Item-based vs User-based Collaborative Recommendation Predictions

Joel Azzopardi

Department of Artificial Intelligence
Faculty of ICT
University of Malta

joel.azzopardi@um.edu.mt

September 2017

Overview

The Problem

The Problem

- Information Overload
- **Information Retrieval** – user 'pulls' relevant information after submitting query.
- **Recommendation Systems** – system 'pushes' relevant information to the user based on user model.
- Main Challenge: handling large amounts of data *efficiently* and *effectively*.

Background

Recommendation Approaches

- **Content-based techniques** – recommendation is performed on the basis of similarity between the content of the different items (documents).
  - Need to extract features from the different items (documents).
  - Does not suffer from *new user/item* problem, and from *sparse matrix* problem.
  - Suitable for items with high turn-over (e.g. news).
- **Collaborative techniques** – recommendation is performed on the basis of what other 'similar' users have found useful.
  - Does not use features from the items/documents.
  - Need to have substantial user-item rating overlap.

Collaborative Recommendation

- More effective than content-based approaches.
- Exploit the fact that humans enjoy sharing their opinions with others.
- 2 main types:
  - **User-based** – an item's recommendation score for a user is calculated depending on that items' ratings by other similar users
  - **Item-based** – item's rating is predicted based on how similar items have been rated by that user.

Research Questions

Research Questions

- What will be the performance of an *ensemble* system combining both *user-based* and *item-based* approaches?
- What is the effect of *Latent Semantic Analysis (LSA)* applied to the collaborative recommendation algorithms?
- What is the *optimal neighbourhood size* for the different collaborative recommendation setups?

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{S} \cdot \mathbf{D}^T$$

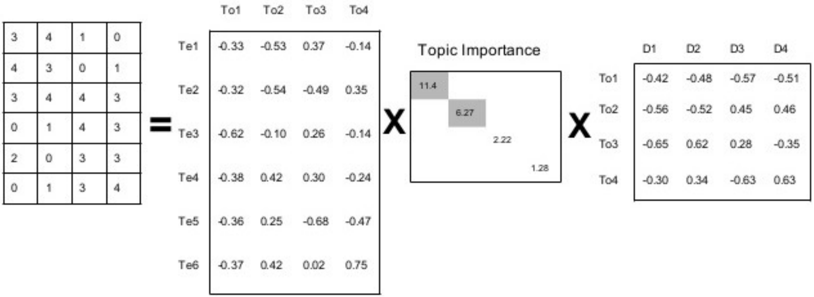


Figure 1 : Latent Semantic Analysis Process, from: *http://www.slideshare.net/vitomirkovanovic/topic-modeling-for-learning-analytics-researchers-lak15-tutorial*, September 2016

Methodology

Collaborative Recommendation Algorithm

```
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
```

Methodology

- Algorithm is based on *k Nearest Neighbours* (*kNN*).
  - Votes are weighted according to neighbours' similarities.
- Use of:
  - *User pair-wise* similarity matrix in user-based recommendation.
  - *Item pair-wise* similarity matrix in item-based recommendation.
- In LSA, these similarity matrices are decomposed, and only the top dimensions are considered.
- Ensemble algorithm:
  - Separate candidate user-item ratings are obtained from *user-based* and *item-based* algorithms.
  - Lists are merged together.
  - Predicted recommendation score is set to the highest weighted candidate score in the merged list.

Evaluation

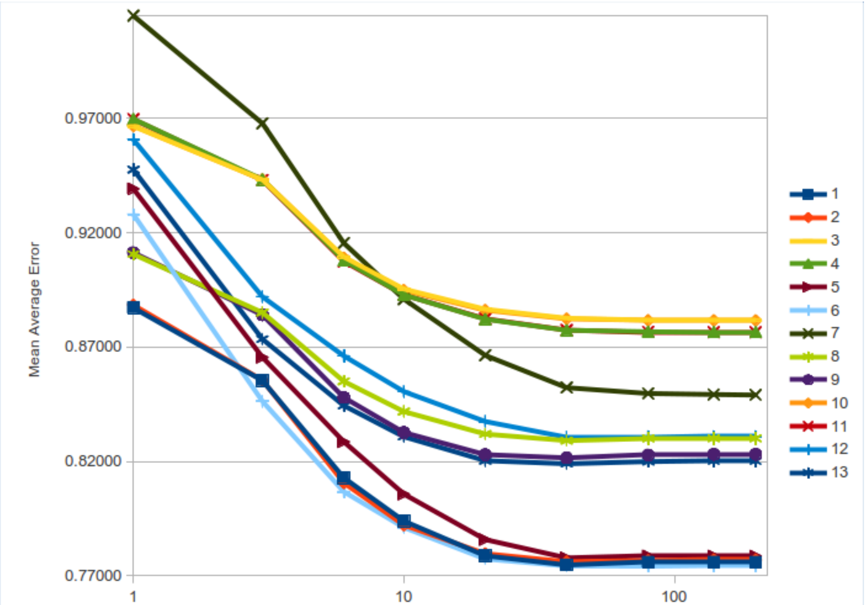
Evaluation Dataset

- *MovieLens 1M* dataset
  - 1000209 ratings
  - 3883 movies
  - 6040 different users
- Split into 80% / 20% for training and testing.
  - Training set consists of the oldest 80% ratings for each user.
  - Rest into test set.
- Metric used: *Mean Average Error* (MAE)
- Neighbourhood sizes: 1, 2, 3, 6, 10, 20, 40, 80, 140, 200

System Configurations Evaluated

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

Results



Mean Average Error
0.97000
0.92000
0.87000
0.82000
0.77000
1
10
100
1
2
3
4
5
6
7
8
9
10
11
12
13

Conclusions

Comparison of the Different Setups

- *Item-based* recommenders perform considerably better than the *user-based* ones.
- LSA has a beneficial effect on *user-based* recommendations, but an overall negative effect on the *item-based* recommendations.
- *Ensemble* system that uses LSA gives best (albeit slightly) results across practically all neighbourhood sizes.

Optimal Neighbourhood Size

- Optimal neighbourhood size seems to be around 40.
- *Item-based* recommenders are most effective with a neighbourhood size of 40 with a slight deterioration of results for larger sizes.
- Performance of *user-based* recommenders keeps improving (albeit very slightly) as neighbourhood sizes are increased.
- Ensemble algorithm that uses LSA obtains the best results with a neighbourhood size of 80, and results degrade slightly with larger neighbourhoods.

Future Work

- Investigation of the different methods of how content-type features may be incorporated in collaborative systems.
- Recommendation over big-data: how to perform distributed recommendation over multiple datasets and merging the recommendation scores.