Final Report

Timolai Andrievich

I. Introduction

Recommender systems are a vital part of current internet. Practically all major businesses, starting from YouTube and Netflix, and ending with Amazon rely on them to attract and maintain customers. While massive advancements have been made in numerous fields of research thanks to artificial neural networks, the workings of the majority of recommender systems stay a trade secret. We aim to explore solutions to the recommendation task, using MovieLens 100K dataset [1].

II. Data Analysis

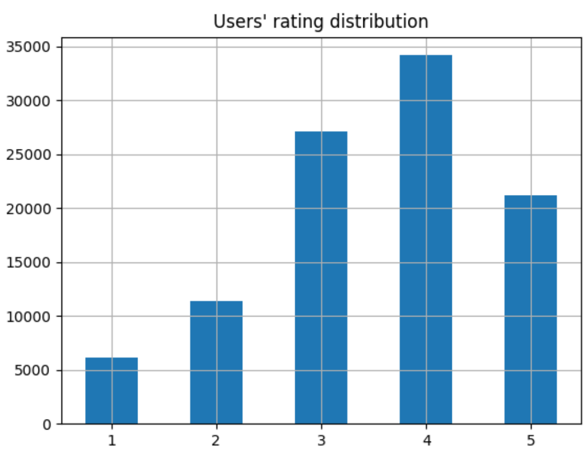

Fig. 1. Distribution of users' ratings

The dataset includes 100K entries, each entry consisting of a user ID, a movie (henceforth referred to as "item"), and a rating the user gave to the movie. As Figure 1 shows, users' ratings are skewed towards the positive side. Figures 2, 3, and 4, provide the information about age, gender, and occupation distribution of users. It is worth mentioning that the gender distribution is skewed towards male, and the mode age is around 30, despite the minority majority of the users being students. As for the films, Figure **??** shows the genre distribution. As seen on the figure, a very small fraction of films is of an unknown genre.

III. Model Implementation

As the base for the model, we used the GraphRec architecture by Rashed et al. [3]. Ironically, however, the graph-derived feature the model got its name from are not used in the modified architecture we ended up using. First, the user and item information are encoded as follows: user age is scaled

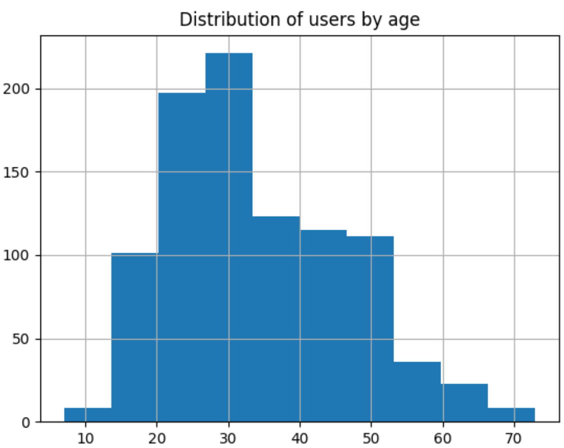

Fig. 2. Distribution of users by age

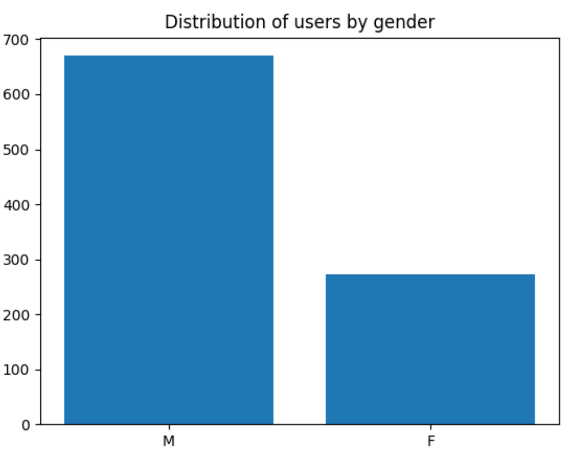

Fig. 3. Distribution of users by gender

and shifted to a range of $[0, 1]$, gender is encoded as either 0 or 1, and occupation is one-hot encoded. The item information is encoded in a similar fashion: the genres of the film are encoded as a binary vector, and the timestamps of the release date of the film are scaled to the $[0, 1]$ range. Here, we deviate from the paper, as the authors of the original architecture break the date up into year, month, etc. Authors also use one-hot encoding for the user and item IDs, however this approach may make it difficult to extend the model to new users and items, thus we learn embeddings for users and items and concatenate them to the encoded information. Then, a single fully connected layer

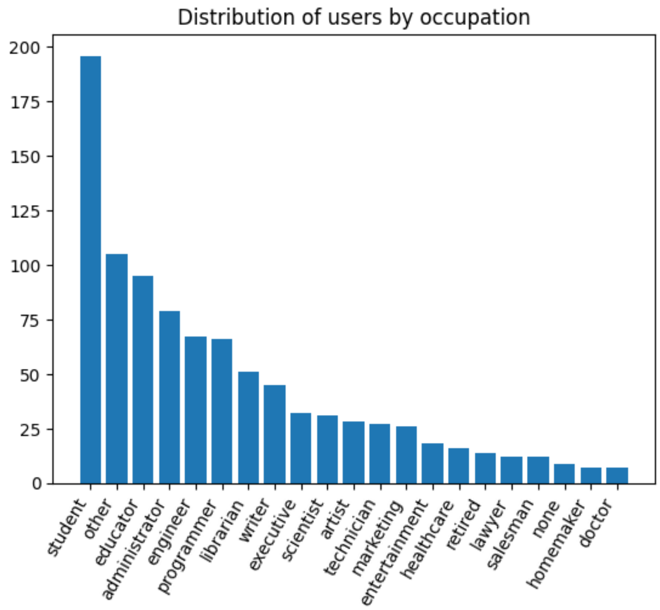

Fig. 4. Distribution of users by occupation

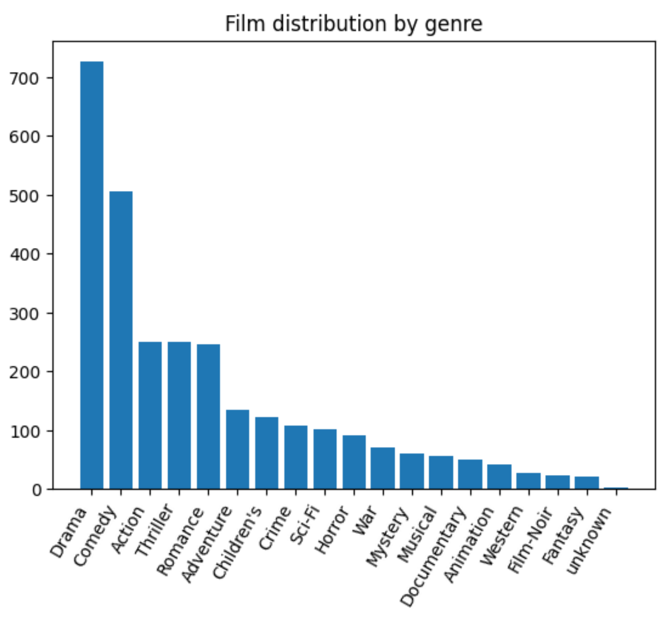

Fig. 5. Distribution of films by genre

followed by a ReLU activation is used to map the features to a 50-dimentional latent space. Separate networks ϕ_u and ϕ_i are used for items and users. Then, to predict the rating of the item, the dot product of the latent-space vectors is taken. See Figure 6 for details.

IV. Model Advantages and Disadvantages

Because the model does not use a big number of parameters, it is computation-light, which may be important to recommend to a vast array of users at once. However, the model heavily depends on already known information, and would have to

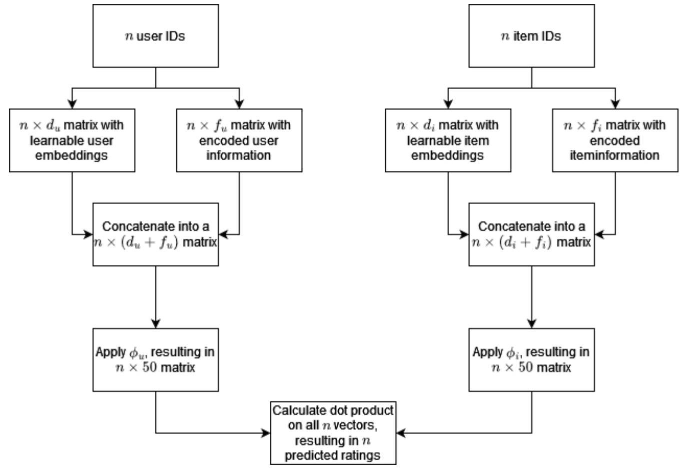

Fig. 6. Architecture of the model

be augmented in some way to deal with the "cold start" problem. Furthermore, because of the lack of graph-derived features present in the GraphRec model, the performance suffers. However, in contrast to GraphRec, the model does not directly utilize the IDs of users and items, and thus extends more easily to new user and movie entries.

V. Training Process

The split of the dataset is 90%/5%/5% for train, validation, and test sets respectively. The model is trained on batches of 64, using an Adam optimizer with a learning rate of 1×10^{-3}. The optimized loss is:

$$\mathcal{L}(\hat{y}, y) = \mathbb{E}\left[(y - \hat{y})^2\right] + \lambda_{\text{user}}||U|| + \lambda_{\text{user}}||I|| \qquad (1)$$

Where U and I are the users and items embeddings, y are the ground truth ratings, $\hat{y}$ are the predicted ratings. In other words, the loss used is the MSE loss with added L2 regularization of user and item embeddings.

VI. Evaluation

We chose RMSE as our evaluation function, as the structure of the dataset is not very well fitted to metrics like precision, recall, and F1 score, as the target variable is the rating of the film, and not a binary value like relevancy to a given query. Furthermore, using RMSE makes comparing the model to the state-of-the-art models more convenient, as the majority of the papers using MovieLens 100K dataset used RMSE as well.

After training, the model scored 0.837 RMSE on the train set, and 0.962 and 0.952 RMSE on validation and test sets respectively. This is a drop of 0.069 from the GraphRec's 0.883 RMSE. However, the model still manages to beat 2014 state of the art - GMC, proposed by Kalofolias et al. [2].

VII. Results

We modified the GraphRec [3] architecture, and implemented it, putting emphasis on the speed of inference, while still maintaining acceptable quality and allowing easier extension of the model in modern neural network frameworks.

References

[1] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[2] Vassilis Kalofolias, Xavier Bresson, Michael Bronstein, and Pierre Vandergheynst. Matrix completion on graphs. *arXiv preprint arXiv:1408.1717*, 2014.

[3] Ahmed Rashed, Josif Grabocka, and Lars Schmidt-Thieme. Attribute-aware non-linear co-embeddings of graph features. In *Proceedings of the 13th ACM conference on recommender systems*, pages 314–321, 2019.