Exploiting Group Recommendation Functions for Flexible Preferences

Senjuti Basu Roy[1], Saravanan Thirumuruganathan[2], Sihem Amer-Yahia[3], Gautam Das[2], Cong Yu[4]

[1]UW-Tacoma, [2]UT-Arlington, [3]CNRS-LIG, [4]Google Research

Motivation

- Group Recommendation is Ubiquitous

business Lunch

playlist of songs for a party

- Exploiting group recommendation functions for maximizing individual satisfaction

Problem Definition

Propose principled solutions to bias the output of boolean group recommendation functions

- in favor of an individual user
- without altering group recommendation functions
- for two most popular group consensus functions
- and various user satisfaction functions

Novel Contributions

- Boolean Preference Model

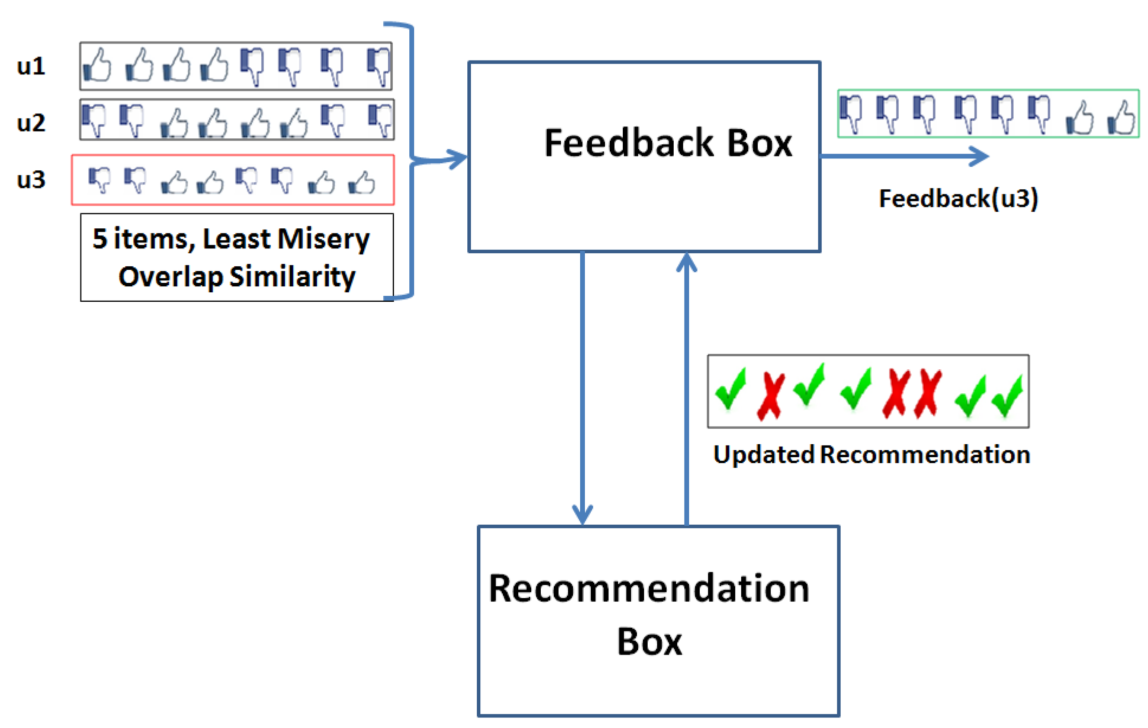
	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
User Preference	0	0	1	1	0	0	1	1

- Group recommendation algorithms for Boolean preferences
- Algorithms to exploit group recommendation algorithms
- Extensions to numerical preferences
- Robustness of group recommendations

Exploiting Group Recommendation Algorithms using Feedback box

- Exploits group recommendation functions to increase individual user satisfaction
- Almost always improves user satisfaction without ever reducing it
- Does not modify semantics of existing group recommendation functions
- Inspired by ideas from Strategic Voting
- Highly efficient (polynomial) as a function of running time of group recommendation algorithms

Interactions in Feedback Box



Group Recommendation and Feedback Algorithms

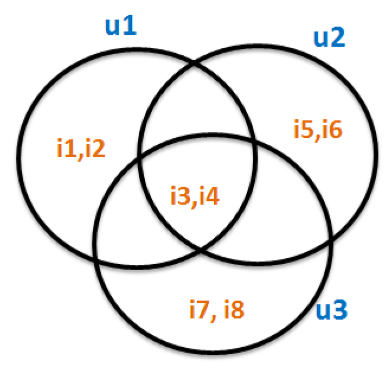
Setting	Recommendation Box	Feedback Box
Aggregated Voting		
Overlap Similarity	R-AGS	FB-AGS
Hamming Distance	R-AGD	FB-AGD
Least Misery		
Overlap Similarity	R-LMS	FB-LMS
Hamming Distance	R-LMD	FB-LMD

Illustrative Example

- 3 users $\{u_1, u_2, u_3\}$ and 8 items $\{i_1, i_2, \ldots, i_8\}$
- Atmost $k = 5$ to be recommended
- User Preferences

User Preferences	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
pref(u_1)	1	1	1	1	0	0	0	0
pref(u_2)	0	0	1	1	1	1	0	0
pref(u_3)	0	0	1	1	0	0	1	1

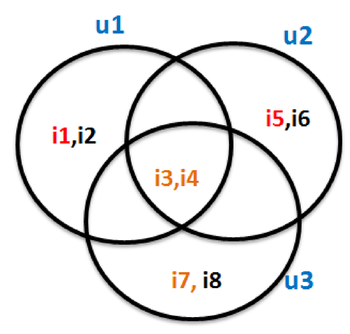
- User Preferences as a Venn Diagram



Recommendation Box in Action

1. Least Misery and Overlap Similarity

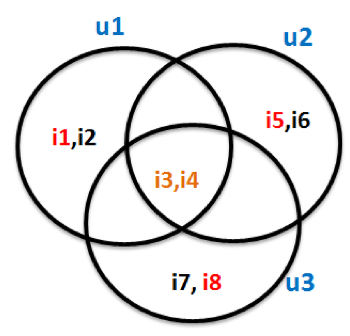
Preferences	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
pref(u_3)	0	0	1	1	0	0	1	1
I_k	1	0	1	1	1	0	1	0



- Satisfaction(u_1, I_k) = 3, Satisfaction(u_2, I_k) = 3, Satisfaction(u_3, I_k) = 3
- I_k maximizes the minimal Overlap Similarity of 3 (max {3,3,3})

2. Least Misery and Hamming Distance

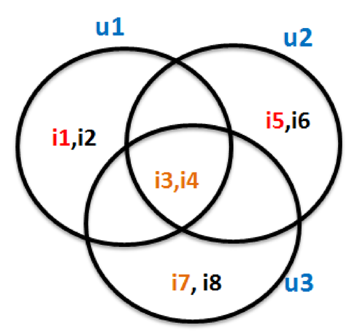
Preferences	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
pref(u_3)	0	0	1	1	0	0	1	1
I_k	1	0	1	1	1	0	1	0



- Distance(u_1, I_k) = 3, Distance(u_2, I_k) = 3, Distance(u_3, I_k) = 3
- I_k minimizes the maximum Hamming Distance of 3 (min {3,3,3})

3. Aggregated Voting and Overlap Similarity

Preferences	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
pref(u_3)	0	0	1	1	0	0	1	1
I_k	1	0	1	1	1	0	1	0



- Satisfaction(u_1, I_k) = 3, Satisfaction(u_2, I_k) = 3, Satisfaction(u_3, I_k) = 3
- I_k provides the maximum Aggregated Overlap Similarity of 9 (3 + 3 + 3)

Feedback Box in Action

1. Least Misery and Overlap Similarity

Preferences	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
pref(u_3)	0	0	1	1	0	0	1	1
I_k	1	0	1	1	1	0	1	0
feedback(u_3)	0	0	0	0	0	0	1	1
New I_k	1	0	1	1	0	0	1	1

- Satisfaction of u_3 with pref(u_3) and I_k = 3
- Satisfaction of u_3 with feedback(u_3) and new I_k = 4

2. Least Misery and Hamming Distance

Preferences	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
pref(u_3)	0	0	1	1	0	0	1	1
I_k	1	0	1	1	1	0	1	0
feedback(u_3)	0	0	0	0	0	0	1	1
New I_k	1	0	1	1	0	0	1	1

- Satisfaction of u_3 with pref(u_3) and I_k = 3
- Satisfaction of u_3 with feedback(u_3) and new I_k = 4

3. Aggregated Voting and Overlap Similarity

- Feedback box returns pref(u_3) as feedback(u_3)
- Not possible to improve user satisfaction with any other preferences

Key Theoretical Results

Recommendation Box:

Setting	Complexity	Proposed Solution
R-AGS	PTIME	Optimal, $O(mn)$
R-AGD	NP-Complete	Approximate, $O(mn)$
R-LMS	NP-Complete	Approximate, $O(kn)$
R-LMD	NP-Complete	Approximate, Polynomial

Feedback Box:

Setting	Effective?	Complexity	Proposed Solution
FB-AGS	No	NP-Complete	Optimal, Linear
FB-AGD	Yes	NP-Complete	Approximate, Polynomial*
FB-LMS	Yes	NP-Complete	Optimal, Linear*
FB-LMD	Yes	NP-Complete	Approximate, Polynomial*

Complexity using an Oracle for Recommendation Box.

Robustness of Group Recommendations

- Ensures generated group recommendations are not *too* different from prior iterations
- Enforced as a *soft* constraint
- Picks an optimal recommendation most similar to previous recommendation
- Implemented via addition of pseudo-users with previous recommendation as preference

Experimental Study Highlights

Feedback Box:

- Feedback box was always beneficial although the utility varies
- Feedback box most useful for Least Misery consensus function
- Feedback box most useful for highly similar groups
- Utility verified both qualitatively and by user studies

Scalability:

- Our algorithms were highly scalable despite NP-Hardness
- Our Recommendation algorithms were scalable for large groups and preferences
- Our feedbox algorithms required only linear invocations of recommendation box

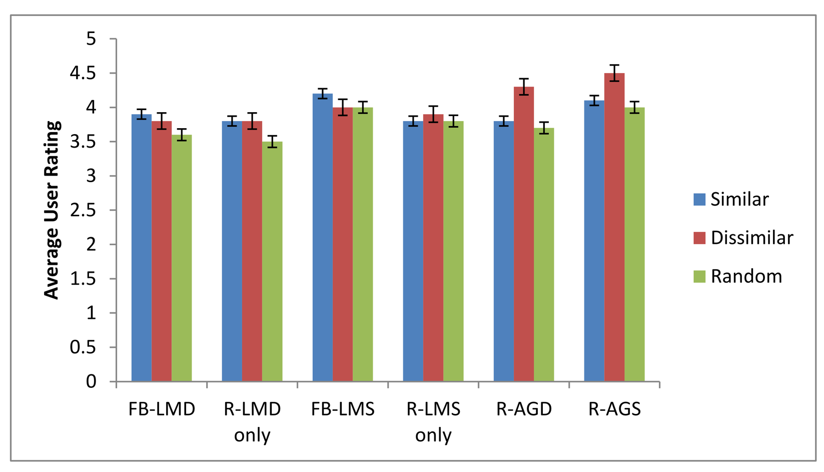
Experimental Results

Datasets:

- MovieLens - 71K users, 10K movies
- Flickr and Lonely Planet - 60K users, 1000 POIs

User Study:

- Flickr Dataset, 270 users, 439 POIs
- Feedback box was universally preferred



User Study: User Rating for Recommendations

Future Directions

1. Exploiting feedback box for a class of recommendation functions.
2. A cost model that will maximize user satisfaction by certain percentage.
3. Invoking Feedback box for a subset of users, as opposed to individual user.

Key References

- S.Amer-Yahia, S.B.Roy, A.Chawla, G.Das, and C.Yu, "Group recommendation: Semantics and efficiency," PVLDB, 2009.
- G.L. Nemhauser et.al, "An analysis of approximations for maximizing submodular set functions-I," Mathematical Programming, 1978.