Trabalho Final

Algoritmos e Estruturas de Dados I

Michel Pires, Centro Federal de Educação Tecnológica de Minas Gerais April 27, 2025

1 Objetivo

O trabalho consiste no desenvolvimento de um sistema de recomendação que, a partir de perfis de usuários e características de itens, seja capaz de sugerir agrupamentos de elementos similares. Cálculos como medida de distância euclidiana, similaridade do cosseno ou Jaccard devem ser considerados como alternativas. As equipes, formadas por no máximo cinco alunos, serão avaliadas conforme critérios de eficiência computacional, qualidade dos resultados, organização do código e documentação, conforme descrito a seguir.

2 Instruções para Obtenção e Processamento dos Dados

Download da Base de Dados: Cada equipe deverá realizar o download manual da base de dados MovieLens 25M por meio do repositório Kaggle, disponível em MovieLens 25M. O arquivo principal a ser utilizado é o `ratings.csv`; entretanto, recomenda-se que as equipes explorem também os demais arquivos disponibilizados, uma vez que informações complementares podem contribuir significativamente para a melhoria da qualidade das recomendações geradas.

Pré-processamento: O pré-processamento deverá seguir os seguintes critérios:

- Utilizar apenas usuários que tenham realizado pelo menos 50 avaliações distintas.
- Utilizar apenas filmes avaliados por pelo menos 50 usuários.
- Remover registros duplicados ou inconsistentes.
- Gerar arquivo de entrada no seguinte formato:

```
usuario_id item_id1:nota1 item_id2:nota2 item_id3:nota3 ...
```

Exemplo: `123 12:4.0 54:3.5 76:5.0 145:2.0`

Note que, cada linha representa um usuário (`usuario_id`) e suas respectivas avaliações (`item_id:nota`).

2.1 Das disposições sobre os arquivos de entrada e saída

Para o arquivo que será utilizado como fonte de informação para a execução das recomentações, tem-se como padrões a serem seguidos:

- Nome do arquivo: `input.dat`
- Localização: Diretório `datasets/`
- Formato: Texto puro (UTF-8)
- Necessidade: Arquivo que representará a base de dados para exploração

Já para os arquivos de exploração e de saída, espera-se que ambos sigam os seguintes padrões:

- Nome do arquivo: `explore.dat`
- Localização: Diretório `datasets/`
- Formato: Texto puro (UTF-8)
- Necessidade: Arquivo que conterá os usuários utilizados para exploração das recomendações
- Nome do arquivo: `output.dat`
- Localização: Diretório `outcome/`
- Formato: Texto puro (UTF-8)
- Necessidade: Arquivo contendo as K recomendações para cada usuário apresentados no arquivo `explore.dat`

3 Definições de Execução

Compilação e Execução: As execuções devem, obrigatoriamente, seguir os padrões apresentados no `Makefile` fornecido como parte do material da disciplina. Esse, por sua vez deve permitir a execução através do padrão:

```
make clean
make
make run
```

Em conformidade com o estabelecido na seção anterior, o arquivo `explore.dat` conterá a lista de `usuario_id`s para os quais o sistema deverá gerar recomendações personalizadas.Para cada `usuario_id` listado, o procedimento a ser seguido consiste em:

- Realizar a busca na base de dados para identificar os filmes previamente avaliados por este usuário.
- A partir desse conjunto de avaliações, calcular a similaridade ou afinidade do usuário em relação aos demais usuários da base, utilizando a métrica de distância ou similaridade definida pelo projeto.

- Selecionar os K usuários mais similares (de maior afinidade) ao usuário em análise.
- Identificar os filmes avaliados positivamente pelos usuários similares, mas ainda não avaliados pelo usuário-alvo, priorizando aqueles com maior grau de sobreposição de interesse.
- A partir dessa análise, gerar as recomendações a serem atribuídas a cada `usuario_id`.

Os resultados deste processo deverão ser armazenados no arquivo `output.dat`, obedecendo o formato estipulado, em que cada linha corresponde a um `usuario_id` seguido pelos `item_id`s recomendados.

3.1 Ambiente de Execução

- Sistema operacional: Linux Ubuntu 24.04 LTS
- Compilador: gcc 13 ou superior
- Uso exclusivo da biblioteca padrão da linguagem C.

3.2 Formato da Saída

O arquivo `output.dat` deverá conter o formato abaixo, sendo o número de recomendações (*Top-N*) será definido via constante global no arquivo `config.h`.

```
usuario_id item_id1 item_id2 item_id3 ...
```

Exemplo: `123 54 76 145`

4 Composição do Ranking

Neste trabalho vamos compor um ranking das execuções. A equipe que apresentar o melhor índice na avaliação fica dispensada da realização da prova final da disciplina. Para ter tal benefício, o algoritmo apresentado não pode ter tempo superior a 2.5 segundos de execução por recomendação / usuário do arquivo `explore.dat`.

Critério	**Peso**	**Descrição**
Tempo de execução	40%	Menor tempo médio de execução (média de 10 execuções).
Qualidade da recomendação	10%	Avaliação da cobertura e precisão das recomendações.
Consumo de memória	10%	Avaliado através de medições via `valgrind --tool=massif`.
Organização do código	20%	Modularização adequada e boas práticas de programação.

Clareza da documentação	20%	Metodologia, estudo de caso, discussão de limitações e melhorias futuras.

5 Critérios de Avaliação Geral

- Código modularizado.
- Documentação clara, metodologias descritas e análise de performance.
- Utilização correta das estruturas de dados estudadas.
- Preservação da integridade do processo de compilação e execução.

6 Responsabilidades das Equipes

- Garantir que o projeto compile e execute conforme especificado.
- Incluir Makefile funcional.
- Assegurar reprodutibilidade do projeto apenas com os arquivos entregues.

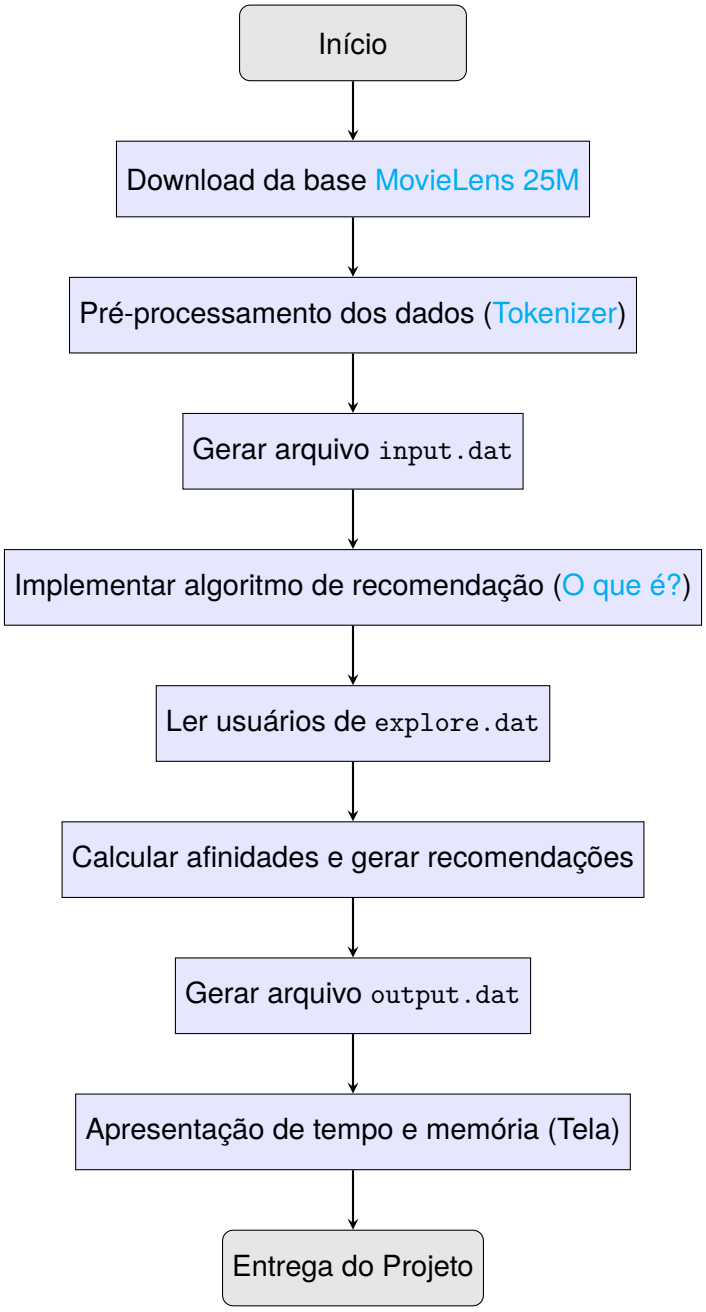
Falhas em compilação ou execução resultarão na desclassificação.

7 Observações Finais

- Bases alternativas são permitidas apenas mediante aprovação prévia do professor.
- Estruturas de dados não estudadas devem ser justificadas e aprovadas.
- Alterações no formato de entrada ou saída após submissão não serão aceitas.

Data das apresentações: 04 a 11 de julho.
Data da entrega por todas as equipes: 03 de julho até as 23h:59m.
Linguagens Permitidas: C e C++
Valor: 30 pontos.

Fluxograma


Início
Download da base MovieLens 25M
Pré-processamento dos dados (Tokenizer)
Gerar arquivo input.dat
Implementar algoritmo de recomendação (O que é?)
Ler usuários de explore.dat
Calcular afinidades e gerar recomendações
Gerar arquivo output.dat
Apresentação de tempo e memória (Tela)
Entrega do Projeto