Recommender Systems

TIETS43

Fairness in Group Recommendations in the Health Domain

Kostas Stefanidis
kostas.stefanidis@uta.fi

Fall 2017
https://coursepages.uta.fi/tiets43/

Motivation (1/2)

The problem:

- Approximately 72 percent of Internet users seek health information online
- It is very difficult for a patient to accurately judge the relevance of some information to his own case
- Doctors and caregivers on the other hand have not the time to provide detailed information to individuals and to small groups of patients
- A real requirement from doctors in iManageCancer EU project: “A tool to automatically recommend useful, high quality documents to similar groups of patients”

Motivation (2/2)

We target at improving the opportunities that patients have to inform themselves via the Web about their disease and possible treatments, and providing to them personalized information

- Deliver relevant information to patients
  - How?
    - Based on their current profile as represented in their personal healthcare record (PHR) data.
- Ensure the quality of the presented information
  - How?
    - Giving medical experts the chance to control the information that is given.
    - Identify once a high quality corpus of interesting documents and those will be automatically recommended to small group of patients according to their problems

Goals

- A model for group recommendations
  - Both group recommendations and notions of fairness are mostly unexplored in the health domain
- Collaborative filtering approach
- Explore different similarity measures that take into consideration specific health-related information to identify the correct set of similar users for a user in question
- Use different designs for aggregating the recommendations for the group
- Suggestions highly related and fair to the patients of the group

Outline

- Single User Recommendations
  - Single User Rating Model
  - User Similarities
  - Example
- Group Recommendations
  - Group Rating Model
  - Fairness in Group Recommendations
  - Aggregations Designs
- Dataset
- Evaluation

Single User Recommendations

Recommendation Model

$I = \{i_1, i_2, \ldots, i_d\}$: set of items

$U = \{u_1, u_2, \ldots, u_n\}$: set of users

$rating(u,i) \in [1,5]$: the preference / rating of a user $u \in U$ for an item $i \in I$

U(i): the subset of users that rated an item *i*

I(u): the subset of items rated by a user *u*

But, typically users rate only a few items (and |I| is too high!)

For an unrated item *i*, estimate its relevance for a user *u*

$\longrightarrow$ $relevance(u,i)$

Single User Recommendations (1/3)

How to estimate *relevance(u,i)?*

- Collaborative filtering idea: use preferences of similar users to u to produce relevance scores for the items unrated by u
  - Similarity is estimated in terms of some similarity / distance function
  - P_u : the set of similar users to u, or peers, taking into consideration a distance threshold

Single User Recommendations (2/3)

Relevance computation based on peers

$$relevance(u,i) = \frac{\sum_{u' \in (P_u \cap U(i))} S(u,u') r(u',i)}{\sum_{u' \in (P_u \cap U(i))} S(u,u')}$$

After estimating the relevance scores of all unrated items for a user u, the items A_u with the top-k relevance scores are suggested to u.

Similarity Between Users

Two ways to find similarities between users

- Using the ratings of users
  - Pearson correlation

- Using the Personal Health Profile (PHR) of users
  - Semantic Similarity function

Rating Similarity Function

Similarity based on user ratings

Assumption: if two users have rated documents in a similar way, then we can say that they are similar, since they share the same interests

$$RatS(u,u') = \frac{\sum_{i \in X} (r(u,i) - \mu_u)(r(u',i) - \mu_{u'})}{\sqrt{\sum_{i \in X} (r(u,i) - \mu_u)^2} \sqrt{\sum_{i \in X} (r(u',i) - \mu_{u'})^2}}$$

Pearson correlation: where $X = I(u) \cap I(u')$ denotes the items that both users have rated, *r(u,i)* is the rating *u* gave to *i*, and μ_u is the mean of the ratings in *I(u)*

Semantic Similarity Function

Similarity based on semantic information

Considers the users health problems: represent health problems utilizing the ICD10 ontology

Two steps approach:

- Find the similarity between pairs of “problems” of two users
- Calculate the overall similarity of the users

Similarity between two health problems (1/7)

Step One: Find similarity between two health problems

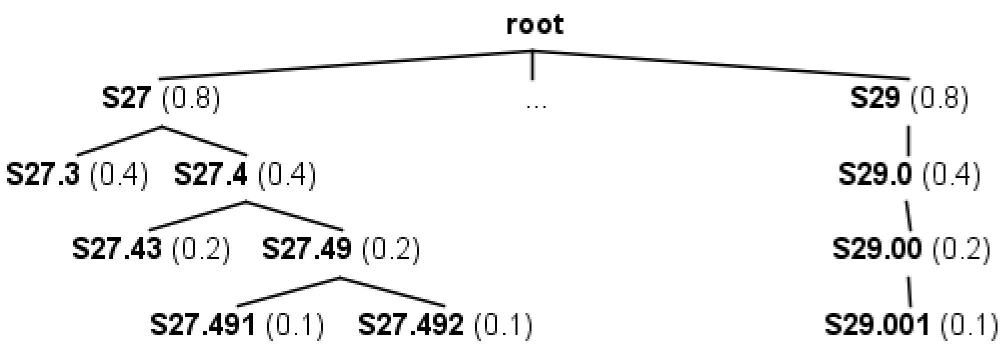
Code Id	Description	Level
S27	Injury of other and unspecified intrathoracic organs	1
S29	Other and unspecified injuries of thorax	1
S27.3	Other injury of bronchus, unilateral	2
S27.4	Injury of bronchus	2
S27.43	Laceration of bronchus	3
S27.49	Other injury of bronchus	3
S27.491	Other injury of bronchus, unilateral	4
S27.492	Other injury of bronchus, bilateral	4

Sibling nodes share different similarity, based on the level they reside

Similarity between two health problems (2/7)

We utilize the ontology tree and assign weights to nodes.

$$weight(A) = w * 2^{\max Level - level(A)}$$

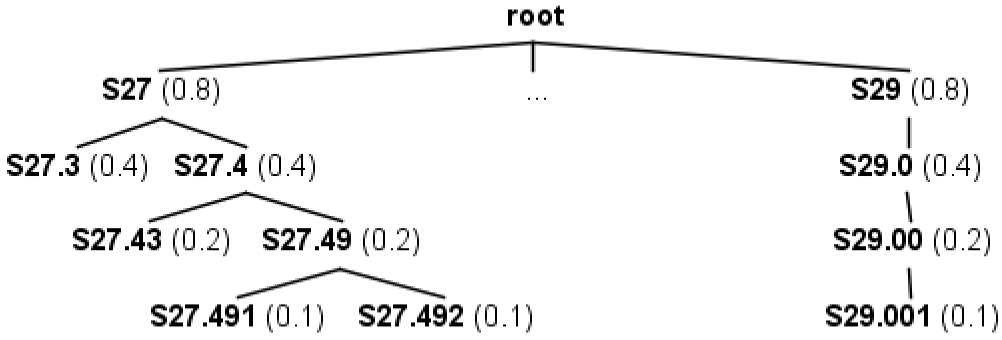


w = 0.1

Similarity between two health problems (3/7)

To find similarity between two nodes:

1. Find their Least Common Ancestor
2. Calculate the distance between each node and the LCA
3. Accumulate and normalize those distances



w = 0.1

Similarity between two health problems (4/7)

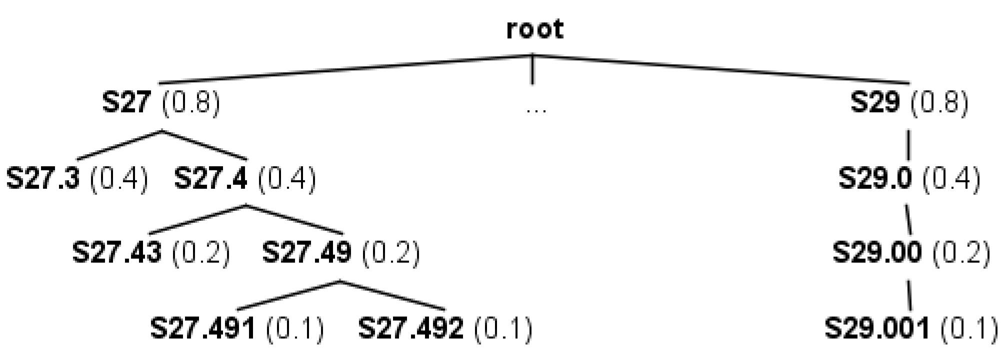
Find their Least Common Ancestor

We allow a node to be descendant of itself.

Examples:

LCA(S27.4, S27.491) = S27.4

LCA(S27, S29.0) = root



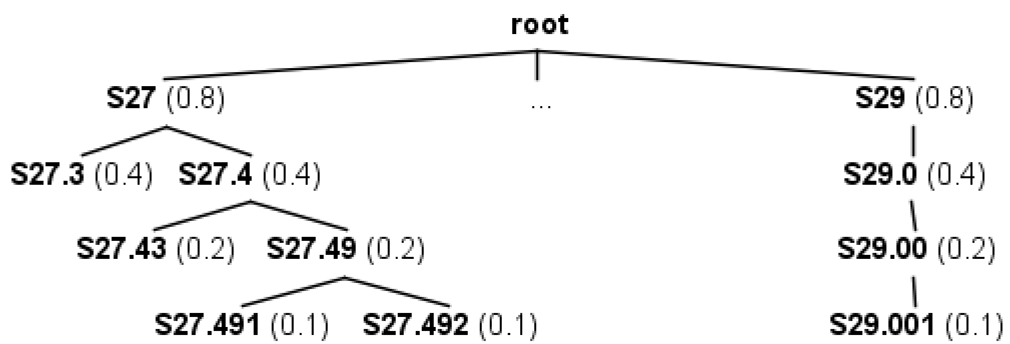
w = 0.1

Similarity between two health problems (5/7)

<u>Calculate the distance between each node and the LCA</u>

1. Find the path connecting the two nodes
   - In the path that connect the nodes A and B, we include A, but not include B. Example: path(S27.491, S27.4) = {S27.491, S27.49}
2. Accumulate the weights of the nodes in the path

<u>Examples:</u>
dist(S27.491, S27.4) = 0.3
dist(S27.492, root) = 1.5



w = 0.1

Similarity between two health problems (6/7)

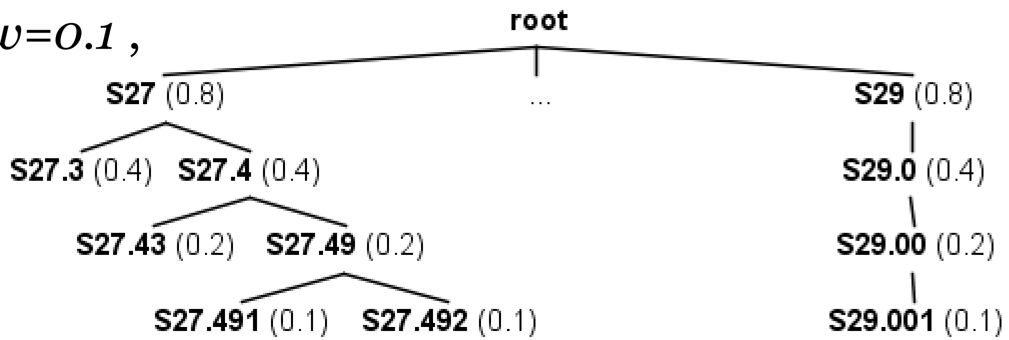
<u>Accumulate and normalize those distances</u>

The final similarity score of two nodes A and B, with LCA(A,B) = C is:

$$simN(A,B) = 1 - \frac{dist(A,C) + dist(B,C)}{maxPath * 2}$$

where *maxPath* is the maximum distance in the tree.

- In the ICD10 ontology tree and *w=0.1* , maxPath = 1.5

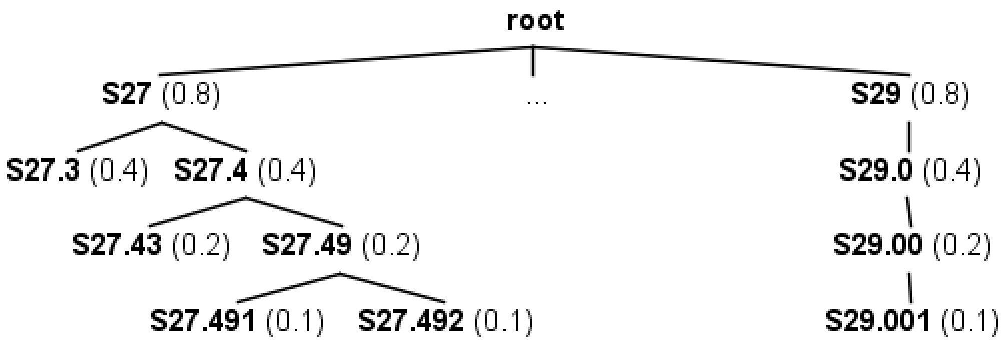


w = 0.1

Similarity between two health problems (7/7)

Examples:

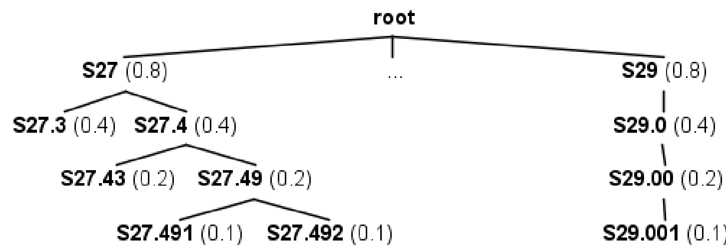
Node A	Node B	LCA(A,B)	simN(A,B)
S27	S29	root	1-(0.8+0.8/3)=0.47
S27.43	S27.49	S27.4	1-(0.2+0.2/3)=0.87
S27.492	S27.49	S27.49	1-(0 + 0.1/3) = 0.97
S27.491	S27.492	S27.49	1-(0.1 + 0.1 /3) = 0.93



w = 0.1

Overall Similarity between two users

Users have multiple of health problems in their profile.
So the final similarity of two users:

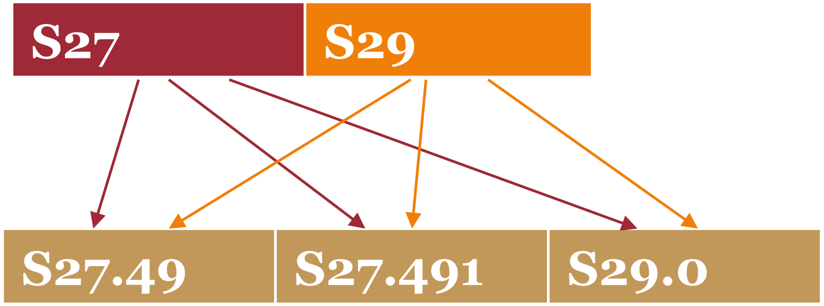


Assume users *x*, *y*.

1. Compare a health problem from x's profile with all the health problem from y's profile.
2. Choose the one with the maximum similarity.
3. Do the same for all problem from x's profile.
4. Accumulate and then average those.

simN(S27, S27.49) = 1-(0+0.6)/3 = 0.8
simN(S27, S27.491) = 1-(0+0.7)/3 = 0.77
simN(S27, S29.01) = 1-(0.8+1,2)/3 = 0.34

x's health problems



y's health problems

simN(S29, S27.49) = 1-(0.8+1.4)/3 = 0.6
simN(S29, S27.491) = 1-(0.8+1.5)/3 = 0.23
simN(S29, S29.0) = 1-(0+0.4)/3 = 0.86

SemS(x,y) = (0.8 + 0.86) / 2 = 0.83

Single User Recommendation-Example (1/3)

#	Name	Problems	Ratings – DocId(score)
1	John Smith	S27.492	10(4), 15(5), 20(3), 30(3), 22(4), 23(2), 36(5)
2	Mary Jane	S27.4, S27.49	10(3), 30(5), 16(3), 19(4), 18(3), 17(4), 35(4)
3	Thomas Murphy	S27.3	16(5), 20(5), 30(2), 25(3), 22(3), 17(5), 36(5)
4	Scott Wilson	S29.00	25(5), 45(2), 19(2), 17(5), 31(5), 35(5)
5	Mia Brown	S29, S27.49	15(4), 20(4), 30(3), 45(4), 22(5), 36(5), 18(5)

Rating Similarity Function: $RatS(u,u') = \dfrac{\sum_{i \in X}(r(u,i)-\mu_u)(r(u',i)-\mu_{u'})}{\sqrt{\sum_{i \in X}(r(u,i)-\mu_u)^2}\sqrt{\sum_{i \in X}(r(u',i)-\mu_{u'})^2}}$

For user John Smith: $\mu_u = 3.71$

For user Mia Brown: $\mu_u = 4$

Common documents: 15, 22, 36

RatS(John Smith, Mia Brown) = 0.897

User	RatS
Mary Jane	-0.993
Thomas Murphy	0.389
Scott Wilson	0
Mia Brown	0.897

Single user Recommendation-Example (2/3)

#	Name	Problems	Ratings – DocId(score)
1	John Smith	S27.492	10(4), 15(5), 20(3), 30(3), 22(4), 23(2), 36(5)
2	Mary Jane	S27.4, S27.49	10(3), 30(5), 16(3), 19(4), 18(3), 17(4), 35(4)
3	Thomas Murphy	S27.3	16(5), 20(5), 30(2), 25(3), 22(3), 17(5), 36(5)
4	Scott Wilson	S29.00	25(5), 45(2), 19(2), 17(5), 31(5), 35(5)
5	Mia Brown	S29, S27.49	15(4), 20(4), 30(3), 45(4), 22(5), 36(5), 18(5)

<u>Semantic Similarity Function</u>

For user John Smith

User	Partial Scores	SemS
Mary Jane	S27.492 - S27.4 = 0.9	0.967
	S27.492 - S27.49 = 0.967	
Thomas Murphy	S27.492 - S27.3 = 0.634	0.634
Scott Wilson	S27.492 - S29.01 = 0.034	0.034
Mia Brown	S27.492 - S29 = 0.234	0.967
	S27.492 - S27.49 = 0.967	

Single user Recommendation-Example (3/3)

$$relevance(u,i) = \frac{\sum_{u' \in (P_u \cap U(i))} S(u,u') r(u',i)}{\sum_{u' \in (P_u \cap U(i))} S(u,u')}$$

User	Sim	Peers	Recommendation List
John Smith	RatS	Thomas Murphy, Mia Brown	18(5), 16(5), 17(5), 28(3), 31(3), 25(3), 45(2)
	SemS	Mary Jane, Thomas Murphy, Mia Brown	18(5), 17(4.4), 35(4), 19(4), 16(3.8), 18(3), 25(3), 28(3), 31(3), 45(2)
Mary Jane	RatS	Scott Wilson	25(5), 31(5), 45(2)
	SemS	John Smith, Thomas Murphy, Mia Brown	15(5), 36(5), 22(4.1) 20(3.8), 25(3), 28(3), 31(3), 23(2), 45(2)

<u>John Smith with SemS</u>

Document 17 voted by:
- Mary Jane - 4
- Thomas Murphy - 5

SemS(John Smith, Mary Jane) = 0.967
SemS(John Smith, Thomas Murphy) = 0.634

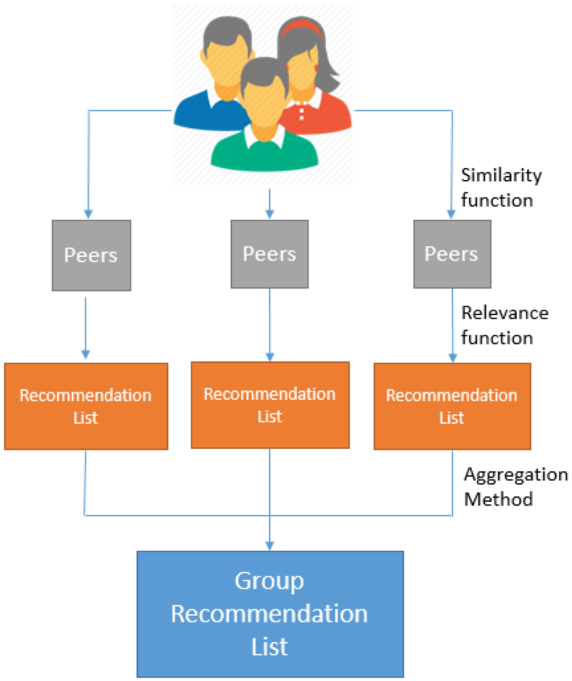
$$relevance(JohnSmith, 17) = \frac{0.967 * 4 + 0.634 * 5}{0.967 + 0.634} = 4.4$$

Group Recommendations

Group Rating Model

Recently, group recommendations have received considerable attention!

Make recommendations to groups of users instead of single users



How:

- Estimate the relevance scores of the unrated items for each user in the group

- Aggregate these predictions to compute the suggestions for the group

$$relevanceG(G, i) = Aggr_{u \in G}(relevance(u, i)).$$

Different designs regarding the aggregation method can be used – each one carries different semantics

Fairness in Group Recommendations (1/3)

- Our goal is to locate suggestions that include data items highly relevant and fair to the patients of a group
  - We actually provide suggestions to a caregiver responsible for a group of patients
- The package selection should be fair to all users in the group!
  - I.e., we do not want to have any user *u* that is the least satisfied user in the group for all items in the recommendations list, <u>that is, all items are **not** related to *u*</u>

Fairness in Group Recommendations (2/3)

We consider a fairness measure that evaluates the goodness of recommendations

Given a user u and a set of recommendations D, we define the degree of fairness of D for u as:

$$fairness(u, D) = \frac{|X|}{|D|},$$

where, $X = A_u \cap D$ and A_u are the items with the top-k relevant scores for u

Intuitively, the fact that recommendations contain a highly relevant item to u, makes both u and his caregiver tolerant to the existence of other items not highly related to the user, considering that there are other members in the group who may be related to these items

Fairness in Group Recommendations (3/3)

The fairness of a set of recommendations D for a set of users G is defined as follows.

$$fairness(G, D) = \frac{\sum_{u \in G} fairness(u, D)}{|G|}.$$

The fairness - aware value of D for G is:

$$value(G, D) = fairness(G, D) \cdot \sum_{i \in D} relevanceG(G, i).$$

Aggregation Designs

$$relevanceG(G, i) = Aggr_{u \in G}(relevance(u, i)).$$

We consider four Aggregation Designs that can be divided into two groups:

1. Score Based Methods
   1. Minimum
   2. Average

2. Rank Based Methods
   1. Borda
   2. Fair

Remember: we have already found the individual recommendation lists for the members of the group

Minimum

$$relevanceG(G, i) = \min_{u \in G}(relevance(u, i)).$$

Assuming we have two members in the group x, y and their corresponding recommendation lists

x's list

DocId	Score
10	5
20	4
30	3

y's list

DocId	Score
20	4
10	2
30	1

We consider that strong user preferences act as a veto

Using the Minimum Aggregation method the group recommendation list will be:

DocId	Score
20	4
10	2
30	1

min(5,2) = 2

Average

$$relevanceG(G,i) = \sum_{u \in G} relevance(u,i)/|G|$$

Having the same assumption as before:

We focus on satisfying the majority of the group members

x's list

DocId	Score
10	5
20	4
30	3

y's list

DocId	Score
20	4
10	2
30	1

Using the Average Aggregation method the group recommendation list will be:

DocId	Score
20	4
10	3.5
30	2

(5+2)/2=3.5

Borda

$$points(G,i) = \sum_{u \in G} (k - (p_u(i) - 1)).$$

Having the users x, y:

We consider the number of times a document wins

x's list

DocId	Score	Points
20	4	3
30	3	2
10	2	1

y's list

DocId	Score	Points
10	5	3
20	4	2
30	1	1

Using the Borda Aggregation method the group recommendation list will be:

DocId	Points
20	5
10	4
30	3

Fair (1/2)

Targeting at increasing the fairness of the resulting set of recommendations, we introduce also the Fair method, which consists of **two** phases

First:

- we consider pairs of users in the group
- A data item *i* belongs to the top-k suggestions for a group G, if, for a pair of users $u_1, u_2 \in G, i \in A_{u_1} \cap A_{u_2}$ and *i* is the item with the maximum rank in A_{u_2}

Second:

- If the value of *k* (i.e. the number of items that need to be provided by the group recommendation list) is greater than the items already found
  - We construct the rest of D, by serially iterating the Au of the group members and adding the item with the maximum rank that does not already exists in D

Fair (2/2)

Having the users x, y:

x's list

DocId	Score
20	4
10	3
30	2

y's list

DocId	Score
10	5
50	3
20	1

We target to increase the fairness of the resulting set

For x:
The item with highest score in y's list that also exists in x's is item 10
For y:
The item with highest score in x's list that also exists in y's is item 20

DocId
10
20
30

Dataset

Dataset

- In order to start the group recommendation process we need three things
  - A document corpus
  - A users-PHR dataset
  - A users-ratings dataset

Dataset Acquisition - PHR dataset

- From EMRBots[1] we acquired:

- A 10.000 chimeric patient profiles
  - These profiles contain the health problems for each patient, that are described using the ICD10 ontology.

1 http://www.emrbots.org

Dataset Creation – Document Corpus

- Generate a *numDocs* number of documents, for each first level category of the ICD10 ontology (i.e. for each node that belongs in the first level of the ontology tree)
- For their corresponding keywords we randomly selected *numKeyWords* words from the description of the nodes in each subsequent subtree
- Randomly select a *popularDocs* number of documents that will be the most popular

Parameter Name	Explanation	Value
numDocs	The number of documents created for each different category of health problems, based on the ICD10 ontology tree	270
numKeyWords	The number of randomly selected keywords, attached to each document	10
popularDocs	The number of documents, that will be most popular in each category, in order to simulate a power law distribution	70

Dataset Creation – Ratings Dataset (1/2)

- Divide the patients into groups.
  - *Sparse*: few number of ratings
  - *Medium*: average number of ratings
  - *Dedicated*: a lot of ratings

- Generate items to rate.
  - *healthRelevant*: documents belonging in the same subtree as a user's health problems
  - *nonRelevant*: the rest

- Generate ratings.
  - Randomly select a value from [1,5] to assign to each rating.

Dataset Creation – Ratings Dataset (2/2)

Partitions	Parameter Name	Explanation	Value
Group Partition	Group *sparse*	The number of ratings given by patients in this group is 20 to 100	50% of all patients
	Group *medium*	The number of ratings given by patients in this group is 100 to 250	30% of all patients
	Group *dedicated*	The number of ratings given by patients in this group is 250 to 500	20% of all patients
Scores Partition	One	The number of ratings that have as value 1	20% of all ratings
	Two	The number of ratings that have as value 2	10% of all ratings
	Three	The number of ratings that have as value 3	30% of all ratings
	Four	The number of ratings that have as value 4	20% of all ratings
	Five	The number of ratings that have as value 5	20% of all ratings
Ratings Partition	*healthRelevant*	The number of documents each user will rate that are relevant to some health problem he/she suffers from	20% of ratings from each user
	nonRelevant	The number of documents that each user will rate that are not relevant to any of his/her health problems	80% of ratings from each user

Evaluation

Similarity Evaluation Measures

For evaluation we use the following measures:

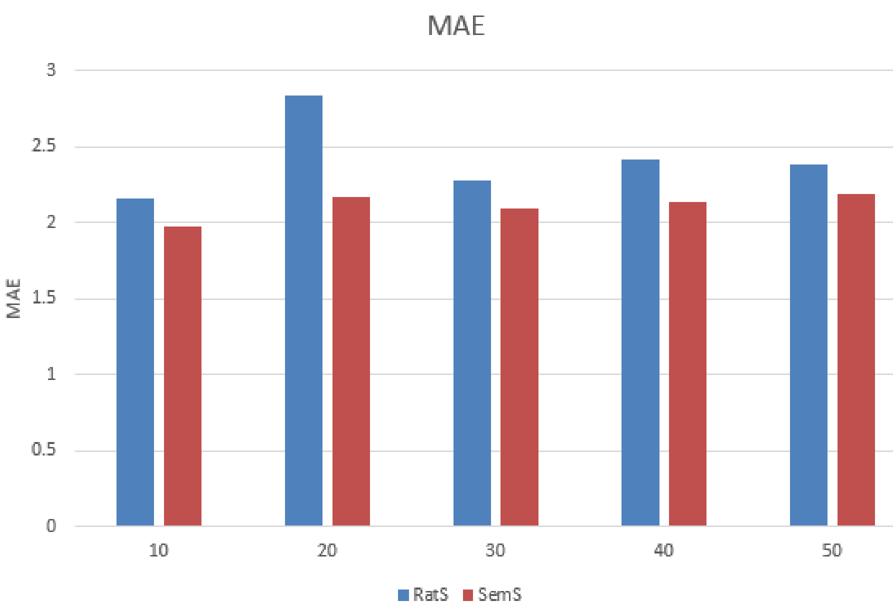
Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |predicted_i - actual_i|$$

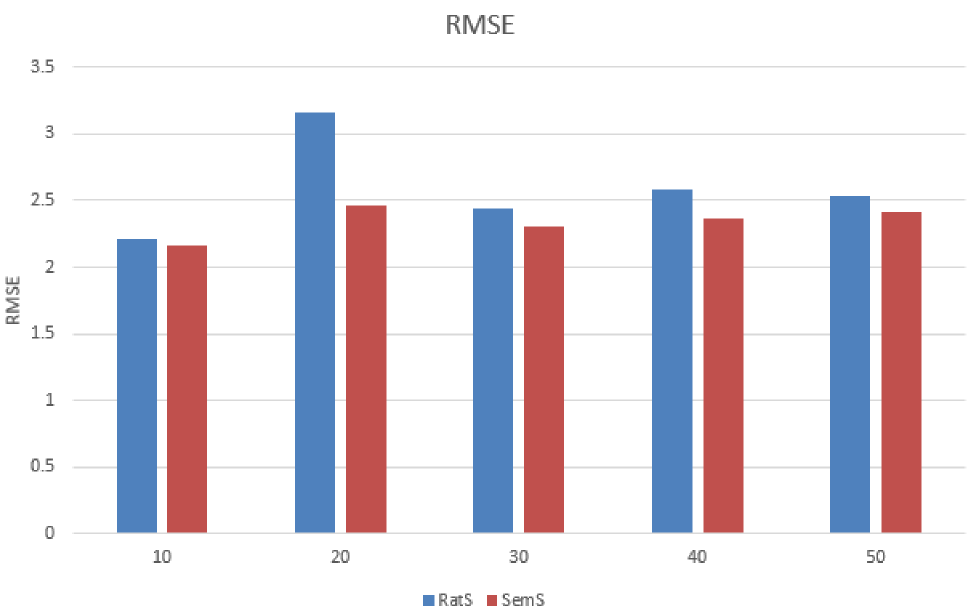
Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (predicted_i - actual_i)^2}$$

RatS vs SemS (1/2)



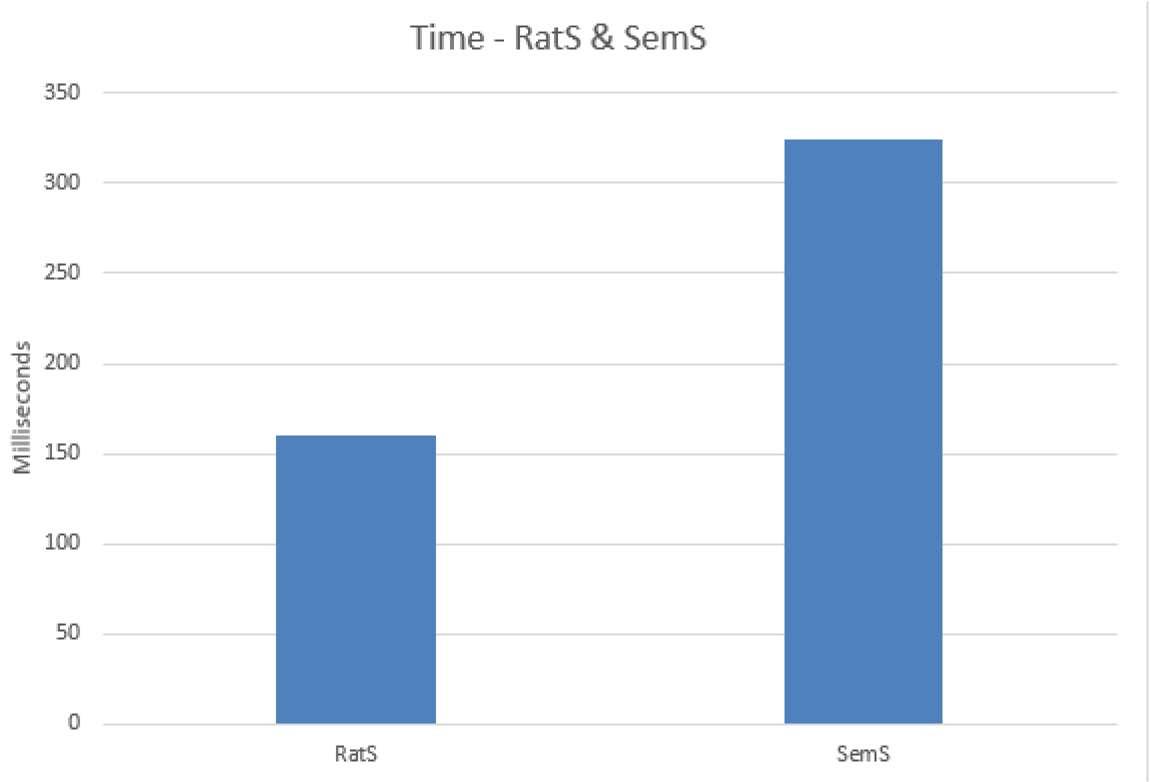
K Number of predictions



K Number of predictions

Tests done for 100 different users

RatS vs SemS (2/2)



Test done for 100 different users

Aggregation Evaluation Measures

In order to evaluate the different aggregation methods, we compare the members individual recommendation lists with that of the group's.

We utilize two distances:

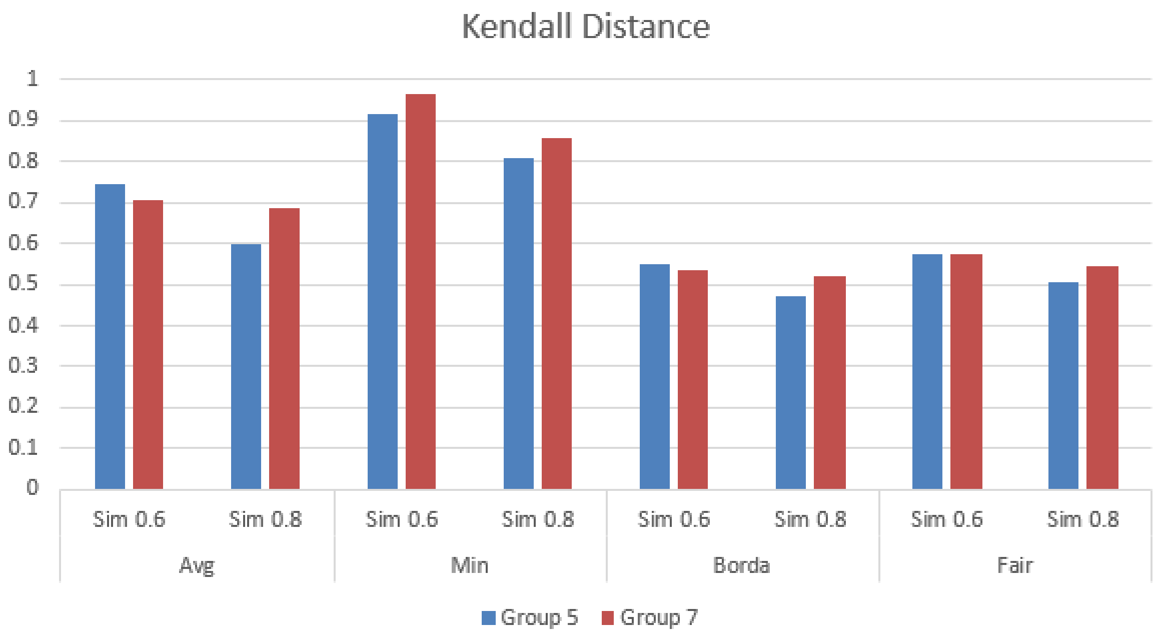
Spearman's footrule distance: is the absolute difference between the ranks assigned to an item in each list.

$$S(t_1, t_2) = \sum_{i=1}^{n} |t_1(i) - t_2(i)|$$

Kendall tau distance: is a metric that counts the number of pairwise disagreements between two ranking lists

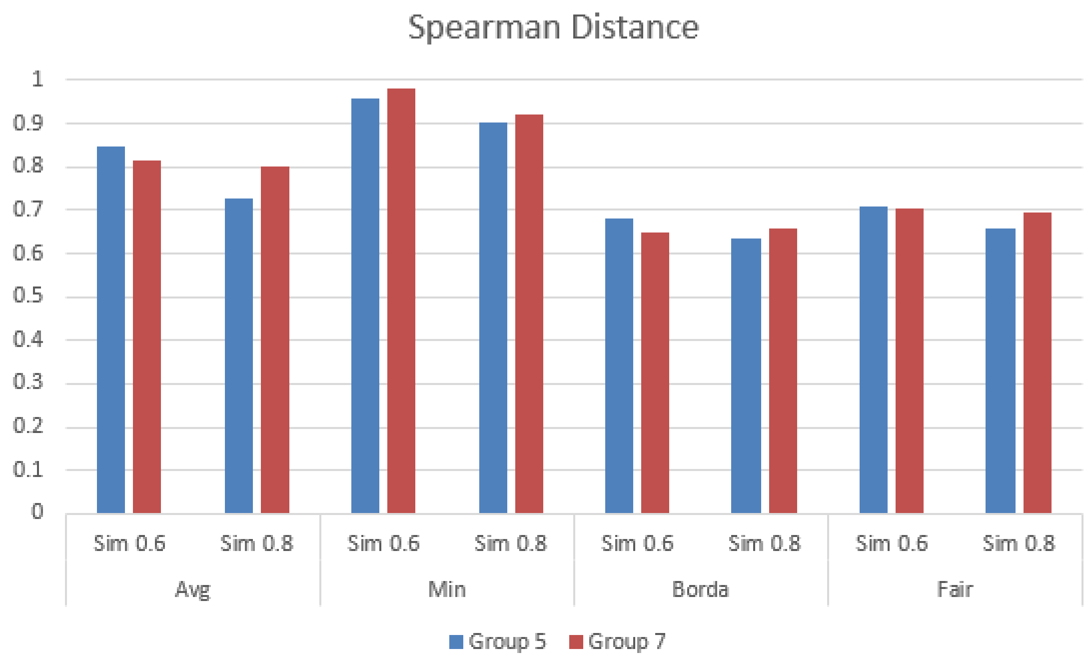
$$K(t_1, t_2) = |\{(i, j) : i < j, (t_1(i) < t_1(j) \wedge t_2(i) > t_2(j)) \\ \vee (t_1(i) > t_i(j) \wedge t_2(i) < t_2(j))\}|$$

Kendall tau Distance



Test done for 10 different groups

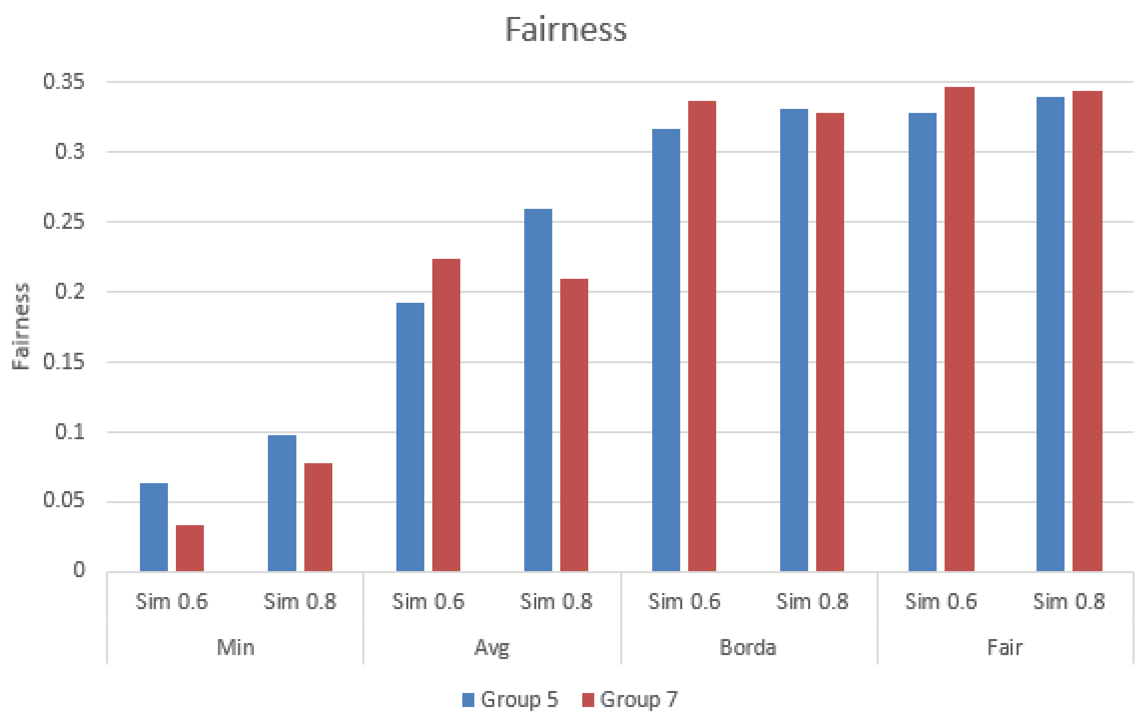
Spearman footrule Distance



Test done for 10 different groups

Fairness

$$fairness(G, D) = \frac{\sum_{u \in G} fairness(u, D)}{|G|}.$$



Test done for 10 different groups

Value (1/3)

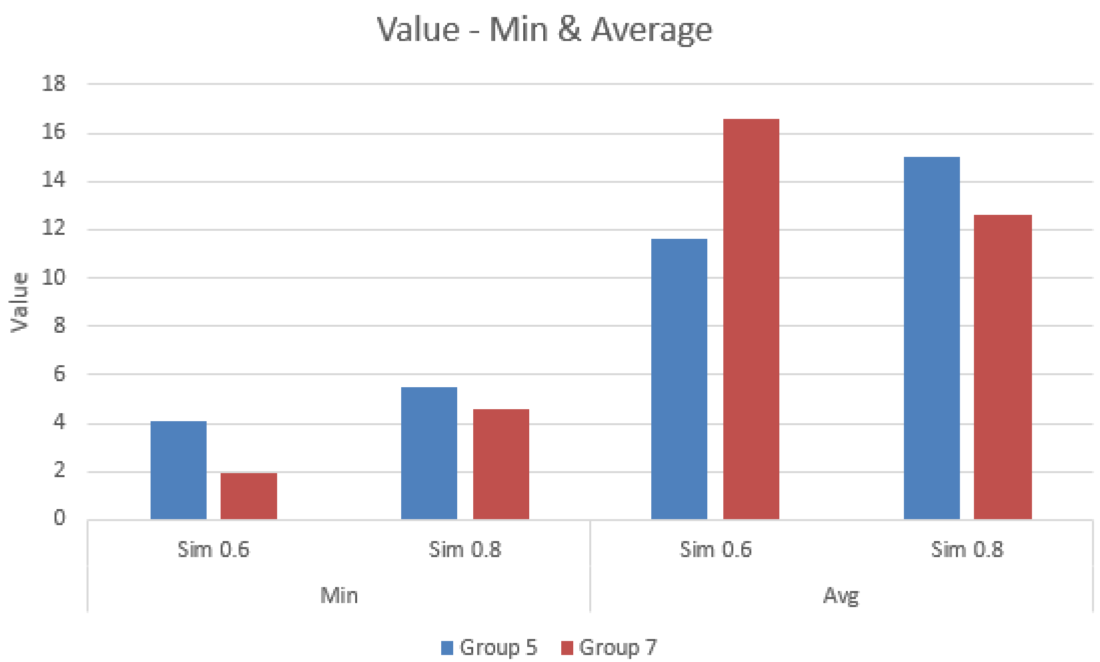
$$value(G, D) = fairness(G, D) \cdot \sum_{i \in D} relevanceG(G, i)$$

Difference between score-based and rank-based aggregation designs:

- In score-based we calculate the actual score of an item.
- In rank-based we calculate the rank of an item.

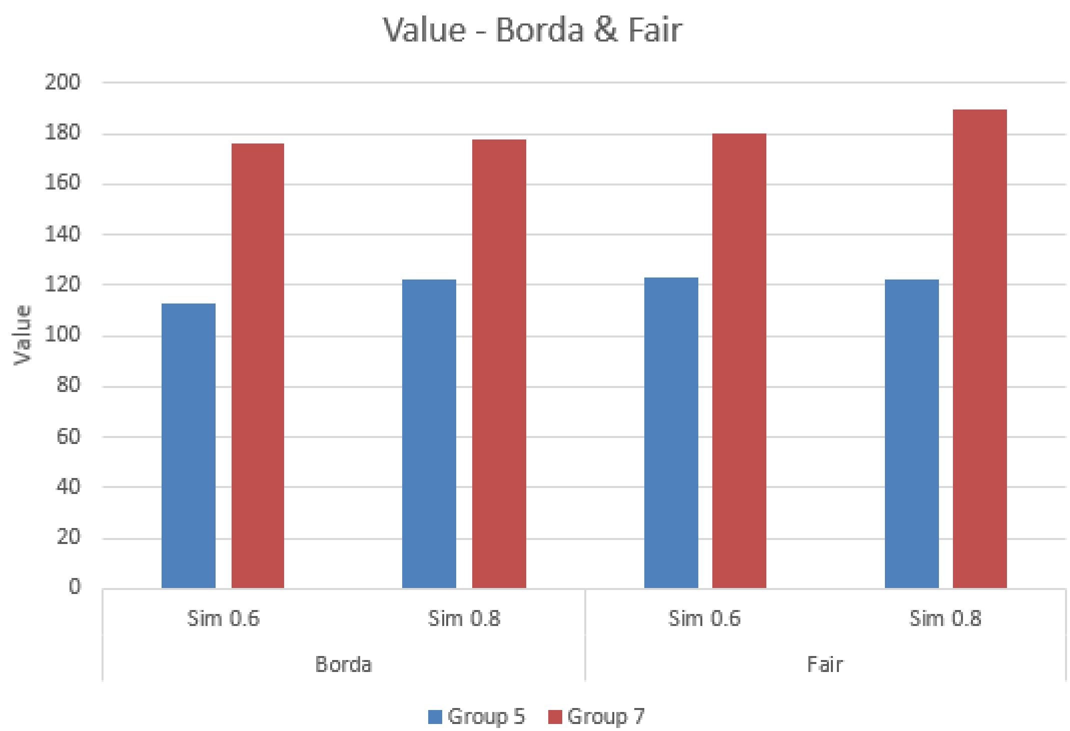
We can not compare designs across aggregation groups!

Value (2/3)



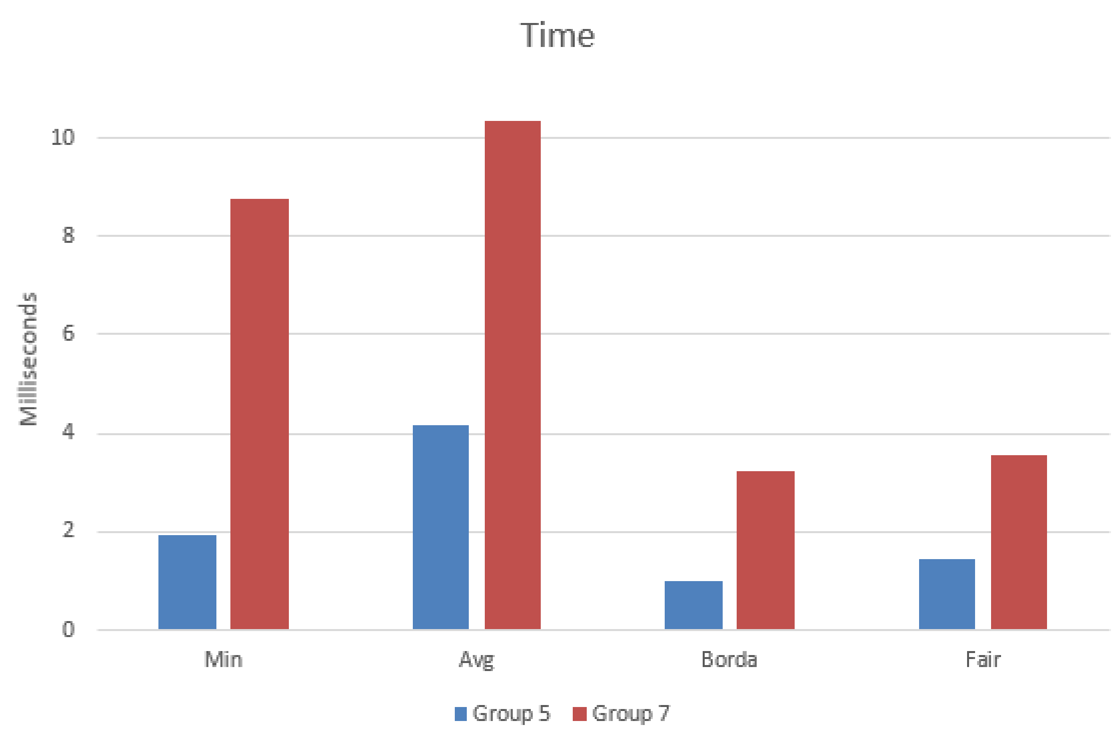
Test done for 10 different groups

Value (3/3)



Test done for 10 different groups

Time



Test done for 10 different groups

Advantages of the method

- Propose an end-to-end framework for group recommendations
  - First time in the health domain
- Incorporate fairness into recommendation process
- Introduce a novel similarity method
  - Outperform rating-based similarity measures
  - Exploit Semantic Information
  - Able to compute users similarities without ratings
- Explore various aggregation policies and introduce the fairness aware policy

Questions ?