Ant-Man

Poorwa Hirve **Payoj Jain**

Given a list of movies by a movie recommender system, which type of recommendations are you more likely to prefer?

A) Movies recommended based on your historical likings

B) Movies based on your likes, and have been rated by some of your friends whose views you trust

C) List of Popular movies

D) A random list of movies

Leveraging Social Connections to Improve Personalized Ranking for Collaborative Filtering

Tong Zhao	The Chinese University of Hong Kong, Hong Kong
Julian McAuley	University of California San Diego, La Jolla, CA
Irwin King	The Chinese University of Hong Kong, Hong Kong

Published in CIKM '14. Shanghai, China, November 2014.

23rd ACM International Conference on Conference on Information and Knowledge Management

Personalized Ranking

- Estimating relative preferences of users in the form of ranking
  - Collaborative filtering can identify users who have similar rankings to you
  - BPR
- Another source which influences user preferences:
  - Social Connections
  - S-BPR
- Ranking Methods
  - Point-wise methods focus on fitting the numeric values of the data
  - Pairwise methods focus on model the preference order of the data

Advantages of Social Connections

- Understand interactions with others along with user preferences
  - A user’s preference can be related to their friend’s preferences
- Can be **more** informative than the user’s feedback
  - One-to-many relationship
  - Increase in data
  - Improved Cold-start recommendations

Leveraging Social Connections

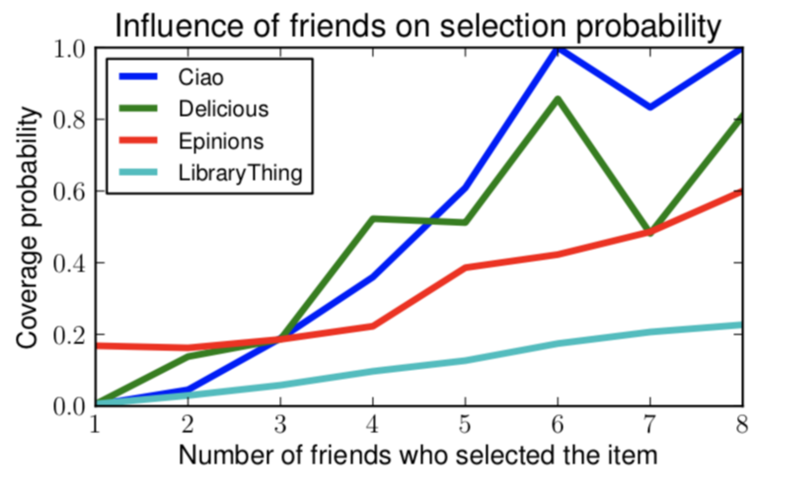
- Main goal of this paper in order to build better models for user preferences
- Assumes explicit numerical ratings
  - In the real world, implicit feedback would have to be taken into consideration
- Tries to understand the underlying mechanisms of how certain items chosen by users' friends influence their own decision
- Datasets:
  - Ciao
  - Delicious (we have used FilmTrust due to unavailability of this dataset)
  - LibraryThing (Lthing)
  - Epinions

Coverage Probability Analysis

Probability of a user selecting an item which their friends have chosen.

Users become monotonically more likely to select an item as more of their friends select it.

Implies that the friends' selections drives the users' selections independently.



Algorithmic Idea

Assumptions 1

Pairwise preference comparison:

item(i) user(u) has selected (x_{ui}) **>**

item(k) their friends have selected (x_{uk}) **>**

item(j) neither the user nor their friends have selected (x_{uj})

$$\boxed{x_{ui} > x_{uk} > x_{uj}}$$

Assumptions 2

Pairwise preference comparison:

1) item(i) user(u) has selected (x_{ui}) **>**

   item(k) their friends have selected (x_{uk})

2) item(i) user(u) has selected (x_{ui}) **>**

   item(j) neither the user nor their friends have selected (x_{uj})

$$\boxed{x_{ui} > x_{uk}\ ,\ x_{ui} > x_{uj}}$$

Contributions

1. Developing a ranking algorithm: Social-BPR (SBPR)
2. Evaluate this model on the datasets and show how the proposed model significantly improves item recommendation performance
3. Experiment on cold-start recommendation problems and note the significant improvements

Methodology

Data Description

- Ciao: Product review website
- Delicious: Bookmarking website
- LibraryThing: Book reviewing website
- Epinions: Consumer review website

Table 1: Statistics of the Datasets.

	Ciao	Delicious	Lthing	Epinions
#Users	1,705	1,670	73,882	41,554
#Item	12,252	52,613	337,561	112,991
#Observed feedback	22,839	71,105	979,053	181,394
#Social relations	47,842	13,246	12,0536	181,304
#Average P've-feedback	13	42	13.25	4.3651
#Average SP've-feedback	342	241	101.08	11.437

Data Selection and Processing

- Filter out ratings greater than or equal to **4.0**
  - Top-N recommendation problem
- Consider the users who have less interactions to be cold-start users
  - Number of interactions less than 5

Problem Definition

User set: **U** (|M| users)

Item set: **I** (|N| items)

Social network **G = (U, E)**, where $(u,v) \in E$

- **Observed items:** $O_u \in I$ for a user u
- **Unobserved items:** $\overline{O_u} \in I$
- **Positive feedback:** $P_u = \{(u, i)\}$, where $i \in O_u$
- **Social feedback:** $SP_u = \{(u, k)\}$, where $k \in O_v \cap \overline{O_u}$,v is friend of u
- **Negative feedback:** $N_u = \{(u, j)\}$, where $j \in \overline{O_u} \cap \overline{O_v} \cap \ldots \cap \overline{O_{v'}}$
- **Social Coefficient:** s_{uk} (constant, common neighbors, common preferences)

Model Assumption

- SBPR-1:
  - $x_{ui} \geqslant x_{uk}$, $x_{ui} \geqslant x_{uj}$, $i \in P_u$, $k \in SP_u$, $j \in N_u$
  - User u prefers its own rated item i over friend's item k, and prefers its item i over an item not observed
- SBPR-2:
  - $x_{ui} \geqslant x_{uk}$, $x_{uk} \geqslant x_{uj}$, $i \in P_u$, $k \in SP_u$, $j \in N_u$
  - User u prefers its own rated item i over friend's item k, and prefers friend's item k over an item j not observed

$$\boxed{x_{ui} = x_u^* \, y_i \,,\; x_{uk} = x_u^* \, y_k \,,\; x_{uj} = x_u^* \, y_j}$$

Posterior Probability for BPR

Let Θ be parameter of the model that determines the personalized ranking. BPR's goal is to maximise the posterior probability:

$$p(\Theta \mid i >_u j) \propto p(i >_u j \mid \Theta).p(\Theta), \quad \Theta \sim \{x_u, y_i, y_j, y_k\} \text{ for SBPR}$$

To maximize posterior, we need to maximize the likelihood. Thus, for maximizing the likelihood, we find the point where the gradient of this likelihood w.r.t. to parameter Θ. For optimisation we perform Stochastic gradient descent.

Objective Function

goal is to maximize the following objective function

$$\sum_{u}\left[\sum_{i\in P_u}\sum_{k\in SP_u}\ln(\sigma(\frac{x_{ui}-x_{uk}}{1+s_{uk}}))+\sum_{k\in SP_u}\sum_{j\in N_u}\ln(\sigma(x_{uk}-x_{uj}))\right]-regularization$$

$x_{ui} = x_u * y_i$, $x_{uk} = x_u * y_k$, $x_{uj} = x_u * y_j$,

s_{uk} is social coefficient

Social coefficient Analysis

- Controls the contribution of social feedback
- Social strength from related users, indicating the preference distance between positive feedback and social feedback.
    - Constant: giving a constant value to each social feedback
    - Common neighbor: more mutual friends implies strong social coefficient which means preference to peers' item is high
    - Common preference: more common items between two friends implies strong social coefficient

Sampling Strategy

- **Uniform Sampling**: Randomly selecting a user and items from Positive, Social and Negative feedback for that user
- **Static Sampling**: Positive feedback is selected uniformly but social feedback is drawn according to Geometric distribution
- **Adaptive Sampling**: Similar to static sampling. Social and negative feedback selections have preference score instead of a static value
- **Dynamic negative sampling**: Negative feedback instances are sampled according to the preference scoring function

Why we chose this paper

- Not much work has been done in ranking optimization using social connections
- Interesting to experiment with how social connections affect ranking
- Simple enough to understand and implement
- Variety of analysis done: AUC, Recall, NDCG, Cold-start
    - Get to implement evaluation metrics, analyze cold-start problem and come up with the objective function for SBPR-1

Which of the following assumptions regarding social feedback can be applied to Social-BPR?

A) User u prefers its rated items (u, i) over its peers' items (u, k). User u prefers its rated items (u, i) over its unrated items (u, j).

B) User u prefers its peers' items (u, k) over its unrated items (u, j). User u prefers its peers' items (u, k) over its items (u, i).

C) User u prefers its rated items (u, i) over its peers' items (u, k). User u prefers its peers' items (u, k) over its unrated items (u, j).

D) A, C

E) B, C

What kind of ranking method is SBPR?

A) Pointwise ranking method

B) Pairwise ranking method

C) Both

D) Neither

Project implementation

https://github.com/payoj21/social-network-recommender-system

Overview of Challenges

- Documentation of the paper
- Data Processing and data structuring
- Cold-Start Users Analysis
- Implementation of BPR and SBPR
- Train-Test Split

Documentation challenges

- Links provided to the datasets did not work
- Delicious dataset was not available
  - Replaced it with FilmTrust
- Interpretation of the problem statement
  - O_U, $\sim O_u$, SP_u, N_u
- Model parameters such as learning rate, regularization
  - Performed many experiments before setting the learning rates for different datasets
- SBPR1-Opt was not provided
  - We formulated the objective function on our own

Data Processing

- Lthing and Epinions were extremely large datasets
  - Billion data points
  - Reduced the datasets to smaller sizes
    - Removed users who have rated 5 ratings or lesser
    - Considered top 15 % of unique users and 15 % of unique items
    - Considered cold start users with number of items rated as ≤ 6

- We are solving a Top-N recommendation problem
  - Eliminate user rating data for ratings 3 and below

- Two .txt files: **ratings_data.txt** and **trust_data.txt**
  - Considered an overlap in the users occurring in both

- Sparse data
  - Solution: Sparse matrices

Cold-Start Users

- Paper has explained how users with ≤ 5 ratings are treated as cold start users
- No explanation for those users who have not rated anything nor do they have any social graph
- We have not considered such people in our model

Implementation of BPR

- LKPy does not have BPR implemented

- We referred to a version of MyMediaLite
  - C# implementation
  - Thanks to Professor Burke for providing a head start with this!

- Extending BPR to SBPR was challenging
  - We ran into NaN errors
  - Learned proper representation of item factors

- We used LKPy for metrics → NDCG, Recall@k

Train - Test Split

- Normally:
  - Divide train and test by some fraction, e.g. 0.2
  - Or, LKPy as we did in our assignments (partition_users)

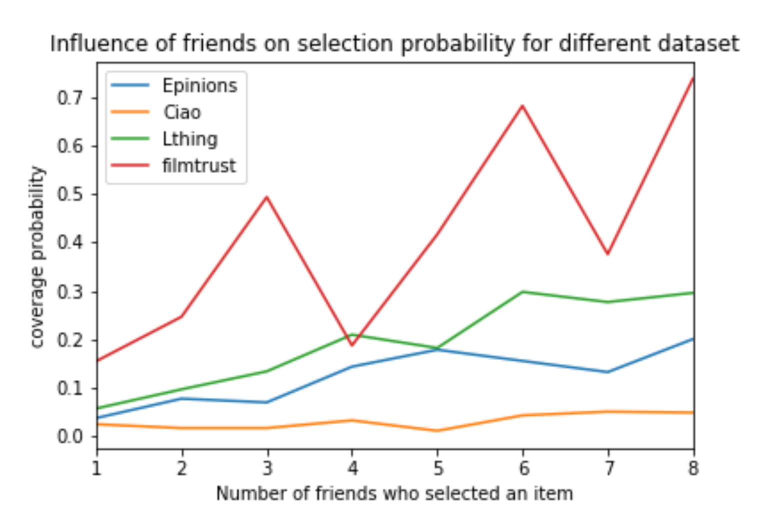
- We are using sparse matrices
  - Difficulty in understanding how data is represented
  - How to access the elements

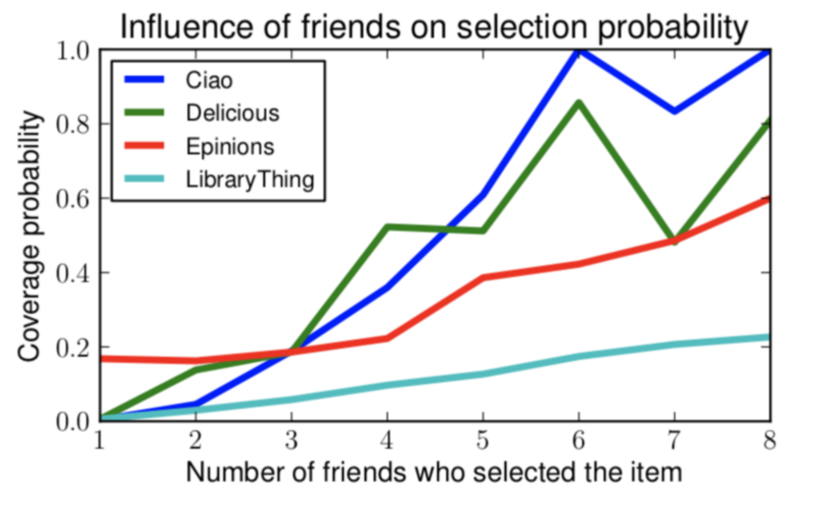
- Split by user
  - Per user, 0.1 samples go to test and 0.9 go to train

- Cross fold validation implemented over 10 folds

Accomplishments

- Social influence for user selection





BPR vs SBPR (Recommendations)

	user	item	score	rank
5	3	18	2.278096	1
6	3	661	2.202410	2
7	3	1309	2.061484	3
8	3	765	2.045426	4
9	3	315	2.033914	5

	user	item	score	rank
5	3	2049	274.739978	1
6	3	1356	6.608586	2
7	3	110	5.024261	3
8	3	1742	4.899271	4
9	3	30	4.842967	5

BPR vs SBPR (NDCG and Recall@5)

user	ndcg	recall
3	0.060790	0.015152
21	0.350497	0.161290
22	0.092858	0.125000
30	0.068066	0.055556
34	0.081963	0.100000

user	ndcg	recall
3	0.064535	0.030303
7	0.177148	0.200000
21	0.350497	0.161290
22	0.228893	0.250000
25	0.127371	0.111111

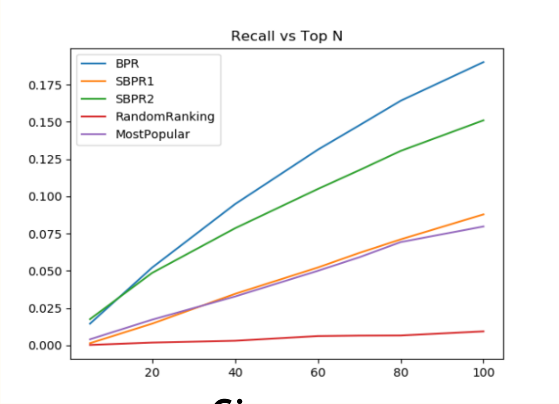
Table 1 : Recommendation performance of different methods on four real-world datasets. The last column shows the improvement of the proposed method compared with the best baseline method

Datasets	Metrics	Rand	Most Popular	BPR	SBPR1	SBPR2	% Improvement
Ciao	**R@5**	0.0004	0.0040	0.0145	0.0014	**0.0176**	**20.8445**
	R@10	0.0011	0.0087	0.0280	0.0050	**0.0287**	**2.5517**
	NDCG	0.0012	0.0089	0.0280	0.0041	**0.0319**	**13.9586**
	AUC	0.5000	0.5006	**0.9100**	0.8740	0.8396	**-7.7370**
filmTrust	**R@5**	0.0050	0.0088	0.0522	**0.1249**	0.1011	**139.2466**
	R@10	0.0106	0.0193	0.1100	**0.2303**	0.2011	**109.2655**
	NDCG	0.0113	0.0202	0.1172	**0.2365**	0.2055	**101.7075**
	AUC	0.4996	0.5057	0.9018	**0.9552**	0.9345	**5.9245**
Lthing	**R@5**	0.0013	0.0030	**0.0170**	0.0140	0.0072	**-17.6004**
	R@10	0.0019	0.0063	**0.0298**	0.0255	0.0124	**-14.4220**
	NDCG	0.0025	0.0095	**0.0454**	0.0372	0.0175	**-18.2214**
	AUC	0.4999	0.5002	**0.8144**	0.7954	0.6593	**-2.3349**
Epinions	**R@5**	0.0006	0.0056	0.0252	**0.0276**	0.0096	**9.3097**
	R@10	0.0019	0.0122	0.0453	**0.0487**	0.0183	**7.7050**
	NDCG	0.0022	0.0142	0.0525	**0.0567**	0.0203	**8.0806**
	AUC	0.4999	0.5005	**0.9272**	0.9272	0.8278	**-0.0060**

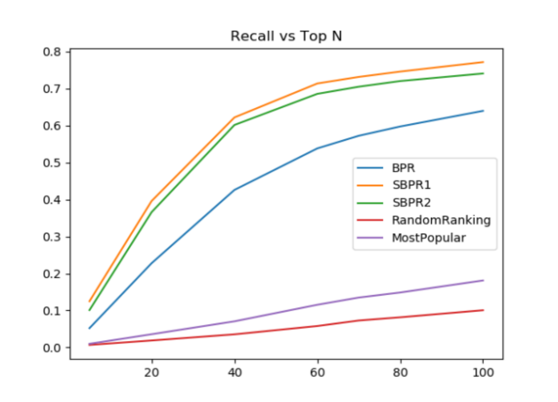
Table 2 : Recommendation for cold-start users.

Datasets	Metrics	Rand	Most Popular	BPR	SBPR1	SBPR2	% Improvement
Ciao	**R@5**	0.0003	0.0043	0.0182	0.0014	**0.0203**	**11.7333**
	R@10	0.0014	0.0097	0.0321	0.0055	**0.0321**	**0.0000**
	NDCG	0.0010	0.0073	0.0248	0.0031	**0.0284**	**14.4049**
	AUC	0.4820	0.5002	0.6187	0.6284	**0.6364**	**2.8607**
filmTrust	**R@5**	0.0041	0.0079	0.0143	**0.1322**	0.0808	**823.6842**
	R@10	0.0102	0.0198	0.0416	**0.2371**	0.1766	**469.6833**
	NDCG	0.0064	0.0157	0.0250	**0.1906**	0.1299	**662.5596**
	AUC	0.4969	0.5006	0.6046	**0.6237**	0.6184	**3.1575**
Lthing	**R@5**	0.0022	0.0014	0.0145	**0.0159**	0.0036	**10.0000**
	R@10	0.0014	0.0051	**0.0283**	0.0239	0.0109	**-15.3846**
	NDCG	0.0006	0.0033	**0.0240**	0.0207	0.0062	**-13.7380**
	AUC	0.5000	0.5000	**0.5174**	0.5166	0.5111	**-0.1487**
Epinions	**R@5**	0.0000	0.0046	0.0239	**0.0293**	0.0131	**22.5806**
	R@10	0.0031	0.0093	0.0463	**0.0502**	0.0231	**8.3333**
	NDCG	0.0022	0.0092	0.0378	**0.0407**	0.0163	**7.7283**
	AUC	0.5100	0.5003	0.5450	**0.5499**	0.5389	**0.9016**

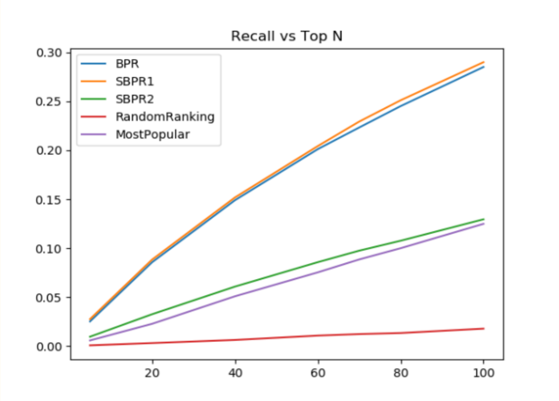
Recall plots



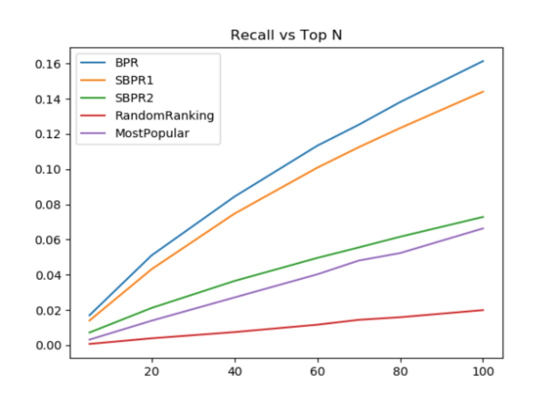
Ciao



FilmTrust

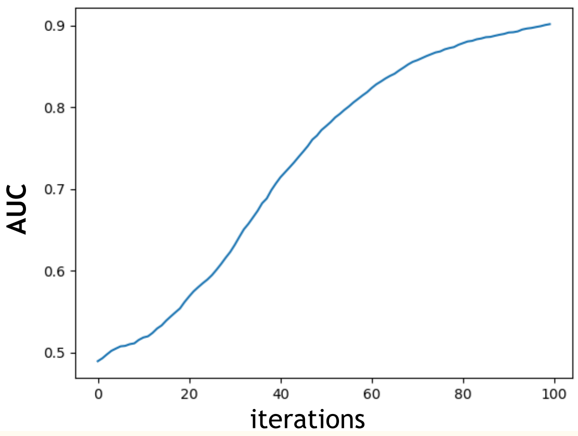


Epinions

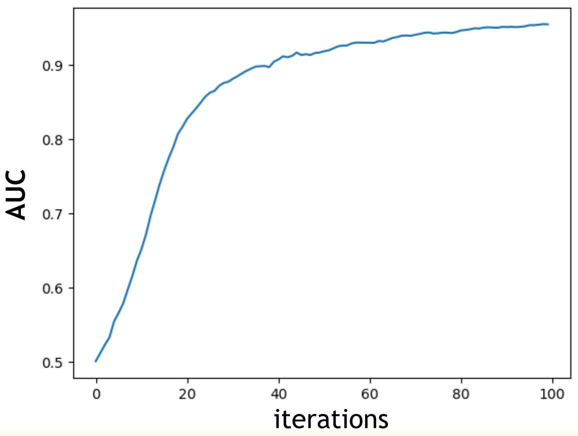


Lthing

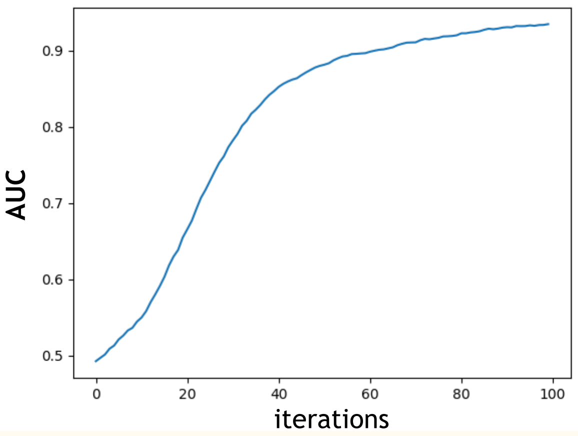
Convergence for filmTrust dataset



BPR

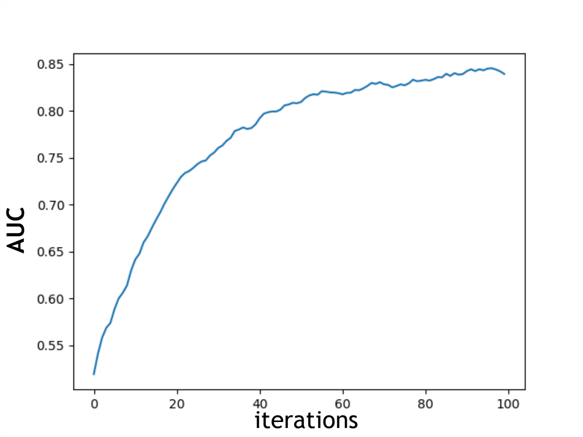


SBPR1

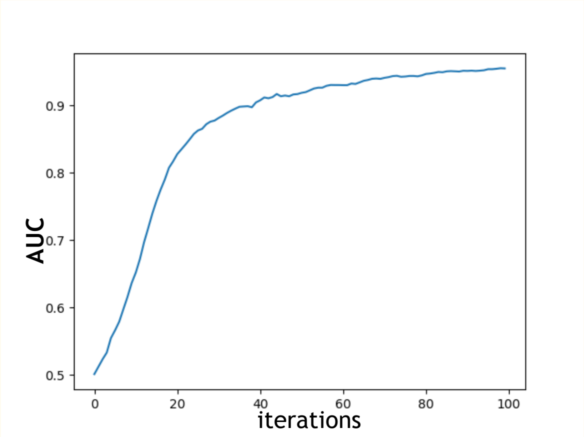


SBPR2

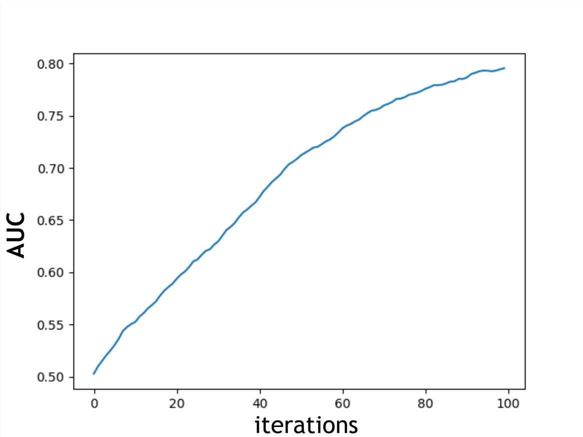
Convergence for SBPR



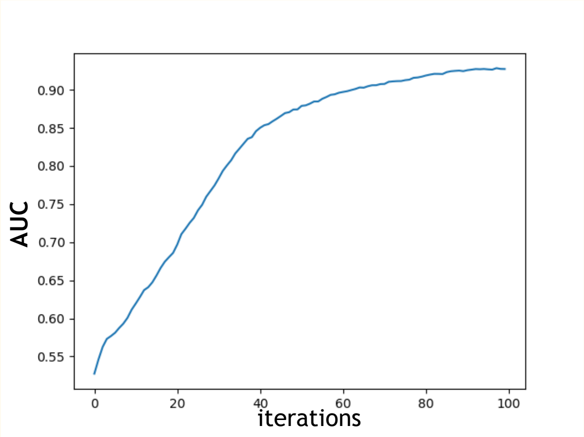
Ciao



filmTrust



Lthing



Epinions

Limitations of Results

- Our results have come from trial and error, and not using Grid Search
- The entire data for Lthing and Epinions not used
- Did not implement all the algorithms
  - Do not know how our interpretation of SBPR compares to those
- We have implemented Uniform Sampling
  - Paper shows to have best results with Static Sampling

If we had more time...

- Use other sampling techniques
  - Static
  - Adaptive
  - Dynamic Negative
- Implement other algorithms
  - GBPR
  - MR-BPR
  - WRMF
  - MMMF

If we had more time...

- Social coefficient analysis
  - Common Neighbors
  - Common Preferences
- Implement our own version of SBPR
  - Negative feedback of a user < negative feedback of his peer
- Try training on the entire dataset for Lthing and Epinions

Thank you