클러스터링 기법을 이용한 하이브리드 영화 추천 시스템

싯소포호트 [1], 펭소니 [1], 양예선 [1], 일홈존 [1], 김대영 [1], 박두순 [1*]
[1] 순천향대학교 소프트웨어융합학과

siet.sophort60@gmail.com, parkds@sch.ac.kr

Hybrid Movie Recommendation System Using Clustering Technique

Sophort Siet[1], Sony Peng[1], Yixuan Yang[1], Sadriddinov Ilkhomjon[1], DaeYoung Kim[1], Doo-Soon Park[1*]
[1]Dept. of Software Convergence, Soonchunhyang University, South Korea

요 약

This paper proposes a hybrid recommendation system (RS) model that overcomes the limitations of traditional approaches such as data sparsity, cold start, and scalability by combining collaborative filtering and context-aware techniques. The objective of this model is to enhance the accuracy of recommendations and provide personalized suggestions by leveraging the strengths of collaborative filtering and incorporating user context features to capture their preferences and behavior more effectively. The approach utilizes a novel method that combines contextual attributes with the original user-item rating matrix of CF-based algorithms. Furthermore, we integrate k-mean++ clustering to group users with similar preferences and finally recommend items that have highly rated by other users in the same cluster. The process of partitioning is the use of the rating matrix into clusters based on contextual information offers several advantages. First, it bypasses of the computations over the entire data, reducing runtime and improving scalability. Second, the partitioned clusters hold similar ratings, which can produce greater impacts on each other, leading to more accurate recommendations and providing flexibility in the clustering process.

keywords: Context-aware Recommendation, Collaborative Filtering, Kmean++ Clustering.

1. Introduction

Recently, the rapid growth of the huge amount of available information has caused people to have difficulty making decisions. To address this issue, Recommender Systems (RS) have been developed to suggest relevant items based on a user's preferences and the experiences of the users(rating, like/dislike, comment). RS plays a crucial role in various domains such as e-commerce, social networks, review sites, mobile applications, and information retrieval [1]. These systems help users to find and consume relevant contents, products, or services by filtering and recommending them based on their past behavior, preferences, and interactions with the system. In addition, RS has become an important research area in machine learning, data mining, and artificial intelligence. etc.

Collaborative Filtering (CF) techniques are commonly used technique in RS to predict a user's interest in an item based on the past rating history of other similar users or items. However, CF-based systems are only consider ratings and do not take into account other relevant contextual information, such as time of day, location, weather, and mood, which can significantly impact the recommendation quality. For example, a person's movie preference may vary depending on the company they're with, the time of day, the weather, or their location.

To address this limitation, context-aware recommendation systems (CARS) have been proposed in the last decade to integrate the current user’s context features into the existing CF framework. CARS utilized k-mean++ algorithm which can help to improve the scalability of the system by clustering users based on their preferences or behavior. This approach can reduce the computational complexity of the system and make it more efficient. CARS will provide more accurate, relevant, and personalized recommendations to the user [2].

The aim of this paper is to propose a Hybrid Recommendation System which utilizes collaborative filtering (CF) integrated with context-aware (CARS) approach. We present a CF framework that partitions the user-item-rating matrix based on various contextual factors, resulting in more accurate and personalized recommendations. Our approach also involves applying Kmean++ Clustering to remove noise outliers, reducing the computational complexity, and improving scalability without sacrificing recommendation quality.

2. Related Works

The field of Recommendation Systems are constantly evolving, and researchers are exploring new techniques to improve their performance and provide more personalized recommendations to users:

A Survey on Personality-Aware Recommendation Systems: This survey paper presents an overview of context-aware recommendation systems and discusses the challenges and opportunities in this area. It provides a taxonomy that categorizes context-aware recommendation systems based on various factors, such as the type of context, the recommendation approach, and the evaluation method [3].

Data Sparsity Issues in Collaborative Filtering Recommender Systems: This paper examines the data sparsity problem in collaborative filtering recommendation systems and proposes various techniques for improving recommendation accuracy, such as data imputation and regularization techniques [4].

Context-Aware Collaborative Filtering: This work proposed a Context-Aware recommendation system that uses collaborative filtering to make recommendations. The system considered the user's context when making recommendations by adjusting the user's preferences based on the context [5].

3. Material and Method

3.1. Data Collection

In this research, the Movielens 1M [6]. The dataset was selected as the primary data source. This dataset, obtained from the Movielens website, comprises 1,000,209 anonymous movie ratings from 6,040 users who joined the website in the year 2000. The ratings cover approximately 3,900 unique movies. The dataset is structured into three distinct CSV files which is a brief description of important attributes such as: user id, movie id, movie title, rating, timestamp, gender, age, occupation, zip code, title, and genres.

3.2. System Architecture

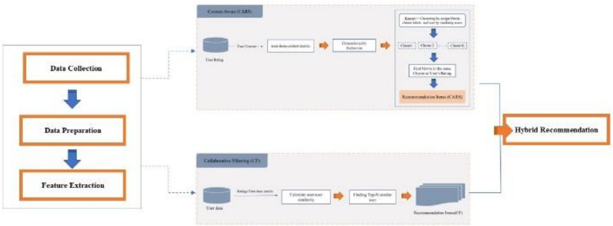

Fig.1. System Architecture of our recommendation system

3.3. Data Preparation

To construct a movie recommendation system, the first step is to gather relevant data on users, movies, and user-item interactions. This data is then subjected to a sequence of cleaning procedures, including removing duplicates, handling missing values, and dealing with outliers, before being transformed and normalized for analysis. Data preparation is crucial for ensuring high-quality data and accurate, reliable, and scalable analyses, predictions, and recommendations. It also allows for the identification of meaningful patterns and trends that are useful in developing collaborative and context-aware recommendation models [3].

3.4. Feature Extraction

Feature extraction is a critical aspect of constructing an effective recommendation system which involves identifying relevant user and item attributes, as well as contextual information, such as time of day or genre. One approach to feature extraction is creating a sparse matrix that represents user-item-context interactions in a collaborative and context-aware using binary encoding or one-hot encoding to transform information into a machine-readable format. Additionally, incorporating features such as average rating, number of ratings, and the time of day or weekday can provide valuable insights into user behavior and preferences, aiding in the grouping or filtering of movies based on various criteria [3].

3.5. Collaborative Filtering

Collaborative Filtering (CF) is a widely used recommendation technique that aims to identify similarities between users or items based on their ratings. CF algorithms can be broadly categorized into classes: Memory-based and Model-based approaches [2].

In this section, we utilized Memory-based algorithms, which rely on identifying the top-N most similar users (i.e., neighbors) to the active user and using a weighted sum of their ratings to predict missing ratings for the active user. To achieve this, we take a user ID and the number of similar users to be returned as input and computing the pairwise similarity scores between the given user and all other users based on their ratings. The Pearson correlation coefficient measures the linear correlation between two sets of ratings. The formula for computing the Pearson correlation coefficient between two users i and j is as follows [2]:

$$\text{Similarity(i, j)} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r_i})(r_{u,j} - \bar{r_j})}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r_i})^2 \sum_{u \in U}(r_{u,j} - \bar{r_j})^2}}$$

Where, $r_{u,i}$ is the rating of user u on item i
$r_{u,j}$ is the rating of user u on item j
U is the set of all users who have rated both items i and j
$\bar{r_i}$ is the average rating of item i by all users in U
$\bar{r_j}$ is the average rating of item j by all users in U

Collaborative Filtering (CF) will retrieve the movies that the top similar users have rated highly and sort them based on their average ratings. It then removes the movies that the given user has already rated from the recommended movies list to avoid recommending movies that the user has already watched. Finally, the CF will return the top-N recommended movies based on their average ratings.

This approach is effective in providing personalized recommendations to users with similar movie preferences and has been widely used in recommender systems. However, it suffers from the cold start problem, where new users with limited ratings history cannot be effectively recommended movies[3].

3.6. Context-Aware Clustering Algorithm

The Context-Aware Clustering Algorithm is a popular method used in recommender systems to improve the accuracy of recommendations by incorporating contextual information. This approach involves three key steps: Contextual Pre-filtering, Contextual Post-filtering, and Contextual Modeling.

The first step is to collect ratings from users for a set of items and consider the context in which the user provided the rating. This context could include factors such as the time of day, day of the week, or the user's occupation. The system then filters out items that are unlikely to be of interest to the user based on their context [3,5].

Next, a user-item-context matrix is created where each user-item pair is represented as a vector, and each vector component represents the strength of the user's preference for that item in that context. However, this matrix can have a high number of dimensions, making it difficult to cluster accurately. To address this, Non-negative Matrix Factorization (NMF) can be applied to reduce the dimensionality of the data and identify patterns in the matrix. NMF factorizes the user-item-context matrix into two non-negative matrices, W and H, where W represents the reduced-dimensional representation of the matrix and H represents the coefficients that describe how each user-item pair can be reconstructed from the latent factors.

After applying NMF, the k-means++ algorithm is then applied to cluster the items based on their similarities in genre or other relevant factors. The algorithm selects initial cluster centroids that are well-spaced and representative of the data distribution. Each item is then assigned to the nearest centroid, and new centroids are calculated based on the mean of the items in each cluster. This process is repeated until the centroids converge to stable positions.

Finally, each cluster is assigned a label to identify it, and this label is used to find items in the same cluster as the user's ratings. Clustering helps to reduce the complexity of similarity computations by confining them within the cluster and generating more accurate recommendations by grouping ratings with similar contexts.

Hybrid Recommendation Systems (CARS, CF) by confining them within the cluster, while grouping ratings with similar contexts enhances the accuracy of recommendations.

4. Conclusion

In conclusion, our proposed hybrid movie recommendation system effectively combines collaborative and context-aware approaches, leveraging contextual information to improve recommendation system accuracy and runtime efficiency. By incorporating clustering techniques which be able to address traditional problems such as cold-start and data sparsity. In Addition, this model prioritizes contextual information such as location, time, and social interactions to provide more personalized and relevant recommendations. In the Future work, we will develop and optimize it for real-world applications and utilized various techniques, such as deep learning and transfer learning to better handle the complexities of large and diverse datasets and incorporate a wider range of context information.

참고문헌

[1] Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R.: Recommendation systems: Algorithms, challenges, metrics, and business opportunities. applied sciences, 10(21), 7748. (2020).

[2] Kannout, E.: Context clustering-based recommender systems. In 2020 15th Conference on Computer Science and Information Systems (FedCSIS) (pp. 85-91). IEEE. (2020, September)

[3] Dhelim, S., Aung, N., Bouras, M. A., Ning, H., & Cambria, E.: A survey on personality-aware recommendation systems. Artificial Intelligence Review, 1-46. (2022)

[4] Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H.: Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. Expert Systems with Applications, 149, 113248. (2020)

[5] Zheng, Y.: Context-aware collaborative filtering using context similarity: an empirical comparison. Information, 13(1), 42. (2022)

[6] Movielens. GroupLens. (2021, December 8). Retrieved January 31, 2023, from https://grouplens.org/datasets/movielen