TSINGHUA UNIVERSITY

FISM: Factored Item Similarity Models for Top-N Recommender Systems

Santosh Kabbur

Computer Science & Engineering, UMN

Present by *Feng Xie*

October 16, 2013

Outline

- Background
- Motivation
- FISM
  - Basic FISM
  - FISMrmse
  - FISMauc
- Evaluation
- Conclusion

What is recommender system?

Information Overload

百万首页 中文版
1baiwan.com
•一万个格子 •一个格子卖价100元
我们的格子值得买吗？
已售出：121，600元
热卖中：878，400元
百万首页是什么？
购买格子
常见问题
媒体报道
买家金榜
日志 Blog
推荐给朋友
English
WIKI
派派 播客
这里是我的地盘
网络电视
我的台球网
售
口碑网
中国最大的餐饮
信息、采购平台
易餐网
成功
这是俺的地盘
SPA
中国水疗健康网
高举领秀思想伟大红旗把无产站长团结起来搞联盟
领秀联盟
WWW.LXSMS.COM

How to solve information overload?

淘宝网

所有类目

虚拟 买车险送暖冬三重豪礼！

话费充值 移动 联通 电信 不停机
游戏 DNF 九阴 新手卡 DN 天龙
彩票 双色球 3D 大乐透 足球单场
网上营业厅 合约机 3G上网 号码
点卡 魔兽 CF 传奇 QQ 网页游戏
机票 酒店 客栈 旅游 门票 国际票

服装 大码/中老年，全场7折封顶

女装 羽绒 毛衣 羊绒 棉衣 连衣裙
男装 羽绒服 毛衣 棉衣 牛仔 卫衣
内衣 文胸 睡衣 内裤 袜子 保暖
新品 呢大衣 裤子 牛仔 T恤 皮衣
外套 衬衫 休闲裤 T恤 夹克 皮衣
童装 套装 T恤 外套 裤子 童鞋

鞋包配饰 2012围巾帽子当道，温暖越冬！

女鞋 新品 雪地靴 靴子 短靴 单鞋
男鞋 休闲 潮流 板鞋 皮鞋 靴子
配饰 围巾 帽子 皮带 手套 毛线
女包 新品 真皮 大牌 欧美 钱包
男包 单肩 钱包 手提 休闲 真皮
旅行箱包 双肩 旅行箱 包 登机

运动户外 运动户外品牌汇，5折起！

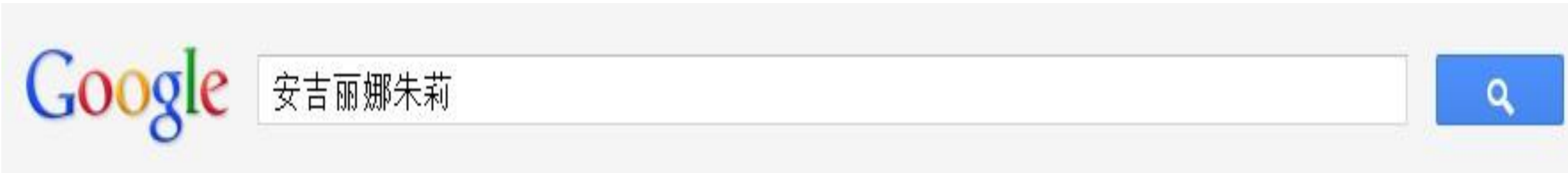
运动鞋 滑板鞋 跑鞋 板鞋 篮球鞋
装备 羽球 山地车 自行车 跑步机
耐克 骆驼 安踏 李宁 匡威 阿迪
运动服 卫衣 长裤 套装 羽绒服
户外 冲锋衣 鱼竿 雪地靴 登山鞋
背包 单肩包 军迷 帐篷 品牌直销

珠宝手表 买黄金过大年，抵抗通胀！

珠宝 钻石 黄金 铂金 施华洛
翡翠 珍珠 琥珀 珊瑚 宝石 碧玺
饰品 项链 手链 发饰 耳饰 手镯
品牌手表 卡西欧 天梭 浪琴 品牌
时装表 果冻表 水钻表 复古表
眼镜 太阳镜 眼镜架 zippo 酒具

数码 中国好路由，全场包邮特惠，爽！

手机 iPhone 安卓 三星 索尼 HTC
智能机 小米 国产 联想 华为 4核
相机 卡片机 单反 微单 佳能 尼康
笔记本 联想 苹果 戴尔 HP 华硕
平板 三星 iPadmini 谷歌 iPad4
电脑 键鼠 显示器 CPU DIY 路由
配件 手机壳 小米 耳机 手机配件
苹果配件 苹果壳 防尘塞 创意
办公 高清投影 打印机 一体机



Google
安吉丽娜朱莉

Baidu百度
新闻 网页 贴吧 知道 音乐 图片 视频 地图 文库 更多»
安吉丽娜朱莉
百度一下

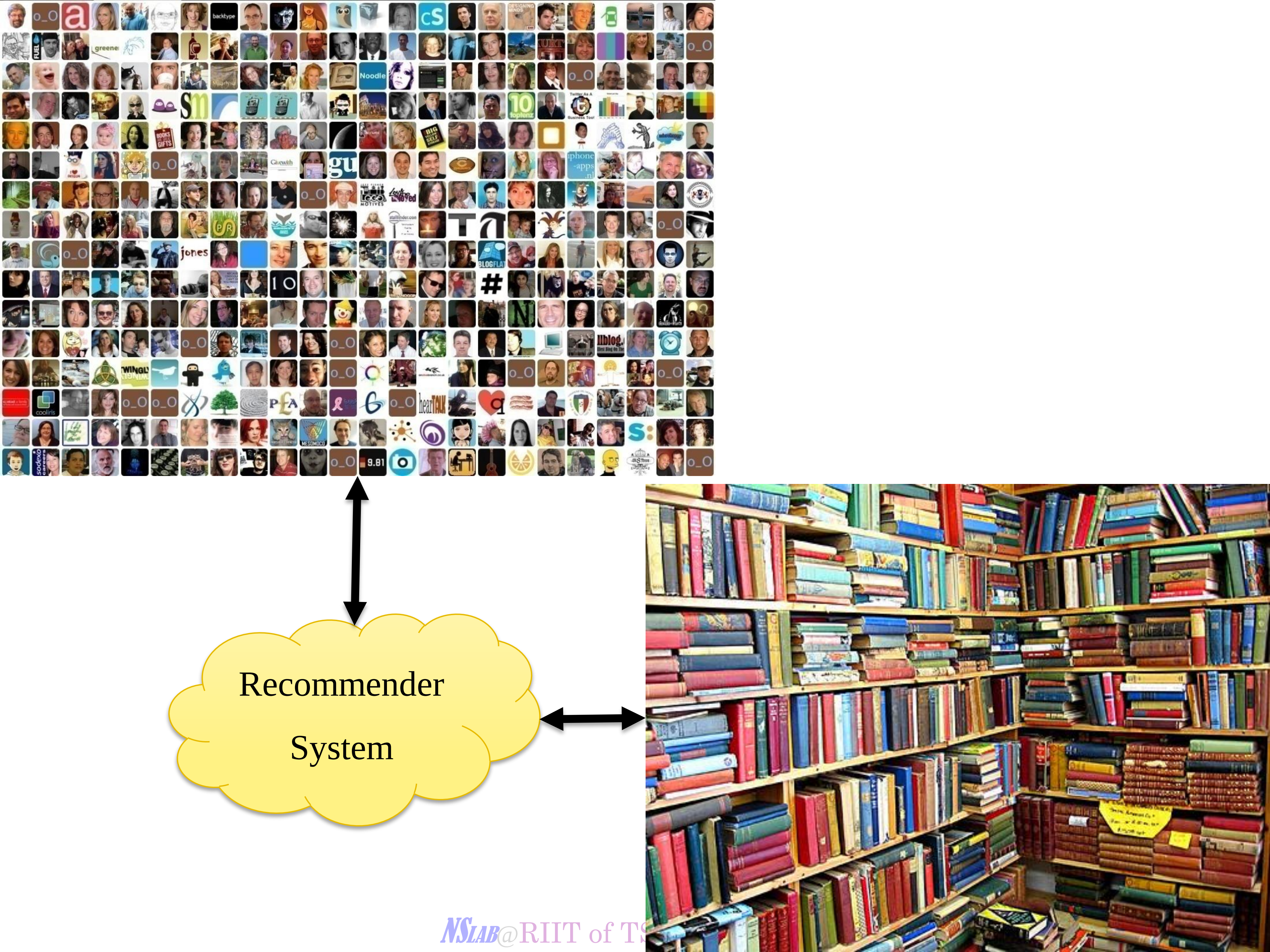
网页 图片 视频 地图 资讯 词典 在线翻译 更多
必应bing
Beta
安吉丽娜朱莉

Search Engine VS. Recommender System

- User will try search engine if
  - they have specific needs
  - they can use keywords to describe needs
- User will try recommender system if
  - they do not know what they want now
  - they can not use keywords to describe needs

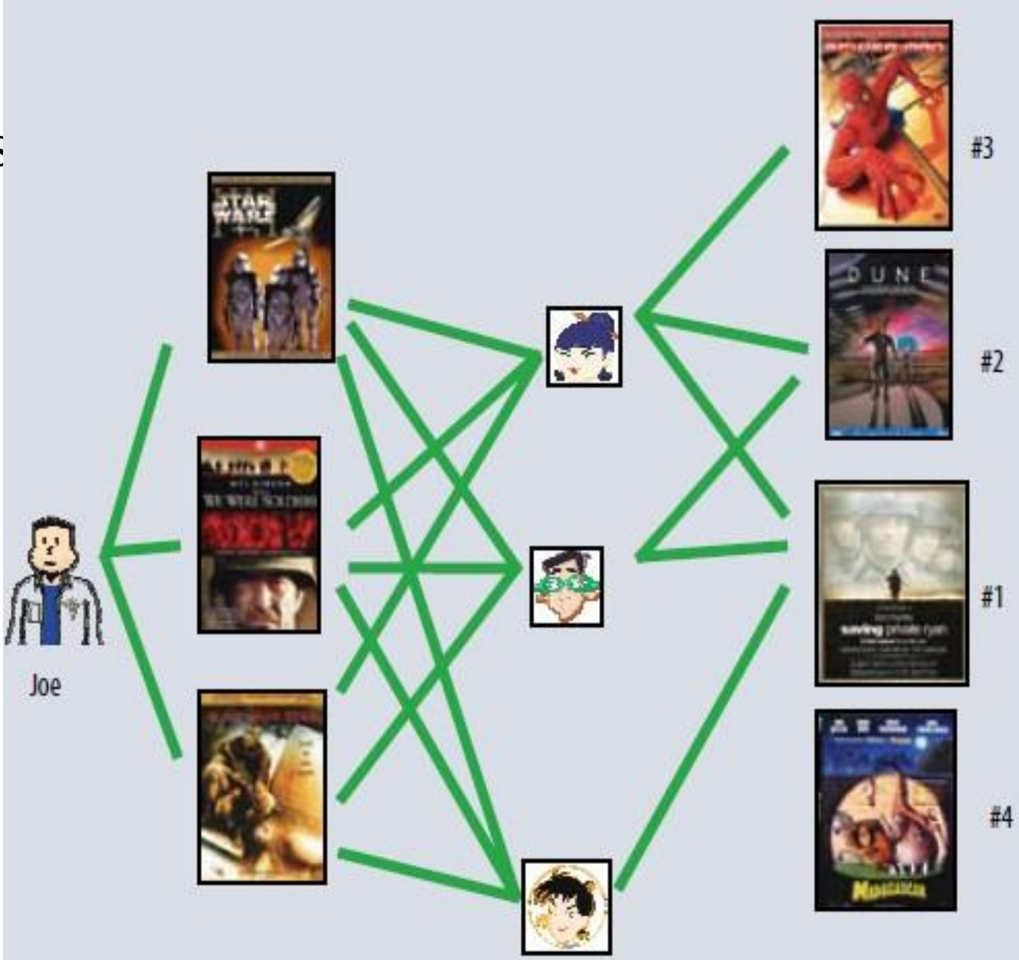
Mission

- Help user find item of their interest
- Help item provider deliver their item to right user
- Help website improve user's loyalty


Recommender
System

Background

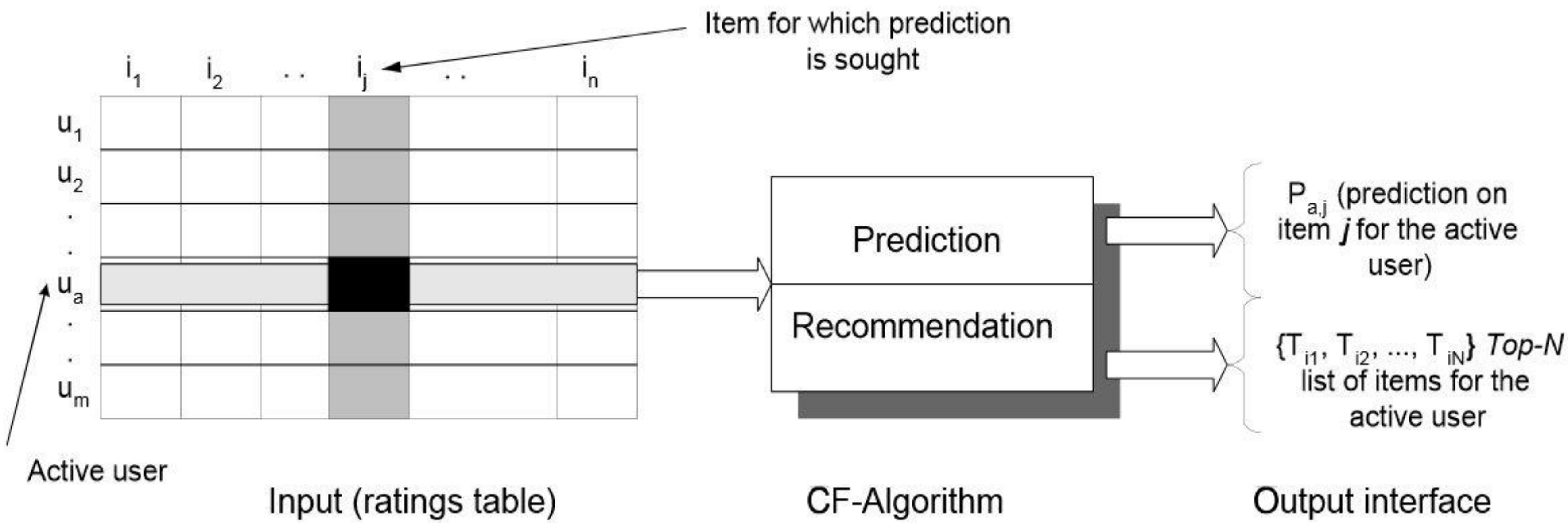
- Content filtering
  - Music Genome Project (Pandora.com)
- Collaborative filtering
  - Neighborhood methods
    - User based
    - Item based
  - Latent factor models



Collaborative Filtering

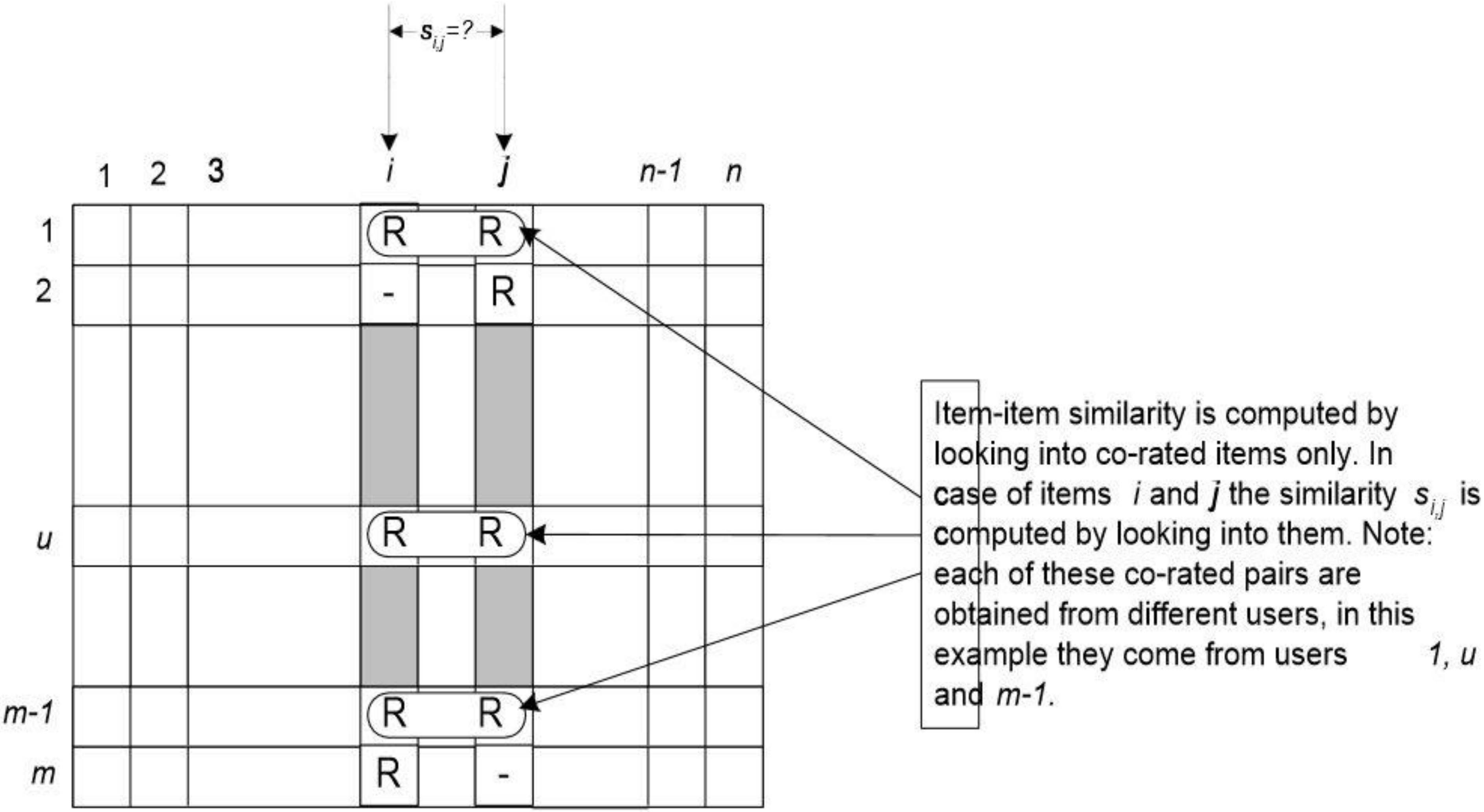
- User based
  - Users with similar history selections will share same future interest
- Item based
  - Users will like items similar to what they consumed before

The Process

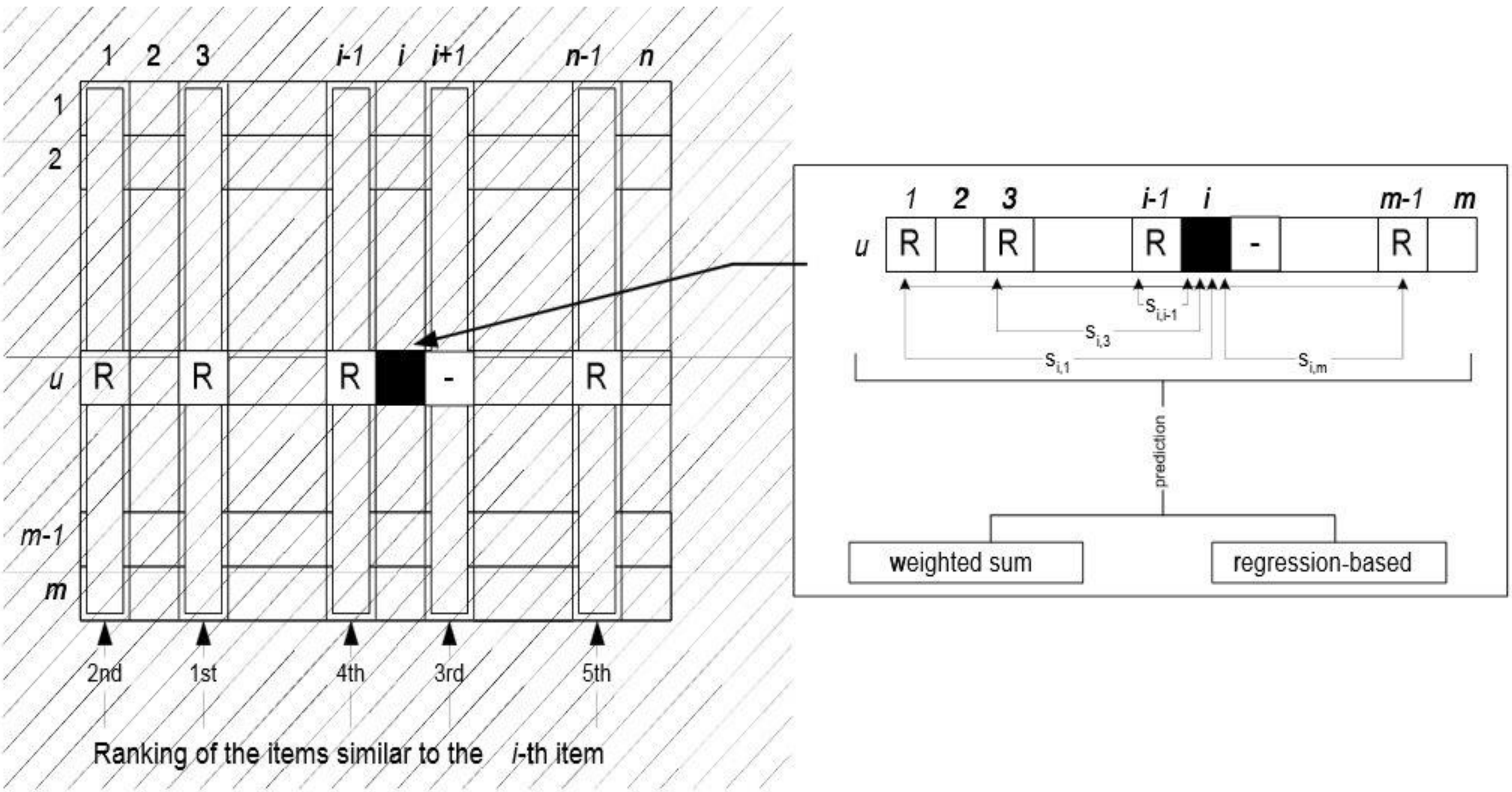


Item for which prediction is sought
i_1 i_2 . . i_j . . i_n
u_1
u_2
u_a
u_m
Active user
Input (ratings table)
Prediction
Recommendation
CF-Algorithm
$P_{a,j}$ (prediction on item *j* for the active user)
$\{T_{i1}, T_{i2}, ..., T_{iN}\}$ *Top-N* list of items for the active user
Output interface

Item based CF



Item based CF



SLIM

$$\tilde{\mathbf{r}}_u = \mathbf{r}_u \mathbf{S}$$

where $\mathbf{r}_u$ is the rating vector of u on all items and $\mathbf{S}$ is a $m \times m$ sparse matrix of aggregation coefficients.

$$\underset{\mathbf{S}}{\text{minimize}} \ \frac{1}{2}\|\mathbf{R} - \mathbf{R}\mathbf{S}\|_F^2 + \frac{\beta}{2}\|\mathbf{S}\|_F^2 + \lambda\|\mathbf{S}\|_1$$

$$\text{subject to} \ \mathbf{S} \geq 0\,, \ \ \text{diag}(\mathbf{S}) = 0,$$

NSVD

In this method, an item-item similarity was learned as a product of two low-rank matrices, $\mathbf{P}$ and $\mathbf{Q}$, where $\mathbf{P} \in \mathbb{R}^{m \times k}$, $\mathbf{Q} \in \mathbb{R}^{m \times k}$, and $k \ll m$.

$$\tilde{r}_{ui} = b_u + b_i + \sum_{j \in \mathcal{R}_u^+} \mathbf{p}_j \mathbf{q}_i^{\mathsf{T}}$$

where b_u and b_i are the user and item biases and $\mathcal{R}_u^+$ is the set of items rated by u. The parameters of this model are estimated as the minimizer to the following optimization problem:

$$\underset{\mathbf{P},\mathbf{Q}}{\text{minimize}} \ \frac{1}{2} \sum_{u \in \mathcal{C}} \sum_{i \in \mathcal{R}_u^+} \| r_{ui} - \hat{r}_{ui} \|_F^2 + \frac{\beta}{2} (\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2)$$

Motivation

- Sparse user-item rating matrix results in Item based and SLIM, which rely on learning similarities between items, fail to capture the dependencies between items that have not been co-rated by at least one user.
- Methods based on matrix factorization, alleviate this problem by projecting the data onto a low dimensional space, thereby implicitly learning better relationships between the users and items (including items which are not co-rated). However, such methods are consistently out-performed by SLIM
- NSVD does not exclude the diagonal entries while estimating the ratings during learning and prediction phases. In this case it can lead to rather trivial estimates, in which an item ends up recommending itself.

Basic FISM

Estimated value

$$\tilde{r}_{ui} = b_u + b_i + (n_u^+)^{-\alpha} \sum_{j \in \mathcal{R}_u^+} \mathbf{p}_j \mathbf{q}_i^\mathsf{T}$$

where $\mathcal{R}_u^+$ is the set of items rated by user u, $\mathbf{p}_j$ and $\mathbf{q}_i$ are the learned item latent factors, n_u^+ is the number of items rated by u, and α is a user specified parameter between 0 and 1.

FISMrmse

Loss function: RMSE

$$\mathcal{L}(\cdot) = \sum_{i \in \mathcal{D}} \sum_{u \in \mathcal{C}} (r_{ui} - \hat{r}_{ui})^2$$

Estimated value

$$\hat{r}_{ui} = b_u + b_i + (n_u^+ - 1)^{-\alpha} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{p}_j \mathbf{q}_i^\mathsf{T}$$

Regularized optimization problem

$$\underset{\mathbf{P},\mathbf{Q}}{\text{minimize}} \ \frac{1}{2} \sum_{u,i \in R} \|r_{ui} - \hat{r}_{ui}\|_F^2 + \frac{\beta}{2}(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) + \frac{\lambda}{2}\|\mathbf{b_u}\|_2^2 + \frac{\gamma}{2}\|\mathbf{b_i}\|_2^2$$

Algorithm 1 FISMrmse:Learn.

procedure FISMrmse_Learn
  $\eta \leftarrow$ learning rate
  $\beta \leftarrow \ell_F$ regularization weight
  $\rho \leftarrow$ sample factor
  $iter \leftarrow 0$
  Init $\mathbf{P}$ and $\mathbf{Q}$ with random values in (-0.001, 0.001)

  while $iter < maxIter$ or error on validation set decreases **do**
    $\mathcal{R}' \leftarrow \mathbf{R} \cup$ *SampleZeros*$(\mathbf{R}, \rho)$
    $\mathcal{R}' \leftarrow$ *RandomShuffle*$(\mathcal{R}')$

    for all $r_{ui} \in \mathcal{R}'$ **do**
      $\mathbf{x} \leftarrow (n_u^+ - 1)^{-\alpha} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{p}_j$

      $\tilde{r}_{ui} \leftarrow b_u + b_i + \mathbf{q}_i^{\mathsf{T}} \mathbf{x}$
      $e_{ui} \leftarrow r_{ui} - \tilde{r}_{ui}$
      $b_u \leftarrow b_u + \eta \cdot (e_{ui} - \lambda \cdot b_u)$
      $b_i \leftarrow b_i + \eta \cdot (e_{ui} - \gamma \cdot b_i)$
      $\mathbf{q}_i \leftarrow \mathbf{q}_i + \eta \cdot (e_{ui} \cdot \mathbf{x} - \beta \cdot \mathbf{q}_i)$

      for all $j \in \mathcal{R}_u^+ \setminus \{i\}$ **do**
        $\mathbf{p}_j \leftarrow \mathbf{p}_j + \eta \cdot (e_{ui} \cdot (n_u^+ - 1)^{-\alpha} \cdot \mathbf{q}_i - \beta \cdot \mathbf{p}_j)$
      end for
    end for
    $iter \leftarrow iter + 1$
  end while

  return $\mathbf{P}, \mathbf{Q}$
end procedure

FISMauc

Loss function: AUC

Given user's rated items in $\mathcal{R}_u^+$ and unrated items in $\mathcal{R}_u^-$

$$\mathcal{L}(\cdot) = \sum_{u \in \mathcal{C}} \sum_{i \in \mathcal{R}_u^+, j \in \mathcal{R}_u^-} \left((r_{ui} - r_{uj}) - (\hat{r}_{ui} - \hat{r}_{uj})\right)^2$$

Estimated value

$$\hat{r}_{ui} = b_u + b_i + (n_u^+ - 1)^{-\alpha} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{p}_j \mathbf{q}_i^\mathsf{T}$$

Regularized optimization problem

$$\underset{\mathbf{P},\mathbf{Q}}{\text{minimize}} \ \frac{1}{2} \sum_{u \in \mathcal{C}} \sum_{i \in \mathcal{R}_u^+, j \in \mathcal{R}_u^-} \|(r_{ui} - r_{uj}) - (\hat{r}_{ui} - \hat{r}_{uj})\|_F^2 + \frac{\beta}{2}(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2) + \frac{\gamma}{2}(\|\mathbf{b_i}\|_2^2)$$

Algorithm 2 FISMauc:Learn.

```
procedure FISMauc_Learn
    $\eta \leftarrow$ learning rate
    $\beta \leftarrow \ell_F$ regularization weight
    $\rho \leftarrow$ number of sampled zeros
    $iter \leftarrow 0$
    Init $\mathbf{P}$ and $\mathbf{Q}$ with random values in (-0.001, 0.001)
    while $iter < maxIter$ or error on validation set decreases do
        for all $u \in \mathcal{C}$ do
            for all $i \in \mathcal{R}_u^+$ do
                $\mathbf{x} \leftarrow 0$
                $\mathbf{t} \leftarrow (n_u^+ - 1)^{-\alpha} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{p}_j$
                $\mathcal{Z} \leftarrow SampleZeros(\rho)$
                for all $j \in \mathcal{Z}$ do
                    $\tilde{r}_{ui} \leftarrow b_i + \mathbf{t} \cdot \mathbf{q}_i^\mathsf{T}$
                    $\tilde{r}_{uj} \leftarrow b_j + \mathbf{t} \cdot \mathbf{q}_j^\mathsf{T}$
                    $r_{uj} \leftarrow 0$
                    $e \leftarrow (r_{ui} - r_{uj}) - (\tilde{r}_{ui} - \tilde{r}_{uj})$
                    $b_i \leftarrow b_i + \eta \cdot (e - \gamma \cdot b_i)$
                    $b_j \leftarrow b_j - \eta \cdot (e - \gamma \cdot b_j)$
                    $\mathbf{q}_i \leftarrow \mathbf{q}_i + \eta \cdot (e \cdot \mathbf{t} - \beta \cdot \mathbf{q}_i)$
                    $\mathbf{q}_j \leftarrow \mathbf{q}_j - \eta \cdot (e \cdot \mathbf{t} - \beta \cdot \mathbf{q}_j)$
                    $\mathbf{x} \leftarrow \mathbf{x} + e \cdot (\mathbf{q}_i - \mathbf{q}_j)$
                end for
            end for

            for all $j \in \mathcal{R}_u^+ \setminus \{i\}$ do
                $\mathbf{p}_j \leftarrow \mathbf{p}_j + \eta \cdot (\frac{1}{\rho} \cdot (n_u^+ - 1)^{-\alpha} \cdot \mathbf{x} - \beta \cdot \mathbf{p}_j)$
            end for
        end for

        $iter \leftarrow iter + 1$
    end while

    return $\mathbf{P}, \mathbf{Q}$
end procedure
```

Evaluation

- Data set
  - ML100K: ML100K-1, ML100K-2, ML100K-3
  - Netflix: Netflix-1, Netflix-2, Netflix-3
  - Yahoo Music: Yahoo-1, Yahoo-2, Yahoo-3
  - Only ML100K-3, Netflix-3, Yahoo-2 are used

Table 1: Datasets.

Dataset	#Users	#Items	#Ratings	Rsize	Csize	Density
ML100K-1	943	1,178	59,763	63.99	50.73	5.43%
ML100K-2	943	1,178	39,763	42.57	33.75	3.61%
ML100K-3	943	1,178	19,763	21.16	16.78	1.80%
Netflix-1	6,079	5,641	429,339	70.63	76.11	1.25%
Netflix-2	6,079	5,641	221,304	36.40	39.23	0.65%
Netflix-3	6,079	5,641	110,000	18.10	19.50	0.32%
Yahoo-1	7,558	3,951	282,075	37.32	71.39	0.94%
Yahoo-2	7,558	3,951	149,050	19.72	37.72	0.50%
Yahoo-3	7,558	3,951	75,000	9.92	18.98	0.25%

Evaluation

- Methodology
  - 5-fold Leave-One-Out-Cross-Validation (LOOCV)
- Metrics
  - HR (Hit Rate)

$$HR = \frac{\#hits}{\#users}$$

  - ARHR

$$ARHR = \frac{1}{\#users} \sum_{i=1}^{\#hits} \frac{1}{pos_i}$$

Evaluation

- Comparison Algorithms
  - ItemKNN(cos)
  - ItemKNN(cprob)
  - ItemKNN(log)
  - PursSVD
  - BPRkNN
  - BPRMF
  - SLIM
  - Basic FISM
  - FISMrmse
  - FISMauc

Experimental Results

$$\hat{r}_{ui} = b_u + b_i + (n_u^+ - 1)^{-\alpha} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{p}_j \mathbf{q}_i^\mathsf{T}$$

Table 2: Performance of different bias schemes.

Scheme	**ML100K**				**Yahoo**			
	Beta	Lambda	Gamma	HR	Beta	Lambda	Gamma	HR
NoBias	8e-4	-	-	0.1281	2e-5	-	-	0.0974
UserBias	6e-4	0.1	-	0.1336	4e-5	0.1	-	0.1012
ItemBias	2e-4	-	0.01	0.1401	4e-5	-	1e-4	0.1007
User&ItemBias	6e-4	0.1	1e-4	0.1090	4e-5	0.1	1e-4	0.0977

Experimental Results

$$\hat{r}_{ui} = b_u + b_i + (n_u^+ - 1)^{-\alpha} \sum_{j \in \mathcal{R}_u^+ \setminus \{i\}} \mathbf{p}_j \mathbf{q}_i^\mathsf{T}$$

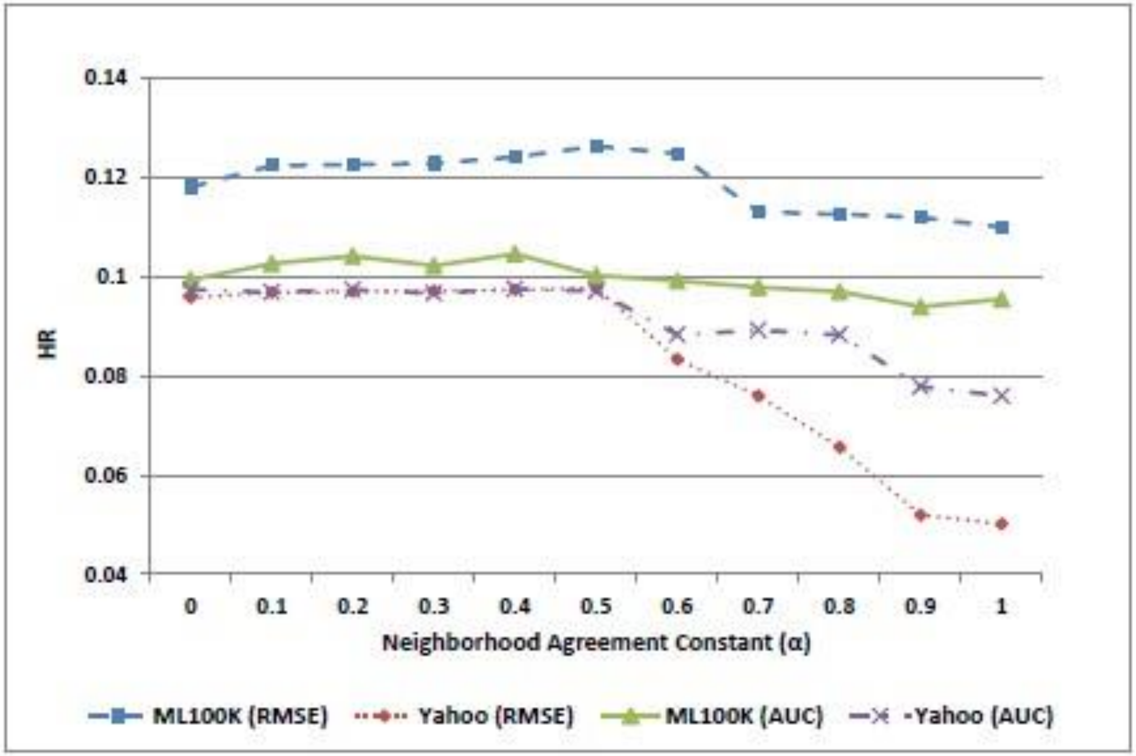


Figure 1: Effect of neighborhood agreement on performance.

Experimental Results

$$S = PQ^T$$

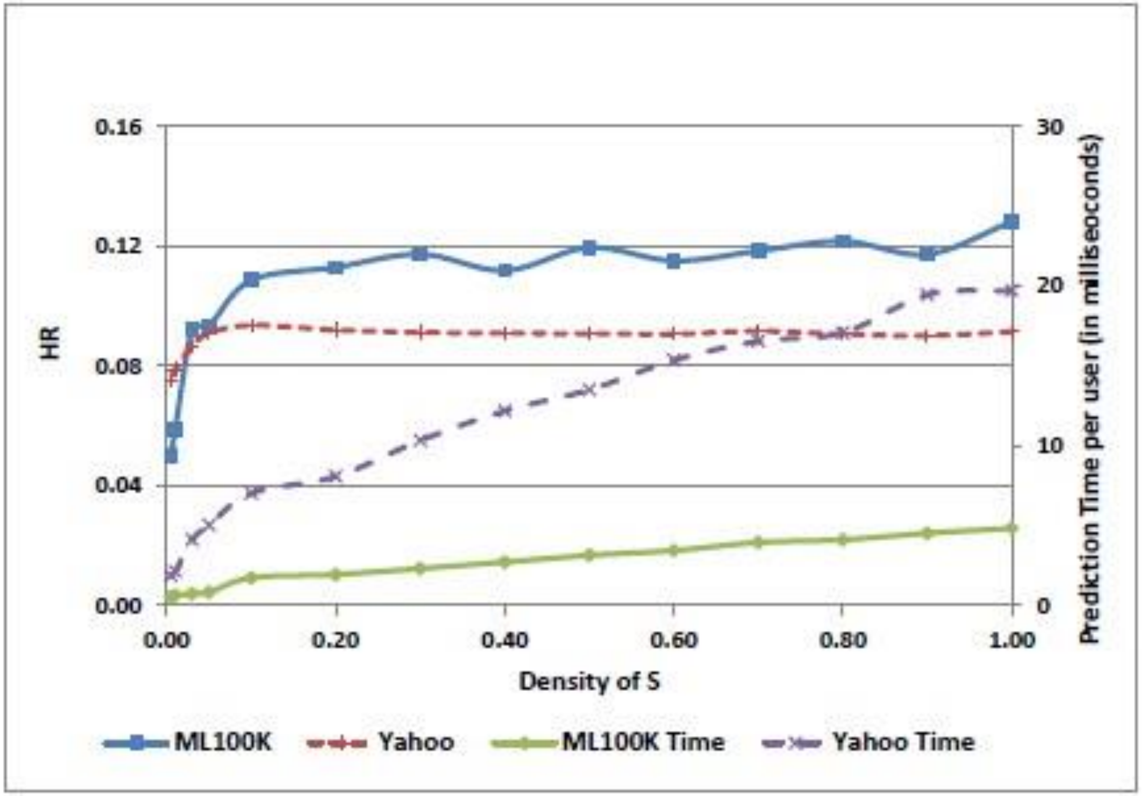
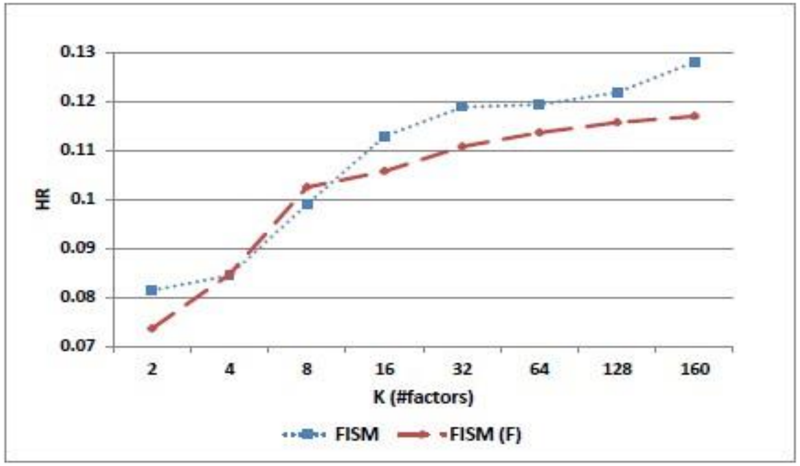
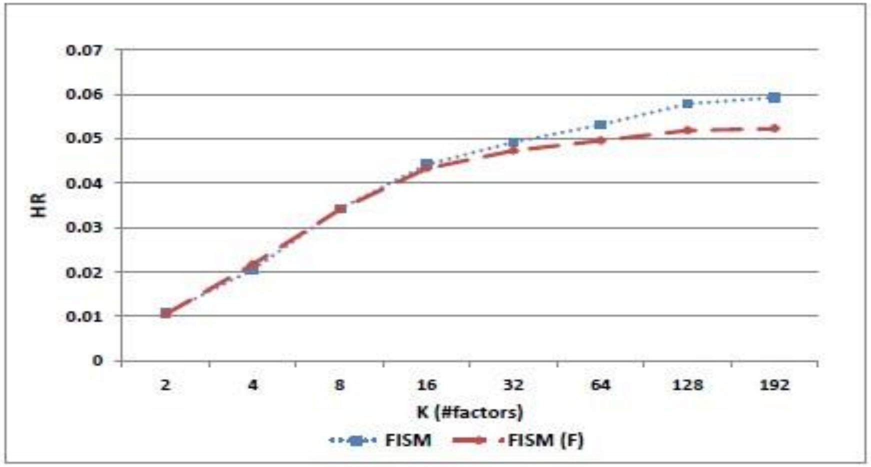


Figure 2: Performance of induced Sparsity on S.

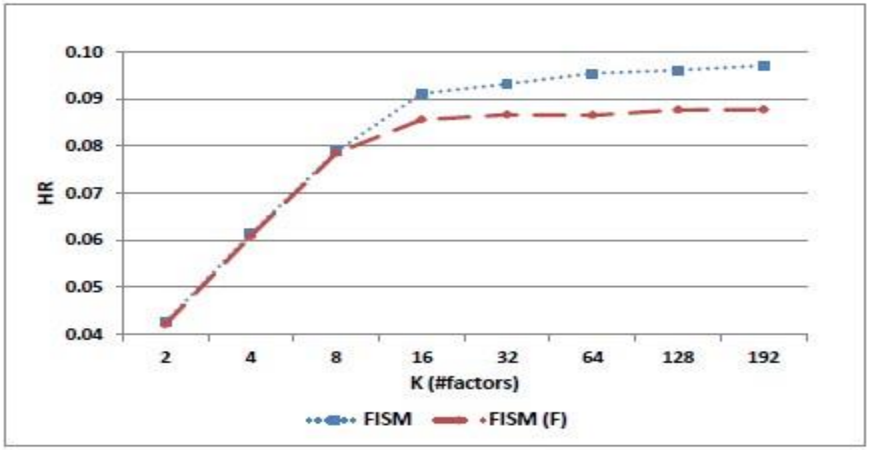
Experimental Results



(a) ML100K dataset.



(b) Netflix dataset.



(c) Yahoo dataset.

Figure 3: Effect of estimation approach on performance.

Experimental Results

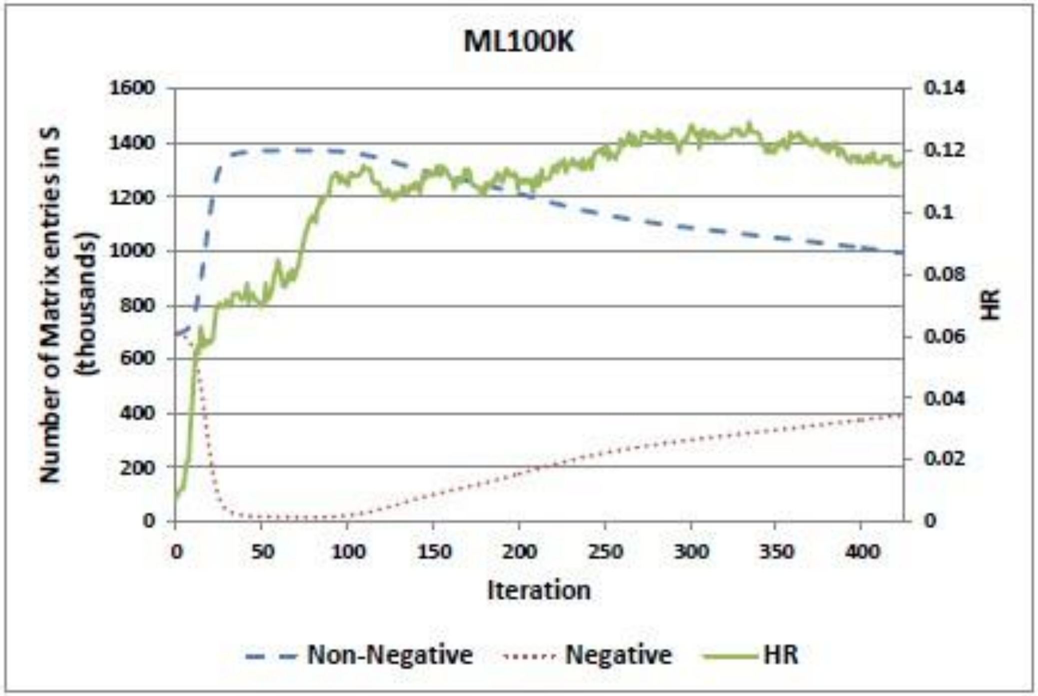
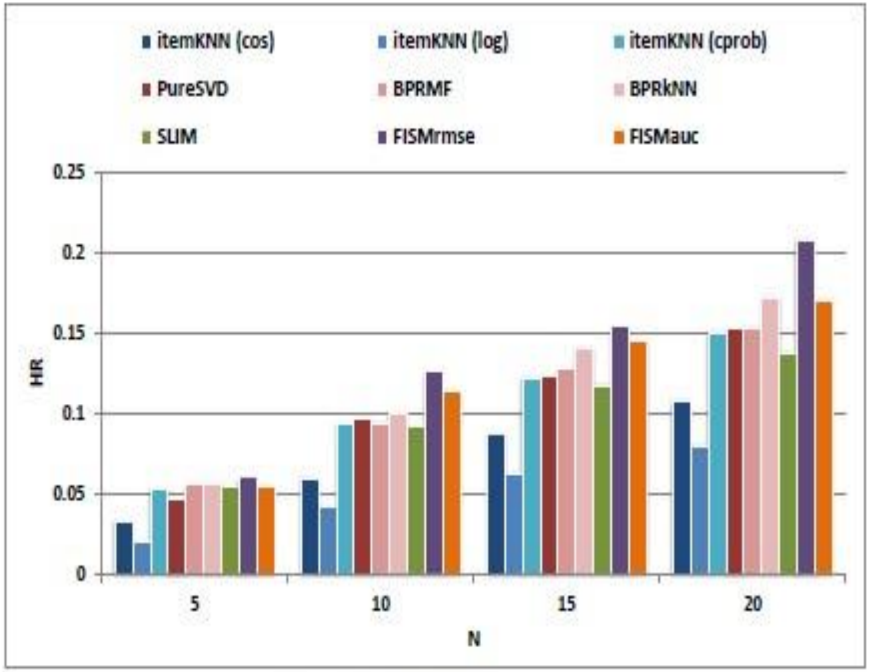
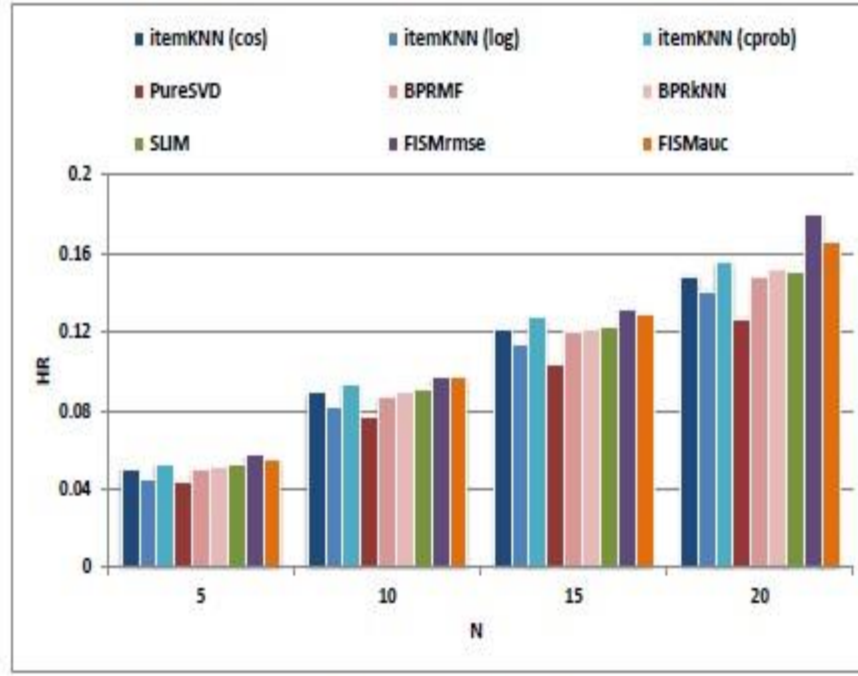


Figure 4: Non-negative and negative entries in S.

Experimental Results



(a) ML100K dataset.



(b) Yahoo dataset.

Figure 5: Performance for different values of N.

Experimental Results

Table 3: Comparison of performance of *top-N* recommendation algorithms with FISM.

Method	ML100K-1					ML100K-2					ML100K-3				
	Params			HR	ARHR	Params			HR	ARHR	Params			HR	ARHR
ItemKNN (cos)	100	-	-	0.1604	0.0578	100	-	-	0.1214	0.0393	100	-	-	0.0602	0.0193
ItemKNN (log)	100	-	-	0.1047	0.0336	100	-	-	0.0809	0.0250	100	-	-	0.0424	0.0116
ItemKNN (cprob)	500	0.6	-	0.1711	0.0581	500	0.3	-	0.1308	0.0440	400	0.1	-	0.0938	0.0293
PureSVD	10	-	-	0.1700	0.0594	10	-	-	0.1362	0.0438	5	-	-	0.0438	0.0316
BPRkNN	1e-4	0.01	-	0.1621	0.0564	1e-5	0.01	-	0.1272	0.0447	1e-5	14	-	0.1006	0.0319
BPRMF	400	0.1	-	0.1610	0.0512	700	0.1	-	0.1224	0.0407	700	0.25	-	0.0943	0.0305
SLIM	0.1	20	-	0.1782	0.0620	0.01	18	-	0.1283	0.0448	1e-4	14	-	0.0919	0.0303
FISMrmse	96	2e-5	0.001	0.1908	0.0641	64	8e-4	0.01	0.1482	0.0462	96	8e-4	0.001	0.1260	0.0384
FISMauc	64	0.001	1e-4	0.1518	0.0504	144	2e-5	5e-5	0.1304	0.0424	144	8e-5	1e-5	0.1140	0.0340

Method	Netflix-1					Netflix-2					Netflix-3				
	Params			HR	ARHR	Params			HR	ARHR	Params			HR	ARHR
ItemKNN (cos)	100	-	-	0.1516	0.0689	100	-	-	0.0849	0.0316	100	-	-	0.0374	0.0123
ItemKNN (log)	100	-	-	0.0630	0.0240	100	-	-	0.0838	0.0303	100	-	-	0.0188	0.0062
ItemKNN (cprob)	20	0.5	-	0.1555	0.0678	500	0.5	-	0.0879	0.0326	200	0.1	-	0.0461	0.0162
PureSVD	600	-	-	0.1783	0.0865	400	-	-	0.0807	0.0297	400	-	-	0.0382	0.0131
BPRkNN	1e-3	1e-4	-	0.1678	0.0781	1e-4	1	-	0.0889	0.0329	0.01	1e-3	-	0.0439	0.0148
BPRMF	800	0.1	-	0.1638	0.0719	700	0.1	-	0.0862	0.0318	5	0.01	-	0.0454	0.0153
SLIM	1e-3	8	-	0.2025	0.1008	0.1	8	-	0.0947	0.0374	1e-4	12	-	0.0422	0.0149
FISMrmse	192	2e-5	0.001	0.2118	0.1107	192	6e-5	0.001	0.1041	0.0386	128	6e-5	0.001	0.0578	0.0185
FISMauc	192	1e-5	1e-4	0.2095	0.1016	240	2e-5	1e-4	0.0979	0.0341	160	4e-4	5e-4	0.0548	0.0177

Method	Yahoo-1					Yahoo-2					Yahoo-3				
	Params			HR	ARHR	Params			HR	ARHR	Params			HR	ARHR
ItemKNN (cos)	100	-	-	0.1344	0.0502	100	-	-	0.0890	0.0295	100	-	-	0.0366	0.0116
ItemKNN (log)	100	-	-	0.1046	0.0358	100	-	-	0.0820	0.0261	100	-	-	0.0489	0.0153
ItemKNN (cprob)	500	0.6	-	0.1387	0.0510	200	0.4	-	0.0908	0.0313	20	0.1	-	0.0571	0.0187
PureSVD	50	-	-	0.1229	0.0459	20	-	-	0.0769	0.0257	20	-	-	0.0494	0.0154
BPRkNN	1e-3	1e-4	-	0.1432	0.0528	1e-3	1e-4	-	0.0894	0.0304	0.1	0.01	-	0.0549	0.0183
BPRMF	700	0.1	-	0.1337	0.0473	700	0.1	-	0.0869	0.0288	10	0.01	-	0.0530	0.0169
SLIM	0.1	12	-	0.1454	0.0542	1e-3	12	-	0.0904	0.0304	0.1	2	-	0.0491	0.0159
FISMrmse	192	1e-4	0.001	0.1522	0.0542	192	2e-5	5e-4	0.0971	0.0371	160	0.002	0.001	0.0740	0.0230
FISMauc	144	8e-5	1e-4	0.1426	0.0488	160	2e-5	5e-4	0.0974	0.0315	176	2e-4	0.001	0.0722	0.0228

Experimental Results

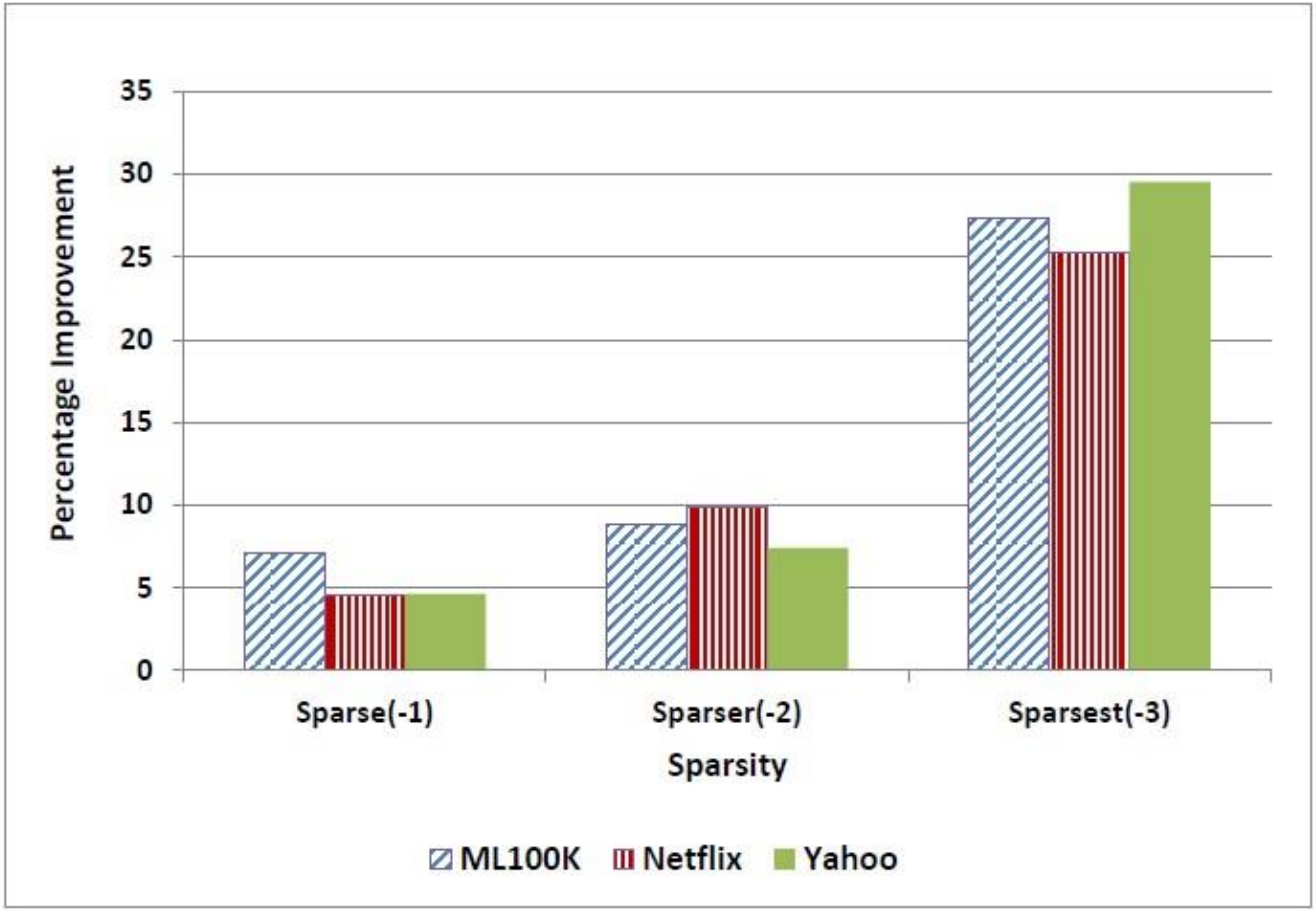


Figure 6: Effect of sparsity on performance for various datasets.

Conclusion

- This paper proposed a factored item similarity based method (FISM), which learns the item similarities as the product of two matrices.
- FISM can well cope with data sparsity problem, and better estimators are achieved as the number of factors increases.
- FISM outperforms other state-of-the-art top-N recommendation algorithms

Discussion

- Why not use RMSE as the metric?
- NSVD doesn't use unrated entries for learning, while FISM does. The comparison seems bias?
- Is there a symmetric method to factor user similarity with two low dimensional matrices?

Questions?

Thank you!