Introduction to Web Science

Assignment 9

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 18, 2016, 10:00 a.m.
Tutorial on: January 20, 2016, 12:00 p.m.

For all the assignment questions that require you to write scripts, make sure to **include the scripts 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.

Group name: uniform
Group members: Pradip Giri, Jalak Arvind Kumar Pansuriya, Madhu Rakhal Magar

1 Generative models (abstract) (10 points)

In the lecture sessions you will learn about 6 potential parts you could find in research paper abstracts. Consider the following research paper abstract[1]

> Hit songs, books, and movies are many times more successful than average, suggesting that "the best" alternatives are qualitatively different from "the rest"; yet experts routinely fail to predict which products will succeed. We investigated this paradox experimentally, by creating an artificial "music market" in which 14,341 participants downloaded previously unknown songs either with or without knowledge of previous participants' choices. Increasing the strength of social influence increased both inequality and unpredictability of success. Success was also only partly determined by quality: The best songs rarely did poorly, and the worst rarely did well, but any other result was possible.

1. Name the 6 potential parts you could find in research paper abstracts.
2. Mark all parts you can find in the given abstract.

Answers: 1. These are the 6 potential parts in research paper abstracts:
I) State the Background and Problem you tackle with your research.
II) Name the methodology you have used.
III) Formulate 1 to 3 precise research question that are answered in your paper.
IV) Talk about your unique solution or idea.
V) Demonstrate the results.
VI) Conclude with a point of impact.

2. Now, we can classify the given research paper abstract into 6 potential parts.
I) State the Background and Problem you tackle with your research.
Hit songs, books, and movies are many times more successful than average, suggesting that "the best" alternatives are qualitatively different from "the rest"; yet experts routinely fail to predict which products will succeed.
II) Name the methodology you have used. We investigated this paradox experimentally, by creating an artificial "music market"
III) Formulate 1 to 3 precise research question that are answered in your paper.
Increasing the strength of social influence increased both inequality and unpredictability of success.
IV) Talk about your unique solution or idea.
by creating an artificial "music market" in which 14,341 participants downloaded previously unknown songs either with or without knowledge of previous participants' choices.
V) Demonstrate the results.
Increasing the strength of social influence increased both inequality and unpredictability

[1] https://www.princeton.edu/~mjs3/salganik_dodds_watts06_full.pdf

of success.

VI) Conclude with a point of impact.

Success was also only partly determined by quality: The best songs rarely did poorly, and the worst rarely did well, but any other result was possible.

2 Meme spreading model (10 points)

We provide you with the following excerpt from the meme paper[2] which will be discussed at the lecture. This part of the paper contains and explanation of their basic model. Your task is to **list five model choices** that stay in conflict with reality and **discuss the conflict**.

> Our basic model assumes a frozen network of agents. An agent maintains a time-ordered list of posts, each about a specific meme. Multiple posts may be about the same meme. Users pay attention to these memes only. Asynchronously and with uniform probability, each agent can generate a post about a new meme or forward some of the posts from the list, transmitting the cor- responding memes to neighboring agents. Neighbors in turn pay attention to a newly received meme by placing it at the top of their lists. To account for the empirical observation that past behavior affects what memes the user will spread in the future, we include a memory mechanism that allows agents to develop endogenous interests and focus. Finally, we model limited attention by allowing posts to survive in an agent's list or memory only for a finite amount of time. When a post is forgotten, its associated meme become less represented. A meme is forgotten when the last post carrying that meme disappears from the user's list or memory. Note that list and memory work like first-in-first-out rather than priority queues, as proposed in models of bursty human activity. In the context of single-agent behavior, our memory mechanism is reminiscent of the classic Yule-Simon model.
>
> The retweet model we propose is illustrated in Fig. 5. Agents interact on a directed social network of friends/followers. Each user node is equipped with a screen where received memes are recorded, and a memory with records of posted memes. An edge from a friend to a follower indicates that the friend's memes can be read on the follower's screen (#x and #y in Fig. 5(a) appear on the screen in Fig. 5(b)). At each step, an agent is selected randomly to post memes to neighbors. The agent may post about a new meme with probability p_n (#z in Fig. 5(b)). The posted meme immediately appears at the top of the memory. Otherwise, the agent reads posts about existing memes from the screen. Each post may attract the user's attention with probability pr (the user pays attention to #x, #y in Fig. 5(c)). Then the agent either retweets the post (#x in Fig. 5(c)) with probability $1 - p_m$, or tweets about a meme chosen from memory (#v triggered by #y in Fig. 5(c)) with probability p_m. Any post in memory has equal opportunities to be selected, therefore memes that appear more frequently in memory are more likely to be propagated (the memory has two posts about #v in Fig. 5(d)). To model limited user attention, both screen and memory have a finite capacity, which is the time in which a post remains in an agent's screen or memory. For all agents, posts are removed

[2] http://www.nature.com/articles/srep00335

after one time unit, which simulates a unit of real time, corresponding to Nu steps where Nu is the number of agents. If people use the system once weekly on average, the time unit corresponds to a week.

Answers:

1. Our basic model we assumes a frozen network of agents in which agent maintains a time-ordered list of posts, each about a specific meme. Agent sends some post to neighbors based on user's interest and focus. In reality, we can not predict user's choice which always differs in different situation.
2. A meme and post disappear from the list or memory like first-in-first-out. But, in real world user can make changes in meme at any time because it remains in user's mind for long time. It depends on user's behavior and social impact.
3. Each user node has a screen where received memes are recorded, and a memory with records of posted memes. An agent is selected randomly to post memes to neighbors according to their probability. But, in real world user selects meme which is more useful.
4. To model limited user attention, both screen and memory have a finite capacity, which is the time in which a post remains in an agent's screen or memory. But, user can recall the post depend on importance of the post.
5. Any post in memory has equal chances to be selected, therefore memes appear more frequently in memory are likely to be propagated. But, in reality the propagation of meme can affect from the number of friends/followers.

3 Graph and its properties (10 points)

Last week we provided you with a graph of out-links[3] of Simple English Wikipedia which should be reused this week.

Write a function that returns the diameter of the given directed network. The diameter of a graph is the longest shortest path in the graph.

Answer

```
import pandas as pd
from copy import deepcopy

def find_all_paths(graph, start_vertex, end_vertex, path=[]):
    """ find all paths from start_vertex to
        end_vertex in graph """
    path = path + [start_vertex]
    if start_vertex == end_vertex:
        return [path]
    if start_vertex not in graph:
        return []
    paths = []
    for vertex in graph[start_vertex]:
        if vertex not in path:
            extended_paths = find_all_paths(graph, vertex, end_vertex, path)
            for p in extended_paths:
                paths.append(p)
    return paths

def diameter(graph):
    """ calculates the diameter of the graph """
    v = list(graph.keys())
    pairs = [(v[i], v[j]) for i in range(len(v)) for j in range(i+1, len(v)-1)]
    smallest_paths = []
    for (start, end) in pairs:
        paths = find_all_paths(graph, start, end)
        try:
            smallest = sorted(paths, key=len)[0]
            smallest_paths.append(smallest)
        except:
            pass

    # longest path is at the end of list,
    # i.e. diameter corresponds to the length of this path
    dia = len(smallest_paths[-1])
    return dia

```

[3] http://141.26.208.82/store.zip

```python
def prepare_dict(graphs):
    """ cleans our dict from
        we found the data as
        {
            "Germany": {"out_links": ["Netherland"]}
        }
    """
    new_dict = dict()
    for key, values in graphs.items():
        new_dict[key] = values['out_links']
    return new_dict


def add_empty_node(dict_of_link):
    pure_dict = deepcopy(dict_of_link)
    all_links = set()
    for key, values in pure_dict.items():
        all_links.add(key)
        all_links |= set(values)
    for key in all_links:
        if key not in pure_dict:
            pure_dict[key] = []
    return pure_dict


def read_file(file_name):
    """ read given file and returns the content"""
    return pd.HDFStore(file_name)

if __name__ == "__main__":
    """ entry point of the application"""

    store = read_file("store.h5")
    df = store['df2']
    # Dictionary of article names and its associated article text in list form
    name_assoctext_dict = df.set_index('name').T.to_dict()
    new_dict = prepare_dict(name_assoctext_dict)
    graph = add_empty_node(new_dict)
    print("The diameter of our outlinks is: " , diameter(graph))
```

```
The diameter of our outlinks is:  9
Closing remaining open files:store.h5...done
```

3.1 Hints

1. You can first write a function that returns the shortest path between nodes and then find the diameter.
2. Do not forget to use proper data structures to avoid a memory shortage.

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment9/` 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`.