Neural Collaborative Filtering Book Reccomender

pnvv44
Department of Computer Science
Durham University
pnvv44@durham.ac.uk

I. INTRODUCTION

A. Domain of Application

The UCSD Book-Graph dataset, obtained from the popular online website Goodreads[1], has been employed in several studies for the evaluation of recommendation algorithms [1]. The platform enables users to rate, review, and recommend books, making it valuable source of data in the literature [2]. The primary focus of our proposed system is to facilitate the discovery of new books, tailored to individual preferences.

B. Related Work

Neural Collaborative Filtering (NCF) [3] is a recent approach to Collaborative Filtering (CF) that utilizes deep neural networks (DNNs) to learn the user-item interactions. It has been shown to be effective in handling large amounts of sparse data, which is a common challenge in recommendation systems [4].

Traditionally, the dot product has been utilized to determine similarity. However, recent advancements [5] have shifted towards utilizing neural networks, specifically multilayer perceptrons (MLPs), to learn the similarity function. The reasoning behind this shift stems from the belief that MLPs, being general function approximators, would offer improved performance over fixed similarity functions like the dot product. Empirical studies have shown that NCF models with deeper neural networks achieve improved results compared to traditional methods [4][6]. Nevertheless, empirical evaluations of properly configured matrix factorization models do not provide evidence supporting the superiority of MLPs, and it presents some limitations such as increased model parameters and increased computational cost during serving [7].

Our paper utilizes a neural network based matrix factorization approach to capture non-linear relationships between users, items and their ratings, balancing the strengths of both matrix factorization and neural networks.

C. Purpose

A system titled *Readable*, was developed to allow users to discover literature that aligns with their personal tastes and interests. The design of the system was informed by an analysis of existing recommendation platforms, with particular attention paid to the user interface and presentation of recommendations on platforms such as Spotify[2].

The system, guided by the mission statement of "find your next great read" adopts a community-based approach, with the primary objective of optimizing the user experience through the incorporation of interactive controls, a minimalistic user interface, and explainable recommendations. Additionally, the system has been designed with security as a key consideration, incorporating validation measures for all inputs to safeguard sensitive user data.

II. METHODS

A. Data Preprocessing & Feature Selection

The UCSD Book-Graph dataset was processed in order to generate the final dataset utilized in our implementation. The selection of features was based on both relevance and computational feasibility. This resulted in the creation of two *.csv* files containing ratings and book information, as shown in Figure 1.

In order to enhance the quality of the dataset, various preprocessing steps were implemented. These included the elimination of unrequired book metadata, removal of null rows, standardization of the *'book_id'* column across both files, and inclusion of books with sufficient ratings. Additional columns in the *ratings.csv* file containing explicit data were also removed to ensure relevance. It is worth noting that *books.csv* was only used for ease of output and was not considered in the prediction model.

ratings.csv	'user_id', 'book_id', 'rating'
books.csv	'book_id', 'title', 'ratings', 'url'

Fig. 1. Dataset *.csv* files

The original dataset is of substantial size yet exhibited a high degree of sparsity. As such, the filtering method described in [3] was applied to retain only users who had provided a minimum of 20 ratings. This resulted in the generation of a user-book matrix R (of ratings on a scale of 1 to 5) with a 0.27% reduction in sparse data. The characteristics of the final dataset are outlined in Table I.

TABLE I. FINAL DATASET STATISTICS

Field description	**Value**
Number of Users	228
Number of Books	59139
Number of Ratings	100000
Data sparsity	99.67%

[1]https://www.goodreads.com/
[2]https://open.spotify.com/

B. Reccomendation Techniques

In our implementation, we adopt a neural network based matrix factorization approach as outlined in I.B. To ensure optimal performance, an embedding size of 50 was selected for both the user and item representations. Smaller embedding sizes are known to work well with sparse datasets [8], and the selected size was found to balance the trade-off between model complexity and performance in our experiments.

$$\text{Loss} = -\frac{1}{\text{output size}} \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i) \tag{1}$$

In terms of optimization, the binary cross entropy loss function (Eq (1)) is minimized using the Adam algorithm, which has proven to be computationally efficient in previous studies [9]. L2 regularization is also employed to prevent over-fitting and effectively learn latent representations of users and items. Figure 2 displays the architecture of the NCF framework.

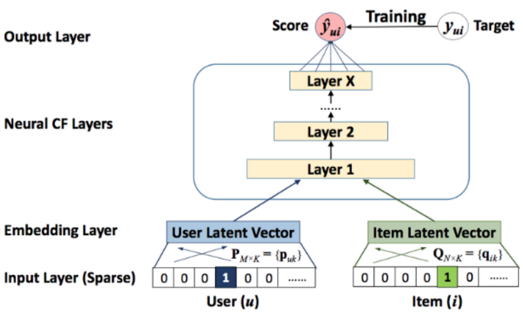

Fig. 2. Neural collaborative filtering framework [6]

C. Evaluation Methods

The evaluation for *Readable* was done using an offline experiment specifically on user 5's ratings.

The prediction accuracy was evaluated using Root Mean Squared Error (RMSE) and is represented by the Eq (2). The metric gauges the system's ability to accurately predict a user's rating for an item. The evaluation involved removal of 2, 4 and 6 books from user 5's profile.

$$\text{RMSE} = \sqrt{\frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}. \tag{2}$$

The accuracy of ranking was measured using Precision and Recall. These metrics are concerned with whether the generated recommendations are relevant to the user. The evaluation involved removal of 10 interaction from user 5's profile. The metrics are is represented by the following equations:

$$recall = \frac{TP}{TP+FN} \tag{3}$$

$$precision = \frac{TP}{TP+FP} \tag{4}$$

III. Implementation

A. Interface

The system prompts users with a login request, which undergoes a validation process to ensure the input is accurate. In case multiple consecutive invalid attempts are made, an error message is displayed, and the system terminates its operation. Upon a successful login, a welcome message is presented, and users are introduced to the main menu.

The main menu comprises several options, including viewing past ratings, receiving new recommendations, exploring popular picks, and logging out. Each option is accompanied by an informative description, which makes the system user-friendly and straightforward to navigate. Future development could include the addition of a feature that enables users to rate books through the interface.

Readable adopts an ethical approach towards explainability and transparency in its generated recommendations. This is achieved by providing concise textual descriptions that inform users about the data that is being collected and processed. In addition, the popular picks option takes inspiration from Netflix's 'Top 10 Movies' display, offering recommendations based on popularity metrics.

The interface, as shown in Figure 3 and 4, is designed with the user experience in mind. As such, a minimalistic display of recommendations serves to prevent information overload.

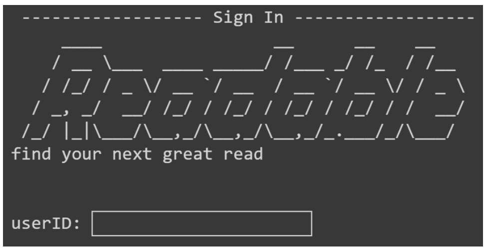

Fig. 3. Login screen

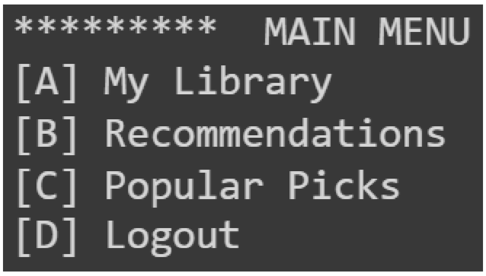

Fig. 4. Main Menu view

B. Reccomendation Algorithm

In our recommendation algorithm, the match score between user and item embeddings is computed through the dot product of their respective vectors. To account for individual user preferences and item popularity, a bias term is added for each user and item.

The final predicted rating is generated by passing the sum of the dot product and bias terms through a sigmoid activation function $\sigma(x) = 1/(1 + e^{-x})$. This activation function maps the output to the interval [0, 1], making it suitable for datasets where ratings are normalized within this range, as is the case in our study.

IV. EVALUATION

A. Accuracy of rating predictions

MAE testings were conducted by removing 2, 4, and 6 books respectively, and calculated as per Eq (1). The evaluation of the NCF model resulted in a negligible difference in scores, likely due to the small proportion of items removed in comparison to the size of the dataset. The results are highlighted in Table II.

It is crucial to acknowledge that achieving an RMSE score of 0 is a highly unlikely scenario in real-world applications due to the presence of inherent inaccuracies in the prediction process. The low RMSE score in this particular case can be attributed to the popularity-based recommender approach, which filters and presents the top 10 rated books each time. The results of the evaluation imply that the books excluded from the analysis were likely not included in the top 10.

Given these considerations, it would be appropriate to assess the performance of the recommender system using a metric that is better suited for comparison between personalized and non-personalized recommendations. Future work could also aim to design more appropriate offline experiments, such as excluding a greater number of books to further evaluate the effectiveness of the recommender system.

TABLE II. RMSE

	2	4	6
NCF	0.059	0.059	0.060
Popularity	0	0	0

B. Accuracy of ranking

10 ratings were removed to conduct the precision and recall evaluation. A high precision value of 1 indicates that all of none of the recommended books are irrelevant. The recall value suggests that 85.71% of the relevant books for the user have been recommended by the system. Overall, these results suggest that the NCF model is performing well in terms of precision but could still improve in terms of recall. Further analysis, such as examining the characteristics of the removed ratings, may help to understand the impact of the removal on the system's performance.

TABLE III. PRECISION & RECALL

	NCF	Popularity
Precision	1.00	1.00
Recall	0.86	1.00

V. CONCLUSION

In conclusion, the present study introduces the *Readable* system, which aims to provide personalized literary recommendations to users. The system employs a neural network-based matrix factorization approach, incorporating the benefits of both matrix factorization and neural networks to capture non-linear relationships between users, books, and their ratings. In terms of user experience, the system prioritizes interactive controls, a user-friendly interface, and explainable recommendations. The security of sensitive user data was also considered in the design, through the use of appropriate validation measures.

A. Limitations & Future work

The limitations of this study include the ineffectiveness of the chosen metrics (RMSE, precision, recall) and the inadequacy of the offline experiment for our neural network matrix factorization book recommendation domain. The implementation was also limited by the restricted dataset and computational constraints, which highlights the limitations of the analysis.

Future work could focus on incorporating implicit data. More appropriate evaluation protocols and metrics, specifically designed for DNN frameworks, such as the strategy of randomly sampling 100 items not interacted by the user [11] and using Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [12] as evaluation metrics, should also be considered.

REFERENCES

[1] Wan, M., & McAuley, J. (2018). Item recommendation on monotonic behavior chains. Proceedings of the 12th ACM Conference on Recommender Systems. https://doi.org/10.1145/3240323.3240369

[2] Wan, M., Misra, R., Nakashole, N., & McAuley, J. (2019). Fine-Grained Spoiler Detection from Large-Scale Review Corpora. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. https://doi.org/10.18653/v1/p19-1248

[3] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web - WWW '17. https://doi.org/10.1145/3038912.3052569

[4] Najafabadi, M. K., Mohamed, A., & Onn, C. W. (2019). An impact of time and item influencer in collaborative filtering recommendations using graph-based model. Information Processing & Management, 56(3), 526–540. https://doi.org/10.1016/j.ipm.2018.12.007

[5] He, X., Du, X., Wang, X., Tian, F., Tang, J., & Chua, T.-S. (2018). Outer Product-based Neural Collaborative Filtering. ArXiv:1808.03912 [Cs, Stat]. https://arxiv.org/abs/1808.03912

[6] Zhao, X., Zhu, Z., Zhang, Y., & Caverlee, J. (2020). Improving the Estimation of Tail Ratings in Recommender System with Multi-Latent Representations. Proceedings of the 13th International Conference on Web Search and Data Mining. https://doi.org/10.1145/3336191.3371810

[7] Rendle, S., Krichene, W., Zhang, L., & Anderson, J. (2020). Neural Collaborative Filtering vs. Matrix Factorization Revisited. Fourteenth ACM Conference on Recommender Systems. https://doi.org/10.1145/3383313.3412488

[8] Liu, S., Gao, C., Chen, Y., Jin, D., & Li, Y. (2021). Learnable Embedding Sizes for Recommender Systems. ArXiv:2101.07577 [Cs]. https://arxiv.org/abs/2101.07577

[9] Kingma, D. P., & Ba, J. (2014, December 22). Adam: A Method for Stochastic Optimization. ArXiv.org. https://arxiv.org/abs/1412.698

[10] Team, K. (2020). Keras documentation: Collaborative Filtering for Movie Recommendations. Keras.io. https://keras.io/examples/structured_data/collaborative_filtering_movielens/

[11] Elkahky, A. M., Song, Y., & He, X. (2015). A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. Proceedings of the 24th International Conference on World Wide Web. https://doi.org/10.1145/2736277.2741667

[12] He, X., Chen, T., Kan, M.-Y., & Chen, X. (2015). TriRank. Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. https://doi.org/10.1145/2806416.280650