A Hybrid Recommender System with Collaborative Filtering and Content-Based Filtering

Hdpb88
Department of Computer Science
Durham University
Durham, United Kingdom

The basic function of a recommender system is to parse a dataset, often too large for a user to manually browse, into a ranked presentation of the information of most interest to a user [7]. A hybrid recommender is such a system that implements two separate methods of recommendation, then combines the results so as to increase accuracy and robustness [6].

Key words: collaborative filtering, recommender systems, content-based filtering, hybrid recommender systems

I. Introduction

A domain in which recommender systems are becoming vital is that of business recommendations. With online representation becoming vital for new businesses, it is becoming increasingly difficult for a user to manually find the businesses they need. The Yelp dataset [1] for example, contains a vast array of businesses, users and interactions between the two. This paper attempts to show the effectiveness of a hybrid recommender system at filtering such a dataset in order to show a user business they will find interesting.

Hybrid recommenders have been explored in various combinations [2, 5, 6, 8, 9] with the addition of a second recommender algorithm frequently producing results surpassing those of their baseline components in multiple metrics. This result is not guaranteed however, some combinations of recommenders yield no benefit or even a hinderance to results [6]. Thus, it is vital that an appropriate combination of algorithms is used and that these algorithms are combined in a way that is conducive to improving their results.

Collaborative Filtering recommender systems involve the generation of similarities between users based on the reviews left by those users or their behaviours [11], then use of these similarities to predict ratings for unseen items. This is one of the more researched methods of recommending items to users [12, 2, 4, 7, 10, 11] and the logic of showing a user items rated well by 'similar' users has been shown to provide satisfactory results for well-populated data. Given the dataset being utilised in this system contains many thousands of user/business ratings, it follows that the literature presents collaborative filtering as a viable option.

Content-Based Filtering is the process of generating similarities between items based on features of those items, then using these similarities in tandem with the user's ratings to predict ratings for unseen items similar to the ones the user liked [6]. Various models of similarity generation have been used [8] however the most common is TF-IDF vectorisation such as in [9] and [13]. As a collaborative/content-based hybrid was shown to be a successful combination in [6], these present strong candidates for the algorithms used in this system.

The aim of this hybrid recommender is to provide a user with a ranked and ordered table of information, filtered from the Yelp dataset [1] into an easily-readable format based around the user's predicted enjoyment of the businesses shown.

II. Methods

The data for this recommender system is a filtered-down version of the Yelp Dataset [1], a set consisting of *business.json*, *review.json*, *user.json*, *checkin.json*, *tip.json* and *yelp_academic_dataset_covid_features.json*. Of these, the files *business.json*, *review.json* and *yelp_academic_dataset_covid_*features, were used to: provide content features about business items, build user profiles from their ratings and to filter businesses by their approach to the pandemic, respectively.

To prepare the data for use in the system, it was first filtered down to a usable size within the memory constraints given (8GB) by filtering the business data based on 'city' to exclusively those in Las Vegas, then the 'category' to only restaurants and finally to those reviewed in the year 2018. These parameters were chosen as Las Vegas is the city with the most businesses, a majority of these were restaurants, and these received the most reviews in 2018.

The filtered data was then prepared by removing unnecessary features. For the content-based filter, only business_ids and categories were needed to run the content-based filter. For the collaborative filter, the reviews table was reduced to just contain user_id, business_id and stars for each review. These were the only features required as collaborative filtering utilises only the recorded ratings of a user/item pairing.

In [6], a Collaborative/Content hybrid yielded an impressive result that was robust to changing datasets and adaptive to new user tastes. It also found that of the variety of ways to hybridise two recommender systems, cascade appeared to be recurringly successful.

The cascade hybridisation method consists of allocating a dedicated recommender system (in this case the collaborative filter) that is prone to ties, then using a secondary system that has its predictions used to break those ties hence the secondary recommender holds no influence over the order of recommendation made by the primary, but still imparts its own predictions onto the final result.

The two recommendation algorithms used in this hybrid are collaborative filtering and content-based filtering.

In both algorithms, a similarity matrix is generated, providing a similarity between each pair of items in the dataset. These similarity matrices are then used to perform a weighted average for each unrated item, of the similarities of that item to those rated by the user. The difference in these algorithms is the method through which the similarity matrices are generated and the data required to do so.

Collaborative filtering similarities are generated through the creation of a vector for each item consisting of the ratings it has been given. The similarities between these vectors form the matrix and are generated using the cosine similarity, defined in [15] as:

$$sim(i,j) = \cos(\vec{\imath},\vec{\jmath}) = \frac{\vec{\imath}\cdot\vec{\jmath}}{\left|\left|\vec{\imath}\right|\right|_2 * \left|\left|\vec{\jmath}\right|\right|_2}$$

The similarity matrix for content-based filtering is generated using TF-IDF vectorisation. TF-IDF vectorisation converts the categories feature of each business into a normalised vector using the equation in [14]:

$$TF-IDF\left(t_k,d_j\right) = TF\left(t_k,d_j\right)\cdot log\frac{N}{n_k}$$

In which, TF is the term frequency within the document, N is the total number of documents and n_k is the number of documents containing the term t.

Once both matrices are generated, the weighted average is taken as shown in [15] as:

$$P_{u,i} = \frac{\sum_{all\ rated\ items\ N}(sim(i,N) * R_{u,N})}{\sum_{all\ rated\ items\ N}(|N|)}$$

This recommender was evaluated using three metrics, evaluating accuracy of rating predictions, accuracy of usage predictions and diversity of recommendations respectively.

The first metric used was root mean squared error (RMSE) which is calculated as such [16]:

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{(u,i)\in T}(\hat{r}_{ui} - r_{ui})^2}$$

In which, T is a testing set of ratings where some removed ratings r_{ui} are known. This metric was chosen as it favours systems with multiple errors so long as they are not disproportionately large. Given the sparsity of the dataset being used, many errors are likely, so it is more worthwhile to minimise their magnitude than frequency.

The second metric evaluated was Precision, defined as the proportion of items the system predicted the user would rate, that were in fact rated by the user. This was chosen as a relatively small number of recommendations were being shown to the user by default, increasing the importance of these few recommended items being ones the user would in fact want to visit.

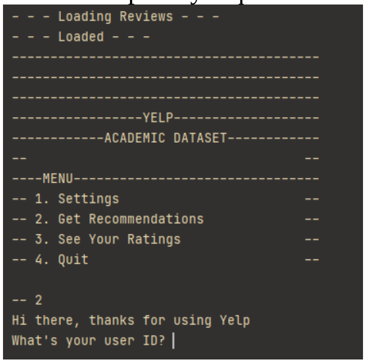
The final metric of evaluation was Diversity defined in [21] as:

$$D = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(1 - similarity\left(c_i,c_j\right))}{n/2 * (n-1)}$$

The lower the similarity, the higher the diversity of the system since it is presenting items of a varied taste.

III. Implementation

When interacting with the system, the current user is determined by requesting their Yelp user ID as shown in the dataset. As the system does not update or edit the user's ratings, the only data gathered about the active user is their ID, which is explicitly requested and temporarily stored.



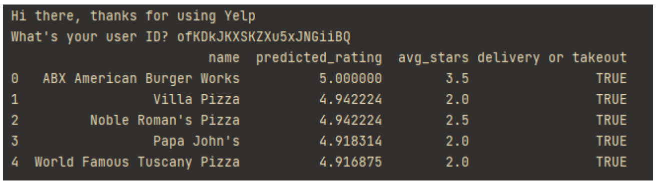
```
- - - Loading Reviews - - -
- - - Loaded - - -
----------------------------------------
----------------------------------------
----------------------------------------
-----------------YELP-------------------
------------ACADEMIC DATASET------------
--                                    --
----MENU--------------------------------
-- 1. Settings                        --
-- 2. Get Recommendations             --
-- 3. See Your Ratings                --
-- 4. Quit                            --

-- 2
Hi there, thanks for using Yelp
What's your user ID? 
```

Figure 1. Input UI

When loading the system by running the hybrid_filt.py file, the user is first informed of the loading of the reviews table stored in the data folder. If the user chooses to receive recommendations, they are asked to input their user ID which is then used to execute the following steps: Retrieve the items rated by the user and the stars given, use the pre-generated similarity matrices to evaluate the similarity of each item to those rated by the user, use these similarities to perform a weighted average of ratings, generating two sets of predictions, use the tie-breaker system to order items given identical predictions by the primary system.

The default number of recommendations shown is 5, however the user can change this functionality in the settings menu. Figure 2 shows the recommendations table.



```
Hi there, thanks for using Yelp
What's your user ID? ofKDkJKXSKZXu5xJNGiiBQ
                          name  predicted_rating  avg_stars delivery or takeout
0   ABX American Burger Works          5.000000        3.5                TRUE
1                 Villa Pizza          4.942224        2.0                TRUE
2         Noble Roman's Pizza          4.942224        2.5                TRUE
3                 Papa John's          4.918314        2.0                TRUE
4  World Famous Tuscany Pizza          4.916875        2.0                TRUE
```

Figure 2. Output UI

IV. Evaluation Results

The hybrid recommender (H) in this paper was tested using the established metrics RMSE, Precision and Diversity on random sets of 500 users with recommendations of size 5 and 30 ($_{5,\ 30}$). The recommender then had the hybrid elements removed so as to reduce it to functioning as a standard collaborative filtering recommender (CF) and tested on the same sets and recommendation sizes as the hybrid.

Metric	H_5	CF_5	H_{30}	CF_{30}
RMSE	1.0357	1.0222	1.0262	1.0192
Precision	0.0008	0.0004	0.0001	0.0001
Diversity	1.9993	1.9999	1.9995	1.9997

[4] Utilised a hybrid recommender system consisting of a linear combination of various recommender algorithms on the Yelp dataset achieving an RMSE of 0.609. [17] Utilised a deep neural network hybrid recommender on the FilmTrust dataset and resulted in an RMSE of 0.805. [18] Utilised a hybrid recommender system to analyse the MovieLens dataset, achieving an RMSE of 0.554. [19] proposed a hybrid recommender system that obtained 68.75 precision on the MetaFilter dataset. [20] Achieved a Diversity of 0.15 when 5

items were recommended to the user by their hybrid recommender. The system in this paper outperforms these examples in Diversity and slightly underperforms in RMSE. The Precision of the system is low, as expected due to the sparsity of the data used.

The recommender implemented in this paper utilises a series of mathematical concepts such as cosine similarity, root mean squared error and tf-idf vectorisation which to the user may be weakly understood if known of at all. Because of this, the user may place disproportional trust in the system. The result of this is the possibility of the system to guide users to choices and diminish their experience of personal identity.

It is the suggestion of this paper that an immediate solution to this ethical issue would be to further enhance the 'Explainability mode' of the system to deliver concrete justification for any recommendations made, however a longer-term, more necessary solution would be to introduce explanations of recommender systems to common web-pages.

A commonly known issue in recommender systems is the tendency of the system to develop some variety of bias towards certain items. In the case of this system, despite scoring well for the diversity of its recommendations, since both collaborative and content-based filtering algorithms suffer from the sparsity problem, items with fewer recommendations will be seen far less in recommendation results, only making them increasingly unlikely to receive ratings, creating a feedback loop. The result is a recommender system biased to prefer frequently rated items.

To address this bias, this paper adding functionality to pair new users to restaurants with equally few reviews in order to increase the dataset and reduce sparsity of business reviews while developing a profile for the new user.

While it can be seen as an advantage to use a third-party dataset taken from within the public domain due to its anonymised state, since the data was not collected by the creator of this recommender system, the collection process cannot fully be vetted. Also, should a user of Yelp request their data to be removed from the dataset, Yelp have no control over the local copy of the data used in this system and thus neither does the user.

Using a cloud-hosted version of this data would help address this issue and so would examination of the privacy statement of Yelp.com which could also be presented to users of this system.

V. Conclusion

Due to, both of algorithms suffering from the cold-start problem [2], this is still a large limitation of this system. Both recommender systems used rely on the availability of user reviews in order to develop a profile of similar users and to predict ratings for similar businesses to those rated.

Another limitation of the system is that it does not contain the ability to store additional reviews of users to those already in the dataset. While this does not impede the function of the system itself as a pure recommender, it is a concern that the user experience could suffer due to the inability of the system to track changes in user behaviour or taste.

The first improvement to the system would be the addition of the ability to create and update user profiles, in collaborative recommenders it is important to observe that user preferences can evolve and change over time.

Additionally, this system could benefit from the addition of contextual data to the recommendation algorithm such as a user's emotional, situational and date-time context which could largely influence the resulting recommendations.

References

[1] https://www.yelp.co.uk/london

[2] Lü, L., Medo, M., Yeung, C.H., Zhang, Y.C., Zhang, Z.K. and Zhou, T., 2012. Recommender systems. *Physics reports*, *519*(1), pp.1-49.

[3] Vargas-Govea, B., González-Serna, G. and Ponce-Medellın, R., 2011. Effects of relevant contextual features in the performance of a restaurant recommender system. *ACM RecSys*, *11*(592), p.56.

[4] Li, Y. and Song, H., CS229 Final Project Predicting Yelp User's Rating Based on Previous Reviews.

[5] Subramanian, R.S. and Gnanasekar, S., Hybrid recommendation system to provide suggestions based on user reviews.

[6] Burke, R., 2007. Hybrid web recommender systems. *The adaptive web*, pp.377-408.

[7] Fakhri, A.A., Baizal, Z.K.A. and Setiawan, E.B., 2019, March. Restaurant Recommender System Using User-Based Collaborative Filtering Approach: A Case Study at Bandung Raya Region. In *Journal of Physics: Conference Series* (Vol. 1192, No. 1, p. 012023). IOP Publishing.

[8] Burke, R., 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, *12*(4), pp.331-370.

[9] Ghazanfar, M.A. and Prugel-Bennett, A., 2010, January. A scalable, accurate hybrid recommender system. In *2010 Third International Conference on Knowledge Discovery and Data Mining* (pp. 94-98). IEEE.

[10] A Restaurant Recommender System Based on User Preference and Location in Mobile Environment

[11] Su, X. and Khoshgoftaar, T.M., 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, *2009*.

[12] Goldberg, D., Nichols, D., Oki, B.M. and Terry, D., 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), pp.61-70.

[13] Lang, K., 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995* (pp. 331-339). Morgan Kaufmann.

[14] Lops, P., De Gemmis, M. and Semeraro, G., 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, pp.73-105.

[15] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J., 2001, April. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295).

[16] Shani, G. and Gunawardana, A., 2011. Evaluating recommendation systems. In *Recommender systems handbook* (pp. 257-297). Springer, Boston, MA.

[17] Kiran, R., Kumar, P. and Bhasker, B., 2020. DNNRec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications*, *144*, p.113054.

[18] Chikhaoui, B., Chiazzaro, M. and Wang, S., 2011, March. An improved hybrid recommender system by combining predictions. In *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications* (pp. 644-649). IEEE.

[19] Riyahi, M. and Sohrabi, M.K., 2020. Providing effective recommendations in discussion groups using a new hybrid recommender system based on implicit ratings and semantic similarity. *Electronic Commerce Research and Applications*, *40*, p.100938.

[20] Zhang, H.R., Min, F., He, X. and Xu, Y.Y., 2015. A hybrid recommender system based on user-recommender interaction. *Mathematical Problems in Engineering*, *2015*.

[21] Bradley, K. and Smyth, B., 2001. Improving recommendation diversity. In *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland* (Vol. 85, No. 94, pp. 141-152).