Item-Based Vs User-Based Collaborative Recommendation Predictions

Joel Azzopardi(✉)

University of Malta, Msida, Malta
joel.azzopardi@um.edu.mt

Abstract. The use of personalised recommendation systems to push interesting items to users has become a necessity in the digital world that contains overwhelming amounts of information. One of the most effective ways to achieve this is by considering the opinions of other similar users – i.e. through collaborative techniques. In this paper, we compare the performance of *item-based* and *user-based* recommendation algorithms as well as propose an *ensemble* that combines both systems. We investigate the effect of applying LSA, as well as varying the neighbourhood size on the different algorithms. Finally, we experiment with the inclusion of content-type information in our recommender systems. We find that the most effective system is the *ensemble* system that uses LSA.

1 Introduction

Recommendation systems counter the information overload problem by 'pushing' relevant information to the user. One of the most effective recommendation approaches is *Collaborative Filtering* – whereby recommendations are issued based on the opinion of other similar users. Collaborative Filtering Methods are typically divided into: *user-based* methods whereby item recommendations are calculated depending on ratings by similar users; and *item-based* methods where the ratings for an item are predicted based on how similar items have been rated [10,11,13]. *Item-based* methods reportedly achieve better performance [9–11,13].

In this research, we use the MovieLens-1M dataset [7] to compare the performance of *user-based* and *item-based* approaches. We also analyse the performance of an *ensemble* of both such approaches – something that we have not encountered in literature. In addition, we use *Latent Semantic Analyses* (LSA) techniques as proposed in [5] – LSA is typically used in content-based systems, but we apply it to collaborative recommender systems. We also investigate the effect of incorporating content-based features (namely movie type information) in our collaborative recommender systems.

2 Related Work

Recommendation systems are typically divided into: *content-based* recommendations whereby recommendations are issued on the basis of similarity between

© Springer International Publishing AG, part of Springer Nature 2018
J. Szymański and Y. Velegrakis (Eds.): IKC 2017, LNCS 10546, pp. 165–170, 2018.
https://doi.org/10.1007/978-3-319-74497-1_16

the items' content and the user model; and *collaborative* recommendations where a user-item rating is predicted based on the ratings provided by similar users, or by the user on similar items [4,12,13].

Content-based approaches are able to calculate recommendations for items that do not have ratings, or that have a high-turn-over – in fact, a popular use for them is online news recommendation [2,8]. Such systems need to perform content analysis [11], and textual content is most generally represented using the *Bag-of-Words* model [4,8]. Keyword-based representations may be further extended using techniques such as *Latent Semantic Analysis* (LSA) [1,5,8].

Collaborative approaches have been found to be more effective than content-based approaches [1,9,10]. Their advantages include simplicity, and increased likelihood of including novelty and serendipity in their recommendations [3,9,11]. On the other hand, they suffer from: *sparsity* – i.e. they require substantial overlap in ratings; and the *cold-start problem* – recommendations for new users or items can not be issued if they have no ratings [3,4,9,11,12]. As described in Sect. 1, such approaches are further subdivided into *User-based* and *Item-based* methods [10,11,13] and *Item-based* methods are found to achieve better results [9–11,13]. However, the evaluation of *ensemble* methods combining both methods seems to have been rarely performed in previous research.

An important aspect underlying collaborative recommendation performance is the neighbourhood size. Prediction accuracy increases in proportion to the neighbourhood size [12]. [9] reports that accuracy improves as the neighbourhood size approaches 30, but then remains quasi uniform for further increments.

Problems faced by collaborative systems can be partially relieved by incorporating content-based features when calculating similarities [4,9,12]. While [9] and [4] report improved results when incorporating content features, the incorporation of content features in [6] caused a deterioration of results. [11] states that there is no consensus on how such techniques should be combined, and suggests a number of different hybrid types. When comparing different hybrid types, [3] found that *cascade recommendation* (i.e. a priority is given to each recommender, and the lower-priority ones are used to break ties) resulted to be the most effective hybrid technique.

3 Methodology

In this research, we compare the performance of *user-based* and *item-based* collaborative recommenders. We analyse the effect of LSA – LSA has been very rarely applied in collaborative scenarios. We also investigate the effectiveness of an *ensemble* combining both *user-based* and *item-based* approaches. Additionally, we identify the best neighbourhood sizes to use. Finally, we also experiment with the incorporation of movie-type information in collaborative recommenders.

3.1 Collaborative Recommendation Algorithms

The *user-based* recommendation algorithm is based on the *kNN* algorithm whereby each 'vote' is weighted by the similarity of the other user to the current

user. The neighbourhood consists of the k most similar users that have also rated the item under consideration. The algorithm is shown below:

```
predictRating-SimUsers (UserSimMatrix, UserID, ItemID, k)
  CandidateRatings ← φ
  SimUsers ← getSimilarUsers (UserSimMatrix, UserID)
  curk ← 0

  while (curk < k)
    user ← getNextMostSimilarUser (SimUsers)
    SimUserRating ← getUserItemRating (user, ItemID)

    if (exists(SimUserRating))
      updateCandidateRatings (CandidateRatings, SimUserRating,
                                          Similarity(user, UserID))
      k ← k + 1
    end if
  end while

  return (getHighestWeightedCandidate (CandidateRatings))
end
```

The parameter $UserSimMatrix$ refers to a matrix containing the similarity between each pair of users. This matrix is constructed by representing each user as a vector of ratings, and using Pearson similarity to calculate each pairwise similarity. In the LSA variant of this algorithm, the user-by-item ratings matrix is decomposed using SVD, and only the top-most dimensions are used to construct the pair-wise user similarity matrix.

The *item-based* recommendation algorithm is equivalent to the *user-based* recommendation algorithm, where instead of user pair-wise similarity, item pair-wise similarity is used. Predicted ratings are calculated based on how the current user rated the items that are most similar to the item in question.

The ensemble algorithm uses both the user pair-wise similarity matrix, and the item pair-wise similarity matrix. It obtains separate candidate recommendations lists ($CandidateRatings$) from both the *user-based* and the *item-based* recommendation algorithms described previously. The predicted recommendation is set to highest weighted candidate after merging both lists together.

3.2 Incorporation of Content-Type Information

The *MovieLens 1M* dataset classified each movie under multiple categories. To incorporate this information, we constructed a feature-by-item matrix where the items are the movies, and the features are the movie categories. We kept the sum of each column in this feature-by-item matrix uniform to 5 (the maximum MovieLens rating). Hence, the category score of each movie is set to be 5 divided by the total number of categories pertaining to that movie. For our hybrid recommendation algorithms, we appended the category-by-item matrix to the original user-by-item matrix. This matrix is then used to calculate the pair-wise similarity matrices in the algorithms described previously.

4 Evaluation

We evaluated our algorithms using the *MovieLens 1M* dataset [7], that consists of 1000209 ratings for 3883 movies by 6040 different users. A typical approach is to use 80% of the dataset for training, and the remaining 20% for testing [12]. We split the dataset on a user by user level – i.e. we placed the oldest 80% of the ratings for each user in the training set, and the remaining (newer) ratings in the test set. This resulted into a training set consisting of 800193 ratings, and a test set composed of 200016 ratings. We evaluated the predictions using *Mean Average Error* (MAE) metric as described in [4,10,11].

We compared a total of 13 different algorithms – summarised in Table 1. In the cases of algorithms that utilised LSA, we evaluated their performance across a range of different dimensions between 100 and 1000 in intervals of 100. The column "LSA Dimensions Used" shows the number of dimensions that gave the best results across all neighbourhood sizes.

Results in Fig. 1 show that *Item-based* recommenders perform considerably better than the *user-based* equivalent. Whilst LSA has a beneficial effect on *user-based* recommendations, it has an overall negative effect on the *item-based* recommendation results. The *ensemble* system that uses LSA (Algorithm 6) gives slightly better results across practically all neighbourhood sizes than the pure *item-based* system that does not use LSA. In fact, the best result obtained is for this *ensemble* setup (Algorithm 6) with a neighbourhood size of 80.

Increasing the neighbourhood size beyond 40 has marginal improvement in recommendation accuracy. Whilst *item-based* recommenders are most effective with a neighbourhood size of 40 with a slight deterioration of results for larger

Table 1. Recommendation algorithms

Algorithm index	Similar items	Similar users	Item category	LSA dimensions used
1	✓			-
2	✓			300
3		✓		-
4		✓		1000
5	✓	✓		-
6	✓	✓		300
7			✓	-
8	✓		✓	-
9	✓		✓	300
10		✓	✓	-
11		✓	✓	1000
12	✓	✓	✓	-
13	✓	✓	✓	300

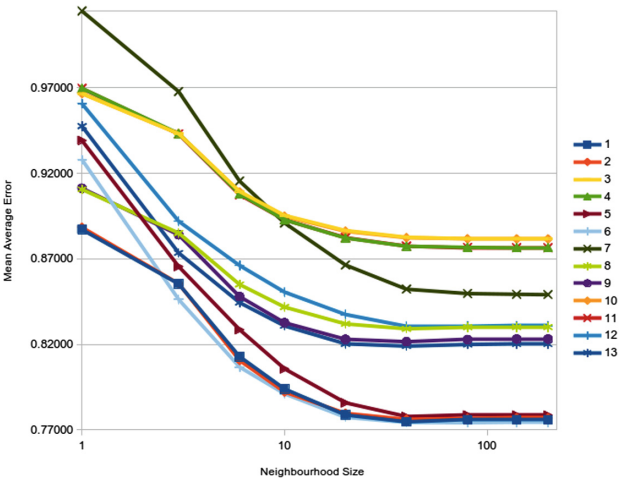

Fig. 1. Results obtained by the different algorithms

sizes, the performance of *user-based* recommenders keeps improving (albeit very slightly) as neighbourhood sizes are increased. The ensemble algorithm that uses LSA (Algorithm 6) obtains the best results with a neighbourhood size of 80, and results degrade slightly with larger neighbourhoods.

Results also show that incorporation of movie-type information cause an overall deterioration of results when used as a hybrid with the other collaborative systems. These observations confirm the observations reported in [6] where hybrid systems performed worse than purely collaborative systems.

5 Conclusions and Future Work

In this research, we compared *item-based* recommendation approaches with analogous *user-based* approaches, as well as with an *ensemble* approach combining both. We analysed the effect of LSA as well as the variation of neighbourhood size. Our best performing system is the *ensemble* system that uses LSA. However, this is followed closely by the *item-based* system that does not use LSA. Our *item-based* system performs best with a neighbourhood size of about 40, whilst the *ensemble* system performs best with a neighbourhood of size 80.

Our experiments have also shown that purely collaborative recommendation systems perform better than hybrid systems that incorporate item types. However, an interesting avenue for future research would be an investigation of the different methods of how content-type features may be incorporated in collaborative systems.

References

1. Azzopardi, J., Ivanovic, D., Kapitsaki, G.: Comparison of collaborative and content-based automatic recommendation approaches in a digital library of Serbian PhD dissertations. In: Calì, A., Gorgan, D., Ugarte, M. (eds.) KEYSTONE 2016. LNCS, vol. 10151, pp. 100–111. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53640-8_9
2. Azzopardi, J., Staff, C.: Automatic adaptation and recommendation of news reports using surface-based methods. In: Pérez, J., et al. (eds.) Highlights on Practical Applications of Agents and Multi-Agent Systems. AINSC, vol. 156, pp. 69–76. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28762-6_9
3. Burke, R.: Hybrid web recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_12
4. Choi, Y.S.: Content type based adaptation in collaborative recommendation. In: Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems, RACS 2014, pp. 61–65. ACM, New York (2014)
5. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
6. Garcin, F., Zhou, K., Faltings, B., Schickel, V.: Personalized news recommendation based on collaborative filtering. In: Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT 2012, pp. 437–441. IEEE Computer Society Washington (2012)
7. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. **5**(4), 19:1–19:19 (2015)
8. Lang, K.: Newsweeder: learning to filter netnews. In: Proceedings of the 12th International Machine Learning Conference, ML 1995, pp. 331–339. Morgan Kaufman (1995)
9. Li, Q., Kim, B.M.: An approach for combining content-based and collaborative filters. In: Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages - Volume 11, AsianIR 2003, pp. 17–24. Association for Computational Linguistics, Stroudsburg (2003)
10. Patel, V., Hasan, M.: Parallel ratio based CF for recommendation system. In: Proceedings of the 7th International Conference on Computing Communication and Networking Technologies, ICCCNT 2016, pp. 34:1–34:4. ACM, New York (2016)
11. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_9
12. Stanescu, A., Nagar, S., Caragea, D.: A hybrid recommender system: User profiling from keywords and ratings. In: Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01, WI-IAT 2013, pp. 73–80. IEEE Computer Society, Washington (2013)
13. Xia, C., Jiang, X., Liu, S., Luo, Z., Yu, Z.: Dynamic item-based recommendation algorithm with time decay. In: 2010 Sixth International Conference on Natural Computation, vol. 1, pp. 242–247, August 2010