Data Mining 2015 - Homework 1

Ludovico Fabbri 1197400

March 21, 2015

1 Problem 1

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 second question.
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?

1.1

Firstly i define two simple events:

- G: ”the kid is a girl”
- B: ”the kid is a boy”

Next we can define the outcomes of the sample space Ω:

$$\Omega = \{BB, BG, GB, GG\}$$

The outcomes of the sample space are equiprobable, so P(BB) = P(BG) = P(GB) = P(GG) = $\frac{1}{4}$.

Because we are interested in the sample space related to the second question, where we assume that at least one kid is a girl, the probability of the outcome (B, B) is equal to zero. Thus the sample space becomes:

$$\Omega = \{BG, GB, GG\}$$

The outcomes of the sample space are equiprobable, so P(BG) = P(GB) = P(GG) = $\frac{1}{3}$

1.2

Accordingly to the sample space defined in the previous section, the requested probability is equal to the probability of the outcome (G, G). Since the outcomes BG, GB, GG are equiprobable, the probability of the the outcome GG is $P(GG) = \frac{1}{3}$.

More formally we can derive the same result starting from the theorem of the conditional probability. Given two generic events A and B, the following expressions are respectively the probability of A to occur given that B occurs and the probability of B to occur given that A occurs:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{1}$$

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \tag{2}$$

From the (2) we can derive:

$$P(A \cap B) = P(B|A) \cdot P(A) \tag{3}$$

And substituting in the (1) we find the Bayes theorem:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{4}$$

We can also define the following event:

- G$_l$: at least one kid is a girl.
  $P(G_l) = P(B,G) + P(G,B) + P(G,G) = \frac{3}{4}$, this is a priori probability

Now using the (4) we can find the probability that given that one of the kids is a girl, the other one is also a girl:

$$P(GG|G_l) = \frac{P(G_l|GG) \cdot P(GG)}{P(G_l)} = \frac{1 \cdot \frac{1}{4}}{\frac{3}{4}} = \frac{1}{3} \tag{5}$$

where $P(G_l|GG)$ is the probability that at least one kid is a girl given that both are girls, which is obviously 1.

1.3

We can use again the (4):

$$P(GG|G_{sun}) = \frac{P(G_{sun}|GG) \cdot P(GG)}{P(G_{sun})} \tag{6}$$

The $P(G_{sun}|GG)$ is the probability that at least one kid is a girl born in Sunday given that the kids are 2 girls, that is equal at two times the probability that one girl is born in Sunday and the other girl is born in another day plus the probability that both the girls are born in Sunday. Thus:

$$P(G_{sun}|GG) = \frac{1}{7} \cdot \frac{6}{7} + \frac{1}{7} \cdot \frac{6}{7} + \frac{1}{7} \cdot \frac{1}{7} = \frac{13}{49} \tag{7}$$

$P(GG)$ is the probability that both kids are girls, which is simply:

$$P(GG) = \frac{1}{4} \tag{8}$$

$P(G_{sun})$ is the probability that at least one kid is a girl born on Sunday, and must be calculated on the entire sample space. So we have four cases to consider: GG(two girls), GB (one girl and one boy), BG (one boy and one girl), BB (two boys). All these outcomes are equiprobable and have probability of $\frac{1}{4}$. In the case of BB the probability is 0, there are no girls. In the case of GB and BG the probability is $\frac{1}{7}$, it's the probability for one girl to be born on Sunday. In the case of GG the probability is the probability of one girl to be born on Sunday plus the probability of the other girl to be born on Sunday minus the probability that both are born on Sunday, so is equal to $\frac{1}{7} + \frac{1}{7} - \frac{1}{7} \cdot \frac{1}{7} = \frac{13}{49}$. So we can calculate:

$$P(GG|G_{sun}) = \frac{P(G_{sun}|GG) \cdot P(GG)}{P(G_{sun})} = \frac{\frac{13}{49} \cdot \frac{1}{4}}{\frac{1}{4}(\frac{1}{7} + \frac{1}{7} + \frac{13}{49})} = \frac{13}{27} \tag{9}$$

2 Problem 2

You are in an airplane that falls in the jungle and you manage to survive. In the jungle there are are two tribes, the Randomukee and the Bugiardukee. The Randomukee are twice as many as the Bugiardukee. Each time you ask a question to a Randomukee he will say the truth with probability 3/4, whereas, each time you as a question to a Bugiardukee he will lie. As you try to find your way out of the jungle, you find a random person from the two tribes. You ask him the question "To get out of the jungle, I have to go left or right?"

1. Define an appropriate probability space that can be used to answer the questions that follow.

2. Assume that the person gave you the answer "right". What is the

probability that the answer is correct?

3. You ask the same person again, and he gives you the same answer. Show that the probability that the answer is correct is 1/2.

4. You ask a third time and you get again the same answer. What is now the probability that the answer is correct?

5. Finally, you ask a fourth time and you get again the answer ”right”. Show that the probability that the answer is correct is 27/70.

6. Assume that the first three times the answer was ”right” but that the fourth one it was ”left”. Show that the probability that the correct answer is ”right” is 9/10.

2.1

We can define the following simple events:

- R: ”meet a Randomukee” - $P(R) = \frac{2}{3}$
- B: ”meet a Bugiardukee” - $P(B) = \frac{1}{3}$
- R_t: ”Randomukee says the truth” - $P(R_t) = \frac{3}{4}$
- B_t: ”Bugiardukee says the truth” - $P(B_t) = 0$
- R_f: ”Randomukee says the false” - $P(R_f) = \frac{1}{4}$
- B_f: ”Bugiardukee says the false” - $P(B_f) = 1$

$$\Omega = \{\} \tag{10}$$

We can use regular expressions to define a sample space for the next answers. These are all the possible outcomes:

- $RR_t\left\{1,i\right\} = \left\{RR_t, RR_tR_t, RR_tR_tR_t, RR_tR_tR_tR_t\right\}$ for i $\in \left\{1,2,3,4\right\}$
- $RR_f\left\{1,i\right\} = \left\{RR_f, RR_fR_f, RR_fR_fR_f, RR_fR_fR_fR_f\right\}$ for i $\in \left\{1,2,3,4\right\}$
- $RR_t\left\{3\right\}R_f = \left\{RR_tR_tR_tR_f\right\}$
- $RR_f\left\{3\right\}R_t = \left\{RR_fR_fR_fR_t\right\}$
- $BB_f\left\{3\right\} = \left\{BB_f, BB_fB_f, BB_fB_fB_f, BB_fB_fB_f\right\}$

If we consider a sample space with no restrictions on the possible outcomes (except for the problem data), we can compute the probability for the outcomes defined above:

- $P(RR_t\left\{1,i\right\}) = \frac{2}{3} \cdot \left(\frac{3}{4}\right)^i$ for i $\in \left\{1,2,3,4\right\}$
- $P(RR_f\left\{1,i\right\}) = \frac{2}{3} \cdot \left(\frac{1}{4}\right)^i$ for i $\in \left\{1,2,3,4\right\}$
- $P(RR_t\left\{3\right\}R_f) = \frac{2}{3} \cdot \left(\frac{3}{4}\right)^3 \cdot \frac{1}{4}$
- $P(RR_f\left\{3\right\}R_t) = \frac{2}{3} \cdot \left(\frac{1}{4}\right)^3 \cdot \frac{3}{4}$
- $P(BB_f\left\{3\right\}) = \frac{1}{3}$

These are the probabilities in our 'reference' sample space. Clearly the sample space is dynamic and will vary depending on the new informations coming from the questions that follow. In the next questions we wil use these reference probabilities to compute the probability in the dynamic (current) sample space.

2.2

The possible outcomes are: RR_t, RR_f, BB_f, so the current sample space is:

$$\Omega_{(2)} = \{RR_t, RR_f, BB_f\} \tag{11}$$

The probability that the answer is correct is equal to the the probability to meet a Randomukee and that he answers the truth, so it is simply equal to:

$$P(RR_t) = \frac{2}{3} \cdot \frac{3}{4} = \frac{1}{2} \tag{12}$$

2.3

In this case the possible outcomes and their probabilities are respectively:

- $RR_tR_t \quad P(RR_tR_t) = \frac{2}{3} \cdot \left(\frac{3}{4}\right)^2 = \frac{3}{8}$
- $RR_fR_f \quad P(RR_fR_f) = \frac{2}{3} \cdot \left(\frac{1}{4}\right)^2 = \frac{1}{24}$
- $BB_fB_f \quad P(BB_fB_f) = \frac{1}{3} \cdot 1^2 = \frac{1}{3}$

These probabilities are related to the reference sample space where also other outcomes are possible, like the outcomes RR_tR_f and RR_fR_t. We are interested to find probability of the outcome RR_tR_t in the current sample space, which is:

$$\Omega_{(3)} = \{RR_tR_t, RR_fR_f, BB_fB_f\} \tag{13}$$

We know that the probability of the sample space must be equal to one ($P(\Omega) = 1$), so we can use a simple proportion to find the probability of the outcome RR_tR_t in the current sample space:

$$[P(RR_tR_t) + P(RR_fR_f) + P(BB_fB_f)] : 1 = P(RR_tR_t) : P_x(RR_tR_t) \tag{14}$$

So the wanted probability is:

$$P_x(RR_tR_t) = \frac{\frac{3}{8}}{(\frac{3}{8} + \frac{1}{24} + \frac{1}{3})} = \frac{1}{2} \tag{15}$$

Alternatively we can use again the Bayes theorem. We define the event SA:

- SA: "the man gives the same answer".

This event is a subset of the sample space:

$$SA = \{BB_fB_f, RR_tR_t, RR_fR_f\} \tag{16}$$

and its probability is:

$$P(SA) = P(BB_fB_f) + P(RR_tR_t) + P(RR_fR_f) = \frac{1}{3} + \frac{2}{3} \cdot \left(\frac{3}{4}\right)^2 + \frac{2}{3} \cdot \left(\frac{1}{4}\right)^2 = \frac{3}{4} \tag{17}$$

So its probability is the sum of the probability to meet a Bugiardukee (which of course will tell two times the false) plus the probability to meet a Randomukee that answers two times the truth plus the probability to meet Randomukee which will answers two times the false. Now we can use the Bayes theorem to find the probability to have the correct answer (from a Randomukee of course) given that we got the same answer to the same question (which is the probability we are looking for):

$$P(RR_tR_t|SA) = \frac{P(SA|RR_tR_t) \cdot P(RR_tR_t)}{P(SA)} \tag{18}$$

$$P(RR_tR_t|SA) = \frac{1 \cdot \frac{2}{3} \cdot \left(\frac{3}{4}\right)^2}{\frac{1}{3} + \frac{2}{3} \cdot \left(\frac{3}{4}\right)^2 + \frac{2}{3} \cdot \left(\frac{1}{4}\right)^2} = \frac{\frac{2}{3} \cdot \left(\frac{3}{4}\right)^2}{\frac{3}{4}} = \frac{1}{2} \tag{19}$$

which is the same probability we found earlier.

2.4

Again, we can use both the methods in 2.3. Using Bayes theorem we have:

$$P(RR_tR_tR_t|SA) = \frac{P(SA|RR_tR_tR_t) \cdot P(RR_tR_tR_t)}{P(SA)} \tag{20}$$

where in this case the SA event and its probability P(SA) are:

$$SA = \{BB_fB_fB_f, RR_tR_tR_t, RR_fR_fR_f\} \tag{21}$$

$$\begin{aligned} P(SA) = P(BB_fB_fB_f) + P(RR_tR_tR_t) + P(RR_fR_fR_f) = \\ = \frac{1}{3} + \frac{2}{3} \cdot \left(\frac{3}{4}\right)^3 + \frac{2}{3} \cdot \left(\frac{1}{4}\right)^3 = \frac{5}{8} \end{aligned} \tag{22}$$

So we have:

$$P(RR_tR_tR_t|SA) = \frac{P(SA|RR_tR_tR_t) \cdot P(RR_tR_tR_t)}{P(SA)} \tag{23}$$

$$P(RR_tR_tR_t|SA) = \frac{1 \cdot \frac{2}{3} \cdot \left(\frac{3}{4}\right)^3}{\frac{1}{3} + \frac{2}{3} \cdot \left(\frac{3}{4}\right)^3 + \frac{2}{3} \cdot \left(\frac{1}{4}\right)^3} = \frac{\frac{2}{3} \cdot \left(\frac{3}{4}\right)^3}{\frac{5}{8}} = \frac{9}{20} \tag{24}$$

Alternativally we can find the the same probability using a proportion like we did at the beginning of 2.3. First we have to define the probability of the possible outcomes:

- $RR_tR_tR_t$ — $P(RR_tR_tR_t) = \frac{2}{3} \cdot \left(\frac{3}{4}\right)^3 = \frac{9}{32}$
- $RR_fR_fR_f$ — $P(RR_fR_fR_f) = \frac{2}{3} \cdot \left(\frac{1}{4}\right)^3 = \frac{1}{96}$
- $BB_fB_fB_f$ — $P(BB_fB_fB_f) = \frac{1}{3} \cdot 1^3 = \frac{1}{3}$

The current sample space is:

$$\Omega_{(4)} = \{RR_tR_tR_t, RR_fR_fR_f, BB_fB_fB_f\} \tag{25}$$

We can use again the (26) to calculate the requested probability:

$$[P(RR_tR_tR_t)+P(RR_fR_fR_f)+P(BB_fB_fB_f)] : 1 = P(RR_tR_tR_t) : P_x(RR_tR_tR_t) \tag{26}$$

$$P_x(RR_tR_tR_tR_t) = \frac{\frac{9}{32}}{(\frac{9}{32} + \frac{1}{96} + \frac{1}{3})} = \frac{9}{20} \tag{27}$$

which is exatcly the same probability calculated with the Bayes theorem.

2.5

For brevity i'll stick only with the proportion method, assuming that with the Bayes theorem we'll get exactly the same result. The possible outcomes and their proabability are respectively:

- $RR_tR_tR_tR_t, \quad P(RR_tR_tR_tR_t) = \frac{2}{3} \cdot \left(\frac{3}{4}\right)^4 = \frac{27}{128}$

- $RR_fR_fR_fR_f, \quad P(RR_fR_fR_fR_f) = \frac{2}{3} \cdot \left(\frac{1}{4}\right)^4 = \frac{1}{384}$
- $BB_fB_fB_fB_f, \quad P(BB_fB_fB_fB_f) = \frac{1}{3} \cdot 1^4 = \frac{1}{3}$

The current sample space is:

$$\Omega_{(5)} = \{RR_tR_tR_tR_t, RR_fR_fR_fR_f, BB_fB_fB_fB_f\} \tag{28}$$

We can use again the (26) to calculate the requested probability:

$$P_x(RR_tR_tR_tR_t) = \frac{\frac{27}{128}}{(\frac{27}{128} + \frac{1}{384} + \frac{1}{3})} = \frac{27}{70} \tag{29}$$

2.6

In this case the possible outcomes are $RR_tR_tR_tR_f$ and $RR_fR_fR_fR_t$, since we know that the Bugiardukee answer always the false, so he can't change answer to the same question. The probabilities of these two outcomes in the primary sample space are:

- $P(RR_tR_tR_tR_f) = \frac{2}{3} \cdot \left(\frac{3}{4}\right)^3 \cdot \frac{1}{4} = \frac{9}{128}$
- $P(RR_fR_fR_fR_t) = \frac{2}{3} \cdot \left(\frac{1}{4}\right)^3 \cdot \frac{3}{4} = \frac{1}{128}$

The current sample space is:

$$\Omega_{(6)} = \{RR_tR_tR_tR_f, RR_fR_fR_fR_t\} \tag{30}$$

We are interested to find the probability of the outcome $RR_tR_tR_tR_f$ in the current sample space. Because $P(\Omega) = 1$, we can write a simple proportion:

$$[P(RR_tR_tR_tR_f) + P(RR_fR_fR_fR_t)] : 1 = P(RR_tR_tR_tR_f) : P_x(RR_tR_tR_tR_f) \tag{31}$$

where $P_x(RR_tR_tR_tR_f)$ is the probability we want to calculate. Thus:

$$P_x(RR_tR_tR_tR_f) = \frac{\frac{9}{128}}{\frac{9}{128} + \frac{1}{128}} = \frac{9}{10} \tag{32}$$

3 Problem 3

Assume that a monkey sits in front of a keyboard and hits randomly the 26 letters, each with the same probability. Assume that it types 100,000,000,000 letters. Let X be the number of times that the word ”mining” appears? What is the expectation of X?

3.1

So we have the random variable X = ”number of times that the word 'mining' appears”. Let's also define this Bernoulli indicator:

$$X_i = \begin{cases} 1 & \text{if the word 'mining' begin at character i of the sequence} \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

where $i \in \{1, 2, 3, ..., 100 * 10^9 - 6 + 1\}$ (6 is the number of characters of the word 'mining').

So we can write the random variable X in function of the X_i indicators:

$$X = \sum_{i=1}^{100*10^9-5} X_i \tag{34}$$

We want to calculate the expectation of X that is:

$$E[X] = E\left[\sum_{i=1}^{100*10^9-5} X_i\right] \tag{35}$$

From the linearity of the expectation we can take out the $\sum$ operator:

$$E[X] = \sum_{i=1}^{100*10^9-5} E\left[X_i\right] \tag{36}$$

We know that the number of letters of the keyboard are 26, and each letter is hit randomly, so the expactation of X_i is:

$$E[X_i] = 1 \cdot \left(\frac{1}{26}\right)^6 + 0 \cdot \left(\frac{25}{26}\right)^6 = \left(\frac{1}{26}\right)^6 \tag{37}$$

Using this expression in the (36) we got in final:

$$E[X] = \sum_{i=1}^{100*10^9-5} \left(\frac{1}{26}\right)^6 \approx 323.7 \tag{38}$$

4 Problem 4

Quite often you can analyze your data just by using simple unix tools. Some useful 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-2015/protected/ratebeerProcessed.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!)

4.1

Assuming that we are in the same path of the ratebeerProcessed.txt file on the console, this is a command line to display the first 10 beers with the highest number of reviews:

```
$ sed 's/[0-9]//g' ratebeerProcessed.txt | uniq -c | sort -n -r |
    head
```

The sed command is used together with the s command to replace all the beer scores number with an empty string, next with uniq -c command we can found the number of occurrences for each beer review (uniq appends that number at the begin of each line), next we sort in reverse by numeric-sort and display only the first ten results by the head command.

Alternatively we can use curl command to download and process data directly from http:

```
$ curl -u datamining2015:sapienzaroma "http://aris.me/contents/
    teaching/data-mining-2015/protected/ratebeerProcessed.txt" |
    sed 's/[0-9]//g' beersReview.txt | uniq -c | sort -n -r | head
```

In both cases the result is the following:

```
   3696 Guinness Draught
   3230 Dogfish Head  Minute Imperial IPA
   3126 Budweiser
   3119 Sierra Nevada Pale Ale &#;Bottle&#;
   3110 Samuel Adams Boston Lager
   3056 Chimay Bleue &#;Blue&#; / Grande R?serve
   2904 North Coast Old Rasputin Russian Imperial Stout
   2872 Stone Arrogant Bastard Ale
   2813 Orval
   2812 Newcastle Brown Ale
```

note: ratebeerProcessed.txt file is not included

5 Problem 5

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.)

5.1

- source code: problem5.py
- input file: ratebeerProcessed.txt
- output file: problem5output.tsv
- packages used: re (for regular expressions)

The source code is in problem5.py file. Packages used are: re for regular expressions. The output is written in a tsv file where each line is formatted this way:

```
beerName ”\t” numberReviews ”\t” averageScore ”\n”
```

Basically the algorithm uses regular expression to find indexes of the score number in each review (tab separated in the line), use these indexes to get the substring review, and update variables for the current number of reviews and current score. I’ve defined a class for a beer entity (BeerEntity) with 3 attributes: name, numReviews, avgScore (the name, the number of reviews and the average score respectively). At runtime it inserts in a list (resultList) each BeerEntity with number of reviews higher than 100. At the end we sort this list using a lambda operator to sort against the avgScore attribute. Then an helper method writes the result list in a .tsv file and prints the result on the console.

```
Westvleteren 12
Three Floyds Oak Aged Dark Lord Russian Imperial Stout
Bells Bourbon Barrel Double Cream/Expedition Stout
Goose Island Rare Bourbon County Stout
Lost Abbey Yellow Bus
De Dolle Stille Nacht Reserva 2000
Russian River Pliny the Younger
AleSmith Barrel Aged Speedway Stout
Cigar City Zhukov?s Final Push
AleSmith Speedway Stout
```

note1: ratebeerProcessed.txt file must be in the same directory of problem5.py file

note2: ratebeerProcessed.txt file is not included

6 Problem 6

[...]Write a program that will download from http://www.kijiji.it and parse all the job positions in Lazio about Informatica/Grafica/Web. Download regular and top announcements, but not sponsored ads. Save in a tab-separated value (TSV) file, for every job (one line per job), the title, short description (from the job summary page), the location, the publication date of the job announcement, and the URL link to its web page.

6.1

- source code: problem6.py
- output file: problem6output.tsv
- packages used:
  - re

- requests
- time
- BeatifulSoup

I used requests package to download html data from the website and BeatifulSoup to parse it and move in the parse tree using built-in functions. Before starting to download/parse/write the data i've used an helper method to find the number of pages (paging) of the requested uri in the website, because it can vary depending on the number of the ads posted online. With that information i build an uri in each iteration of the algorithm in the form:

```
http://www.kijiji.it/offerte-di-lavoro/offerta/annunci-lazio/
    informatica-e-web/?=p
```

where p is the number of the page. In each iteration of the algorithm we use a library function of beatifulsoup to find all the ads of the page (excluding sponsorized ones). For each ads we move in the parse tree to find also the uri for that ads and use again requests and beatifulsoup to get the long description of the ads. All the requested parameters are encoded in UTF-8 and wrote in .tsv file, the format of each line is:

```
title + "\t" shortDescription "\t" locale + "\t" timestamp "\t" +
    adsLink "\t" longDescription "\n"
```