Introduction to Web Science

Assignment 7

Jun Sun
junsun@uni-koblenz.de

Iryna Dubrovska
idubrovska@uni-koblenz.de

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

Submission until: December 21, 2021, 24:00 CET
Tutorial on: January,6, 2021, 16:00 CET

This assignment focuses on the concepts of 1) **Generative Models for the Web**, 2) **Modeling the Web as a graph** and 3) **Programming in Python**. Some of the tasks may require you to do additional research extending the lecture. Please keep the citation rules in mind.

For all the assignment questions that require you to write a 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.

Date: 21/12/2021

Team Name: Boehm

Abhinav Ralhan (abhinavr8@uni-koblenz.de)

Fatima Akram (fatimaakram9396@uni-koblenz.de)

Hammad Ahmed (hammadahmed@uni-koblenz.de)

Vishal Vidhani (vvidhani@uni-koblenz.de)

1. Python Programming. Generative Models for the Web (25 points)

Your task is to write a script that would read provided text file Text_sample.txt and do the following:

1.Create a probabilistic generative model of text by sampling from the distribution of words' lengths and frequency of each character in the given text (similar to the one presented in the video slides).

Modelling choices:

- Your model should generate one word at a time according to the distribution of the word lengths.
- Within each word, generate characters according to the distribution of character frequencies.
- Your model should only consider lowercase letters [a-z] and numbers [0-9]. All upper-case letters should be converted to lowercase. Other characters such as punctuation should be excluded.

Generate a text of 5000 words with your model. Please upload your generated text as an individual file. You do not need to include it into the PDF document.

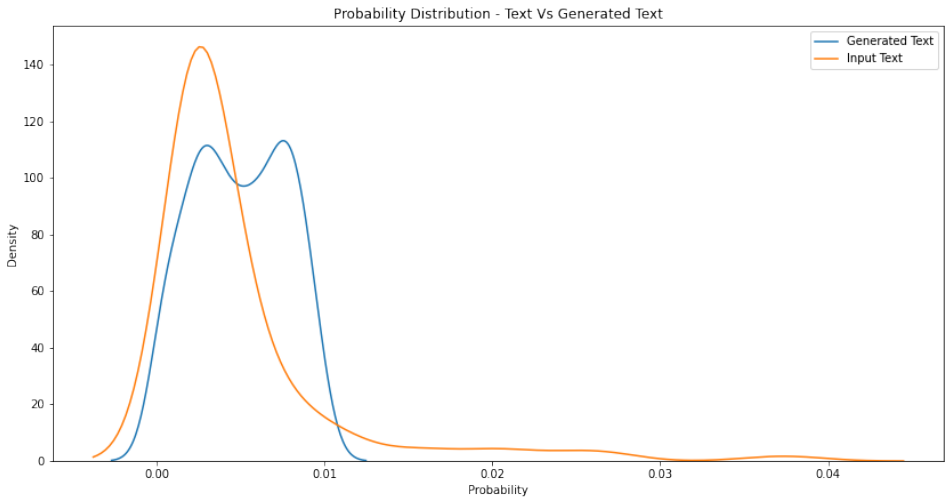
Please find:

Generated Text File: q1/q1-1-generatedWords.txt

Related python code: q1/q1-1-code.py

2.Plot the probability distribution of word lengths of the original text and the text you generated in one plot. Save the plot as PNG file.

Please find below image in q1/q1-2-ProbabilityDistribution.png



3. Discuss the resulting plot and generated text (max 200 words)
You are allowed to use the following libraries: string, re, numpy, matplotlib.

From the above plots, we can see that the input text follows a normal distribution which is also right skewed in nature. The generated text does not follow a normal distribution. Infact, it is a bimodal distribution with two peaks. Also, probability distribution of generated text does not have any extreme outliers like in the input text. Here, generated text captures most of the words present in the input text.

2. Modeling the Web as a graph (30 points)

1.Have a look at the adjacency matrix A (Figure 1). Based on the matrix, define which type of graph it represents and reconstruct the graph. (8 points)
Name another type of adjacency matrix which can be used for more efficient representation of this kind of graphs. (2 points)

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

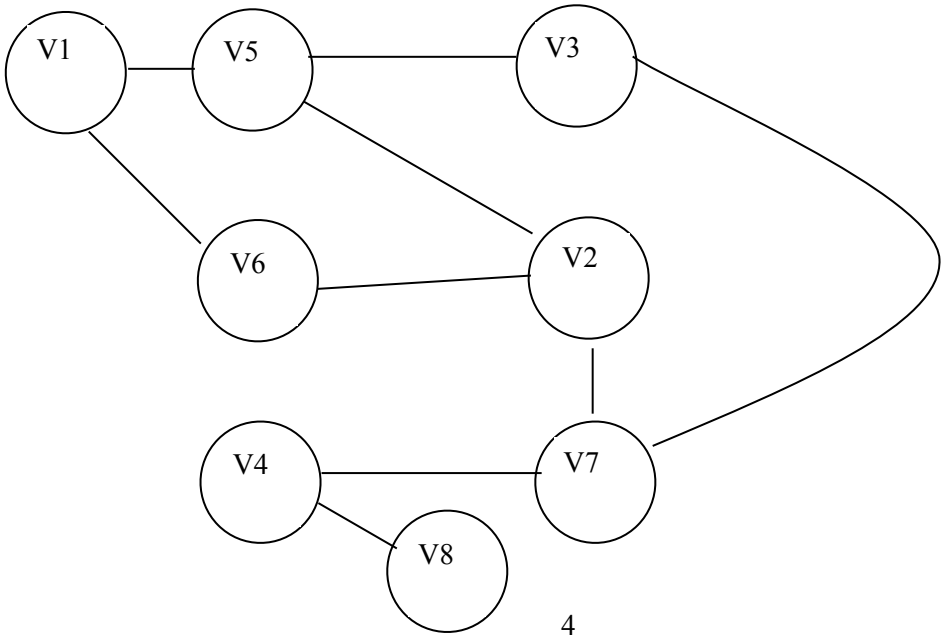
Figure 1: Adjacency matrix

Solution:

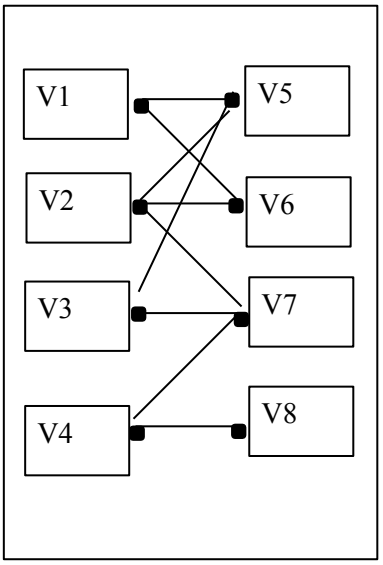
In the above given matrix A, the order of matrix is 8*8 where the no. of vertices are equals to 8. Let nodes be V1, V2, V3, V4, V5, V6, V7, V8.

The above given matrix is the type of "**Bipartite Graph**", where we can clearly see that Vertices V1, V2, V3, V4 has disjoint split with the vertices V5, V6, V7, V8. There aren't any edges connect within the vertex in a subset of the vertices.

Now, lets create a graph from the above adjacency matrix:



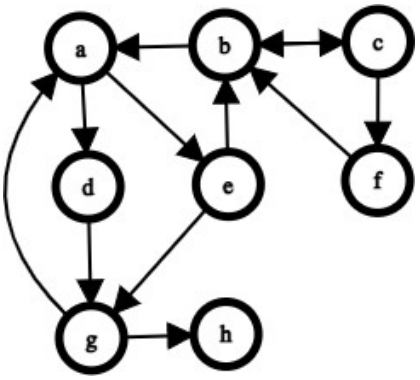
Bipartite Representation



Another efficient way to represent this adjacency matrix is the adjacency list, as we can see in the matrix that the graph is very sparse. Many of the vertices are not connected with each other, so using the adjacency list will be more efficient for the given matrix.

We can also use the Seidel adjacency matrix, known as (-1,1,0) adjacency matrix. Where diagonals are 0, non-adjacency vertices are -1 and adjacency vertices are 1.

2. Consider directed graph G pictured below (Figure 2).



Write down the adjacency matrix A of the graph G. (8 points)

Solution:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Calculate the in- and outdegree of every vertex. (4 points)

Solution:

Vertex	**In – degree**	**Out - degree**
A	**2**	**2**
B	**3**	**2**
C	**1**	**2**
D	**1**	**1**
E	**1**	**2**
F	**1**	**1**
G	**2**	**2**
H	**1**	**0**

Graph G contains cycles. List all the cycles you can detect. (4 points)

Solution: We have listed down 14 simple cycles (no repeated nodes) in given graph.

a -> d -> g -> a
a -> e -> b -> a
a -> e -> g -> a
b -> a -> e -> b
b -> c -> b
b -> c -> f -> b
c -> b -> c
c -> f -> b -> c
d -> g -> a -> d
e -> b -> a -> e
e -> g -> a -> e
f -> b -> c -> f
g -> a -> e -> g
g -> a -> d -> g

Write down the definition of the diameter as a mathematical formula, and find out the diameter d of graph G. (4 points)

Solution:

$$D(G)=\sup_{x,y\in V(G)} d(x,y)$$

$$\forall x, y \epsilon V(G) \exists z1 \ldots zn(x \rightarrow z1) \wedge (z1 \rightarrow z2) \wedge (z2 \rightarrow y)$$

The diameter of Graph G is 6, as the longest shortest path in the graph G is

d -> g -> a -> e -> b -> c -> f = 6

3. Short Questions (13 points)

3.1. You have learnt from the lectures about Gini coefficient as a measure of fairness. Explain what Gini coefficient and Lorenz curve show. Give an example of a case where Gini coefficient can be applied (excluding economic field and example from the lecture) (5 points)

Solution

The Gini coefficient indicates the measure of inequality. It lies in the range from 0 to 1; where 0 shows perfect equality and 1 indicates perfect inequality. It is generally used to demonstrate the inequality in incomes.

Generally a Lorenz curve is a graphical representation of the distribution of income or wealth in a population. It is a mathematical function which is made out of incomplete set of data so this may not be the most reliable measure of inequality or distribution.

Although Gini coefficient and Lorenz curve are mostly used to measure wealth and income inequality, it can also be used to measure the average hospital charges per diagnosis, or number of uploads or downloads in YouTube.

3.2. Explain why you should be careful with increasing the number of parameters in your generated model even though they may yield a better result. (4 points)

Solution

We should be careful while increasing the number of parameters in a model because as much as we increase the number of parameters our model would keep on getting complex and complicated. It would also be risking overfitting and not being generalized on the unseen data.

3.3. Explain what is the difference or relation between a graph and a tree. (4 points)

Solution

Both the graphs and tree are linear data structures. They comprise of edges vertices and path, however their difference lies in the fact that trees can not include loops while graphs can. In a tree there can be only one unique path between any two nodes, however in a graph there can be more than one and not necessarily unique paths between two certain nodes. A tree is a hierarchical structure whereas a graph is the network structure.

Important Notes

SUBMISSION

- Solutions have to be submitted to the SVN repository. Use the directory name groupname/assignment7/ 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 SVN 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 consistentindentation.
  - 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 pdfLaTeX template was adapted by Jun Sun based on the LuaLaTeX version by Lukas Schmelzeisen.

LaTeX

Use pdflatex assignment_X.tex to build your PDF.