Problem Set #1

ECE 4424 / CS 4824 - Machine learning - Fall 2020
Virginia Tech

August 30, 2020

- Feel free to collaborate with other classmates in doing the homework. Please indicate your collaborators with their student ID. You should, however, write down your solution yourself. Please try to keep the answers brief and clear.
- Whenever you need clarification, please post the related questions on Piazza under the corresponding homework folder.
- Total: 100 points
- **Due date: 9/18/2020, 5PM ET**
- Late submission: each student will have a total of **four** free late (calendar) days to use for homeworks. Once these late days are exhausted, any assignments turned in late will be penalized 20% per late day. However, no assignment will be accepted more than three days after its due date. Each 24 hours or part thereof that a homework is late uses up one full late day.

1 Supervised learning

Question 1.1 (10 points): Imagine you are a machine learning consultant for an online movie streaming company. Due to COVID-19, they have observed an upward trend of customer registrations. However, they also realized that a growing number of their existing customers have cancelled their subscriptions to use a competitor's product, a problem known as *customer churn*. As a consultant, you are asked to suggest a way in which **supervised learning** could be used to address this issue. For each proposed method:

- Describe the supervised learning setup: (2 pts) what is the data you need (you can be as imaginative as possible, as long as the data can be realistically and legally collected); (4 pts) what is the feature space $\mathcal{X}$, label space $\mathcal{Y}$, training loss function $\ell(\cdot, \cdot)$, and hypothesis space $\mathcal{H}$; (1 pt) and whether it is a classification or regression problem.
- (3 pts) Describe how the model will be used by the business and how it can address the customer churn issue.

2 K-nearest neighbor

Question 2.1 (15 points): Email spam detection models often rely on a bag-of-words representation, which describes an email based on how many times a particular word occurs. A dictionary consists of all the words that have appeared in the training set. Thus, a dictionary of size d corresponds to a feature vector of dimension d. The table below lists the bag-of-words representation for the following five emails and labels that indicate whether they are spam emails or not.

Training data: "money, money, money"–spam; "free money for free gambling fun"–spam; "gambling for fun"–spam; "machine learning for fun, fun, fun"–not spam; "free machine learning"–not spam

	Bag-of-words							
ID	money	free	for	gambling	fun	machine	learning	spam
1	3	0	0	0	0	0	0	true
2	1	2	1	1	1	0	0	true
3	0	0	1	1	1	0	0	true
4	0	0	1	0	3	1	1	false
5	0	1	0	0	0	1	1	false

Now, consider the query "machine learning for free". Answer the following questions and show the calculation.

a) (3 pts) What would a 1-nearest neighbor model using Euclidean distance predict?

b) (3 pts) What would a 3-nearest neighbor model using Euclidean distance predict?

c) (3 pts) What would a 3-nearest neighbor model using Manhattan distance predict?

d) (3 pts) The bag-of-words representation is sparse (i.e., there are a lot of zero entries). The cosine similarity between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by:

$$d_{\text{cosine}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|},$$

where $\boldsymbol{x} \cdot \boldsymbol{y}$ represents the inner product, and $\|\boldsymbol{x}\|$ is the ℓ_2-norm (i.e., length) of the vector $\boldsymbol{x}$. This distance is often a good choice when dealing with sparse non-binary data. What would a 3-nearest neighbor model using cosine similarity predict?

e) (3 pts) Define a distance function that is different from the above. Show that a 3-nearest neighbor model using this distance predicts the label `not spam`.

3 Probability and linear algebra review

Question 3.1 (10 points): Two dice are thrown. Let E be the event that the sum of the dice is even, let F be the event that at least one of the dice lands on 6 and let G be the event that the numbers on the two dices are equal. Find $\mathbb{P}(E)$, $\mathbb{P}(F)$, $\mathbb{P}(G)$, $\mathbb{P}(E \cup F)$, $\mathbb{P}(E \cap F)$, $\mathbb{P}(F \cup G)$, $\mathbb{P}(F \cap G)$. Note: $\mathbb{P}(E \cup F)$ indicates the probability of either E or F happens; $\mathbb{P}(E \cap F)$ indicates the probability of both E and F happen.

Question 3.2 (10 points): A pair of dice is rolled until either the two numbers on the dice agree or the difference of the two numbers on the dice is 1 (such as a 4 and a 5, or a 2 and a 1). Find the probability that you roll two dice whose numbers agree before you roll two dice whose numbers differ by 1.

Hint: Let E_n denote the event that the numbers agree on the nth roll and that on the first $n-1$ rolls the dice neither agree nor differ by one. Compute $\mathbb{P}(E_n)$ and argue that $\sum_{n=1}^{\infty} \mathbb{P}(E_n)$ is the desired probability. Then calculate this sum.

Question 3.3 (10 points): Urn A has 99 red balls and 1 green ball. Urn B has 1 red ball and 99 green balls. An urn is picked at random and a ball is chosen from it. If the ball is red, what is the probability that it came from urn A?

Question 3.4 (10 points): Suppose the matrix $\boldsymbol{A}$ has orthogonal columns $\boldsymbol{w}_1, ..., \boldsymbol{w}_n$ with lengths $\sigma_1, ..., \sigma_n$. Recall that SVD of $\boldsymbol{A}$ is of the form $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Please specify what are $\boldsymbol{U}, \boldsymbol{\Sigma}$ and $\boldsymbol{V}$.

4 Programming assignment: KNN (35 points)

For the following programming assignment, please download the datasets and iPython notebooks from Canvas and submit the following:

- Completed and ready-to-run iPython notebooks. Note: we will inspect the code and run your notebook if needed. If we cannot run any section of your notebook, you will not receive any points for the task related to that section.
- Responses (texts, codes, and/or figures) to the following problems/tasks

In this programming exercise, you will build a KNN classifier and apply it to a handwritten digit dataset (MNIST).

Task P1 (2 pts): Complete the code section to calculate the Euclidean distance. Copy the corresponding code here.

Task P2 (7 pts): Complete the code sections for `find_KNN` and `KNN_classifier`. Copy the corresponding code here.

Task P3 (2 pts): Find one example of success case and one example of failed case for 1-nearest neighbor (i.e., $K = 1$). Print the outputs of the code and copy them here.

Task P4 (4 pts): What is the error of 3-nearest neighbor classifier with Euclidean distance? How long does it take? (also report the specs of the computer used to run the program)

Task P5 (4 pts): Complete the definition of `manh_dist` and copy the code here. What is the error of 3-nearest neighbor classifier with Manhattan distance? How long does it take?

Task P6 (8 pts): Define your own distance function and write down the mathematical definition. Copy the code here. What is the error of 3-nearest neighbor classifier with Manhattan distance? How long does it take? Note: you will only get full points if the self-defined distance function can improve over the Euclidean distance in terms of accuracy (worth 2 pts).

Task P7 (8 pts): Implement the 5-fold cross validation to choose the best K (number of nearest neighbors) between 1 and 10 for KNN with **Euclidean distance**. Copy the code in the solution file and plot the 5-fold validation error with respect to K. Also plot the test error on the same figure. Which K would you choose? What are some other observations you can make?