Recommender Systems: Summative Assignment

Department of Computer Science
Durham University

I. INTRODUCTION

A. Domain

Recommender systems (RS) are a type of information filtering technique which, given a profile of user preferences, aim to predict item ratings and output suggested items a user may like [L1]. Various techniques such as collaborative, content-based, knowledge-based, and context-aware recommendation are prevalent in the domain. Hybrid recommenders combine multiple RS techniques, aiming to improve recommendation performance and overcome limitations such as data sparsity and the cold-start problem.

B. Related Work Review

RS applications are relevant to a wide range of industries, including e-commerce, movies/TV shows, music, news, and restaurants. In developing the proposed system, various hybrid schemes and their constituent recommender algorithms were considered. Advantages and disadvantages of various hybrid schemes, as well as their implementation and performance analysis, were considered in light of discussion put forth by Burke et al [3]. Upon exploring the dataset, it was evident that matrix scarcity (i.e. limited user-item ratings) posed a significant limitation to collaborative filtering, both in terms of rating prediction and quality of output recommendations; most users left very few reviews, and those businesses which they did review comprised an extremely small sub-set of the entire dataset. A similar problem was addressed in Vall et al., where content-based filtering was used to supplement collaborative filtering performance in a music playlist hybrid recommender; despite "the vast majority of songs occurring only in very few playlists," it was observed that one can "gather rich song-level side information from … audio signal, text descriptions from social-tagging platforms, … listening logs… [to leverage] CF to infrequent songs" [2]. A similar approach was taken in the proposed system; recommendation algorithms and implementation details are further discussed in II.

C. Purpose/Aims

In this task, the aim was to design and develop a hybrid RS for a particular service category using business, review, and user data supplied in the Yelp dataset. The aim of the proposed system is to produce relevant, useful recommendations for its users; the resulting system should strike a balance between accuracy and novelty of recommendations. By implementing a hybrid recommendation scheme, it was intended that proposed recommendations would better suit a user's taste than those produced solely by collaborative filtering. Incorporating multiple recommendation techniques served to improve the diversity of suggested businesses, as well as take into account both past user behaviors and current user sentiment (i.e. considering a user's habitual/historical preferences vs. imminent state at the time of recommendation).

II. METHODS

A. Data Description

The Yelp dataset consists of business, review, and user data; in addition to pairs of (user, business) ratings and review content, each user and business entity is accompanied by a range of qualitative attributes and other characteristics that can be used to indicate preference (categorization, atmosphere, price range, accessibility, Wifi, parking, etc.) [7]. The dataset covers 10 major metropolitan areas and includes location data such as address, city, state, and latitude/longitude coordinates for each business. Business images, check-in history, and review sentiment (e.g. reactions, upvotes to individual reviews) are also included, however these were not utilized in the proposed system so as to retain a dataset of manageable size; business and review data were prioritized in the RS implementation.

B. Data Preparation, Feature Selection

The first step of the data preparation process involved cleaning the business.json file; this primarily included limiting the data to relevant business categories and geographical areas so as to produce a dataset of manageable size, yet sufficient for robust recommendation. In considering time frame for possible recommendations, all establishments that were now closed (i.e. out of business) were removed from the dataset. Further, businesses were limited to the 'Restaurant' and 'Bars' service categories; these categories are closely related as eating/drinking establishments, and were most widely represented in the complete dataset. Many qualitative attributes in the business.json file such as meal suitability (e.g. good for lunch, dinner), atmosphere (e.g. casual, romantic, trendy), and delivery/takeaway options were specific to food establishments; out of interest to retain this qualitative data and use it to construct additional features which could be used for recommendation, I opted to limit businesses in the dataset to the restaurant and bars. Additionally, unlike other large service sectors such as salons or home improvement businesses, it was noted that eating establishments generally fall under multiple categories; this can include generic categorization such as "Chinese" as well as niche labeling such as "Dim Sum" or "Wok". Such specification labeling proved useful when assembling a corpus of "bag-of-words" document representations for each restaurant; these terms were used to generate keyword vectors which were utilized in content-based filtering (discussed further in D). Boolean attributes such as 'BusinessParking' or 'Wifi' were one-hot encoded; particularly sparse attributes were dropped from the dataset altogether.

After limiting the dataset to businesses which matched 'Restaurants' or 'Bars', it was observed that many irrelevant categories remained (~460 total categories); categories deemed irrelevant to said establishments were subsequently dropped, leaving ~200 categories. To aid in keyword vector construction, the 'categories' column was expanded into individual columns, one for each category term present in the dataset. Columns for niche categories which were sparsely represented in the data were dropped; however, niche category terms remained in the bundled "categories" column to ensure use in construction of a vector space model. Lastly, categories sharing a common theme were grouped ito a newly engineered feature; e.g. categories representing different Japanese dishes (e.g. sushi, izakaya, ramen, etc.) were unified under a 'Japanese Food' attribute. This enabled knowledge-based filtering by a user's indicated cuisine preferences in the RS implementation. At this stage, the number of columns representing expanded categories totaled ~120.

Observing the Yelp Covid-19 data, it was noted that the primary indicator for a business' compliance with Covid-19 restrictions included availability for delivery or takeaway. This attribute was one-hot encoded to represent lack or presence of delivery options. The user is prompted to indicate whether they prefer eat-in or takeaway dining before recommendations are constructed. Once data preparation was completed for the business.json file, user reviews were mapped to businesses by performing an inner merge on business_id. Initially, only users with 2+ reviews were retained in the dataset; however, this resulted in an extremely sparse matrix (0.06%), thereby hindering collaborative filtering performance. Thus, only users with 10+ reviews for restaurants in the cleaned business data were retained; this resulted in an improved sparsity of 0.35%. At this stage, the cleaned dataset was 1 GB in size; though a significant reduction, this was still far too large to be easily malleable in a pandas dataframe. Thus, I proceeded to restrict the number of metropolitan areas represented in the business.json file; though initially all were included, only Ontario (the largest of the 10 groups) was ultimately considered for generating recommendations. The final dataset comprised 6419 users and 8113 businesses.

C. Hybrid Scheme

As discussed, implementing a hybridized scheme aimed to overcome limitations posed by the baseline collaborative filtering implementation; namely, rating matrix sparsity. Furthering considerations in Vall et al., a content-based filtering technique was used to supplement the baseline method; doing so facilitated use of the abundant restaurant feature data which is otherwise not taken into consideration when collaborative filtering alone is used. In constructing a hybrid scheme, various architectures put forth by Burke et al. were considered; ultimately, as the collaborative and content-based approaches used different in their nature of output (see D; done for purposes of overcoming user-item rating sparsity), a mixed hybrid model was selected. Each respective recommender first independently generates a list of candidates ranked by a corresponding scoring criteria; then, a linear combination of the output scores is used to generate an overall score for each candidate. Though a linear combination is used, construction of an overall "combined" score does not follow the weighted hybrid approach, as the two scores are independent; e.g. a single candidate does not pass through both recommenders before scores are averaged.

Next a cascade model is used; after being ranked by overall score, the output set of candidate restaurants proceed to the next stage of the recommendation process. The top-k candidates (selected k=12 of 18 initial candidates, s.t. the bottom third are excluded) continue to a third, knowledge-based recommender which compares candidate restaurant characteristics to user preferences indicated via questionnaire responses. Final output rankings of the candidate restaurants are tweaked depending on how closely they align with user questionnaire preferences.

D. Recommendation Techniques/Algorithms

As outlined in part C, collaborative, content-based, and knowledge-based recommendation techniques are utilized in the proposed RS. Implementation details of each algorithm are discussed in section III, part A. The below figure illustrates the overarching hybrid recommendation algorithm and its respective components.

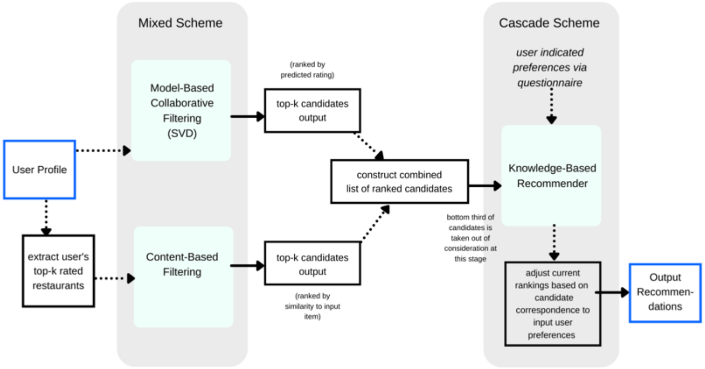

Mixed Scheme
Cascade Scheme
user indicated preferences via questionnaire
(ranked by predicted rating)
Model-Based Collaborative Filtering (SVD)
top-k candidates output
User Profile
construct combined list of ranked candidates
Knowledge-Based Recommender
bottom third of candidates is taken out of consideration at this stage
extract user's top-k rated restaurants
Content-Based Filtering
top-k candidates output
(ranked by similarity to input item)
adjust current rankings based on candidate correspondence to input user preferences
Output Recommen-dations

Fig. 1. Flowchart diagram of recommendation techniques employed in the proposed hybrid RS

E. Evaluation Methods and Results

Initially, both memory-based and model-based collaborative filtering were implemented; the 2 schemes were compared, and performance on test and training sets was evaluated. Ultimately, I proceeded to utilize the model-based method (SVD) as lower RMSE values were observed when quantifying performance; in contrast to a memory-based approach, model-based methods are generally better-suited to overcome sparsity problems often seen in large datasets [].

i. Accuracy of rating predictions

Hyperparameter tuning was conducted on the *k* parameter (indicating the number of matrix latent factors) for the model-based SVD implementation. Output RMSE for differing k = 10, 20, 50, 100, 200, 400, 600 were computed, where RMSE is calculated as in [1, W9 p.10]. k=50 was found to be optimal, returning a RMSE value of 3.68 . Given that the hybrid implementation incorporates a knowledge-based recommender at the third step, RMSE representing the complete system cannot be reported. I thus additionally considered rule coverage of the KB component, comparing the number of features addressed by KB constraints to the total number of qualitative attributes in the dataset [6]. Assessing the cleaned dataset, it is noted that there are a total of 38 engineered features (excluding one hot encodings for individual categories); in analyzing the querying structure of the KB recommender, 34 distinct features are utilized in rule constraints, suggesting a rules coverage of 34/38 = 0.894.

ii. Accuracy of usage predictions

Usage prediction accuracy is generally assessed via an offline experiment, where precision (TP/(TP+FP)) and recall (TP/(TP+FN)) are computed after visualizing output recommendations for a user whose known interactions with the dataset have been hidden [1, W9 p.12]. Again, because the implemented system incorporates KB filtering (which is dependent on the user's current sentiment and imminent responses), generic usage predictions cannot be computed; thus, interactions with the RS were undertaken to gauge **user perception** of the system [6]. For example, a sample analysis resulted in the RS output shown in Figure 2 (see further, *Output Interface*); at the input stage, it was indicated that the user (6 in dropdown list) was looking for a venue suitable for working at lunch time, and enjoyed Desserts, Ice Cream, and Cafes. It was observed that the output contains recommendations generated by both collaborative and content based approaches, and that the recommended restaurants do in fact correspond to the user's indicated preferences; the output includes 3 coffee shops which meet hasWifi criteria and 2 bakeries, which align with a preference for dessert foods at lunch time.

iii. Coverage

Next, item-space and user-space coverage were assessed by the considering the number of items for which predictions can be made / the set of all available items (or users, respectively) [1, W9 p.19]. Rather than considering the entire Yelp dataset (for which coverage would be minimal), I opted to consider only restaurants located within the Ontario metropolitan area for user-space coverage; as the minimum number of reviews per user in the dataset was restricted to 10+ and all recommended businesses require at least one review, naturally complete coverage could not be achieved.

Item-space crvge. (all metro areas): 8113/30469 restaurants = 0.27
Item-space crvge. (Ontario): 8113/8278 restaurants = 0.98
User-space crvge (Ontario, 10+ reviews/user): 6419/96,666 users = 0.07

For comparison, user-space coverage (Ontario, 2+ reviews/user): 42,538/96,666 users = 0.44. As quality recommendations were prioritized, I opted to require 10+ reviews/user so as to reduce sparsity from 0.06% to 0.35%.

iv. Explainability

Lastly, a user-centered approach was taken to assess RS explainability and transparency. A checklist of criteria for good explanation were matched against the system implementation; it was vital that explanations supplied to the user be useful and easy to understand, s.t. one can make efficient and effective decisions with the RS output. It was ensured that the RS' explainability features included 3 main components: type of recommendation technique used to generate candidate, numerical indicator (e.g. ranking score) used to select a candidate, and filtering criteria used to match a KB query presented in human-readable form. All of these criteria were fulfilled in the implemented system, as demonstrated in the output interface.

III. IMPLEMENTATION

A. Input Interface

The RS operates using a command-line interface (CLI). First, the one is prompted to select an active user for the recommendation; one can either select a user id from a predefined list (10 users randomly sampled from the dataset) or by typing a user id of their choosing. Next, the user is presented with a questionnaire, the responses to which are utilized by the knowledge-based recommender component to gauge current user sentiment and preferences. Questionnaire components include:

- Covid-19: delivery/take-away/drive-thru preferences
- Cuisine preferences: can select up to 3 from list
- Occasion: Work, casual meal, meeting with friends, family outing, date; appropriate features gauged for each
- Request for parking availability on-site
- Time-sensitivity: check current t.o.d. or allow manual selection; restaurants serving meals appropriate for that time of day are prioritized (e.g. morning ~ breakfast)

The CLI was designed to be simple to use for a range of potential users, regardless of technical ability or pre-existing knowledge of the data; all inputs require a single key press, consisting of y/n or a single digit number.

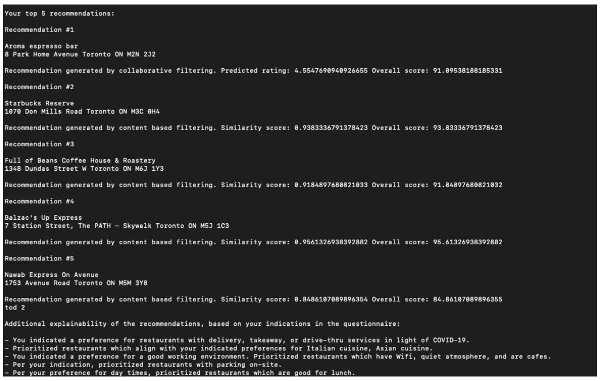
B. *Recommendation Algorithm*

To perform collaborative filtering, a user-item rating matrix is first constructed. Next, a model-based approach is adopted to compensate for sparsity issues encountered in the memory-based implementation; after normalizing the ratings matrix (so as to remove bias introduced by user tendencies to rate lower/higher than average), singular value decomposition (SVD) is used to generate a prediction matrix. For a given user id, the top-k (k set to 9) restaurants sorted by predicted rating are returned. Next, in content-based filtering, a keyword vector is constructed via "bag of words" approach, taking into account all qualitative restaurant attributes and category information. Features are weighted according to TF-IDF, after which cosine similarity is computed to gauge business' degree of similarity to one another. After examining user profile data supplied in the user.json file, I opted to implement keyword vectorization on the restaurant data as it was much more plentiful and contained features which intuitively correspond to indications of preference (e.g. matching a user to their past cuisine preference). Hence, for a given user_id, the top-k rated restaurants are extracted (selected k=3 to return 9 candidates), and the top 3 similar restaurants are returned for each. Rather than returning 9 candidates for a user's single top-rated restaurant, I opted to consider 3 candidates * 3 restaurants as the top rated restaurants may be fundamentally different in nature; e.g. a user may have rated a fancy restaurant highly for special occasions, as well as a fast food venue for casual meals – both are valuable indications of user preference and hence should be taken into consideration by the RS.

The output candidates of the collaborative and content-based recommenders are merged (total 18), and the top 12 ranked candidates (excluding the bottom third) proceed to the third knowledge-based recommender step. To rank the 18 candidates by an "overall score", a linear combination is used to quantify predicted ratings and similarity scores using a single metric. Lastly, the constraint-based KB recommender compares each of the candidates to received questionnaire input, and assesses the degree to which each restaurant fulfills the user's current sentiment. Despite introducing a slight information acquisition bottleneck by requiring users to answers a series of questions, the KB-component helps overcome the cold-start problem and introduces a notion of novelty to the system; rather than strictly on past user behavior history as an indicator of preference, the RS can also take into account what the user is feeling at the time of recommendation (e.g. maybe the user opts to try a new cuisine which they previously did not try or visit frequently). The candidate ranks are thus adjusted at the KB-step, with restaurants fulfilling more of the questionnaire criteria being ranked more highly. Of the remaining candidates, the top 5 are returned as recommendations to the user.

C. *Output Interface*

5 recommended restaurants are output to the user with corresponding prediction/similarity scores (depending on whether the recommendation was initially generated via collaborative or content-based filtering) and explainability regarding preferences indicated in the questionnaire. Output recommendations were capped at 5 so as not to overwhelm the user and disorient their selection of a restaurant to visit. Recommendations are displayed on the screen with adequate spacing, and relevant business information such as business name (as opposed to business_id, which is not human identifiable) and address are shown. The user is then prompted to receive new recommendations with the option to modify the active user, or to quit the program.



```
Your top 5 recommendations:

Recommendation #1

Aroma espresso bar
8 Park Home Avenue Toronto ON M2N 2J2

Recommendation generated by collaborative filtering. Predicted rating: 4.5547690940926655 Overall score: 91.09538188185331

Recommendation #2

Starbucks Reserve
1070 Don Mills Road Toronto ON M3C 0H4

Recommendation generated by content based filtering. Similarity score: 0.9383336791378423 Overall score: 93.83336791378423

Recommendation #3

Full of Beans Coffee House & Roastery
1348 Dundas Street W Toronto ON M6J 1Y3

Recommendation generated by content based filtering. Similarity score: 0.9184897680821033 Overall score: 91.84897680821032

Recommendation #4

Balzac's Up Express
7 Station Street, The PATH - Skywalk Toronto ON M5J 1C3

Recommendation generated by content based filtering. Similarity score: 0.9561326938392882 Overall score: 95.61326938392882

Recommendation #5

Nawab Express On Avenue
1753 Avenue Road Toronto ON M5M 3Y8

Recommendation generated by content based filtering. Similarity score: 0.8486107089896354 Overall score: 84.86107089896355
tod 2

Additional explainability of the recommendations, based on your indications in the questionnaire:

- You indicated a preference for restaurants with delivery, takeaway, or drive-thru services in light of COVID-19.
- Prioritized restaurants which align with your indicated preferences for Italian cuisine, Asian cuisine.
- You indicated a preference for a good working environment. Prioritized restaurants which have Wifi, quiet atmosphere, and are cafes.
- Per your indication, prioritized restaurants with parking on-site.
- Per your preference for day times, prioritized restaurants which are good for lunch.
```

Fig. 2. Sample output returned to a user who preferred bakeries, cafes, desserts and wanted a working environment at lunch-time

IV. FURTHER EVALUATION, ETHICS

A. *Baseline, Domain Comparison*

In addition to the evaluation metrics discussed in II (E), the proposed RS was compared to a baseline recommender (i.e. strictly collaborative filtering) and similar recommenders in literature. When implementing the collaborative filtering scheme, performance of both memory-based and model-based implementations was evaluated; when comparing performance based on predicted rating RSME, model-based significantly outperformed memory-based (3.87 vs ~14). As discussed in II (E), addition of a KB component did not allow for calculation of RMSE on the final hybrid implementation; however, through observing various use cases of the implemented system, it is evident that the hybrid RS presents a much more comprehensive set of recommendations. As evidenced by system output, content-based filtering is used to generate many of the top-ranking candidates in cases where matrix sparsity results in poor rating prediction by collaborative filtering. Further, observing that output recommendations often match closely with user questionnaire responses suggests that the constraint-based KB queries are a very effective means of adjusting final output rankings.

Looking to literature, it is difficult to establish a direct comparison as no research papers implementing a similar hybrid structure on the Yelp dataset were found. However, collaborative-content-based hybrids such as that put forth by Vall, et al. can be considered to gauge metrics such as RMSE, particularly with regards to the effect of matrix sparsity on recommendation [2]. Further, it is of interest to analyze system performance alongside that of other restaurant recommenders, considering both hybrid and traditional implementations; however, this has been omitted here for brevity.

B. *Ethical Issues* [4]

i. User Privacy

Firstly, users may be concerned about the use of their data, whether this be data existing in the dataset or collected during recommendation via the CLI questionnaire. It is unclear to the users whether questionnaire data is stored or discarded after use, and, if stored, whether it is subject to further processing which may infringe on a user's privacy. For example, data may be used to interpret user lifestyle patterns from their past behaviors and reviewed businesses; e.g. restaurant price point may suggest one's financial status, types of businesses visited may suggest habitual routines. Personal information such as the user's name is also deanonymized in the dataset. This concern can be mitigated by serving users a disclaimer about data usage before they interact with the RS, or otherwise stating that any stored personal data is anonymized or altogether deleted.

ii. User Autonomy and Personal Identity

The KB component of the recommender makes a series of assumptions about the meaning of a user's response; e.g. what constitutes a user's ideal "dining out with friends" may not correspond to that implied by the imposed KB constraints. One might argue that the system may encroach on user autonomy by using overly generalized preference indicators, thereby limiting the range and "uniqueness" of restaurants to which a user may be exposed. Further, restaurants with more ratings will naturally be more prominently featured as users are more likely to be deemed similar to their raters; this can "run the risk of insulating users," introducing self-reinforcing biases and "filter bubbles" (i.e. a herd mentality approach) which favor towards recommending the same businesses for many users [4]. Some users may only be inclined to leave reviews when their experience was particularly good or bad, hence skewing a business' overall rating and their "true" user profile. This can be mitigated by implementing a more fine-grained questionnaire and corresponding KB component (e.g. prompting a user to type in their own words rather than select from a dropdown) s.t. truly custom recommendations are queried. Perhaps introducing NLP or deep learning methods may help further refine user profiles and consequently better tailor output recommendations.

iii. Business/Provider Interests

Lastly there is concern that Yelp or a listed business itself may alter items' performance within the RS algorithm by modifying individual listings to match recommendation parameters; e.g. what a business categorizes itself as directly impacts the output of content and knowledge-based components. More generally, businesses with more complete profiles will perform better by virtue of supplied information; a business which closely meets a user's preference criteria may not be recommended simply because of missing or incorrect information listed on its Yelp profile. Further, businesses may be giving users incentives (e.g. discounts) in return for a positive review, thereby increasing their number reviews from unique users and introducing bias to rating scores. It is difficult to mitigate this concern without fundamentally altering the RS algorithm (e.g. diversifying the types of queries made by the KB component) or standardizing the contents of a Yelp business profile (e.g. requiring all businesses to have a fully completed profile before their listing is released for users to review and included in the RS).

V. CONCLUSION

A. Limitations

Key limitations included data sparsity (particularly with regards to (user, item) ratings) and dataset size handled by the RS. The sparse ratings matrix hindered collaborative-filtering performance, particularly when a memory-based implementation was considered; additional ratings per user would improve overall performance and prediction accuracy metrics such as RMSE. Additionally, as recommendations were to be issued by the CLI in a timely manner, the size of the dataset was restricted; this was also done to fulfill upload restrictions upon submission. Using a larger dataset would enable performing recommendations in other businesses categories and/or metro areas.

B. Further Developments

As discussed, a more robust implementation could incorporate a larger subset of Yelp data so as to expand to more business categories or metro areas. One might consider introducing additional recommenders or new recommendation techniques such as NLP on the review text itself (e.g. sentiment analysis, examining repeated phrases) or deep learning algorithms. Additionally, one can extend the system to accept new user ratings directly through the CLI, s.t. the active user can rate a new business and immediately update their user profile used for recommendation.

REFERENCES

[1] Hadzidedic, Suncica. Recommender Systems lecture notes, weeks 1-10.

[2] Vall, A., Dorfer, M., Eghbal-zadeh, H. *et al.* Feature-combination hybrid recommender systems for automated music playlist continuation. *User Model User-Adap Inter* 29, 527–572 (2019).

[3] Burke, R. (2007). Hybrid web recommender systems. In The adaptive web (pp. 377-408). Springer, Berlin, Heidelberg.

[4] Milano, S., Taddeo, M. & Floridi, L. Recommender systems and their ethical challenges. *AI & Soc* 35, 957–967 (2020).

[5] Garrido, Angel & Pera, Maria & Ilarri, Sergio. (2014). SOLE-R: A Semantic and Linguistic Approach for Book Recommendations. Proceedings - IEEE 14th International Conference on Advanced Learning Technologies, ICALT 2014. 10.1109/ICALT.2014.155.

[6] Burke, Robin. Knowledge-based recommender systems. University of California, Irvine. https://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day6/burke-elis00.pdf

[7] Yelp Dataset, https://www.yelp.com/dataset.