Introduction to Machine Learning **Spring Semester**

Homework 5: May 4, 2022

Due: May 18, 2022

Theory 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. **(20 points) AdaBoost.** Let $x_1, \ldots, x_m \in \mathbb{R}^d$ and $y_1, \ldots, y_m \in \{-1, 1\}$ 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 $h_{t+1} \neq h_t$.

3. **(20 points) Sufficient Condition for Weak Learnability.** Let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a training set and let $\mathcal{H}$ be a hypothesis class. Assume that there exists $\gamma > 0$, hypotheses $h_1, \ldots, h_k \in \mathcal{H}$ and coefficients $a_1, \ldots, a_k \geq 0$, $\sum_{i=1}^k a_i = 1$ 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 $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subseteq \mathbb{R}^d \times \{-1, 1\}$ 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 $h_1, \ldots, h_k \in \mathcal{H}$ and $a_1, \ldots, a_k \geq 0$ 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.)

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

Programming Assignment (30 points)

Submission guidelines:

- Download the supplied files from Moodle (2 python files and 1 tar.gz file). Written solutions, plots and any other non-code parts should be included in the written solution submission.
- Your code should be written in Python 3.
- Your code submission should include these files: `adaboost.py`, `process_data.py`.

1. **(30 points) AdaBoost.** In this exercise, we will implement AdaBoost and see how boosting can be applied to real-world problems. We will focus on binary sentiment analysis, the task of classifying the polarity of a given text into two classes - positive or negative. We will use movie reviews from IMDB as our data. Download the provided files from Moodle and put them in the same directory:

   - `review_polarity.tar.gz` - a sentiment analysis dataset of movie reviews from IMDB.[1] Extract its content in the same directory (with any of zip, 7z, winrar, etc.), so you will have a folder called `review_polarity`.
   - `process_data.py` - code for loading and preprocessing the data.
   - `skeleton_adaboost.py` - this is the file you will work on, change its name to `adaboost.py` before submitting.

   The main function in `adaboost.py` calls the `parse_data` method, that processes the data and represents every review as a 5000 vector $\mathbf{x}$. The values of $\mathbf{x}$ are counts of the most common words in the dataset (excluding stopwords like "*a*" and "*and*"), in the review that $\mathbf{x}$ represents. Concretely, let $w_1, w_2, \ldots, w_{5000}$ be the most common words in the data. Given a review r_i we represent it as a vector $\mathbf{x}_i \in \mathbb{N}^{5000}$ where $x_{i,j}$ is the number of times the word w_j appears in the review r_i. The method `parse_data` returns a training data, test data and a vocabulary. The vocabulary is a dictionary that maps each index in the data to the word it represents (i.e. it maps $j \to w_j$).

   (a) **(10 points)** Implement the AdaBoost algorithm in the run `adaboost` function. The class of weak learners we will use is the class of hypotheses of the form:

   $$h(\mathbf{x}_i) = \begin{cases} 1 & x_{i,j} \leq \theta \\ -1 & x_{i,j} > \theta \end{cases}, \quad h(\mathbf{x}_i) = \begin{cases} -1 & x_{i,j} \leq \theta \\ 1 & x_{i,j} > \theta \end{cases}$$

   That is, comparing a single word count to a threshold. At each iteration, AdaBoost will select the best weak learner. Note that the labels are $\{-1, 1\}$. Run AdaBoost for $T = 80$ iterations. Plot the training error and the test error of the classifier corresponding to each iteration t (as a function of t), that is, $sign\left(\sum_{j=1}^{t} \alpha_j h_j(\mathbf{x})\right)$. Include a single plot containing both the training error and the test error.

   (b) **(10 points)** Run AdaBoost for $T = 10$ iterations. Which weak classifiers did the algorithm choose? Pick 3 that you would expect to help to classify reviews and 3 that you did not expect to help, and explain possible reasons for the algorithm to choose them.

[1] `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

(c) **(10 points)** In the lecture you saw that AdaBoost works towards minimizing the average exponential loss:

$$\ell_{exp}(\boldsymbol{\alpha}) = \frac{1}{m}\sum_{i=1}^{m} e^{-y_i \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}_i)}$$

Run AdaBoost for $T = 80$ iterations. Plot ℓ_{exp} as a function of t, for both the training and test sets. Explain the behavior of the loss.