Movie Recommendation System

Dhruv Kaushik MT18037 | **Gurpreet Singh** MT18098 | **Wrik Bhadra** MT18027

IIIT Delhi

Problem Statement

Recommender systems have become ubiquitous in our lives. Be it e-commerce websites or social media platforms, recommender systems add the "what-next" factor to it. Due to the advances in recommender systems, users constantly expect good recommendations. They have a low threshold for services that are not able to make appropriate suggestions. This has led to a high emphasis by tech companies on improving their recommendation systems. However, the problem is more complex than it seems.

In this project, we aim to help users instantly discover movies to their liking, regardless of how distinct their tastes may be.

Motivation

Given that huge amount of movies are available all over the world, it is challenging for a user to find the appropriate movies suitable to his/her tastes. Different users like different movies or actors. It is important to find a method of filtering irrelevant movies and/or find a set of relevant movies. Movie recommendation does exactly this. Such a system has lots of implications and is inspired by the success of recommendation systems in different domains viz. books, news articles etc.

In regard to this, we recount an experience with a friend of ours when he had approached us to get a movie recommendation and we had casually asked him to look up IMDb.

Tools and Technologies

- Language: Python
- Libraries
  - NLTK
  - Scikit-learn
  - GenSim

Evaluation Metric

The system will be evaluated on the RMSE (Root Mean Square Error) score as given below:

$$RMSE = \sqrt{\sum_{i=1}^{N} (pred_i - true_i)^2 / N}$$

Literature Review

Content-based filtering makes recommendation based on similarity in item features. Popular techniques in content-based filtering include the term-frequency/inverse-document-frequency (tf-idf) weighting technique in information retrieval [1][2] and word2vec in natural language processing. An extension of word2vec, called doc2vec [3] is used to extract information contained in the context of movie descriptions. Content-based filtering works well when there hasn't been enough users or when the contents haven't been rated. Collaborative filtering recommends items that similar users like, and avoids the need to collect data on each item by utilizing the underlying structure of users' preference. One major approach in collaborative filtering is neighborhood model [4]. The neighborhood model recommends the closest items or the closest user's top rated items.

Dataset Description

Dataset is provided at Kaggle as The Movie Dataset at TheMoviesDataset.

The dataset contains 45,000 movies listed in the Full Movie Lens Dataset. All the movies were released before July 2017. Each data point has features like cast, genre, revenue, language, release date, etc.

The whole dataset is rated by 270,000 users and total ratings done by these users were 26 million. Ratings are on scale 1-5.

Methodology

The methodology we plan to follow is given below:

Method 1: Our approach is to rate the movie based on a cosine similarity between the tf-IDF vector of test data point against the tf-IDF vector of the known ground truth. More similarity implies a similar rating.

Method 2: Use doc2vec to capture the similarity effect of synonyms and understand the context of words. Doc2vec would be used to generate unique feature vector representing an entire document (movie).

Method 3: Predict user's rating of movie i using a weighted sum of movie i's rating from the k nearest users based on their ratings' similarity score.

Timeline

Mid Evaluation
- Data pre-processing
- Method 1 implementation

Final Evaluation
- Method 2 and Method 3 implementation
- Analysis of all the methods

References

[1] A. Tuzhilin and G. Adomavicius. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge & Data Engineering, vol. 17, no.6, pp. 734-749, 2005.

[2] G. Salton. Automatic Text Processing. Addison-Wesley (1989)

[3] https://radimrehurek.com/gensim/models/doc2vec.html

[4] D. Billsus and M. J. Pazzani. Learning Collaborative Information Filters. Proceedings of the International Conference on Machine Learning. 1998.