A Battle Royale of Machine Learning Models for Recommender Systems

CSE 5713 - Data Mining

Department of Computer Science and Engineering
University of Connecticut

Presented by Shravan Kumar Gajula, Razzakuddin Mohammed, Nithin Magatala

Agenda

- Introduction
- Data Mining Techniques
- Related works
- Dataset
- Content Based Recommendation System
- Collaborative Filtering Recommendation System
- Hybrid Recommendation System
- Applications of Data Mining
- Conclusion

Introduction

- In today's data-driven world, businesses and organizations are increasingly relying on personalized recommendations to enhance customer experience, increase engagement, and boost revenue. Recommendation systems, powered by data mining techniques, analyze user behavior and preferences to provide tailored suggestions for products, services, or content.
- This project aims to explore the fundamentals of recommendation systems in data mining, and explain about recommendation engine using content based, collaborative filtering techniques and hybrid RS.
- The project will cover various aspects of recommendation systems, including data preprocessing, feature engineering, model selection, and evaluation.

Data Mining Techniques

- Matrix Factorization: This technique analyzes user-movie interaction and decomposes the user-movie rating matrix into smaller matrices to identify user preferences and recommend movies based on those preferences.
- Clustering: It is a technique that groups movies into clusters based on their attributes such as genre, actors, and directors. It then recommends movies from the same cluster to users who have watched and liked a movie from that cluster.
- KNN: K-Nearest Neighbors is a simple data mining algorithm which we have used in this movie recommendation systems. It works by finding the K most similar movies to a target movie based on distance metrics such as Euclidean distance or cosine similarity.

Related works

Recommender Systems by Charu C. Aggarwal

- This book describes a comprehensive introduction to the field of recommendation systems, covering both traditional and modern techniques.

R. Vijayarani and V. M. Pandit, "Hybrid Recommender System Using Item Clustering and User-Based Collaborative Filtering,"

- The paper proposes a hybrid recommender system that utilizes item clustering and user-based collaborative filtering to improve recommendation accuracy.

S. Amara and R. R. Subramanian, "Collaborating personalized recommender system and content-based recommender system using TextCorpus,"

- The paper presents a hybrid recommender system that combines the collaborative filtering and content-based approaches using a text corpus. The proposed system showed improved performance in recommending items to users compared to the individual approaches.

Dataset

- For the dataset preparation, we have taken the data regarding multiple social media platforms happening in the world for that we need to import the dataset/s that we have gathered for the Data Mining project at hand.
- Importing the dataset is one of the important steps in data preprocessing in Data Mining. However, before we can import the dataset/s, we have set the current directory as the working directory.
- Dataset contains the ratings provided by customers on a five-star scale, with whole numbers ranging from 1 to 5. The MovieIDs range from 1 to 17770, and each ID corresponds to a specific movie in the dataset. The CustomerIDs range from 1 to 2649429.

Content based Recommendation System

- Content-based filtering is a type of recommendation system that recommends items to users based on the features or characteristics of the items themselves.
- Uses a machine learning algorithm to induce a profile of the users preferences from examples based on a featural description of content.
- Content-based filtering starts by analyzing the attributes of the items, such as metadata, text, or keywords, to create a profile that represents the item's characteristics.
- The system then measures the similarity between the user profile and the item profiles using a similarity metric, such as cosine similarity or Jaccard similarity.
- To overcome the limitations of content-based filtering, it is often combined with other approaches, such as collaborative filtering or hybrid filtering.

Results for CBF

```
#Sort movies based on score calculated above
q_movies = q_movies.sort_values('score', ascending=False)

#Print the top 15 movies
q_movies[['title', 'vote_count', 'vote_average', 'score']].head(10)
```

	title	vote_count	vote_average	score
1881	The Shawshank Redemption	8205	8.5	8.059258
662	Fight Club	9413	8.3	7.939256
65	The Dark Knight	12002	8.2	7.920020
3232	Pulp Fiction	8428	8.3	7.904645
96	Inception	13752	8.1	7.863239
3337	The Godfather	5893	8.4	7.851236
95	Interstellar	10867	8.1	7.809479
809	Forrest Gump	7927	8.2	7.803188
329	The Lord of the Rings: The Return of the King	8064	8.1	7.727243
1990	The Empire Strikes Back	5879	8.2	7.697884

- Sort movies based on score calculated above then we will get the above output as suggestions
- Selects the top 10 most similar movies from the sorted list of similarity scores.

```
get_recommendations('The Dark Knight Rises')
```

```
65                                The Dark Knight
299                                Batman Forever
428                                Batman Returns
1359                                       Batman
3854    Batman: The Dark Knight Returns, Part 2
119                                Batman Begins
2507                                   Slow Burn
9           Batman v Superman: Dawn of Justice
1181                                       JFK
210                                Batman & Robin
Name: title, dtype: object
```

```
get_recommendations('The Avengers')
```

```
7                 Avengers: Age of Ultron
3144                              Plastic
1715                              Timecop
4124                   This Thing of Ours
3311                Thank You for Smoking
3033                         The Corruptor
588      Wall Street: Money Never Sleeps
2136          Team America: World Police
1468                          The Fountain
1286                           Snowpiercer
Name: title, dtype: object
```

Advantages of Content based Filtering RS

- Content-based filtering provides personalized recommendations based on the user's preferences and characteristics of the items.
- Content-based filtering does not require user feedback or ratings to provide recommendations, which makes it useful for new users or items with limited feedback.
- The recommendations provided by content-based filtering are easy to explain and understand, as they are based on the features and characteristics of the items.
- Content-based filtering can be scaled to handle large datasets and high traffic, as the computation of the item profiles can be done offline.

Disadvantages of Content based Filtering RS

- Content-based filtering is not effective for new users or items with limited data, as it relies on the user's past interactions or the item's attributes to provide recommendations.
- Content-based filtering may suffer from overfitting, as it may recommend items that are too similar to the users previous interactions and do not generalize well to new or different contexts.
- Content-based filtering does not consider social influence or the opinions of other users in the recommendation process, which may limit the discovery of popular or trending items.

Collaborative Filtering Recommendation System

- Collaborative filtering recommender systems make personalized recommendations by using the preferences of similar users or items to predict ratings for a particular user.
- Collaborative filtering can be implemented in two ways: user-based and item-based, where the former identifies similar users and recommends items based on their preferences, while the latter identifies similar items and recommends them based on their attributes.
- Collaborative filtering can suffer from the cold-start problem, where new users or items have limited data for prediction, and also from the sparsity problem, where data is often incomplete due to the vast item space and low user-item interaction.

Collaborative Filtering Recommendation System

- The code loads movie rating data using pandas and converts it to a dataset using Surprise library's Reader and Dataset modules.
- The SVD algorithm is applied to the dataset using cross-validation to evaluate its performance using RMSE and MAE measures.
- Three recommender algorithms (SVD, user-based KNN, and item-based KNN) are evaluated using cross-validation with RMSE and MAE measures.
- The SVD algorithm is fit to the training data and used to predict the rating of a particular user for a particular movie.
- The Surprise library is used throughout the code for loading data, defining algorithms, and evaluating their performance.

Results

```
SVD - Mean RMSE: 1.008, Mean MAE: 0.778
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
KNNBasic - Mean RMSE: 0.967, Mean MAE: 0.744
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
KNNBasic - Mean RMSE: 0.936, Mean MAE: 0.722
```

```
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
Computing the msd similarity matrix...
Done computing similarity matrix.
```

Results

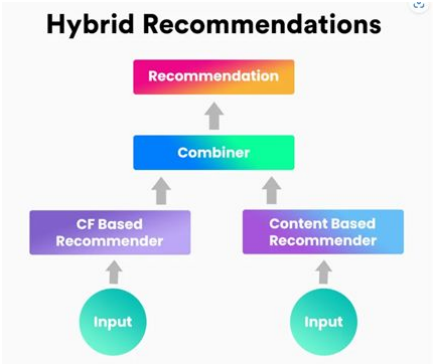
+ Code + Text

```
[18] svd.predict(1, 302, 3)
```

```
Prediction(uid=1, iid=302, r_ui=3, est=2.030815072584304, details={'was_impossible': False})
```

Hybrid Recommendation System

- Hybrid Recommender Systems combine two or more recommendation algorithms to improve the quality of recommendations. For example, a hybrid recommender system may combine Collaborative Filtering and Content-Based Filtering.
- We brought together ideas from content and collaborative filtering to build an engine that gave movie suggestions to a particular user based on the estimated ratings that it had internally calculated for that user.



Results

```
hybrid(1, 'Avatar')
```

	title	vote_count	vote_average	year	id	est
1011	The Terminator	4208.0	7.4	1984	218	3.083605
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	2.947712
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	2.935140
1621	Darby O'Gill and the Little People	35.0	6.7	1959	18887	2.899612
974	Aliens	3282.0	7.7	1986	679	2.869033
8401	Star Trek Into Darkness	4479.0	7.4	2013	54138	2.806536
2014	Fantastic Planet	140.0	7.6	1973	16306	2.789457
922	The Abyss	822.0	7.1	1989	2756	2.774770
4966	Hercules in New York	63.0	3.7	1969	5227	2.703766
4017	Hawk the Slayer	13.0	4.5	1980	25628	2.680591

```
hybrid(500, 'Avatar')
```

	title	vote_count	vote_average	year	id	est
8401	Star Trek Into Darkness	4479.0	7.4	2013	54138	3.238226
974	Aliens	3282.0	7.7	1986	679	3.203066
7265	Dragonball Evolution	475.0	2.9	2009	14164	3.195070
831	Escape to Witch Mountain	60.0	6.5	1975	14821	3.149360
1668	Return from Witch Mountain	38.0	5.6	1978	14822	3.138147
1376	Titanic	7770.0	7.5	1997	597	3.110945
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	3.067221
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	3.043710
1011	The Terminator	4208.0	7.4	1984	218	3.040908
2014	Fantastic Planet	140.0	7.6	1973	16306	3.018178

Applications of Data Mining

Marketing

- Analysis of consumer behavior
- Advertising campaigns
- Targeted mailings
- Segmentation of customers, stores or products

Finance

- Creditworthiness of clients
- Performance analysis of finance investments
- Fraud detection

Manufacturing

- Optimization of resources
- Optimization of manufacturing processes
- Product design based on customers requirements

Health Care

- Discovering patterns in X-ray images
- Analyzing side effects of drugs
- Effectiveness of treatments

Conclusion

- This project explored various machine learning models for building recommender systems, including collaborative filtering, content-based filtering, and hybrid approaches. The results showed that each model has its own strengths and weaknesses and that the performance of the models is highly dependent on the dataset and evaluation metrics used.
- Hybrid models, which combine both collaborative and content-based filtering, generally performed better than the individual approaches, showing the benefits of leveraging multiple techniques to build more accurate and effective recommender systems.
- Recommender systems have numerous practical applications in industries such as e-commerce, entertainment, and social media, and the insights from this project can help inform the development of more effective recommendation systems in various domains.

References

- https://www.amazon.com/Recommender-Systems-Textbook-Charu-Aggarwal/dp/3319296574
- https://ieeexplore.ieee.org/document/9074360
- https://www.slideshare.net/SalilNavgire/data-mining-and-recommendation-systems
- https://www.researchgate.net/publication/293646845_A_Hybrid_Approach_for_Movie_Recommendation_via_Tags_and_Ratings
- https://github.com/spChalk/Movie-Recommendation-System

Thank you