RS Methods Research and Report Writing

skww86
Computer Science
Durham University
Durham, United Kingdom
skww86@durham.ac.uk

I. Introduction

This paper will focus on the domain of movie recommendations, within the context of online streaming platforms. This is a market that has been rapidly growing in recent years and has huge potential in the future. Traditional technology giants such as Apple have even entered this market and are now competing against fellow organisations such as Amazon and Netflix. Thus, it is clear to see why having innovative, state of the art recommender systems within this domain is so appealing in the modern day. Specifically, the reason why there is such a need for recommender systems in online movie streaming platforms is because these platforms tend to purchase the rights to a very large number of movies. It would not be an optimal experience for a user if they had to browse through the platforms themselves in order to locate a movie they would like to watch, so instead a crucial tactic would be to recommend the user movies that they are likely to enjoy – thus, they would be satisfied with the overall service making them more likely to renew their subscription.

II. Methods

This paper focuses on two recommender system techniques: a collaborative filtering method, and a content-based filtering method.

A. Data Description

The dataset used shall be the MovieLens dataset, collected by GroupLens Research [1]. This dataset boasts 25 million user ratings, across 62,000 different movies, with those ratings having been submitted by 162,000 unique users. This makes it an incredibly resourceful data set. Each entry within the dataset includes the user, the movie, a timestamp, and the given rating, which can take any value between 0.5 and 5 in increments of 0.5. The movie is represented by a unique identifier which can be cross referenced with another file which stores the movie's title along with its associated genre(s). Additionally, each rating associates several tags to the movie, based on what keywords the user in question would associate with the movie, further increasing the depth and complexity of the data set.

B. Data Preparation and Feature Selection

The data preparation will be different for each of the two recommendation techniques that will be implemented. For the collaborative filtering technique, parsing the data will allow for the creation of a user by movie matrix, such that each row corresponds to a unique user, and each column corresponds to a movie. The values in the matrix provide the rating value. This matrix can then be used to lookup any given user's rating on any given movie. Note that there will of course be empty values within this matrix, since not all 162,000 different users will have rated all 62,000 movies. However, it is likely that such a matrix of size 162000 x 62000 will not be viable to use within processing on a personal computer anyway, so rows or columns that have the least data in this sense (i.e. those rows/columns that would be of least use to the algorithm) could be prioritised for removal so that the matrix is stripped down to a more manageable size.

For the content-based filtering technique, item vectorisation will be undertaken to create detailed vectors for each item (i.e. movie). This will be achieved by using the data (which includes relevant tags associated to each movie) to create a matrix of movies vs tags, allowing a program to be able to use this matrix to lookup any given tag's relevance to any given movie [5]. This matrix can then be juxtaposed with the user rating data such that we now have the feature vectors and the ratings together; this data would now be in a suitable format to pass to a machine learning regression model to train on. The matrix can then be split appropriately into test and train sets for the machine learning model to use.

C. Recommendation Techniques/Algorithms

The first technique used, collaborative filtering, works by considering the ratings submitted by other users in order to make predictions about what the given user's potential rating would be for a movie that they have not yet seen. Given the size of this dataset, it would not be viable or efficient to consider ratings submitted by all the other users (there are 162,000 different users), so a nearest neighbour approach will be used; ratings submitted by a suitable set of users, who are deemed to be the most similar users to the given user (in terms of rating history) are considered. The technique can then employ latent factor models in order to compute the recommendations. Using the user by movie matrix constructed during the data preparation (which we will refer to as R), the matrix can be factorised into user and item embedding matrices U and V respectively such that UV^T is an accurate approximation to R. To establish U and V and perform this factorisation, an objective function must be derived which represents the error between the ratings in $R' = UV^T$ and the original matrix R; the machine learning algorithm Stochastic gradient descent [6] can then be used to minimise this objective function (which is acting as the loss function) to find the optimal U and V matrices, in order to produce the optimal recommendations. There are a few reasons why this technique seems suitable for the purpose of this recommender system, the domain and the available data. Since the recommendations are originating from data based on what other users are rating, there is scope for a more diverse range of movies to be recommended, which would not be the case in other techniques, such as content-based filtering. In this domain that can be very useful, because it might mean more novel or niche movies could be recommended to the user who might then enjoy them and be more likely to continue their subscription rather than being recommended the same set of movies that they might have already heard of or watched in the past. Further, collaborative filtering techniques have a track record of providing accurate recommendations to users in this context, such as its use in the winning entry to the Netflix Prize,

which involved the exact same domain that is being considered in this paper.

The second technique used, content-based filtering, operates by predicting how a user would rate a movie based on their previous ratings of other similar movies. Data from other users is not taken into account; thus, this technique can provide more personalised recommendations to users, which can be very useful in this domain since there are often users who prefer very niche/specific genres of movies and may not wish to simply be recommended whatever is trending at the time. The technique works by creating detailed vectors for each movie, which in this dataset, considers the movie genre and the user associated tags for each movie. A function is to be created [4] that takes in such a vector and returns the most probabilistic rating that the given user whose data the model trained on would give for that movie. A deep neural network [3] would perhaps be most appropriate to carry this out, given its ability to handle the depth and complexity of the item vectors, and its flexibility with regards to allowing additional query features. This method is less susceptible to the cold start problem, which has the potential to be quite damaging in this domain. This is because in order for the model to make a prediction on a new user, it simply needs to train on a new predictive model, and if a new movie is put onto the streaming platform, the algorithm only needs to calculate its vector to proceed. This is computationally more efficient and has better scalability implications than say other models such as matrix factorisation which would have to refactorize the entire matrix in the event of a new user. This is important in this a domain; these online streaming platforms are extremely popular and are constantly gaining new users and adding in new movies, so the recommender system must be well equipped to deal with this.

D. Evaluation Methods

To evaluate the matrix factorisation undertaken in the collaborative filtering based technique, a proportion of users and movies can be extracted from the data set. The remaining data can then be treated as a training set, and can be used to compute matrices U and V in the same way as before. This embedding function can then be applied to the test set to produce matrices again that correspond to the test set. This can then be compared to the original ratings matrix using RMSE [2], which should give us a holistic evaluation of the embedding functions, and the matrix factorisation process that was undertaken. A similar measure can be applied for the deep neural network that is used in the content-based filtering approach. The data corresponding to each user should be appropriately split into a test and train set. The test set can then be ran on the deep neural network (once it has trained on the training set), and an RMSE precision value can be computed based on the comparison between the results of the test set after having run it on the neural network, and the actual user ratings data. This should allow for a succinct evaluation of the neural network itself.

III. Conclusion

There are of course a few limitations with each of these techniques. The matrix factorisation method, used in the collaborative filtering based approach, is not particularly adept in terms of dealing with the addition of new items. Since the embeddings are computed using a static data matrix, based on all the movies' collective data at the time of computation, if a new movie was added to the platform, the entire factorisation would have to be carried out again. However, this could be mitigated by simply approximating the embedding for a new movie each time. With regards to the deep neural network that is used within the content-based filtering approach, it has the potential to end up being fairly computationally expensive to run. This is because the application owner will need to decide how often they wish to 'refresh' the neural network. Over time, users will continue to interact and rate more and more movies, which ultimately changes the nature of what the training data would be for that user should a neural network be established for that user. Too few refreshes and the recommender system is missing out on the most recent data and risking inaccuracy, but too many refreshes and there are implications for the availability of computational resources.

References

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872

[2] C. C. Aggarwal, *Recommender Systems*. Springer, 2016.

[3] Cataldo Musto, Tiziano Franza, Giovanni Semeraro, Marco de Gemmis, Pasquale Lops. Deep content-based recommender systems exploiting recurrent neural networks and linked open data. In Adjunct Publication of the 26th conference on user modeling, adaptation and personalization

[4] K. Bougiatiotis and T. Giannakopoulos, "Enhanced movie content similarity based on textual, auditory and visual information," *Expert Systems with Applications*, vol. 96, 2018.

[5] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems," *Big Data Mining and Analytics*, vol. 1, no. 4, 2018.

[6] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.