Data Mining

Homework 1

Due: 23/10/2022, 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 Gianluca (`decarlo.1805894@studenti.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 Gianluca.

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. A family has two kids, each being a boy or a girl with probability 1/2 and born in a random day of the week.

1. Define a sample space sufficient to answer the third question and define the probabilities to the points in the sample space.

2. If we know that one kid is a girl, what is the probablity that the other kid is a girl?

3. If we know that one kid is a girl born on Sunday, what is the probablity that the other kid is a girl?

Problem 3. A group of n men and m women go to a Chinese restaurant and sit in a round table, such that each person has two other persons next to him/her.

1. Describe a sample space that describes the random process.

2. Find the expected number of men who will be sitted next to at least one woman.

Problem 4. In the class we saw a question about people getting random jackets. Here we will see a variation. Assume that n people enter a restaurant and they leave their umbrellas at the entrance. In their way out, people leave one by one. The first person being in a hurry, did not search for his umbrella in his way out, but picked one at random and left. From that point on, each person searches for his own umbrella. If he finds it he takes it and leaves. Otherwise, he takes a random one from the ones left (and leaves). The problem is to compute the probability that the last person gets his own umbrela. The answer is probably not obvious. We will find it in two ways.

1. First we will do simulations. Write a python program that tries different values of n and computes the probability. To compute the probability you will need, for each value of n, to perform several trials and see how many times you succeed. From the results of your simulations make a conjecture for the correct answer.

2. Try to prove your conjecture. Probably the simulations in the first part will help you. (There are at least two ways to solve this problem; one is more straightforward but requires a few calculations, instead the second needs a good argument.)

Problem 5. 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)).

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 $G_{n,p}$?

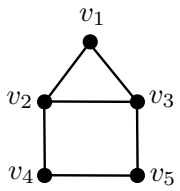

Figure 1: The *house subgraph*.

Problem 6. In this exercise we will get some practice in using Python and some libraries, for downloading web pages, parsing them, and performing some first analysis. We will obtain data about apartments available for rent in Rome.

For downloading the web pages you may use some package such as `Requests`. To parse the page you can either use regular expressions through the package `re` (it is anyway a good idea become familiar with regular expressions), or, probably better, use an HTML/XML parser. The `Beautiful Soup` package is a good one but it loads the whole file in memory. This is fine for this problem, since the pages to parse are small, but be careful if you want to use it on large XML files; for those ones check the `lxml` library.

Write a program that will download from `http://www.kijiji.it` and parse all the apartments for rent in Rome. Download regular and top announcements, but not sponsored ads. Save in a tab-separated value (TSV) file, for every apartment (one line per apartment), the *title*, *short description* (from the result page), *the location*, the *price*, the *timestamp* of the apartment announcement, and the *URL link* to its web page. **Because you will make a lot of calls to the kijiji site, make sure that you have a delay (use: `sys.sleep()`) between different downloads of kijiji pages, to avoid being blocked.**

After you download the web pages, compare the different locations by calculating for each of them:

1. The number of announcements.

2. The average apartment price.

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.