Relatório Técnico:

TDE5 - Projeto Colaborativo 2

Grupo: Alex Menegatti Secco, Mariana de Castro e Tarso Bertolini Rodrigues

1. Introdução

O presente relatório descreve o funcionamento de um sistema computacional desenvolvido para analisar relações entre atores e diretores com base em dados extraídos de títulos de plataformas de streaming, tais como Netflix, Amazon Prime e Disney+. A análise se fundamenta na construção e exploração de grafos, visando identificar propriedades estruturais das redes de colaboração entre os profissionais do cinema. O código implementado realiza a leitura dos dados, a modelagem de grafos e a aplicação de métricas de teoria dos grafos para extrair informações relevantes. Para garantir escalabilidade e eficiência computacional, a biblioteca `igraph` foi adotada como base para todas as operações com grafos.

2. Objetivo do Código

O sistema tem como principal objetivo construir representações em grafo das relações extraídas do conjunto de dados e aplicar algoritmos de análise estrutural sobre essas representações. Os grafos construídos servem para capturar dois tipos de relações principais:

- **Grafo direcionado**: modela a relação entre atores e diretores, onde cada aresta direcionada conecta um ator a um diretor com o qual trabalhou.

- **Grafo não-direcionado**: modela a relação de co-participação entre atores, conectando atores que atuaram juntos em uma mesma produção.

A partir desses grafos, são extraídas métricas clássicas da teoria dos grafos, como componentes conexas, árvore geradora mínima e centralidades (grau, intermediação e proximidade).

3. Estrutura e Funcionamento do Código

3.1 Leitura e Pré-processamento dos Dados

A leitura do arquivo CSV é realizada por meio do módulo `csv` da linguagem Python. O código percorre cada linha do arquivo e extrai os campos de interesse: `director` e `cast`.

Para garantir a uniformidade dos dados, é aplicada uma função de limpeza que transforma os nomes para letras maiúsculas e remove espaços em branco indesejados.

Esse pré-processamento é essencial para evitar a criação de múltiplos vértices para um mesmo nome que aparece com variações tipográficas. Os nomes padronizados são então utilizados como base para construir as arestas dos grafos.

3.2 Construção dos Grafos com igraph

Para maximizar o desempenho na manipulação de grandes volumes de dados, os grafos são construídos utilizando a biblioteca `igraph`, que é otimizada para processamento eficiente de grafos com dezenas ou centenas de milhares de vértices.

Durante a construção, os nomes dos profissionais são convertidos em índices numéricos, conforme exigido pelo modelo interno da biblioteca. Em seguida, as arestas são adicionadas com base na lógica da relação representada:

- No grafo direcionado, para cada combinação ator-diretor, uma aresta é criada do ator para o diretor.

- No grafo não-direcionado, todos os pares de atores que aparecem em um mesmo título são conectados entre si.

Após a adição das arestas, os nomes reais dos profissionais são reatribuídos aos vértices, utilizando o atributo `name`.

3.3 Atividades de Análise

Após a construção dos grafos, são realizadas seis atividades principais, descritas a seguir.

Atividade 1 – Contagem de vértices e arestas

Esta etapa consiste em exibir a quantidade total de vértices e arestas presentes nos dois grafos. Essa contagem fornece uma visão geral da complexidade e densidade das redes, o que é relevante para estimar o custo computacional das análises subsequentes.

Atividade 2 – Identificação de componentes conexas

A análise de componentes conexas permite compreender a fragmentação da rede. Para o grafo direcionado, são extraídas as componentes fortemente conexas, ou seja, subconjuntos nos quais cada vértice é alcançável a partir de qualquer outro vértice por meio de caminhos direcionados. Já no grafo não-direcionado, são identificadas componentes conexas simples, que representam grupos de atores interligados por meio de colaborações diretas ou indiretas.

Atividade 3 – Cálculo da árvore geradora mínima

Nesta etapa é gerada a árvore geradora mínima (MST) a partir do grafo não-direcionado, utilizando o algoritmo interno da biblioteca `igraph`. A MST representa uma subestrutura do grafo que conecta todos os seus vértices com o menor número possível de arestas e custo total mínimo. Embora os pesos das arestas não sejam considerados neste contexto, a estrutura resultante fornece uma visão simplificada da conectividade essencial da rede.

Atividade 4 – Cálculo da centralidade de grau

A centralidade de grau é uma métrica simples e eficiente que mede o número de conexões diretas que um vértice possui. Neste código, a centralidade é normalizada pela quantidade máxima possível de conexões, considerando o tamanho total da rede. Esta métrica é útil para identificar atores com alta participação em colaborações.

Atividade 5 – Cálculo da centralidade de intermediação

A centralidade de intermediação (betweenness centrality) quantifica o quanto um vértice participa como intermediário nos caminhos mais curtos entre outros pares de vértices. Como o cálculo exato dessa métrica é computacionalmente custoso em redes grandes, a implementação utiliza a versão com o parâmetro `cutoff`, que limita o cálculo a caminhos com no máximo três arestas. Essa abordagem permite uma estimativa mais rápida, ainda que menos precisa, da importância estrutural de um vértice.

Atividade 6 – Cálculo da centralidade de proximidade

A centralidade de proximidade (closeness centrality) mede a inversa da soma das distâncias mínimas entre um vértice e todos os demais vértices alcançáveis. Esta métrica indica o quão central um vértice está dentro da estrutura global do grafo. A versão normalizada da métrica é utilizada, como previsto pela biblioteca `igraph`.

4. Estratégias de Otimização

Durante o desenvolvimento do código, foram identificados e solucionados gargalos de desempenho, sobretudo nas métricas de centralidade. A versão inicial, baseada em estruturas de dados personalizadas com dicionários aninhados, foi substituída pelo uso integral da biblioteca `igraph`. Essa substituição proporcionou ganhos substanciais em velocidade de execução e consumo de memória.

Além disso, foram adotadas práticas específicas para acelerar o cálculo de métricas pesadas, como:

- Utilização de índices inteiros em vez de strings nos grafos.
- Limitação da profundidade de caminhos na centralidade de intermediação.
- Execução de métricas individualizadas apenas para vértices específicos, evitando cálculos globais desnecessários.

5. Conclusão

O sistema desenvolvido apresenta uma solução robusta e eficiente para a análise de grafos extraídos de dados textuais sobre produções audiovisuais. A estrutura do código reflete um raciocínio lógico bem segmentado, onde cada etapa do processamento é implementada com clareza e foco em desempenho.

A adoção da biblioteca `igraph` permitiu escalar a solução para conjuntos de dados com dezenas de milhares de vértices, mantendo tempos de execução aceitáveis para análises interativas. As métricas aplicadas oferecem diferentes perspectivas sobre a estrutura da rede, permitindo tanto análises locais (por vértice) quanto globais (por componente).

O resultado final é um sistema capaz de apoiar investigações sobre redes de colaboração artística, com aplicações potenciais em estudos de redes sociais, análise de dados culturais e visualização de grafos.