A Hybrid Restaurant Recommender System

mvlk94 - Durham University

I. Introduction

A. Domain

Recommender systems (RS) 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.

B. Related Work

Achakulvisut et al investigated text pre-processing for CBF as they worked to recommend scholarly articles [2]. 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 [3]. 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. [4] 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 therefore appears that a hybrid RS outperforms its individual components. It is particularly important to group methods that combat each others' downfalls.

C. 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! [3]. The interface should be usable, processing transparent and results explainable.

II. Methods

A. 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.

B. Data Preparation and Feature Selection

We filtered the dataset to entries relating to restaurants since they are the most common category, responsible for 30% of the dataset. More data results in more accurate recommendations. 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. This was filtered, retaning only common words across the dataset. For instance, the word `irish` only appeared twice, so its instances were removed. We set the cutoff at 10 occurences, since "Vegan", a common diet, appeared this many times. As done in [2], words were converted to their "stem". Key words were then weighted using a frequency measure, described in the following section. .

C. Hybrid Scheme and Recommendation techniques

Our RS uses a Cascade methodology, where one technique selects immediate recommendations, and the second refines these, producing a final list [5]. The first stage uses CBF, where item attributes are represented as vectors. These are produced from the business' "bag of words" using a metric such term frequency inverse document frequency (tfidf). The similarity between all of these are determined, using a method such k-Nearest-Neighbours (kNN). This assumes users will like items similar to those they liked before. Produced is a list of the most related items which is inputted into the CF. Here, user profiles are established and items are recommended based on preferences of similar users. Our CF is model-based, where ratings are used to learn a predictive model as opposed to where stored ratings are used directly in the prediction. It is reliant on existing ratings. We estimate what the user in question would rate each input restaurant. Our model used single value decomposition (SVD) to represent items and users in a latent space, and predict ideal parameters using stochastic gradient descent. We output a ranked list of restaurants with the highest estimated ratings. In addition to this cascade, an element of geographical context is exploited. The RS only analyses restaurants located in states of places they have rated before. This minimises computation whilst avoiding recommending an Pizza place in New York when the user lives in and only has history in California.

D. Evaluation methods

We evaluate the performance of our RS 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 [6]. We choose this over mean averge error

(MAE) because it gives disproportionately higher weightings to estimates further from the average. RMSE is calculated $\text{RMSE} = \sqrt{\frac{1}{|\mathcal{J}|}\sum_{(u,i)\in\mathcal{J}}\left(\hat{r}_{ui}-r_{ui}\right)^2}$

Accuracy of usage predictions with recall. Also known as the true-positive rate, this evaluates whether items predicted are used by the user. A selection of a test user's history is hidden and the RS's ability to reproduce these are 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)

From this, we calculate recall as $\frac{tp}{tp+fp}$. This typically improves with longer recommendation lists [6].

Catalog Coverage. This evaluates the proportion of items that the system can ever recommend. For instance, an item may not be suggested if it has no ratings. Prediction coverage is the set of items for which predictions can be made, divided by the set of available items, $\frac{|I_p|}{|I|}$. Catalog coverage is borne from this, and is calculated as the observed recommendations of all users divided by the set of all available items, $\frac{|\cup_{j=1...N} I_L^j|}{|I|}$ [6].

Explainability. This refers to the quality of explanations behind recommendations, judged against criteria. For example, do they help users make informed decisions effectively? And do they improve transparency of the RS? Finally, we look at whether they match the mechanisms used to generate the predictions? [7]

III. Implementation

A. Input interface

The interface is easy to follow, with simple controls and the majority of input being multiple choice. Validation is incorporated throughout with informative error messages. It is usable for users of all abilities. To recognise the active user, as soon as the system is opened, they enter their id or create one if they are new. Informative transparency messages are displayed at points of input, indicating the purpose of storing their data. 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 can write a review. Preferences can be declared from a multiple choice list e.g. family-friendly restaurants, which are then updated in the database for future recommendations. These preferences affect the output presentation, allowing users to make effective informed decisions.

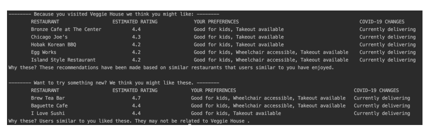
B. Recommendation algorithm implementation

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 I *B*. To take advantage of the user's context, restaurants are filtered according to the their visited states. Terms in the "bag of words" column of the dataset are weighted according to the tf-idf 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 [8]. For example, "food" may be a common word, but holds little discriminatory power. Each item is represented as an equal-length vector in a feature vector space. To use kNN we determine the most similar restaurant to the input using cosine similarity between vectors. This is calculated as

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

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 CF. We use a probabilistic model-based filter, specifically SVD to transform items and users to the same latent factor space. 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 and 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 an item and user.We minimise errors 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]. 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 CF applied to the pre-content filtered dataset.

C. Output interface

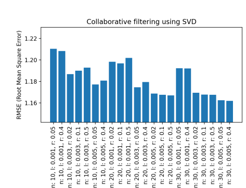


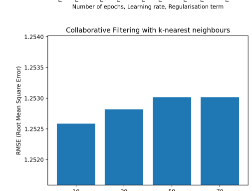
The recommendations are presented in a readable, clean table format, with no unnecessary information. User preferences in the recommendations are also highlighted. Wheelchair accessibility is a preference which would help those with disabilities. Additionally, whether or not the restaurant is delivering under COVID restrictions is outputted. Simple explanations for results are provided. 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.

IV. Evaluation results

A. Comparison againt baseline implementation

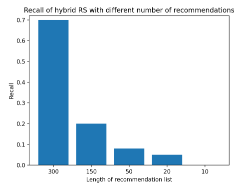
RMSE We test our hybrid implementation against a baseline basic k-NN collaborative RS. We test a range of parameters to find optimal results for both. The best RMSE scores produced were 1.2526 and 1.1604 for the baseline with k = 10, and hybrid respectively. Our system produces a 7.4% decrease.

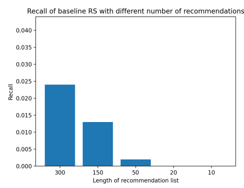




Recall As indicated by the graphs, our hybrid produced significantly higher recall than a baseline SVD CF system. Particularly, the hybrid produced a recall score of 0.68 with 300 recommendations, compared to 0.02. Additionally, the baseline produced no true positives with a list of length 20. This could result from users prefering restaurants with similar attributes, where content proves useful.

Coverage For our hybrid we focussed on businesses with at least 3 ratings and reviews, for improved collaborative results. 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, 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 Two sets of recommendations are made from the hybrid and pure CF methods, and are displayed with estimated ratings. Separate explanations for each technique's results are made to ensure they acccurately match the mechanisms. We use simple terminology so users can combine separate explanations and estimates to make an informed, effective decision on which to follow up. This consequently improves transparency. A baseline CF comprises fewer stages than a cascade model, resulting in fewer explanation for the user to make an informed decision upon.

B. Comparison againt hybrid recommenders in related studies

RMSE DNNRec is a hybrid recommender system that addresses the cold start problem in collaborative filtering through deep learning, using embeddings to represent users and items, and learns non-linear latent factors to fill in knowledge gaps. Testing on the MovieLens 100K dataset, their RMSE was 0.935, a 25% decrease from our 1.253 RMSE [11].

Recall Yang et al. [10] combined content and collaborative filtering in a hybrid job recommendation system. Particularly, 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.

C. Ethical issues

There are potential ethical issues that can be associated with our RS. The dataset may be anonymised but there is still a significant risk of re-identification if data is leaked. Quasi identifiers, such as preferred restaurants and explicitly set preferences (e.g. wheelchair access) can be pieced together. To combat this, data can be encrypted so it can not be interpreted by a third party. Additionally, there is a risk of the system suggesting age-inappropriate places. To overcome this, it could be asked whether the user is of drinking age and adult venues can be labelled, and results filtered accordingly. Finally, bias could be prevalent in the dataset. If those who contributed to the data were certain demographic, under-represented minorities are less likely to recieve accurate predictions. The effect of this can be reduced by asking for user feedback, and adapt the system accordingly.

V. Conclusion

A. Limitations and further developments

A clear limitation in our hybrid system is the potential bias within the dataset and its ratings, as discussed in the previous section. Although we attempt to provide transparency with messages such as "Users similar to you liked", further analysis could be done to provide more precise information e.g. how the users are similar. In addition, the prediction coverage of our system is low, since businesses are only considered if they have been reviewed. In future we could using embeddings and learn non-linear latent factors to further combat sparsity for collaborative filtering as was done in [11]. 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] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] S. Milano, M. Taddeo, and L. Floridi, "Recommender systems and their ethical challenges," *AI & SOCIETY*, vol. 35, no. 4, pp. 957–967, 2020.

[8] 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.

[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] 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.

[11] 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.