Data Mining Technology for Business and Society

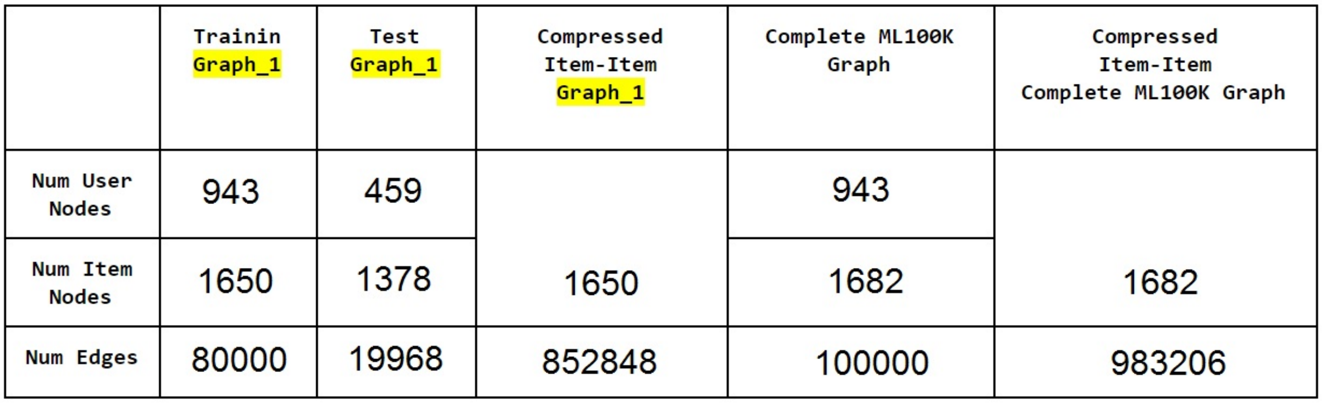
Homework # 3

1536242 - Paolo Tamagnini

June 2016

1 Table

Table containing a simple statistic on the used data-set:



	Trainin Graph_1	Test Graph_1	Compressed Item-Item Graph_1	Complete ML100K Graph	Compressed Item-Item Complete ML100K Graph
Num User Nodes	943	459	1650	943	1682
Num Item Nodes	1650	1378		1682	
Num Edges	80000	19968	852848	100000	983206

2 Results

The average normalized DCG lower bound is:

```
average_normalized_DCG_LOWER_BOUND_over_all_training_set_test_set_couples
                 = 0.821618754446
```

The average normalized DCG is:

```
average_normalized_DCG_for_PERSONAL_recommendation_over_all_training_set_
test_set_couples = 0.937451807435
```

3 Recommendation for Groups

We want to make a recommendation for a group G of users. Each user u in the group has a weight w, proportional on how his/her taste influence the recommendation for the entire group. The user has also ratings r on different movies x, that we will use to find his movie taste.

To compute a recommendation for the entire group I could have taken two paths.
The first path was to compute a personalized page rank for each member of the group and then to merge the results using the weight w of each member. This path is really basic and imprecise.
The second path, the one I took, was to compute directly a personalized page rank on the entire group. To do that I needed a preference vector able to represents the ratings r and the weights w of each member of the group.

I decided to compute such overall preference vector by compiling the relative dictionary *overallPrefVector*. Such dictionary describes the function $d(x)$, so that for each movie key x the value will be:

$$d(x) = \sum_{u \in G} r_{u,x} \cdot w_u$$

If a user in the group didn't rate the movie, it will be like if it rated it 0.

The page rank will take such overall preferential vector as input and, after normalizing it, will compute the score for each movie. The personalization will be due the teleporting probabilities, which will point just towards movies rated from at least one user of the group. Such probabilities will be proportional to the number of members in the group who rated the relative movies, to such members influences w and ratings r.

Once we have the personalized page rank score for each movie in the data-set, we take away any of the ones rated from at least a user in the group, since they cannot be recommended. Then we sort our page rank score in descending order. The list of movies we recommend will have as first the most recommended and as last the least recommended.