A Hybrid Restaurant Recommender System

Durham University

Monday 8th February, 2021

1 Introduction

1.1 Domain

Recommender systems use machine learning techniques to learn patterns in data and make predictions on items a user would like [1]. We investigate a hybrid of two recommendation techniques, collaborative filtering (CF) and context based filtering (CBF) in the domain of restaurant recommendations in the USA. We demonstrate empirically that the proposed method outperforms the separate techniques.

1.2 Related Work

Achakulvisut et al investigated text pre-processing for CBF as they worked to recommend scholarly articles [3]. So that related words are treated the same, stemming is suggested. For example, *computer* and *computation* would both be represented as *comput*. They demonstrated that these pre-processed keywords improved recommender performance significantly.

MyEat! is a restaurant recommendation mobile app that adds an element of context to its results [4]. Implementing a location-based service, candidate suggestions are limited to nearby stores, before CBF determines restaurant and user similarities. Not only does location filtering speed computation, but coverage and novelty of recommendations increase since results will be change across locations.

Wang et al. [5] successfully addressed data sparsity issues of CF by combining ratings of the same item by other users, ratings of the same user on different items and ratings of similar items by similar users.

It is therefore evident that a hybrid recommender system outperforms its individual components. It is particularly important to group methods that combat each others' downfalls.

1.3 Purpose

The aim of this current system is to combine content, collaborative filtering, geographical contextual filtering and specific user preferences to improve restaurant recommendations and mask issues of individual techniques as done in MyEat! [4]. The interface should be usable, processing transparent and results explainable.

2 Methods

2.1 Data Description

Data used is from the *Yelp!* dataset. It is divided into 5 JSON files, e.g. `business.json, user.json`. It covers 1.9M user profiles, 8M reviews and 209K businesses based in the USA. Businesses span across a wide range of categories - from restaurants to gyms.

2.2 Data Preparation and Feature Selection

We filtered businesses to restaurants since they are the most common type, responsible for 30% of the dataset and more data results in more informed, accurate recommendations. The review dataset was reduced to reviews for the restaurants, their ratings, the business id and user id in question. Only users with at least 3 reviews were used for collaborative filtering. Each business' attributes, categories and review keywords were combined into a "bag of words" column. Only common words across the dataset were retained. For instance, the word `irish` only appeared twice, so its instances were removed. As done in [3], words were converted to their "stem". Key words were then weighted using a frequency measure, described in section 2.4.

2.3 Hybrid Scheme and Recommendation techniques

The recommender system uses a Cascade methodology, where one technique selects immediate recommendations, and the second refines these, producing the final list [6]. The first stage uses CBF, where item attributes are represented as vectors and the similarity with other vectors are determined, using a metric such as tf-idf, term frequency inverse document frequency. We work assuming users will like items similar to those they liked before. Outputted is a list of the most related items which is refined by a collaborative filter. Collaborative filtering establishes user profiles and recommends items based on preferences of similar users. They can be memory-based, where stored ratings are used directly in the prediction, or model-based, where ratings are used to learn a predictive model. It does not require content, but is reliant on existing ratings. We estimate what the user in question would rate each input restaurant. Outputted is a ranked list of the restaurants with the greatest estimated rating. In addition to this cascade, a geographical context-based filter is used. The RS only analyses restaurants located in states of places they have rated before. This minimises computation whilst avoiding recommending an Italian in New York when the user lives in and only has history in California.

2.4 Evaluation methods

We evaluate the performance of our recommender system using offline experiments, observing four dimensions.

Accuracy of rating predictions with root mean squared error (RMSE). Predicted ratings $\hat{r}_{ui}$ are generated for a test set $\mathcal{J}$ of user-item pairs *(u, i)* for which the true ratings r_{ui} are known [10]. RMSE is calculated RMSE = $\sqrt{\frac{1}{|\mathcal{J}|}\sum_{(u,i)\in\mathcal{J}}\left(\hat{r}_{ui}-r_{ui}\right)^2}$

Accuracy of usage predictions with recall. Evaluates whether items predicted are used by the user. Selecting a test user, and hiding some of their selections, the ability of the recommender system to produce the hidden items is determined. The four outcomes can be expressed as a confusion matrix.

	Recommended	**Not recommended**
Used	True-Positive (tp)	False-Negative (fn)
Not used	False-Positive (fp)	True-Negative (tn)

Recall, also referred to as the true-positive rate is then calculated as $\frac{tp}{tp+fp}$. This typically improves with longer recommendation lists [10].

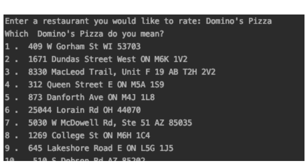
Catalog Coverage. Evaluates the proportion of items that the system can recommend. We calculate the percentage of all items that can ever be recommended. For instance, an item may not be suggested if it has no ratings. Catalog coverage is calculated as the observed recommendations of all users divided by the set of all available items, $\frac{|\cup_{j=1...N} I_L^j|}{|I|}$.

Explainability Since recommendations are broken down into estimated ratings, hybrid and pure collaborative recommendations, and this made clear with simple terminology to the user, they can combine this information to make an informed, effective decision on which to follow up. This consequently improves transparency. Separate explanations for the hybrid and collaborative results allow the explanation to accurately match the recommendation mechanism. On the other hand, a baseline collaborative system could explain how they were generated with this technique, but fewer stages compared to a cascade model result in less explanation and information for the user to make an informed decision upon.

3 Implementation

3.1 Input interface

The interface is simple and easy to follow, with simple controls, the majority of input being multiple choice. For every input, validation is incorporated with informative error messages. These make it is usable for all users. To recognise the active user, upon opening the system, the user can enter their user id or create one if they are new. There are several informative transparency messages, informing the data being stored and for which purposes. For example, a message is shown upon account creation, indicating the usage of their user information. In order to generate predictions, one must enter a restaurant name. If there is more than one with the same name, a multiple choice list of all the addresses are ouputted for the user to select from.



The user must input a rating (1-5) and have an option to write a review. They can also indicate preferences from a multi-choice list e.g. restaurants with wheelchair accessibility, which are then updated in the user database for future recommendations. Preferences affect the output presentation, allowing users to make effective informed decisions.

3.2 Recommendation algorithm

A user's rating and/or review acts as a start point, and is added to the dataset after pre-processing as described in section *2.2*. To take advantage of the user's context, the state of this business, along with all other businesses the user has previously rated is noted, and the potential restaurants to be recommended are filtered according to these. Terms in the "bag of words" column of the dataset are weighted according to the tf-idf, term frequency inverse document frequency feature selection metric. The advantage of this choice is that rare terms are not less relevant than those frequent, yet multiple occurences of a term doesn't make it less relevant than a single occurence. [7] For example, "food" may be a common word, but holds little discriminatory power. Once each item is represented as an equal-length vector in a feature vector space. Using KNN we determine the most similar restaurant to the input using cosine similarity between vectors. We use the following formula:

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k {w_{ki}}^2} \cdot \sqrt{\sum_k {w_{kj}}^2}}$$

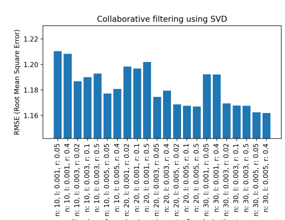
where d_i and d_j are two item vectors, and w_k is the weighting of word k. The top 100 most similar are fed as input to the collaborative filter. We use a probabilistic model-based filter, which uses ratings to learn a predictive model and transform items are users to the same latent factor space. Specifically, we use single value decomposition (SVD) where items are represented as vector q_i and users as p_u.The final rating is represented $\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$, where μ is the overall average rating, b_i and b_u are observed deviations of user u and item i from the average. $q_i^T p_u$ represents the dot product between item and user. In our implementation we minimise error using stochastic gradient descent, by calculating the prediction error for each rating and modifying parameters b_i, b_u, q_i and p_u in the opposite direction of the gradient [9]. This determines the given user's estimated ratings for all the inputted restaurants. The restaurants with the top five highest estimated ratings are presented to the user. For increased diversity of predictions, we also output the top three results of collaborative filtering applied to the pre-content filtered dataset.

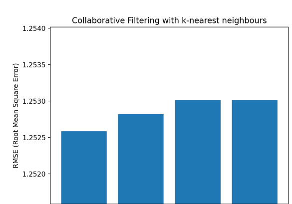
3.3 Output interface

Before entering any information, the output displays a message informing the user how inputted data will be stored and used for transparency. The recommendations are presented in a readable, clean table format, with no unnecessary information. User preferences in the recommendations are also highlighted. Wheelchair accessibility is a preference which would help those with disabilities. Finally, whether or not the restaurant is delivering during COVID, an important factor during the pandemic, is outputted. Simple explanations for the results are outputted. These all aid an informed decision for the user. Since none of the outputs are visual, a text reader would be able to share the information for those with visual impairments.

4 Evaluation results

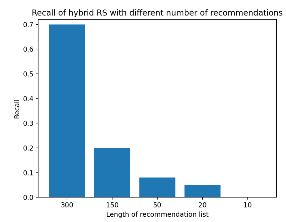
4.1 Comparison againt baseline implementation

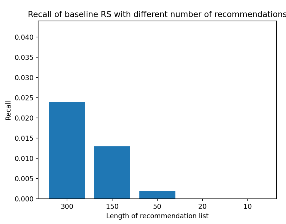




RMSE Our hybrid implementation uses a content-based recommender followed by SVD collaborative filtering. We test our hybrid implementation against a baseline basic k-NN collaborative recommender system, with a range of parameters to find an optimal result. 1.2526 was the best result using the baseline with k = 10, whereas the hybrid produced an optimal RMSE of 1.1604, a 7.4% decrease.

Recall As indicated in the graphs, recall was significantly higher using the hybrid over a SVD baseline collaborative system, with a recall of 0.68 recall with 300 recommendations, compared to 0.02. There were no true positives with a list of length 10 for the hybrid, whereas the baseline provided no true positives at a list length of 20. This could be because of the narrowing of results to a relevant geographical location in the hybrid. Users could also prefer restaurants with similar attributes, where content proves useful.





Coverage Our prediction coverage was the same for our hybrid and baseline. We used businesses with at least 3 ratings and reviews, because our collaborative made use of the ratings. This filtered 64k businesses to 12k, an 18% prediction coverage. With respect to catalog coverage, we did not have the computational power to process the coverage of our hybrid with all users. Instead, we used a subset of 50 randomly selected users for comparison purposes, and generated 15 recommendations each. 368 distinct businesses were suggested by the hybrid, but only 125 by the baseline. This is a significant 194% increase in coverage.

Explainability

4.2 Comparison againt hybrid recommenders in related studies

RMSE DNNRec is a hybrid recommender system that uses addresses the cold start problem in collaborative filtering by pairing it with deep learning, using embeddings to represent users and items and learn non-linear latent factors to fill in knowledge gaps. Testing on the MovieLens 100K dataset, their recall was 0.935, a 25% decrease from our 1.253 RMSE.

Recall Yang et al. [11] combined content and collaborative filtering in a hybrid job recommendation system. However, they incorporate costs for false postives and negatives so the trade-off between precision and recall are fine-tuned. Across tests they consistently achieved a recall of 1.0, a 47% increse from our 0.68 recall.

4.3 Ethical issues

There are potential ethical issues that can be associated with our recommender system. The dataset may be anonymised but there is still a significant risk of re-identification if data is leaked. Firstly, there is a real risk of sensitive data being leaked. Quasi identifiers, such as preferred restaurants and explicitly set preferences (e.g. wheelchair access) can be pieced together, even more so since first names are stored in the dataset. To combat this, data can be stored in a decentralised database to minimise the risk of a leak, and encrypted so it can not be interpreted by a third party.

There is a risk of the system recommending age-inappropriate food venues. For instance, recommending those under 21 in the US bars or other adult venues. To overcome this, it could be asked whether the user is of drinking age, and filter results accordingly. Adult venues can be manually labelled, or determined using attributes - although this risks being unreliable.

In addition, bias could be prevalent in the dataset. People who have contributed to the data could be from a certain demographic or hold a paticular political standpoint, making it less likely for recommendations to be made for restaurants typically attended by under-represented minorities. Although difficult, the effect of this can be reduced with increased transparency of where the predictions have come from, and asking the user for recommendation feedback, so to adapt more to the particular user in future.

5 Conclusion

5.1 Limitations and further developments

A clear limitation in our hybrid system is the potential bias within the dataset and its ratings, as discussed in section *4.3*. Although we attempt to provide transparency with messages such as "Users similar to you liked", further analysis could be done and more precise information could be provided as to how the users are similar to further explainability of the recommender system. In addition, the prediction coverage of our system is low, since businesses are only considered if they have been reviewed. This could be overcome in future by allowing unrated businesses to be content analysed solely off attributes. This could be extended further by then using embeddings and learning non-linear latent factors to further fill in sparsity for collaborative filtering as was done in [12]. Finally methods to learn about user's context, such as age or potential disability, could be used to refine recommendations further, although it comes as a tradeoff with the storing of more sensitive information.

References

[1] M. A. Ghazanfar and A. Prugel-Bennett, "A scalable, accurate hybrid recommender system," in *2010 Third International Conference on Knowledge Discovery and Data Mining*, pp. 94–98, IEEE, 2010.

[2] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*, pp. 325–341, Springer, 2007.

[3] T. Achakulvisut, D. E. Acuna, T. Ruangrong, and K. Kording, "Science concierge: A fast content-based recommendation system for scientific publications," *PloS one*, vol. 11, no. 7, p. e0158423, 2016.

[4] A. Gupta and K. Singh, "Location based personalized restaurant recommendation system for mobile environments," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 507–511, IEEE, 2013.

[5] J. Wang, A. P. De Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 501–508, 2006.

[6] A. S. Lampropoulos, P. S. Lampropoulou, and G. A. Tsihrintzis, "A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis," *Multimedia Tools and Applications*, vol. 59, no. 1, pp. 241–258, 2012.

[7] T. Joachims, "A probabilistic analysis of the rocchio algorithm with tfidf for text categorization.," tech. rep., Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[8] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[9] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, vol. 2007, pp. 5–8, 2007.

[10] G. Shani and A. Gunawardana, “Evaluating recommendation systems,” in *Recommender systems handbook*, pp. 257–297, Springer, 2011.

[11] S. Yang, M. Korayem, K. AlJadda, T. Grainger, and S. Natarajan, “Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive statistical relational learning approach,” *Knowledge-Based Systems*, vol. 136, pp. 37–45, 2017.

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