CSE 573: Semantic Web Mining

Group 21: Project Proposal

Movie Recommendation System

Group Members

- Ninad Nale | nnale@asu.edu | 1225710226
- Gaurav Hoskote | ghoskote@asu.edu | 1225134352
- Rishabh Saraf | rasaraf@asu.edu | 1225332719
- Herambh Shah | hshah75@asu.edu | 1225467789
- Amey Bhilegaonkar | abhilega@asu.edu | 1225368924
- Pravinkumar Tiwari | ptiwar23@asu.edu | 1225419676

Abstract

In today's world, where we have countless options for streaming movies, online shopping, and social networking, a crucial component in all these platforms is the one that provides recommendations based on user preferences. A recommender system is essential to tailor the content on any platform to the user's choices and needs. With the constant influx of new movies and TV series from streaming services like Netflix, Peacock, etc. we need recommendation systems to help users discover content they'll enjoy. These systems analyze the kind of content users prefer to watch and suggest personalized content based[7] on relevant data. This not only makes it easier for users to find what they like but also benefits businesses by identifying popular content that can boost revenue by increasing the user interaction and influx. Using clustering algorithms, movie recommendation[5] systems navigate a vast amount of data to provide accurate suggestions. In this study, we have evaluated different models like Collaborative Filtering[1], Matrix Factorization[2], and Item-based KNN[5], testing them on the 20M MovieLens[4] dataset to predict user movie preferences.

Relevant keywords: Recommendation System[6], Clustering, Collaborative Filtering[1], Matrix Factorization[2], K-Nearest-Neighbors.

Problem Definition

Movies serve as the cornerstone of the entertainment industry. In the age of on-demand entertainment, where platforms like Netflix, Disney, Peacock and others skillfully use technology and data to curate a sizable movie collection and deliver personalized recommendations influenced by users' viewing habits, movies serve as the foundation of the entertainment industry and have a significant economic impact on the world. The rapid growth of on-demand entertainment platforms, such as Netflix, Disney, Peacock, and others, has led to an overwhelming choice of movies for users. While these platforms effectively employ advanced technology and data-driven approaches to curate vast movie collections and provide personalized recommendations, the primary challenge is to deliver precise and engaging movie suggestions. The problem is that users often face decision fatigue and may struggle to discover movies that align with their preferences, resulting in suboptimal user engagement and potential loss of revenue for these platforms. To address this issue, our project aims to explore and evaluate two fundamental recommendation algorithms: Content-Based[7] Recommendation Systems[6] and Collaborative Filtering[1] Recommendation Systems[6]. Specifically, we will delve into the application of the K-Nearest-Neighbor algorithm and incorporate advanced Collaborative Filtering[1] techniques, including Matrix Factorization and Restricted Boltzmann Machine. Through this research, we seek to enhance the effectiveness of movie recommendations, thereby improving user satisfaction and ultimately contributing to the economic success of internet-based entertainment companies. Our goal is to assess the performance of different algorithms for the process of movie recommendation and find an efficient algorithm for this use-case. More information on the dataset used in this project may be found in the DataSet section below.

Data Set Description

For this project, we've chosen the MovieLens[4] 20M Dataset as it fits our needs well. This dataset is organized into six files: tag.csv, rating.csv, movie.csv, link.csv, genome_scores.csv, and genome_tags.csv, all saved in CSV format. The dataset includes movie titles, user-applied tags, and ratings from a diverse group of people. It comprises 20,000,263 ratings and 465,564 tag applications across 27,278 movies. This information was generated by 138,493 users between January 9, 1995, and March 31, 2015. All selected users had rated at least 20 movies.

Key Features of the Dataset:

- Movie Metadata: The dataset includes comprehensive information about movies, such as titles, release years, genres, and tags. This metadata serves as the foundation for content-based[7] recommendation techniques.
- User Ratings: Over 20 million user ratings provide insights into user preferences and the core data for collaborative filtering approaches.

- User-Item Interactions: Beyond ratings, the dataset includes user-item interactions, such as user tags and user-movie timestamps. These interactions add depth to our analysis, helping us understand the dynamics of user behavior and preferences over time.
- Data Diversity: With a diverse range of movies, genres, and user profiles, the dataset enables us to handle a wide array of user preferences and movie types, ensuring that our recommendation system[6] caters to a broad audience.

- **tag.csv** contains tags assigned to movies by users, with attributes: userId, movieId, tag, and timestamp.
- **rating.csv** includes user ratings for movies, ranging from 0.5 to 5 in increments of 0.5, with columns: userId, movieId, rating, and timestamp.
- **movie.csv** lists movie titles with corresponding genres, having columns: movieId, title, and genres.
- **link.csv** provides identifiers in an external movie database like IMDB, with columns: movieId, imdbId, and tmdbId.
- **genome_scores.csv** presents data on the relevance of movie-tags, with columns: movieId, tagId, and relevance.
- **genome_tags.csv** offers tag descriptions, with columns: tagId and tag.

State of the Art Methods & Algorithms

Movie recommender systems aim to provide personalized suggestions to users based on their preferences and interests. Several state-of-the-art methods and algorithms have been developed to achieve this goal. Here, we present a clear and concise overview of these methods and algorithms.

Content-Based Filtering:

Content-Based Filtering[7] focuses on the attributes of items (in this case, movies) to recommend similar items to users based on their past preferences. This method uses features such as genre, release date, director, actors, and critic ratings to make recommendations. The algorithm creates a user profile based on their interactions with items and calculates the similarity between items using measures like cosine similarity. Recommendations are made by selecting the top K nearest neighbors and predicting the user's rating using a weighted average of the neighbors.

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\cdot\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Where Ai and Bi are ith components of A and B respectively.

Collaborative Filtering:

Collaborative Filtering[1] leverages the collective behavior users to make recommendations. It focuses on the relationships between users and items and can be divided into two main types:

a. User-based Collaborative Filtering: This method finds users with similar preferences to the target user based on their past behavior (e.g., ratings). It then recommends items that these similar users have liked or interacted with. Similarity between users can be calculated using measures like Pearson correlation coefficient, cosine similarity, or Jaccard similarity.

b. Item-based Collaborative Filtering: This method finds items similar to the ones the target user has liked or interacted with. It then recommends these similar items to the user. Similarity between items can be calculated using the same measures as in user-based collaborative filtering[1], but applied to the item-user preference matrix.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r	=	correlation coefficient
xi	=	values of the x-variable in a sample
$\bar{x}$	=	mean of the values of the x-variable
yi	=	values of the y-variable in a sample
$\bar{y}$	=	mean of the values of the y-variable

Matrix Factorization:

Matrix Factorization[2] is a dimensionality reduction technique that decomposes the user-item preference matrix into two lower-dimensional matrices, one representing users and the other representing items. Popular matrix factorization[2] techniques for recommender systems include Singular Value Decomposition (SVD) and Alternating Least Squares (ALS). These techniques help in predicting user preferences and handling sparse data.

K-Nearest Neighbors (KNN):

KNN[5] is a simple and effective algorithm for recommendation systems[6]. It can be used for both user-based and item-based collaborative filtering. The algorithm finds the K most similar users or items to a given user or item and recommends items based on the preferences of those similar users or items. The similarity can be measured using various distance metrics, such as cosine similarity or Pearson correlation coefficient.

Research Plan:

After the topic of the project was finalized, we started the first phase of our project with studying the available content in the form of research papers and blogs, related to the domain of recommender systems and movie recommendation. Once we got familiar with different models being used to build recommendation systems[6], we started the exploratory analysis for the Movielens[4] dataset that we are building this project upon. This has helped us find relevant patterns in the dataset and key attributes we need to incorporate into the discussed models. After this our research plan has been as follows:

Phase 1: Model Selection & Familiarization

The emphasis will be on choosing specific algorithms for content-based[7] filtering and collaborative filtering[1], with a central focus on matrix factorization[2] techniques. Our approach will involve a careful evaluation of the pros and cons associated with each method, ensuring that the selected algorithms align optimally with the objectives of our project. Simultaneously, we will initiate the essential process of data preprocessing for the MovieLens[4] dataset. This phase encompasses data cleaning, addressing missing values, and data quality assurance. We will also explore feature engineering techniques to extract attributes that are pertinent to the development of our recommendation models. This phase lays a strong foundation for the subsequent stages of the project, shaping the direction of our research and development efforts.

Phase 2: Model Implementation & Evaluation

In this phase we will focus on model implementation and evaluation. We will begin by implementing a Content-Based Filtering[7] model, which incorporates movie attributes such as genre, cast, and tags. This model will be carefully fine-tuned to ensure optimal performance, and its effectiveness will be assessed using common metrics like Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Simultaneously, we will develop a Collaborative Filtering[1] model, utilizing matrix factorization techniques[2]. To provide a robust evaluation, we will experiment with various matrix factorization[2] methods, including Singular Value Decomposition (SVD) and Alternating Least Squares (ALS). The effectiveness of the collaborative filtering[1] model will be assessed through a broader set of metrics, including precision, recall, and F1-score. Additionally, in this phase, we will explore the concept of Hybrid Models, which aim to leverage the strengths of both content-based[7] and collaborative filtering[1] techniques. These hybrid models will be implemented and rigorously evaluated to determine their potential to enhance movie recommendations. By comparing the performance of these models, we will gain valuable insights into the most effective approaches for our recommendation system[6].

Phase 3: Evaluation and Reporting

In this phase, we will focus on evaluation, documentation, and presentation. We will conduct a thorough evaluation of our recommendation models, emphasizing their accuracy, coverage, and user engagement metrics. Subsequently, we will compile our findings and insights into a comprehensive project report, detailing our methodology, results, challenges encountered, and recommendations for future enhancements. Finally, we will prepare a presentation to showcase the project's methodologies, demo and work towards the given problem statement.

Evaluation Plan:

The mentioned metrics - Precision, Recall, F1-measure, False-positive rate, Mean Average Precision, Mean Absolute Error, and Area Under the ROC Curve (AUC) - are commonly used to evaluate recommender systems. Here's how each metric can be used and interpreted in the context of recommender systems:

Precision:

- Use: Measures the proportion of correctly recommended items (relevant items) out of all items recommended.
- Interpretation: Higher precision indicates a higher ratio of relevant recommendations among the total recommendations, implying the system is suggesting relevant items to users.

$$Precision = True\ Positives/(True\ Positives + False\ Positives)$$

Recall:

- Use: Measures the proportion of correctly recommended items out of all relevant items.
- Interpretation: Higher recall suggests that the system is effective at capturing a significant portion of all the relevant items.

$$Precision = True\ Positives/(True\ Positives + False\ Negatives)$$

F1-Measure:

- Use: Harmonic means of precision and recall, providing a balance between the two metrics.
- Interpretation: F1-measure is useful when both precision and recall are important. It ensures a trade-off between precision and recall, giving a single metric for evaluation.

$$F1\ Measure = 2 * Precision * Recall/(Precision + Recall)$$

4. **False-Positive Rate:**

- Use: Measures the proportion of incorrectly recommended items out of all irrelevant items.

- Interpretation: A lower false-positive rate implies that the system is effectively minimizing irrelevant recommendations.

$$False\ Positive\ Rate\ = False\ Positives/(False\ Positives\ +\ True\ Negatives)$$

Mean Average Precision (MAP):

- Use: Averages the precision at different recall levels, giving a more comprehensive assessment.
- Interpretation: Higher MAP indicates that the recommender system consistently provides relevant recommendations across varying recall levels.

$$MAP\ =\ (\sum Precision\ at\ Ki\ *\ Rel(i))/Total\ Relevant\ Items$$

Mean Absolute Error (MAE):

- Use: Measures the average difference between predicted ratings and actual ratings.
- Interpretation: A lower MAE suggests a smaller discrepancy between predicted and actual ratings, indicating better prediction accuracy.

Area Under the ROC Curve (AUC):

- Use: Evaluates the model's ability to distinguish between positive and negative items.
- Interpretation: A higher AUC value (closer to 1) indicates that the recommender system can effectively rank positive items higher than negative ones.

These metrics collectively provide insights into the performance of a recommender system in terms of accuracy, relevance, trade-offs between precision and recall, and prediction errors. We will consider evaluating our movie recommender system using a combination of these metrics which will help us understand its strengths and weaknesses and in fine-tuning the system for optimal performance.

Project Timeline

Task	Description	Deadline
Research and compile relevant papers	Conducted comprehensive research on the problem statement, pinpointed subtasks, and delved into state-of-the-art methods.	09/29/2023

Understand and preprocess dataset	Understand the dataset and apply data preprocessing techniques to clean the data.	10/06/2023
Evaluate and finalize algorithms	Examine various algorithm and tested on small block of dataset	10/13/2023
Implementation	Implement model and check its accuracy	10/23/2023
Comparative analysis	Conduct comparisons based on metrics derived from diverse models.	11/03/2023
Project presentation	Prepare a presentation that articulates the problem statement and outlines the proposed solution	11/06/2023
Demo preparation	Deliver the presentation	11/24/2023
Project report	Prepare a summary of the project's details and document them in a project report.	11/30/2023

Division of Work

- Read & analyze research papers - All group members
- Preprocessing of dataset - Rishabh, Herambh
- Evaluate and finalize algorithms - All group members
- Implementation - Pravin, Ninad
- Comparative analysis - Amey, Gaurav
- Project presentation - All group members
- Project demo - All group members
- Project report - All group members

References

1. R. Zhang, Q. -d. Liu, Chun-Gui, J. -X. Wei and Huiyi-Ma, "Collaborative Filtering for Recommender Systems," 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan, China, 2014, pp. 301-308, doi: 10.1109/CBD.2014.47.
2. Yehuda et al. “Matrix Factorization Techniques for Recommender Systems”. In:Pub-lished by the IEEE Computer Society.42 n.8 (Aug.2009).
3. Yuan Yao, Hanghang Tong, Guo Yan, Feng Xu, Xiang Zhang, Boleslaw K. Szymanski, Jian Lu:Dual-Regularized One-Class Collaborative Filtering.
4. Grouplens, “20M MovieLens Dataset”, Available at - https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset/data
5. Analytics Vidhya, "Movie recommendation system using KNNs", Available at - https://www.analyticsvidhya.com/blog/2020/08/recommendation-system-k-nearest-neighbors
6. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, 17(6):734–749, 2005.
7. Content-based Recommender System for Movie Website: https://www.diva-portal.org/smash/get/diva2:935353/FULLTEXT02.pdf