CS-E4650 Methods of Data mining

Exercise 4 / Autumn 2023

4.1 PageRank and HITS

Learning goal: The idea of PageRank and HITS algorithms.

Figure 1 shows the linkage structure of web pages and Table 1 lists the keywords that occur in the pages. The task is to evaluate PageRank and hubs and authority values of pages given a query. In this task, you can use any of the existing PageRank and HITS (simulation) tools (you can find also online calculators). Note that different tools may use different initialization or scaling but the top results should be the same. If the tool allows you to adjust the teleportation probability, use 0.10 or 0.15.

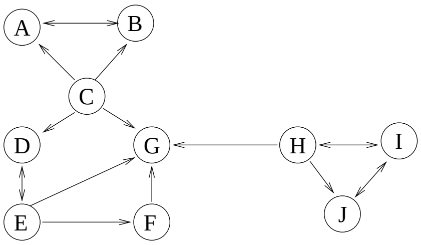

Figure 1: Linkage structure of web pages.

a) Evaluate PageRank values for all pages. What would be the most reputable sources containing query words i) "PageRank" or ii) "teleportation"?

b Construct the HITS graph (base set and its edges) for query "PageRank". Include pages in the root set, all pages pointed by the root pages and all pages pointing to the root pages. Then calculate the hubs and authority values. Which page is the best authority on the topic? What about the best hub?

c) Repeat b) for query "teleportation".

d) Compare the results with PageRank and HITS. Do they agree on the most reputable sources?

Table 1: Keywords that occur in pages A–J.

id	keywords
A	authority, page, reputable, source
B	hub, page, link, good, source
C	PageRank, HITS, ranking, algorithm
D	reputable, page, link, PageRank
E	reputation, visit, frequency, random, surfer
F	random, surfer, trap, dead-end
G	PageRank, teleportation, random, surfer, model
H	teleportation, travel, planet
I	Star Trek, transporter, teleportation
J	beam, Scotty, transporter

4.2 Collaborative filtering for movie recommendations

Learning goal: How to use neighbourhood-based collaborative filtering in recommender systems; problems of adjusted cosine similarity.

Table 2 presents movie ratings by 6 users on 6 movies. The latex source of the table is available on the course page (mratingstable.tex). The ratings are between 1 (didn't like at all) to 5 (fantastic movie) and 0 means a missing rating (the user hasn't watched the movie). The users are notated $u1, \dots, u6$ and movies $m1, \dots, m6$. The task is to apply recommender systems for rating prediction using neighbourhood-based collaborative filtering (see Aggarwal 18.5.2 and an example in the lecture).

a) Calculate mean ratings per user. Use all non-missing ratings in the calculation. These are needed in parts b) and c).

b) Calculate required pairwise similarities between users[1] using a modified Pearson correlation r ("Pearson" in Aggarwal Equation 18.12). Use the mean values calculated in part a. Remember that the correlation is calculated only over co-rated movies.

c) Predict missing ratings using two nearest neighbours ($K = 2$) and an extra requirement that the similarity is $r \geq 0.5$. Tell if the movie is recommended to the user (if the user would like it more than average).

   Report if some prediction cannot be made (not enough sufficiently similar neighbours with required ratings).

[1] Note: similarity between $u2$ and $u3$ is not needed, so 14 similarities.

d) Consider the item-based way of predicting the missing ratings of movies $m3$ and $m4$ with adjusted cosine similarity, as suggested in Aggarwal 18.5.2.2. Why it is not a good solution here? Suggest an alternative item-based solution that could be used instead (no need to calculate the actual predictions).

Table 2: Movie ratings (scale 1–5) by 6 users ($u1$–$u6$) on 6 movies ($m1$–$m6$). Special value 0 means a missing rating.

	$m1$	$m2$	$m3$	$m4$	$m5$	$m6$
u1	3	1	2	2	0	2
u2	4	2	3	3	4	2
u3	4	1	3	3	2	5
u4	0	3	4	4	5	0
u5	2	5	5	0	3	3
u6	1	4	0	5	0	0

4.3 Distances between molecular structures

Learning goal: The concept of maximum common subgraph (MCS) and related distance measures.

Figure 2 shows an example of four molecular graph structures: Niacin (vitamin B1), Nicotine (active ingredient in tobacco), psilocin (active ingredient in "magic mushrooms"), and proline (amino acid). The node labels correspond to atoms (carbon, oxygen or nitrogen)[2].

a) Determine the nearest neighbour for each molecule using Union-normalized MCS distance (*Udist* in slides, see Aggarwal Eq. 17.2).

b) Determine the nearest neighbour for each molecule using Max-normalized MCS distance (*Mdist* in slides, see Aggarwal Eq. 17.3).

c) Under which conditions are *Udist* and *Mdist* equivalent? I.e., give conditions related to some graphs G_1, G_2 and $MCS(G_1, G_2)$ such that $Udist(G_1, G_2) = Mdist(G_1, G_2)$.

[2] For simplicity hydrogen atoms and double bonds between atoms are not presented.

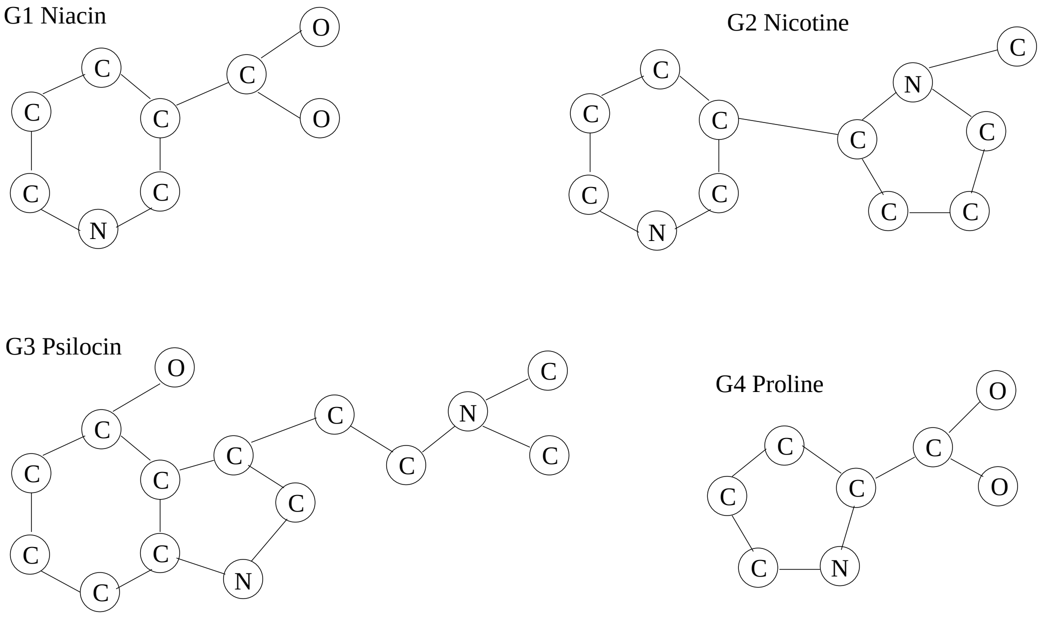

Figure 2: Four graphs corresponding to molecular structures.

4.4 Homework: Core communities

Learning goals: How to find core communities in a social networks; designing search algorithms for graph-form data.

Let us notate a social network as graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. One analysis task is to find the core of the network consisting of tightly interconnected actors (nodes) $\mathbf{S} \subseteq \mathbf{V}$. In this task, we use the following measure to evaluate interconnectedness:

$$ic(\mathbf{S}) = \frac{\sum_{i \in \mathbf{S}} \sum_{j \in \mathbf{S}} w(i,j)}{|\mathbf{S}|},$$

where $w(i,j)$ is the weight of the edge between vertices i and j, if it exists, and 0 otherwise, and $|\mathbf{S}|$ is the cardinality of the set $\mathbf{S}$. Note that in unweighted graphs, $w(i,j) \in \{0,1\}$.

a) Propose a greedy algorithm to find a core $\mathbf{S} \subseteq \mathbf{V}$ with high ic. The algorithm does not need to find the globally optimal solution, but it should converge to a good local optimum.

b) Load the Dolphin data (dolphins.txt) from MyCourses. The first line tells the number of nodes and each subsequent line indicates the two endpoints of an **undirected** edge. Estimate some reasonable lower and upper bounds of *ic* for the optimal core in the Dolphin data (these can be used to estimate goodness of your solution).

c) Apply your algorithm on the Dolphin data. As a result, provide the maximum *ic* you achieved and the vertices of the corresponding subgraph (in ascending order).

d) Visualize the graph using different colour for the core nodes **S**. For visualization, you can use Gephi (`https://gephi.org/`). What is your conclusion, does your solution capture the interconnected core well?

Parts of the report:

1. Cover page: title (course name and assignment number), names and student ids of all participants of the team.
2. Section 1 "Methods": Describe briefly the methods: how did you invent the algorithm (e.g., searched literature, applied a method you knew from another context – and give references to possible sources), how did you estimate the lower- and upperbounds, what programming language and tools you used in the implementation. If you made any extra experiments, tell them briefly.
3. Section 2 "Algorithm": Describe here your algorithm. Give clear and compact pseudocode and explain the idea. Tell if there are any special assumptions or properties (e.g., if the result depends on the order of execution).
4. Section 3: "Experiments on the Dolphin data": Describe here results of your experiments on the Dolphins data, including lower- and upperbounds on *ic*, the actual result, its visualization, and your conclusions on the quality.
5. (Optional) Section 4: "Extra experiments". If you made extra experiments (e.g., analyzed other networks or compared different algorithm approaches), describe them here.
6. Section "Appendix": Include here the code of your program.

Produce a pdf report including all parts and submit it in MyCourses before the deadline. Recall that this task is homework that is done in groups of 2–3 students – you cannot do the task alone or in a larger group. You can search collaborators in zulip, exercise sessions, or ask help from the TAs.