DATA MINING TECHNOLOGY FOR BUSINESS AND SOCIETY HW2

Cecilia Martinez Oliva, Giorgio Giannone

Contents

Introduction

In this work we have to solve recommendation problems using link-analysis techniques. First we test the method with a 5-fold cross validation and then we create a movie recommendation for groups of users.

1. Testing the method

We have 942 users who rated 1682 movies, 100000 ratings overall (released in 4/1998). The data are splitted in 5 different train/test for the validation of the Topic-Specific-PageRank for the recommendation.

Train - Test_1

	Training Graph_1	**Test Graph_1**	**Compressed Item-Item Graph_1**
Num User Nodes	943	459	1650
Num Item Nodes	1650	1378	
Num Edges	80000	19968	1360425

Train - Test_2

	Training Graph_2	**Test Graph_2**	**Compressed Item-Item Graph_2**
Num User Nodes	943	653	1648
Num Item Nodes	1648	1386	
Num Edges	80000	19964	1357128

Train - Test_3

	Training Graph_3	Test Graph_3	Compressed Item-Item Graph_3
Num User Nodes	943	869	1650
Num Item Nodes	1650	1391	
Num Edges	80000	19964	1360425

Train - Test_4

	Training Graph_4	Test Graph_4	Compressed Item-Item Graph_4
Num User Nodes	943	923	1660
Num Item Nodes	1660	1372	
Num Edges	80000	19973	1376970

Train - Test_5

	Training Graph_5	Test Graph_5	Compressed Item-Item Graph_5
Num User Nodes	943	927	1660
Num Item Nodes	1650	1375	
Num Edges	80000	19964	1360425

2. The Average-Normalized-Discounted-Cumulative-Gain

For each user, for each couple of train/test we compute the nDCG and the final score is 0.94324. We underline though that the minimum score is 0.83637 (see minimum_discounted_cumulative_ gain in the "`Network_Based_Recommendation_System_FUNCTIONS.py`" file).

3. Movies recommendation for groups

Finally we have to recommend movies for groups of users where each user has a different weight.

We suppose that if the rating is bigger than 2 the user likes the movie, otherwise he doesn't and his preference for that movie will be:

- r*1/w if the user's rating for the movie is less than 3
- r*w otherwise

To compute the preference vector for the PageRank we sum the preference vector of each user belonging to the group and divide by the total sum of the components (for the normalization). Then we have to compute the PageRank and to sort the list of recommended movies.

We suppose that users don't want to watch again movies already rated. However, in the code, line 141 can be commented (see "## COMMMENT .... ##"). For instance, in the case of your girlfriend who wants to watch Titanic for the 30th time or your brother The gladiator for the 100th. ':()