CDS DS210: Programming for Data Science
Report

-Anika Bhati

Section A: About the dataset

I used a dataset extracted from goodreads.com. The source of this data set is https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home[1][2]. This data set contains information about books and their genres, user reviews, etc. Listed below are the statistics:

Genre	**# books**	**# reviews**
Comics	999	75000
Mystery	999	100000
Poetry	999	145000
Children	999	56000

Section B: Running the project

Clone the GIT repository locally from https://github.com/anikab21/good_reads_v3

Perform a cargo run

Follow the instructions on the screen (***there should be a step where it asks the user if they want a deep dive of the iteration. To skip this, and directly reach the densest subgraph, enter 'n'*)

Section C: Description of the project

1. Once opened in the terminal, the user-friendly interface will guide the user through the data taking into consideration how the user wants the data to be picked.
2. The master data consists of 999 books from the Children's, Comics, Mystery, and Poetry categories. The user can select the data for analysis by simply selecting a genre and the no. of books associated with the respective genre.
3. It also provides the user with the option to add more genres if the user is interested in relationships across genres.
4. Then it will process the data to build a graph, in which each book is a vertex and an edge is drawn between any two books with a common reader. After its construction, the summary statistics of the graph are also displayed
5. Followed by this is the densest subgraph analysis by performing iterations and outputs the iteration # with the densest subgraph *(greedy algorithm method: refer to section D)*

---

[1] Mengting Wan, Julian McAuley, "Item Recommendation on Monotonic Behavior Chains", in *RecSys'18*. [bibtex]

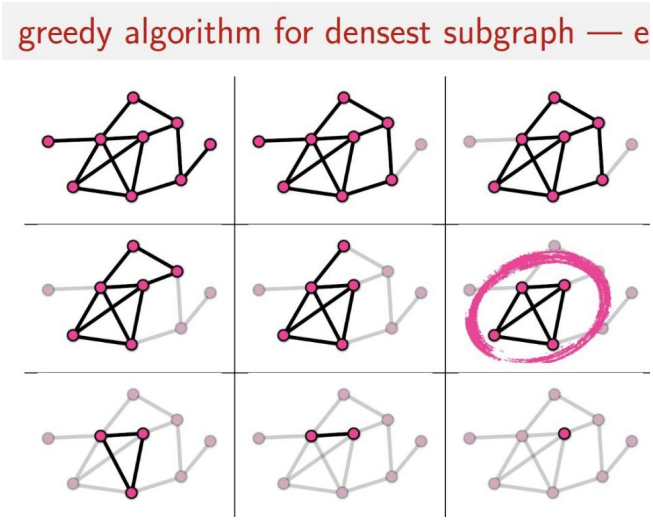
[2] Mengting Wan, Rishabh Misra, Ndapa Nakashole, Julian McAuley, "Fine-Grained Spoiler Detection from Large-Scale Review Corpora", in *ACL'19*. [bibtex]

The user has the opportunity to inspect each iteration result or directly skip to the final densest subgraph.

<u>Section D:</u> Algorithms implemented

I chose to explore the concept of the densest subgraph. In essence, this is when one vertice/node has multiple edges that connect it to another vertex.

To explain this mathematically, let V be the set of vertices of the graph.
For any $V' \subseteq V$,
$E(V')$ is used to denote the subset of edges of the graph that have both endpoints in V'.
<u>Therefore</u>, the density of a subgraph induced by V' is
$|E(V')|/|V'|$
Using this formula, my goal was to find and identify a subset $V' \subseteq V$ that maximized the density.



One type of algorithm in the densest graph is the greedy algorithm which was used in my project[3]. This is an algorithm strategy that outputs the optimal choice at each step, and gradually leads us to a globally optimal solution. In essence, this algorithm continues to eliminate nodes after each iteration and finally outputs the iteration where the densest subgraph was produced. Here's a picture to exemplify this phenomenon[4]:

As one can see, the algorithm goes through 9 iterations and eliminates a vertex at each step. After the algorithm is left with only one vertex, the optimal solution (6th iteration) is identified.

<u>Section E:</u> Output

My algorithm's output consisted of the Graph stats and the Densest subgraph (including each iteration). The graph stats provided the user with information about the vertices, edges, and the chosen genre/s. Whereas, the densest subgraph provided the user with information about which nodes form the densest sub-graph within the graph. By looking at the densest sub graph, we can infer the common readers of multiple books. As we hypothesized that clusters would form by genres, we saw a similar result in the output as well. In addition, the algorithm also reports the

[3] *Cs.Umd.Edu*, 2023, http://www.cs.umd.edu/~samir/grant/ICALP09.pdf. Accessed 4 May 2023.
[4] "Dense Subgraph Discovery: Theory And Applications (Tutorial SDM 2021)". *Tsourakakis.Com*, 2021, https://tsourakakis.com/dense-subgraph-discovery-theory-and-applications-tutorial-sdm-2021/. Accessed 2 May 2023.

result of each iteration for the user to understand how it landed on the optimal result (ie densest sub-graph).

The output of the Graph Stats:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
✔ Shall we proceed with the graph build? · yes
Great!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Graph built with:
    Vertices: 923
    Edges: 26254
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The graph stats are:
shape: (2, 2)
```

genre	count
str	u32
CHILDREN	367
MYSTERY	556

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

The output of the densest subgraph analysis:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
✔ Shall we proceed with the densest subgraph analysis? · yes
Great!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
? I will be performing 922 iterations
✔ I will be performing 922 iterations
Would you like to see result of each iteration? · no
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
THE DENSEST SUBGRAPH IS:
shape: (2, 2)
```

genre	count
str	u32
CHILDREN	118
MYSTERY	129

<u>Section F:</u> Interesting findings

When a sample of 1600 books is selected, out of which comics, mystery, children and poetry and equally divided (400 each), here's how the outputs look:

```
~~~~~~~~~~~~~~~~~~~~~~~~~
 Graph built with:
    Vertices: 1506
    Edges: 56326
~~~~~~~~~~~~~~~~~~~~~~~~~
The graph stats are:
shape: (4, 2)
```

genre	count
str	u32
POETRY	371
CHILDREN	374
MYSTERY	377
COMICS	384

```
~~~~~~~~~~~~~~~~~~~~~~~~~
THE DENSEST SUBGRAPH IS:
shape: (4, 2)
```

genre	count
str	u32
POETRY	69
MYSTERY	86
COMICS	96
CHILDREN	105

The results signify that all genres are equitably divided telling us, the vertices are all almost the same. In this example's densest subgraph, we see that Children's books dominate the sub graph, followed by comics, mystery and poetry. This algorithm reveals how strongly connected genres all to each other.

Similarly, when a sample of 1600 books is selected where mystery books are 1000 and the others are 200 each, here's how the outputs look:

```
~~~~~~~~~~~~~~~~~~~~
 Graph built with:
    Vertices: 1520
    Edges: 62373
~~~~~~~~~~~~~~~~~~~~
The graph stats are:
shape: (4, 2)
```

genre	count
str	u32
POETRY	181
CHILDREN	187
COMICS	190
MYSTERY	962

```
THE DENSEST SUBGRAPH IS:
shape: (4, 2)
```

genre	count
str	u32
POETRY	22
COMICS	34
CHILDREN	36
MYSTERY	249

Here, since we are overloading with one genre (mystery), the graph stats display a larger number of vertices as compared to any other genre. A similar result, is reflected in the densest subgraph as well where mystery has the highest count. From these outputs, we can deduce that mystery books are not strongly connected to children's book indicating that most mystery books are made for adults in real life.

Ultimately, the output is always predicted by the number and the proportion in which we are selecting books of genres.

Section G: Project's Relevance

This project gave me plenty knowledge of graph traversals by coming out with an insghfutl view of the data underneath. The selected data set was huge and this algorithm gave me the opportunity to organize the data using attributes of my preference.

This kind of research can act as a starting point to move towards a more complex business solutioning such as recommending books based off of commonalities between genres For instance, helping readers identify the most popular choice of genre or book after exploring a particular genre/book. E-commerce sites like Amazon could use this algortim to generate more business revenue potential.

Overall, this approach could be applied in real life to discover connections between variables outside of books by using clustering at different granularities to identify the hierarchy of clusters within any data set.