Name:	Mark Gameng
IIT ID Number:	A20419026

CS 579: Online Social Network Analysis

Homework I - Graph Essentials, Data Mining

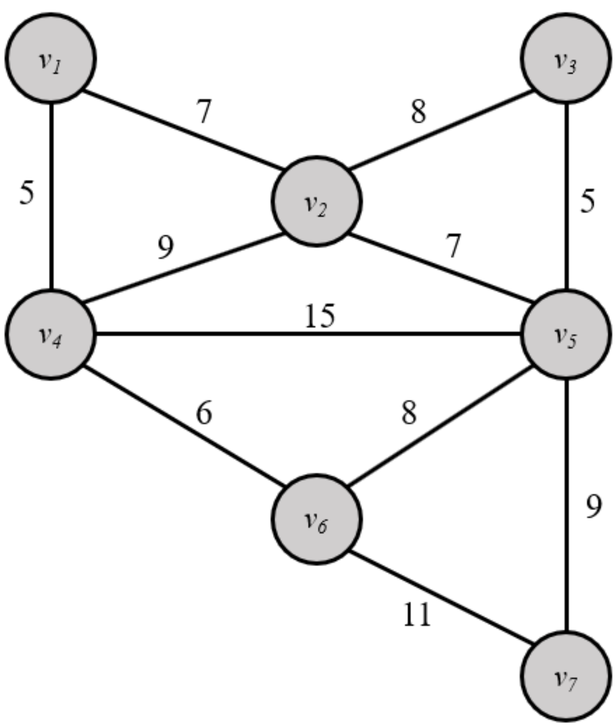
Prof. Kai Shu
Due at 2021 Feb. 7th, 11:59 PM

This is an ***individual*** homework assignment. Please submit a digital copy of this homework to **Blackboard**. For your solutions, even when not explicitly asked, you are supposed to concisely justify your answers.

1. **[Graph Algorithms]**

   (a) Compute the shortest path between v_1 and other nodes using Dijkstra's algorithm for the following graph.

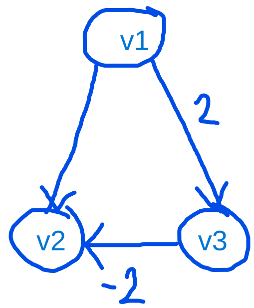
v4 visited -> 5
v2 visited -> 7
v6 visited -> 11
v5 visited -> 14
v3 visited -> 15
v7 visited -> 22



Node	**Distance from v_1**
v_2	7
v_3	7 + 8 = 15
v_4	5
v_5	7 + 7 = 14
v_6	5 + 6 = 11
v_7	5 + 6 + 11 = 22

Name:	Mark Gameng
IIT ID Number:	A20419026

(b) In the space below, draw a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers.



Using Dijkstra's algorithm starting from v1...

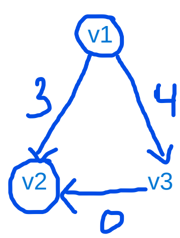
v2 = v1 -> v2 = 1
v3 = v1 -> v3 = 2

However, shortest path to v2 is incorrect and should be...

v2 = v1 -> v3 -> v2 = 2 + -2 = 0

(c) Argue whether "Algorithm 1" below always produces the shortest paths from one source node to others for graphs that have negative weights but do not have negative cycles.

Algorithm 1 basically makes all the weights non-negative by subtracting each of the weights with the most negative weight (min weight). Then it uses the original Dijkstra algorithm with the modified weights to find shortest paths.
However, this is still incorrect with negative weights. Using the example shown above, algorithm 1 still produces the incorrect shortest path. Using the algorithm below, the weights becomes ...



Algo results in
v2 = v1 -> v2 = 3
v3 = v1 -> v3 = 4

However, as shown above, the correct shortest path is
v2 = v1 -> v3 -> v2
v3 = v1 -> v3

Another example ... with v1 to v3 and only 2 paths
v1 -> v2 (-4) -> v3 (6) ... cost = 2
v1 -> v4 (-3) -> v5 (2) -> v3 (2) ... cost = 1
so 2nd one is shortest

However ... using algorithm 1
v1 -> v2 (0) -> v3 (10) ... cost = 10
v1 -> v4 (1) -> v5 (6) -> v3 (6) ... cost = 13
now, 1st one is shortest

Algorithm 1: Dijkstra Algorithm for graphs with negative weights.

Input : Adjacency Matrix M, Source node s.
Output: Shortest Path from s to other nodes.

$C \leftarrow$ Find minimum weight in M
for all i and j:
    $M[i, j] \leftarrow M[i, j] - C$
return Dijkstra(M, s) *// use the original Dijkstra algorithm to find the shortest paths*

2. **[Network Algorithms]** For a real-world social network, is BFS or DFS more desirable? Why? Provide details.

I think BFS is more desirable in a social network as most people would want to find and see people that are 'closer' to them. For example, in Facebook, I want to see my suggested friends/pages to be closely related to my friends or pages that I already followed. I wouldn't want a friend suggestion to be a random person that isn't even a friend of any of my friends. I remember seeing that social media and people nowadays have an average degree of separation of 3 or 4, somewhere around there. That means that all people are 4 or fewer social connections away from each other.
Ideally, in Facebook for example, I would only want my suggested friends to be 1 or 2 social connections away from me. In this case, BFS is much better and more desirable. BFS, when looking at it this way, starts at the root and explores all the nodes at the present social connection level before moving on to the next connection level. So, starts off with people at 1st degree, people that you know, and then to the 2nd degree, everyone that they know.

Name:	Mark Gameng
IIT ID Number:	A20419026

3. **[Decision Tree and Data Types]** Consider the given dataset below. Answer the following questions:

Instance	Age	Income	Student	Credit Rating	Buy Computer
1	25	High	No	Fair	No
2	20	High	No	Excellent	No
3	32	High	No	Fair	Yes
4	45	Medium	No	Fair	Yes
5	41	Low	Yes	Fair	Yes
6	41	Low	Yes	Excellent	No
7	36	Low	Yes	Excellent	Yes
8	27	Medium	No	Fair	No
9	30	Medium	Yes	Fair	Yes
10	42	Medium	Yes	Fair	Yes
11	29	Medium	Yes	Excellent	Yes
12	31	Medium	No	Excellent	Yes
13	33	High	Yes	Fair	Yes
14	41	Medium	No	Excellent	No

(a) Specify the data types (Nominal, Ordinal, Interval, Ratio) for each of the four attributes (Age, Income, Student, Credit Rating) in the given data.

	Age	Income	Student	Credit Rating
Data Type	Ratio	Ordinal	Nominal	Ordinal

meaningful zero point

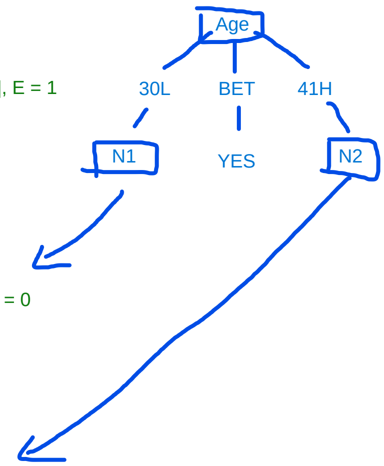
(b) Now assume that we have discretized the real-value "Age" attribute into three categories: 1) 30L: "Age" ≤ 30, 2) 41H: "Age" ≥ 41, and 3) BET: $31 \leq$ "Age" ≤ 40. What is the new data type for the "Age" attribute given this change?

	Age
Data Type	Ordinal

Intrinsic order to them
30L < BET < 41H

Name:	Mark Gameng
IIT ID Number:	A20419026

(c) Using the ID3 algorithm that we discussed in the class, generate the decision tree for the given dataset. Assume that "Buy Computer" attribute is the class label and the "Age" attribute is discretized as we discussed in previous question. Note that there could be more than one tree that fits the same data and we only need one! Show all your work for each step in making decision tree and explain how you select decision tree nodes and branches.



Root: [9, 5], E = 0.94
Age30L [2, 3], E = 0.97 | AgeBET [4, 0], E = 0 | Age41H [3, 2], E = 0.97
Age, Gain = 0.94 - (5/14)0.97 - 0 - (5/14)0.97 = 0.247
IncomeLow [2, 1], E = 0.92 | IncomeMedium [5, 2], E = 0.86 | IncomeHigh [2, 2], E = 1
Income, Gain = 0.94 - (3/14)0.92 - (7/14)0.87 - (4/14)1 = 0.022
StudentNo [3, 4], E = 0.98 | StudentYes [6, 1], E = 0.59
Student, Gain = 0.94 - (7/14)0.98 - (7/14)0.59 = 0.155
CreditFair [6, 2], E = 0.81 | CreditExcellent [3, 3], E = 1
Credit, Gain = 0.94 - (8/14)0.81 - (6/14)1 = 0.048

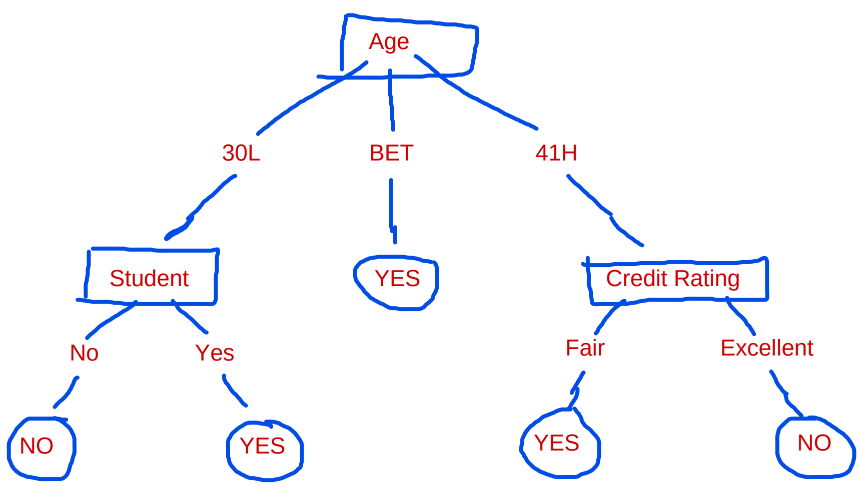
AGE as root, because it has highest gain which is 0.247

N1(30L): [2, 3], E = 0.97
IncomeLow [0, 0], E = 0 | IncomeMedium [2, 1], E = 0.92 | IncomeHigh [0, 2], E = 0
Income, Gain = 0.97 - 0 - (3/5)0.92 - (2/5)0 = 0.418
StudentNo [0, 3], E = 0 | StudentYes [2, 0], E = 0
Student, Gain = 0.97 - (3/5)0 - (2/5)0 = 0.97
CreditFair [1, 2], E = 0.92 | CreditExcellent [1, 1], E = 1
Credit, Gain = 0.97 - (3/5)0.92 - (2/5)1 = 0.018

N1, use Student, because it has highest gain which is 0.97

N2(41H): [3, 2], E = 0.97
IncomeLow [1, 1], E = 1 | IncomeMedium [2, 1], E = 0.92 | IncomeHigh [0, 0], E = 0
Income, Gain = 0.97 - (2/5)1 - (3/5)0.92 - 0 = 0.018
StudentNo [1, 1], E = 1 | StudentYes [2, 1], E = 0.92
Student, Gain = 0.97 - (2/5)1 - (3/5)0.92 = 0.018
CreditFair [3, 0], E = 0 | CreditExcellent [0, 2], E = 0
Credit, Gain = 0.97 - (3/5)0 - (2/5)0 = 0.97

N2, use Credit, because it has highest gain which is 0.97



Name:	Mark Gameng
IIT ID Number:	A20419026

4. **[Naive Bayes Classification]** Using the Naive Bayes algorithm and the table given in question 3, what would be the label for the following instance. Assume that "Buy Computer" attribute is the class label and the "Age" attribute is discretized as we discussed in 3.(b).

	Age	**Income**	**Student**	**Credit Rating**	**Buy Computer**
Instance 15	26	Low	Yes	Fair	?

P(Buy = Y | i15) = P(i15 | Buy = Y) * P(Buy = Y) / P(i15)
= P(Age = 30L, Income = Low, Student = Yes, CreditRating = Fair | Buy = Y)
* (P(Buy = Y) / P(i15))
= P(Age = 30L | Buy = Y) * P(Income = Low | Buy = Y) * P(Student = Yes | Buy = Y)
* P(CreditRating = Fair | Buy = Y) * (P(Buy = Y) / P(i15))
= (2 / 9) * (2 / 9) * (6 / 9) * (6 / 9) * ((9 / 14) / P(i15))
= (144 / 6561) * (9 / 14P(i15)) = 1296 / 91854P(i15)
= (8 / 567) * P(i15)

P(Buy = N | i15) = P(i15 | Buy = N) * P(Buy = N) / P(i15)
= P(Age = 30L, Income = Low, Student = Yes, CreditRating = Fair | Buy = N)
* (P(Buy = N) / P(i15))
= P(Age = 30L | Buy = N) * P(Income = Low | Buy = N) * P(Student = Yes | Buy = N)
* P(CreditRating = Fair | Buy = N) * (P(Buy = N) / P(i15))
= (3 / 5) * (1 / 5) * (1 / 5) * (2 / 5) * ((5 / 14) / P(i15))
= (6 / 625) * (5 / 14P(i15)) = 30 / 8750P(i15)
= (3 / 875) * P(i15)

Thus,

P(Buy = Y | i15) = (8 / 567) * P(i15) ~= 0.014 * P(i15)
P(Buy = N | i15) = (3 / 875) * P(i15) ~= 0.003 * P(i15)

P(Buy = Y | i15) > P(Buy = N | i15)
So, Buy Computer = YES

Name:	Mark Gameng
IIT ID Number:	A20419026

Notes