Recommender Systems

I. Introduction

The inundation of digital content and data on the Internet hinders a user's ability to navigate a service to access compelling content. To ensure the vast amount of rich information often provided by a service can be exploited, recommender systems are crucial. These systems filter information to present items that align with a user's preferences, interests and recorded behaviour [1] . By learning user preferences and exploiting this ethically, recommender systems provide a personalized experience which amplifies user satisfaction and engagement.

Recommender systems are particularly critical in the movie domain where it is infeasible to explore the entire domain manually. The value of these systems in streaming services, is showcased through Netflix's recommender systems which account for 80% of stream time and were valued at an estimated $1 billion in 2016 [2]. These systems ameliorate user engagement and satisfaction which is conducive to increasing the revenue for a business through increased consumption. Moreover, as a result of the Netflix prize, state of the art recommender systems are often established in this domain, and go on to be applied very effectively in this and other domains. Thus, we have chosen the movie domain as the sector to implement our recommender systems.

We explore two state-of-the-art deep model-based recommender systems, a novel content-collaborative hybrid approach and a solely collaborative approach. We aim to produce two recommender systems that foster user satisfaction in different ways; one by capturing the complex user-item interaction structures and a second that exploits temporal data. We contrast the different methodologies and evaluate which is most effective with respect to customer satisfaction and engagement - metrics conducive to profit. We also comment on which feedback form (explicit or implicit) is more preferable to exploit from the perspective of our results.

II. Methods

We utilize the MovieLens 1M dataset which contains 1,000,209 anonymous ratings made by 6040 users with a rating system from 1 to 5. It has a sparsity of 95.73% with approximately 3,900 movies. This sparisty was not reduced (by discarding certain users or items) as it reflects the sparse nature of real world data and allows our findings to be applicable in real-world systems. This dataset is advantageous as it contains abundant demographic data pertaining to users, temporal data and the ability to gather additional data on each item through its link to IMDB. This collection of data enables a deep and reflective user profile to be constructed which is conducive to our aims.

Matrix factorisation (MF) is well utilized in the context of recommender systems. However, the use of the inner product (taken between the user and item latent vectors) places severe limitations on the algorithm's ability to encapsulate the complicated structure of user-item interaction data [3]. In the movie domain, user-item interactions can be very complex. For example, one user can hold positive opinions about two films that differ tremendously in genre and stylistic choices. To produce a recommender system that produces dependable recommendations, in line with our aims, and is suitable for this domain (by accounting for these complex structures) NeuMF is chosen [3]. NeuMF is a state-of-the-art model that retains the success of MF and advances it further.

NeuMF builds upon the success of MF by introducing nonlinearities through neural architecture. The inner product is replaced with a deep neural architecture whereby the sparse user and item vectors are projected to dense vectors in a shared latent space using a fully connected embedding layer. These embeddings are then fed into the general matrix factorization layer (GMF) - a multi-layered neural network. The GMF layer learns certain latent feature interactions.

The model is furnished with the ability to model non-linear interactions through the use of a multi-layer perceptron (MLP). The latent embeddings are concatenated and passed through the MLP which learns the interaction function between user and item latent features. These two separate architectures are concatenated together and passed through a dense layer to map to a prediction.

The user profile is learned by the neural network by passing training data and minimizing the binary cross-entropy loss:

$$-\sum y_{ui}\log(\hat{y_{ui}}) + (1 - y_{ui})\log(1 - y_{ui})$$

Where y_{ui} corresponds to the rating given by user i and $\hat{y_{ui}}$ is the model's prediction. This model utilizes implicit feedback. The MovieLen's explicit feedback is converted to implicit feedback through mapping all interacted items with a one and zero elsewhere. This is a suitable mapping as it captures the noisy interactions one would expect in implicit feedback. The final model is passed a user and an item identifier and outputs the confidence of the user liking the item.

[3] identifies the user and item through a one-hot encoded vector, not utilizing any content or demographic information to make these vectors expressive. This report makes a novel contribution by crafting vectors for the users and items that incorporate demographic and content information. This contributes to the neural architecture learning a more descriptive user profile in line with our aims, and exploits the descriptive information available (unique to the movie domain) to help create a model capable of modeling the complex dynamics

in the movie domain. To develop the item vectors, content pertaining to each movie was extracted from the IMDB page for each film.

The poster associated with a film encodes very expressive data as it is crafted to align with the target audience's preferences and can be indicative of the genre. Moreover, as films are visual mediums, the poster is very closely aligned with the style and design features of the film making it a very expressive representation of the film. To exploit this information, the movie posters were embedded in a vector space by passing the images of the posters through the pre-trained convolutional neural network ResNet50 [4] with 50 convolution layers. The information is encoded through the shrinking convolution features maps produced by each layer. As ResNet50 has been trained on over a million images it is capable of creating rich feature representations for the posters.

A second source of valuable information is the description of the film which summarizes the film's storyline and contains information pertaining to plot points, emotional components, main characters and the movie's setting. These factors encode information that motivates a user's preference for a film. To embed this content into a vector space, it was encoded through the Bidirectional Encoder Representations from Transformers (BERT) [5]. The BERT model, a state-of-the-art model, produces a vector with high and telling semantic meaning.

Finally, the names of the leading star and director were one-hot encoded to produce a third vector. Often directors have a signature style and so user preferences can be aligned with the director. Moreover, users are often motivated to watch a film because of their favourite star. Other features, such as genre, were disregarded as this information was already encoded in the features selected (such as description) and it would reduce the generalisability of the model to purposefully include redundant features.

To combine these three distinct vector spaces the tensor product was taken. This yielded a very high dimensional space which was projected to lower dimensional space by compressing the vectors using an autoencoder. An autoencoder was chosen instead of principal component analysis as autoencoders are more suitable for modelling complex nonlinear functions which are present in this data. This produced item vectors with a dimensionality of eight.

The user vector was created by first one-hot encoding gender and occupation. The zip code was then converted to longitude and latitude so there was a more meaningful relation between different location representations. To reduce the number of features, the filter method chi-squared was applied [6]. This measures the dependence between variables to remove variables that are independent to the rating. This is fast to compute and produces a feature set that is not constrained by the assumptions of a predictive model (as would be the case if a wrapper method was applied). This enables the features generated to be more generalizable and so allows the report to isolate the effectiveness of the recommender method.

The second recommender system is conducive to the movie domain as it accounts for the temporal evolution of the user profile. A user's preference of movie is often dictated by their personal circumstances and lifestyle at the time. For example, a user might prefer lighter comedies in periods of stress or scientific documentaries aligning with her studying at school. Furthermore, new movies, emerging actors and rising directors induce changing trends that will also contribute to a user's preferences evolving quickly. Moreover, there are seasonal changes in preferences whereby a user might prefer romantic films near Valentine's Day or Christmas films approaching Christmas. Changes in movie perception also contribute to a dynamic user profile. [7] proposes Recurrent Recommender Networks (RNN) which endow a traditional low-rank matrix factorization model with a Long Short-Term Memory (LSTM) neural architecture to account for this dynamic evolution of the user profile.

The latent user vector for user i at time t are given by passing the previous latent vector at time $t-1$ and an embedded vector y_t into the LSTM

$$u_{it} = LSTM(u_{it-1}, y_t), m_{jt} = LSTM(m_{jt-1})$$

where

$$y_t := W_{embed}(x_{it}, \tau_t)$$

where x_{it} is the rating vector for a given user at time t. The predicted rating is given by

$$\hat{r}_{ij|t} = f(u_{it}, m_{jt}, u_i, m_j)$$

$$:= \langle \tilde{u_{it}}, \tilde{m_{jt}} \rangle + \langle u_i, m_j \rangle$$

Where $\tilde{u_{it}}$ and $\tilde{m_{jt}}$ are learned affine transformations with additional embedding layers given by

$$\tilde{u_{it}} = W_{user} u_{it} + b_{user}, \tilde{m_{jt}} = W_{movie} m_{jt} + b_{movie}$$

. The parameters are learnt through an alternating subspace descent strategy.

To evaluate the accuracy of the recommender systems, accuracy of rating predictions, item rankings and usage predictions were considered. Accuracy of rating predictions is not appropriate as the first recommender system deals with implicit feedback (predicting a value between 0 and 1) whereas the second deals with explicit feedback (predicting a value between 1 and 5). Therefore, it would not be a fair metric as the error would be larger for the second recommender system. As the user will only consider the top proportion of recommendations (as movies can be quite long), the bottom proportion of recommendations do not impact the user's experience. As accuracy of usage predictions takes into account all recommendations it is not appropriate for this domain. We use the accuracy of item rankings metric as we can evaluate the top recommendations (which will have the biggest impact on user experience). Specifically, normalized discounted cumulative gain (NDCG) was chosen as it measures the usefulness of a movie ranking based on its position and allows movies to be equally relevant (for example when a user rates two movies the same). It is given by the formula (where g_{ui} is the relevance of the recommendation):

$$\frac{1}{N}\sum_{u=1}^{N}\sum_{j=1}^{J}\frac{g_{ui}}{\max(1, \log(j)}$$

Broadening a user's horizons ameliorates user engagement and satisfaction [8] which is in line with this report's aims. Failure to return diverse and novel recommendations can confine users in filter bubbles [9] leaving them unengaged as they are recommended more of the same. Therefore, to accommodate the fluidity of user preferences and amplify user satisfaction, the diversity and novelty of recommendations are important [10]. Additionally, if streaming services recommend narrow recommendations the diversity of recommendations can collapse via a butterfly effect as the recommender systems reduce the profitability of novel movies potentially resulting in a collapse in diversity within the industry itself.

As the first recommender system was fed an expressive vector representing the item, it could be argued it would not be fair to compare the two systems utilizing diversity as this system received more information regarding the similarity of items. Therefore novelty was chosen, with $p(i)$ representing the proportion of users that rated item i and R taken to be 10:

$$\sum_{i\in R}\frac{\log_2 p(i)}{R}$$

III. Implementation

The implement and design decisions are discussed in the video.

A. Output

```
Please enter a number for how many recommendations you want:
5
Recommendation 1. Unforgettable (1996) with a confidence of 19.0%
Recommendation 2. Quick and the Dead, The (1995) with a confidence of 18.0%
Recommendation 3. Nueba Yol (1995) with a confidence of 18.0%
Recommendation 4. Gumby: The Movie (1995) with a confidence of 18.0%
Recommendation 5. Rumble in the Bronx (1995) with a confidence of 18.0%


These recommendations have been generated because you rated the films Welcome to the Dollhouse
(1995), Muriel's Wedding (1994), All That Jazz (1979), West Side Story (1961), Election (1999)
highly
```

IV. Evaluation

For both metrics the ratings dataset was randomly split into train and test datasets, with 80% being used to train the model and 20% being used for evaluation. These experiments were repeated across 3 seeds, with the results being averaged:

Results		
	NCDG	***Novelty***
NeuMF (with content)	0.78	1.91
RNN	0.77	2.91
NeuMF	0.74	-

Evidently, both recommender systems score well in accuracy. These scores indicate that the top recommendations presented to a user are relevant, which allows us to reach our target aim of cultivating user satisfaction. It indicates that there is value in exploiting the temporal aspect of the data as we argued. However, we note NGFC does not penalize irrelevant recommendations so an additional ranking metric (such as the Spearman correlation) would need to be evaluated to conclude concretely on this. We conclude the choice of explicit or implicit feedback does not appear to make a notable difference to the performance of the recommender systems. We note these scores could be improved further through refining the neural architecture by experimenting with deeper and shallower networks.

However, there is a disparity in the novelty, with the second recommender system producing more novel recommendations. This could be because the first recommender system is over-specialising after being fed the content vectors. To alleviate this problem, this report could pursue the integration of a genetic algorithm, which [11] showed to be promising. Alternatively, we could inject a random component into the item and user vectors or introduce a filtering step whereby items that are too similar to items the user has rated previously are removed.

This report also finds a gain in performance when compared to the purely collaborative version of NeuMF confirming our hypothesis that exploiting the rich content information available in the movie domain ameliorates performance.

V. Conclusion

This report has implemented two state-of-the-art recommender systems, making contributions to the first one. We have met the aims of the report and have reflected on specific paths of further work to bolster performance. These systems have demonstrated the value in exploiting content and temporal data in producing a recommender system that cultivates user satisfaction. Additional online experiments could be undertaken to strengthen this conclusion.

References

[1] F. Isinkaye, Y. Folajimi and B. Ojokoh, "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal, vol. 16, no. 3, pp. 261-273, 2015. Available: 10.1016/j.eij.2015.06.005.J.

[2] H. Steck, L. Baltrunas, E. Elahi, D. Liang, Y. Raimond and J. Basilico, "Deep Learning for Recommender Systems: A Netflix Case Study", AI Magazine, vol. 42, no. 3, pp. 7-18, 2021.

[3] He, L. Liao, and H. Zhang , "Neural Collaborative Filtering", 2017.

[4] He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", arXiv.org, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[5] Devlin, M. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[6] K. Hopf and S. Reifenrath, "Filter Methods for Feature Selection in Supervised Machine Learning Applications – Review and Benchmark", arXiv.org, 2021. [Online]. Available: https://arxiv.org/abs/2111.12140.

[7] C. Wu and A. Ahmed, "Recurrent Recommender Networks — Proceedings of the Tenth ACM International Conference on Web Search and Data Mining", Dl.acm.org, 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3018661.3018689.

[8] F. Passino and L. Maystre, "Where To Next? A Dynamic Model of User Preferences — Proceedings of the Web Conference 2021", Dl.acm.org, 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3442381.3450028.

[9] T. Nguyen, "Exploring the filter bubble — Proceedings of the 23rd international conference on World wide web", Dl.acm.org, 2021.

[10] P. Gravino, B. Monechi and V. Loreto, "Towards novelty-driven recommender systems", Comptes Rendus Physique, vol. 20, no. 4, pp. 371-379, 2019. Available: 10.1016/j.crhy.2019.05.014.

[11] O. Stitini and S. Kaloun, "An Improved Recommender System Solution to Mitigate the Over-Specialization Problem Using Genetic Algorithms", Electronics, vol. 11, no. 2, p. 242, 2022