EXPLOITING GROUP RECOMMENDATION FUNCTIONS FOR FLEXIBLE PREFERENCES

Senjuti Basu Roy – UW Tacoma
Saravanan Thirumuruganathan – UT Arlington
Sihem Amer-Yahia – CNRS LIG
Gautam Das – UT Arlington
Cong Yu – Google Research

Motivation

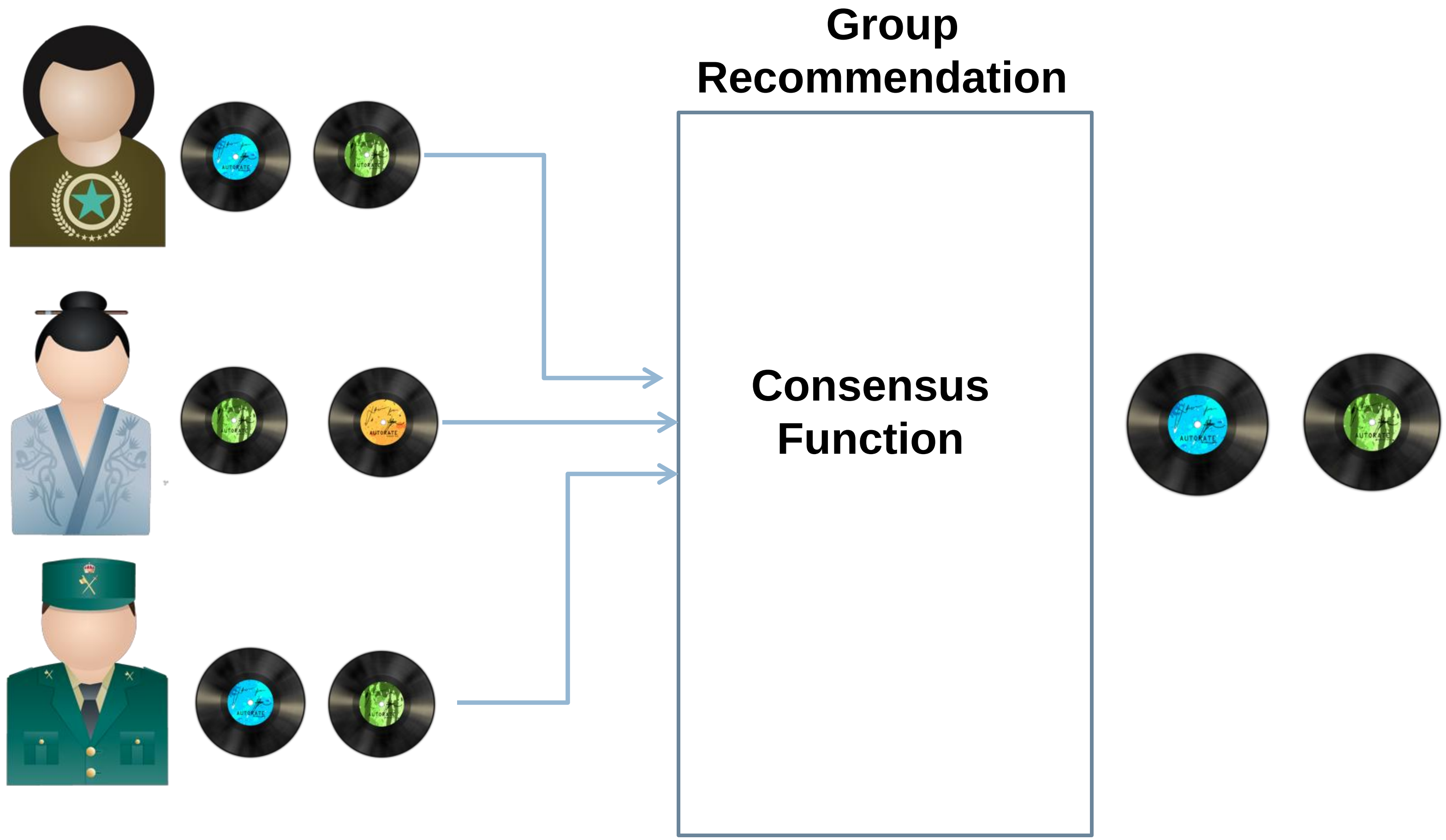
business Lunch

movies for a group of friends

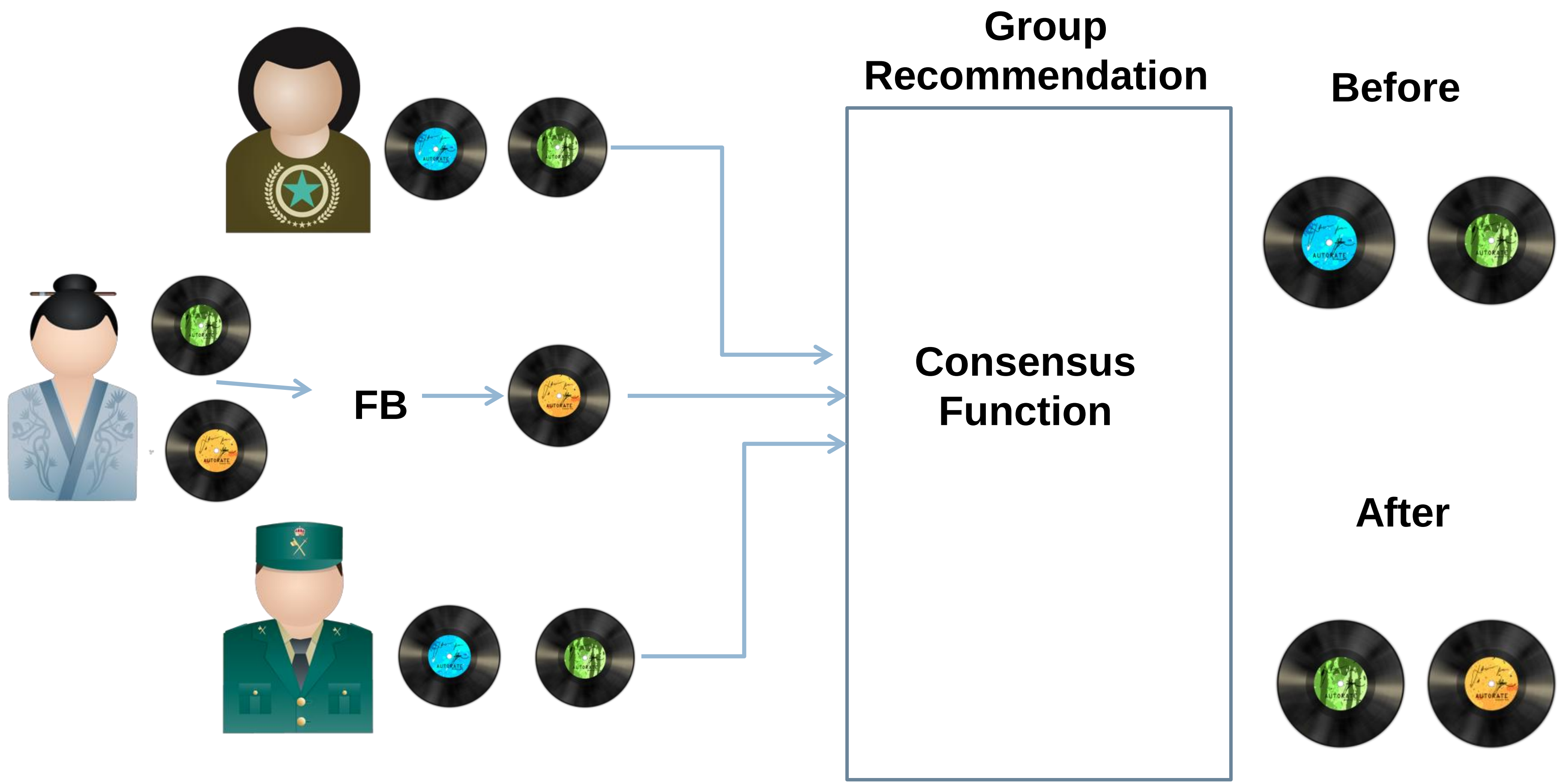
playlist of songs for a party

a travel destination for family

Motivation



Motivation



Contributions

- Boolean preference model
  - Gives rise to unexpected challenges
- Algorithms for group recommendation
- Algorithms for maximizing individual preferences (Response Box)
- Recommendation Robustness

Boolean Preference Model

Items

Users

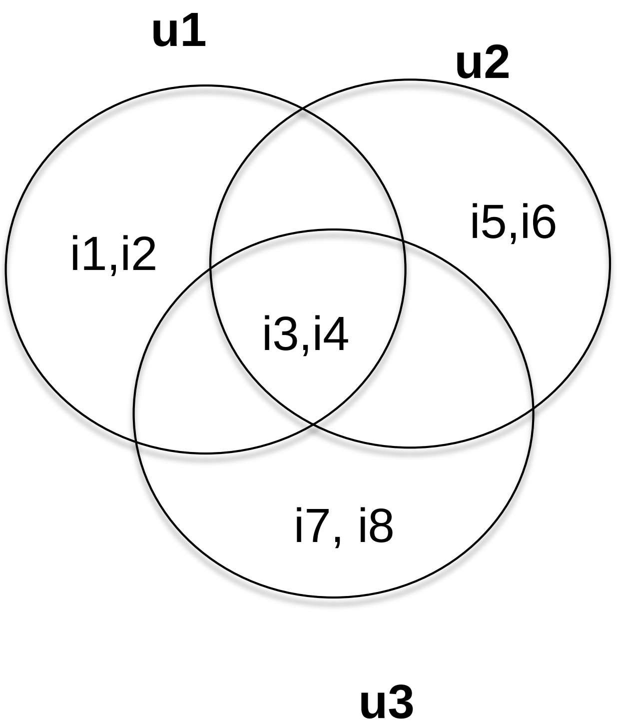
	i1	i2	i3	i4	i5	i6	i7	i8
u1	1	1	1	1	0	0	0	0
u2	0	0	1	1	1	1	0	0
u3	0	0	1	1	0	0	1	1

Boolean Preference Model

Items

Users

	i1	i2	i3	i4	i5	i6	i7	i8
u1	1	1	1	1	0	0	0	0
u2	0	0	1	1	1	1	0	0
u3	0	0	1	1	0	0	1	1



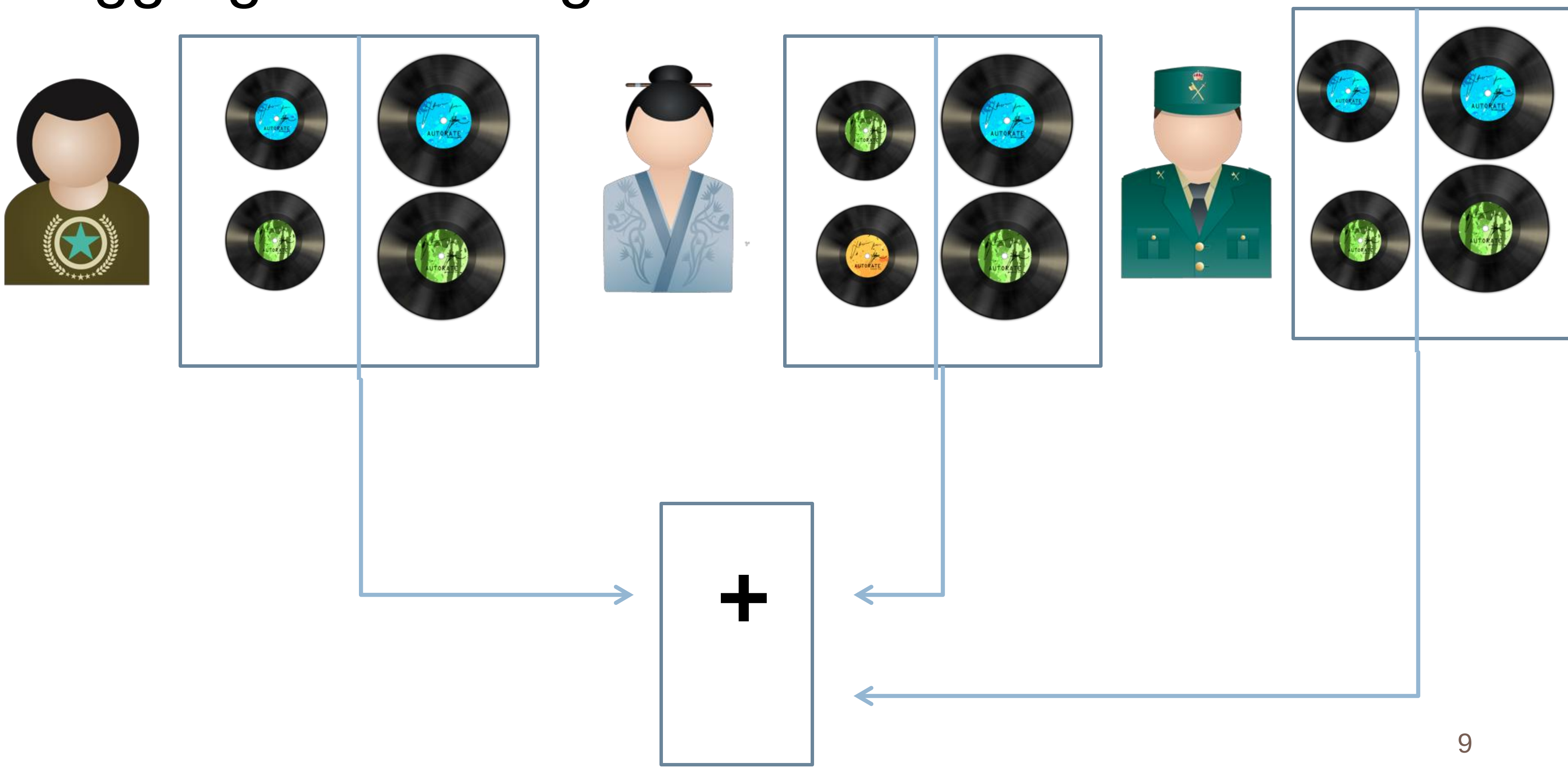
Which k <= 5 items to recommend

User Satisfaction

- User preference A, recommendation B
- Jaccard Index: $J(A,B) = \dfrac{|A \cap B|}{|A \cup B|}.$
- Overlap Similarity : $OS(A, B) = |A \cap B|$
- Hamming Distance

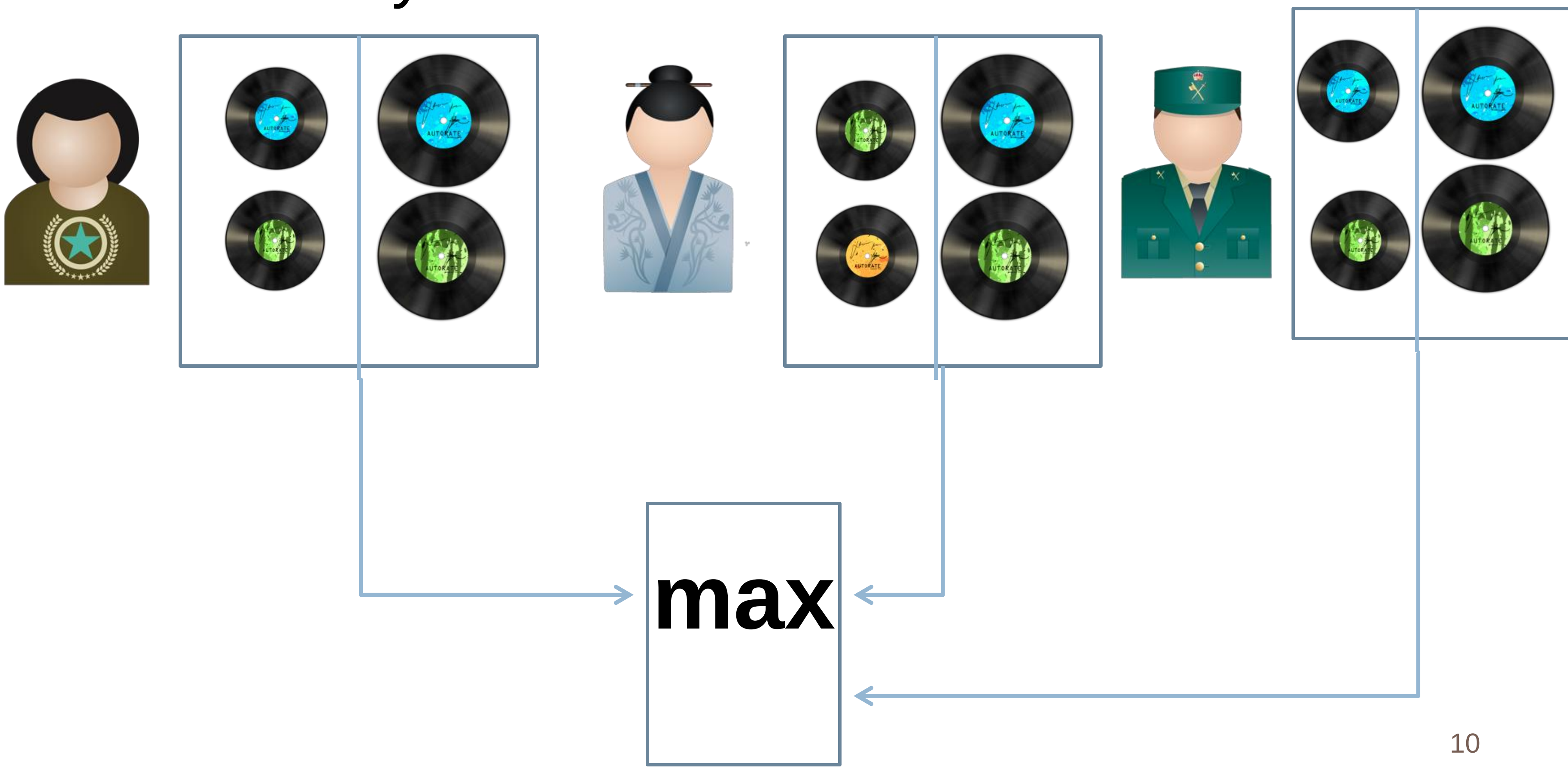
Group Consensus

- Aggregated Voting

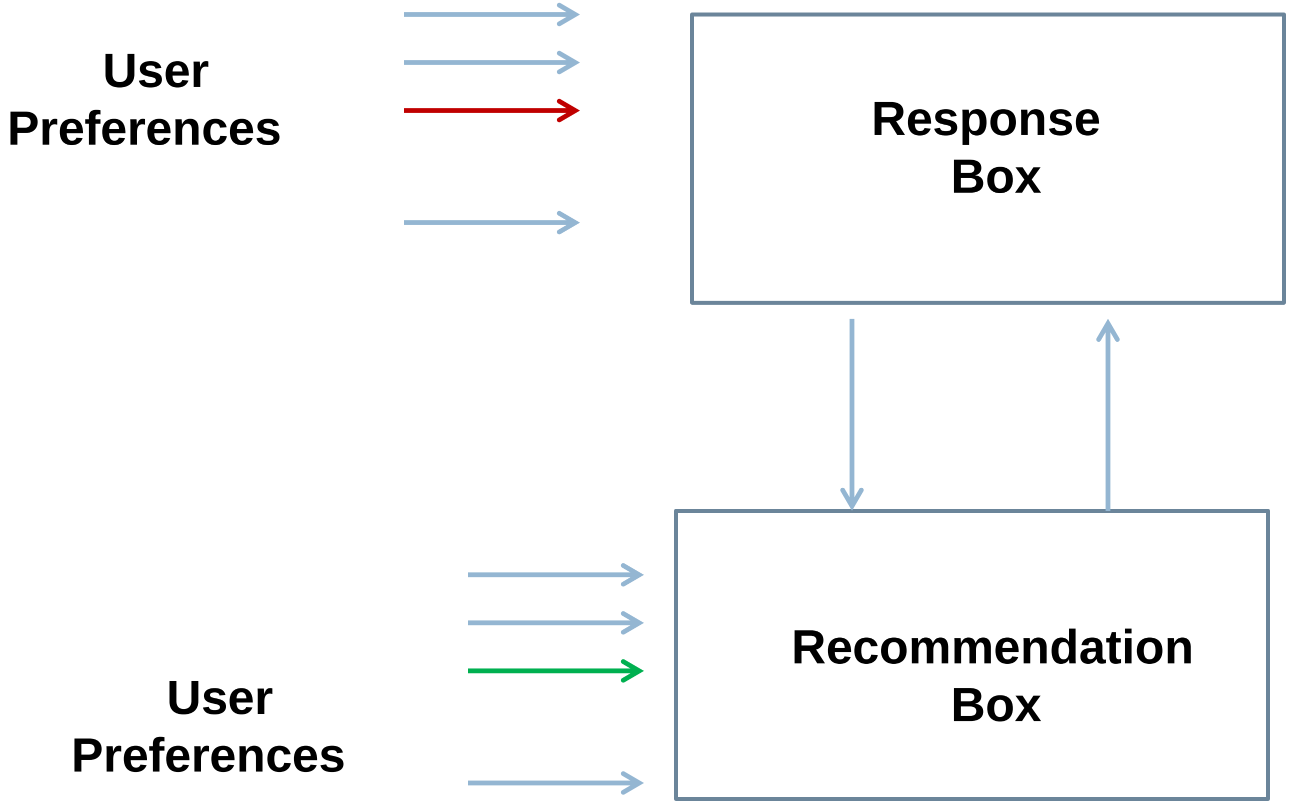


Group Consensus

- Least Misery



Pipeline

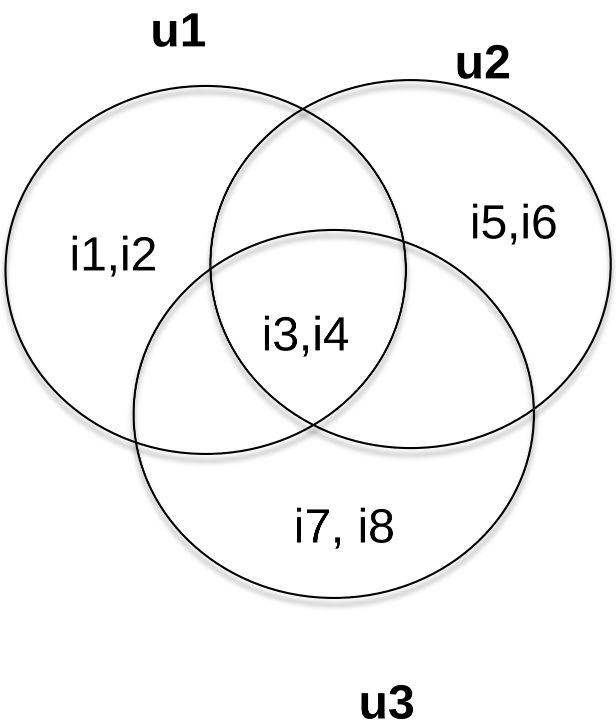

User
Preferences
Response
Box
Recommendation
Box
User
Preferences

Recommendation Box

- Aggregated Voting
  - Overlap Similarity
  - Hamming
- Least Misery
  - Overlap Similarity
  - Hamming

Recommendation Box

- Aggregated Voting and Overlap Similarity, k <=5



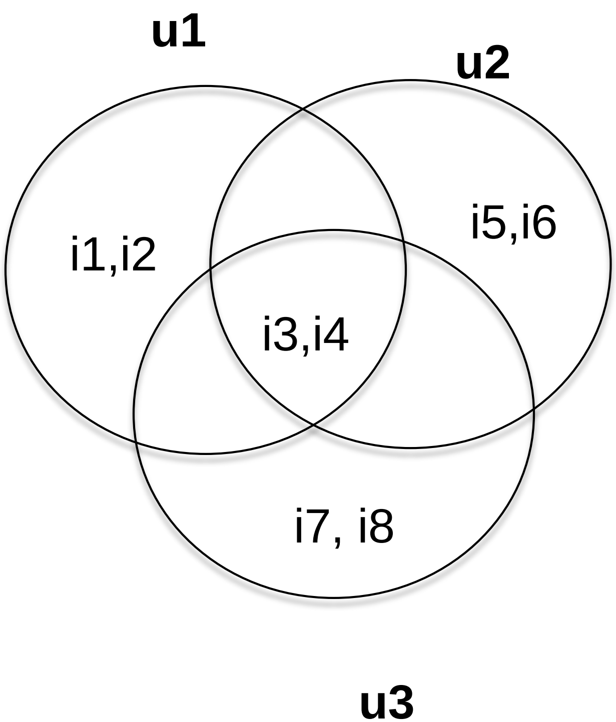
items	i1	I2	I3	I4	I5	I6	I7	i8
Pref(u1)	1	1	1	1	0	0	0	0
I^k	1	0	1	1	1	0	1	0

Recommendation = {i1,i3, i4, i5, i7}

Overlap Similarity between (u1, I^k) = 3
Overlap Similarity between (u3, I^k) = 3
Overlap Similarity between (u2, I^k) = 3
Aggregated Voting = 3 + 3 + 3 = 9

Recommendation Box

- Least Misery and Overlap Similarity, k <=5



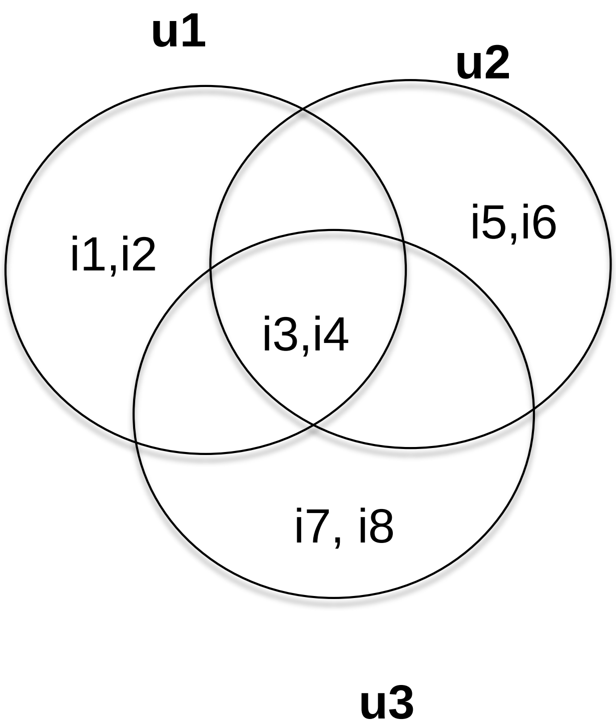
Recommended Items to G , **I^k = {i1,i3, i4, i5, i7}**, that **maximizes the minimum overlap similarity** of 3

Misery between (u1, I^k) under Overlap Similarity = 3
Misery between (u2, I^k) under Overlap Similarity = 3
Misery between (u3, I^k) under Overlap Similarity = 3

Least Misery = min (3,3,3) = 3

Recommendation Box

- Aggregated Voting and Hamming, k <=5



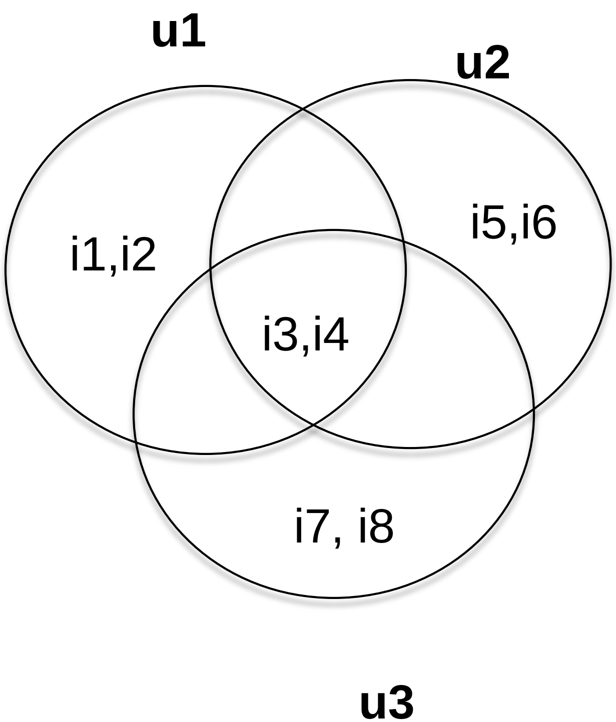
items	i1	I2	I3	I4	I5	I6	I7	i8
Pref(u1)	1	1	1	1	0	0	0	0
I^k	1	0	1	1	1	0	1	0

Recommendation = {i1,i3, i4, i5, i7}

Hamming distance between (u1, I^k) = 3
Hamming distance between (u3, I^k) = 3
Hamming distance between (u2, I^k) = 3
Aggregated Voting = 3 + 3 + 3 = 9

Recommendation Box

- Least Misery and Hamming, k <=5

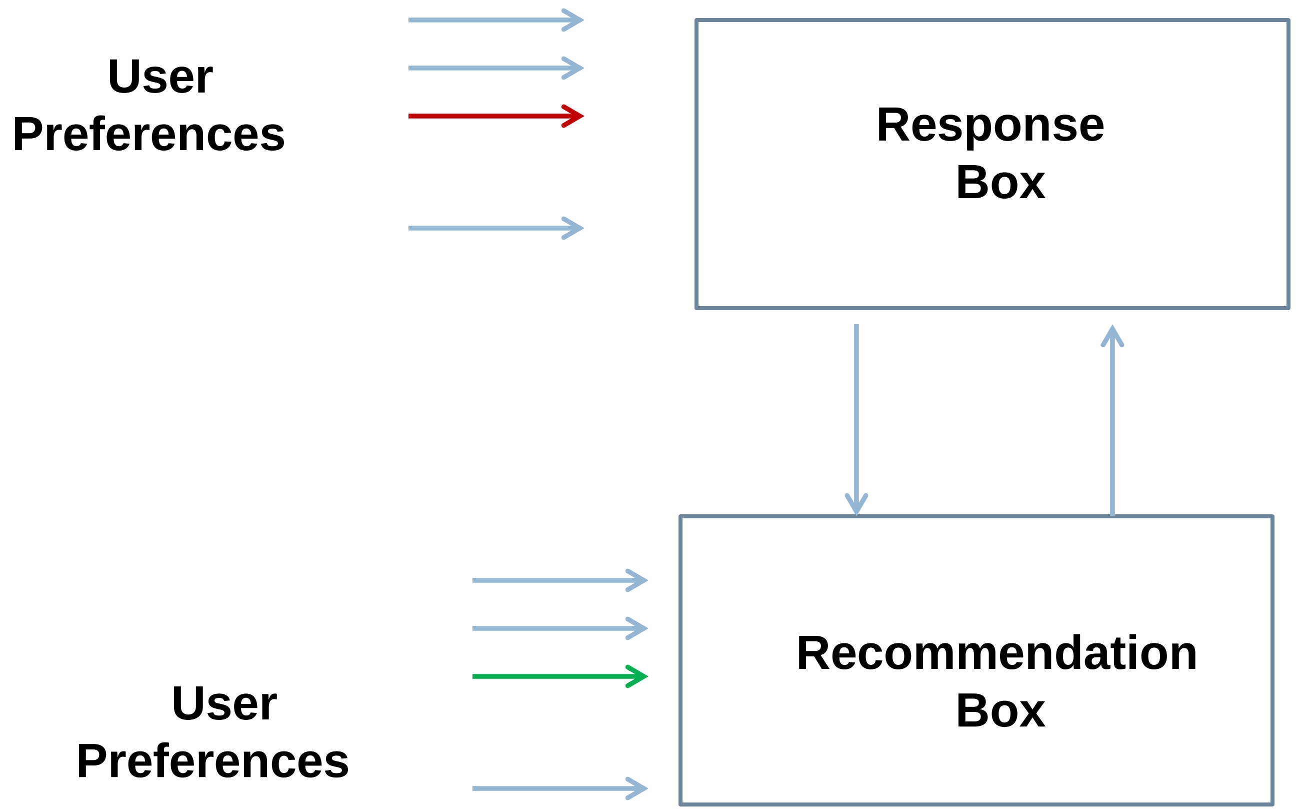


Recommended Items to G , **I^k = {i1,i3, i4, i5, i7}**, that **maximizes the minimum Hamming distance** of 3

Misery between (u1, I^k) under Hamming distance = 3
Misery between (u2, I^k) under Hamming distance = 3
Misery between (u3, I^k) under Hamming distance = 3

Least Misery = min(3,3,3) = 3

Pipeline


User
Preferences
Response
Box
Recommendation
Box
User
Preferences

Response Box

Least Misery, Overlap Similarity k <=5

- u3's satisfaction in generated recommendation, $(u3, I^k) = 3$
- What if, Feedback box internally changes pref(u3) to a different feedback, feedback(u3) = {i7,i8}
- Using pref(u1), pref(u2), feedback(u3), generated $I^{k'} = \{i1, i3, i4, i7, i8\}$

items	I1	I2	I3	I4	I5	I6	I7	i8
Pref(u3)	0	0	**1**	**1**	0	0	1	1
I^k = satisfaction 3	1	0	1	1	1	0	1	0
$I^{k''}$ = satisfaction 4	1	0	1	1	0	0	1	1

Summary of Results

	Aggregated Voting	Least Misery
Overlap Similarity	**Recommendation Generation:** Proposed algorithm: Optimal Algorithm Complexity: $O(m \times n)$;	**Recommendation Generation:** NP-hard Problem Proposed algorithm: Greedy Approximation Algorithm Approximation factor: $(1-1/e)$; Complexity: $O(k \times n)$;
	Feedback Box: **not useful**	**Feedback Box:** **Useful** NP-hard (as the recommendation counterpart is NP-hard) Proposed algorithm: Optimal Algorithm considering an oracle for recommendation computation; Complexity: $O(m)$;

Summary of Results

	Aggregated Voting	Least Misery
Hamming Distance	**Recommendation Generation:** NP-hard problem Proposed algorithm: based on centroid computation relaxing integrality constraint, followed by deterministic rounding. Complexity: O(m×n); **Feedback Box: Useful** NP-hard problem (as the recommendation counterpart is NP-hard) Proposed algorithm: based on Quadratic Programming formulation relaxing integrality constraint, followed by deterministic rounding. Complexity: polynomial, as it becomes a convex optimization problem on positive definite matrices;	**Recommendation Generation:** NP-hard problem Proposed algorithm: based on Quadratic Programming formulation relaxing integrality constraint, followed by deterministic rounding; Complexity: polynomial, as it becomes a convex optimization problem on positive definite matrices; **Feedback Box: Useful** NP-hard problem (as the recommendation counterpart is NP-hard) Proposed algorithm: based on Quadratic Programming formulation relaxing integrality constraints, followed by deterministic rounding. Complexity: polynomial, as it becomes a convex optimization problem on positive definite matrices;

Recommendation Algorithm

- Least Misery and Overlap Similarity, k <=5
  - NP-Hard problem (from Hitting set)
- Greedy Approximation
  - Satisfy the most miserable user at each step
  - Approximation factor of 1 – 1/e

Response Box Algorithm

- Least Misery and Overlap Similarity, k <=5
- NP-Hard
- Optimal algorithm with recommendation oracle
- Optimal Feedback(u) should contain items (Y-X)
  - Y is the set of items present in I_k excluding u
  - X = (Ik $\wedge$ pref(u))

Experimental Results

- Datasets
  - Movie lens 10 M user rating file
  - Flickr data, 12 popular cities across the world from LonelyPlanet
- User Feedback from Flickr Log, Movielens
- Performed experiments
  - Performance Experiments
  - Quality Experiments (Offline and Online User Studies in AMT for 200 users)

Experimental Study Highlights

- Usefulness of feedback box
  - For any recommendation function and similarity measure, this box was found useful
  - The degree of usefulness varies
  - Feedback box most useful for Least Misery Consensus function
  - Feedback box most useful for similar group
- Performance Results
  - Despite being NP-hard, proposed algorithms are highly scalable

Summary

- Usefulness of feedback box
  - For any recommendation function and similarity measure, this box was found useful
  - The degree of usefulness varies
  - Feedback box most useful for Least Misery Consensus function
  - Feedback box most useful for similar group
- Performance Results
  - Despite being NP-hard, proposed algorithms are highly scalable

Experimental Study Highlights

- Efficient Algorithms are necessary for online Recommendation Computation for Individual Users and Group of users
- Enabling Interactivity raises opportunities and computational challenges
- Binary Preference Model raises unexpected computational challenges for both the cases
- A feedback box maximizes the individual user preference under the semantics of traditional group recommendation functions.