Movie Recommender System

Zhengqi Dong, Yuntian He
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH 43210
{dong.760, he.1773}@osu.edu

1 Problem Statement

"What movie should I watch tonight?" Have you ever heard your friends or family members ask you this question? As for me — yes, and more than once. The need to build robust movie recommendation systems is extremely important given the huge demand for personalized content of modern consumers. An example of recommendation system is such as this:

- User A watches Game of Thrones and Breaking Bad.
- User B does search on Game of Thrones, then the system suggests Breaking Bad from data collected about user A.

Here is a formal formulation of **Movie Recommendation Problem**. Given a set of users $\mathcal{U}$, movies $\mathcal{M}$, rating score field $\mathcal{S}$, and a rating record set $\mathcal{R} \subseteq \mathcal{U} \times \mathcal{M} \times \mathcal{S}$, we aim to provide user $u \in \mathcal{U}$ with recommendations on movies in $\mathcal{M}$ that he has never watched. In general, there are two kinds of queries for this problem:

- **Rating prediction:** The goal is to learn a function $f_s : \mathcal{U} \times \mathcal{M} \to \mathcal{S}$ to predict the rating that user u would give to an unseen movie m.
- **Top-k recommendation:** The goal is to learn a function $f_t : \mathcal{U} \to \mathcal{M}^k$ to predict k unseen movies in which user u might be interested.

Note that the second query can be answered from the first one by simply sorting all movies with their predicted ratings and confidence scores. Since it is hard to find the ground truth data of top-k recommendation, in this project we will focus on the first query for the problem.

2 Proposed approach

Our goal is to propose a embedding-based recommendation system where both content-based and collaborative filtering are considered, and the architecture is shown in Figure 1.

First we will build a heterogeneous network with user rating records and metadata of movies, more specifically, a network with nodes including not limited to users, movies, and genres. We will perform over the network some meta-path-based random walks which, for example, contains movies seen by the same user, or movies of the same genre, so that these movies will have close embedding vectors.

In addition, we also plan to leverage other metadata of movies for embedding them. Intuitively, movies having similar stories may interest the same user. We can measure the similarity of two movies based on the embeddings of their summary.

Finally, we will train a classifier with embeddings learned from the modules mentioned above as input and rating data as output. The classifier can be logistic regression, deep neural network, and so on.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

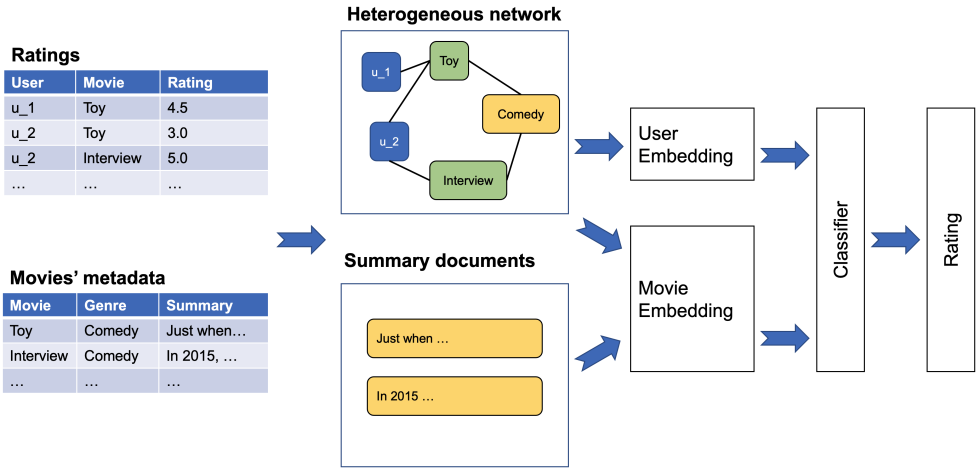

Figure 1: System architecture

3 Experiment setup

3.1 Datasets

We currently have three alternative datasets for building this project. Table 1 shows the characteristics of each dataset.

The Movies Dataset on Kaggle This dataset[1] contains 26 million ratings and 750,000 tag applications from 270,000 users on all the 45,000 movies in the dataset. Besides, it has metadata of each movie including cast, crew, language, overview, etc.

MovieLens Dataset This dataset[2] is collected by GroupLens Research from `movielens.org`, containing 25 million movie ratings applied to 62,000 movies by 162,000 users. Each rating record has the following format: *(userId, movieId, rating, timestamp)*, and each movie is represented as *(movieId, title, genres)*.

Netflix-Movie Recommendation Dataset This dataset[3] comes directly from Netflix, collecting ratings on 17,770 movies from 480,000 users.

Table 1: Sample table title

Name	# ratings	# users	# movies
Movies	26M	270K	45K
MovieLens	25M	162K	62K
Netflix	20M	480K	18K

3.2 Baselines

In the experiment, we plan to evaluate our proposed methods including:

- using embeddings from heterogeneous network embedding only

[1] https://www.kaggle.com/rounakbanik/the-movies-dataset
[2] https://grouplens.org/datasets/movielens/
[3] https://www.kaggle.com/netflix-inc/netflix-prize-data

- using embeddings from summary only
- using embeddings from both modules

We are going to compare our proposed methods with one or both of the following methods:

- SVD-based collaborative filtering
- Skip-gram model: movie2vec[4] is implemented based on item2vec[1], generating embeddings of movies with tags considered only.

3.3 Metrics

We plan to evaluate the quality of each method using the following metrics:

- Macro-F1 score
- Micro-F1 score
- RMSE of predictions

4 Related work

4.1 Classic recommendation techniques

Content-Based Filtering The Content-Based Recommender solely relies on the similarity of the items being recommended. For instance, if you like an item, then you will also like a "similar" item. It generally works well when it's easy to determine the context/properties of each item.

Collaborative Recommendations The collaborative recommendations are based on the similarity between the preference of users and not the content of the product, so the recommended items will depend on what other similar users liked. For instance, if user A likes movies 1, 2, 3 and user B likes movies 2,3,4, then they have similar interests and A should like movie 4 and B should like movie 1.

4.2 Heterogeneous network embedding

Traditional network embedding methods like DeepWalk[6] and node2vec[4] are designed networks where nodes are of the same type, such as users in a social network. However, sometimes we will put objects of different types in a network, which comprise an heterogeneous network. The challenge of embedding a heterogenous network comes from the capturing the relationship of objects of different types. Metapath2vec[3] perform network embedding by feeding a heterogeneous skip-gram model with meta-path-based random walks, enabling the modeling of structural and semantic correlations in the network simultaneously.

4.3 Document embedding

In recent years, extensive research has been conducted on learning representations of textual information like words and documents. Doc2vec[5] is a representative work for generating a fixed-length representation for a variable-length pieces of texts like a sentence or a document. Tweet2vec[2] aims to embed a tweet and predict its hashtag by leveraging Bi-GRU encoder and softmax function.

References

[1] Oren Barkan and Noam Koenigstein. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2016.

[2] Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W Cohen. Tweet2vec: Character-based distributed representations for social media. *arXiv preprint arXiv:1605.03481*, 2016.

[4] https://github.com/lujiaying/MovieTaster-Open

[3] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[4] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[5] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[6] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.