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: XXXX

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.

Answers

1. See Figure 1 for the plot and Listing 1 for the code
2. As we can see, the plot shows that the average entropy of the users stays at the same level over time, where the system's entropy increases. This directly correlates with the papers result, that users only have a limited breath of attention.

Listing 1 Task 3

```
1: from collections import Counter
2: from math import log2
3: import pandas as pd
4: import numpy as np
5: import matplotlib.pyplot as plt
6: from matplotlib.dates import YearLocator, MonthLocator,
       ↪ DateFormatter
7:
8: data = pd.read_csv('onlyhash.data', sep='\t', names=['user', 'date
       ↪ ', 'tags'])
9:
10:
11: def entropy(vector):
12:     size = vector.size
```

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

```
13:     frequencies = Counter(vector).values()
14:     probabilities = list(map(lambda frequency: frequency / size,
            ↪ frequencies))
15:     return - sum(map(lambda p: log2(p) * p, probabilities))
16:
17:
18: date_to_tweets = dict(list(data.groupby('date')))
19: date_to_user_to_tweets = {date: dict(list(tweets.groupby('user')))
      ↪  for date, tweets in date_to_tweets.items()}
20:
21: date_to_system_entropy = {date: entropy(tweets['tags']) for date,
      ↪ tweets in date_to_tweets.items()}
22:
23: date_to_mean_user_entropy = {date: np.mean([entropy(tweets['tags'
      ↪ ]) for user, tweets in user_to_tweets.items()]) for
24:                               date, user_to_tweets in
                                     ↪ date_to_user_to_tweets.items()}
25:
26: dates = list(sorted(date_to_tweets.keys()))[46:-3]
27: years = YearLocator()
28: months = MonthLocator()
29: yearsFmt = DateFormatter('%Y')
30: fig, ax = plt.subplots()
31: fig.suptitle('system entropy and average user entropy over time')
32: ax.plot_date(dates, [date_to_system_entropy[date] for date in
      ↪ dates], '-', label='system entropy', color='#F44336')
33: ax.plot_date(dates, [date_to_mean_user_entropy[date] for date in
      ↪ dates], '-', label='average user entropy',
34:              color='#3F51B5')
35: ax.xaxis.set_major_locator(years)
36: ax.xaxis.set_minor_locator(months)
37: ax.autoscale_view()
38: ax.fmt_xdata = DateFormatter('%Y-%m-%d')
39: ax.set_xlabel('Time')
40: ax.set_ylabel('Entropy')
41: plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
42: plt.show()
```

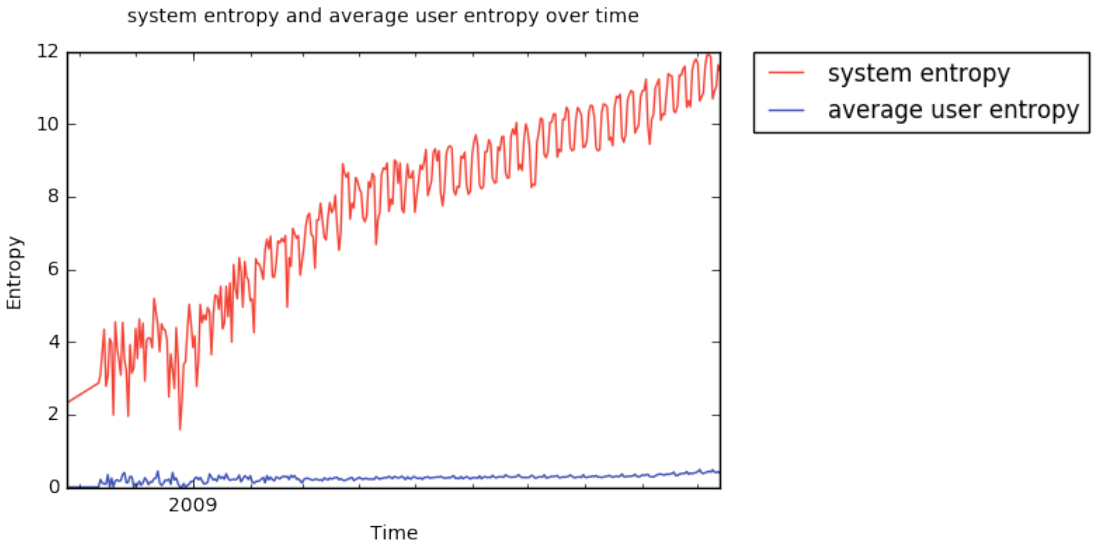

Figure 1: Task 1 - Plot

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.

Solution One concern is that the sample size S in the Gini-coefficient is not static, the table number is said to be infinite. However, as the formula contains S^2 in the denominator, we assume that we do not regard ∞ as the sample size.

Listing 2 Task 2

```
import random
import json
import matplotlib as pl
import matplotlib.pyplot as pp


def gini(sample):
    """
    Calculates the Gini coefficient
    :param sample: The sample to evaluate
    :return: Returns the Gini coefficient
    """
    num = sum(abs(x_i - x_j) for x_i in sample for x_j in sample)
    den = 2 * len(sample) * sum(x_i for x_i in sample)
    return num / den

```

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

```
17:
18: def generate_chinese_restaurant(customers):
19:     # First customer always sits at the first table
20:     tables = [1]
21:     ginis = [gini([1])]
22:
23:     # for all other customers do
24:     for cust in range(2, customers + 1):
25:         # rand between 0 and 1
26:         rand = random.random()
27:         # Total probability to sit at a table
28:         prob = 0
29:         # No table found yet
30:         table_found = False
31:         # Iterate over tables
32:         for table, guests in enumerate(tables):
33:             # calc probability for actual table an add it to
34:             # total probability
35:             prob += guests / cust
36:             # If rand is smaller than the current total prob.,
37:             # customer will sit down at current table
38:             if rand < prob:
39:                 # incr. #customers for that table
40:                 tables[table] += 1
41:                 # customer has found table
42:                 table_found = True
43:                 # no more tables need to be iterated, break out
44:                 # for loop
45:                 break
46:         # If table iteration is over and no table was found, open
47:         # new table
48:         if not table_found:
49:             tables.append(1)
50:
51:         ginis.append(gini(tables))
52:     return tables, ginis
53:
54:
55: restaurants = 1000
56:
57: network = generate_chinese_restaurant(restaurants)[0]
58: with open('network_' + str(restaurants) + '.json', 'w') as out:
59:     json.dump(network, out)
60:
61: pl.style.use('ggplot')
62:
63: fig, ax = pp.subplots()
64: ax.set_xlabel("Run number")
65: ax.set_ylabel("Gini coefficient")
```

```
66:
67: for run in range(5):
68:     gini_curve = generate_chinese_restaurant(restaurants)[1]
69:     xs = range(len(gini_curve))
70:     ys = gini_curve
71:
72:     ax.plot(xs, ys, label='Run %d' % run)
73:
74: ax.legend(loc='lower right')
75: pp.show()
```

The result of five independent runs is shown in the figure 2

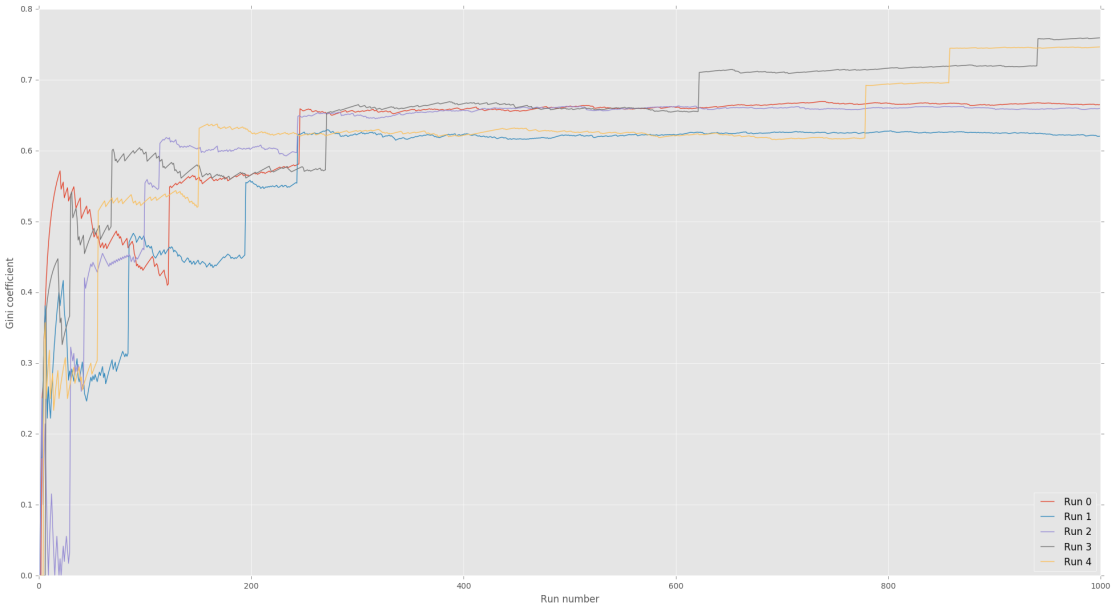

Figure 2: Five independent runs of the Chinese Restaurant program, Gini coefficients plotted for the tables' distribution of customers.

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.

Results:

The height and arguments are the following:

Person 1 Height = 125 m Argument: I assumed that angle of the cable car to be 5 degree and the distance to the fortress is about 500m. I used to live in Koblenz-Ehrenbreitstein and the distance to the fortress should be about the same as the length of the pfaffendorfer bridge. So tan(5) about 0.1, therefore h 125

Person 2 Height = 234 m Argument: Basically because the first segment to the gate is about 20 meter and then this times 6 plus height of that building over sea level plus another 40 at the end because the road gets steeper. 40 meant to be height of building plus steep part. So 74 plus 160.

Person 3 Height = 150 m Argument: Initially evaluated as 200 meters just approximately, but agreed with the arguments of person 1 and decided to decrease the height to 150.

Mean: (125+150+234)/3 = 170 m Variance: ((125 - 170)**2 + (150 - 170)**2 + (234 - 170)**2)/2 = 2173,7 Standard deviation: 46.6

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.

Estimations for height and arguments: Person 1 Height = 400 m Argument: ”As for the tower I say 400. It’s more like it”

Person 2 Height = 384 Argument: Unfortunately did not see it, so I had to google to find out the answer.

Person 3 Height = 384 Argument: Decided to agree with person 3 even though there was a prohibition to google the result.

Mean: (384 + 384 +400)/3 = 389.3 Variance: ((389.3 - 384)**2 +(389.3 - 384)**2 + (400-389.3)**2)/3 = 56.89 Standard deviation: 7.54

Inference: Personal point of view depends heavily on the collective opinion and could be influenced by the external arguments.

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.

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`.