


Provided by the author(s) and University College Dublin Library in accordance with publisher policies. Please cite the published version when available.

Title	Using social ties in group recommendation
Author(s)	Bourke, Steven; McCarthy, Kevin; Smyth, Barry
Publication date	2011-08-31
Publication information	AICS 2011 : Proceedings of the 22nd Irish Conference on Artificial Intelligence and Cognitive Science : 31 August - 2 September, 2011 : University of Ulster - Magee
Conference details	Paper presented at the 22nd Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2011), University of Ulster, Northern Ireland, 31 August - 2 September, 2011
Publisher	Intelligent Systems Research Centre
Link to online version	<a href="http://isrc.ulster.ac.uk/aics2011/images/files/AICS_2011_Proceedings.pdf">http://isrc.ulster.ac.uk/aics2011/images/files/AICS_2011_Proceedings.pdf</a>
Item record/more information	<a href="http://hdl.handle.net/10197/3448">http://hdl.handle.net/10197/3448</a>

Downloaded 2018-02-22T16:42:55Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd\_oa) 

Some rights reserved. For more information, please see the item record link above.



# Using Social Ties In Group Recommendation

Steven Bourke, Kevin McCarthy, Barry Smyth

CLARITY: Centre for Sensor Web Technologies,  
School of Computer Science and Informatics,  
University College Dublin (UCD), Belfield, Dublin 4, Ireland.  
`firstname.lastname@ucd.ie`

**Abstract.** The social web is a mass of activity, petabytes of data are generated yearly. The social web has proven to be a great resource for new recommender system techniques and ideas. However it would appear that typically these techniques are not so social, as they only generate recommendations for a user acting alone. In this paper we take the social graph data and preference content (via Facebook) of 94 user study participants and generate social group recommendations for them and their friends. We evaluate how different aggregation policies perform in deciding the final group recommendation. Our findings show that in an offline evaluation an aggregation policy which takes into consideration social weighting outperforms other aggregation policies.

**Keywords:** Recommender systems, Collaborative Filtering, Group Recommendation, Aggregation, Social network

## 1 Introduction

The social web has changed the way people view the Internet. Initially, the Internet was primarily used for research and the sharing of ideas, with very few people contributing to the overall content of the Internet. However, in recent times we've seen the social web grow and services such as Facebook and Twitter allow for the mass population to fill the Internet with social graph details, public and private messages, and most importantly, from our point of view, detailed user profiles. In this paper, we present an investigation into the effects of different aggregation policies in a group recommender system when used at the consensus negotiation stage, we do this using a collaborative filtering recommendation technique. We apply these aggregation policies to information gathered from a user study that leveraged the Facebook social graph. The goal of our work is to investigate if the social graph is actually beneficial when trying to recommend items, in this case TV and movie content to groups of people. Typically on the social web, collaborative filtering recommendation systems generate recommendations to single users, this can make the social web appear to be somewhat unsocial. Our proposal is a group recommendation consensus negotiation technique, which uses aggregation policies alongside social metrics. We evaluate our

---

Supported by Science Foundation Ireland under Grant No. 07/CE/11147 CLARITY

aggregation policies across three types of groups, the first group type is based on a test subject selecting friends who they deem to be experts in their field (In this case TV and Movie content), the second group type is based on who communicates the most with our test subject via Facebook. The third group type is based on who has the most similar Facebook profile to the user study participant's Facebook profile. The rest of the paper is structured as follows, Section 2 covers related work, Section 3 covers our approach, Section 4 is our evaluation, Section 5 our results and finally Section 6 cover our conclusions.

## 2 Related Work

In this section, we shall review the relevant background material from the group recommender system field. Group recommender systems can be classified as either active or passive group recommender systems [2]. An active group recommender system involves the group members interacting with the system to shape the final result, for instance users of the CATS system [6] can critique preferences during the recommendation process. The critiquing interactions from the group members will affect the final recommendation returned to the group. On the other hand a passive group recommender system uses the previously known user models for each group member to generate a group recommendation. An example of a passive group recommender system is the PolyLens system[8], PolyLens is built on top of MovieLens[3], which is a well known movie recommender system. When generating recommendations for groups, PolyLens will then use each group members MovieLens profile to determine a movie recommendation for the entire group. This is achieved by using an aggregation policy called least misery. An aggregation policy is a way for a recommender system to algorithmically predict the level of interest in an item[9], this could be for a single user or even a group of users. Aggregation policies are widely used in both single user and group based recommendation systems. Aggregation policies originate from the concept of social choice, which was first introduced by Arrow[1], who is a well known economist. Social choice is a process which allows groups to decide on a final outcome, its theoretical grounding lays firmly in the field of voting theory. Masthoff[5] performed a study where by a number of these social choice theories are used as aggregation policies in a group recommender system which recommends TV content. The study shows that different types aggregation policies can have substantially different outcomes for the exact same group. This is because some aggregation policies may place an emphasis on individual user satisfaction, where as other aggregation policies could place their emphasis on the group satisfaction. Whether or not a group recommender system is passive or active they will still need to overcome four of the key challenges that face group recommenders which are, preference elicitation, recommendation generation, presentation and explanation [4, 10]. Our approach would be considered a passive group recommender as a user would not interact directly with the group recommendation stage. Also our work focuses on the consensus negotiation stage.

### 3 Approach

In this section, we describe the user study we performed which is the basis of the dataset we collected for the group recommendation work carried out in this paper. After which we will then describe our group recommendation approach. Our group recommendation approach uses single user recommendation for individual group members, we then merge these results to represent the group model. At this point we use an aggregation policy to identify the final recommendation for the group.

#### 3.1 User Study

We performed a live user study during May 2011, the user study was completed via a Facebook application by the study participants. When a participant signed into the application we extracted their Facebook profile as well as their entire social graph. This typically involved a participants friends and all the movie and TV content associated with each profile. The TV and Movie content extracted from the user profiles does not have any score to indicate how strong the user's preference for the content actually is. This means that we must represent the information in a unary fashion meaning it either exists in a user profile or it does not. Additionally we captured metrics such as who the participant communicated with the most frequently via Facebook. This was done by extracting and processing a user's Facebook feed. We also asked the test subjects to identify at least ten friends they felt were experts or had good taste in movie and TV content. In total 94 people participated in the user study, the majority of the participants were from Ireland(51), USA(14) and Philippines(9) with the rest coming from a number of countries around the world. On average each participant had 232 friends and 31 interests in their profile. All test subject participants allowed us access to their social graph data for our experiments. All user and item (TV and Movie) information are represented by numbers. We do not store any information about a users Facebook feed other than a count of the interactions which took place.

#### 3.2 Individual Recommender System

In this section we shall discuss the recommender system which we used to generate recommendations for an individual group member. It is important to note that the user preferences gathered from our user study are represented in a unary scale, meaning that an item either exists in a users preference model or it does not. Typically user preferences for items used in collaborative filtering recommendation systems operate on a scale of 1 to 5. Therefore when we want to predict a users interest in an item we must use an adapted version of user based collaborative filtering, we use an approach proposed by Mild[7]. It is worth noting at this point, that while Mild's work talks about binary data, it is in actual fact dealing with unary data. The predictive algorithm for the technique proposed by Mild et al. is the core difference between recommendation systems

that operate across a non unary rating scale, the predictive algorithm we use can be seen in Equation 1. To use the predictive algorithm we must firstly identify a user’s nearest neighbours using a user to user similarity metric from standard collaborative filtering. We use Tanimoto’s Coefficient to calculate user to user similarity, the  $i$  most similar user’s will form a cluster of similar users known as a users nearest neighbours. We use Tanimoto’s instead of the standard Pearson’s correlation Coefficient because Tanimoto does not require any numerical value associated with the user preferences to calculate similarity. The predictive value (from Mild) is calculated with the function  $P_{a,j}$  (See Equation 1) where we calculate our target user  $a$ ’s interest in the item  $j$ , to do this we count the number of occurrences of item  $j$  in the target user’s nearest neighbourhood, using the weighting function  $w(a,i)$  (Tanimoto’s coefficient Equation 2) to generate the predictive value of an item. We set  $k$  as the number of occurrences an item appears in the users neighbourhood so we can normalise the predictive value across a 0 to 1 scale.  $C_{ij}$  relates to co-occurring items between users  $a$  and  $i$ .

$$p_{a,j} = k \sum_{i=1}^N w(a,i)c_{i,j} \quad (1)$$

$$w(a,i) = \frac{n(c_a \cap c_i)}{n(c_a \cup c_i)} = \frac{n(c_a \cap c_i)}{n(c_a) + n(c_i) - n(c_a \cap c_i)} \quad (2)$$

Using this technique we can generate a *top-n* list of recommendations for each user which we will use in the group recommendation stage.

### 3.3 Group Recommendation Approach

In this section we will describe how we generate recommendations for a group. The entire process can be broken down into four steps. At the beginning (**Step one**) of the group recommendation we have a group of  $N$  members. The next step (**Step two**) is to generate a *top-n* list of recommendations for each individual group member. We then take each individual group member’s list of recommendations and merge them into a group list (**Step three**), we use our recommendation algorithm to generate a score for each item in the group list, for each user. At this point each user should have a score associated with each item in the group list. At **Step four** we apply a aggregation strategy to the group list so that we can generate a final recommended item for the group. We will explain the different aggregation policies we use and how they work in the next section. We will use the example group model listed in Table 1 for the source of our aggregation strategy explanations.

### 3.4 Aggregation Policies

We use aggregation policies to decide what item should be recommended to the group. This is done by using the aggregation policy on the list of potential group recommendations which we discussed in the previous section. The point of our

aggregation policy is to identify the most relevant item to recommend. Whichever item best meets the constraints of the aggregation policy will be recommended to the group. The aggregation policy is useful as it is likely each individual user in the group will have different preferences for different items. In the case of the work carried out here, we use six well known aggregation policies which are come from social choice theory [1, 9]. A problem group recommenders need to overcome is how to adapt to the preferences of the group as a whole based on information about individual user’s likes, this issue is considered to be one of the fundamental challenges for group recommenders. For instance, suppose the group contains three people: Bob, Peter and Jane (As in Table 1). We will explain how our chosen aggregation policies perform in a given test scenario. If a group

	A	B	C	D	E	F	G	H	I	J
Bob	10	4	3	6	10	9	6	8	10	8
Peter	1	9	8	9	7	9	6	9	3	8
Jane	10	5	2	7	9	8	5	6	3	6

**Table 1.** Group estimation and aggregation table

recommender system is aware that these three individuals are present and has a list of their top ten recommendations, how would the aggregation policy decide what item should be recommended to the group. Table 1 gives example predicted ratings on a scale of 1 (Hate) to 10 (Love). As mentioned we use six aggregation policies from the work performed by Masthoff[5]. These aggregation policies are as follows; respect, borda count, least misery, most pleasure, multiplicative and approval, the examples given for how each aggregation policy works in relation to Table 1 is below. The first item to reach a 'winning' position will be selected, currently we do not have a mechanism in place to handle ties.

- Respect: Usually this aggregation policy uses a weighting assigned to each group member when calculating a group score. If Jane (See Table 1) is given the highest weighting, then A’s group rating is 1. If Bob has the highest weighting, then it is 10. This policy is where we use our social weighting to determine the group recommendation. When using the respect policy the aggregation weighting for each user will be based on the group formation strategy. This is explained more in Section 4.1.
- Borda Count: Counts points from items rankings in the individuals, preference lists, with bottom items getting 0 rank points, the first from bottom getting one rank point, etc. For example A’s group rating is 17, namely 0 (last for Jane) + 9 (first for Bob) + 8 (shared top 3 for Peter).
- Least Misery: Takes the minimum of individual ratings. B’s group rating is 4, namely the smallest of 4, 9, 5.
- Most Pleasure: Takes the maximum of individual ratings. B’s group rating is 9, namely the largest of 4, 9, 5.

- Multiplicative: Multiplies individual ratings. B’s group rating is 180, namely  $4 \times 9 \times 5$ .
- Approval Voting: Counts the individuals with ratings for the item above a approval threshold (e.g. 6). B’s group rating is 1 and F’s is 3. For our experiments we set the threshold to 0.7 from a scale of 0 to 1.

In this section we have reviewed how we generate group recommendations using single group members, the group recommender approach and finally a number aggregation policies. We use the single user recommender system to identify items which are likely to of interest to individual group members, we perform our group recommendation approach as described in Section 3.3 to identify items of interest to the group as a whole through different aggregation policies as described in Section 3.4. We use the aggregation policy to select which item should be selected as the group recommendation.

## 4 Evaluation

In this section we will explain the evaluation set up and evaluation metrics that we use to measure the performance of the aggregation policies. We performed an offline evaluation, using the information collected from our initial live user study as the underlying datasource.

### 4.1 Setup

As previously mentioned our dataset is based on information captured during a live user study from which 94 individuals took part. To use the information from the user study in the group recommendation process we generate groups using three different strategies for each study participant which are as follows,

- G1** - This group is formed on the basis of the users who were selected as experts by the user who partook in our user study.
- G2** - This group is formed on the basis of the users who communicate the most often with the user who partook in our user study.
- G3** - This group is formed on the basis of the users profile being the most similar to our target user who partook in the experiment. We use Tanimoto’s coefficient (Equation 2) to determine user similarity.

In all cases each group has a seed user who is responsible for the structure of the group, namely the user who partook in the user study. For each of these 94 users we generate groups with a size of 3, 5 and 10 using G1, G2 and G3 as group formation strategies. This means that in total we generate 846 groups which are based on 94 seed users. If we think back to the aggregation policies described in Section 3.4 we mentioned an aggregation policy called respect, which had a weighting applied to it. Well in the context of our user study, the weighting is applied to the group formations in the following ways.

- G1** The weightings are applied based on which user was selected first. Therefore the user who was selected first by our study participant will get the second highest weighting (The first highest going to the study participants). The weighting score will decrease as we go through the group members.
- G2** The friend who communicates with our target user the most has the second highest weight, and the user who communicates with our user the least has the lowest weighting. As in G1 the study participant has the highest weighting.
- G3** The user with the highest similarity score gets the second highest weighting, while the user with the lowest similarity gets the lowest weighting. As in G1 and G2 the study participant has the highest weighting.

## 4.2 Evaluation Metrics

To evaluate our aggregation policies we run the group recommendation process as described in Section 3.3 using each of the aggregation policy described in Section 3.4. The item with the best predicted value within the context of the applied aggregation policy is deemed to be the item that should be recommended to the group. To determine the overall group satisfaction with the final recommendation we use Equation 3 where  $g$  is the target group,  $j$  is the target item, we then calculate the average predicted score for the group, using  $p_{m,j}$ , which is the prediction algorithm described previously. Individual group members are represented as  $m$  and  $j$  is the item.

$$GroupScore(g, j) = AVG\left(\sum_{m=1}^g p_{m,j}\right) \quad (3)$$

To calculate our benchmark metric, we took each group member’s list of nearest neighbours and calculate how interested each neighbour is in the item that is being recommended to the group. We then average this number and use that as our benchmark metric. This is shown in Equation 4. In Equation 4  $g$  is the group we are calculating the bench mark metric for. We use  $j$  as the item that was selected by the aggregation policy for the group. The next step is to predict a value for the item using the prediction function  $p_{i,j}$  (See Equation 1) for each user  $i$ . The user  $i$  is taken from the group members  $m$  nearest neighbourhood  $n$

$$BenchMark(g, j) = \sum_{m=1}^g (AVG\left(\sum_{i=1}^n (p_{i,j})\right)) \quad (4)$$

In this section we have described the different group formation approaches as well as the two metrics we use to measure the performance of our group recommender. We have shown the three different group formation strategies we evaluate our aggregation policies against, as well as the different sized groups. In total we evaluate the aggregation policies against 846 different groups.



## 5 Results

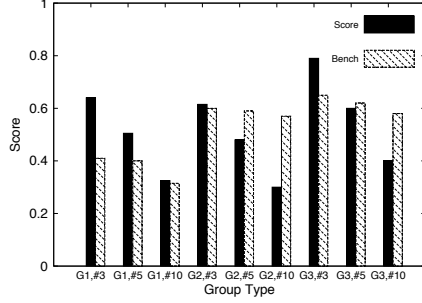
In this section we shall review the results from our offline evaluation for our aggregation policies. Our results can be found in Figure 5. Our results show that *Respect* and *Multiplicative* generally perform the best overall. We can see that not only do both of these aggregation policies return a high satisfaction score, but also that they both nearly always surpassed the benchmark scores. The only time *Respect* clearly outperforms *Multiplicative* is for group type *G1*. In all other group formation approaches both aggregation strategies perform to a similar level. This is interesting because our *Respect* aggregation policy is the only aggregation policy considering any social metrics when calculating the final group recommendation. As mentioned *Respect* is providing additional weighting to different users in the group consensus stage, which is in effect making some users more important than others. A common theme mentioned in the social choice literature [1] is that it is incredibly difficult to find a group outcome that makes everyone happy, and that one potential way to avoid such a situation is to use a dictator. Obviously the use of a dictator would be counter intuitive in a group recommendation system. We believe this to be the case because if a user found their preferences were never considered in a group recommendation they would be less likely to trust it. Therefore our *Respect* approach which allows for more influential users to exist would appear to be a good alternative to having a dictator.

The *Least Misery* aggregation strategy is a good example of a policy that tries to satisfy the group as opposed to individuals. The *Most Pleasure* aggregation strategy is a good example of an aggregation policy that tries to satisfy individual group members. As mentioned in Masthoff[5] these conflicting objectives can impact the quality of aggregation policies. What we see from our results is that *Least Misery* is unable to find an item as highly received as *Most Pleasure* but the performance of *Least Misery* across the larger group types does not fall off as dramatically as *Most Pleasure*. We can definitely see a substantial difference between *Least Misery* and *Most Pleasure* when comparing the aggregations satisfaction score to the bench mark score. *Most pleasure* performs the poorest of all the aggregation strategies when considering pure satisfaction against the benchmark score.

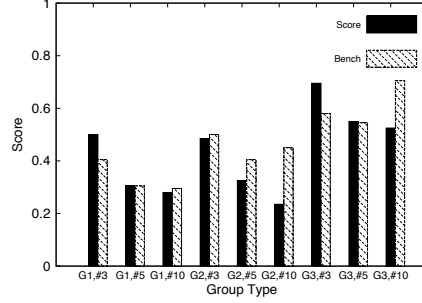
When looking at *Borda* and *Approval* we can see that both perform significantly better for groups *G3*, which as mentioned previously are based on users with the most similar profiles. Unsurprisingly we can see that the performance of groups formed based on user to user similarity perform quite well across the board. Both approaches perform well when comparing the satisfaction score to the bench mark score, but generally speaking the scores themselves are lower than other policies. Both *Borda* and *Approval* perform poorly for group types *G1* and *G2*.

Overall the only aggregation policy that performs particularly poorly is *Most Pleasure*. We can see that *G1* is the most difficult group to find a good recommendation for, but usually the item that is selected provide an adequate satisfaction

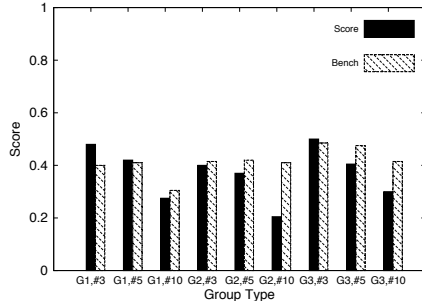
score when compared to our benchmark scores.  $G3$  on the other hand appears to be easier to find a good recommendation for.



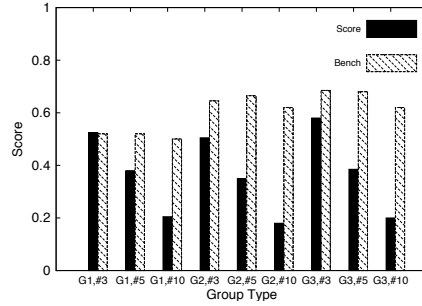
(a) Aggregation Policy: Respect



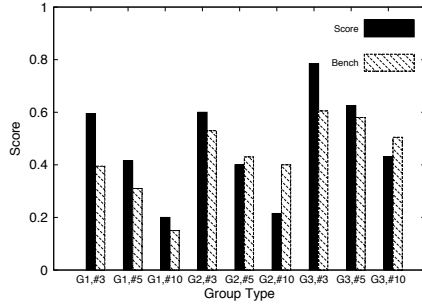
(b) Aggregation Policy: Approval



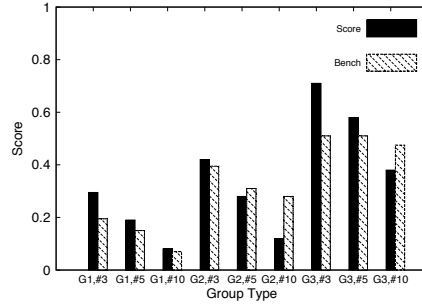
(c) Aggregation Policy: Least Misery



(d) Aggregation Policy: Most Pleasure



(e) Aggregation Policy: Multiplicative



(f) Aggregation Policy: Borda

## 6 Conclusion

In this paper we have presented our work which covers different aggregation policies applied to the consensus negotiation stage of group recommenders which use social graph information. Our work demonstrated that an aggregation policy that

can use social metrics is beneficial when trying to find an item which can be recommended to a group. However other aggregation policies also performed well in our offline evaluation. Our approaches show that having additional information, namely social metrics, can help in a group decision process by placing different weighting scores on users with strong social connections. In our future work we will take a more in depth look into forming different group models which can better take advantage of the social data on offer. We will also look to explore different group recommendation techniques.

## References

1. Arrow, K.J.: *Social Choice and Individual Values*. Yale University Press, 2 edn. (1970)
2. Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Rueda-Morales, M.A.: Managing uncertainty in group recommending processes. *User Modeling and User-Adapted Interaction* 19, 207–242 (2009)
3. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 230–237. SIGIR '99, ACM (1999)
4. Jameson, A.: More than the sum of its members: challenges for group recommender systems. In: *Proceedings of the working conference on Advanced visual interfaces*. pp. 48–54. AVI '04, ACM (2004)
5. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction* 14, 37–85 (2004)
6. McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P.: Group recommender systems: a critiquing based approach. In: *Proceedings of the 11th international conference on Intelligent user interfaces*. pp. 267–269. IUI '06, ACM (2006)
7. Mild, A.: An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data. *Journal of Retailing and Consumer Services* 10(3), 123–133 (2003)
8. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: *Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*. pp. 199–218. Kluwer Academic Publishers (2001)
9. Pennock, D.M., Horvitz, E., Giles, C.L.: Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering. In: *Proceedings of the National Conference on Artificial Intelligence AAAI*. pp. 729–734 (2000)
10. Ricci, F., Rokach, L., Shapira, B., Kantor, P.E. (eds.): *Recommender Systems Handbook - Chapter 21: Group recommender systems: Combining individual models*. Springer (2011)