Which Recommender System Can Best Fit Social Learning Platforms?

Soude Fazeli, Babak Loni, Hendrik Drachsler, Peter Sloep

Open Universiteit Nederland, Centre for Learning Sciences and Technologies, Netherlands
Soude.fazeli@ou.nl, hendrik.drachsler@ou.nl, peter.sloep@ou.nl

Multimedia Computing Group, Delft University of Technology, Netherlands
b.loni@tudelft.nl

Abstract. This study aims to develop a recommender system for social learning platforms that combine traditional learning management systems with commercial social networks like Facebook. We therefore take into account social interactions of users to make recommendations on learning resources. We propose to make use of graph-walking methods for improving performance of the well-known baseline algorithms. We evaluate the proposed graph-based approach in terms of their F1 score, which is an effective combination of precision and recall as two fundamental metrics used in recommender systems area. The results show that the graph-based approach can help to improve performance of the baseline recommenders; particularly for rather sparse educational datasets used in this study.
Keywords. Recommender system, graph, teacher, social learning platform, social network, sparsity, performance

1 Introduction

Vassileva [1] introduces social recommender systems as a practical solution to help users in finding suitable resources that can support their learning process. Social recommenders are mainly based on two methods, either Collaborative Filtering (CF) algorithms or content-based algorithms. CF algorithms pass recommendations on to a user based on the opinions of many other users and their feedback on items. They first find like-minded users and create a network of so-called nearest neighbors; then they predict an item's rating for a target user on the basis of the ratings given by the nearest neighbors to this item. Content-based algorithms are based on preferences of a user summarized in a user model. They recommend an item to a user by comparing the representation of the item's content with the user's model. Due to this content dependence, CF algorithms have been applied more widely in social recommender systems because they are more flexible and require user opinions and feedback only instead of the actual content description, as do content-based algorithms.

In this research, we focus on interactions and collaboration between users in a social learning platform developed by the eContentPlus Open Discovery Space (ODS) project[1]. Social learning platforms combine traditional learning management systems (LMS) with commercial social networks like Facebook to provide easy content creation, access, sharing, bookmarking, etc. Beside forums and chat communities often provided in an LMS, they let users establish more connections and improve their networks of peers. To recommend the most suitable resources to the ODS users, we decided to use collaborative filtering algorithms since they focus on the similarities and overlaps of users' social activities.

Most of the CF algorithms employ similarity measures to build the nearest neighbors network that allows a recommender algorithm to learn. Such algorithms try to find like-minded users and introduce them as nearest neighbors of a target user for whom recommendations are generated. Although this kind of CF algorithms has proven quite successful in both research and practical use cases, they rely on the full user-item matrix data. That matrix, however, is not always available, particularly not in the educational domain as it involves fewer users and fewer transactions compared to e-commerce applications [2]. This problem originates from the sparse ratings of neighbors (the sparsity problem). When rating data are sparse, users are likely to receive irrelevant recommendations. Therefore, we aim to take advantage of a graph-based approach, which extends and improves the baseline nearest neighbour CF by invoking graph search algorithms. There exist quite a few approaches on improving performance of recommenders by using graph-walking algorithms [3]–[7]. Almost all use data that include either actual social (or trust) relations between users, content features of items, or tags assigned to the items. None of these, however, were available for this study. Therefore, our main research question is:

RQ: How to generate more accurate and thus more relevant recommendations for the users in social learning platforms by employing graph-walking methods?

Our overall aim is to find out which recommender algorithm best fits a social learning platform such as the ODS platform. In doing so, we follow the research method proposed by Manouselis et al. for evaluating TEL recommender systems [8]. After developing the conceptual model based on a literature review and after carrying out an interview study [9], we conducted an offline data study to investigate if and how the use of a graph-walking recommenders can help to improve prediction accuracy of the recommendations made based on data collected from platforms similar to the future ODS dataset. In this paper, we present results of this offline data study. As a further step, we intend to run a user study to measure user satisfaction on the recommendations generated.

The rest of this paper is organized as follows: Section 2 gives a brief review of related studies. The proposed graph-based approach used to collect the recommendations is described in Section 3. Sections 4 and 5 present the experimental study and results, respectively. Then, we explain the practical implications of the experiment in Section 6 and conclude by giving an overview of future work in Section 7.

---

[1] http://opendiscoveryspace.eu

2 Related works

Manouselis et al. [10] investigated which collaborative filtering algorithm supports multi-attribute ratings of the users within an online community composed of teachers from all over the Europe. The authors reported the results on different variations of their proposed multi-attribute collaborative filtering algorithm. Cechinel et al. [11] applied several memory-based collaborative filtering algorithms on the MERLOT repository to investigate which of the algorithms used performs best. Their study focused on evaluating the collaborative algorithms in terms of the automated quality assessment of learning resources within the MERLOT dataset. Koukourikos et al. [12] proposed to use sentiment analysis to enhance collaborative filtering algorithms. Such techniques take into account the opinions of a user on the quality of the resources before recommending resources to other users. They studied performance of their proposed approach on the MERLOT repository. Although all studies mentioned presented useful insights in applying recommender system algorithms to educational datasets, none of them dealt adequately with the sparsity problem. Verbert et al. [2] presented a dataset-driven study by testing different classical collaborative filtering algorithms on a set of educational datasets, including the Travel well, MACE, and Mendeley datasets from the dataTEL project [13]. They proposed using the implicit data of users such as tags, downloads, etc. However, their approach fails in cases in which not even the implicit data of users are sufficient to find similarity patterns between users. Manouselis et al. [14] compared the results of evaluating multi-criteria algorithms on a real dataset of original data collected from the Organic.Edunet portal and a synthetic dataset including real data plus some simulated data. They simulated how the users would have rated the learning resources and then added these simulated ratings to the real dataset. But it remains unclear in how far actual user ratings match the simulated ones.

Finally, an as yet rarely used approach in the educational domain is the state-of-the-art Matrix Factorization (MF) method. It was able to tackle the sparsity problem in ACM recommender systems research by decomposing the sparse user-item ratings matrix into two matrices using latent features of the items. Manouselis et al. [8], reported only one study on MF [15]. Thai-Nghe et al. showed that MF has the potential to take into account temporal effects such as the increasing knowledge of learners. However, they did not focus on making recommendation on learning resources.

In an attempt to improve prediction accuracy of recommendations and thus to overcome sparsity, graph-based algorithms have emerged. However, very few educational applications are known. Anjorin et al. [7] aimed to make use of the tags assigned by users in a social platform called CROKODIL. They extended an existing approach, based on the PageRank algorithm. Their study used a dataset that is similar to ODS. As with ODS, extra information about learning resources or tagging data was lacking. Therefore, they could not make use of tags and keywords assigned to the learning resources.

3 A graph-based approach

We propose to employ graph-walking algorithms in order to improve the prediction accuracy of recommendations. Such an approach first forms a graph in which nodes are users and edges are similarity relations between users. Then, it collects recommendations for a target user by walking through the target user's neighbors.

3.1 Creating the graph

We take into account a Social Index (S-index) for each user, which is inspired by the H-index and calculated using the algorithm 1. The H-index is an indicator of publications of an author. It combines information on the number of publications of some author with the number of citations [20]. Similarly, the S-index of a user *u* shows not only how many times user *u* has been selected as a neighbor, but also how much the user *u* contributed to interactions on items in common with her neighbors [6]. In this study, the S-index ranges from 1 to 100, the *similarityScore* between two users from 0 to 1. The S-index helps us to extend and improve finding like-minded users (neighborhoods). We use it for sorting list of raters of an item. This list helps us to discover new neighbors for a user, who have been excluded when walking through the created graphs but still can be a useful source of information.

Algorithm 1 Computing S-index for user u

```
upon event (COMPUTE S-INDEX| u, NeighborsList)
    SortedNeighborsList ← SortDescendingBySimilarityScore(NeighborsList);
    FinalNeighborsList ← Normalize(SortedNeighborsList,MaximumSindex);
    Sindex ← 0;
    for ( similarityScore(u,n); n in FinalNeighborsList) do
        if Sindex <= similairtyScore then
            Sindex= Sindex+1;
        else
            Break;
        end if
end for
updateSindex(Sindex);
end event
```

3.2 Collecting recommendations

The graph-based approach uses a modified BFS (Breadth First Search) graph search algorithm to traverse the implicit social network created using S-index and items' raters lists. We chose BFS among the well-known walking algorithms like depth first search to first poll the direct neighbors when collecting recommendations in the created user graph. The inferred neighborhoods, therefore, are not limited to the *k* nearest neighbors only; instead we provide dynamic neighborhoods beyond *k* for each target user depending on the new neighbors the graph-based approach helps us to infer. We formalize this procedure as follows:

```
G(V,E) = CreateSocialGraph();   // V contains users
                                // E contains similarity relations between users
for all u ∈ V do
    ComputeSindex(u, N);        // N contains users who have user u as their neighbor
    G(V,E') ← BFS(u, G(V,E));      // E ⊂ E' where E' contains:
                                // 1. explicit similarity relations (u,n) ∈ E and
                                // 2. new inferred relations (u, n')
    TopItems ← CollectRecommendations(u, G(V, E'));
    UpdateSindex(u,N');         // N' contains new neighbors found
    UpdateSocialGraph();
end for
```

Moreover, we followed a discounting mechanism when collecting recommendations from the neighbors who appear in the BFS result. For this, we propagate the similarity scores between users who have no direct connection yet by multiplying the interconnecting users' similarity scores. This guaranties that the inferred similarity score is always smaller than the actual values of the interconnecting edges.

4 Experimental study

In order to address the research question described in Section 1, we conducted an offline experiment to compare a graph-based approach with baseline algorithms. But first, we briefly describe the classification categories of CF algorithms according to their *type* and *technique* [16].

Type refers to model-based and memory-based algorithms. Model-based algorithms rely on probabilistic approaches to create a model of users' preferences. Examples of model-based algorithms are neural networks, Bayesian networks, and algebraic approaches such as those using eigenvectors. Memory-based algorithms use statistical and mathematical approaches based on the users' data stored in memory. Examples are the Pearson correlation coefficient, Tanimoto-Jaccard coefficient, and Euclidean distance. In general, model-based algorithms are faster than memory-based algorithms because they develop models of users' preferences offline. However, they require a full set of users' preferences to create a user model. Moreover, model-based algorithms often prove to be costly in terms of required resources and maintenance efforts. Therefore, choosing what type of CF to use is a trade-off that depends on the use case's limitations. In this study, we use both memory-based and model-based algorithms to find out which one can best help to tackle the sparsity problem.

The *technique* of CF algorithms often refers to user-based and item-based algorithms [16]. User-based algorithms try to find patterns of similarity between users in order to make recommendations; item-based algorithms follow the same process but are based on similarity between items. Here, we are interested in both user-based and item-based CFs because we focus on users' interactions and activities.

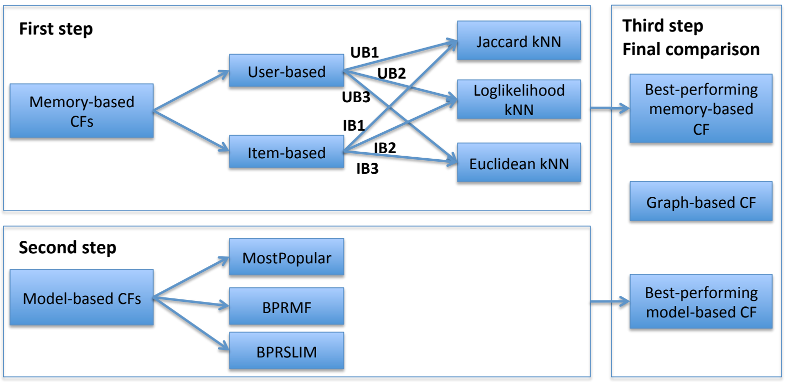

Fig. 1. Experimental method used in this paper

As shown in Fig. 1, the experimental study in this paper consists of three steps:

1. Memory-based CFs: we compare several both user-based and item-based k Nearest Neighbors (kNN) that can process binary data in terms of their F1 score. We choose the best-performing baseline kNN as the underlying layer for the graph-based approach.
2. Model-based CFs: we compare several model-based CFs, which are appropriate candidates for the binary data we have in this study.
3. Ultimately, we compare the graph-based approach with the best performing CFs selected from each of the previous two steps. We discuss the results to find out which of the CFs used performs best for our data and, thus, can be selected to be integrated in the ODS social platform.

In the experiment, we used Apache Mahout[2] and MyMediaLite[3] as open source frameworks; they provide implementations of the baseline collaborative filtering algorithms. Moreover, both frameworks enable us to evaluate the performance of baseline algorithms. The proposed graph-based approach has been implemented in Java.

4.1 Candidate memory-based CFs

As indicated, memory-based CF algorithms try to find similarity between users' opinions, interests, and actions on the items (from here on, we use items for learning resources). To measure similarity, we need to select a similarity measure that is able to process educational datasets including user interactions data. The educational datasets used in this study consist of too few explicit user preferences. But they do provide implicit user preferences such as views, downloads, tags, etc., which show users' interest in particular items as binary indicators (interested in the item: yes/no). Some

[2] http://mahout.apache.org/
[3] http://mymedialite.net/

of the similarity measures such as Pearson correlation are not suitable for this kind of data because they require explicit user preferences. Among the popular similarity measures, the Tanimoto-Jaccard coefficient (Jaccard coefficient, from here on), Log-likelihood, and Euclidean distance are most appropriate if the data includes implicit user preferences in binary format.

4.2 Candidate model-based CFs

Among the model-based CFs that make use of implicit user preferences, the Bayesian Personalized Ranking (BPR) method proposed by Rendel et al. [17] in our opinion best suits the data used in this study. Rendel et al. [17] aimed to optimize the learning process for the task of personalized ranking on a set of items. They applied their BPR to the state-of-the-art matrix factorization models to improve the learning process in the Bayesian model used (BPRMF) [18]. In addition, we use an extended version of BPR with Sparse Linear Methods (SLIM) [19] for item ranking optimized for BPR optimization criterion. The SLIM method [19] generates top recommendations by aggregating positive user feedback on items. This approach ‘learns’ a sparse aggregation coefficient matrix from aggregated users’ feedback and can produce fast and accurate recommendations. Beside these two methods based on BPR, 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.

Table 1. Details of the selected datasets

Dataset	Number of users	Number of items	Transactions	Sparsity (%)	Source
MACE	631	12,571	23,032	99.7096	MACE portal[4]
OpenScout	331	1,568	2,560	99.5067	OpenScout portal[5]
MovieLens	941	1,512	96,719	93.6953	GroupLens research[6]

4.3 Datasets

We selected the MACE and OpenScout datasets for the following reasons:

- The datasets contain social data of users such as ratings, tags, reviews, etc. on learning resources. So, their structure, content and target users are quite similar to the ODS dataset.

[4] http://mace-project.eu/
[5] http://learn.openscout.net/
[6] http://grouplens.org/datasets/movielens/

- Running recommender algorithms on these datasets enables us to conduct an offline experiment for studying the recommender algorithms before going online with the actual users of the ODS.
- Both MACE and OpenScout datasets comply with the CAM (Context Automated Metadata) format [21], which provides a standard metadata specification for collecting and storing social data. CAM will also be applied in the ODS project for storing the social data.

Since the educational domain lacks a 'golden' standard dataset to run data studies, such as the MovieLens dataset, we also tested the MovieLens dataset as a reference dataset. Table 1 provides an overview of these datasets. Note that the educational datasets MACE and OpenScout clearly suffer from extreme sparsity.

5 Results

The offline experiment in this study gauges the F1 score. We chose F1 due to the type of users data in this study and also in typical social learning platforms, which are implicit user preferences in binary format. We could have made use of other common metrics, such as MAE, RMSE and nDCG. However, these only work if we have explicit user preferences available, like 5-star ratings This is hardly ever the case in educational settings. Another advantage of F1 is that it combines precision and recall, which are both important metrics to evaluate accuracy and coverage of recommendations generated [22]. F1 ranges from 0 to 1. In this experiment, we split users' ratings randomly into two sets: a training set (80%) and test set (20%). The sets include actual and predicted relevance indictors of users. We computed F1 for the top 10 recommendations of the result set for each user. These settings are commonly used for empirical studies on recommender systems [22].

5.1 Memory-based CFs

First, we evaluated several baseline k-nearest neighbor (kNN) CFs on the similarity measures: Jaccard coefficient, Loglikelihood ratio, and Euclidean distance. We did so to find out which of them performs best on the data used in this study. Figure 1 shows the result of the F1 for testing the following baseline CFs:

1. User-based Jaccard kNN (UB1)
2. User-based Loglikelihood kNN (UB2)
3. User-based Euclidean kNN (UB3)
4. Item-based Jaccard kNN (IB1)
5. Item-based Loglikelihood kNN (IB2)
6. Item-based Euclidean kNN (IB3)

The used datasets are: MACE, OpenScout, Travel well and MovieLens. The horizontal axis (x) indicates different sizes of neighborhood (*k*) and the vertical axis (y) shows the values of F1. As Fig. 1 shows, the F1 value of the used algorithms provides different patterns depending on the used datasets. In general, user-based Jaccard kNN (UB1) provides the best F1 scores for all the datasets used: MACE (exceeding 7.7%), OpenScout (exceeding 10%), and MovieLens (exceeding 20%); only with an exception for OpenScout when k=20. Among the datasets used in Fig. 2, MovieLens owns the highest F1 values of all. The main reason for this (expected) result is the larger size of the dataset in terms of number of user transactions. The F1 result for MovieLens is consistent with the previous study by Verbert et al. [2] for user-based Jaccard kNN (UB1) (± 0.2). For both MACE and OpenScout, although UB1 has the best F1 results for all sizes of neighbors, its F1 values fluctuate while *k* increases and thus, it does not follow a clear pattern according to the sizes of neighborhoods. Unlike MACE and MovieLens, the F1 results for OpenScout quite declines by increasing *k*.

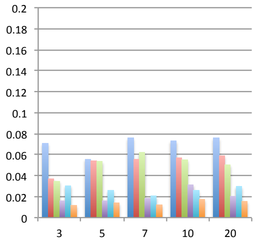

Fig. 2. F1 of memory-based CFs on MACE

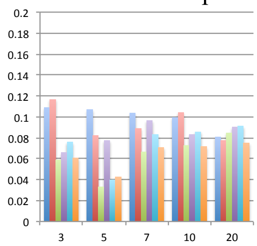

Fig. 3. F1 of memory-based CFs on OpenScout

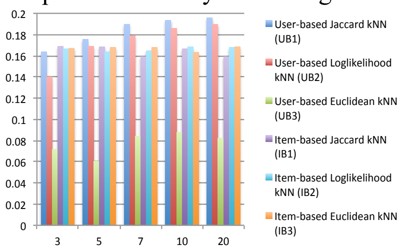

Fig. 4. F1 of memory-based CFs on MovieLens

F1 results of the item-based CFs are dataset-dependent; similar to the user-based CFs. Fig. 2 shows that for MACE, the user-based CFs (UB1, UB2, UB3, and UB4) outperform the item-based ones (IB1, IB2, IB3) for all *k,* with quite a large difference. The smallest difference (0.4%) is between UB2 and IB2 for *k*=3, the largest one (7.5%) is between UB4 and IB3 for *k*=20. This is unexpected since MACE has many more items (5,696) than users (105).

For OpenScout, IB1 and IB2 perform best, right after the user-based CF (UB1) for all sizes of neighborhoods and even they have better F1 than UB1's F1 when *k*=20. For MovieLens, the item-based CFs perform quite well and quite close to the best performing CFs: UB1 for the smaller sizes of neighbors (*k*=3,5). However, for *k* larger than 5, the F1 of the item-based CFs decreases compared to UB1's F1. For *k*=20, the difference between the F1 of IB3 as the best item-based CF and the best-performing user-based UB1 is more than 3%. In summery, CF recommenders that make use of similarities between users perform better than those that make use of similarities between items. We decided to select the Jaccard kNN for the ultimate comparison with the other candidate CFs.

5.2 Model-based CFs

We now report the F1 results using model-based CFs on the same datasets as were used in previous sections: MACE, OpenScout, and MovieLens. As explained in section 3.1, we choose to use three model-based CFs:

1. The BPR method using Matrix Factorization (BPRMF)
2. The BPR method using SLIM (BPRSLIM)

3. MostPopular (a well-known baseline CF often used in recommender research)

Fig. 2 shows the F1 results. The horizontal axis (x) again shows the datasets and the vertical axis (y) indicates the values of F1 for the model-based methods. Similar to the results for memory-based CFs, the F1 scores of model-based CFs are also dataset-dependent and MovieLens again scores best (exceeding 11%). This refers to the lowest sparsity of this dataset compared to others (see Table1). In general, BPRSLIM performs best for all the datasets: MACE (6.76%), OpenScout (8.53%), and MovieLens (18%). BPRMF stands in the second place for MACE and MovieLens, providing F1= 5.6% and 17.7%, respectively. For OpenScout, the differences between F1 of BPRSLIM and the others are quite large (the lowest gap is around 6%). We decided to choose BPRMF since it best performs among the model-based methods used and for all the datasets.

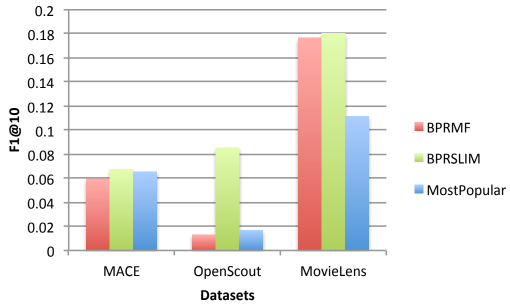

Fig. 5. F1 of the model-based CFs for all datasets used

Final results and discussion

In this step, we compare the graph-based approach with the best performing CFs from the previous steps: Jaccard kNN (memory-based) and BPRMF (model-based). Recall that the graph-based approach is a memory-based and user-based CF. For making a fair comparison, the graph-based CF also employs the same similarity measure used in the best performing baseline CF: the Jaccard coefficient. For the MACE and MovieLens datasets, we choose neighborhoods of size k=20, for which the F1 score is the highest (see Figure 1). Unlike this, for OpenScout, the best F1 score is for k=3.

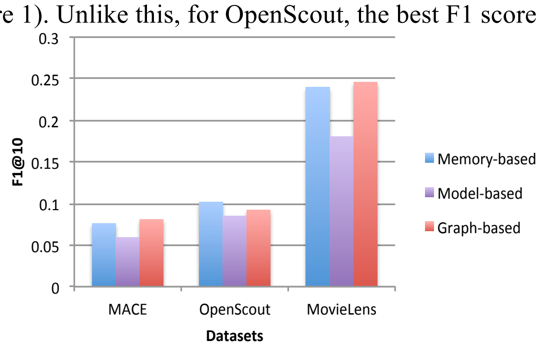

Fig. 6. F1 of the graph-based CF and the best performing baseline memory-based and model-based CFs, for all the datasets used

The graph-based approach collects recommendations for a target user from the neighbors reachable at a maximum path length (L). In the baseline kNN method, L always equals 1, which imposes the constraint of including only directly connected users to a target user. For the graph-based approach, choosing a value for L involves a trade-off. Increasing the L provides us with higher coverage but lower precision. Moreover, choosing larger path lengths can be more risky because of including malicious users in the recommendation procedure. Therefore, we set the maximum path length at L=3. Fig. 3 shows the F1 results of best performing memory-based CF (Jaccard kNN), model-based CF (BPRSLIM) compared to the graph-based CF. The horizontal axis (x) indicates the datasets used and the vertical axis (y) shows the values of F1.

As Fig. 3 shows, the graph-based approach performs best for MACE (8%) and MovieLens (24%) and the selected memory-based and model-based CFs stand in second and third place right after the graph-based CF. For OpenScout, the memory-based approach performs better with a difference of almost 1%. Note that the size of neighborhoods selected for Jaccard kNN on OpenScout was k=3, which is the smallest k (see Figure1) whereas k was set to 20 for the graph-based CF on other datasets. We did so because our strategy was to select the best-performing memory-based and model-based from steps 1 and 2 of this experiment. The Jaccard kNN performed best for k=3 in the case of OpenScout. If we consider the same k as we used for the graph-based (20), the Jaccard kNN's F1 (8%) is lower than the graph-based F1 (9.1%) for the OpenScout dataset. This shows that the graph-based approach performs well for all the datasets used.

In conclusion, according to the aggregated results presented in Fig.4, the graph-based approach can help to deal with the sparsity in the educational data coming from the social learning platforms. This is reflected by an improved F1, which is an effective combination of precision and recall of the recommendation made.

6 Practical implications and limitations

In the current study, it was difficult to make a comparison with the findings of related empirical research studies, such as the ones by Verbert et al. [2] and Manouselis et al. [10]. One of the reasons could be the use of different versions of the same dataset because the collected data belongs to different periods of time. For instance, for the MACE dataset, different versions are available. In fact, no unique version has been fixed for running the experiments, nor for making a comparison, in the community for Technology-Enhanced Learning (TEL) recommender system [13]. This problem originates from the already mentioned lack of a golden standard dataset in the educational domain, like the MovieLens dataset in the e-commerce world. In fact, it seems the TEL community, instead of aiming for their own single golden standard, should collect several representative datasets that can be used as a main set of references for different data studies on personalization and recommender systems. This observation was already made by the dataTEL project [13].

7 Conclusion and further work

The main goal of our study was to identify the most appropriate recommender algorithm that can support users to find useful resources in a social learning platform, such as the ODS platform. We conducted an offline data-driven study to evaluate a set of candidate recommender algorithms on a set of representative datasets similar to the ODS future dataset, as well as MovieLens from the ACM recommender systems community. We proposed a graph-based approach that aims to improve the process of neighborhood formation and thus, to improve the performance of baseline methods. The experimental study presented in this paper consists of three steps. First, we investigated which memory-based nearest neighbor methods best performs for the educational data used. Second, we evaluated state-of-the-art model-based methods using matrix factorization and Bayesian models. Ultimately, we evaluated the graph-based approach in comparison with the best-performing methods from the first and second steps of the experiment. The results showed that the graph-based approach outperforms baseline CFs and thus can tackle the sparsity problem in the data coming from social learning platforms. At present, we are working on using the matrix factorization (MF) methods in the graph-based approach, and to investigate whether this hybrid approach can improve the performance of each of the methods alone. The results presented in this paper serve as an initial step to investigate a recommender algorithm that can best fit the social learning platforms similar to the one used for ODS. As a further step, we intend to study usability of the selected recommender approaches by evaluating user satisfaction on novelty and diversity of the recommendations made.

Acknowledgements. This paper is part of a doctoral study funded by NELLL (the Netherlands Laboratory for Lifelong Learning) and the Open Discovery Space project. Open Discovery Space is funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D. This document does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of its content. The work of Hendrik Drachsler has been supported by the EU project LinkedUp FP7 Program. Last but not least, we would like to thank the FIT Fraunhofer Institute, especially Dr. Martin Wolpers and Katja Niemann, for providing us with both the MACE and OpenScout datasets on a short notice. Without their support, this study would not have been possible.

References.

[1] J. Vassileva, "Toward Social Learning Environments," *Learning*, vol. 1, no. 4, pp. 199–214, 2009.

[2] 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.

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

[4] 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.

[5] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09*, 2009, p. 203.

[6] 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.

[7] M. Anjorin, T. Rodenhausen, R. D. García, and C. Rensing, "Exploiting Semantic Information for Graph-Based Recommendations of Learning Resources," in *In proceedings of 7th European Conference on Technology Enhanced Learning (EC-TEL 2012)*, 2012, pp. 9–22.

[8] N. Manouselis, H. Drachsler, K. Verbert, and E. Duval, *Recommender Systems for Learning*. Springer Berlin Heidelberg, 2012, pp. 1–61.

[9] S. Fazeli, H. Drachsler, F. Brouns, and P. Sloep, "Towards a Social Trust-aware Recommender for Teachers," *Spec. issue Recomm. Syst. Technol. Enhanc. Learn. Res. Trends Appl.*, 2013.

[10] N. Manouselis, R. Vuorikari, and F. Van Assche, "Collaborative recommendation of e-learning resources: an experimental investigation," *J. Comput. Assist. Learn.*, vol. 26, no. 4, pp. 227–242, Jul. 2010.

[11] C. Cechinel, S. Sicilia, Miguel-Ángel Sánchez-Alonso, and E. García-Barriocanal, "Evaluating collaborative filtering recommendations inside large learning object repositories," *Inf. Process. Manag.*, 2012.

[12] A. Koukourikos, G. Stoitsis, and P. Karampiperis, "Sentiment Analysis: A tool for Rating Attribution to Content in Recommender Systems," in *in Manouselis N., Drachsler H., Verbert K., Santos O.C. (Eds.), Proceedings of the 2nd Workshop on Recommender Systems in Technology Enhanced Learning 2012, 7th European Conference on Technology Enhanced Learning (EC-TEL 2012)*, 2012, pp. 61–70.

[13] H. Drachsler, T. Bogers, R. Vuorikari, K. Verbert, E. Duval, N. Manouselis, G. Beham, S. Lindstaedt, H. Stern, M. Friedrich, and M. Wolpers, "Issues and

considerations regarding sharable data sets for recommender systems in technology enhanced learning," *Procedia Comput. Sci.*, vol. 1, no. 2, pp. 2849–2858, 2010.

[14] N. Manouselis, G. Kyrgiazos, and G. Stoitsis, "Revisiting the Multi-Criteria Recommender System of a Learning Portal," in *in Manouselis N., Drachsler H., Verbert K., Santos O.C. (Eds.), Proceedings of the 2nd Workshop on Recommender Systems in Technology Enhanced Learning 2012, 7th European Conference on Technology Enhanced Learning (EC-TEL 2012)*, 2012, pp. 35–48.

[15] N. Thai-Nghe, L. Drumond, A. Grimberghe, L. Schmidt-Thieme, and A. Krohn-Grimberghe, "Recommender system for predicting student performance," *Procedia Comput. Sci.*, vol. 1, no. 2, pp. 2811–2819, Jan. 2010.

[16] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering Recommender Systems," *Adapt. Web Lect. Notes Comput. Sci.*, vol. 4321/2007, pp. 291–324, 2007.

[17] 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.

[18] S. Rendle, "Context-Aware Ranking with Factorization Models," *Stud. Comput. Intell.*, vol. 330, 2011.

[19] X. Ning and G. Karypis, "SLIM: Sparse Linear Methods for Top-N Recommender Systems," in *IEEE 11th International Conference on Data Mining*, 2011, pp. 497–506.

[20] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 102, no. 16569, 2005.

[21] H. Schmitz, M. Scheffel, M. Friedrich, M. Jahn, K. Niemann, M. Wolpers, and S. Augustin, "CAMera for PLE," vol. 5794/2009, pp. 507–520, 2009.

[22] 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.