[Y=1]} + \mathbb{P}_S[Y=0]\log\frac{1}{\mathbb{P}_S[Y=0]} = \log(2)$

$$\begin{aligned}H[Y|X_i] &= \mathbb{P}_S[X_i = 1]H[Y|X_i = 1] + \mathbb{P}_S[X_i = 0]H[Y|X_i = 0] = \\ &\mathbb{P}_S[X_i = 1]\left(\mathbb{P}_S[Y = 1|X_i = 1]\log\frac{1}{\mathbb{P}_S[Y = 1|X_i = 1]}\right. \\ &\qquad\left. + \mathbb{P}_S[Y = 0|X_i = 1]\log\frac{1}{\mathbb{P}_S[Y = 0|X_i = 1]}\right) \\ &+\mathbb{P}_S[X_i = 0]\left(\mathbb{P}_S[Y = 1|X_i = 0]\log\frac{1}{\mathbb{P}_S[Y = 1|X_i = 0]}\right. \\ &\qquad\left. + \mathbb{P}_S[Y = 0|X_i = 0]\log\frac{1}{\mathbb{P}_S[Y = 0|X_i = 0]}\right)\end{aligned}$$

- $H[Y|X_1] = 0.75\left(\frac{2}{3}\log\left(\frac{3}{2}\right) + \frac{1}{3}\log(3)\right) + 0.25\cdot 0$
- $H[Y|X_2] = H[Y|X_3] = 0.5(0.5\log(2) + 0.5\log(2)) + 0.5(0.5\log(2) + 0.5\log(2)) = \log(2)$ This is also the case of uniform probability which gives us the maximum value of entropy function $\log(L)$ (L=2 in our case).

Therefore, $\arg\max_{i\in\{1,2,3\}} G(S,i) = 1$ and X_1 is chosen to be the root.

Now we divide the data to 2 sets:

$$S_0 = \{points\, with\, X_1 = 0\} = \{((0,0,1),0)\}$$
$$S_1 = \{points\, with\, X_1 = 1\} = \{((1,1,1),1),((1,0,0),1),((1,1,0),0)\}$$

In this point S_0 has a unique label so this is a leaf labeled by 0.

We do the same calculation as before on S_1 on the set $\{X_2, X_3\}$ and we chose the next split node:

Probabilities table:

	X_2	X_3	Y
0	1/3	2/3	1/3
1	2/3	1/3	2/3

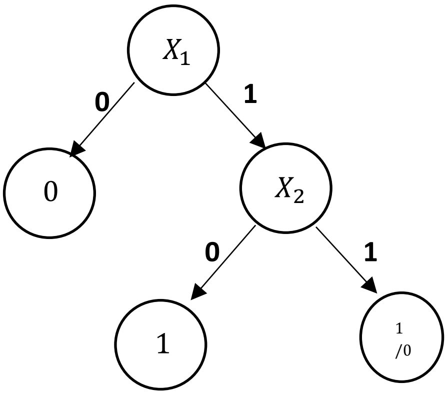
Conditional probabilities: $\mathbb{P}_{S_1}[Y = 1|X_i = j]$

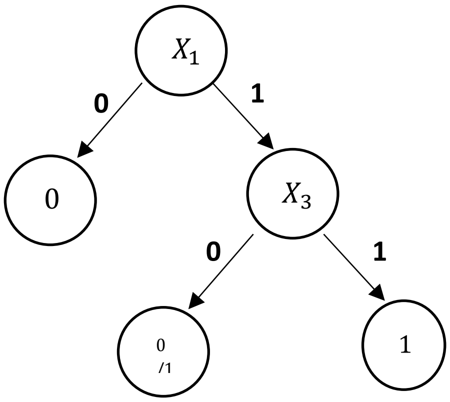
	$X_2 = 1$	$X_2 = 0$	$X_3 = 1$	$X_3 = 0$
$Y = 1$	0.5	1	1	0.5

So,

- $H[Y|X_2] = 2/3(\log 2) + 1/3(0) = 2/3 \cdot \log(2)$
- $H[Y|X_3] = 1/3(0) + 2/3(\log 2) = 2/3 \cdot \log(2)$

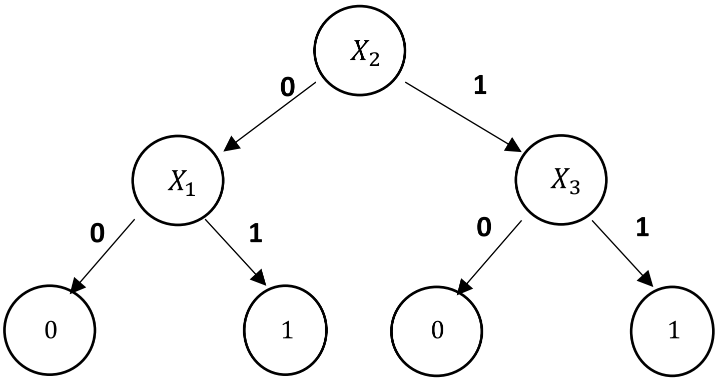
Therefore, $I(Y; X_2) = I(Y; X_3)$ and the split can be done on both of them. Then we need to stop building the tree and we determine the value of leaves by the majority of labeling. We should show that in any case, the train error is at least $1/4$. Let's see our 2 optional trees:





Where we write 0/1 since the majority of both are equal. In any case, it can be shown that we have only 1 error for the data set for this leaf, which gives us error of at least $1/4$.

b. Decision tree with zero training error on the data:



(2) Properties of KL divergence.

a. Solution:

$$D_{KL}(p,q) = \sum_{x,y\in X} p(x,y)\cdot\log\left(\frac{p(x,y)}{q(x,y)}\right) = \sum_{x,y\in X} p_1(x)p_2(y)\cdot\log\left(\frac{p_1(x)p_2(y)}{q_1(x)q_2(y)}\right)$$

$$= \sum_{x,y\in X} p_1(x)p_2(y)\cdot\left(\log\left(\frac{p_1(x)}{q_1(x)}\right)+\log\left(\frac{p_2(y)}{q_2(y)}\right)\right)$$

$$= \sum_{x,y\in X} p_1(x)p_2(y)\log\left(\frac{p_1(x)}{q_1(x)}\right)+\sum_{x,y\in X} p_1(x)p_2(y)\log\left(\frac{p_2(y)}{q_2(y)}\right)$$

$$= \sum_{y\in X} p_2(y)\sum_{x\in X} p_1(x)\log\left(\frac{p_2(x)}{q_2(x)}\right)+\sum_{x\in X} p_1(x)\sum_{y\in X} p_2(y)\log\left(\frac{p_2(y)}{q_2(y)}\right)$$

$$= D_{KL}(p_1,q_1)+D_{KL}(p_2,q_2)\ \blacksquare$$

b. Solution:

$$I(Y;X) = H[Y]-H[Y|X]$$

$$= -\sum_{y\in\mathcal{X}} \mathbb{P}_Y[y]\log\mathbb{P}_Y[y] + \sum_{x\in\mathcal{X}}\mathbb{P}_X[x]\sum_{y\in\mathcal{X}}\mathbb{P}_{X,Y}[y|x]\log\mathbb{P}_{X,Y}[y|x]$$

$$= -\sum_{y\in\mathcal{X}} \mathbb{P}_Y[y]\log\mathbb{P}_Y[y] + \sum_{x,y\in\mathcal{X}}\mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log\mathbb{P}_{X,Y}[y|x]$$

$$= \sum_{x,y\in X}\mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log\left(\mathbb{P}_{X,Y}[y|x]\right) - \sum_{y\in Y}\log(\mathbb{P}_Y[y])\sum_{x\in X}\mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]$$

$$= \sum_{x,y\in X}\mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log\left(\mathbb{P}_{X,Y}[y|x]\right) - \mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log(\mathbb{P}_Y[y])$$

$$= \sum_{x,y\in X}\mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log(\mathbb{P}_X[x]) + \mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log\left(\mathbb{P}_{X,Y}[y|x]\right)$$
$$-\mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log(\mathbb{P}_X[x]) - \mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log(\mathbb{P}_Y[y])$$

$$= \sum_{x,y\in X}\mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log\left(\mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\right) - \mathbb{P}_X[x]\mathbb{P}_{X,Y}[y|x]\log(\mathbb{P}_X[x]\mathbb{P}_Y[y])$$

$$= \sum_{x,y\in X}\mathbb{P}_{X\times Y}[(x,y)]\log\mathbb{P}_{X\times Y}[(x,y)] - \mathbb{P}_{X\times Y}[(x,y)]\log\mathbb{P}_{X\otimes Y}[(x,y)]$$

$$= \sum_{x,y\in X}\mathbb{P}_{X\times Y}[(x,y)]\log\frac{\mathbb{P}_{X\times Y}[(x,y)]}{\mathbb{P}_{X\otimes Y}[(x,y)]} = D_{KL}\left(P_{X\times Y},P_{X\otimes Y}\right)\ \blacksquare$$

(3) Sufficient Condition for Weak Learnability

(a) Take expectation of both sides of (1) wrt D:

$$\gamma \le \sum_{i=1}^{n} D(i)y_i\sum_{j=1}^{k} a_j h_j(x_i) = \sum_{j=1}^{k} a_j\sum_{i=1}^{n} D(i)y_i h_j(x_i)$$

Since $a_j \ge 0, \sum_{j=1}^{k} a_j = 1$, there exists j such that $\sum_{i=1}^{n} D(i)y_i h_j(x_i) \ge \gamma$. If $y_i = h_j(x_i)$ then $y_i h_j(x_i) = 1$ and otherwise $y_i h_j(x_i) = -1$. Therefore,

$$\sum_{i=1}^{n} D(i)y_i h_j(x_i) = \sum_{\substack{1\le i\le n\\ y_i = h_j(x_i)}} D(i) - \sum_{\substack{1\le i\le n\\ y_i \ne h_j(x_i)}} D(i) = 1-2\sum_{\substack{1\le i\le n\\ y_i \ne h_j(x_i)}} D(i) \ge \gamma$$

Hence, $\mathbb{P}_{i\sim D}\left[y_i \ne h_j(x_i)\right] = \sum_{\substack{1\le i\le n\\ y_i\ne h_j(x_i)}} D(i) \le \frac{1}{2}-\frac{\gamma}{2}$ ■

(b) Set $k = 4d - 1, a_i = \frac{1}{4d-1}\ \forall i$. Define the following hypotheses from $\mathcal{H}$:

$$\forall j = 1,2,\dots,d :\ h_{2j-1}(x) = \begin{cases} 1 & x \geq b_j \\ -1 & x < b_j \end{cases}$$

$$\forall j = 1,2,\dots,d :\ h_{2j}(x) = \begin{cases} 1 & x \leq b_j \\ -1 & x > b_j \end{cases}$$

$$\forall j = 2d+1, \dots, 4d-1 :\ h_j(x) = -1$$

Therefore, for $\gamma = \frac{1}{4d-1} > 0$ it holds that:

- If $y_i = +1$ then:

$$y_i \sum_{j=1}^{k} a_j h_j(x_i) = y_i \sum_{j=1}^{2d} \frac{1}{4d-1} h_j(x_i) + y_i \sum_{j=2d+1}^{4d-1} \frac{1}{4d-1} h_j(x_i)$$
$$= \frac{2d}{4d-1} - \frac{2d-1}{4d-1} = \frac{1}{4d-1} = \gamma$$

- If $y_i = -1$ then:

$$y_i \sum_{j=1}^{k} a_j h_j(x_i) = y_i \sum_{j=1}^{2d} \frac{1}{4d-1} h_j(x_i) + y_i \sum_{j=2d+1}^{4d-1} \frac{1}{4d-1} h_j(x_i) = 0 + \frac{2d-1}{4d-1}$$
$$\geq \frac{1}{4d-1} = \gamma$$

(4) Sparsity of LASSO estimator.

Denote the objective be $f(a) = \frac{1}{2}\|y - Xa\|_2^2 + \lambda\|a\|_1$. So, we can write f as follows:

$$f(a) = \frac{1}{2}\|y - Xa\|_2^2 + \lambda\|a\|_1 \Rightarrow f(a) = \sum_{j=1}^{d} -z_j a_j + \frac{1}{2}\sigma_j a_j^2 + \lambda|a_j|$$

Where $z_j = \sum_{i=1}^{n} y_i X_{ij}$.

- $\|y - Xa\|_2^2 = \langle y - Xa, y - Xa\rangle = \langle y, y\rangle - 2\langle y, Xa\rangle + \langle Xa, Xa\rangle = n - 2\sum_{j=1}^{d}\sum_{i=1}^{n} y_i X_{ij} a_j + a^T X^T X a = n + 2\sum_{j=1}^{d}\left(-\sum_{i=1}^{n} y_i X_{ij}\right) a_j + \sum_{j=1}^{d} \sigma_j a_j^2$
- $\lambda\|a\|_1 = \sum_{j=1}^{d} \lambda|a_j|$
- We can remove n from the objective function since the minimum does not depend on additive constant.

The derivative w.r.t a: $\nabla_j f(a) = -z_j + \sigma_j a_j + \lambda\delta_j$ where $\delta_j = \begin{cases} 1 & a_j > 0 \\ 0 & a_j = 0 \\ -1 & a_j < 0 \end{cases}$.

Since f is convex function in a then there is minimum where $\nabla_j f(a) = 0$. We shell show that $\hat{a}_j^{lasso} = \frac{sign(z_j)}{\sigma_j}\max(0, |z_j| - \lambda)$ gives $\nabla_j f(\hat{a}^{lasso}) = 0$.

Consider the first case where $sign(z_j) = +1$. So, $\hat{a}_j^{lasso} \geq 0$.

- If $\hat{a}_j^{lasso} = 0$ then $\delta_j = 0,\ z_j - \lambda \leq 0 \Longrightarrow \nabla_j f(\hat{a}_{lasso}) = -z_j + \sigma_j \frac{1}{\sigma_j}\max(0, z_j - \lambda) = 0$
- If $\hat{a}_j^{lasso} > 0$ then $\delta_j = 1,\ z_j - \lambda > 0 \Longrightarrow \nabla_j f(\hat{a}_{lasso}) = -z_j + \sigma_j \frac{1}{\sigma_j}\max(0, z_j - \lambda) + \lambda = \max(-(z_j - \lambda), 0) = 0$

In the same manner, if $sign(z_j) = -1$, Then:

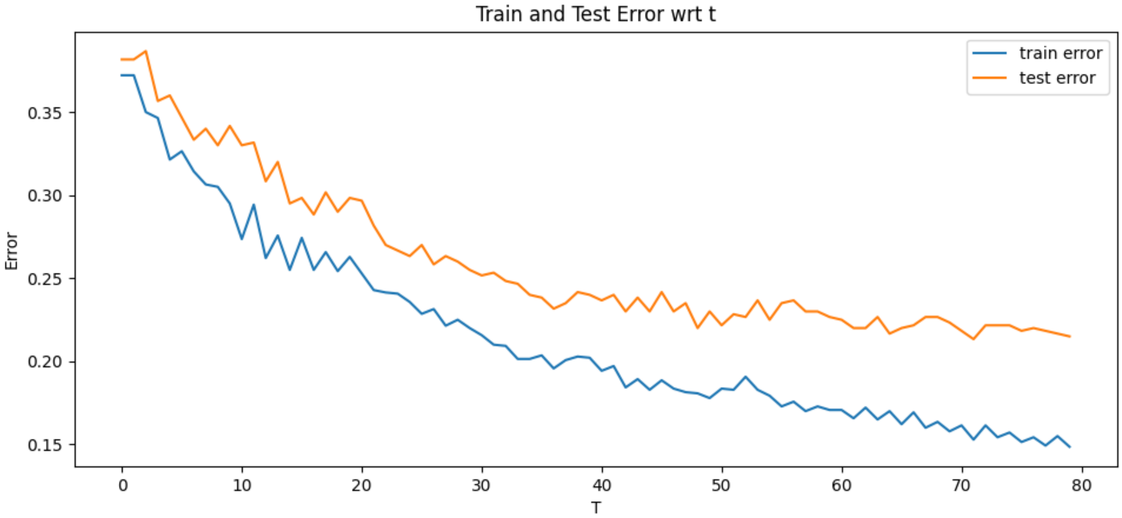
$\nabla_j f(\hat{a}_{lasso}) = -z_j + \sigma_j \frac{sign(z_j)}{\sigma_j}\max(0, |z_j| - \lambda) + \lambda\delta_j = -\max(0, -z_j - \lambda) - (z_j + \lambda) = -\max(-(z_j + \lambda), 0) = 0$

Introduction to Machine Learning **Liad Iluz**
Assignment 5 **208427435**

Programming Assignment

(1) AdaBoost.

(a) Implementing AdaBoost and plotting train and test error over the iteration number:



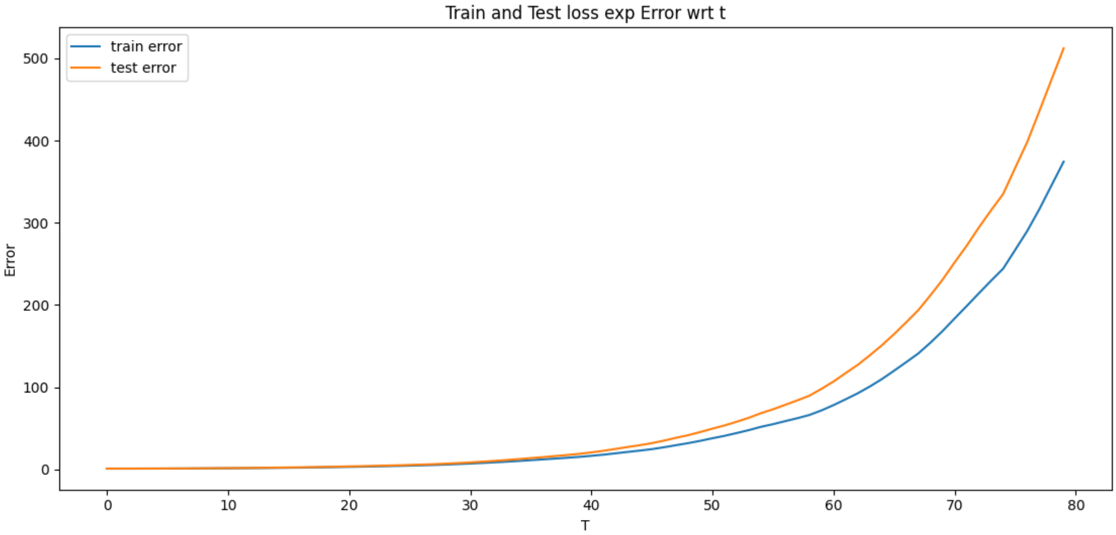
(b) Weak classifiers during the 10 iterations AdaBoost:

```
Weak classifier at t=0: h_pred=1.0     h_index=26.0[bad]              h_theta=0.0
Weak classifier at t=1: h_pred=-1.0    h_index=31.0[many]             h_theta=0.0
Weak classifier at t=2: h_pred=-1.0    h_index=22.0[life]             h_theta=0.0
Weak classifier at t=3: h_pred=1.0     h_index=311.0[worst]           h_theta=0.0
Weak classifier at t=4: h_pred=-1.0    h_index=37.0[great]            h_theta=1.0
Weak classifier at t=5: h_pred=1.0     h_index=372.0[boring]          h_theta=0.0
Weak classifier at t=6: h_pred=-1.0    h_index=282.0[perfect]         h_theta=0.0
Weak classifier at t=7: h_pred=1.0   h_index=292.0[supposed]          h_theta=0.0
Weak classifier at t=8: h_pred=-1.0    h_index=196.0[performances]      h_theta=0.0
Weak classifier at t=9: h_pred=1.0     h_index=88.0[script]           h_theta=0.0
```

Weak classifiers that I would expect to help to classify reviews:
All the blue weak classifiers are good ones, since the classifier's word of each of them represent the review as good or bad review. For the good words such as *great* or *perfect*, the algorithm picks classifier which return -1 if those words appear 1 or 0 times respectively; while for bad words such as *bad*, *worst* or *boring* the classifier return 1 only if each of them appears once in the review.

Weak classifiers that I would not expect to help to classify reviews:
The orange weak classifiers are unexpected ones. Each of them used a generic word such as *many*, *supposed* or *script*, without a strongly meaningful for good or bad review. The algorithm picks the words *supposed* and *script* as are related to bad review while it picks *many* as good review word.

(c) Exponential loss over 80 iterations of AdaBoost:



The loss increases in both cases: train and test. Initially, the incremental is small and the loss of both cases is getting around the value of 0-20 during the first 30 iterations. Afterwards, the loss grows fest. The test loss exceeds the train loss every time.

I expect that the loss will decrease over time since $sign(\sum_{t=1}^{T} \alpha_t h_t(x_i))$ is getting close to y_i as t grows, so $y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)$ should be a positive value and so $e^{-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)}$ should be small number for all i. As we saw in lecture, the AdaBoost does also coordinate descent over the exp loss, so the expected graph should be decreasing over time.