A Context Aware Restaurant Recommender System Using Content-Boosted Collaborative Filtering

Max Woolterton
Department of Computer Science
Durham University
Durham, United Kingdom
max.woolterton@durham.ac.uk

I. Introduction

This paper proposes a restaurant recommender system (RS) by building on the works of various authors, with the aim of aiding users in finding restaurants they will enjoy either in their local area or further afield. Collaborative filtering (CF) (Herlocker et al. [1]), content-based filtering (CB/CBF) and context aware (CA) (Zeng et al. [2]) approaches are all utilised and combined in a hybrid scheme (HS), similar to that introduced by Melville et al [3], which we name CACBCF. It will be apparent that RSs can become complex systems made of many and differing components, as is well demonstrated in the literature survey by Burke [4] which compares some 41 HSs. Such systems result in a myriad of ethical issues, as surveyed and investigated in the works of Milano et al. [5] and Germano et al. [6].

II. Methods

A. Data Description

The dataset used in this study is the Yelp Dataset, alongside Yelp's COVID-19 Data. The dataset includes anonymised user reviews of businesses in various sectors, such as restaurants, retail and professional services, across North America. These reviews include a description of the experience and a rating out of 5, together with the relevant user and business ids. The dataset also contains summarised details for each user and business, e.g. number of previous reviews, average star rating along with business tags and categories. For the geocoding services, MapQuest's Geocoding API is used.

B. Data Preparation

The dataset was first filtered so as to include only businesses containing the word "restaurant" in the pre-defined Yelp business tags. For this study restaurants were then narrowed down to those in Quebec, so that the whole training could be run on a modest machine. If a user had submitted multiple reviews for one restaurant only the most recent was retained. After examining the review count distributions, businesses and users with less than 5 reviews were filtered out to reduce the dimensionality and size of the dataset while avoiding the cold start problem and restricting the RS to restaurants for which a meaningful prediction could be produced. Finally, businesses with an average rating of below 3 stars were excluded to limit the recommender to restaurants of at least a mediocre quality. The pre-processing stage resulted in a final count of 4787 users, 1339 businesses and 56765 reviews, with an average review count of 12 per user and 42 per business. The resultant user rating matrix was 98.9% sparse.

C. Feature Selection

The features extracted from the dataset were the star rating, the user id and the business id from each review, along with business tags (e.g. food type) and specific relevant information about the businesses (i.e. business name, opening hours, address and coordinates) to aid users in their decisions. Subsequently, the business tags were pre-processed into a Bag of Words representation for each business using *term frequency-inverse document frequency* (TFIDF) vectorization. The constituent words of tags were chosen as tokens rather than the tags themselves in order to allow links between similar tags. A manual list of stop words was introduced to remove resultant duplicated and/or irrelevant tokens. Thereby a "business vector" was stored for each business to be later used for CBF. An (optional) additional feature used in our algorithm is the distance between the user and the restaurant (calculated using the haversine formula) so that local recommendations can be prioritised if desired.

D. Hybrid Scheme and Recommendation Techniques

In aid of constructing a HS the literature survey by Burke [4] proved invaluable. The schemes were narrowed down to those most appropriate to the dataset, namely those using CF and CBF, with, for example, those involving knowledge based techniques being discounted due to the deep domain knowledge and expertise required. A *feature augmentation* (FA) scheme, namely Melville et al.'s "content-boosted collaborative filtering" [3], was selected as a base for our hybrid, due to its impressive reported results and its ability to overcome rating sparsity and the cold start problem (thanks to its use of CB FA). In this scheme the primary recommender uses CF and the contributing recommender uses CBF. A CA component was added, which permits user-tuned novelty settings and facilitates consideration of COVID-19 requirements and the user's current travel preferences.

For CF a *neighbourhood-based algorithm* (NBA, Herlocker et al. [1]) was implemented on the recommendation of Melville et al. [3], and because of its superior performance when compared in early experimentation to *singular value decomposition* (SVD), one of the most popular model-based CF techniques. Namely, a memory-based user-user k-nearest neighbours (k-NN) algorithm, with the neighbourhood size k chosen to be 30 based on the studies of Herlocker et al. [1]. To summarise, to predict user i's ratings for all restaurants, users are first compared using a similarity metric and then a mean-adjusted weighted sum of the k most similar users' ratings is calculated. User-user was chosen over item-item due to the nature of the FA scheme, which, for example, would suffer in performance without the use of user-specific *hybrid cosine correlation weights* (3) (HCCW). Based on the extensive studies conducted by Liu et al. [7], the similarity metrics used for both NBAs were narrowed down to *adjusted cosine similarity* (ACOS) and *cosine similarity*, with experiments showing considerable performance boost over the *Pearson correlation coefficient* (PCC) used in Melville et al. [3].

Due to the quality of predictions of this HS being dependent upon the performance of the CBF, 2 algorithms were implemented and compared. Firstly, following Melville et al. [3], a model-based naive

Bayesian classifier, owing to its historically high performance, its low time complexity and the benefits it offers in the scalability of a RS. Secondly, another NBA was chosen, one designed for scalability, as is required for generating and maintaining a full pseudo-user ratings matrix (see Section III-B). The predicted rating of item i for a user u is calculated as a mean-adjusted weighted sum of the ratings of all items rated by u, with the weightings being equal to the cosine similarity between each item's business feature vector and that of i. The algorithm designed is effectively k-NN with k set to the number of ratings of each user. This choice is made to exploit vectorization and enable all items to be predicted at once.

E. Evaluation Methods

To evaluate the system an offline experiment was conducted with 4 metrics being used. Firstly, for accuracy and usage evaluation, the ratings dataset was randomly split into train and test subsets, with 80% being used to train the model and 20% being used for evaluation (ratings kept hidden from the recommender). These experiments were repeated across 3 seeds, with the results being averaged. For accuracy of rating predictions, predictions were generated for all ratings in the test set, and the root-mean-squared error (RMSE) was evaluated between these predictions and the test ratings. RMSE was selected due to its penalisation of larger errors as, given the domain, larger errors in recommendations have a greater impact on the user's dining experience than smaller errors. For accuracy of usage predictions both precision and recall were deemed relevant hence a weighted harmonic mean, the F-score was selected. Specifically, for ease of comparison with related studies, they were weighted equally with the F_1-score being used. The F_1-score was also evaluated over all items used (rated) in the test set, with a positive result defined as an item i being in the top n recommendations for the user u in question. The system's novelty was evaluated by the following formulae:

$$Novelty(R) = \frac{\sum_{i \in R} -log_2\ p(i)}{|R|}, \quad R = \bigcup_{u \in U} Recommend_n(u) \tag{1}$$

where U is the set of all users, $p(i)$ is the proportion of users who have rated restaurant i, and R is the union over the top n recommendations generated for each user u. In our experiments, $n = 10$ is chosen for both experiments, as this is thought to accurately approximate the choice of an average user. Novelty was chosen as a metric, as recommending niche and diverse experiences is an important quality in determining a user's satisfaction and therefore it is important for a RS not to simply tend towards recommending the most popular restaurants. Additionally, this causes diversity of recommendations to collapse via a butterfly effect, with the RS then causing a negative economic impact on all other restaurants, potentially resulting in a collapse in diversity within the industry itself (Germano et al. [6]). Finally, to evaluate the explainability of the RS, a user-centered approach has been adopted, with the criteria suggested by Milano et al. [5] being used to assess the quality of the factual explanations given, namely: "the purpose of the recommendation for the user; whether the explanation accurately matches the mechanism by which the recommendation is generated; whether it improves the system's transparency and scrutability; and whether it helps the user to make decisions more efficiently (e.g. more quickly), and more effectively, e.g. in terms of increased satisfaction."

III. Implementation

A. Input Interface

Users are asked either to login using their user id or register as a new user. On registration, the user is asked to rate some of their favourite restaurants in the area using the RS's search function for name or address. Whenever submitting a review, the user is always reminded that their review will be stored and reused for all future predictions for themselves and others, but their name never shown. The active user is then shown the main menu and asked whether they would like to submit a new review, retrieve recommendations or exit. When retrieving recommendations the user is asked for their input with regards to the following: number of restaurants to display, any takeaway/delivery COVID-19 driven restrictions, novelty preferences (new and old, only new or a preference for new over old ("weighted")), address and the maximum distance D they would like to travel (if they opt for local recommendations).

B. Recommendation Algorithm

In our HS ("Fig. 1") a sparse user ratings matrix is created from existing users' reviews and passed to the CB predictor along with the businesses' feature vectors, in order to generate a full "pseudo-user ratings matrix". This matrix consists of users' actual ratings along with the CB predictions for unrated businesses and is fed into the CF component with the active user's ratings to produce personalised predictions for all restaurants. Finally, these predictions are given to the CA recommender alongside the user's inputted preferences to filter and produce the final recommendations. CF predictions $P_{a,i}$ for an active user a for item i are produced via the following formulae:

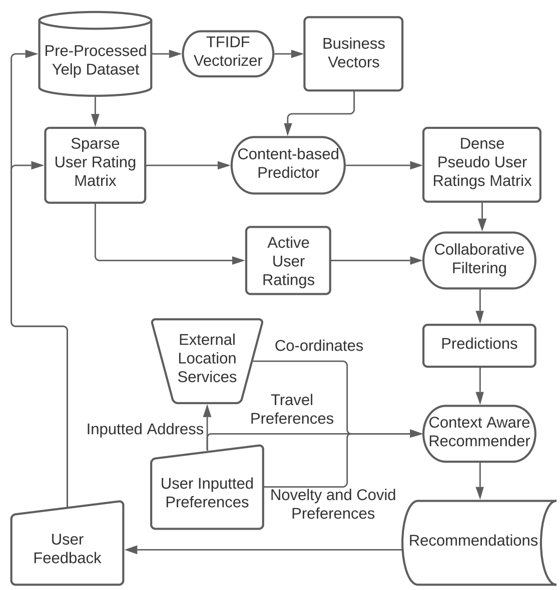

Fig. 1. Hybrid Scheme Framework

$$P_{a,i} = \bar{r}_a + \frac{\sum_{\tilde{u} \in \{N+\tilde{a}\}} (r_{\tilde{u},i} - \bar{r}_u) \cdot W_{a,\tilde{u}}}{\sum_{\tilde{u} \in \{N+\tilde{a}\}} W_{a,\tilde{u}}} \tag{2}$$

$$W_{a,\tilde{u}} = \begin{cases} ACOS(a,\tilde{u}) \cdot ((\frac{2m_a m_u}{m_a+m_u}) + m_{a\cap u}) & : \text{if } \tilde{u} \neq \tilde{a} \\ m_a \cdot max(\{W_{a,\tilde{\mu}} \mid \tilde{\mu} \in N\}) & : \text{if } \tilde{u} = \tilde{a} \end{cases} \tag{3}$$

$$ACOS(a,\tilde{u}) = \frac{\sum_{j \in R_a} (r_{a,j} - \bar{r}_a)(r_{\tilde{u},j} - \bar{r}_u)}{\sqrt{\sum_{j \in R_a} (r_{a,j} - \bar{r}_a)^2} \sqrt{\sum_{j \in R_a} (r_{\tilde{u},j} - \bar{r}_u)^2}} \tag{4}$$

$$m_u = \begin{cases} \frac{|R_u|}{100} & : \text{if } |R_u| < 100 \\ 1 & : \text{otherwise} \end{cases}, \qquad \bar{r}_u = \frac{\sum_{j \in R_u} r_{u,j}}{|R_u|} \tag{5}$$

$$m_{a\cap u} = \begin{cases} \frac{|R_a \cap R_u|}{50} & : \text{if } |R_a \cap R_u| < 50 \\ 1 & : \text{otherwise} \end{cases} \tag{6}$$

where neighbourhood N is the set of n (30) most similar pseudo-users $\tilde{u}$ to user a and R_u is the set of items j for which a user u has given a rating $r_{u,j}$. The pseudo-user $\tilde{a}$ is also considered and given a "self-weighting" in (3). The architecture of Melville et al. [3] was enhanced whereby to determine N active user a is compared rather than $\tilde{a}$ (using ACOS in lieu of PCC) and the maximum in (3), originally a static parameter (=2), was introduced to allow the parameter to dynamically adjust to the neighbourhood and similarity metric. The additional weights in the HCCW (3) are to devalue pairs with few co-rated items along with inaccurate CBF pseudo-vectors due to low user rating counts, when evaluating user similarities.

In the recommendation stage if a user opts for local recommendations, restaurants are restricted to those within the radius D selected and a vicinity weighting of $(1 - \frac{d_{u,i}}{D})$, where $d_{u,i}$ is the distance between u and i, similar to that of Zeng et al. [2], is added to the predictions for the sake of sorting, but not shown to the user. If "weighted" novelty mode is selected, before this stage, previously rated restaurants' predictions are discounted by 0.8 for sorting.

C. Output Interface

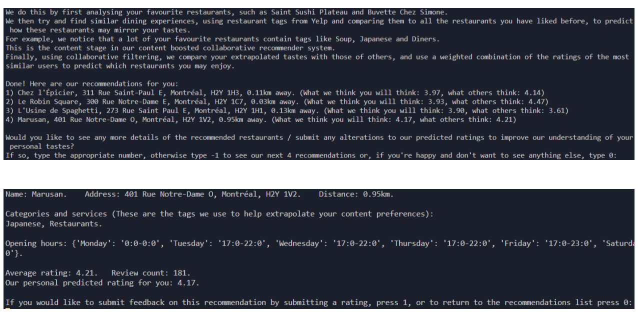

Fig. 2. Top: Main Output Interface, Bottom: "More Details"

Once recommendations are shown to the user, they are immediately able to offer feedback in the form of amending a prediction, which then has a live effect on the recommendations and predictions within seconds, as the RS is fed this new feedback.

IV. Evaluation Results

A. Comparison against Baseline Implementation

	HS-Bayes	HS-NBA	CBF-Bayes	CBF-NBA	CF
RMSE	1.01	<u>1.00</u>	1.18	1.12	**0.994**
F_1-score	0.0072	<u>0.0074</u>	0.0044	0.0046	**0.016**
Novelty	<u>6.00</u>	5.96	6.45	**6.46**	5.14

We compare 2 versions of CACBCF with the corresponding CBF-only baselines and an imputed (with user mean ratings) CF baseline. Along with taking into account content preference, tackling the sparsity, cold start and first rater problems for CF, CACBCF can be seen to combine the strengths of CBF and CF whilst mitigating the downsides of both (low accuracy/usage (CBF) and lack of novelty (CF)). The improved HS-NBA performs better in 2 of the 3 metrics than HS-Bayes, but does not scale as well, thus is only suitable for a smaller dataset.

The factual explanations given by our RS to users on presentation of recommendations of restaurants, which we expressly predict they will enjoy, accurately reflect the underlying mechanisms, while improving the transparency and scrutability of the RS, by giving an exact description, in simplified terms, of how these recommendations were produced along with examples of influential personal data utilised; for example, a user's favourite restaurants and some influential restaurant categories for CBF. Overall, the explanations given should help users identify enjoyable experiences more efficiently and effectively due to increased trust in and understanding of the recommendations.

B. Comparison against Hybrid Recommenders in Related Studies

	SVD	a	WBG	c	d	e
RMSE	1.57	1.53	1.49	1.15	1.24	**1.09**

In a related hybrid restaurant recommender study by Sawant and Pai [8] on the Yelp dataset, 5 schemes of increasing complexity and 2 baselines are compared, with RMSE values shown in the table above. Namely, "hybrid cascade of k-nearest neighbour clustering" (a), "weighted bi-partite graph projection" (WBG), "clustered WBG" (c), "multi-step random walk WBG" (d) and "cascaded clustered multi-step WBG" (e) schemes were implemented and compared, along with a SVD baseline. The schemes include a combination of knowledge-based, CB and CF techniques. Our scheme, also using CBF and CF techniques along with similar data preparation, can be seen to vastly outperform those proposed by Sawant and Pai [8].

C. Ethical Issues

There are many privacy risks related to both CACBCF and RSs in general (Milano et al. [5]), one such example is the risk arising from the storage of user information, e.g. data being leaked and/or attempts at user de-anonymization. Such a risk could be mitigated by storing user data in separate and decentralised databases. Furthermore, users' autonomy could also be at risk, by the RS "attempting to "addict" them to some types of contents, or by limiting the range of options to which they are exposed" (Milano et al. [5]). A potential mitigation would be to consider the serendipity and diversity of recommendations as additional metrics when designing and evaluating a RS. There is also no consideration of users' requirements on the grounds of religion or principle, e.g. halal, kosher, vegan. These requirements could easily be considered by the CA recommender.

V. Limitations and Further Developments

Despite having improved the results of Melville et al. [3], CACBCF remains the sum of its parts and so improving any component (TFIDF,CBF,CF,ACOS,CA) would result in further amelioration. For example, a custom-built similarity metric such as that in Liu et al. [7] could be considered. Including further features, such as the results of sentiment analysis on the reviews themselves, could prove valuable. The RS could be made time aware too, with lunch/dinner preferences differentiated and only open restaurants considered.

References

[1] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *ACM SIGIR Forum*, vol. 51, no. 2. ACM New York, NY, USA, 2017, pp. 227–234.

[2] J. Zeng, F. Li, H. Liu, J. Wen, and S. Hirokawa, "A restaurant recommender system based on user preference and location in mobile environment," in *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, 2016, pp. 55–60.

[3] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," *Aaai/iaai*, vol. 23, pp. 187–192, 2002.

[4] R. Burke, "Hybrid web recommender systems," *The adaptive web*, pp. 377–408, 2007.

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

[6] F. Germano, V. Gómez, and G. L. Mens, "The few-get-richer: a surprising consequence of popularity-based rankings," 2019.

[7] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowledge-Based Systems*, vol. 56, pp. 156–166, 2014.

[8] S. Sawant and G. Pai, "Yelp food recommendation system," 2013.