Introduction to Web Science

Assignment 6

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

René Pickhardt
rpickhardt@uni-koblenz.de

Korok Sengupta
koroksengupta@uni-koblenz.de

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

Submission until: December 6, 2016, 10:00 a.m.
Tutorial on: December 9, 2016, 12:00 p.m.

Please look at the lessons 1) **Simple descriptive text models** & 2) **Advanced descriptive text models**

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:India
Team Members: Jasvinder Kaur, Jalpa Patel, Amani Gaddamedi

1 Digging deeper into Norms (10 points)

You have been introduced to the concept of a norm and have seen that the uniform norm $||\cdot||_\infty$ fullfills all three axioms of a norm which are:

1. Positiv definite
2. Homogeneous
3. Triangle inequality

Recall that for a function $f : M \longrightarrow \mathbb{R}$ with M being a finite set[1] we have defined the L_1-norm of f as:

$$||f||_1 := \sum_{x \in M} |f(x)| \tag{1}$$

In this exercise you should

1. calculate $||f-g||_1$ and $||f-g||_\infty$ for the functions f and g that are defined as
   - $f(0) = 2, f(1) = -4, f(2) = 8, f(3) = -4$ and
   - $g(0) = 5, f(1) = 1, g(2) = 7, g(3) = -3$
2. proof that all three axioms for norms hold for the L_1-norm.

1.1 Hints:

1. The proofs work in a very similar fashion to those from the uniform norm that was depicted in the videos.
2. You can expect that the proofs for each property also will be "three-liners".
3. Both parts of this exercise are meant to practice proper and clean mathematical notation as this is very helpfull when reading and understanding research papers. Discuss in your study group not only the logics of the calculation and the proof (before submission) but try to emphasize on the question whether your submission is able to communicate exactly what you are doing.

Answer:

[1] You could for example think of the function measuring the frequency of a word depening on its rank.

1.

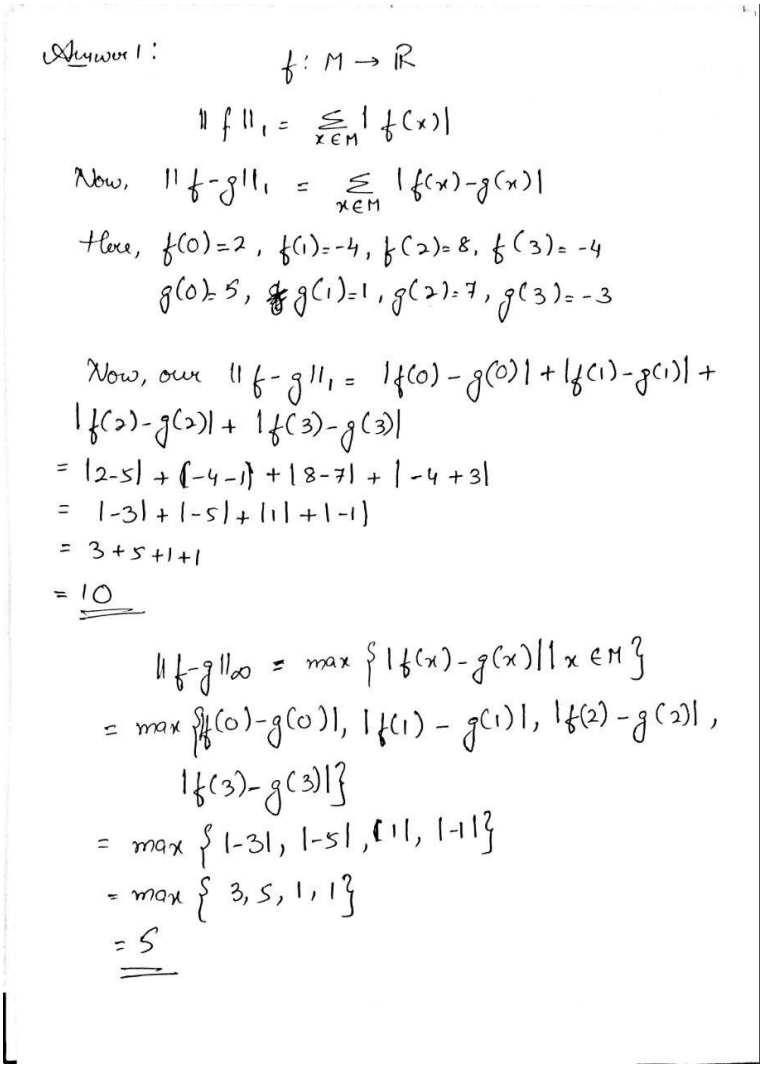


Figure 1: The calculate $||f-g||_1$ and $||f-g||_\infty$ for the functions f and g

2.

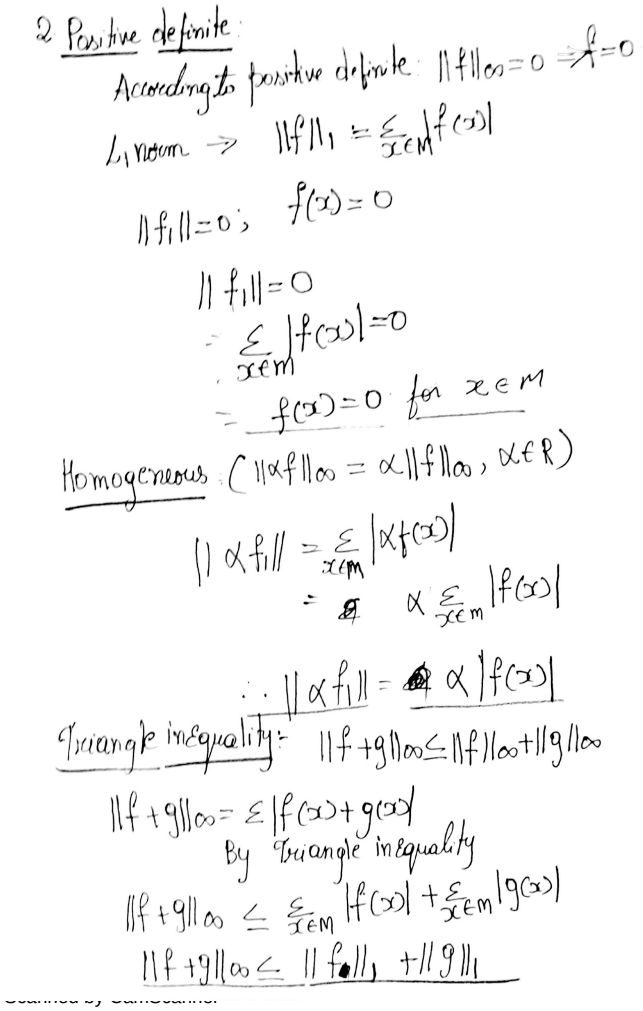


Figure 2: proof of three axioms

2 Coming up with a research hypothesis (12 points)

You can find all the text of the articles from Simple English Wikipedia at http://141.26.208.82/simple-20160801-1-article-per-line.zip each line contains one single article.

In this task we want you to be creative and do some research on this data set. The ultimate goal for this exercise is to practice the way of coming up with a research hypothesis and testable predictions.

In order to do this please **shortly**[2] answer the following questions:

1. What are some obervations about the data set that you can make? State at least three obervations.
2. Which of these observations make you curious and awaken your interest? Ask a question about why this pattern could occur.
3. Formulate up to three potentiel research hypothesis.
4. Take the most promesing hypothesis and develop testable predictions.
5. Explain how you would like to use the data set to test the prediction by means of descriptive statistics. Also explain how you would expect your outcome.

   (If you realize that the last two steps would not lead anywhere repeat with one of your other research hypothesis.)

2.1 Hints:

- The first question could already include some diagrams (from the lecture or ones that you did yourselves).
- In step 3 explain how each of your hypothesis is falsifiable.
- In the fifth step you could state something like: "We expect to see two diagrams. The first one has ... on the x-axis and ... on the y-axis. The image should look like a ... The second diagram ...". You could even draw a sketch of the diagram and explain how this would support or reject your testable hypothesis.

2.2 Answers:

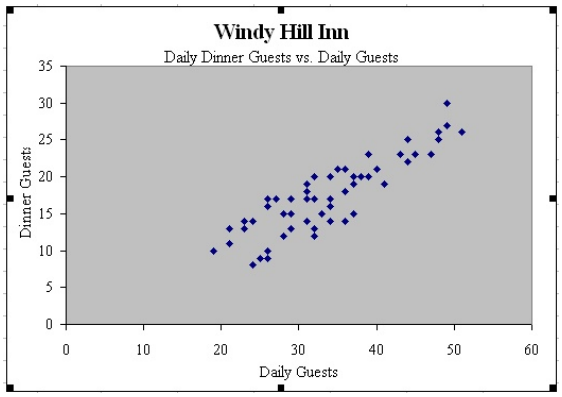
- Answer1: Observations which we strive are: 1. Capital and small letters are counted as different.
  2. Frequency of the unique words per article 3. Frequency of wordlength per article

[2] Depending on the question shortly could mean one or two sentences or up to a thousand characters. We don't want to give a harsh limit because we trust in you to be reasonable.

4. Occurrence of auxiliary verbs, prepositions, adverbs, prepositions.
5. Total number of articles

- Answer 2: Observation which makes us curious is:
  Frequency of wordlength per article
  Median of the word length frequency
- Answer3: Potential research hypothesis are:
  1. Articles in the dataset, having higher median frequency of word size are more difficult to understand.
  2. Articles having similar frequency of wordsize are similar to each other where similarity is measured as the difference between respective median word lengths.
  3. Article size is directly proportional to the median wordlength of article.
- Answer 4:the most promising hypothesis is:
  Article size is directly proportional to the median wordlength of article. We are not taking hypothesis 1 stating that the articles in the dataset, having higher median frequency of word size are more difficult to understand because the difficulty level of an article cannot be judged by the machine. It varies from person to person. And, the same is the case with hypothesis 2 that articles having similar frequency of wordsize are similar to each other where similarity is measured as the difference between respective median word lengths. But, the similarity measure in two articles can only be given by human experts if,in actual, they are similar or not to each other.
- Answer 5:We need to measure the length of the articles,frequency of word length per article,median of the frequency of wordlength of the article
  We, then plot the the length of article vs frequency of word length in a scatter plot, where the length of article is plotted on x axis and that frequency is plotted on y axis. Ideally, the curve should come out to be proportional between the two quantities.

Figure 3: Expected plot



3 Statistical Validity (8 points)

In the above question, you were asked to formulate your hypothesis. In this one, you should follow your own defined roadmap from task 2 validate (or reject) your hypothesis.

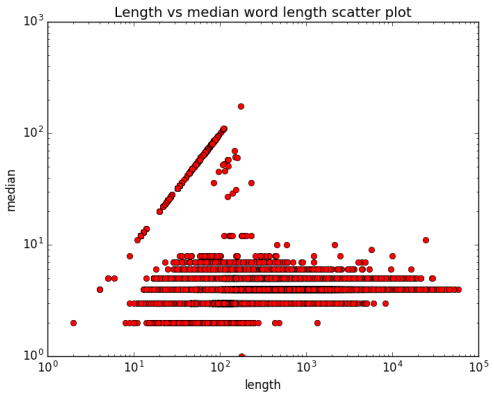
3.1 Hints:

- In case feel uncomfortable to test one of the predictions from task 2 you can "steal" one of the many hypothesis (and with them imlicitly associated testable predictions) or diagrams depicted from the lecture and reproduce it. However in that case you cannot expect to get the total amount of points for task 3.

3.2 Answer:

- Ideally, the curve should come out to be proportional between the two quantities. But, the output of our scatter plot is not as expected. Many points are below the straight rising curve, denoting that many median values are not rising proportionally as the length of article increases. Further statistical values can be found, but from the naked eye, it comes to be in a plot different from our expected one.
  So, this falsifies our proposed hypothesis.

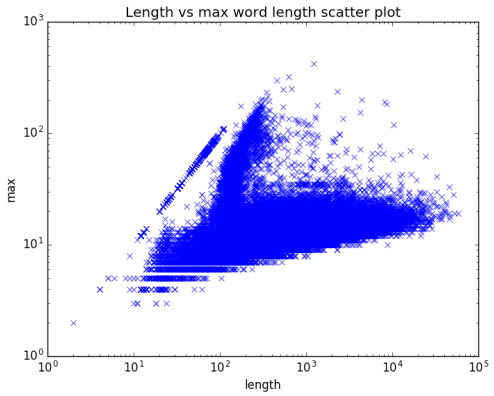
Figure 4: Actual obtained plot



- Now, median is not a suitable statistic because it ignores the outliers and their complexity as bigger length words will always occur in the outliers. Measuring the max is a better substitute to the median.
  Max denotes the max wordlength of the article.
  Words of higher wordlength have higher probability to occur in the articles of bigger size.

So, max shows a better correlation to the article length. And it is not falsified from the scatter plot involved.

Figure 5: Actual obtained plot



- So, we can still work more and find more better statistics to give a better modelling.
- Running our python code: python India_assignment6.py "dataset" where dataset is target dataset.

1. India Assignment6

```
#it replaces all dots,commas etc in the file by spaces and takes o(n)
import India_stats as stat
import matplotlib.pyplot as plt
import sys
def word_frequency_in_article(file_name):
  article_list = []
  article_hash = {}
  with open(file_name,'r') as file:
    for line in file:
      ##article_list extracted by code for extracting article
      #articles_count
      article_hash = read_and_countwords(line)
      line = line.strip()
      if line:
        article_list.append(article_hash)
  return article_list
def article_lengths(file_name):
  length_of_article = []
  with open(file_name, 'r') as file:
    for line in file:
      line = line.strip()
      if line:
        length_of_article.append(len(line))
  return length_of_article
```

```
25:
26: def read_and_countwords(text):
27:   words_count = {}
28:   text.strip()
29:   cleandot_data = text.replace('.',' ')
30:   cleancomma_data = cleandot_data.replace(',',' ')
31:   cleanbracket1_data = cleancomma_data.replace('(',' ')
32:   cleanbracket2_data = cleanbracket1_data.replace(')',' ')
33:   cleancolon_data = cleanbracket2_data.replace(':',' ')
34:   cleanspace_data = cleancolon_data.replace(' ',' ')
35:   cleanquotes_data = cleanspace_data.replace('"',' ')
36:   word_list = cleanquotes_data.split()
37:   words_count = count_length_of_words_in_list(word_list)
38:   return words_count
39:
40:
41: #it returns the count of the word from the given list
42: def count_length_of_words_in_list(word_list):
43:   frequency_of_word_size = {}
44:   for word in word_list:
45:     length_of_word = len(word)
46:     frequency_of_word_size[length_of_word] = frequency_of_word_size.get(lengt
47:   return frequency_of_word_size
48: if __name__ == '__main__':
49:   if len(sys.argv) > 1:
50:     filename_to_be_read = sys.argv[1]
51:   else:
52:     filename_to_be_read = '/home/jass/Documents/jass/WebScience/india/assignm
53:   article_hash_list = word_frequency_in_article(filename_to_be_read)
54:   medians = [stat.median_of_freq(article_hash)
55:   for article_hash in article_hash_list]
56:   length = article_lengths(filename_to_be_read)
57:   plt.title('Length vs median word length scatter plot')
58:   plt.xlabel('length')
59:   plt.ylabel('median')
60:   plt.xscale('log')
61:   plt.yscale('log')
62:   plt.plot(length, medians, 'ro')
63:   plt.show()
64:   maxs = [max(article_hash.keys())
65:   for article_hash in article_hash_list]
66:   plt.plot(length, maxs, 'bx')
67:   plt.title('Length vs max word length scatter plot')
68:   plt.xlabel('length')
69:   plt.ylabel('max')
70:   plt.xscale('log')
71:   plt.yscale('log')
72:   plt.show()
```

2. India Stats

```
# -*- coding: utf-8 -*-
import numpy as np
import csv
"""
Created on Tue Nov 29 19:17:08 2016

@author: Amani
"""
def dict_to_sorted_lists(freq):
  xs = list(sorted(freq))
  ys = []
  for x in xs:
    ys.append(freq[x])
  return [xs, ys]

def cumulate_freq(freq_array):
  ys = freq_array[1]
  for iter in range(len(ys) - 1):
    ys[iter + 1] = ys[iter] + ys[iter + 1]
  return [freq_array[0], ys]


def cdf(cumulated_freq_array):
  ys = cumulated_freq_array[1]
  length = len(ys)
  max_val = ys[length - 1]
  cumul_ys = [(val / max_val) for val in ys]
  return [cumulated_freq_array[0], cumul_ys]

def median_of_freq(freq):
  xs_ys = dict_to_sorted_lists(freq)
  cum_xs_ys = cumulate_freq(xs_ys)
  cdf_xs_ys = cdf(cum_xs_ys)

  cdf_ys = cdf_xs_ys[1]
  for iter in range(len(cdf_ys)):
    if cdf_ys[iter] >= 0.5:
      return cdf_xs_ys[0][iter]



if __name__ == '__main__':
  a = { 26:21, 4:10, 43:4 }
  b = dict_to_sorted_lists(a)
  c = cumulate_freq(b)
  cdfs = cdf(c)
  med = median_of_freq(a)
  print(med)
```

```
49:  print(cdfs)
```

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment6/` 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`.