Daniel Vidal 1634599
Neil de la Fuente 1630223
Jordi Longaron 1630483

Report on Session 1: Data acquisition and storage

Introduction

In this session, we got access to the Spotify API to retrieve information from the Spotify data. This first session consists of the acquisition of data, showing where we can obtain it, how to do it, and showing the obstacles we may encounter during the process using Python scripts (or colab Notebooks).

Questions to answer in the report:

1. Provide the order and size of the graphs g_B and g_D.

Graph g_B has order 779 and size 4000
Graph g_D has order 554 and size 4020

(a) Explain why, having explored the same number of nodes, the order of the two graphs (g_B and g_D) differs.

Because they follow different strategies:

BFS explores the graph in a breadthward motion and uses a queue data structure to remember to get the next vertex to start a search when a dead end occurs in any iteration. It visits nodes level by level. For example, starting from a certain node, it first explores all its neighbors before moving onto the neighbors' neighbors. As a result, BFS provides a "flatter" graph structure, capturing the immediate neighbors first and then moving on to more distant nodes.

On the other hand, DFS uses a stack data structure to remember to get the next vertex to start a search when a dead end occurs. DFS explores as far as possible along each branch before backing up. It dives deeper into the graph, exploring a single path as far as it can go before backtracking and exploring another path. As a result, DFS can provide a "deeper" graph structure compared to BFS.

(b) Justify which of the two graphs should have a higher order.

BFS explores all the neighbors of the current node before moving on to the next level neighbors. In this particular situation where every node has exactly 20 related nodes, BFS will expand these 20 before it goes to the next depth level. This behavior gives BFS the potential to touch a greater number of unique nodes faster than DFS.

On the other hand, depth-first search (DFS) goes as deep as possible down one path before backtracking. So it would explore one node from the 20 related nodes, then move to one node from the related nodes of the recently visited node, and so on. In such scenarios, DFS is likely to explore fewer unique nodes if the number of nodes to be expanded is capped.

(c) Explain what size the two graphs should have.

The size of graphs should be closer to 4000.

2. Indicate the minimum, maximum, and median of the in-degree and out-degree of the two graphs (g_B and g_D). Justify the obtained values.

Graph gB	Minimum	Maximum	Median
in-degree	0	34	3
out-degree	0	20	0

Graph gD	Minimum	Maximum	Median
in-degree	0	69	2
out-degree	0	20	0

The minimum in-degree is because the root has 0 indegree in both graphs, the minimum out-degree is 0 because of the leaf nodes that were not expanded. Given that each non-leaf node has 20 out-degree neighbors, it's possible that the nodes visited by DFS tend to have a higher in-degree because they might be more well-connected or popular artists. DFS, by its nature of exploring deeper paths, could be reaching these nodes more readily. Meanwhile, BFS could be capturing a more localized snapshot of the network, including potentially less popular artists (with a lower in-degree) that are closely connected to the seed node.

3. Indicate the number of songs in the dataset D and the number of different artists and albums that appear in it.

(a) Explain why the number of artists is between 200 and 400, considering the input graphs.

(b) Justify why the number of songs you obtained is correct, considering the input graphs.

(c) Justify why the number of retrieved albums is correct.