Data Mining

Homework 1

Due: 13/11/2016, 23:59.

Instructions—Read carefully!

You must hand in the homeworks electronically and before the due date and time.
You should do the homeworks in groups of 3 people. Please send the teams to Aris by email by 26/10. If you are not able to find a team within that date, send an email to Aris to create one or more teams.

Handing in: You must hand in the homeworks by the due date and time by an email to Aris 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. In the text you must also mention the team members. 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 2 days after you submit then contact the Aris.

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 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. A group of n men and m women go to a Chinese restaurant and sit in a round table, such that each person has to other person 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. For this question you have to implement a search engine for recipes. It has several parts that you need to implement. For the linguistic analysis you can use the NLTK Python library.

1. First you need to download the recipes. We will use the recipes at `http://www.bbc.co.uk/food/recipes`. You will need to find a way to download all the recipes from the site. You can use any method you want. It is important to put some time delay between requests; at least 1sec between two requests. For that you can use the `time.sleep` command of Python.
2. After you download them you have to preprocess them. For each recipe store all the information (title, who wrote it, preparation time, cooking itme, number of people it serves, dietary information, ingredients, and method). Note that some fields (e.g., dietary information) may be missing. Store the final output as a large single tab-separated file. After that, you can do whatever preprocessing you think is essential (e.g., stopword removal, normalization, stemming).
3. The next step is to build a search-engine index. First, you need to build an inverted index, and store it in a file. Build an index that allows to perform proximity queries using the cosine-similarity measure. Then build also a query-processing part, which, given some terms it will bring the most related recipes.
4. **Extra credit:** Use your imagination to think of features you would like such a service to have. For example, you may want to weigh different the ingredients based on the quantity. You may want to try to find methods that take care of ingredients that are written in different ways, refering though to the same thing. You may want to give different weights to the title–ingredients–method. You may want to provide a query that satisfies some people, such as, vegetarians, lactose intolerant, and so on. Other ideas might be on the presentation of the results. Or you can find photos from google images with final food, and so on.

Hand in the code, along with some examples of queries and screenshots of the results. Try short queries, such as a few terms, as well as long queries, which can be other recipes, where the goal would be, for example, to find the types of food that are more similar to *parmigiana di melanzane*.