CS 59000SA **Fall 2018**

Assignment 1

Instructor. Ninghui Li

Due: 11:59pm Sep 7th, 2018

Please typeset your homework solutions, ideally using LaTeX. No hand written assignment will be graded.

1. (12 pts) Consider an experiment where a coin is tossed repeatedly.

- If the coin is a fair coin, what is the probability that the coin turns up 5 heads after 10 tosses?
- With a fair coin, what's the expected number of times you need to toss to observe head for the first time?
- With a coin that gives head with probability p, what's the probability that the coin turns up k heads after n tosses, where $n \geq k$.
- With a coin that gives head with probability p, what's the expected number of coin tosses to observe head for the first time?
- With a coin that gives head with probability p, let the random variable X be the number of heads minus the number of tails after n tosses. What is the expected value of X? What is the variance of X?

2. (8 pts) You are being tested for arachnophobia. It is known that 30% of people have arachnophobia. There are three spider pictures; they include a Goliath birdeater tarantula spider, a black widow spider, and a Brazilian wandering spider. Those who have arachnophobia shiver $\frac{9}{10}$ of the time when shown one of the pictures. Those who do not have arachnophobia shiver $\frac{1}{5}$ of the time they are shown a picture of a black widow. (And whether a person shivers or not when seeing one picture is independent from whether the person shivers or not when shown another picture.)

- (a) You are shown three pictures and never shiver. What is the probability you have arachnophobia?
- (b) You are shown three pictures and always shiver. What is the probability you have arachnophobia?
- (c)You are shown three pictures and shiver exactly twice. What is the probability you have arachnophobia?
- (d)You are shown three pictures and shiver exactly once. What is the probability you have arachnophobia?

3. (10 pts) Tom runs a blivet-making factory. Unfortunately, due to circumstances beyond his control, 10% of the blivets that he makes are defective. Defective blivets fail 25% of the time when one tests it, but good (non-defective) blivets never fail. How many times does Tom have to test a blivet that comes off his assembly line, in order for him to be 98% sure that the blivet is good?

4. (10 pts) Prove that the Jaccard Distance is a metric.

5. (10 pts) Consider the following variant of the Monty Hall problem. Suppose there are 7 curtains with 1 car and 6 goats behind them. The host knows which curtains are hiding the cars. The contestant chooses one of the curtains. The host randomly chooses two curtains that have goats behind them and reveal the curtain, and gives the contestant the opportunity to switch. Then the host randomly chooses two more curtains that have goats behind them and reveals them, and again gives the contestant the opportunity to switch. What's the probability of winning the car for each of the following strategies (a) does not switch; (b) switch after the first pair of goats are revealed but not after the second pair; (c) does not switch after the first pair, but switch after the second pair is revealed; (d) switch both times?

6.(10 pts) Find one example of data mining applications in recent news articles (like the Target pregnancy prediction models we discussed in class). Answer the following:

- . Briey summarize the article and include a reference.
- . Identify the data analysis task in the application.
- . Outline the hypothesis that prompted the analysis.
- . Describe the data that was analyzed.
- . If KNN is used for this, is it suitable? Justify your answer.

7.(40 pts) Python: SSH Attempt Analysis

Secure Shell(SSH) is a cryptographic network protocol for operating network service securely over an unsecured network. It provides a secure channel in a client-server architecture. In this assignment, you will be asked to write a program to analyze the behavior of SSH connections for a log.

Description of SSH log: SSH logs are files recording SSH connection activities. In this assignment you will be given a **.csv** file contains multiple SSH connection activities. Each line of the file represents a SSH connection with attributes in the following order: ts, uid, orig_host, orig_port, resp_host, resp_port, status, direction, client_string, server_string, and resp_size. ts(time stamp) is a floating point number represent when the SSH connection was detected. uid is a string represent a unique connection ID. orig/resp host/port represent the host IP address and service port of origin/response. Direction can be either "Outbound" or "Inbound". Client and server strings are software strings from the client and server indicating which SSH software were used. The final attribute, resp_size is a count represents the amount of data returned by the server. For example:

1.23 Ab3T 192.168.201.98 49495 192.168.28.254 22 failure INBOUND OPENSSH SSH-1.99-CISCO

This is a SSH connection activity captured at time stamp 1.23 with uid Ab3T. The host IP address and port of the origin is 192.168.201.98 and 49495. Similarly, 192.168.28.254 and 22 are the host IP address and port for the response. This **INBOUND** connection was **failed**. The SSH software used by the client and server are OPENSSH and SSH-1.99-CISCO correspondingly. The amount of data returned by the server attribute is missing.

Task 1: Simple Counting

Counting is one of the fundamental techniques used in data analysis. In this task, you are asked to implement a program that support counting(**task1.py**). The program takes **two input parameters**. The first parameter is a file contains SSH log as described above. The second parameter is one of the attribute names. To be more specific, it can be one of the followings:

- ts
- uid
- id.orig_h
- id.orig_p
- id.resp_h
- id.resp_p
- status
- direction
- client_string
- server_string
- resp_size

The program should count the occurrence of each unique value under the given attributes. For example, if the given attributes is "direction", then the program should output how many connections are Outbound and Inbound.

The output contains two parts: text output and figure output. For the text part, counted frequency will be printed line by line with the following format:

attribute_value frequency

The text output must also be sorted in descending order w.r.t. the occurrence of values. We break **ties** by the ascending alphabet order of the attribute values. For example, if both Outbound and Inbound show up 10 times, the program will output the following to break tie:

INBOUND 10
OUTBOUND 10

The output from the figure part will be a bar chart. The X and Y axis will be attribute values and their frequency counts. If there are more than 10 attributes values, plot the top 5 and bottom 5 bars (w.r.t. the frequency). The title of X axis should be the attribute name. Each figure should have a title name "attribute_name k" where k is the number of different attribute values. The color of the bar should be magenta.

Tips: Recall what we covered in the python tutorial lecture, dictionary data structure can be very useful in this task. Also you should look for a sorting algorithm from libraries instead of implement one by yourself.

Task 2: Counting with two attributes Sometime people care about the frequency of SSH connection satisfying multiple conditions. For example, failed connection may come from a particular SSH software. In this task, you are asked to implement a program (**task2.py**) that support counting with two attributes. The program takes the SSH file log as the only input. It outputs the result of all possible(55) pair-combination of attributes. To be more specific, for each pair, one should output attribute1_attribute2.txt and attribute1_attribute2.jpg. In total there will be 55 text files and 55 figures.

Text output is similar to task 1. Each line contains the frequencies of two attribute values. The attribute pair should be formed based on the list in Task 1. To be more specific, given two attribute types, the one listed first in the list will be attribute 1. The result will be sorted in descending order w.r.t. frequencies. Tie breaking is based on the ascending alphabet order of the attribute value 1, then attribute value 2 if necessary. For example, 5 failed Inbound SSH connection should be printed as the following:

failure INBOUND 5

Each **Figure output** is a **heatmap** with each cell represent the density of a specific attribute pairs. The X-Axis will be attribute 1 and Y-Axis will be attribute 2. Attribute values are sorted in ascending alphabet order on the axis. You can choose your favorite color for the heatmap as long as lower RBG value indicate lower frequency.

Hint: Using attribute values as keys for dictionary does not work in this task. You should consider a different way of hashing. For example, creating a dictionary of dictionaries.

Grading:

- Task 1: 40%
  - 10% Breaking ties
  - 15% Text output
  - 15% Figure output
- Task 2: 50%
  - 10% Breaking ties
  - 10% Text output
  - 10% Figure output
  - 10% File IO

- Coding standard: 10 %, your program should be easy to read. (Friendly variable names, sufficient comments, well structured, and etc)

Your program will be checked by a grading script first (mainly use "diff" to check the output). If any test fails, TA will check your program and output to assign partial credits. Partial credits are assigned based on the correctness of code instead of number of tests you can pass. For example, a program passing 8 out of 10 tests does not mean 80 points, it can be a lower score. On the other hand, a program passed 0 of the test due to a simple typo may deserve most of the points.