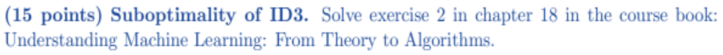
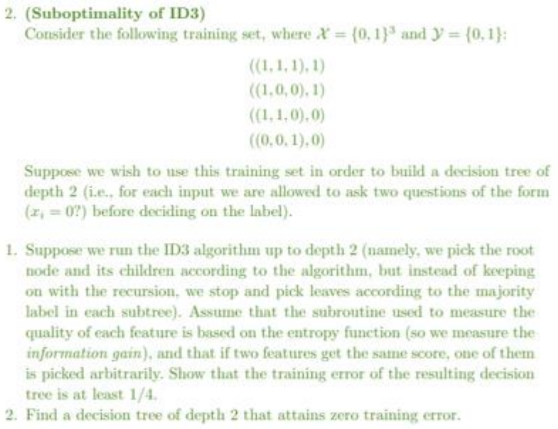
Introduction to Machine Learning: Ex05

Theoretical Questions

1.



(15 points) Suboptimality of ID3. Solve exercise 2 in chapter 18 in the course book: Understanding Machine Learning: From Theory to Algorithms.



2. **(Suboptimality of ID3)**
Consider the following training set, where $\mathcal{X} = \{0,1\}^3$ and $\mathcal{Y} = \{0,1\}$:

$$((1,1,1),1)$$
$$((1,0,0),1)$$
$$((1,1,0),0)$$
$$((0,0,1),0)$$

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).

1. 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.
2. Find a decision tree of depth 2 that attains zero training error.

Annotation: $x \in X \rightarrow x = (b_0, b_1, b_2)$ so we mark the question $q_i(G)$ as the question about b_i on group G.

a. Training Error of ID3

Using the ID on the given set S:
$$S = \begin{cases} x_0 = (1,1,1) & y_0 = [1] \\ x_1 = (1,0,0) & y_1 = [1] \\ x_2 = (1,1,0) & y_2 = [0]: \\ x_3 = (0,0,1) & y_3 = [0] \\ A = (q_0, q_1, q_2) & \end{cases}$$

1st Iteration

$A = \phi \wedge \exists x_i, x_j \in X. st\ y_i \neq y_j$ *so we calculate:* $j = argmax_{i \in A}(H(Y) - H(Y|X_i))$ *when* $P_Y[0] = P_Y[1] = \frac{1}{2}$

$$j = argmax_{i\in A}\Big(P_Y[1]\log(P_Y[1]) - P_y[0]\log(P_Y[0]) - P_{X_i}[1]P_{Y,X_i}[1,1]\log(P_{Y,X_i}[1,1]) - P_{X_i}[1]P_{Y,X_i}[0,1]\log(P_{Y,X_i}[1,1]) - P_{X_i}[0]P_{Y,X_i}[1,0]\log(P_{Y,X_i}[1,1]) - P_{X_i}[0]P_{Y,X_i}[0,0]\log(P_{Y,X_i}[0,0])\Big)$$

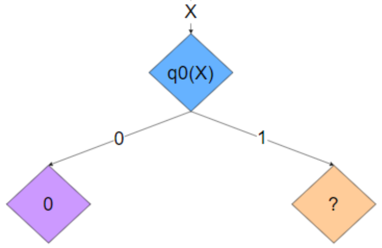
Since $P_Y[1]\log(P_Y[1]) = P[Y=0]\log(P_Y[0])$ *we get* $= P_Y[1]\log(P_Y[1]) - P_Y[0]\log(P_Y[0]) = 0$

Observing $P_{X_0}[0] = \frac{1}{4}, P_{X_1}[0] = P_{X_2}[0] = \frac{1}{2}$ *will let us see that*

for $i \in (1,2)$: $P_{Y,X_i}[1,1] = P_{Y,X_i}[1,0] = P_{Y,X_i}[0,1] = P_{Y,X_i}[0,0\ ] = \frac{1}{2}$

for $i = 0$: $P_{Y,X_i}[1,1] = \frac{2}{3}, P_{Y,X_i}[1,0] = P_{Y,X_i}[0,1] = \frac{1}{2}, P_{Y,X_i}[0,0] = \frac{1}{2}$

so: $j = argmax(0.4774, 0.301, 0.301)$ *meaning our first question will be* $q_0(X) = \{X_0, X_1\}\ s.t\ X_0 = \{x_3\}. X_1 = \{x_1, x_2, x_3\}$



2nd Iteration

For the 2nd iteration we have to iterate only on X_1 (since X_0 has a single value) and we have two possible splits $A = \{q_1, q_2\}$ *.*

Using each of them will result a single error which is 25% :

$q_1(X_1) = \{\widetilde{X_0}, \widetilde{X_1}\}\ s.t\ \widetilde{X_0} = \{x_1\}, \widetilde{X_1} = \{x_0, x_2\}$ *and since we have no more steps, we will have to choose arbitrary between the labels of x_0, x_2 which are different: hence 1 mistake.*

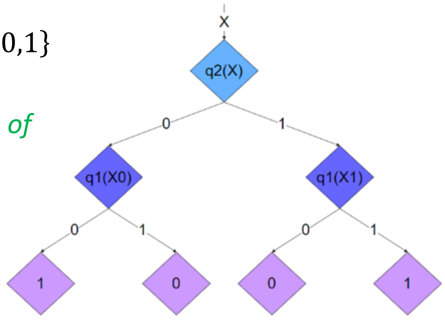
$q_2(X_1) = \{\widetilde{X_0}, \widetilde{X_1}\}\ s.t\ \widetilde{X_0} = \{x_0\}, \widetilde{X_1} = \{x_1, x_2\}$ *and since we have no more steps, we will have to choose arbitrary between the labels of x_0, x_1 which are different: hence 1 mistake.*

b. finding a DT that yields zero training error on the given set

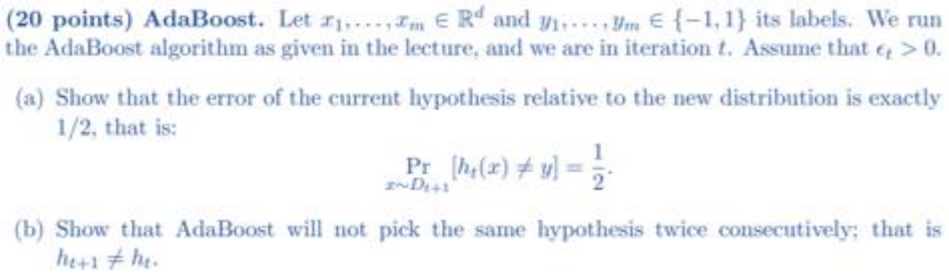
Staring with $q_2(X)$ for splitting will result two groups $X_0, X_1 \subseteq B^3 \times B$ *when* $B = \{0,1\}$

$$X_0 = \{((1,1,0),[0]), (1,0,0),[1]\}\quad X_1 = \{((0,0,1),[0]), (1,1,1),[1]\}$$

Since each one of the groups is to be divided once more, and because each group is of size 2 we can split each group by the element which is different:
in our case $q_1(X_0), q_1(X_1)$



2.



(20 points) AdaBoost. Let $x_1, \dots, x_m \in \mathbb{R}^d$ and $y_1, \dots, y_m \in \{-1, 1\}$ its labels. We run the AdaBoost algorithm as given in the lecture, and we are in iteration t. Assume that $\epsilon_t > 0$.

(a) Show that the error of the current hypothesis relative to the new distribution is exactly 1/2, that is:

$$\Pr_{x \sim D_{t+1}} [h_t(x) \neq y] = \frac{1}{2}.$$

(b) Show that AdaBoost will not pick the same hypothesis twice consecutively; that is $h_{t+1} \neq h_t$.

a. Error of current Hypothesis

Lets mark $\delta_t(i) = \begin{cases} 1 & h_t(x_i) \neq y_1 \\ 0 & h_t(x_i) = y_1 \end{cases}$ *and* $\Delta_t = \{i \mid \delta_t(i) = 1\}$

From the properties of the distribution D that for $X \sim D_{t+1}$*:*

$$P_X[h_t(x) \neq y] = \sum_{i=1}^{n} D_{t+1}(i) \cdot \delta_t(i) = \sum_{i=1}^{n} \frac{D_t(i) e^{-w_t y_i h_t(x_i)}}{Z_t} \cdot \delta_t(i) = \frac{e^{wt}}{Z_t} \sum_{i \in \Delta_t}^{n} D_t(i)$$

Since $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$ *and* $w_t = \frac{1}{2} log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ *we get:*

$$P_X[h_t(x) \neq y] = \frac{\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{2\sqrt{\epsilon_t(1-\epsilon_t)}} \cdot \epsilon_t = \frac{1}{2} \frac{\sqrt{1-\epsilon_t}\sqrt{\epsilon_t}}{\sqrt{\epsilon_t(1-\epsilon_t)}} = \frac{1}{2}$$

b. Showing that AdaBoost will not pick same hypothesis twice sequentially

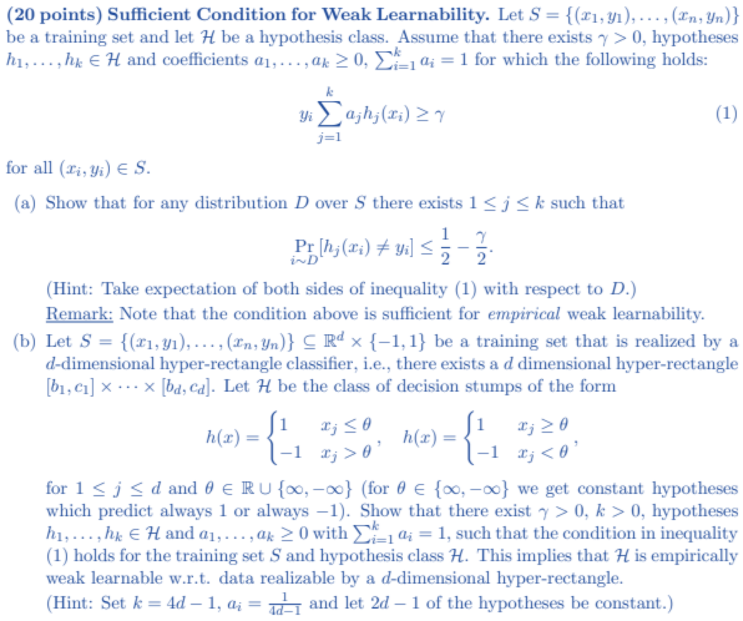
We know that Adaboost satisfies $\forall T . \exists \gamma > 0 \; s.t \; \epsilon_T < \frac{1}{2} - \gamma < \frac{1}{2}$

We will assume (in contradiction) that $\exists t . h_t = h_{t+1}$ *then: for* $X \sim D_{t+1}$: $P_X[h_t(x) \neq y)] = P_X[h_{t+1}(x) \neq y)] = \frac{1}{2}$

which when combined with previous constraint $\frac{1}{2} - \gamma < \frac{1}{2}$ *we require* $\gamma \equiv 0$ *which is a contradiction:*

therefore $h_t \neq h_{t+1}$

3.



(20 points) Sufficient Condition for Weak Learnability. Let $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ be a training set and let $\mathcal{H}$ be a hypothesis class. Assume that there exists $\gamma > 0$, hypotheses $h_1, \dots, h_k \in \mathcal{H}$ and coefficients $a_1, \dots, a_k \geq 0$, $\sum_{i=1}^{k} a_i = 1$ for which the following holds:

$$y_i \sum_{j=1}^{k} a_j h_j(x_i) \geq \gamma \tag{1}$$

for all $(x_i, y_i) \in S$.

(a) Show that for any distribution D over S there exists $1 \leq j \leq k$ such that

$$\Pr_{i \sim D}[h_j(x_i) \neq y_i] \leq \frac{1}{2} - \frac{\gamma}{2}.$$

(Hint: Take expectation of both sides of inequality (1) with respect to D.)

Remark: Note that the condition above is sufficient for *empirical* weak learnability.

(b) Let $S = \{(x_1, y_1), \dots, (x_n, y_n)\} \subseteq \mathbb{R}^d \times \{-1, 1\}$ be a training set that is realized by a d-dimensional hyper-rectangle classifier, i.e., there exists a d dimensional hyper-rectangle $[b_1, c_1] \times \cdots \times [b_d, c_d]$. Let $\mathcal{H}$ be the class of decision stumps of the form

$$h(x) = \begin{cases} 1 & x_j \leq \theta \\ -1 & x_j > \theta \end{cases}, \quad h(x) = \begin{cases} 1 & x_j \geq \theta \\ -1 & x_j < \theta \end{cases},$$

for $1 \leq j \leq d$ and $\theta \in \mathbb{R} \cup \{\infty, -\infty\}$ (for $\theta \in \{\infty, -\infty\}$ we get constant hypotheses which predict always 1 or always -1). Show that there exist $\gamma > 0$, $k > 0$, hypotheses $h_1, \dots, h_k \in \mathcal{H}$ and $a_1, \dots, a_k \geq 0$ with $\sum_{i=1}^{k} a_i = 1$, such that the condition in inequality (1) holds for the training set S and hypothesis class $\mathcal{H}$. This implies that $\mathcal{H}$ is empirically weak learnable w.r.t. data realizable by a d-dimensional hyper-rectangle.

(Hint: Set $k = 4d - 1$, $a_i = \frac{1}{4d-1}$ and let $2d - 1$ of the hypotheses be constant.)

a. *Showing that for any distribution D over S:* $\exists 1 \leq j \leq k \; s.t. P_i\left[h_j(x_i) \neq y_i\right] \leq \frac{1}{2}(1 - \gamma)$

Using the hint:

$$E\left[y_i \sum_{j=1}^{k} a_j h_j(x_i)\right] \geq \gamma \rightarrow \sum_{j=1}^{k} a_j \cdot E\left[y_i h_j(x_i)\right] \geq \gamma$$

We will now assume (in contradiction) that the condition isn't satisfied:

$\forall 1 \leq i \leq j. P_i\left[h_j(x_i) \neq y_i\right] > \frac{1}{2}(1 - \gamma)$ *so* $P_i\left[h_j(x_i) = y_i\right] = 1 - P_i\left[h_j(x_i) \neq y_i\right] < \frac{1}{2}(1 + \gamma)$

so we see that $E\left[y_i h_j(x_i)\right] = P_i\left[h_j(x_i) = y_i\right] - P_i\left[h_j(x_i) \neq y_i\right] < \frac{1}{2}(1 + \gamma) - \frac{1}{2}(1 - \gamma)$ *meaning* $E\left[y_i h_j(x_i)\right] < \gamma$ *so*

$$\mathbb{E}\left[y_i \sum_{j=1}^{k} a_j h_j(x_i)\right] = \sum_{j=1}^{k} a_j \mathbb{E}\left[y_i h_j(x_i)\right] < \gamma \sum_{j=1}^{k} a_j$$

Since $\sum_i a_i = 1$ *we get* $\mathbb{E}\left[y_i \sum_{j=1}^{k} a_j h_j(x_i)\right] < \gamma$, *which contradict inequality #1. Hence our assumption (of unsatisfied condition) was wrong: and the condition is satisfied.*

b. *Showing inequality #1*

Observation: a constant hypothesis $h_b \in \mathcal{H}$ *satisfied* $\exists b \in \{-1,1\}. \forall x. h_b(x) = b$ *can be set by using the right* θ.

Using the hint: setting $k = 4d - 1$ *and choosing the hypotheses* $h_1, \cdots, h_k$

We will set the first $2d - 1$ *hypotheses as* $h_{-1}(\cdot)$

We will set the last $2d - 1$ *hypotheses as those which mark the d-dimensional hyper-rectangle for which the training error is 0.*

We will set the coefficients $\forall i \in [4d - 1]. a_i = \frac{1}{4d-1}$ *and will mark it as* β. *Observe that* $(4d - 1)\beta = 1$.

$$R = y_i \sum_{j=1}^{k} a_j h_j(x_i) = y_i \sum_{j=1}^{4d-1} \beta \, h_j(x_i) = y_i \left((1 - 2d)\beta + \sum_{j=2d}^{4d-1} h_j(x_i) \cdot \beta \right)$$

If $y_i = -1$ then x_i isn't in the hyper-rectangle, which means that at least one coordinate of x_i is outside of the hyper-rectangle: $\exists j \in [2d, 4d - 1]. h_j(x_i) = -1$ so

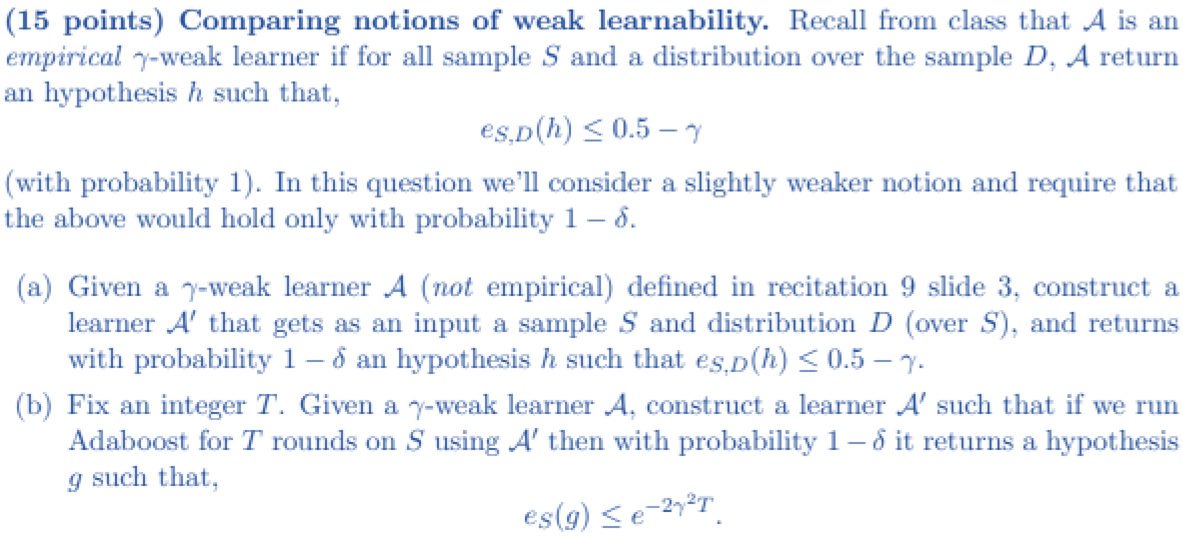
$$R = y_i \sum_{j=1}^{k} a_j h_j(x_i) \geq -1\big((1 - 2d)\beta + (2d - 1)\beta - \beta\big) \rightarrow R \geq \beta$$

If $y_i = 1$, then x_i is in the hyper-rectangle which means $\forall j \in [2d, 4d - 1]. h_j(x_i) = 1$

$$R = y_i \sum_{j=1}^{k} a_j h_j(x_i) = 1\big((1 - 2d)\beta + 2d\beta\big) = \beta$$

Setting $\gamma = \beta = \frac{1}{k}$ will result satisfaction of the inequality.

4.



(15 points) Comparing notions of weak learnability. Recall from class that $\mathcal{A}$ is an *empirical* γ-weak learner if for all sample S and a distribution over the sample D, $\mathcal{A}$ return an hypothesis h such that,

$$e_{S,D}(h) \leq 0.5 - \gamma$$

(with probability 1). In this question we'll consider a slightly weaker notion and require that the above would hold only with probability $1 - \delta$.

(a) Given a γ-weak learner $\mathcal{A}$ (*not* empirical) defined in recitation 9 slide 3, construct a learner $\mathcal{A}'$ that gets as an input a sample S and distribution D (over S), and returns with probability $1 - \delta$ an hypothesis h such that $e_{S,D}(h) \leq 0.5 - \gamma$.

(b) Fix an integer T. Given a γ-weak learner $\mathcal{A}$, construct a learner $\mathcal{A}'$ such that if we run Adaboost for T rounds on S using $\mathcal{A}'$ then with probability $1 - \delta$ it returns a hypothesis g such that,

$$e_S(g) \leq e^{-2\gamma^2 T}.$$

a. *Constructing $A': S \times D \to H$ that return $h \in H$ st $e_{S,D}(h) \leq \frac{1}{2} - \gamma$ with probability $1 - \delta$*

Since we know there is a γ-weak-learner (A) we know by definition (course book, page 131 definition 10.1) that there are functions $N: (0,1) \to \mathbb{N}$, $f: S \to \{\pm 1\}$ s.t for every D over S when S is constructed from n i.i.d samples:

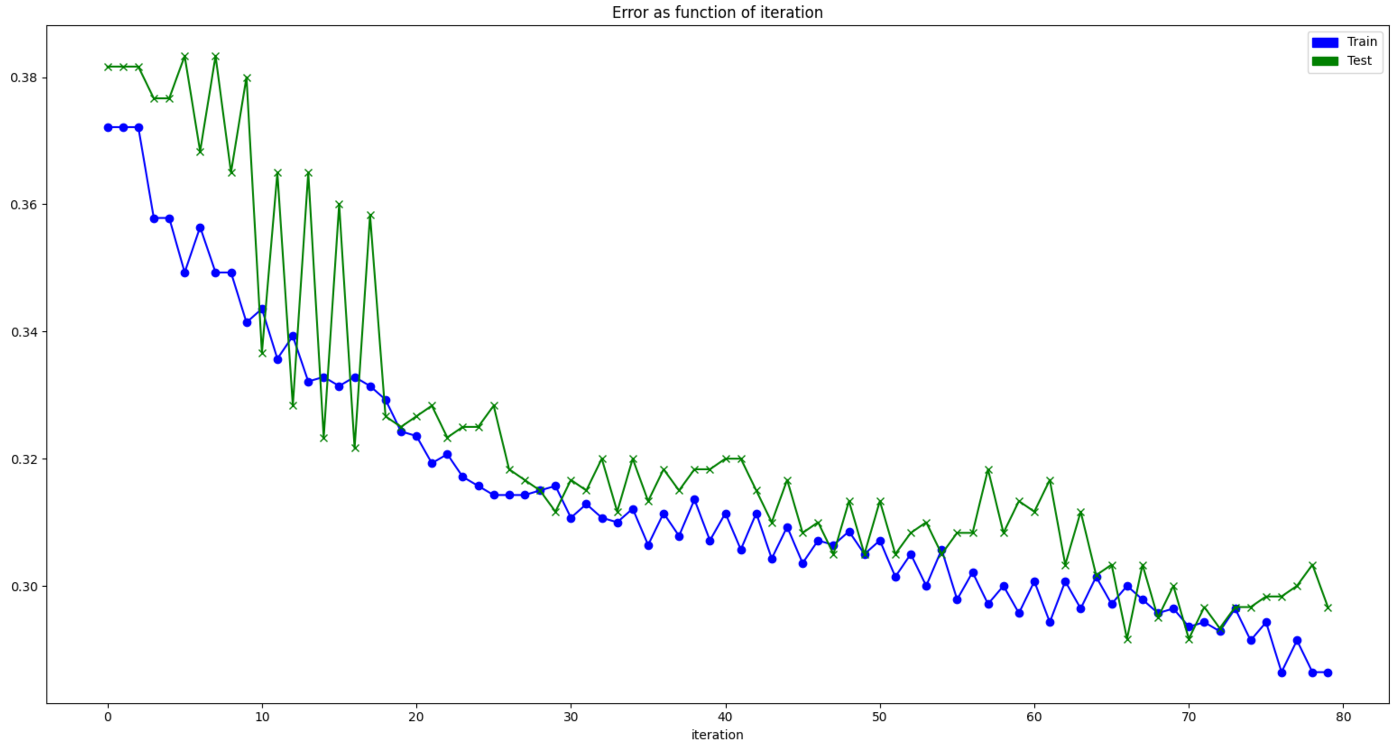
$$\forall \delta \in (0,1). n > N(\delta) \to \; e_{S,D}(h) \leq \frac{1}{2} - \gamma \; with\; probability\; 1 - \delta$$

So all A' has to do it to run A instead

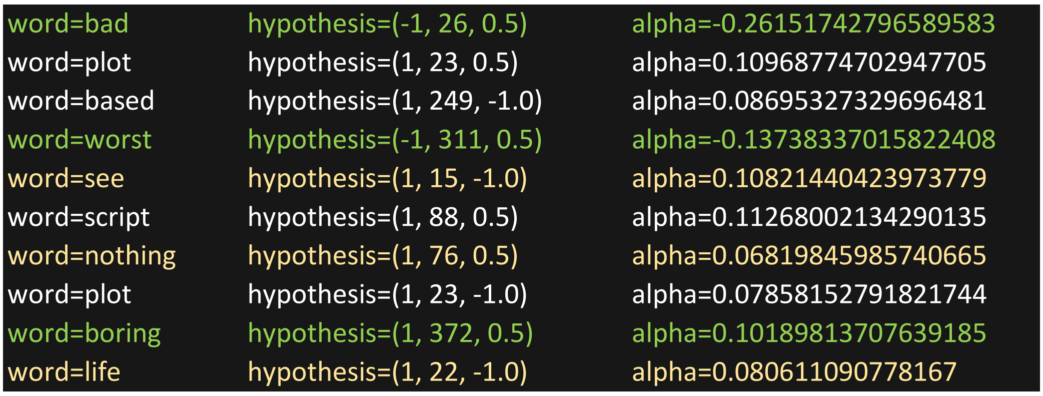
b. *Constructing $A': S \times D \to H$ that return $g \in H$ st $e_s(g) \leq e^{-2\gamma^2 T}$ with probability $1 - \delta$ for a constant T*

Programming Assignment

1.



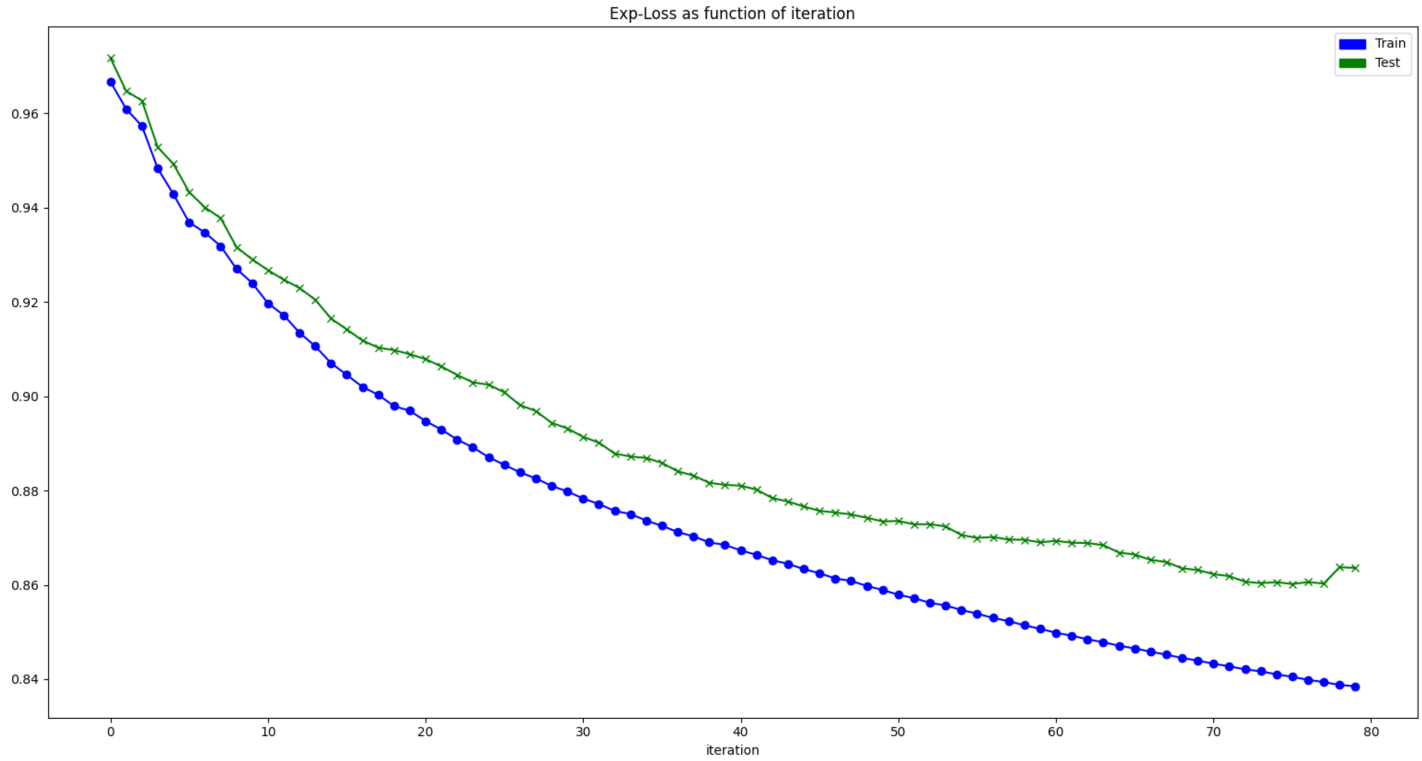
2. Running for T=10, we result that those are the top 10 words that helped labeling the movie.
The 3 words that I expected to be used are in green an the 3 that I wasn't expecting are in yellow.



```
word=bad          hypothesis=(-1, 26, 0.5)        alpha=-0.26151742796589583
word=plot         hypothesis=(1, 23, 0.5)         alpha=0.10968774702947705
word=based        hypothesis=(1, 249, -1.0)       alpha=0.08695327329696481
word=worst        hypothesis=(-1, 311, 0.5)       alpha=-0.13738337015822408
word=see          hypothesis=(1, 15, -1.0)        alpha=0.10821440423973779
word=script       hypothesis=(1, 88, 0.5)         alpha=0.11268002134290135
word=nothing      hypothesis=(1, 76, 0.5)         alpha=0.06819845985740665
word=plot         hypothesis=(1, 23, -1.0)        alpha=0.07858152791821744
word=boring       hypothesis=(1, 372, 0.5)        alpha=0.10189813707639185
word=life         hypothesis=(1, 22, -1.0)        alpha=0.080611090778167
```

My explanation for the 'surprising' words are due to the fact that human formulate sentences (especially when writing reviews) the way they speak, which may be figurative. For example 'life' can be used as part of 'the best movie I've seen in my life' or 'see' can be used as 'I want to see it again'.
While the expected word are adjectives such as: bas, worst, boring, where a person can use a single word to describe a movie.

3.



As we expected we received that $\ell_{\exp}(\alpha)$ is monotonically decreasing for every iteration in AdaBoost when checked for training set. This is happing because the AdaBoost algorithm minimizing $ExpLoss$ (like coordinate decent). we can see that when we are looking on the whole test set we can see that the $ExpLoss$ is also decreasing in general , but not monotonically due to differences in the distribution of the data.