Content-Based Personalized Recommender System Using Entity Embeddings

Xavier Thomas
Manipal Institute of Technology, Udupi, Karnataka, India
xavier.thomas1@learner.manipal.edu

Abstract

Recommender systems are a class of machine learning algorithms that provide relevant recommendations to a user based on the user's interaction with similar items or based on the content of the item. In settings where the content of the item is to be preserved, a content-based approach would be beneficial. This paper aims to highlight the advantages of the content-based approach through learned embeddings and leveraging these advantages to provide better and personalized movie recommendations based on user preferences to various movie features such as genre and keyword tags.

Introduction

There are mainly two approaches to building a recommender system – Collaborative Filtering (CF) and Content-based (CB) recommending. CF systems work by collecting user feedback in the form of ratings for items in a given domain and exploit similarities and differences among profiles of several users in determining how to recommend an item. On the other hand, content-based methods provide recommendations by comparing representations of the content contained in an item to representations of content that interests the user. (Melville, Mooney and Nagarajan 2002)

The collaborative method creates a N x M matrix, with N users and M items. Wherein each user provides a rating for an item and this rating is stored as the element in the matrix. This matrix is utilized to provide similarity scores between users and recommends items that are highly rated by similar users. In the context of a movie recommendation system, which the paper describes. The collaborative method can lead to data sparsity, as a majority of the movies are unrated by the user. This method also undergoes a cold-start issue which causes newly added movies to have less significance in recommendations due to the lower number of ratings.

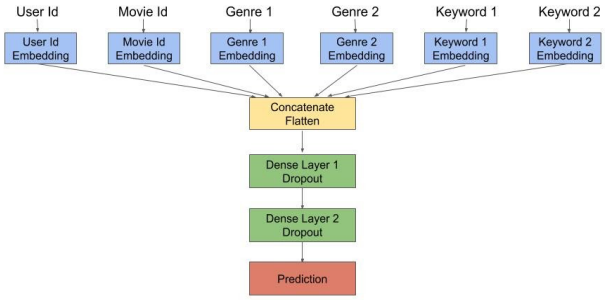


Figure 1: The proposed model takes the movie features as input, converts it into an embedding, and learns to predict the rating.

A content-based method does not suffer the problems stated above and can be used to build personalized recommender systems. This paper uses the MovieLens dataset to build this system.

Methodology

The main idea behind building a content-based recommender system is to recommend similar items based on the content of the item. This paper uses the features genre and keywords attributed to each movie from the MovieLens Dataset. Today's Recommendation systems highly rely on similarity-based recommendations between users, which could fail to adapt to a user's unique taste. Learning an embedding space help to differentiate movies based on their content. This approach is utilized in finding movie and user embeddings so that similar movies and users are grouped respectively. Furthermore, this method is aimed to make recommender systems personalized by capturing the user's unique taste in the embedding space.

User Id	14	289	2
Movie Id	570	112	1127
Genres	Drama, Crime	Horror, Crime	Drama, Romance
Genre 1	5	12	5
Genre 2	11	11	21
Keywords	Mafia, 1970s	Supernatural, Scary	Based on book, Love
Keyword 1	22	27	25
Keyword 2	11	44	51
Rating	4.5	5	2.5

Table 1: Categorical variables Genre and Keywords are mapped to respective integer values.

The proposed model takes as inputs the User Id, Movie Id, Genre 1, Genre 2, Keyword 1, and Keyword 2. Where Genres and Keywords of the movie are categorical variables that are mapped to integer values as shown in the example in Table 1. The model learns to predict the Rating by converting the inputs into Embeddings and uses dense neural network architectures to output a predicted Rating.

In a more concise view, the model aims to learn a function

$$y = f(x_1, x_2, ..., x_n) \quad (1)$$

Given the inputs $(x_1, x_2, ..., x_n)$ the model generates an output prediction of the movie rating. Where the inputs are first converted to Embeddings by mapping each input value to a vector as

$$e_i: x_i \mapsto \mathbf{x}_i \quad (2)$$

Entity Embeddings The model aims to map categorical variables in a function approximation problem into Euclidean spaces, which are the entity embeddings of the categorical variables. The mapping is learned by a neural network during the standard supervised training process (Guo and Berkhahn 2016). Entity Embeddings are advantageous in the setting of a movie recommendation system, as it performs well on datasets with lots of high cardinality features such as the User Id and Movie Id in the dataset, and more importantly, it reveals the intrinsic properties of the categorical variables which helps in the content-based clustering of movies and users. The embeddings are utilized to make recommendations personalized, by aggregating the movie embeddings of the movies a user rated highly, it is possible to extract the user's interests.

Empirical Experiments and Results

The model is tested on a subset of the MovieLens dataset. The dataset contained 77167 entries, 5071 unique movies, and 671 unique users. The observations obtained are as follows: 1) A content-based approach performs well in the setting of a movie recommendation system. Table 2 shows the performance. 2) The embeddings learned was able to express the intrinsic properties of movies and users. 3) The system can be made personalized by aggregating embeddings of the user's top movies which captures the user's unique taste. Then searching the embedding space to return movies with features similar to the user's interest. An example is shown in Table 3, the model captures the user's interest in Romance and Comedy to recommend similar movies.

MSE	RMSE	MAE
0.387	0.622	0.45

Table 2: The evaluation metrics calculated by comparing the predicted output to the actual output are shown.

User Id: 288	
Top Rated Movies	Recommended Movies
The Fox and the Hound, 10 Things I Hate About You, 500 Days of Summer, 13 going on 30. An American Tail: Fievel Goes West	The Mirror Has Two Faces, Sleepless in Seattle, Syriana, The Notebook

Table 3: Recommendations based on the user profile.

Conclusion

This abstract proposes a recommender system using Entity Embeddings to be used in content-based environments like that of movie recommendation. This approach preserves the content of the item and can make recommendations personalized.

References

Cheng Guo and Felix Berkhahn 2016. Entity embeddings of categorical variables.

Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. In *ACM Trans. Interact. Intell. Syst.5, 4, Article 19 (Dec. 2015), 19 pages.*

P. Melville, R.J. Mooney, and R. Nagarajan 2002. Content-Boosted Collaborative Filtering for Improved Recommendations. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI – 2002),* 187-192, Edmonton, Canada, July 2002.

Supplementary Material for *Content-Based Personalized Recommender System Using Entity Embeddings*

Xavier Thomas
Manipal Institute of Technology, Udupi, Karnataka, India
xavier.thomas1@learner.manipal.edu

Abstract

Recommender systems are a class of machine learning algorithms that provide relevant recommendations to a user based on the user's interaction with similar items or based on the content of the item. This paper aims to highlight the advantages of the Content-based approach through learned embeddings and leveraging these advantages to provide better movie recommendations based on user preferences to various movie features such as genre and keyword tags of a movie.

A. DATASET

A subset (composed of metadata and ratings datafile) of the MovieLens 100k dataset is used to build the below dataset.

userId	MovieId	title	genres	keywords	rating
2	10	GoldenEye	['Adventure', 'Action', 'Thriller']	['cuba', 'falsely accused', 'secret identity', 'computer virus', 'secret base', 'secret intelligence service', 'kgb', 'satellite', 'special car', 'cossack', 'electromagnetic pulse', 'time bomb', 'st. petersburg russia', 'ejection seat', 'red army']	4.0
4	10	GoldenEye	['Adventure', 'Action', 'Thriller']	['cuba', 'falsely accused', 'secret identity', 'computer virus', 'secret base', 'secret intelligence service', 'kgb', 'satellite', 'special car', 'cossack', 'electromagnetic pulse', 'time bomb', 'st. petersburg russia', 'ejection seat', 'red army']	4.0
7	10	GoldenEye	['Adventure', 'Action', 'Thriller']	['cuba', 'falsely accused', 'secret identity', 'computer virus', 'secret base', 'secret intelligence service', 'kgb', 'satellite', 'special car', 'cossack', 'electromagnetic pulse', 'time bomb', 'st. petersburg russia', 'ejection seat', 'red army']	3.0

The Dataset is converted into the following,

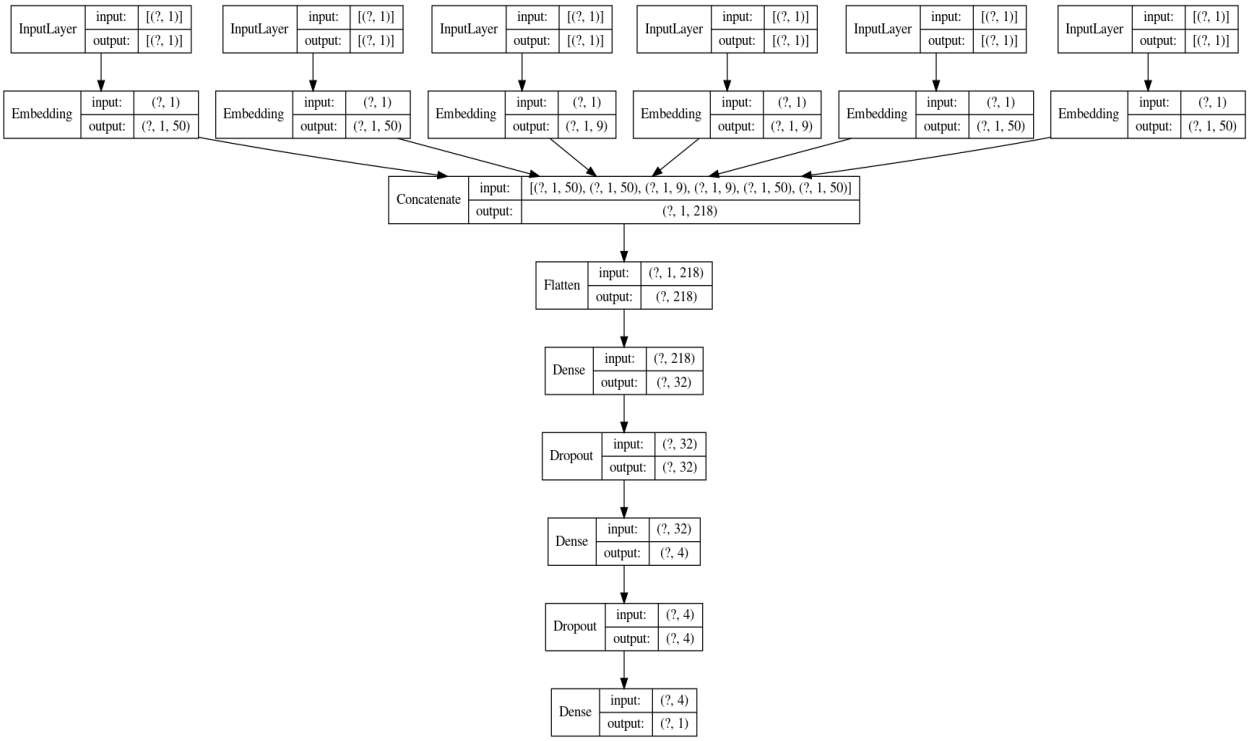
	userId	id	rating	title	genres	keywords	genre1	genre2	key1	key2
13256	3	570	5.0	The Godfather	[Drama, Crime]	['italy', 'love at first sight', 'loss of father', 'patriarch', 'organized crime', 'mafia', 'lawyer', 'italian american', 'crime family', 'rise to power', 'mob boss', '1940s']	5	4	166	150
13257	4	570	2.5	The Godfather	[Drama, Crime]	['italy', 'love at first sight', 'loss of father', 'patriarch', 'organized crime', 'mafia', 'lawyer', 'italian american', 'crime family', 'rise to power', 'mob boss', '1940s']	5	4	166	150
13258	7	570	5.0	The Godfather	[Drama, Crime]	['italy', 'love at first sight', 'loss of father', 'patriarch', 'organized crime', 'mafia', 'lawyer', 'italian american', 'crime family', 'rise to power', 'mob boss', '1940s']	5	4	166	150

Where the inputs to the model are userId, id (movie id), genre1, genre2, key1, key2.

The genres column is mapped to corresponding integer values.
Example: Drama -> 5, Crime -> 4
Similarly, from the keywords column, the most occurring keywords are found from the keywords list which are then mapped to integer values at key1 and key2.

B. THE MODEL

The Detailed figure of the model is shown below. The inputs of the model are userId, id (movie id), genre1, genre2, key1, key2 in this order.



- The Embedding size of each input is taken to be min((no. of unique values)//2, 50), which resulted in the following sizes: userId, id, key1, key2 – 50 and genre1, genre2 – 9
- The first two dense layers use a RELU activation function, the output dense layer uses a custom activation function described as sigmoid(x) * 6
- The Dropout Layer is introduced to prevent the model from overfitting.
- The output of the model is a predicted rating.

C. WORKING

An **Embedding** is a mapping of a discrete categorical variable to a vector of continuous values, i.e. they are learned continuous vector representation of discrete values. The major advantages of using Embeddings are:

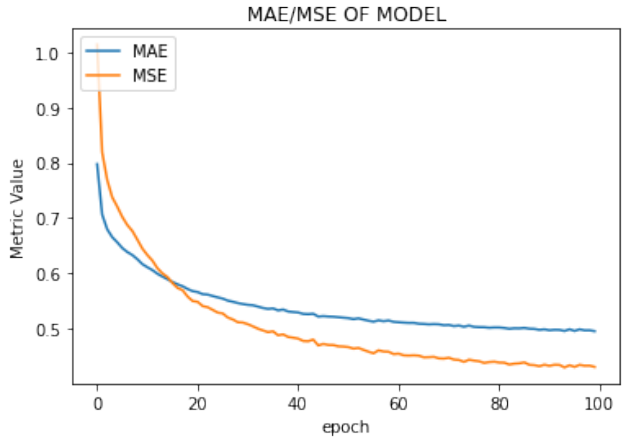
- Converts high dimensional data to low dimensional data representation, which facilitates easier training.
- Similar items are grouped together

In the context of movie recommendations, Embeddings are useful for:

1. Finding similar items in the Embedding space – To find similar movies and users
2. Used as input to a machine learning model for a supervised task – To predict the movie rating
3. For capturing intrinsic properties of the item – To capture user's movie interests through embedding aggregation.

The Embeddings form the weights of the network which are adjusted to minimize the loss on the task, which is to minimize the difference between actual rating and predicted rating. The reasoning behind the model is that through this process of predicting ratings, similar items i.e. similar movies and users come closer in the Embedding space. Thus, **the model can predict the movie rating and express the intrinsic properties of these items through the same learning procedure**.

D. PERFORMANCE



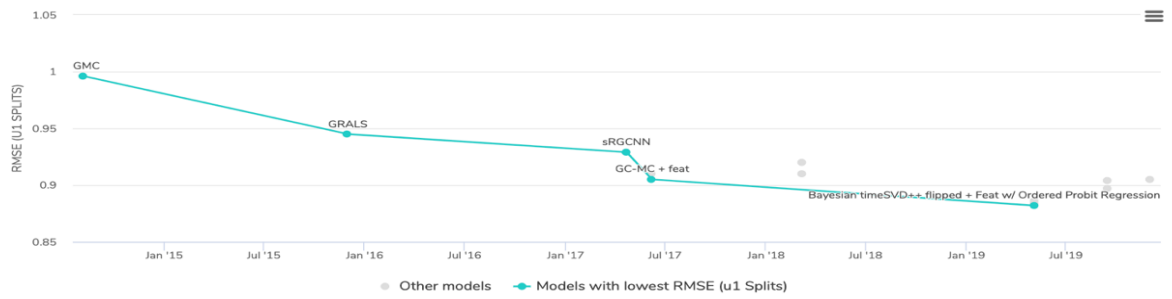
MSE	RMSE	MAE
0.387	0.622	0.45

The model is trained on a subset of the MovieLens-100k dataset. It contains :
Number of unique movies: 5072
Number of unique users: 671
Total Number of entries in the Data Frame: 77167

Entries are dropped from the original dataset if the data in one of the input columns is null. A majority of the movies were dropped because genre information was not available.

The observed RMSE is found to be significantly better than previous approaches on the MovieLens-100k dataset. A Comparison with other approaches are shown in the figure below.

Recommendation Systems on MovieLens 100K



View RMSE (u1 Splits) All models Edit

RANK	MODEL	RMSE (U1 SPLITS)	RMSE (RANDOM 90/10 SPLITS)	EXTRA TRAINING DATA	PAPER	CODE	RESULT	YEAR
1	Bayesian timeSVD++ flipped + Feat w/ Ordered Probit Regression	0.882		✓	On the Difficulty of Evaluating Baselines: A Study on Recommender Systems			2019
2	Bayesian timeSVD++ flipped + Feat	0.884		✓	On the Difficulty of Evaluating Baselines: A Study on Recommender Systems			2019
3	Bayesian timeSVD++ flipped	0.886		✓	On the Difficulty of Evaluating Baselines: A Study on Recommender Systems			2019
4	GraphRec + Feat	0.897	0.883	✓	Attribute-aware non-linear co-embeddings of graph features			2019
5	GraphRec	0.904	0.887	×	Attribute-aware non-linear co-embeddings of graph features			2019

The above figure is obtained from https://paperswithcode.com/sota/collaborative-filtering-on-movielens-100k

Example Recommendations from the Model

Recommendations based on Movie Embeddings. The movie Embedding Space is searched for similar items.

EX 1

Movie – Star Wars (A New Hope)
Recommended Movies – ['Before Sunset', 'Return of the Jedi', 'The Dark Knight', 'The Empire Strikes Back']

EX 2

Movie - The Lord of the Rings: The Fellowship of the Ring
Recommended Movies - ['Gladiator', 'Return of the Jedi', 'Star Trek', 'Star Wars', 'The Lord of the Rings: The Return of the King', 'The Lord of the Rings: The Two Towers', 'The Matrix']

In both examples, it is observed that the model could recommend the remaining movies of the trilogy based on the content of the input movie.

Similarly, the User Embedding Space can be searched for similar users.

Personalized Recommendations. The user profile is first obtained, which contains movies the user has rated highly. The Average Movie Embedding is found from the user profile to capture the user's interests, this is then used to search the Embedding Space for movies that share features with the user's interests.

EXAMPLE - User Id: 288

User Profile

	userId	id	title	genre1	genre2	key1	key2	rating	genres
14016	288	697	The Fox and the Hound	1	2	0	6	4.5	[Adventure, Animation, Drama]
35773	288	1685	10 Things I Hate About You	3	13	410	625	5.0	[Comedy, Romance, Drama]
36167	288	5653	(500) Days of Summer	3	5	44	54	4.5	[Comedy, Drama, Romance]
53160	288	4086	13 Going on 30	3	7	27	373	4.5	[Comedy, Fantasy, Romance]
73919	288	1385	An American Tail: Fievel Goes West	1	2	366	569	4.5	[Adventure, Animation, Family]

If 'w' represents the Embeddings, w_{avg} = (w[697] + w[1685] + w[5653] + w[4086] + w[1385]) / 5

w_{avg} is used to search the embedding space for similar items.

From the User profile, it is observed that the user has interests in movies with genres Romance, Comedy.

Recommended Movies - ['13 Going on 30', 'Sleepless in Seattle', 'Syriana', 'The Mirror Has Two Faces', 'The Notebook']

Thus, the model could capture the user's interest to recommend similar movies with genres Romance, Comedy.

E. FURTHER WORK

- With a bigger dataset and more input features, it is believed that the model could perform better at predicting ratings and extracting intrinsic properties from the items.