Introduction to Machine Learning **Spring Semester**

Homework 5: June 11, 2023

Due: June 27, 2023

Theory Questions

1. **(10 points) Suboptimality of ID3.** Solve exercise 2 in chapter 18 in the course book: Understanding Machine Learning: From Theory to Algorithms.

2. **(20 points) Properties of KL divergence**. Recall the definition of the KL-divergence from slide 16 in lecture 7.

   (a) Let p_1, p_2, q_1, q_2 be distributions over $\mathcal{X}$ (you can assume for simplicity that they are discrete) such that p_1 is independent of p_2 and q_1 is independent of q_2. Denote the product distributions $p = p_1 \times p_2$ and $q = q_1 \times q_2$ over $\mathcal{X}^2$ (i.e. $p(x_1, x_2) = p_1(x_1)p_2(x_2)$ for any $x_1, x_2 \in \mathcal{X}$ and similarly for q). Prove the following:

   $$D_{KL}(p, q) = D_{KL}(p_1, q_1) + D_{KL}(p_2, q_2).$$

   (b) Let X and Y be two discrete random variables over a domain $\mathcal{X}$ and denote by $P_{X \times Y}$ their joint distribution. Denote by $P_{X \otimes Y}$ the product distribution of X and Y, i.e.:

   $$P_{X \otimes Y}(x, y) = P(X = x) \cdot P(Y = y).$$

   Recall the definition of the *mutual information* between X and Y, denoted $I(Y; X)$ (see the last slide of recitation 7). Prove the following relation between the mutual information and the KL-divergence:

   $$I(Y; X) = D_{KL}(P_{X \times Y}, P_{X \otimes Y}).$$

   Can you give an intuitive explanation for this identity?

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, the condition defined in lecture #9 for the Adaboost analysis.

(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 $[b_1, c_1] \times \cdots \times [b_d, c_d]$ which contains all of the positive points in S and doesn't contain the negative points in S. 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. **(20 points) Sparsity of LASSO estimator.** Consider the solution for LLS with ℓ_1 regularization, also known as LASSO:

$$\hat{\mathbf{a}}^{\text{lasso}} = \arg\min_{\mathbf{a}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{a}\|_{\mathbf{2}}^{\mathbf{2}} + \lambda\|\mathbf{a}\|_{\mathbf{1}}.$$

Show that if $X^T X = diag(\sigma_1, ..., \sigma_d)$ where $\sigma_j > 0$ for all j then,

$$\hat{a}_j^{\text{lasso}} = \frac{sign(z_j)}{\sigma_j} \max(0, |z_j| - \lambda),$$

where $z_j = \sum_{i=1}^{n} y_i X_{ij}$.
(**Hint:** First, rewrite the objective as $\sum_{j=1}^{d} \left(-z_j a_j + \frac{1}{2}\sigma_j a_j^2 + \lambda|a_j|\right)$. Then, for each j divide the optimization to cases according to the sign of z_j).

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.