Designing and Developing a Personalised Recommender System

wcrr51
Department of Computer Science
Durham University
Durham, United Kingdom
wcrr51@durham.ac.uk

I. Introduction

A. Domain of Application

The chosen domain of application for this recommender system is restaurants.

One place the restaurant domain differs from a domain involving a selection of products is that of item accessibility. It is assumed that any recommendable product, if chosen, will be readily available - if a product is not available in an area, either the user is not recommended it, recommended an alternative, or provided with an alternative means of access.

Conversely, willingness to go to a restaurant is far more dependent on location, particularly when recommendations are desired - if someone wanted to travel a non-trivial distance, it is most likely they already know what they want. Given this, it is proposed the recommender system take into account the proximity of users to restaurants.

B. Review of Related Work

[1] presents a hybrid recommender using the Yelp dataset with content-based and collaborative filtering. It implements, compares and evaluates several learning algorithms when applied to find both user and item profiles.

[2] demonstrates a non-hybrid CF recommender which uses the Yelp dataset. It uses matrix factorisation to learn a set of latent factors contributing to rating scores.

C. Motivations and Aim

[3] has shown that an increased variety of available products tend to negatively affect sale conversion rates. A key reason is that being presented with lots of options to pick from can be overwhelming, especially when the user does not know exactly what they want. Hence, while many restaurants of differing varieties are available, it is better to present users with a subset of these.

Recommender systems aim to determine which subset of these products should be shown to the user, looking for the highest likelihood of the user purchasing them. This rationale also applies to restaurants - especially in an age of deliveries and takeaways due to the pandemic. The ultimate goal is to provide relevant personalised recommendations to known-users.

II. Methods

A. Data Description

The Yelp dataset [4] spans between 2004–2019 and covers a wide number of businesses across many sectors. While the majority of the data is for US metropolitan areas, it also covers parts of Canada.

It consists of data tables where each record is a single JSON object with attributes representing the respective item. These tables/item-types include businesses, users, reviews, tips, check-ins and Covid-19-related data. From here, only the business, user, review, and Covid-19 data will be considered.

The dataset does not include a business unless it has had at least three reviews and only includes reviews that have flagged for recommendation by Yelp. For the 36,370 sampled restaurants, there was a mean of 110, median of 40 and standard deviation of 248 sampled reviews. For the 1,259,528 sampled users, there was a mean of 3, median of 1 and standard deviation of 9 sampled reviews.

B. Data Preparation and Feature Selection

Where users have reviewed a business more than once, their most recent review is considered. Intuitively, this makes sense as the most up-to-date opinion of the user is likely to be the most desirable.

Data exploration showed that the median number of sampled reviews per restaurant is 40, and per user is 1. Compared to the product of the number of sampled businesses and users, the number of sampled reviews is very sparse. To overcome this, placing a lower-limit on the number of reviews at 40 per restaurant provides a good compromise between sparsity while still using all samples in the upper two quartiles. Similarly, a lower-limit of 4 will be placed on the number of reviews per user, thus retaining the majority of samples in the 4^{th} quartile. This reduces the number of users and businesses with minimal impact on the number of reviews benefiting in a performance increase.

Recommendations can make use of many business features. The ones used here are: name, coordinates, categories, whether delivery/takeaway is offered (from Covid-19 data). The name and categories of restaurants have a large correlation with the type of food they offer, making this a useful feature for similarity. The location is very important for providing nearby recommendations.

C. Recommendation Techniques

The two main components of the recommender system are collaborative filtering (CF) and content-based filtering (CBF), both of which are defined in [5].

For memory-based collaborative filtering, k-nearest neighbours (kNN) with the cosine similarity metric [6] is used on the sparse user-item rating matrix at runtime. For input k, matrix M of size $m \times n$ and vector v of length m, kNN finds the k positions in M whose vectors (length m) have the greatest similarity to v.

By nature, reviews alone do not necessarily paint the entire picture. For example, a user may give a restaurant a negative review because they were put off by one particular thing that no one else would consider. Content-based filtering aims to help counteract/dilute this by basing recommendations and restaurant features. This is achieved by constructing an item-item similarity matrix based on their features and recommending items based on this.

D. Hybrid Scheme

Weighted hybridisation combines the candidate recommendations. The similarity scores for the candidate recommendations provided by CF and CBF are each multiplied by a scalar quantity.

An example of why this is useful is that if both CF and CBF were weighted equally, the recommendations would almost always be dominated by restaurant chains as they tend to have similar/the same features (except for location) and thus get a high similarity rating from CBF. Because of this, a weighting of 0.7 is used for CBF and 12.0 is used for CF.

E. Evaluation Methods

Root mean squared error (RMSE) as defined in [7] is a metric for testing accuracy of rating predictions and will be applied on CF and the hybrid system independently. Knowing this is useful because it helps identify whether any advantage is being gained from using the hybrid model.

Item-space prediction coverage [8] is a valuable metric given the high sparsity of the review data. The definition used here for I_p will be the set of restaurants that can be recommended given the thresholding which turns out to be $18,213$. This gives a prediction coverage of $\frac{18,213}{36,370} \approx 0.5 \approx 50\%$.

For the domain of this recommender system, novelty [9] evaluation is not as applicable. While it is not necessarily desirable for a user to visit a new restaurant each time they go, the recommender system still filters out candidate recommendations that the user has previously positively rated.

To evaluate explainability a list of criteria is required. The relevant ones for this evaluation include system usability, whether the system matches the motivations and aim, and whether the user is informed of how recommendations are generated.

III. Implementation

A. Input Interface

The input command-line interface (CLI) is a state-based system. A pre-set list of commands with specified arguments are used for data input, which in turn updates the internal state. For example, when a user runs the `userid <id>` command, the user data corresponding to the provided ID is fetched and stored. In addition to being shown on start, a `privacy` command is provided to give the user information about the data and how it's being used. Since both the CBF and CF recommenders have their models trained on load, it is not possible to update user profiles. It would not be a difficult adjustment to enable such changes and re-train the models in real-time.

B. Recommendation Algorithm

The system is provided with the selected user ID and desired recommendation count (n). Previously reviewed restaurants are identified and calls down to the CF and CBF recommenders to retrieve the n most similar candidates per reviewed restaurant. These are pooled together and have their similarities normalised. They are sorted by similarity and the n most similar candidates are returned as the actual recommendations.

The CBF recommender uses the `TfidfVectorizer` to pre-calculate phrase features and combines a resultant item-item similarity matrix with the similarity matrix from the other features. For rating predictions, CBF finds the k nearest items to that being rated and returns them.

The CF recommender uses a sparse user-item matrix and generates an item-item cosine similarity matrix to provide fast kNN lookup. CF uses a similar algorithm to CBF for rating prediction except that it uses kNN on user similarity.

C. Output Interface

Restaurant recommendations are provided in a user-friendly table-like format and consist of the recommendation number (rank), restaurant name and location, score, predicted star rating and whether the restaurant offers delivery or takeaway. As part of the Covid-19 adjustments there is an option for displaying restaurants with takeaway or delivery services.

During loading, RAM usage spikes between 7–9 gigabytes, decreasing when temporary calculation data is freed up.

IV. Evaluation

A. Comparison against Baseline

The RMSE for rating predictions through random review sampling stabilised at around 0.85 stars for CF and 0.92 for the hybrid system. This means the hybrid was worse at rating prediction than CF alone. This is most likely from CBF overfit causing unrealisitc similarity values.

As past reviews are excluded from results, novelty was not specifically measured. However, though testing of the hybrid system there are many cases that CBF overrules CF and recommends clumps of chain restaurants, which, at a higher level of abstraction, would lower novelty. Interestingly this did not occur in CF alone.

Finally, for both CF and the hybrid system the coverage remained constant at approximately 50%. These results make sense since the subset of restaurants that can be recommended by both CF and CBF are the same.

B. Comparison against Related Hybrid Recommenders

[1] also uses the Yelp dataset. Combined with the fact it is a hybrid using CBF and CF make it a valuable comparison to this system. Their best model achieves nearly half the RMSE of their naive baseline at about 1.1 stars. Given they were using more advanced ML methods it would appear this recommender has over-fit the rating data.

[10] uses review text and ratings to produce impressive RMSE values (around 0.6 stars) by combining between 4 and 5 sub-models with a linear combination hybrid scheme. Their memory-based CF was able to predict ratings with an RMSE of about 1.1 but over-fit when changing to a model-based CF.

C. Ethical Issues

1) Inferring Private Information: For example, health-based data knowledge and recommendations. It is directly related to the tradeoff of personalisation vs privacy. Users with specific dietary requirements are inclined to visit restaurants who can accommodate these. Apart from being explicitly mentioned in reviews, a machine learning model may be able to infer it from patterns in reviews and ratings, thus giving the system characteristics of a health recommender system [11]. This could lead to private medical conditions being deduced (as in [12], [13]) and at worst result in targeted advertising. To mitigate this, users could opt-out of having diet-related features influence recommendations.

2) Fairness to Businesses: Particularly during a pandemic, when the internet is relied on for recommendations, a business can be made or broken by the exposure it receives. Those in control of how much exposure a business has, could be responsible for its success or failure. In this system, this issue arises during pre-processing when restaurants with a low number of reviews are removed.

To mitigate this, care must be taken not to introduce bias that may unfairly exclude businesses, particularly small ones. This may take the form of having a random or periodic quota for recommending businesses based more on CBF than CF or filtering out the most popular recommendations. Alternatively, the user could be asked if they would try out a smaller, less rated, related business.

3) Inadvertent Discrimination: In machine learning, discrimination is a large ethical topic. Users could inadvertently end up grouped together by protected characteristics if the underlying ML models learn from such features [14]. This may place users in stereotypes or trap them in *echo chambers*, preventing them from discovering a more diverse array of restaurants. For restaurants, it may reduce the diversity of their customer base.

A mitigation for this would be trying to ensure the features in use remain as neutral as possible. As with the previous issue, this could be mitigated by occasionally introducing solely content-based recommendations.

V. Conclusion

A. Limitations

While measures have been taken to reduce the sparsity of the data, it is still a significant problem. For n users, m restaurants, and R reviews, the rating matrix of size (n, m) has a density of $\rho = \frac{R}{nm}$. This means that the preprocessed dataset (of $n \approx 2.1 \times 10^5$ users, $m \approx 1.8 \times 10^4$ restaurants, and $R \approx 2.3 \times 10^6$ reviews) has density of $\rho \approx 6.0 \times 10^{-4} \approx 0.06\%$ (this is six times the original sampled density).

B. Further Developments

1. Investigate the incorporation of the *tip* and *check-in* data in CBF.

2. Make use of the friends list provided for each user would allow a friend graph to be constructed to help determine trustworthiness and reliability of users based on that of their friends.

3. Improve recommendations and overcome the issue of sparsity, pass the filtering models over to cutting edge deep-learning models (as assessed in [15]).

4. Expand the domain to include more of the businesses in the Yelp dataset, as a way to test generalisation with minimal effect on recommendations for the original domain.

References

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

[2] M. C. De Paolis Kaluza, "Recommender system for yelp dataset," 2016, unpublished.

[3] A. Chernev, U. Böckenholt, and J. Goodman, "Choice overload: A conceptual review and meta-analysis," *Journal of Consumer Psychology*, vol. 25, no. 2, pp. 333–358, 2015.

[4] Yelp Inc., "Yelp Dataset," 2020, data retrieved from https://www.yelp.com/dataset.

[5] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

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

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

[8] M. Ge, C. Delgado-Battenfeld, and D. Jannach, "Beyond accuracy: evaluating recommender systems by coverage and serendipity," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 257–260.

[9] M. Kaminskas and D. Bridge, "Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 7, no. 1, pp. 1–42, 2016.

[10] Y. Li and H. Song, "Predicting yelp user's rating based on previous reviews," 2016, unpublished.

[11] M. Wiesner and D. Pfeifer, "Health recommender systems: concepts, requirements, technical basics and challenges," *International journal of environmental research and public health*, vol. 11, no. 3, pp. 2580–2607, 2014.

[12] A. Gottlieb, G. Y. Stein, E. Ruppin, R. B. Altman, and R. Sharan, "A method for inferring medical diagnoses from patient similarities," *BMC medicine*, vol. 11, no. 1, pp. 1–10, 2013.

[13] R. M. Merchant, D. A. Asch, P. Crutchley, L. H. Ungar, S. C. Guntuku, J. C. Eichstaedt, S. Hill, K. Padrez, R. J. Smith, and H. A. Schwartz, "Evaluating the predictability of medical conditions from social media posts," *PloS one*, vol. 14, no. 6, p. e0215476, 2019.

[14] B. Abdollahi and O. Nasraoui, "Transparency in fair machine learning: the case of explainable recommender systems," in *Human and machine learning*. Springer, 2018, pp. 21–35.

[15] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.