Which Recommender System Can Best Fit Social Learning Platforms?

Soude Fazeli, PhD candidate, OUNL
Babak Loni, TU Delft
Dr. Hendrik Drachsler, OUNL
Prof. dr. Peter Sloep, OUNL

Welten Institute
Research Centre for Learning, Teaching and Technology

Open Universiteit
welten-institute.org

Open Discovery Space (ODS)

A socially-powered, multilingual open learning infrastructure in Europe

Recommendations!

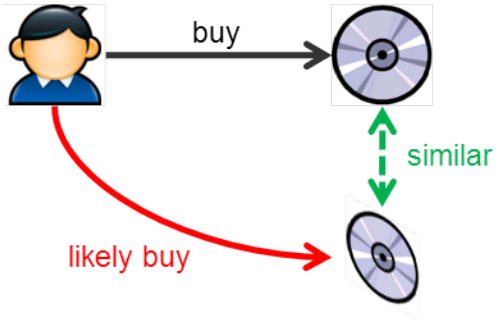
Which recommender approach best fits ODS platform?

Welten Institute
Research Centre for Learning, Teaching and Technology

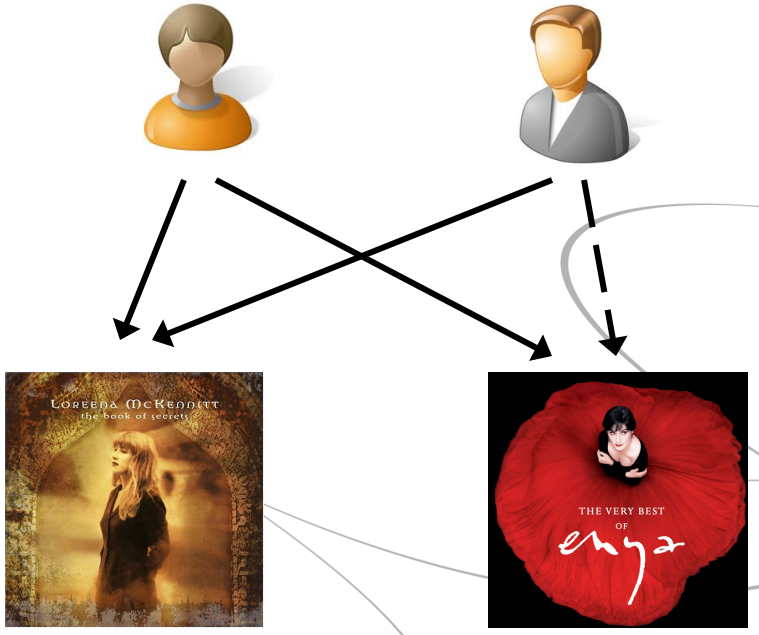
Open Universiteit
welten-institute.org

Recommender algorithms

1. Content-based



2. Collaborative filtering ✓



Similarity

Sparsity!

Learning domain has its own data, limitations, and expectations

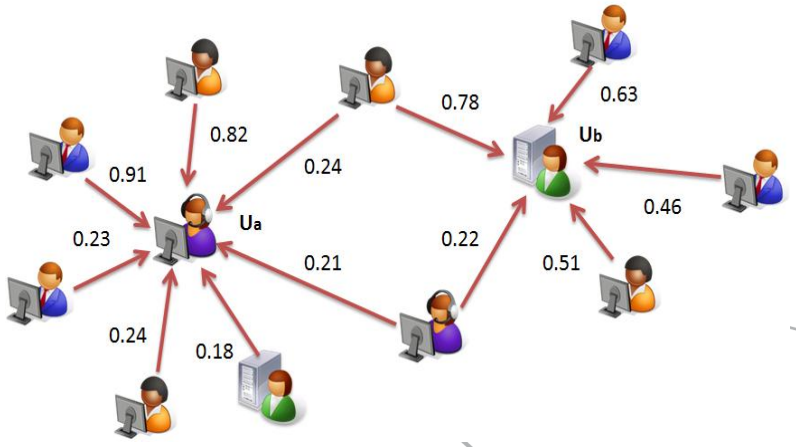
- Too sparse data
- Too few 5-star ratings
- Often no proper tags and annotations
- Can not use only popular reference datasets like MovieLens, Netflix, etc.

Dataset	Users	Learning objects	Transactions	Sparsity (%)
MACE	105	5,696	23,032	99.71
OpenScout	331	1,568	2,560	99.51
MovieLens 100k	941	1,512	96,719	93.69

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

A graph-based recommender system

- Implicit networks: a graph
  - Nodes: users
  - Edges: similarity relationships
  - Weights: similarity values
- Improve the process of finding nearest neighbors
  - Social Index (S-index) for each user
  - H-index: the impact of publications of an author



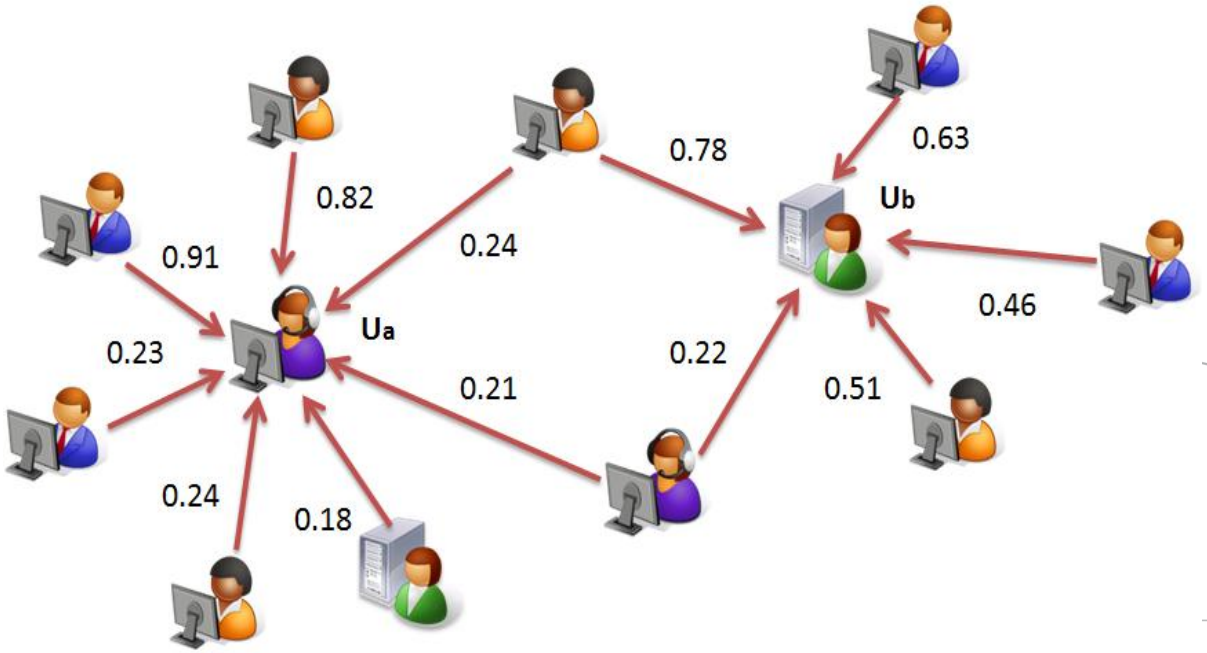
Creating the graph

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
```

Indegree (u_a) = 7
Indegree (u_b) = 5

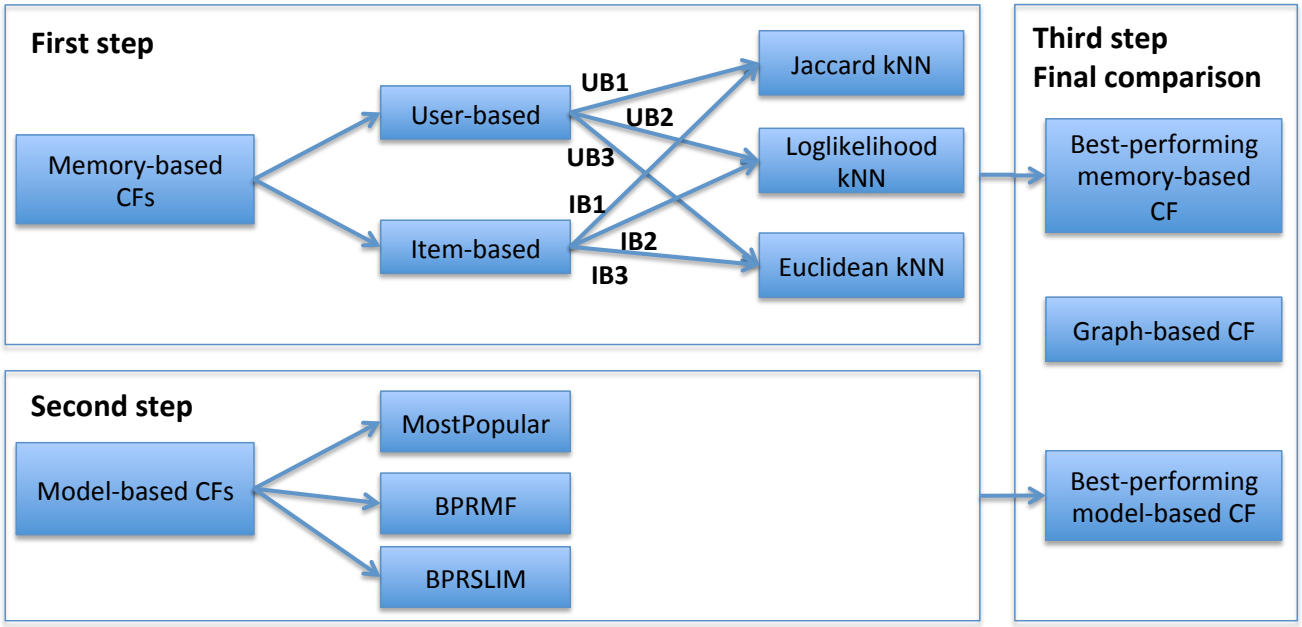
S-index (u_a) = 2
S-index (u_b) = 4



Collecting recommendations

```
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
```

Experimental study
1. The method

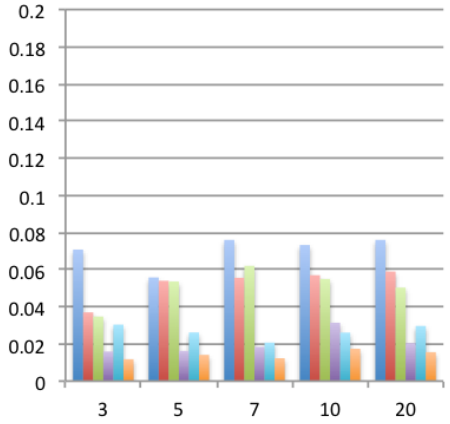


Experimental study
2. Data

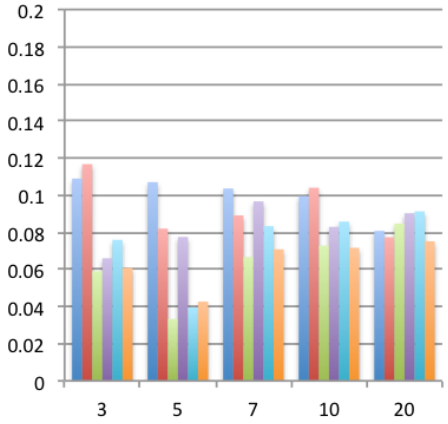
1. The datasets contain social data of users such as ratings, tags, reviews, etc. on learning resources. So, their structure, content and target users are quite similar to the ODS dataset.
2. Running recommender algorithms on these datasets enables us to conduct an offline experiment for studying the recommender algorithms before going online with the actual users of the ODS.
3. Both MACE and OpenScout datasets comply with the CAM (Context Automated Metadata) format. CAM is also applied in the ODS project for storing the social data.

Dataset	Users	Learning objects	Transactions	Sparsity (%)
MACE	105	5,696	23,032	99.71
OpenScout	331	1,568	2,560	99.51
MovieLens	941	1,512	96,719	93.69

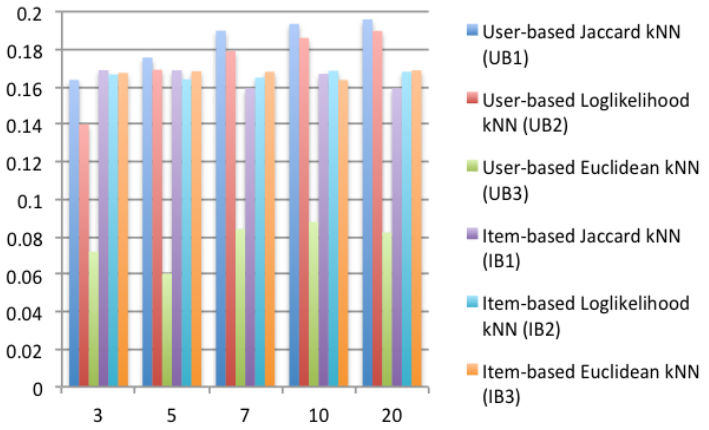
Experimental study
3. Results
3.1. Memory-based CFs



OpenScout



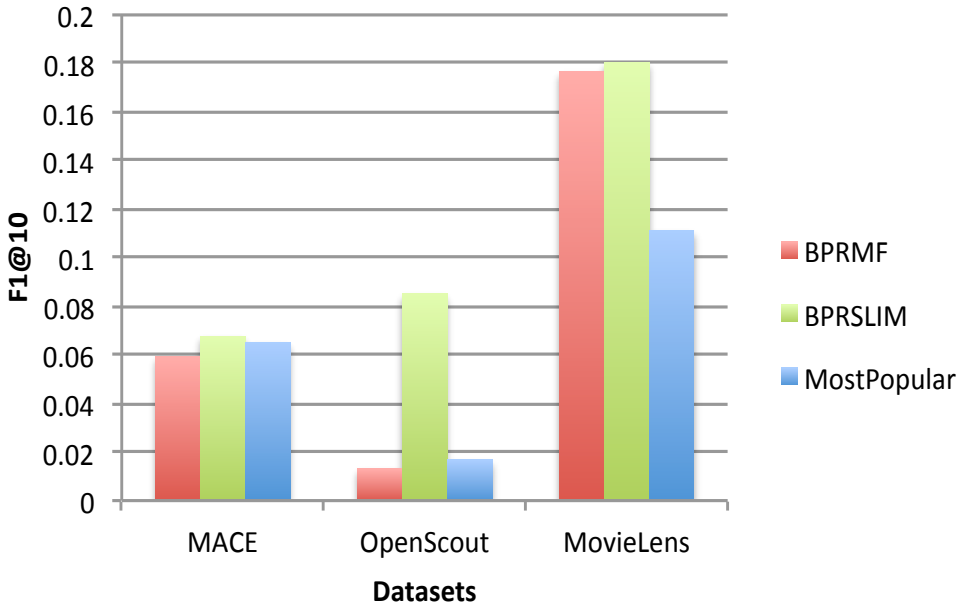
MACE



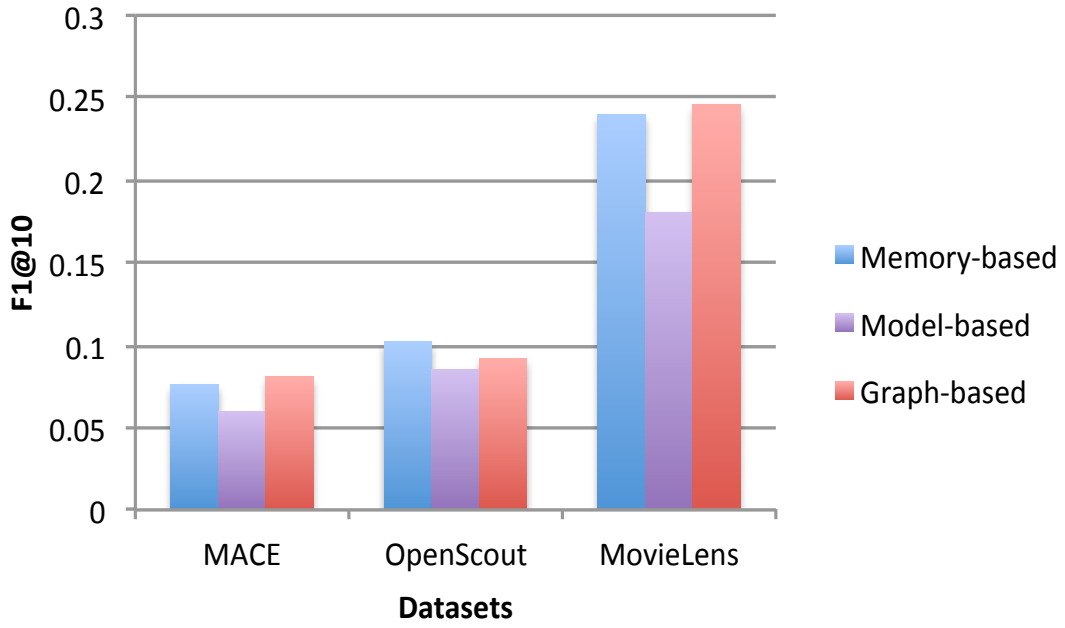
MovieLens

X-axis: Size of neighbors
Y-axis: F1@10

Experimental study
3. Results
3.2. Model-based CFs



Experimental study
3. Results
3.3. Final step



OpenU MOOC platform

General information	OpenU courses (aka masterclasses)	105
	OpenU students	3462
	Total completed activities	42829
	Total messages	19137

Method	k	Pre@5	Pre@10	Recall@5	Recall@10
User-based Jaccard kNN (UB1)	k=5	0.42606	0.24104	0.90788	0.94808
	k=10	0.43609	0.24549	0.91970	0.95683
	k=20	0.43702	0.24707	0.91756	0.95642
	k=30	0.43998	0.24769	**0.92032**	**0.95826**
	k=50	0.43620	0.24591	0.91323	0.95230
	k=100	**0.44284**	**0.24981**	0.91756	0.95473
User-based Cosine kNN (UB2)	k=5	0.43078	0.24302	0.91978	0.94376
	k=10	0.44166	0.24755	0.92289	0.95825
	k=20	0.44492	0.25093	0.92505	0.96219
	k=30	**0.44663**	0.25164	**0.92506**	**0.96244**
	k=50	0.44563	**0.25193**	0.92095	0.95988
	k=100	0.44284	0.24981	0.91756	0.95473
Graph-based CF (UB3)	k=5	0.44766	0.25102	0.87819	0.88342
	k=10	0.45611	0.25341	0.87983	0.87525
	k=20	0.45683	0.25675	0.87902	0.88230
	k=30	0.45842	0.26008	0.88060	0.88644
	k=50	0.45705	0.26465	0.88357	0.88976
	k=100	**0.49008**	**0.26833**	**0.88790**	**0.89058**
Item-based Jaccard kNN (IB1)	k=5	**0.15968**	**0.13857**	**0.30803**	**0.47686**
	k=10	0.12419	0.12182	0.21029	0.39230
	k=20	0.08152	0.09076	0.10608	0.25036
	k=30	0.04956	0.06156	0.03782	0.10692
	k=50	0.01441	0.01901	0.00878	0.02196
	k=100	0.0	0.0	0.0	0.0
Item-based Cosine kNN (IB2)	k=5	**0.15796**	**0.14105**	**0.28232**	**0.48504**
	k=10	0.12704	0.12906	0.19357	0.40850
	k=20	0.07601	0.09283	0.07728	0.24361
	k=30	0.05245	0.06278	0.04102	0.11007
	k=50	0.01654	0.02043	0.01001	0.02385
	k=100	0.0	0.0	0.0	0.0

Conclusion

- The aim is to support users of social learning platforms in finding relevant resources
- Graph-based recommender systems can help to deal with sparsity problem in educational domain

Ongoing and Further work

- The graph-based recommender already has been integrated with ODS platform
- Using trust relationships
  - Explicit vs implicit trust relations (accepted paper at #RecSys2015 in Silicon Valley, US, Oct. 2014)
- User study (November- December 2014) to evaluate user satisfaction
  - Novelty, serendipity, diversity: very important for learning domain

@SoudeFazeli

soude.fazeli[at]ou[dot]nl