Data Mining

Homework 1

Due: 1/11/2020, 23:59

Instructions

You must hand in the homeworks electronically and before the due date and time.

The first homework has to be done by each **person individually**.

Handing in: You must hand in the homeworks by the due date and time by an email to Andrea (`mastropietro@diag.uniroma1.it`) that will contain as attachment **(not links to some file-uploading server!)** a .zip file with your answers. The filename of the attachment should be
`DM_Homework_1__StudentID_StudentName_StudentLastname.zip`;
for example:
`DM_Homework_1__1235711_Robert_Anthony_De_Niro.zip`.
The email subject should be
`[Data Mining] Homework_1 StudentID StudentName StudentLastname`;
For example:
`[Data Mining] Homework_1 1235711 Robert Anthony De Niro`.
After you submit, you will receive an acknowledgement email that your project has been received and at what date and time. If you have not received an acknowledgement email within 2 days after you submit then contact Andrea.

The solutions for the theoretical exercises must contain your answers either typed up or hand written clearly and scanned.

For information about collaboration, and about being late check the web page.

Problem 1. We shuffle a standard deck of cards, obtaining a permutation that is uniform over all 52! possible permutations.

1. Define a proper probability space Ω for the above random process. What is the probability of each element in Ω?
2. Find the probability of the following events:
   (a) The first three cards include at least one ace.
   (b) The first five cards include exactly one ace.
   (c) The first three cards are all of the same rank (they are the same number or both are J, or all three are Q, etc.)
   (d) The first five cards are all diamonds.
   (e) The first five cards form a full house (three of one rank and two of another rank).
3. (Optional) Develop some small programs in Python to perform simulations to check your answers.

Problem 2. You throw a set of 3 regular dice again and again, until for the first time you see a sum of 11 or a sum of 16.

- Design an appropriate probability space for the above process.

- What is the probability that you stop because you see a sum of 16?

Problem 3. There are various tests for detecting the SARS-CoV-2 virus with different characteristics. They differ in the accuracy, cost, equipment needed, how early they can detect the virus, and so on.

A test is characterized by its *sensitivity* and its *specificity*. Assume that we perform the test on a large representative fraction of the population. We define:

- True positives (TP): The number of people who *have* the virus and their test result was *positive*.
- False positives (FP): The number of people who *do not have* the virus and their test result was *positive*.
- True negative (TN): The number of people who *do not have* the virus and their test result was *negative*.
- False negative (FN): The number of people who *have* the virus and their test result was *negative*.

We define the sensitivity and the specificity of the test as:

$$\textit{Sensitivity} = \frac{TP}{TP + FN} \qquad \textit{Specificity} = \frac{TN}{TN + FP}.$$

The rapid test developed by the E25Bio company is an example an antigen test. It is very inexpensive, uses the saliva, and can be performed at home, returning the results in 15min. For the purpose of this exercise, let us assume that the sensitivity of the test is 84.7% and the specificity is 85.7%.[1]

Assume that 1% of the population has currently COVID-19. We take a person selected uniformly at random and we perform the aforementioned test. The result is positive.

What is the probability that this person is infected with COVID-19? First define an appropriate sample space, then the events required so that you can compute the probability, and then compute it.

What do you think is the usefulness of such a test? (There is not a unique clear answer; feel free to answer what you believe.)

Problem 4. The Erdős-Rényi $G_{n,p}$ random-graph model, is a mathematical model for creating random graphs. Fix a positive integer n and a value $p \in [0,1]$. Then a graph created according to the $G_{n,p}$ model, has n nodes, and each pair of distinct nodes is connected with an edge with probability p; all pairs being independent from each other.

1. Define an appropriate probability space Ω to describe the $G_{n,p}$ model. What is its size?
2. What is the probability of each element of Ω?
3. What is the probability that a graph created according to $G_{n,p}$ contains exactly two cycles of length $n/2$ and no other edges (assume that n is even for this)? A cycle is a sequence of vertices $v_1, v_2, \ldots, v_\ell, v_1$ ($\ell \geq 3$), with $v_i \neq v_j$ for $i,j \in \{1, \ldots, \ell\}$, such that each edge (v_i, v_{i+1}) exists in the graph (as well as (v_ℓ, v_1)).

[1] Numbers taken from `https://doi.org/10.1101/2020.09.01.20184713`, although these are only approximations to the sensitivity and the specificity of the test.

4. What is the probability that a graph created according to $G_{n,p}$ contains exactly two cycles of any length (the two cycles can have different length) with no nodes in common, and no other edges?

5. In a graph created by the $G_{n,p}$ model, what is the expected degree of a node?

6. In a graph created by the $G_{n,p}$ model, what is the expected number of edges?

7. We define a *house subgraph* to be a subgraph of 5 nodes, $v_1,\ldots,v_5$ in which the edges $\{v_1,v_2\}$, $\{v_1,v_3\}$, $\{v_2,v_3\}$, $\{v_2,v_4\}$, $\{v_3,v_5\}$, $\{v_4,v_5\}$ exist, and no other edge between them exists (see Figure 1). What is the expected number of house subgraphs in a random graph according to $G_{n,p}$?

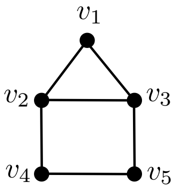

Figure 1: The *house subgraph.*

Problem 5. Quite often you can analyze your data just by using simple unix tools. Some usefull commands are the `grep`, `sort`, `uniq`, `cut`, `sed`, `awk`, `join`, `head`, `tail`, `wget`, `curl`. You can find more information using the `man` command or by checking the web. Shell scripting can help you even more.

As a simple exercise do a simple analysis of the reviews in
`http://aris.me/contents/teaching/data-mining-2020/protected/beers.txt`.
After you download and unzip the file, use some of the commands above to find the 10 beers with the highest number of reviews. (**Hint:** You can do it with a single command line, by chaining commands through pipes!)

Problem 6. We will now go one step further and start practicing with Python. Write a Python program to find the top-10 beers with the highest average overall score among the beers that have had at least 100 reviews. (You may need to preprocess the file first.)

Problem 7.

We continue getting familiar with Python by starting playing with some APIs. There are several libraries that we will use often, and eventually we should learn them: `requests` is used to download web pages, `beautiful soup` is used to parse them, if they are sufficiently small `lxml` or regular expressions (package `re`) are used for larger pages, and so on.

Here instead we will play a bit with Twitter and Google maps. Our goal is to download a stream of tweets in Rome as they are created and create a web page that displays their location on Google maps. For that we need to access two APIs, the Twitter streaming API and the Google maps API.

1. First we will obtain the stream of tweets as they are generated. There are various ways to access the Twitter API. The most direct is to use the package `python-twitter`, which gives

direct access to the API. However, there are several high-level packages, which hide several of the details. One of the easiest ones is the `twython` package. You are encouraged to use that one, but feel free to use whichever you prefer. One more possible library is `tweepy`.

To access the Twitter streaming API, you should register as a user, create a Twitter application, and generate four keys, which allow you to authenticate from your application. After you do that, you can use the `twython` library to set a query and listen for tweets that are being created in Rome (for that you need to define a bounding box to filter tweets and keep only tweets that are created in Rome). Thus the goal of this part is to grab the tweets in Rome as they are generated. In particular, we are interested in the location of the tweet, the screen name of the user, and the text of the tweet.

To learn more, you need to consult the documentation for `twython` and that of the Twitter API.

2. For the second part you are asked to plot the point locations on Google maps. A library for that is `gmplot`. Use this (or some other, if you prefer) library to save the points in an html file as points on a Google map, as they are being collected from Twitter. As title for each point use the

   `@screen_name:  tweet` .

   The title is the text displayed when the mouse goes over the point. In order to use `gmplot` you may need a Google maps API key. Use the guide here `https://developers.google.com/maps/documentation/javascript/get-api-key` to generate the key. This library is based on the original library `pygmaps`, which is available only on Google Archive and is no longer maintained.