Introduction to Machine Learning - Exercise 1

Due Date: November 9th, 2021

Yosi Shrem, Yael Segal, Roni Chernyak and Yossi Keshet

October 26th, 2021

1 ERM

1.1

As mentioned in the class, a learning algorithm receives as input a training set S sampled from an unknown distribution $\mathcal{D}$ and labeled by some target function f. Since the learner does not know what $\mathcal{D}$ and f are, we use a training set of examples, which acts as a snapshot of the world that is available to the learner. In ERM we would like to find a solution that works well on that data.

An axis aligned classifier in the plane is a classifier that assigns the value 1 to a point if and only if it is inside a certain rectangle. Formally, given real numbers $a_1 \leq b_1, a_2 \leq b_2$, define the classifier $h_{(a_1,b_1,a_2,b_2)}$ by

$$h_{(a_1,b_1,a_2,b_2)}(x_1,x_2) = \begin{cases} 1 & \text{if } a_1 \leq x_1 \leq b_1 \textit{ and } a_2 \leq x2 \leq b_2 \\ 0 & \text{otherwise} \end{cases}$$

Let A be the algorithm that returns the smallest rectangle enclosing all positive examples in the training set. Explain whether A is an ERM or not.

Note: We rely on the realizability assumption. In another words, we assume that there is a rectangle that classifies correctly all the data points.

1.2

Let $\mathcal{H}$ be the hypothesis space of binary classifiers over a domain $\mathcal{X}$. Let $\mathcal{D}$ be an unknown distribution over $\mathcal{X}$, and let f be the target hypothesis in $\mathcal{H}$. Denote $h \in \mathcal{H}$.

Let us define the *true error* of h as,

$$L_{\mathcal{D}}(h) = \mathbb{P}_{x\sim\mathcal{D}}[h(x) \neq f(x)]$$

Let us define the *empirical error* of h over the training set S as,

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{[h(x)\neq f(x)]}$$

where m is the number of training examples.

Show that the expected value of $L_S(h)$ over the choice of S equals $L_{\mathcal{D}}(h)$, namely,

$$\mathbb{E}_{S\sim\mathcal{D}}\big[L_s(h)\big] = L_{\mathcal{D}}(h)$$

2 Image Compression

Guidelines

1. You are not allowed to use external packages other than `os, sys, numpy` and `matplotlib`.
2. In order to submit your solution please upload your files to `Submit` and check your inbox for the feedback mail.
3. Technical questions about this exercise should be asked at the course' piazza.
4. Private/Personal issues regarding the deadline should be directed to **Yosi shrem**.

In this part of the exercise we will use the k-means algorithm for image compression, i.e. you should implement the k-means algorithm on the **image pixels** and then replace each pixel by its centroid.
You should implement the k-means algorithm as described in class (in recitation 2 presentation).
You can use the following python snippet for reading, normalizing and reshaping the image so it will be ready for training.

```
import matplotlib.pyplot as plt
import numpy as np
import sys

image_fname,centroids_fname,out_fname = sys.argv[1],sys.argv[2],sys.argv[3]
z=np.loadtxt(centroids_fname) #load centroids

orig_pixels = plt.imread(image_fname)
pixels = orig_pixels.astype(float)/255.
# Reshape the image(128x128x3) into an Nx3 matrix where N = number of pixels.
pixels = pixels.reshape(-1, 3)
```

The image and the centroids initialization will be provided to you as an argument to your program. The run command to your program should be:

```
$ python ex1.py <image_path> <centroinds_init_path> <output_log_fname>
For example:
$ python ex1.py dog.jpeg cents1.txt out.txt
```

When displaying your compressed image you should get similar results to the following:

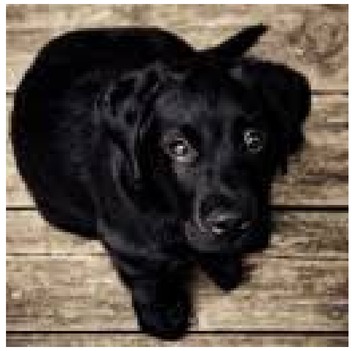

Figure 1: dog.jpeg

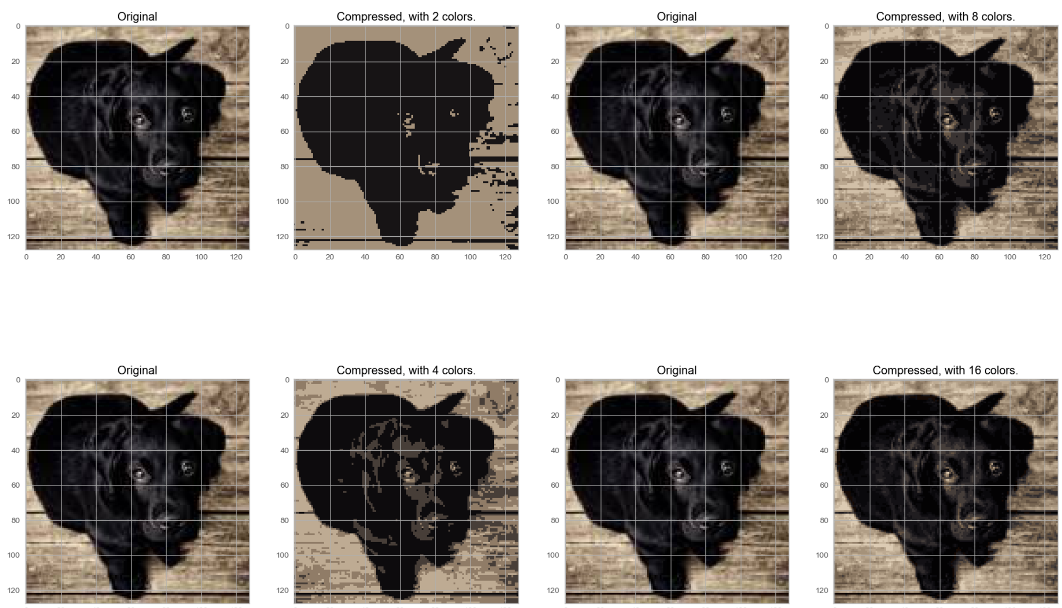

Figure 2: Results

Reproducibility. Originally, the initial centroids in k-means are randomly generated. For reproducible purposes we provided you with the centroids initialization. Moreover, In case when 2 centroids are evenly close to a certain point, the one with the lower index "wins". To evaluate your program outputs, we provided you with 2 examples - cents1.txt and cents3.txt for centroid initialization, and with out1.txt and out3.txt the requested outputs. Please note that given these pre-defined values, your sequence of centroid updates should be deterministic and not random in any way.

Your code should run for **20 iterations or until convergence**. We define convergence where all the centroids don't change. Your program should **create** a file named **<output_log_fname>(3rd arg to your program)**, consisting of your centroids after each centroid update.

For example, when using cents1.txt, the requested output should be:

```
[iter 0]:[0.1327 0.1135 0.1088],[0.6819 0.6071 0.5152]
[iter 1]:[0.1022 0.0879 0.0899],[0.6549 0.5801 0.4896]
...
[iter 7]:[0.0918 0.0793 0.0837],[0.6435 0.569  0.4796]
[iter 8]:[0.0918 0.0793 0.0837],[0.6435 0.569  0.4796]
```

```
use the following line to match your output to the requested format:
f"[iter {iter}]:{','.join([str(i) for i in new_z])}"
```

As you can see the algorithm converged and stopped(same centroids). For consistency and speedup purposes, **after each centroids update, use the built-in** *round*(4) **function on each dimension** - precision of 4 digits.

```
>>> i=np.array([0.123456,0.987654])
>>> i.round(4)
array([0.1235, 0.9877])
```

Figure 3: round(4)

Once submitted you will get a feedback email describing whether the expected output matches the requested format- check it and correct if needed. Part of your grade will consist of automatic checks - follow the format guidelines.

3 What to submit?

You should submit the following files:

- A `txt` file, named `details.txt` with your name and ID.
- A `PDF` file named `ex1.pdf` with your answers to 1.1 and 1.2.
- Python 3.6+ file named `ex1.py`. The main function writes the centroids updates to <output_log_fname> as explained above.
- A `PDF` report named `report.pdf` including the following plots: The average loss/cost value as a function of the iterations for $k = 2, 4, 8, 16$ (4 plots in total, x-axis iterations, y-axis avg/total loss). Explain shortly about your centroids initialization process. You can initialize the centroids as you wish in this part.

  Overall : `ex1.py, ex1.pdf, details.txt` and `report.pdf`

Good Luck!