FairGRecs: Fair Group Recommendations by Exploiting Personal Health Information

Maria Stratigi[1], Haridimos Kondylakis[2], and Kostas Stefanidis[1]

[1] University of Tampere, Finland, {maria.stratigi,kostas.stefanidis}@uta.fi
[2] ICS-FORTH, Greece, kondylak@ics.forth.gr

Abstract. FairGRecs aims to offer valuable information to users, in the form of suggestions, via their caregivers, and improve as such the opportunities that users have to inform themselves online about health problems and possible treatments. Specifically, FairGRecs introduces a model for group recommendations, incorporating the notion of fairness. For computing similarities between users, we define a novel measure that is based on the semantic distance between users' health problems. Our special focus is on providing valuable suggestions to a caregiver who is responsible for a group of users. We interpret valuable suggestions as ones that are both highly related and fair to the users of the group.

1 Introduction

During the last decade, the number of users who look for health and medical information has dramatically increased. However, it is still very hard for a patient to accurately judge the relevance of some information to his own case and to identify the quality of the provided information. The optimal solution for patients, however, is to be guided by healthcare providers to more optimal resources over the Web [1]. Delivering accurate sources to a patient, increases his/her knowledge and changes the way of thinking which is usually referred as patient empowerment. As a result, the patient's dependency for information from the doctor is reduced. Moreover, patients feel autonomous and more confident about the management of their disease [9]. To achieve this, health providers have the history of their patient's and their interests, in order to make an informed decision about the information that would likely be beneficial for the patients. However, health providers have less and less time to devote to their patients. As such, guiding each individual patient appropriately is a really difficult task. On the other hand, the use of group-dynamics-based principles of behavior change, have been shown to be highly effective in enhancing social support. In those cases, a caregiver guides patient groups to more optimal resources over the Web. However, if identifying online information content for a single patient is a difficult task, identifying information for a group of participants is a really challenging one.

To this direction, we focus on recommending interesting health documents selected by health professionals, to groups of users, incorporating the notion of fairness, using a collaborative filtering approach. Our motivation is to offer a list of recommendations to a caregiver who is responsible for a group of patients. The recommended documents need to be relevant, based on the patients current profiles. To exploit patients profiles,

we use the data stored in their personal health-care record (PHR) data. These patients do not necessarily suffer from the same health problems, but a variety of them. As such, we introduce the notion of fairness in the recommendation process.

More specifically, the contributions of our work are the following: a) We demonstrate the first group recommendation model incorporating fairness in the health domain; b) We propose a novel semantic similarity function that takes into account the patients medical profiles, showing its superiority over a traditional measure; c) We introduce a new aggregation method that encapsulates the notion of fairness; d) We explore 5 different aggregation methods; e) We present the first synthetic dataset and the corresponding engine for constructing it, for benchmarking works in the area. To our knowledge, this is the first work that introduces fair group recommendations in the health domain. A preliminary abridged version of this paper appears in [8].

2 Single User Recommendations

Assume a recommender system in the health domain, where I is a set of data items to be rated and U is the set of patients in the system. A patient, or user, $u \in U$ might rate an item $i \in I$ with a score $r(u, i)$, as in $[1, 5]$. Typically, the cardinality of the item set I is high and users rate only a few items. The subset of users that rated an item $i \in I$ is denoted by $U(i)$, while the subset of items rated by a user $u \in U$ is denoted by $I(u)$.

For the items unrated by the users, recommender systems estimate a relevance score, denoted as $relevance(u, i)$, $u \in U$, $i \in I$. To estimate the relevance score of an item, we follow the collaborative filtering approach. First, similar users are located via a *similarity function* that evaluates the proximity between two users. Then, items relevance scores are computed for individual users, taking into account their most similar users. Instead of only using classical similarity notions, we exploit the similarity in patient profiles (their diseases), improving the quality of the recommendations.

Similarity based on ratings. Two users are similar if they have rated data items in a similar way, i.e., they share the same interests. We calculate their similarity based on their ratings, by exploiting the Pearson correlation metric:

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

μ_u is the mean of the ratings in $I(u)$.

Similarity based on semantic information. In the health domain, usually two people have similar interest in health documents if they have similar health problems. ICD10[3] is a standard medical classification ontology, which we exploit to record and identify similarities between health problems and eventually between users. The ICD10 taxonomy can be represented as a tree, with health problems as its nodes. In the 2017 version of ICD10, there are 4 levels in the tree, in addition to the root level. Sibling nodes that belong to lower levels share greater similarity than siblings that belong to upper levels. Because of this, we assign different weights to nodes according to their level. These

[3] `http://www.icd10data.com/`

weights will help us differentiate between siblings nodes in the various levels; we want sibling nodes in the higher levels to share greater similarity than those in the lowest. Formally, for a node A in the ontology tree, $weight(A) = w * 2^{maxLevel - level(A)}$, where w is a constant, $maxLevel$ is the maximum level of the tree and $level(A)$ is a function that returns the level of each node. In addition, let $anc(A)$ be the direct ancestor of A, and LCA(A,B) be the lowest common ancestor of the nodes A and B. For computing the distance between A and B, we compute their distance from $LCA(A, B)$ = C. The distance between A and C is calculated by accumulating the weight of each node in the path, as $dist(A, C) = \sum_{n \in path(A,C)} weight(n)$. In overall, the similarity between A and B is: $simN(A, B) = 1 - \frac{dist(A,C)+dist(B,C)}{maxPath*2}$, where $maxPath = dist(root, L)$. L is a leaf node in the highest level.

Overall similarity between two users. Let $Problems(u)$ be the list of health problems of user $u \in U$. As such, given two users u and u', we calculate their overall similarity by taking into consideration all possible pairs of health problems between them. Specifically, we take one by one all the problems in $Problems(u)$ and calculate the similarity with all the problems in $Problems(u')$. For each distinct problem from u, we take into account only the health problem of u' that has the maximum similarity.

Definition 1 (SemS). *Let u and u' be two users in U. The similarity based on semantic information between u and u' is defined as: $SemS(u, u') = \frac{\sum_{i \epsilon Problems(u)} ps(i,u')}{|Problems(u)|}$, where $ps(i, u') = max(\forall_{j \epsilon Problems(u')}\{simN(i, j)\})$.*

Single User Rating Model. Let P_u denote the set of the most similar users to u, hereafter, referred to as the *peers* of u. If u has expressed no preference for an item i, the relevance of i for u is estimated as: $relevance(u, i) = \frac{\sum_{u' \in (P_u \cap U(i))} S(u,u')r(u',i)}{\sum_{u' \in (P_u \cap U(i))} S(u,u')}$, where S is either $RatS$ or $SemS$.

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

3 Group Recommendations

Our goal is to provide valuable suggestions to a caregiver who is responsible for a group of patients. We interpret valuable suggestions as suggestions that are both highly related and fair to the patients of the group.

Group Rating Model. Most previous works focus on recommending items to individual users. Recently, group recommendations that make recommendations to groups of users instead of single users (e.g., [5, 6]), have received considerable attention. Commonly, a method for computing group recommendations first estimates the relevance scores of the unrated items for each user in the group, and then, aggregates these predictions to compute the suggestions for the group. Formally, the relevance of an item for a group is computed as follows:

Definition 2 (Relevance). *Let U be a set of users and I be a set of items. Given a group of users G, $G \subseteq U$, the group relevance of an item $i \in I$ for G, such that, $\forall u \in G$, $\nexists rating(u, i)$, is: $relevanceG(G, i) = Aggr_{u \in G}(relevance(u, i))$.*

As in single user recommendations, the items with the top-k relevance scores for the group are recommended to the group.

Fairness in Group Recommendations. Given a particular set of recommendations for a caregiver, it is possible to have a user u that is the least satisfied user in the group for all items in the recommendations list, that is, all items are not related to u. Therefore, although the caregiver may like as a whole the set of recommendations, the package selection is not fair to u. In actual life, where the caregiver is concerned for the needs of all patients in his group, we should recommend items that are both strongly relevant and fair to the majority of the group members. In particular, to increase the quality of the recommendations for the caregiver, we consider, similar to [7], a fairness measure that evaluates the goodness of the recommendations as a set. This way, 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 $X = A_u \cap D$.

Intuitively, the fact that the group recommendations contain some highly relevant items to u, makes both u and his caregiver tolerant to the existence of other items that are not highly related to u, considering that there are other members in the group who may be related to these items. Then, the fairness of a set of recommendations D for a set of users G is defined as follows.

Definition 3 (Fairness). *Given a group G and a set of recommendations D, the fairness of D for G is defined as:* $fairness(G, D) = \frac{\sum_{u \in G} fairness(u,D)}{|G|}$.

Finally, we define the fairness-aware value of D for G as follows: $value(G, D) = fairness(G, D) \cdot \sum_{i \in D} relevanceG(G, i)$.

Aggregation Designs. We distinguish between the score-based and rank-based designs. In a *score-based design*, the prediction for an item is computed taking into account the relevance of the item for the group members. Firstly, we consider that strong user preferences act as a veto; this way, the predicted relevance of an item for the group is equal to the minimum relevance of the item scores of the members of the group: $relevanceG(G, i) = \min_{u \in G}(relevance(u, i))$. Alternatively, we focus on satisfying the majority of the group members and return the average relevance for each item: $relevanceG(G, i) = \sum_{u \in G} relevance(u, i)/|G|$.

In a *rank-based design*, we aggregate the group members recommendations lists by considering the ranks of their elements. Specifically, following the Borda count method [2], each data item gets 1 point for each last place received in the ranking, 2 points for each next to last place, and so on, all the way up to k points for each first place received in the ranking. The item with the largest point total gets the first position in the aggregated list, the item with the next most points takes the second position, and so forth, up to locate the best k items. Overall, the points of each item i for the group G is computed as follows: $points(G, i) = \sum_{u \in G}(k - (p_u(i) - 1))$, where $p_u(i)$ represents the position of item i in A_u.

Targeting at increasing the fairness of the resulting set of recommendations, we introduce also the $Fair$ method, which consists of two phases. In the first phase we consider pairs of users in the group, in order to identify what to suggest. In particular, a data item i belongs to the top-k suggestions for a group G, if, for a pair of users $u_1, u_2 \in G$, $i \in A_{u_1} \bigcap A_{u_2}$, and i is the item with the maximum rank in A_{u_2}. For

locating fair suggestions, initially, we consider an empty set D. Then, we incrementally construct D by selecting, for each pair of users u_x and u_y, the item in A_{u_x} with the maximum relevance score for u_y. If k is greater than the items we found using the above method, then we construct the rest of D, by serially iterating the A_u lists of the group members and adding the item with the maximum rank that does not exist in D.

Regardless of how similar the group members are, the first phase of the algorithm may yield few items (i.e., less than k). Moving to the second phase, we can assume a pseudo hierarchy inside the group members, meaning that the members that will be checked first, will have more relevant items for them, in the group list. So, by rearranging the order of the group members, we can influence the fairness achieved for each individual member. On the other hand, if we produce all top-k items from the first phase, then a number of items in the group list, may change accordingly to what order we examine the members. Again the vast majority of the items will be included, regardless of the members order. In both cases, the fairness of the list for the group does not change.

4 Experimental Evaluation

Dataset. In our experiments, we exploit 10.000 chimeric patient profiles [3] preserving the characteristics that exist in a real medical database. The patients health problems are described using the ICD10 ontology. Based on these profiles, we synthetically generated a document corpus and user ratings as follows. Initially, we generated $numDocs$ documents, for each first level category of ICD10. For their corresponding keywords, we randomly selected $numKeyWords$ words from the description of the nodes in each subsequent subtree. We assume that all patients have given $numRatings$ ratings. Specifically, we have divided the patients into three groups – $sparse$, $regular$ and $dedicated$. The users in each group have given *few*, *average* and *a lot* of ratings, respectively. When ranking items based on human preferences, they tend to follow the power law distribution. To depict this, we have randomly selected $popularDocs$ documents that will be the most popular. Given that patients are interested not only in documents regarding their health, but also to some extent in others as well, for each patient, we have divided the ratings into $healthRelevant$ and $nonRelevant$. Finally, for each rating generated in the previous step, we assigned randomly, *a value* in the range of 1 to 5. In our experiments, we set *numDocs*=270, *numKeyWords*=10 and *popularDocs*=70.

Aggregation Methods. In our evaluation, we use the Minimum, Average, Borda and Fair designs. As a baseline, we will employ the Round-Robin aggregation design, considering each member of the group individually, and for each one, taking the item in his/her list with the highest score that does not already exist in the group list.

Evaluation Measures. For our experiments, to calculate the semantic similarity *SemS*, we use $w = 0.1$. To evaluate the similarity functions, we used the Mean Absolute Error (MAE) and the Root Mean Square error (RMSE). To quantify the success of each aggregation method, we compute the distance of each user's top-k recommendation list with that of the group. To calculate the final score, we take the average of those. For calculating the distance, we used the Kendall tau and the Spearman footrule distance.

Evaluation of Similarity Functions. To compare the two similarity functions, we focused on single users recommendations. In Figure 1, we see the different values of MAE and RMSE for several k values. In all cases, the semantic similarity function gave better results than the rating function. This shows the added value of our solution on calculating effectively the similarity between users.

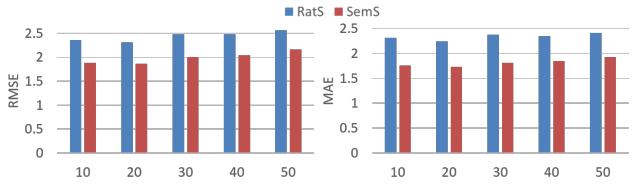

Fig. 1: The RMSE and MAE for different K

Evaluation of Aggregation Methods with Different Similarity Functions. To compute the distance between the top-k lists of the group members, and the group recommendation lists, we use the Kendall tau and Spearman footrule distances. We randomly selected 10 different groups that share the same *group similarity*. Group similarity is the similarity of all pairs of users in the group, averaged over the number of pairs. After generating group recommendations, we calculate for each member of the group the Kendall and Spearman distance. The distance score for the aggregation method is the averaged score of the summation of these distances over the number of group members. Following the same procedure for all 10 groups, the overall score for each aggregation method is the mean of the previously calculated scores, over the number of different groups. To further supplement our findings, we compare the Kendall and Spearman distance for 10 different groups of size 5. The results are shown in Figure 2 (a) and (b); $SemS$ offers better results than $RatS$, regardless of the aggregation design used.

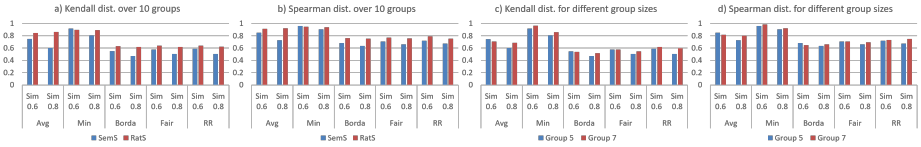

Fig. 2: The Kendall and the Spearman distance.

Evaluation of Aggregation Methods with Different Group Size. To get more accurate results, we study different sized groups, namely groups with 5 and 7 members. Using the Kendall distance (Figure 2(c)), the rank-based methods give better results than the score-based methods. This is because the score-based methods consider the whole user's list, while the rank-based ones consider only the top-k items. The Minimum and Average designs, take into account the scores given to an item. For example,

given one item, if a member of the group has a radically different relevance score for it than the rest, for Minimum, his opinion will act as veto, while in the case of Average, its group relevance will be brought down and might not make it into the group list. The rank-based methods are able to include more items from the members individual top-k recommendation lists, and hence give lower distances. Round-Robin is the worst method, while Borda and Fair have similar results. As expected, when the group similarity gets higher, all methods provide better results. Finally, the size of the group, given that we consider groups with the same similarity, slightly affects the quality of the results of the employed aggregation methods, and overall, the bigger the size of the group, the higher the Kendal tau distance. Figure 2(d) shows similar results for Spearman.

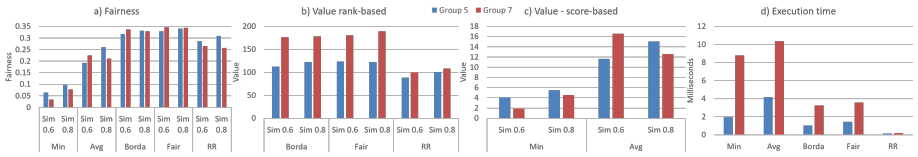

Fig. 3: Fairness, value and execution times.

Fairness. Since with fairness, we measure in essence how many of the items in an individual top-k list made it in the group recommendation list, which is inherently what we shown in Figure 2 (c) and (d), we expect complementary results. In fact, the Minimum method that had the highest distance produces the lowest fairness. Although Borda gave marginally lower distance, Fair now gives better fairness. An explanation for this is that during the first phase of the Fair algorithm, we take into account items regardless of their relevance score – we actually want to include in the group list items that are relevant to most users, leading to higher fairness. Finally, the Round-Robin method gives lower fairness than the rest of the rank-based methods, because with Round-Robin, we do not consider the rest of the group members when constructing the recommendations list, but we consider each member individually from the others.

Value. Because of the inherent differences of the score-based and the rank-based designs, we cannot directly compare them regarding their value. To elaborate more, when we aggregate using the score-based techniques, for any given item, we directly compute its corresponding relevance score in the final group recommendation list. On the other hand, with the rank-based techniques, what we actually calculate is the rank that a specific item will have in the group recommendation list. But to find the value of a group recommendation list, we need the relevance score of all its items. Thus, we define the relevance score of an item in a group recommendation list, that is produced by a rank-based technique as the summation of its rank in each individual recommendation list of the group members. If an item is not present in a list, then it's score is 0. As it is apparent, we cannot directly compare score-based and rank-based aggregation methods. The score-based ones give a relevance score to an item in the range of [1,5], while the rank-based approaches, given that the group recommendation list consist of k items and the size of the group is s, give a relevance score in the range $[1, s*k]$. In Figure 3(b), we see

the *value* for the score-based aggregation methods. As expected, the Average method offers much better results. In Figure 3(c), we compare the rank-based methods. These results complement those in Figure 3(a). The Fair algorithm offers better overall value for the group recommendation list. The worst results are presented by Round-Robin; the fairness that Round-Robin offers is highly incidental and the relevance scores for those are low, since most users do not share the same items with the same high scores.
Execution time. In Figure 3(d), we show the time needed to aggregate the individual lists for each method. For each group size, we took randomly 10 different groups with the same group similarity. Average is the most time costly method. The time needed for Fair is marginally more, than the one needed for Borda. Nevertheless, the execution time is really small (at most 4 msecs), offering better results (Figure 3(a)).

5 Conclusions

In this work[4], we investigate how fairness can be modeled in group recommendations in the health domain. Specifically, we proposed a new similarity function that takes into account information provided by a patient's profile. As in modern Personal Health Systems [4], patient information is represented using standard terminologies, in this work, as a proof of concept, we employ the ICD10 ontology. Our experiments confirm that our proposed similarity function gives better results than traditional similarity functions based on ratings. We proceed even further, to explore and compare 4 different aggregation methods. Our experiments demonstrate the good behavior of our solution with respect to its target, i.e., to increase the fairness and utility of the suggested results.

References

1. G. M. Berg, A. M. Hervey, and D. A. et al. Evaluating the quality of online information about concussions. *JAAPA*, 27:1547–1896, 2014.
2. P. Emerson. The original borda count and partial voting. *Social Choice and Welfare*, 40(2):353–358, 2013.
3. U. Kartoun. A methodology to generate virtual patient repositories. *CoRR*, abs/1608.00570, 2016.
4. H. Kondylakis, E. G. Spanakis, S. Sfakianakis, V. Sakkalis, M. Tsiknakis, K. Marias, X. Zhao, H. Yu, and F. Dong. Digital patient: Personalized and translational data management through the myhealthavatar EU project. In *IEEE EMBC*, pages 1397–1400, 2015.
5. E. Ntoutsi, K. Stefanidis, K. Nørvåg, and H. Kriegel. Fast group recommendations by applying user clustering. In *ER*, 2012.
6. E. Ntoutsi, K. Stefanidis, K. Rausch, and H. Kriegel. Strength lies in differences: Diversifying friends for recommendations through subspace clustering. In *CIKM*, 2014.
7. S. Qi, N. Mamoulis, E. Pitoura, and P. Tsaparas. Recommending packages to groups. In *ICDM*, 2016.
8. M. Stratigi, H. Kondylakis, and K. Stefanidis. Fairness in group recommendations in the health domain. In *ICDE*, 2017.
9. M. Wiesner and D. Pfeifer. Adapting recommender systems to the requirements of personal health record systems. In *IHI*, 2010.

[4] The work was partially supported by the EU project iManageCancer (H2020, #643529), and the TEKES Finnish project Virpa D project.