Project Proposal: Content-Based Filtering Algorithm

Pump Vanichjakvong

January 18, 2024

1 Academic Reference

We plan to implement a content-based filtering algorithm based on the following academic reference:

- **Title:** A Content-Based Movie Recommendation System
- **Authors:** Suja Cherukullapurath Mana and T. Sasipraba
- **Published In:** Journal of Physics: Conference Series, Volume 1770, Number 1, 2021
- **Link:** `https://iopscience.iop.org/article/10.1088/1742-6596/1770/1/012014/pdf`

2 Algorithm Summary

The algorithm we aim to implement is a content-based recommendation system. Content-based filtering recommends items to users based on the characteristics and features of the items and the user's preferences. In our case, we are developing a recommendation system for anime shows. The algorithm takes into account various factors, such as user reviews, show genres, show length, and the aired date. Additionally, we consider user preferences like age group and time period preferences for anime shows. The algorithm calculates the similarity between a user's profile and the features of anime shows to recommend the most relevant shows to the user. Cosine similarity is a fundamental component of the algorithm used to measure the similarity between user preferences and show features.

3 Function I/O

Our implementation will consist of the following core functions/struct:

3.1 AnimeShow Struct

struct AnimeShow {
`int episodes;`
`double score;`
`std::string binary_genre;`
};

- **Attributes:**
  - `episodes` – Number of episodes in the anime show.
  - `score` – Score or rating of the anime show.
  - `binary_genre` – Genre of the anime show encoded in binary format.

3.2 Functions

3.3 `double dotProduct(const std::vector<double> vec1, const std::vector<double> vec2)`

- **@param** vec1 – The first vector for dot product calculation.
- **@param** vec2 – The second vector for dot product calculation.
- **@return** The dot product of vec1 and vec2.

Test Cases:

- **Test 1:** Test for Regular Vectors.
- **Test 2:** Test for Empty Vectors.

3.4 `double magnitude(const std::vector<double> vec)`

- **@param** vec – The vector to calculate the magnitude.
- **@return** The magnitude of the vector.

Test Cases:

- **Test 1:** Test for Regular Vector.
- **Test 2:** Test for Empty Vector.

3.5 `double cosineSimilarity(const AnimeShow show1, const AnimeShow show2)`

- **@param** show1 – The first anime show for comparison.
- **@param** show2 – The second anime show for comparison.

- **@return** The calculated cosine similarity value between show1 and show2.

Test Cases:

- **Test 1:** Test for Different Shows.
- **Test 2:** Test for Similar Shows.

3.6 `std::vector<AnimeShow> findSimilarAnimes(const std::vector<AnimeShow> database, const std::vector<AnimeShow> favoriteAnimes, int k)`

- **@param** database – A vector of AnimeShow containing the entire database of anime shows.
- **@param** favoriteAnimes – A vector of AnimeShow containing the user's favorite anime shows.
- **@param** k – The number of recommended shows to return.
- **@return** A vector of AnimeShow representing the titles of recommended similar anime shows.

Test Cases:

- **Test 1:** Test with Common Shows.
- **Test 2:** Test for Varying 'k' Values.

Both test cases aim to validate the function's effectiveness on a small-scale sample.

4 Math Behind the Algorithm

The core mathematical concept used in our content-based recommendation algorithm is cosine similarity. Cosine similarity is a measure of similarity between two non-zero vectors in an inner product space. In our case, we use it to measure the similarity between the user's preferences (represented as a vector) and the features of anime shows (also represented as vectors).

Cosine similarity between two vectors A and B is calculated as:

$$cosine\_similarity(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Where:

- $A \cdot B$ represents the dot product of vectors A and B.
- $\|A\|$ and $\|B\|$ represent the magnitudes (or Euclidean norms) of vectors A and B, respectively.

In the context of our algorithm, the user's preferences and anime show features are represented as vectors, and the cosine similarity score is computed to measure how similar the user's preferences are to the features of each anime show. Higher cosine similarity scores indicate greater similarity, which helps in recommending anime shows that align with the user's preferences.

5 Data Description

The data will be sourced from MyAnimeList containing comprehensive information about anime shows and user reviews. Upon extraction, this data will be meticulously processed into a CSV format, ensuring structured representation of the various attributes. The dataset will encompass extensive details about anime show ratings, and hot encoded type stream of genres (1 present 0 not present), as well as the show's name to be able to return readable results.

6 Project Proposal Overview

Our primary objective is to develop and implement a content-based filtering algorithm tailored for recommending anime shows to users based on their preferences and the comprehensive features present in the dataset. Cosine similarity algorithm will serve as a fundamental mathematical measure, enabling the evaluation of the likeness between user preferences and the distinctive attributes characterizing each anime show. The project will encompass the entirety of the software development life cycle, commencing with data collection and processing, integrating the calculation of cosine similarity, and ultimately, culminating in the delivery of the final anime show recommendations to users.