Introduction to learning and analysis of big data
Exercise 3

Prof. Sivan Sabato

Fall 2022/3

Submission guidelines, **read and follow carefully**:

- The exercise **must** be submitted in pairs.
- Submit via Moodle.
- The submission should include two separate files:
  1. A pdf file that includes your answers to all the questions.
  2. The code files for the python question. You must submit a copy of the shell python file provided for this exercise in Moodle, with the required functions implemented by you. **Do not change the name of this file!** In addition, you can also submit other code files that are used by the shell file.
- Your python code should follow the course python guidelines. See the Moodle website for guidelines and python resources.
- Before you submit, **make sure that your code works in the course environment**, as explained in the guidelines. Specifically, **make sure that the test `simple_test` provided in the shell file works**.
- You may only use python modules that are explicitly allowed in the exercise or in the guidelines. If you are wondering whether you can use another module, ask a question in the exercise forum. No module containing machine learning algorithms will be allowed.
- For questions, use the exercise forum, or if they are not of public interest, send them via the course requests system.
- Grading: Q1(a): 8 points, Q1(b): 7 points Q1(c)-(e): 3 points each. Q2: 16 points. Q3-Q8: 10 points each

Question 1. (Do this question after we learn about k-means in class)

(a) Implement the k-means heuristic algorithm for Euclidean metric which we learned in class. The function should be implemented in the submitted file called "kmeans.py". The first line in the file (the signature of the function) should be:

```
def kmeans(X, k, t)
```

The input parameters are:

- k - the number of clusters
- t - the number of iterations to run
- X - a 2-D matrix of size $m \times d$. Row i in this matrix is a vector with d coordinates that describes example x_i from the training sample.

The function returns the variable C, which is a column vector of length m, where $C(i) \in \{1, \ldots, k\}$ is the identity of the cluster in which x_i has been assigned.

(b) Implement the single-linkage algorithm that we learned in class, again using the Euclidean metric. The function should be implemented in the submitted file called "singlelinkage.py". The first line in the file (the signature of the function) should be:

```
def singlelinkage(X, k)
```

The input parameters are:

- k - the number of clusters
- X - a 2-D matrix of size $m \times d$. Row i in this matrix is a vector with d coordinates that describes example x_i from the training sample.

The function returns the variable C, which is a column vector of length m, where $C(i) \in \{1, \ldots, k\}$ is the identity of the cluster in which x_i has been assigned.

(c) Run your k-means code on an **unlabeled** random sample of size 1000 generated from all the digits in the MNIST data file `mnist_all.mat`, with $k = 10$. Use the resulting clustering and the true labels of the points in the sample, to provide a table showing, for each cluster, (1) what is its size (2) which label is most common in it, and (3) what percentage of the points in the cluster have this label. Report the classification error on the sample, that would result if we classified all the points in each cluster using the cluster's most common label. Explain your calculation.

(d) Repeat (c) for your single linkage algorithm, again reporting the table and the classification error. Which clustering algorithm worked better for this problem?

(e) Run k-means and single-linkage again on the same data set from MNIST, this time with $k = 6$. Again provide the table of results and the classification errors. Considering the way the two algorithms work, explain the differences in their results when moving from $k = 10$ to $k = 6$.

Question 2. Implement the ridge-regression algorithm. **No need to submit your code**. Run the algorithm on the dataset `regdata.mat` provided on the course web page, you may use the following code to load the math file with python:

```
import scipy.io as sio
data = sio.loadmat('regdata.mat')
```

Run the regression using $\lambda \in \{0, 1, 2, \ldots, 30\}$ on the training set X, Y provided in the data file. Try training-set sizes between 10 and 100. For each training set size that you try, find the value of λ that obtains the smallest mean-squared-error (the average squared loss) on the test set provided in the data file.

(a) Submit a plot of the value of λ that minimizes the mean squared error on the test set as a function of the training set size m.

(b) What trend to you expect in the plot based on what we learned in class? Explain.

(c) Did you get this trend in the plot you submitted? If there are any differences, explain why they could occur.

(d) In this data set, the label y of each example x was generated by setting $y = \langle w, x\rangle + \eta$, where w is a fixed vector which is the same for all examples in the data set, and η is a standard Gaussian random variable, $\eta \sim N(0, \sigma)$ for some $\sigma > 0$. η is drawn independently for each example in the data set. What is the Bayes-optimal predictor for this problem with respect to the squared loss? And how about the absolute loss? Prove your claims.

Question 3. Let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Consider a Gradient Descent algorithm that attempts to minimize the following objective:

$$\text{Minimize}_{w\in\mathbb{R}^d}\ \lambda\|w\| + \sum_{i=1}^{m}(\langle w, x_i\rangle - y_i)^2.$$

(a) Suppose that Gradient Descent is run on S with a step size η. Calculate the the formula for $w^{(t+1)}$ as a function of $w^{(t)}$ and η. Explain the steps of your derivation.

(b) What would the update step for $w^{(t+1)}$ be in Stochastic Gradient Descent for the same objective?

Question 4. Consider a fully-connected feed-forward neural network architecture with the following number of neurons in each layer:

- Input layer: 4
- 1st hidden layer: 2
- 2nd hidden layer: 5
- output layer: 3.

The activation function used in the network is the sigmoid. The output label is determined using the multiclass ψ that we saw in class.

(a) Draw a graph $G = (V, E)$ that describes the neural network architecture.

(b) What is the input domain $\mathcal{X}$?

(c) What is the set of output labels $\mathcal{Y}$?

(d) Write down the hypothesis class described by this network architecture.

Question 5. Define the **depth** of a tree as the maximal path length from the root to a leaf. Let $\mathcal{X} = [0, 1]^d$ and $\mathcal{Y} = \{0, 1\}$. Let $\bar{\mathcal{H}}_n \subseteq \{0, 1\}^{\mathcal{X}}$ be the hypothesis class consisting of decision trees with depth at most n and binary attribute tests of the form "$x(i) \geq \theta$?", for $\theta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$.

(a) Prove that $|\bar{\mathcal{H}}_n| \leq (3d + 2)^{2^{n+1}}$.

(b) Danny got a random sample $S \sim \mathcal{D}^m$, and ran the ID3 algorithm with pruning, getting a decision tree T_S of depth k. Danny claims that given $\delta, \epsilon \in (0,1)$, by setting m to be large enough, it is possible to make sure that with a probability of $1-\delta$ over the choice of S,

$$\mathrm{err}(T_S, \mathcal{D}) \leq \inf_{T \in \mathcal{H}_k} \mathrm{err}(T, \mathcal{D}) + \epsilon.$$

Is Danny correct? Explain.

Question 6. Consider the following distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \{-1,1\}^2$ and $\mathcal{Y} = \{-1,1\}$.

x(1)	x(2)	y	prob.
-1	-1	$+1$	$6/60$
-1	$+1$	$+1$	$24/60$
$+1$	-1	$+1$	$2/60$
$+1$	$+1$	$+1$	$8/60$
-1	-1	-1	$5/60$
-1	$+1$	-1	0
$+1$	-1	-1	$11/60$
$+1$	$+1$	-1	$4/60$

(a) Does the Naive-Bayes assumption hold for this distribution? Prove your claim.

(b) Suppose we had a sample $S \sim \mathcal{D}^m$, such that the frequencies of the possible (x, y) in the data set was *exactly* the same as in the distribution, and suppose that we then ran the Naive-Bayes algorithm on this data set. What predictor would we get from this algorithm? Prove your claim.

Question 7. (Do this question after we learn about PCA in class) In an experiment, several measurements were taken at times $t = 1, 2, \ldots, m$. At each time t, the measurements taken were $x_t(1), x_t(2), x_t(3), x_t(4)$. This created a data set $S = x_1, \ldots, x_m$, where x_t is a vector in $\mathbb{R}^4$ which includes all the measurements from time t. PCA was performed on the data set S to reduce its dimensionality from 4 to 2.

(a) In one experiment, it turned out that in all times t, $x_t(3) = 3x_t(1) + x_t(2)$, and $x_t(4) = 2x_t(2) - 4x_t(3)$. What will be the distortion of the PCA in this case? Prove your claim.

(b) In another experiment, it turned out that at all times t, $x_t(3) = (x_t(1))^2 + (x_t(2))^3$, and $x_t(4) = (x_t(3) - x_t(1))^2$. Show an example of experiment results that satisfy these equations such that the distortion of the PCA is larger than the distortion you showed for the experiment in (a). You may choose m as you like.

Question 8. (Do this question after we learn about Maximum Likelihood estimation in class) Let $\mathcal{X} = \{0, 1, 2\}$. Let $\Theta \subseteq [0,1]^3$ such that for $\theta \in \Theta$, $\theta(1) + \theta(2) + \theta(3) = 1$. Define a *Trinomial* distribution $\mathcal{D}_\theta$ for $\theta \in \Theta$ as follows: $\mathbb{P}_{X \sim \mathcal{D}_\theta}[X = i] = \theta(i)$. Assume that we have a sample $S = x_1, \ldots, x_m \sim \mathcal{D}_\theta^m$.

(a) Let $\Theta' = \{\theta \in \Theta \mid \theta(1) = 3\theta(2)\}$. Give an explicit formula for the value of the maximum likelihood estimator $\hat{\theta}$ using $x_1, \ldots, x_m$, assuming that $\theta \in \Theta'$. Prove your claim.

(b) Consider a distribution which is a mixture of k densities, each density coming from $\{f_\sigma \mid \sigma > 0\}$, where f_σ is the density of a Gaussian random variable $N(1, \sigma^2)$.

- Write down a parametrized expression for the mixture distribution. Define a parameter set Θ which includes all (and only) the possible parameter settings of this mixture distribution.
- Define a multinomial random variable Z over $\{1, \ldots, k\}$. Suppose that we get an augmented sample $(x_1, z_1), \ldots, (x_m, z_m)$, with $S = (x_1, \ldots, x_m)$, $Z = (z_1, \ldots, z_m)$. Write down the augmented log-likelihood $L(S, Z; \theta)$, where $\theta \in \Theta$, and derive the maximum-likelihood estimator for θ, assuming that both S and Z are given.