Data Mining

Homework 1

Due: 25/10/2015, 23:59.

Instructions

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

Handing in: You must hand in the homeworks by the due date and time by an email to the instructor that will contain as attachment (not links!) a .zip or .tar.gz file with all your answers and subject
`[Data Mining class] Homework #`
where # is the homework number. After you submit, you will receive an acknowledgement email that your homework has been received and at what date and time. If you have not received an acknowledgement email within 1 day after you submit then contact the instructor.

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

The solutions for the programming assignments must contain the source code, instructions to run it, and the output generated (to the screen or to files).

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

Most of the questions are not very hard but require time and thought. **You are advised to start as early as possible, to work in groups, and to ask the instructor in case of questions.**

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 two cards include at least one ace.
   (b) The first five cards include at least one ace.
   (c) The first two cards are a pair of the same rank (they are the same number or both are J, or both 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. Aris and Evimaria each pick a different page from the textbook IIR. Your goal is to find who chose the page that appears earlier in the book. Note that we do not choose the pages randomly; in fact, we may choose the pages adversarially so that we make your task really hard (for example, by choosing pages 1 and 2). However, to help you we give you the chance to ask one of us randomly (each with probability 1/2) what is the page number he/she has chosen. Note that it is easy to find a strategy with probability of winning exactly 1/2; for example, pick Aris or Evimaria with probability 1/2. Surprisingly, there are ways to achieve a probability of winning strictly greater then 1/2. Devise a strategy that does that.

Hint: Assume that you know that a given number x is between the two page numbers. Then you know that if the random person you picked reveals a number $y > x$ then you need to respond that the other person has the picked the smallest page number. Of course you do not really know x but this hint should help you.

Problem 3. 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. model.

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 size $n/2$ and no other edges (assume that n is even for this)?
4. What is the probability that a graph created according to $G_{n,p}$ contains exactly two cycles of any size and no other edges? (A single node is also a cycle.)
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, $v_1, \dots, v_5$ 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 $G_{n,p}$?

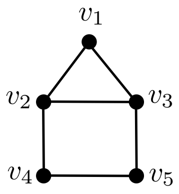

Figure 1: The *house subgraph.*

Problem 4. 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 the package `Requests` or the package `urllib2`. 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 and the tutorial at

`http://www.ibm.com/developerworks/xml/library/x-hiperfparse/`

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.