Supporting Users of Open Online Courses with Recommendations: an Algorithmic Study

Soude Fazeli
Delft University of Technology, Open University of the Netherlands
Delft, the Netherlands
e-mail: s.fazeli@tudelft.nl

Enayat Rajabi
Dalhousie University
Halifax, NS, Canada
e-mail: rajabi@dal.ca

Leonardo Lezcano
eBay Inc.
San Jose, the United States
e-mail: llezcano@ebay.com

Hendrik Drachsler, Peter Sloep
Open University of the Netherlands
Heerlen, the Netherlands
e-mail:{hendrik.drachsler,peter.sloep}@ou.nl

***Abstract*— Almost all studies on course recommenders in online platforms target closed online platforms that belong to a University or other provider. Recently, a demand has developed that targets *open* platforms. Such platforms lack rich user profiles with content metadata. Instead they log user interactions. We report on how user interactions and activities tracked in open online learning platforms may generate recommendations. We use data from the OpenU open online learning platform in use by the Open University of the Netherlands to investigate the application of several state-of-the-art recommender algorithms, including a graph-based recommender approach. It appears that user-based and memory-based methods perform better than model-based and factorization methods. Particularly, the graph-based recommender system outperforms the classical approaches on prediction accuracy of recommendations in terms of recall.**

Keywords: recommender systems; collabortive filtering; open online course; performance; accuracy; matrix factorization

I. INTRODUCTION

Online learning platforms have become quite popular over the last decades [1]. Among those are platforms that offer open online courses. Such courses cover a particular topic, there are learning objectives, in the online platform one conducts the course's business [2]. The platforms provide learners with a social setting that is beneficial for learners to improve their learning outcomes [3]. Collaborating with others helps learners to feel less isolated, it provides them with a sense of belonging [4].

One of the main concerns of learners in online course platforms is to decide which course to take. This has become more essential for those learners who are graduate students since their preference goes to courses that can help them to make progress towards their career goals. Recommender systems have a solid track record in online learning platforms: they point out content to users that they might be interested in but are unaware of [5]. Although studies have been carried out on recommending courses to students in online learning platforms, almost all were on "closed online course platforms" that typically belong to a specific university. Examples are CourseAgent of the University of Pittsburgh [6] and CourseRank of Stanford University [7]. With the advent of open platforms, such as Coursera, edX, Udacity, MiriadaX, FutureLearn or OpenU, the need arises for recommenders that are free from curricular concerns but address the individual information needs of the learners. Moreover, most of the existing course recommenders base their recommendations on either learner or course content metadata. In many open online course platforms, often no comprehensive metadata about the learners are available. Typically, learners provide as little data they can, merely to get through the sign-up procedure. Also, availability of rich content metadata for open online courses is often an issue. Data regarding user interactions and activities become richer and more extensive as learners spend more time online and also more and more learners take advantage of the open online offers. It is a generic form of data that is common to all online learning platforms. Here we investigate how to make recommendations by using learner interactions. Specifically, our research question is:

How best to support learners in open online course platforms with recommendations based on their activities within the platform?

We conduct a study with data from OpenU, an open online learning platform developed by the Open University of the Netherlands to cater for the needs of learners other than their regular curricular students (professionals seeking professional development opportunities, 'passers by', alumni, etc. [8]). The OpenU courses require one to sign up but are otherwise open to anybody and free of charge.

To develop a recommender system one first needs to find out about the available input data. The user activities in the OpenU dataset are mainly implicit, coming from tracking data, like signing up for a course or contributing to a forum.

Therefore, Collaborative Filtering (CF) recommenders can be applied. Such methods make recommendations for a target user based on other users' opinions and interests. Content-based methods should be used when, first, rich content data is available (not the case in this study) and, second, users' rating information (5-star, binary, unary) is not available. However, "even if very few ratings are available, simple rating-based predictors outperform purely metadata-based ones" [9],. This is due to the difference between the item descriptions and the items themselves, and users often rate how much they like an items, not their descriptions.

CF methods come in two main categories: memory-based and model-based [10]. The former use statistical and mathematical approaches to infer similarity between users based on the users' data stored in memory. A well-known example is the k-Nearest Neighbour method (kNN, with neighbourhood size k), which uses similarity metrics such as the Pearson correlation coefficient, Cosine similarity, and the Jaccard coefficient. Model-based algorithms rely on probabilistic approaches to create a model of users' feedback. Examples of model-based algorithms are matrix factorization, and Bayesian networks. Model-based algorithms are faster than memory-based algorithms because they develop models of users' feedback offline. However, to create a user model, they require a full set of users' feedback. Model-based algorithms are also costly in terms of required resources and maintenance. Therefore, choosing a CF method depends on one's intentions and the purpose of the recommender system. Here we use both memory-based (both user-based and item-based) and model-based algorithms. User-based collaborative filtering methods try to find similarity between users, item-based collaborative filtering are based on similarity between items.

We study the following hypotheses, where performance is measured in terms of prediction accuracy of the recommendations generated:

H1: Item-based methods outperform user-based methods.

H2: State-of-the-art matrix factorization methods as model-based CFs outperform memory-based methods.

This paper is structured as follows: Section 2 describes the experimental method (dataset, algorithms, experimental setting). Section 3 presents the experimental results. Section 4 discusses which recommender algorithm is a good candidate for the open online learning platform used in this study. Section 5 draws conclusions and discusses opportunities for future work.

II. METHOD

A. Data

The data used in this study comes from the OpenU open online learning platform. The Open University of the Netherlands has developed the OpenU platform in 2009, which provides a broad, regional or national system for lifelong learning, where various institutional providers may offer their courses and programs for part-time study [11].

The OpenU dataset contains interactions data (92689 events) of 3462 students with 105 courses. The data is too sparse in terms of user transactions (98.14%) to make recommendations with traditional approaches such as CF. The data encompasses the time period from March 2009 until September 2013. The dataset is sufficiently large to apply recommender systems. The vast majority of participants in OpenU courses are professionals in various domains though most work in the educational sector [8].

Upon signing up users receive a student ID and can register for free in any OpenU course they are interested in. Figure 1 shows how course completion is related to the students' activities and interactions. It shows how frequently OpenU students communicate in the courses they attend. Each blue X represents the Percentage of Online Interactions (POI) for a given student and a given course, relative to the highest online interactions of a student in that course. Online interactions are calculated based on a student's contributions to chat sessions and forum messages. The black line in Figure 1 shows that the course completion rate for OpenU students goes up dramatically with increases in students' interactions with their course-mates and the academic staff.

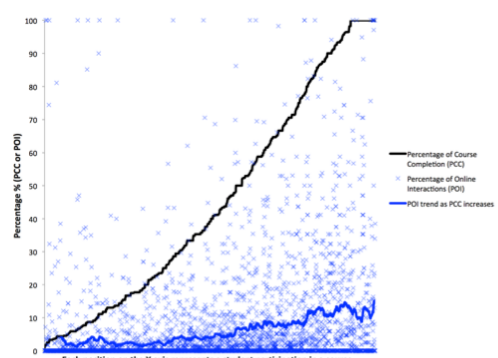

Figure 1. Interlinking and rating process

B. Algorithms

We evaluate applications of several state-of-the-art recommender algorithms as well as a graph-based recommender proposed in our previous work (ref removed).

1) Memory-based recommender systems

Most CF algorithms are based on kNN methods; they have proven to be quite successful [12]. kNN finds like-minded users and introduces them as the target user's nearest neighbours. kNN algorithms create a graph of users with edges as similarity relations. The appropriate similarity measure depends on whether explicit (e.g. 5-star ratings) or implicit user feedback (e.g. views, downloads, clicks, etc.) is available. The OpenU data is implicit user feedback: (userID,itemID) tuples – item refers to course in the current study. This kind of data is known as positive feedback only

[10]. So similarity measures such as Pearson correlation are not suitable (they require explicit user feedback). The Jaccard coefficient and Cosine are appropriate, though, since they use implicit user feedback in binary format [13].

2) A graph-based recommender systems

In spite of their popularity, kNN methods have two shortcomings. First, they usually do not work well with sparse user feedback as is often the case in the educational domain [5]. Second, they are only limited to k neighbours for each user. Thus users without an interest in a common set of items cannot be connected, even though they might be a good source of information for each other. Therefore, the implicit user networks inferred by these methods are always affected by this constraint. This affects the process of knowledge sharing and peer collaborations in online learning platforms, the social nature of which is intended exactly to foster such processes.

To address the sparsity issue and the restriction to k neighbours only, we employed a graph-based approach [14], [15]. It improves upon the kNN's process of finding neighbours by invoking graph search algorithms. First a graph is formed with nodes as users and edges as similarity relations between them. Then, recommendations for a target user are collected by 'walking' through the target user's neighbours. The graph-based approach is memory-based and user-based. This kind of approaches exist already and report positive effects in different domains [16]–[19]. However, almost all use data regarding either social relations between users or inter-user trust relations; these are not available for our datasets. Also, they all focus on performance measures and not specifically on the evolution of implicit user networks and neighbourhood formation.

3) Model-based recommender systems

Among the model-based CFs that make use of implicit user feedback, the Bayesian Personalized Ranking (BPR) method proposed by Rendle et al. [20] in our opinion best suits the data used in this study. They applied their BPR to the state-of-the-art matrix factorization models to improve the learning process in the Bayesian model used (BPRMF). In addition, we also use the baseline MostPopular approach, which makes recommendations based on general popularity of items. In this method, items are weighted based on how often they have been seen in the past.

III. EXPERIMENTAL SETTING

We use precision and recall to measure accuracy of the recommendations generated (Herlocker et al., 2004). *Precision* is defined as the ratio of the number of relevant items recommended to the total number of recommended items. *Recall* shows the probability that a relevant item is recommended: the number of relevant items divided by the total number of relevant items in the entire test set. Precision and recall range from 0 to 1. The number of courses in this experiment is 105 the number of top-n items to be recommended is 5 (approx. 5% of the courses) and 10 (approx.10% of the courses). A random 80% of the data was assigned to a training set, the rest constituted the test set. These metrics and settings are commonly used for empirical studies on recommender systems [21]. We used MyMediaLite [22] as an open source framework to test and to evaluate classical recommender algorithms.

IV. RESULTS

Table 1 shows the precision and recall of the recommendations generated. For each memory-based CF algorithm, we evaluated six neighbourhood sizes (k={5,10,20,30,50,100}). Model-based algorithms use latent factors; we tried three different numbers (3, 5 and 10). Values for the highest-scoring neighbourhood size are in bold, the highest values among all are underlined. Table 1 shows that the memory-based and user-based CFs (UB1, UB2, UB3) outperform both the item-based (IB1, IB2) and the model-based (MB1, MB2).

Relative to any CF, the graph-based CF (UB3) provides the best recall values for all sizes of neighbourhoods. For no CF the recall of the algorithms increases monotonously with neighbourhood size n. However, the graph-based CF (UB3) shows the largest overall increase: from 0.443 (k=5) to 0.507 (k= 100). Jaccard kNN (UB1) is performs second-best, right next to the graph-based CF (UB3). However, it provides a lower recall (0.469) than UB3 (0.507) for the largest size of neighbours k=100 (Recall@10); the same holds for Recall@5. As for precision, the user-based Jaccard kNN (UB1) performs best when k=30 with 0.169 (Prec@5) and 0.119 (Prec@10). However, the precision results of user-based Cosine kNN (UB1) when k=30 are close to UB2's: 0.167 (Prec@5) and 0.118 (Prec@10).

Table 1 also shows that the user-based CFs (UB1, UB2, and UB3) outperform the item-based ones (IB1, IB2) for all k, with quite a large difference. Both precision and recall of the item-based CFs decline when the size of neighbourhoods increases. The item-based Jaccard CF (IB1) provides the highest recall: 0.239 (Recall@5) and 0.313 (Recall@10) when k=5. The highest recall value provided by user-based CFs comes from UB3. For the same k it is 0.422 (Recall@5) and 0.463 (Recall@10). The highest recall of the item-based CFs (IB1 at 0.313 for k=5) is much smaller than the lowest recall of the user-based CFs (UB2 at 0.367 for k=5); the same holds for Recall@10. The precision values in Table 1 are similar to the one for recall.

Finally, Table 1 shows that the user-based CFs (UB1, UB2, and UB3) outperform the model-based ones (MB1, MB2) with a large difference. The BPRMF (MB2) performs better than MostPopular (MB1). The highest recall values of the model-based CFs are 0.266 (MB2's Recall@5 for f=5) and 0.354 (MB2's Recall@10 for f=10) whereas the lowest recall values of the user-based CFs (UB2's Recall@5=0.367 for k=5; UB2's Recall@10=0.415 for k=5). The BPRMF show the best precision values b(MB2) when f=5: 0.112 (MB2's Prec@5) and 0.082 (MB2's Prec@10), which are still much smaller than the lowest precision values of the user-based CFs (UB3's Prec@5=0.102 for k=5; UB3's Prec@10=0.086 for k=5).

V. Discussion

Contrary to what the recommender systems literature suggests [10], user-based CFs exceeded all expectations. Since the number of items (courses in this data study) is much smaller than the number of users for our dataset, item-based results were expected to trump the user-based ones. So, recommenders that make use of similarities between users perform better than those that make use of similarities between items (courses). Therefore, we reject H1.

Furthermore, the user-based recommenders (UB1, UB2, UB3), which are memory-based, widely outperform the model-based ones (MB1, MB2). Rather, we expected the model-based CFs to perform better since they often prove to outperform prediction accuracy of recommendations particularly when explicit user feedback is available (e.g. 5-star ratings) [23]. As a result, the memory-based recommenders perform better than the model-based ones on implicit user feedback. So we also reject, hypothesis H2.

Although the user-based CFs (UB1, UB2, UB3) perform best in terms of both precision and recall, there is no single best-performing one. If one is interested in more accurate recommendations, the graph-based approach (UB3) is the best candidate since it features the best recall. However, if one wants to achieve higher precision, the Jaccard kNN (UB1) and Cosine kNN (UB2) are best, though with a small difference. The graph-based CF (UB3) scores best for the larger neighbourhood sizes (k=50 and k=100). Perhaps, the reason for this is that it is able to find more new neighbours for a target user by traversing the graphs inferred for larger values of k. A larger network increases the chance of establishing more relations, which then improves the process of making recommendations for the target users.

I. Conclusion and further work

This study sought to find out how best to generate personalized recommendations from user activities within an open online course platform. We collected data from the OpenU open online course platform, in use at the Open University of the Netherlands. Seven different algorithms were tested. A graph-based CF algorithm was proposed to augment the accuracy of the generated recommendations. It was also used to improve the process of finding likeminded users. The results show that user-based and memory-based methods perform better than model-based and factorization methods. Particularly graph-based collaborative filtering algorithms outperform classical user-based, item-based, and model-based ones on the prediction accuracy of recommendations in terms of recall. However, the Jaccard nearest neighbours algorithm seems to be best for obtaining a high precision of the recommendations. Therefore, choosing the best-performing recommender depends on whether one wants to maximize precision or recall. In realistic situations one is likely to be interested in a trade-off between both. We conclude that, if one chooses the algorithms suggested here, recommenders can significantly contribute to the user experience in open online course platforms. As a corollary to our findings we note that, as the performance of graph-based recommender goes up with increasing neighbourhood sizes, our conclusions should hold even more strongly for the large datasets typically found in MOOCs, with massive student numbers in the (tens) of thousands [24].

TABLE I. Performance comparison of the recommender algorithms on precision and recall (the higher, the better); highest values for each algorithm are emphasized and best overall values underlined. Bold and underlined values are explained in the main text.

Method	**k**	**Prec@5**	**Prec@10**	**Recall@5**	**Recall@10**
User-based Jaccard kNN (UB1)	k=5	0.150	0.106	0.369	0.420
	k=10	0.155	0.109	0.377	0.434
	k=20	0.162	0.114	**0.453**	0.458
	k=30	0.169	0.119	0.406	0.465
	k=50	0.166	0.118	0.415	**0.473**
	k=100	0.165	0.116	0.408	0.469
User-based Cosine KNN (UB2)	k=5	0.148	0.104	0.367	0.415
	k=10	0.153	0.108	0.374	0.429
	k=20	0.161	0.114	0.394	0.451
	k=30	**0.167**	**0.118**	0.408	0.463
	k=50	0.166	0.116	**0.416**	0.468
	k=100	0.164	0.117	0.406	**0.472**
Graph-based CF (UB3)	k=5	0.102	0.086	0.422	0.443
	k=10	0.109	0.093	0.435	0.471
	k=20	0.115	0.101	0.448	0.462
	k=30	0.112	0.102	0.456	0.471
	k=50	**0.121**	0.106	**0.481**	0.488
	k=100	0.119	**0.108**	0.467	**0.507**
Item-based Jaccard KNN (IB1)	k=5	**0.109**	**0.063**	**0.239**	**0.313**
	k=10	0.107	0.055	0.212	0.276
	k=20	0.034	0.058	0.022	0.010
	k=30	0.019	0.026	0.010	0.026
	k=50	0.005	0.006	0.002	0.004
	k=100	0.00008	0.00011	0.00001	0.00003
Item-based Cosine KNN (IB2)	k=5	**0.096**	**0.117**	**0.237**	**0.302**
	k=10	0.054	0.114	0.180	0.259
	k=20	0.048	0.060	0.038	0.098
	k=30	0.022	0.028	0.012	0.030
	k=50	0.006	0.007	0.002	0.005
	k=100	0.006	0.007	0.002	0.004
Model-based MostPopular (MB1)	--	**0.094**	**0.061**	**0.208**	**0.291**
Model-based BPRMF (MB2)	f=3	0.095	0.064	0.212	0.273
	f=5	**0.112**	**0.082**	**0.266**	0.307
	f=10	0.109	0.083	0.261	**0.354**

In future work, we first intend to integrate the selected recommender algorithm in the OpenU platform to provide real-time recommendations. Second, we want to study how the graph-based approach can help to improve the process of finding like-minded neighbours in terms of social network analysis (SNA). Results from a preliminary study were quite promising. When testing the various algorithms, we already ran an SNA to investigate and study the formation of networks. Our preliminary findings show that the most active users exert a strong effect on the recommendations made. Third, and in line with the suggestions of Manouselis et al. in their work [5], we intend to run a user study to investigate if the selected recommender algorithm is suitable from a users' perspective as well. We want to measure novelty and serendipity of the graph-based recommendations through such a user study.

References

[1] T. Anderson, *The Theory and Practice of Online Learning (p. 484).* Athabasca, Canada: Athabasca University, 2008.

[2] I. Falconer, L. McGill, A. Littlejohn, C. Redecker, J. C. Muñoz, and Y. Punie, *Overview and Analysis of Practices with Open Educational Resources in Adult Education in Europe*. Luxembourg, Luxembourg, 2013.

[3] P. Sloep and L. Kester, "From Lurker to Active Participant," in *Learning Network Services for Professional Development*, 2009, pp. 17–25.

[4] S. Fetter and P. Sloep, "Fostering Online Social Capital through Peer-Support," *Int. J. Web-based Communities*.

[5] N. Manouselis, H. Drachsler, K. Verbert, and E. Duval, *Recommender Systems for Learning*. Springer Berlin Heidelberg, 2012.

[6] R. Farzan and P. Brusilovsky, "Social navigation support in a course recommendation system," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4018 LNCS, pp. 91–100, 2006.

[7] B. Bercovitz, F. Kaliszan, G. Koutrika, H. Liou, Z. M. Zadeh, and H. Garcia-molina, "CourseRank : A Social System for Course Planning," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 1107–1109.

[8] W. Rubens, M. Kalz, and R. Koper, "Improving The Learning Design of Massive Open Online Courses," *Tojet Turkish Online J. Educ. Technol.*, vol. 13, no. 4, pp. 71–80, 2014.

[9] I. Pilászy and D. Tikk, "Recommending new movies: even a few ratings are more valuable than metadata," in *In Proceedings of the third ACM conference on Recommender systems (RecSys '09)*, 2009, pp. 93–100.

[10] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, vol. 54. Springer New York Dordrecht Heidelberg London, 2011.

[11] H. Hermans, "OpenU: design of an integrated system to support lifelong learning," Open University of the Netherlands, 2015.

[12] A. Bellogín, P. Castells, and I. Cantador, "Neighbor Selection and Weighting in User-Based Collaborative Filtering : A Performance Prediction Approach," *ACM Trans. Web, Press*, vol. 1, no. 212, 2014.

[13] K. Verbert, H. Drachsler, N. Manouselis, M. Wolpers, R. Vuorikari, and E. Duval, "Dataset-driven research for improving recommender systems for learning," in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 2011, pp. 44–53.

[14] S. Fazeli, B. Loni, H. Drachsler, and P. Sloep, "Which Recommender system Can Best Fit Social Learning Platforms?," in *9th European Conference on Technology Enhanced Learning, EC-TEL 2014*, 2014, pp. 84–97,.

[15] S. Fazeli, A. Zarghami, N. Dokoohaki, and M. Matskin, "Mechanizing social trust-aware recommenders with T-index augmented trustworthiness," in *Trust, Privacy and Security in Digital Business 7th International Conference, TrustBus 2010, LNCS 6264*, 2010, pp. 202–213.

[16] J. Golbeck, "Computing and applying trust in web-based social networks," University of Maryland at College Park, College Park, MD, USA, 2005.

[17] P. Massa and P. Avesani, "Trust-aware Recommender Systems," in *RecSys '07 Proceedings of the 2007 ACM conference on Recommender systems*, 2007, pp. 17–24.

[18] S. Fazeli, A. Zarghami, N. Dokoohaki, and M. Matskin, "Mechanizing social trust-aware recommenders with T-index augmented trustworthiness," in *Proceedings of the 7th international conference on Trust, privacy and security in digital business*, 2010, pp. 202–213.

[19] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec," in *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, 2008, p. 931.

[20] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback," in *UAI '09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.

[21] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004.

[22] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "MyMediaLite: a free recommender system library," in *RecSys '11 Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 305–308.

[23] Nan Zheng, Qiudan Li, Shengcai Liao, and Leiming Zhang, "Which photo groups should I choose? A comparative study of recommendation algorithms in Flickr," *J. Inf. Sci.*, vol. 36, no. 6, pp. 733–750, 2010.

[24] K. Jordan and M. Weller, "Completion Data for MOOCs," *The Ed Techie, blog. Retrieved from http://nogoodreason.typepad.co.uk/no_good_reason/2013/12/completion-data-for-moocs.html*, 2013.