COMP5511: Artificial Intelligence Concepts

Assignment

Spring 2021

Answers Submission Due: 23:59, Apr 18, 2021.

Important Notes.

- This is an *individual* assessment. So, no discussion (in any forms) is allowed among classmates.
- You can use any programming language you like. For easy checking, we suggest you provide solutions with one of the following: Python, Java, or C++. If you do not select the suggested language, please email and let us know what language you will use and how we can run the code by *18:00, Apr 12, 2021*. Otherwise, it will be assumed that you take our suggestions.
- Please submit the compressed folder (in zip or rar) with all the question answers. Grateful if the folder can be named as the student ID, such as "20123456D.zip" or ""20123456D.rar". For the i-th question below, you may create a folder named as "Qi" (e.g., Q1), which contains the code answers for Q_i and the readme.txt file indicating how to run the code.
- It is highly recommended that the codes are well commented, so that the TA can easily read them. It is for the case that the implementation is imperfect (with bugs) and we need to somehow find scores from the codes to see if your algorithm is designed in a correct way.
- The compressed folder should be submitted to the *blackboard*.[1] The full mark is 100' and submission entry is: Assessments/Assignment. For Question 2 and 3 below, we'll also provide the input data in the form of the compressed folder "Assignment Data" available in the same entry. You'll find two folders there, one named as Q2 and the other Q3, for the input of Question 2 and 3 respectively. As for Question 1, we won't provide the testing

[1] `learn.polyu.edu.hk`

cases but their forms is provided in the question description below. No late submission is allowed and don't forget to double check if the submission is saved successfully before leaving.

- When handling the paths for file loading and saving, please use relative paths for the TAs to run your codes easily in a different environment.

- Last but not least, best of the luck for this assignment! :)

Question 1. Recall that in Quiz 1, we handle the following question with A* algorithm. As it is said in the story, a traveler wants to travel from a site to another. The map can be represented as a graph, such as the one shown in Figure 1, while this time we generalize the problem into a map with N nodes, where V_0 denotes the start point while V_N the target point. Suppose that traveling from node V_i to V_j will result in the cost of $C(V_i,V_j) = (i+j)\cdot W(V_i,V_j)$, where the weight $W(V_i,V_j)$ is denoted as edge weight of V_i and V_j (such as the numbers appearing near the edges linking two nodes in Figure 1). This is defined as the cost of the edge V_i-V_j. For example, the weight of edge V_2-V_4 is 4 and the cost traveling from V_2 to V_4 should be $(2+4)\cdot 4 = 24$, so does the cost from V_4 to V_2. The cost of the path $V_{i_1}V_{i_2}...V_{i_k}$ (where $i_1,i_2,...,i_k \in \{0,1,2,...,N\}$ and are distinct with each other[2]) is the sum of cost of the $k-1$ involving edges, i.e.,

$$\sum_{j=1}^{k-1} C(V_{i_j},V_{i_{j+1}}) = \sum_{j=1}^{k-1} (i_j + i_{j+1})\cdot W(V_{i_j},V_{i_{j+1}}) \tag{1}$$

For example, the cost through the path $V_0V_1V_4V_7$ will result in the cost:

$$(0+1)\cdot 1 + (1+4)\cdot 5 + (4+7)\cdot 4 = 70$$

Our goal is to help the traveler find a path from V_0 to V_N that exhibits the smallest cost. The input describing how the graph looks like is an file in "input.txt", which can be considered to be put in the same folder of the code file (with the main function). The first line is N (the number of nodes on the graph map), followed by the lines indicating the edge weights, where each line shows node ID in one side (i), node ID in the other side (j), and the weight of the edge ($W(V_i,V_j)$), separated with the spaces " ". For example, here comes the input describing the graph in Figure 1:

[2]By definition, a vertex on the graph appears at most once on a path.

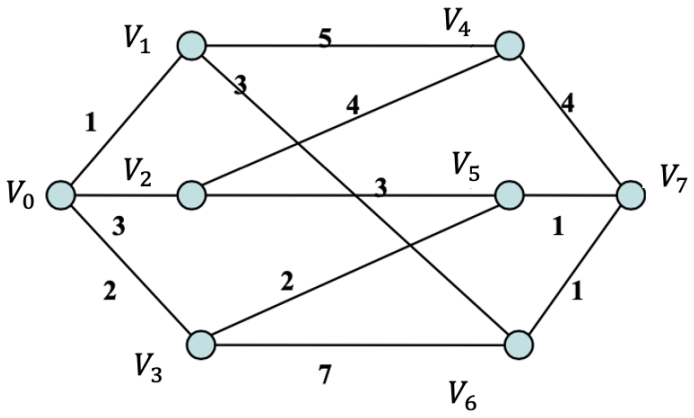

Figure 1: A graph representing the traveler's map. Vertices (nodes on the graph) are in blue and the numbers near the edges denote the weights of the edges. For example, the weight of V_2-V_4 is 4, i.e., $W(V_2, V_4) = 4$.

```
7
0 1 1
0 2 3
0 3 2
1 4 5
1 6 3
2 4 4
2 5 3
3 5 2
3 6 7
4 7 4
5 7 1
6 7 1
```

The expected output is one line showing the nodes forming the goal path (with the smallest cost in sum), starting with 0 and ending with N, displayed by their node IDs separated by the space " ". For example, as you have probably known, the goal path corresponding to the graph in Figure 1 is $V_0 \to V_3 \to V_5 \to V_7$. So the output is "0 3 5 7". You can simply print the output line on the screen at the end of the computing process. If there are more than one goal paths, you can print any one of them.

Note. Please use A* algorithm for implementation. Uninformed search or brute-force solutions can only earn half of the score. (30')

Question 2. Sentiment classification for tweets are very helpful for people to understand the public opinions from users. Here, we will implement a Naive Bayes classifier to classify the sentiment polarity of the tweets. In the class, we discussed the case with only positive and negative polarity, you can generalize that to the classification problem involving three classes — positive, negative, and neutral, where neutral means that the tweet does not exhibit subjective polarity of positive or negative, such as "*I did the laundry on the weekend.*", which tends to be objective. We've released the training data and validation data, named as "Train.tsv" and "Valid.tsv", where you can learn the model on the training data and examine the model performance on the validation data. For both files, each tweet record is presented in a line, separated into "tweet ID", "user ID", "sentiment label" (positive, neutral, or negative), and "Tweet text" by tab ($\backslash t$). To help us understand your model performance, we also provided the test file ("Test.tsv") with the "sentiment label" field hidden. So, there are only three fields, "tweet ID", "user ID", and "Tweet text" separated by the tab ($\backslash t$).

Please implement Naive Bayes model from scratch (without using any external packages or libraries), train the parameters (priors and likelihoods) on the training data, examine the preliminary performance on the validation data, and use the trained model to predict the sentiment labels for the test tweets.

Here in addition to the codes and the readme.txt file, it is required to submit prediction results in a file named as "prediction.tsv" put on the same directory with the codes. The "prediction.tsv" should contain 3,217 lines, where each shows one of the sentiment labels ("positive", "neutral", or "negative"); for the i-th line, it contains the prediction label for the test tweet on the i-th line of "Test.tsv"

Note. In readme.txt, besides the words describing how to run the training and testing codes, please also indicate the classification accuracy obtained by your model on the validation set. (50')

Question 3. We learned the K-means clustering methods in the class. You can implement the algorithm from scratch (without using external packages or libraries) and group the points in 3 clusters. The input points are put in a file named as "Points.csv": each line shows a 2D vector (point) where the first and second entry are separated by the comma ",". When running the codes, it is required to generate a figure visualizing the clustering results. In visualization, you can first draw a scatter plot with the input vectors (x-axis corresponding to the first entry while y-axis the second entry), and then color each point in red, green, and blue, indicating the cluster it belongs to.

Note. It is assumed that the "Points.csv" file is put in the same folder as the codes (with the main function). In K-means implementation, everything should be handled from scratch, while for visualization purpose, you can attach any graphics

or visualization packages you want. For initialization, the representatives of the three classes are $(40, 40)$, $(100, 0)$, and $(0, 100)$, respectively. (20')