Introduction to Web Science

Assignment 10

Prof. Dr. Steffen Staab
staab@uni-koblenz.de

René Pickhardt
rpickhardt@uni-koblenz.de

Korok Sengupta
koroksengupta@uni-koblenz.de

Olga Zagovora
zagovora@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: January 25, 2016, 10:00 a.m.
Tutorial on: January 27, 2016, 12:00 p.m.

For all the assignment questions that require you to write code, **make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.**

Team Name: papa
Members:
Brigitte Aznar
Bonasmitha Behura
Ilia Tugushi

1 Modeling Twitter data (10 points)

In the meme paper[1] by Weng et al., in Figure 2[2] you find a plot, comparing the system entropy with the average user entropy. Your task is to reproduce the plot and corresponding calculations.

1. We provide you with the file 'onlyhashtag.data', containing a collection of hashtags from tweets. Use this data to reproduce the plot from the paper. Once you have the values for average user entropy and system entropy calculated per day create a scatter plot to display the values.
2. Interpret the scatter plot and compare it with the authors interpretation from the graph showed in the paper. Will the interpretations be compatible to each other or will they contradict each other? Do not write more than 5 sentences.

1.1 Hints

1. Use formulas from the lecture to calculate the entropy for one user and the system entropy.
2. Do not forget to give proper names of plot axes.

```
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter


def get_hash(user_name, dataframe):
    dataframe = dataframe[dataframe["user_name"] == user_name]
    hashtag_list = dataframe["hashtag_list"].values
    hashtag_list = [value for sublist in hashtag_list for value in sublist]
    return hashtag_list

def entropy(counter_list):
    c = Counter(counter_list)
    ent = 0.0
    for k,v in c.items():
        prob = float(v)/len(counter_list)
        ent = ent - prob * math.log2(prob)
    return ent

def get_hash_date(date, dataframe):
    dataframe = dataframe[dataframe["date"] == date]
```

[1] http://www.nature.com/articles/srep00335
[2] Slide 27, Lecture Meme spreading on the Web

```
    hashtag_list = dataframe["hashtag_list"].values
    list_hash = [val for sublist in hashtag_list for val in sublist]
    return list_hash

def plotting(sorted):
    length=len(sorted)
    x = [x for x in range(0,length)]
    plt.title("System Entropy(sorted) and user Entropy (Average) per day")
    plt.xticks(np.arange(0,max(x),40))
    plt.yticks(range(0,int(max(sorted.sys_entropy)+2)))
    plt.xlabel("Days Rank")
    plt.ylabel("Entropy")
    plt.scatter(x,sorted.entropy.values,label='User Entropy',color="g")
    plt.scatter(x,sorted.sys_entropy.values,label='System
        Entropy',color="y")
    plt.show()

def main():

    data =
        pd.read_table('onlyhash.data',names=["user_name","date","hashtag"])
    data["hashtag_list"] = data.hashtag.apply(lambda x: x.split(" "))

    users = data.user_name.unique()
    index = [i for i in range(1, len(users)+1)]

    dates = data.date.unique()
    index2 = [i for i in range(1, len(dates)+1)]

    df = pd.DataFrame(users, index=index, columns=['user'])
    df["hashtags"] = df.user.apply(lambda x: get_hash(x, data))
    df["user_entropy"] = df.hashtags.apply(lambda x: entropy(x))

    data["entropy"] = data.hashtag_list.apply(lambda x: entropy(x))

    grp = data.groupby(["date"]).entropy.mean()
    user_entropy_by_day = grp.to_frame()

    df2 = pd.DataFrame(dates, index=index2, columns=['date'])
    df2["hashtags"] = df2.date.apply(lambda x: get_hash_date(x, data))
    df2["sys_entropy"] = df2.hashtags.apply(lambda x: entropy(x))

    user_entropy_by_day['date'] = user_entropy_by_day.index

    entropy_df = pd.merge(df2, user_entropy_by_day, on='date', how='outer')

    sorted = entropy_df.sort_values(by="sys_entropy")
    plotting(sorted)
```

```
if __name__ == "__main__":
    main()
```

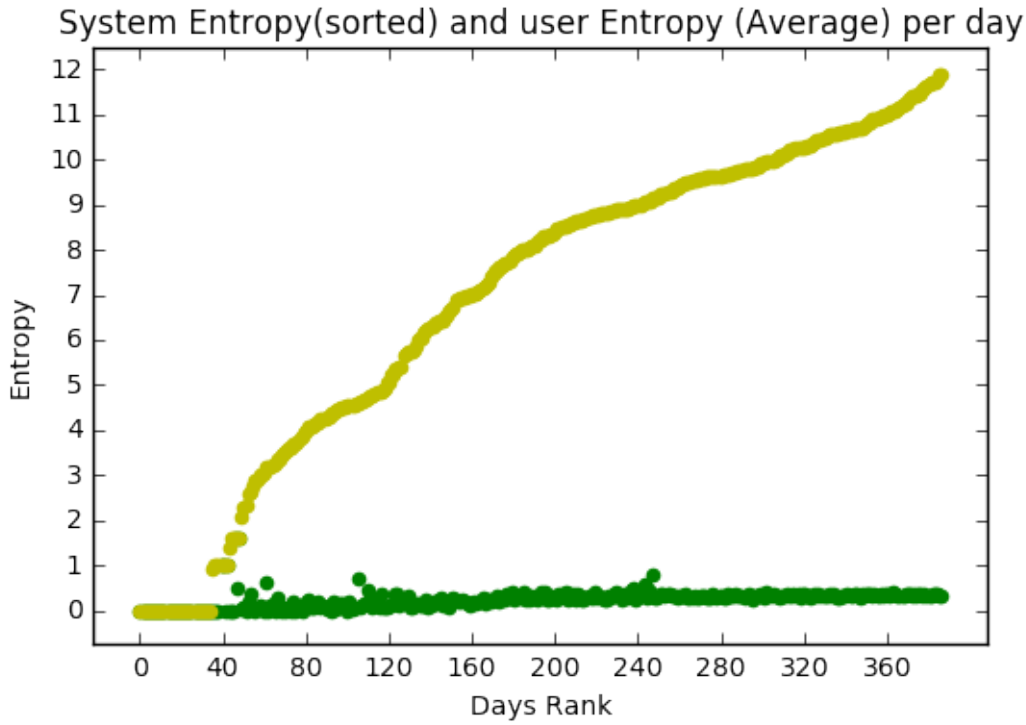

Figure 1: Visualization of entropy

We can see that as the author mentioned, user's attention remains constant trough the days, contrary to the system's entropy which grows almost exponentially. We interpret this as "No matter how many new memes are there out there, users will only be able to manage on average the same amount of memes".

2 Measuring inequality (10 points)

We provide you with a sample implementation of the Chinese Restaurant Process[3].

Assume there is a restaurant with an infinite number of tables. When a new customer enters a restaurant he chooses an occupied table or the next empty table with some probabilities.

According to the process first customer always sits at the first table. Probability of the next customer to sit down at an occupied table i equals ratio of guests sitting at the table (c_i/n), where n is the number of guests in the restaurant and c_i is the number of guests sitting at table i.
Probability of customer to choose an empty table equals : $1 - \sum_{i=1}^{S} p_i$, where S is the number of occupied tables and $p_i = c_i/n$.

Provided script simulates the process and returns number of people sitting at each table. We will study restaurants for 1000 customers. Now you should modify the code and evaluate how unequal were the customers' choices of tables.

Calculate the Gini- coefficient measuring the inequality between the tables, until the coefficient stabilizes. Do five different runs and plot your results in a similar way that plots in the lecture slides are done, cf. Slide 32 and Slide 33.

Answers:

```
import random
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

def plot(ginis,restaurants):
    # The y values. Cumulative percentage of incomes
    j = 0
    for restaurant in restaurants:
        y, y_pe = [], []
        i = 0
        restaurant = sorted(restaurant)
        for rest in restaurant:
            prob = rest*100/sum(restaurant)
            if i == 0:
                y.append(prob)
            else:
                y.append(prob + y[i-1])
            i += 1

        # The perfect equality y values. Cumulative percentage of incomes.
```

[3]File "chinese_restaurant.py"; Additional information can be found here: https://en.wikipedia.org/wiki/Chinese_restaurant_process

```
        y_pe = np.linspace(0.0, 100.0, len(y))

        plt.title("Gini " + str(round(ginis[j],2)))
        plt.plot(y, label='lorenz', color='r')
        plt.plot(y_pe, label='perfect_equality', color='b')
        plt.ylabel('Percentage of guests')
        plt.xlabel('No. of tables')
        x_legend = mpatches.Patch(color='red', label='observations')
        y_legend = mpatches.Patch(color='blue', label='perfect equality')
        plt.legend(handles=[x_legend, y_legend], loc=5)
        plt.show()
        j += 1

def gini(restaurant):
    ordered = sorted(restaurant)
    height, area_under_curve = 0, 0
    for value in ordered:
        height += value
        area_under_curve += height - value / float(2)
    area_under_equity = height * len(restaurant) / float(2)
    gini = (area_under_equity - area_under_curve) / area_under_equity
    return gini


def generateChineseRestaurant(customers):
    # First customer always sits at the first table
    tables = [1]
    #for all other customers do
    for cust in range(2, customers+1):
           # rand between 0 and 1
           rand = random.random()
           # Total probability to sit at a table
           prob = 0
           # No table found yet
           table_found = False
           # Iterate over tables
           for table, guests in enumerate(tables):
               # calc probability for actual table an add it to total
                   probability
               prob += guests / (cust)
               # If rand is smaller than the current total prob., customer
                   will sit down at current table
               if rand < prob:
                   # incr. #customers for that table
                   tables[table] += 1
                   # customer has found table
                   table_found = True
                   # no more tables need to be iterated, break out for loop
                   break
```

```
            # If table iteration is over and no table was found, open new
                table
            if not table_found:
                tables.append(1)
    return tables

restaurants = 1000
network, ginis = [], []
for iterations in range(1, 6):
    network.append(generateChineseRestaurant(restaurants))
for net in network:
    ginis.append(gini(net))

plot(ginis,network)
```

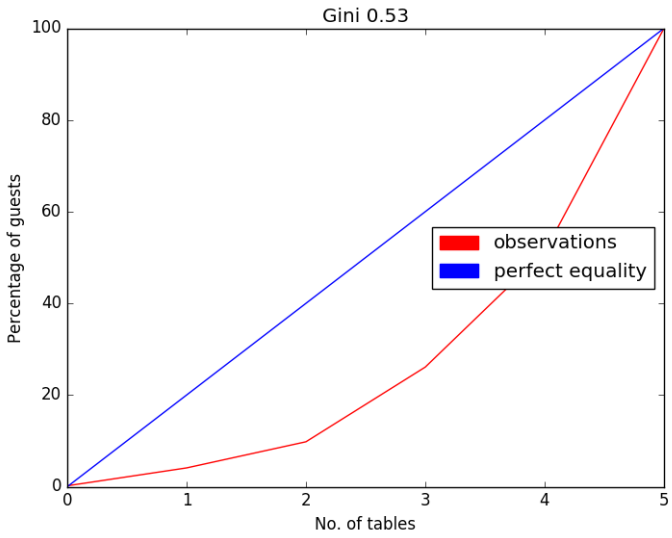

Figure 2: Gini Equality for the first iteration

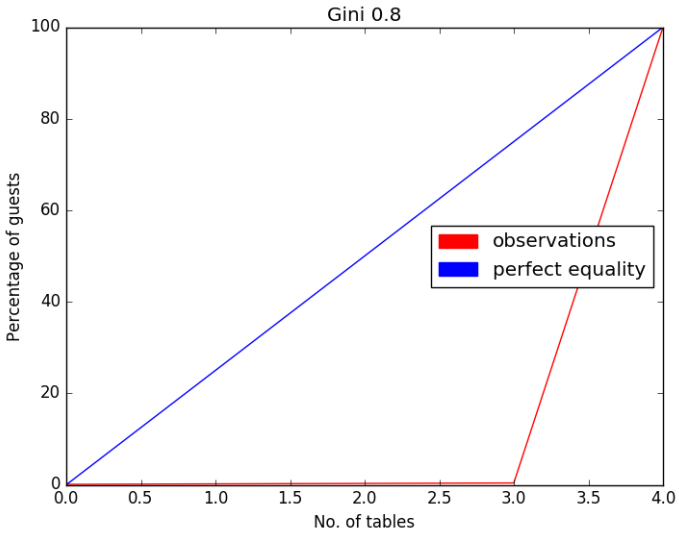

Figure 3: Gini Equality for the second iteration

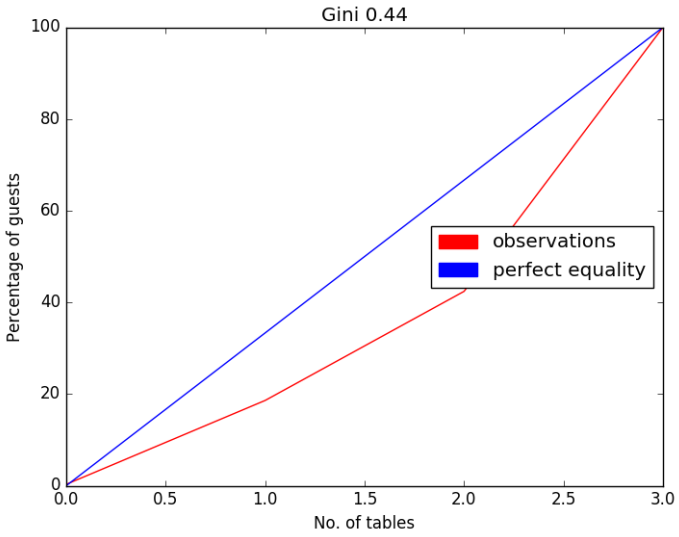

Figure 4: Gini Equality for the third iteration

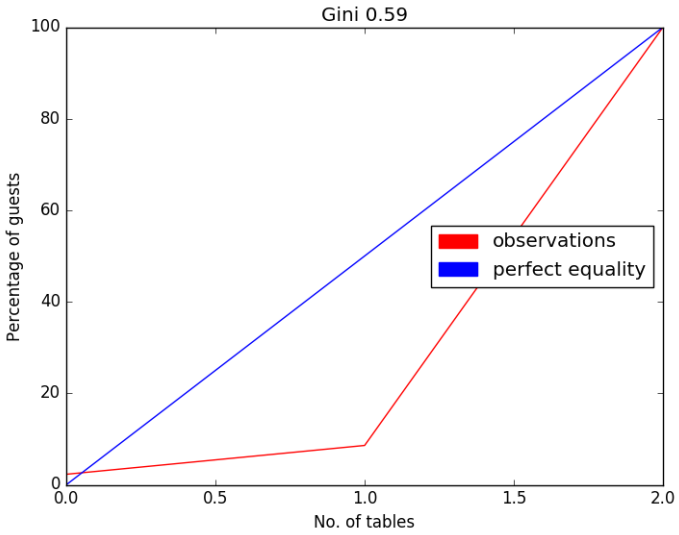

Figure 5: Gini Equality for the forth iteration

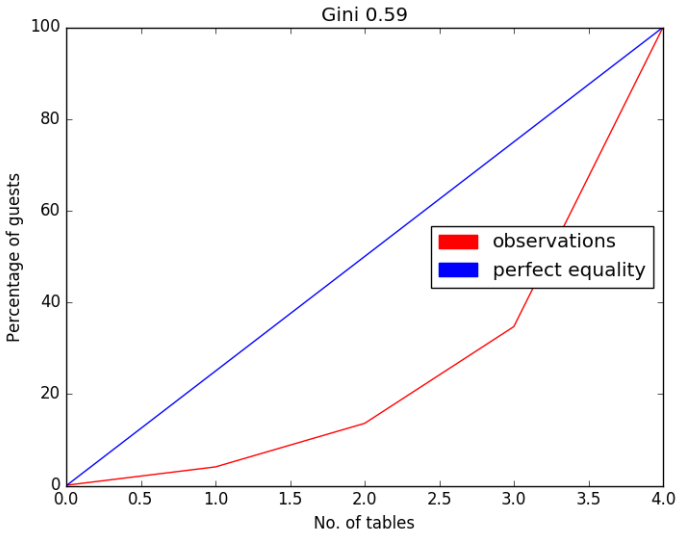

Figure 6: Gini Equality for the fifth iteration

3 Herding (10 points)

Let us consider the altitude of Koblenz to be 74 m above sea level. You are asked to figure out the height of the Ehrenbreitstein Fortress and the Fernmeldeturm Koblenz without googling.
The exercise is split in two parts:

Part 1 : The Secret
In *complete secrecy*, each member of the team will write down their estimated height of the Ehrenbreitstein Fortress without any form of discussion. Please keep in mind that you need to have reasons for your assumption. Once you are done, then openly discuss in the group and present you values in a tabulated format with the reasons each one assumed to arrive at that value.

Part II : The Discussion
Discuss amongst yourself with valid reasoning what could be the height of the Fernmeldeturm Koblenz. Only after discussing, each member of the group is asked to arrive at a value and present this value in a tabulated format as was done in Part I.

Calculate the Mean, Standard Deviation and Variance of your noted results for both the cases and explain briefly what you infer from it.

Note: This exercise is for you to understand the concepts of herding and not to get the perfect height by googling information. There is in fact no point associated with the height but with the complete reasoning that you provide for your answers.

Monument	Height (in meters)	Reasoning
Fortress	240	The lenght of the seilbahn don't seem that long so the mountain can't be that tall
Fortress	300	Taking into consideration that Koblenz is at a 74 m altitude, as the fortress stands on small hill it can't be taller than estimated.
Fortress	375	Compared to Deutsches Eck the hill seems tall enough for me to reach about this height.
Tower	400	I just assume the tower is in a much higher hill than the fortress. Therefore i think is this much higher
Tower	450	Based on my initial guess about the fortress. I believe that the tower is a copule of hundred meters taller than this hill altitude.
Tower	450	I think this tower is located on a hill that is considerably much taller than the fortress.

```
The mean height fot the fortres is  of 305.0 m
The standard deviation for the fortress is of 55.23 m
The Variation for the fortress is of 3050.0 m


The mean height fot the tower is  of 328.75 m
The standard deviation for the tower is of 63.08 m
The Variation for the tower is of 3979.69 m
```

Figure 7: Results

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment10/` in your group's repository.
- The name of the group and the names of all participating students must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the github repository. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This latex template was created by Lukas Schmelzeisen for the tutorials of "Web Information Retrieval".

LaTeX

Currently the code can only be build using LuaLaTeX, so make sure you have that installed. If on Overleaf, there's an error, go to settings and change the LaTeXengine to `LuaLaTeX`.