

# 推荐系统

## 今日头条

推荐

热点

视频

图片

段子

社会

娱乐

科技

体育



乌克兰首都发生汽车炸弹爆炸 国防部情报局局长丧生



大驱来啦！我国新型万吨级驱逐舰首舰刚刚下水

军事



海外网 · 660评论 · 刚刚

要闻

社会

娱乐

体育

军事

明星



# 推荐系统

- ✎ 19444人在进行视频或语音聊天
- ✎ 62.5万部优酷土豆视频被观看
- ✎ Facebook共产生701,389账号登陆
- ✎ App Store上已有51,000个app被下载
- ✎ . . .





# 推荐系统

豆瓣电影

电影、影人、影院、电视剧

影讯&购票 选电影 电视剧 排行榜 分类 影评 2016年度榜单 2016观影报告



## 你可能喜欢的电影

电影主页 影评 问答 在看 想看 看过 豆列 豆瓣主页

看过 ... (993部)



## 排行榜

专属你的购物指南

啤酒

手机通讯

书包

双肩包

炒锅



【京东超市】雪花啤酒 (Snowbee)



【京东超市】麒麟 (Kirin) 一番榨



【京东超市】德国原装进口啤酒 奥丁



【京东超市】青岛啤酒 (Tsingtao)



【京东超市】德国进口 (Eichbau)



【京东超市】朝日啤酒 (清爽生) 50

# 推荐系统

- 推荐之王系统
- 35%的销售额

亚马逊



- 订单贡献率  
10%

京东

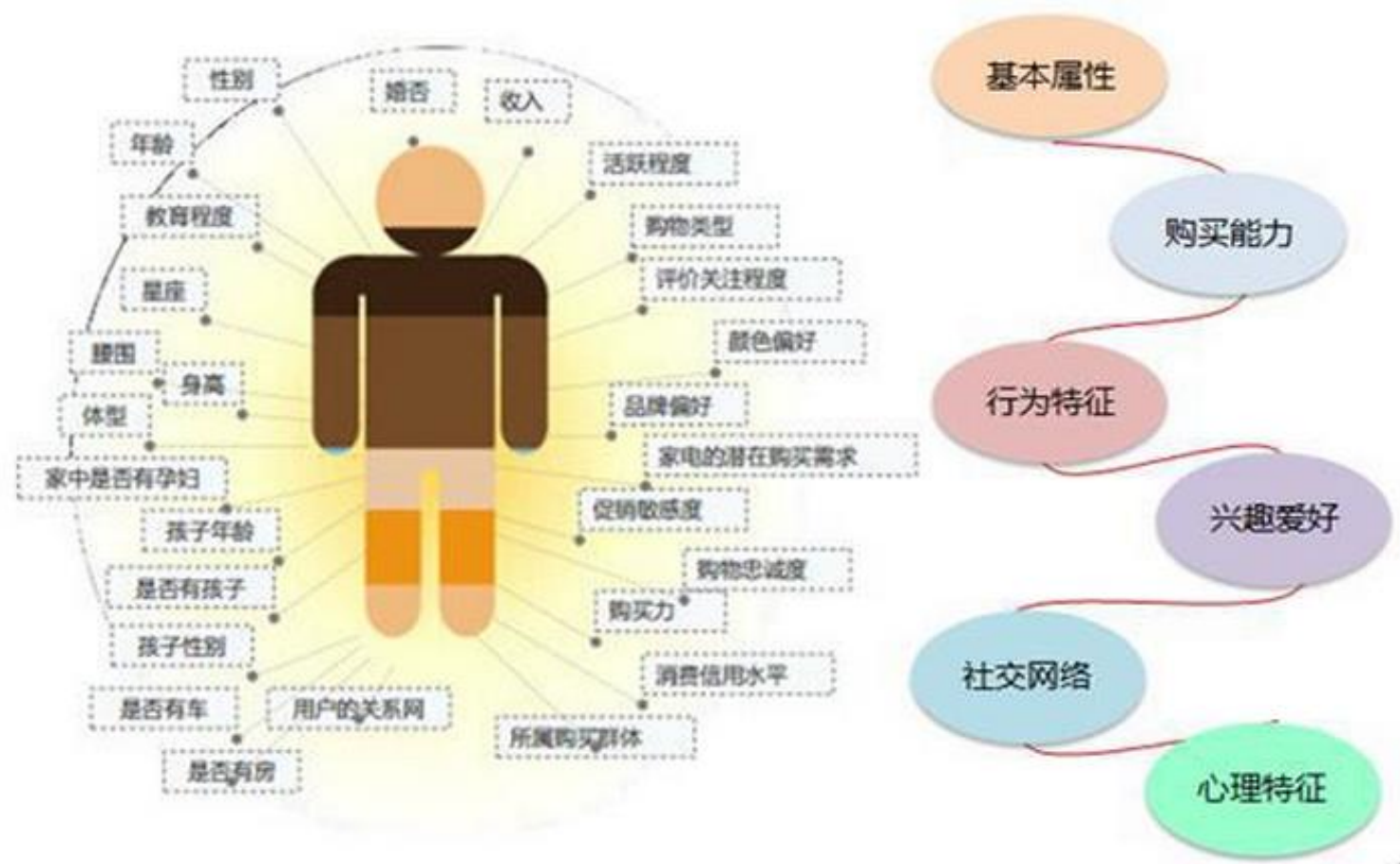


- 基于数据挖掘的推荐引擎产品
- 5.5亿装机
- 月活1.3亿日活6000万,
- 每日使用时长76分钟

头条



# 推荐系统



# 协同过滤

✎ 如果你现在想看个电影，但你不知道具体看哪部，你会怎么做？



☆ 如何确定一个用户是不是和你有相似的品位？

☆ 如何将邻居们的喜好组织成一个排序的目录？


# 协同过滤

---

✓ 要实现协同过滤，需要的步骤？

 1.收集用户偏好

 2.找到相似的用户或物品

 3.计算推荐



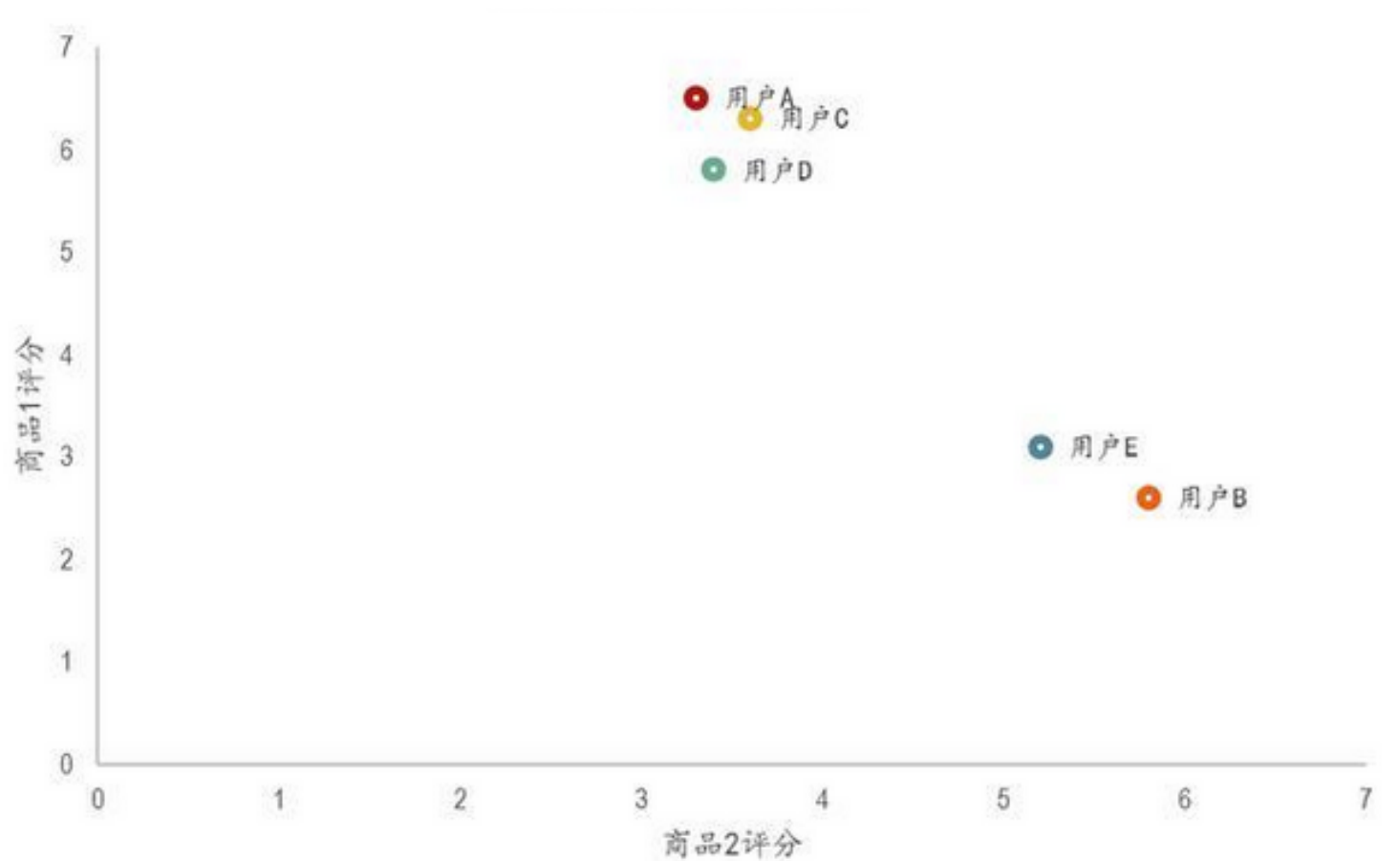
# 协同过滤

用户行为	类型	特征	作用
评分	显式	整数量化的偏好，可能的取值是 $[0, n]$ ； $n$ 一般取值为 5 或者是 10	通过用户对物品的评分，可以精确的得到用户的偏好
投票	显式	布尔量化的偏好，取值是 0 或 1	通过用户对物品的投票，可以较精确的得到用户的偏好
转发	显式	布尔量化的偏好，取值是 0 或 1	通过用户对物品的投票，可以精确的得到用户的偏好。 如果是站内，同时可以推理得到被转发人的偏好（不精确）
保存书签	显示	布尔量化的偏好，取值是 0 或 1	通过用户对物品的投票，可以精确的得到用户的偏好。
标记标签 (Tag)	显示	一些单词，需要对单词进行分析，得到偏好	通过分析用户的标签，可以得到用户对项目的理解，同时可以分析出用户的情感：喜欢还是讨厌
评论	显示	一段文字，需要进行文本分析，得到偏好	通过分析用户的评论，可以得到用户的情感：喜欢还是讨厌



# 相似度计算

	商品1	商品2
用户A	3.3	6.5
用户B	5.8	2.6
用户C	3.6	6.3
用户D	3.4	5.8
用户E	5.2	3.1



# 相似度计算

✓ 欧几里德距离 ( Euclidean Distance )

$$d(x, y) = \sqrt{(\sum (x_i - y_i)^2)} \quad sim(x, y) = \frac{1}{1 + d(x, y)}$$

✓ 皮尔逊相关系数 ( Pearson Correlation Coefficient )

$$p(x, y) = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{(n-1) s_x s_y} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}$$

✓ Cosine 相似度 ( Cosine Similarity )

$$T(x, y) = \frac{x \bullet y}{\|x\|^2 \times \|y\|^2} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \sqrt{\sum y_i^2}}$$

# 相似度计算

✓ 皮尔逊相关系数 ( Pearson Correlation Coefficient )

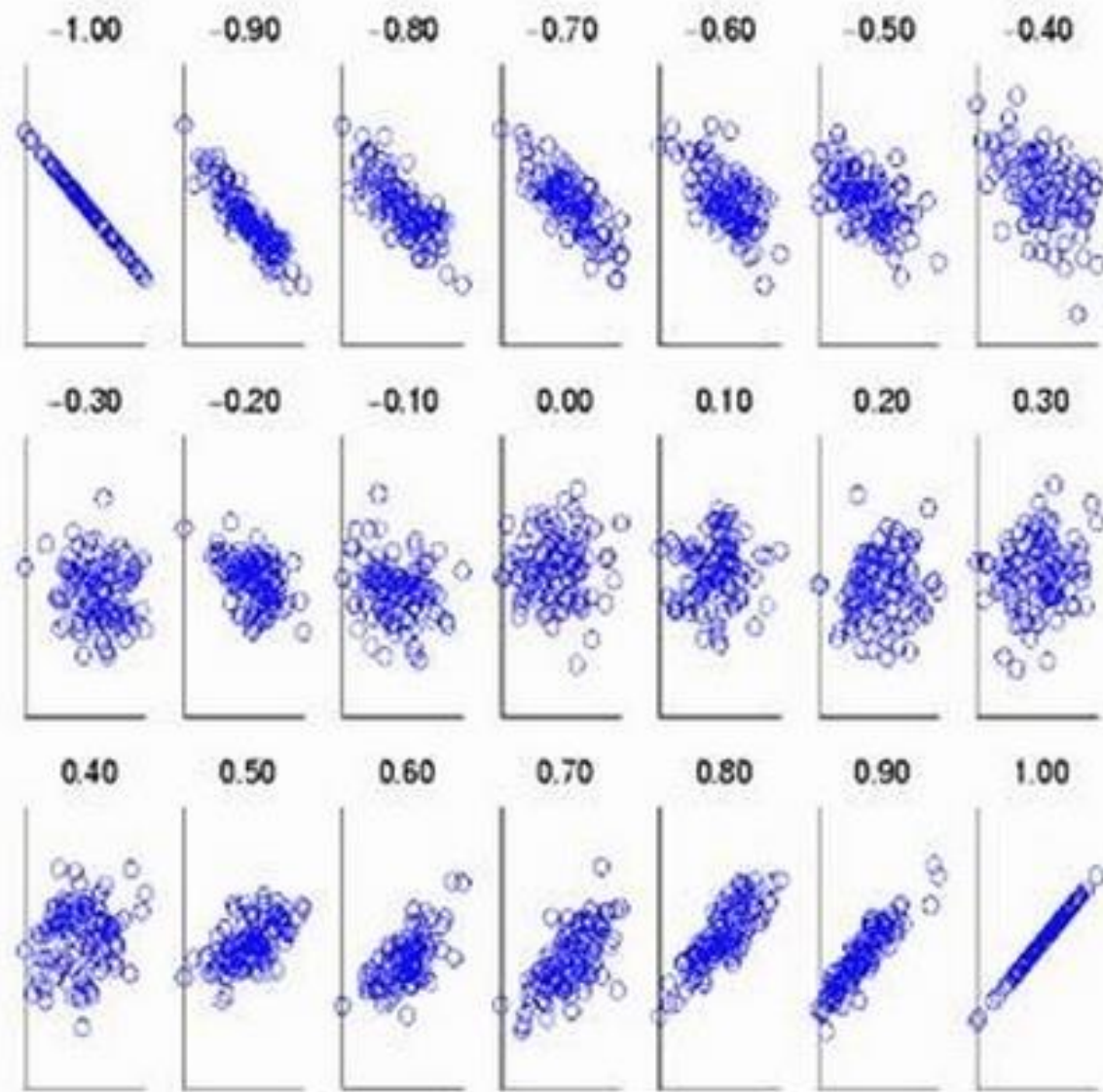
✎ 协方差 
$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1}$$

✎ 皮尔逊相关系数 
$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y};$$

✎ Pearson相关系数是用协方差除以两个变量的标准差得到的

# 相似度计算

✓ 皮尔逊相关系数



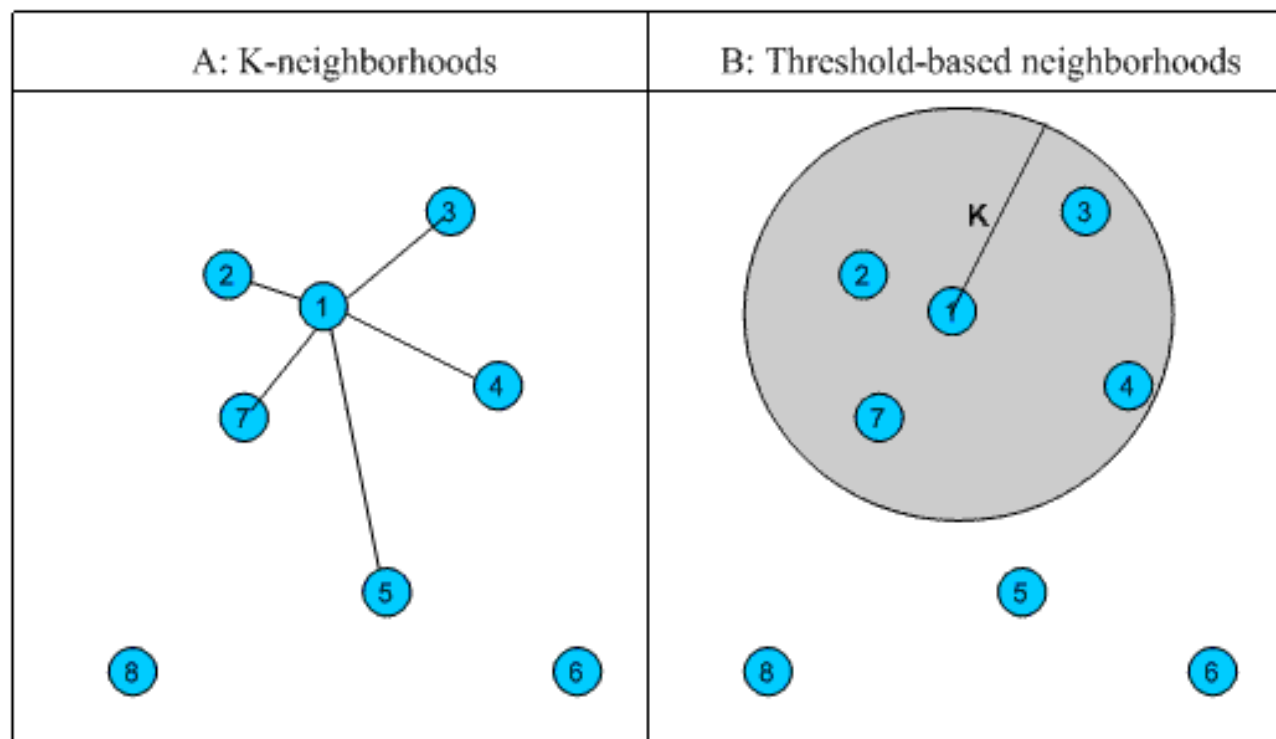


# 相似度计算

## ✓ 邻居的选择

✎ A. 固定数量的邻居

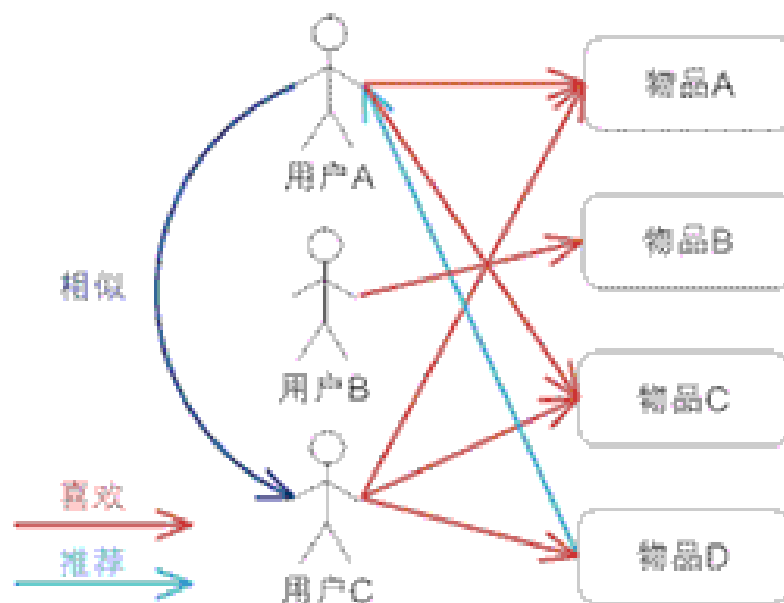
✎ B. 基于相似度门槛的邻居



# 协同过滤

## ✓ 基于用户的协同过滤

用户/物品	物品A	物品B	物品C	物品D
用户A	√		√	推荐
用户B		√		
用户C	√		√	√



# 协同过滤

✓ 基于用户的协同过滤要解决的问题

✎ 已知用户评分矩阵Matrix R ( 一般都是非常稀疏的 )

✎ 推断矩阵中空格empty cells处的值

1	0	3	0	5
0	0	0	0	0
0	0	0	0	0
2	0	4	0	6
0	0	0	0	0
0	0	0	0	0

# 协同过滤

---

## ✓ UserCF存在的问题issues

✎ 对于一个新用户，很难找到邻居用户。

✎ 对于一个物品，所有最近的邻居都在其上没有多少打分。



# 协同过滤

## ✓ 基础解决方案

✎ 相似度计算最好使用皮尔逊相似度

✎ 考虑共同打分物品的数目，如乘上 $\min(n, N)/N$      $n$ :共同打分数  $N$ :指定阈值

✎ 对打分进行归一化处理

✎ 设置一个相似度阈值

# 相似度计算

---

✓ 基于用户的协同过滤为啥不流行？

✎ 1.稀疏问题

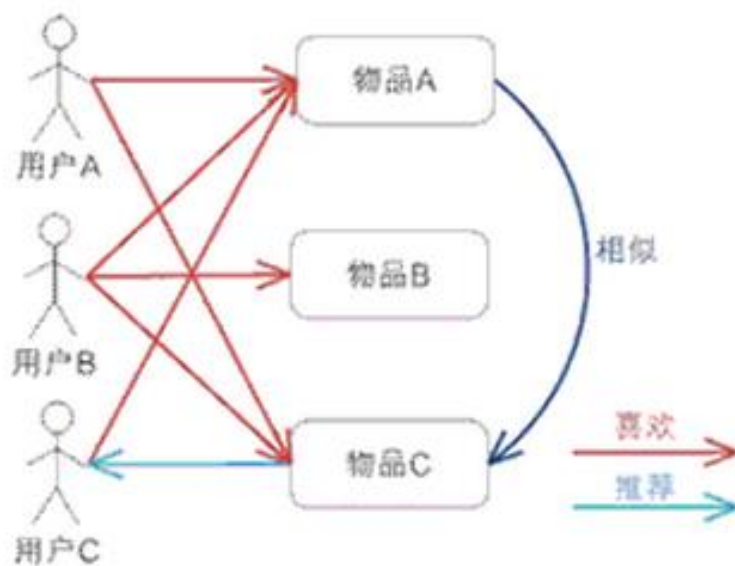
✎ 2.数百万的用户计算，这量？

✎ 3.人是善变的

# 协同过滤

## ✓ 基于物品的协同过滤

用户/物品	物品A	物品B	物品C
用户A	√		√
用户B	√	√	√
用户C	√		推荐



# 协同过滤

---

✓ 基于物品的协同过滤优势！

✎ 计算性能高，通常用户数量远大于物品数量

✎ 可预先计算保留，物品并不善变



# 协同过滤

	users											
	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3			5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- unknown rating
  - rating between 1 to 5

	users												sim(1,m)
	1	2	3	4	5	6	7	8	9	10	11	12	
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		<u>0.41</u>
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		<u>0.59</u>

$$r_{51} = (0.41 \cdot 2 + 0.59 \cdot 3) / (0.41 + 0.59) = 2.6$$

# 协同过滤

---

✓ 用户冷启动问题

✎ 引导用户把自己的一些属性表达出来

✎ 利用现有的开放数据平台

✎ 根据用户注册属性

✎ 推荐排行榜单

# 协同过滤

---

✓ 物品冷启动问题

✎ 文本分析

✎ 主题模型

✎ 打标签

✎ 推荐排行榜单

# 协同过滤

	UserCF	ItemCF
性能	适用于用户较少的场合，如果用户过多，计算用户相似度矩阵的代价交大	适用于物品数明显小于用户数的场合，如果物品很多，计算物品相似度矩阵的代价交大
领域	实效性要求高，用户个性化兴趣要求不高	长尾物品丰富，用户个性化需求强烈
实时性	用户有新行为，不一定需要推荐结果立即变化	用户有新行为，一定会导致推荐结果的实时变化
冷启动	在新用户对少的物品产生行为后，不能立即对他进行个性化推荐，因为用户相似度是离线计算的； 新物品上线后一段时间，一旦有用户对物品产生行为，就可以将新物品推荐给其他用户	新用户只要对一个物品产生行为，就能推荐相关物品给他，但无法在不离线更新物品相似度表的情况下将新物品推荐给用户 (但是新的item到来也同样是冷启动问题)
推荐理由	很难提供令用户信服的推荐解释	可以根据用户历史行为归纳推荐理由



# 协同过滤

---

✓ 基于用户的推荐

 实时新闻

 突然情况

✓ 基于物品的推荐

 图书

 电子商务

 电影

 . . .

# 隐语义模型

---

## ✓ 隐语义模型

✎ 从数据出发，进行个性化推荐

✎ 用户和物品之间有着隐含的联系

✎ 隐含因子让计算机能理解就好

✎ 将用户和物品通过中介隐含因子联系起来

# 隐语义模型




✓ 隐语义模型

📎 分解




Rating Matrix (N x M)

			
	5	3	5
	4	2	1
	0	3	3

User Feature Matrix (F x N)

			
$f_1$	1	-4	1
$f_2$	-2	0	-3
$f_3$	0	-5	1

Movie Feature Matrix (F x M)




			
$f_1$	-1	0	-2
$f_2$	4	-4	1
$f_3$	0	2	2

📎 组合







User Feature Matrix (F x N)

			
$f_1$	1	-4	1
$f_2$	-2	0	-3
$f_3$	0	-5	1

Movie Feature Matrix (F x M)


			
$f_1$	-1	0	-2
$f_2$	4	-4	1
$f_3$	0	2	2


Rating Matrix (N x M)

			
	5	3	5
	4	2	1
	0	3	3

# 隐语义模型

## ✓ 隐语义模型

  $R_{UI} = P_U Q_I = \sum_{k=1}^K P_{U,k} Q_{k,I}$

  $C = \sum_{(U,I) \in K} (R_{UI} - \hat{R}_{UI})^2 = \sum_{(U,I) \in K} (R_{UI} - \sum_{k=1}^K P_{U,k} Q_{k,I})^2 + \lambda \|P_U\|^2 + \lambda \|Q_I\|^2$

	item 1	item 2	item 3	item 4			class 1	class 2	class 3			item 1	item 2	item 3	item 4
user 1	R11	R12	R13	R14	=	user 1	P11	P12	P13	X	class 1	Q11	Q12	Q13	Q14
user 2	R21	R22	R23	R24		user 2	P21	P22	P23		class 2	Q21	Q22	Q23	Q24
user 3	R31	R32	R33	R34		user 3	P31	P32	P33		class 3	Q31	Q32	Q33	Q34
<b>R</b>						<b>P</b>					<b>Q</b>				

# 隐语义模型

## ✓ 隐语义模型求解

✎ 梯度下降方向：

$$\frac{\partial C}{\partial P_{Uk}} = -2(R_{UI} - \sum_{k=1}^K P_{U,k} Q_{k,I}) Q_{kI} + 2\lambda P_{Uk}$$

$$\frac{\partial C}{\partial Q_{kI}} = -2(R_{UI} - \sum_{k=1}^K P_{U,k} Q_{k,I}) P_{Uk} + 2\lambda Q_{kI}$$

✎ 迭代求解：

$$P_{Uk} = P_{Uk} + \alpha((R_{UI} - \sum_{k=1}^K P_{U,k} Q_{k,I}) Q_{kI} - \lambda P_{Uk})$$

$$Q_{kI} = Q_{kI} + \alpha((R_{UI} - \sum_{k=1}^K P_{U,k} Q_{k,I}) P_{Uk} - \lambda Q_{kI})$$

# 隐语义模型

## ✓ 隐语义模型负样本选择

✎ 对每个用户，要保证正负样本的平衡（数目相似）

✎ 选取那些很热门，而用户却没有行为的物品

✎ 对于用户—物品集  $K \{(u,i)\}$   
其中如果  $(u, i)$  是正样本，则有  $r_{ui} = 1$ ，负样本则  $r_{ui} = 0$

# 隐语义模型

## ✓ 隐语义模型参数选择

✎ 隐特征的个数 $F$ ，通常 $F=100$

✎ 学习速率 $\alpha$ ，别太大

✎ 正则化参数 $\lambda$ ，别太大

✎ 负样本/正样本比例 ratio

ratio	准 确 率	召 回 率	覆 盖 率
1	21.74%	10.50%	51.19%
2	24.32%	11.75%	53.17%
3	25.66%	12.39%	50.41%
5	26.94%	13.01%	44.25%
10	27.74%	13.40%	33.87%
20	27.37%	13.22%	24.30%



# 隐语义模型

## ✓ 协同过滤VS隐语义

✎ 原理：协同过滤基于统计，隐语义基于建模

✎ 空间复杂度，隐语义模型较小

✎ 实时推荐依旧难，目前离线计算多

✎ 隐语义模型咋解释呢？不解释

# 评估指标

✓ 评估标准：

✎ 准确度：
$$\text{RMSE} = \frac{\sqrt{\sum_{u,i \in T} (r_{ui} - \hat{r}_{ui})^2}}{|T|}$$

✎ 召回率：
$$\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}$$

令  $R(u)$  是根据用户在训练集上的行为给用户作出的推荐列表,  $T(u)$  是用户在测试集上的行为列表

# 评估指标

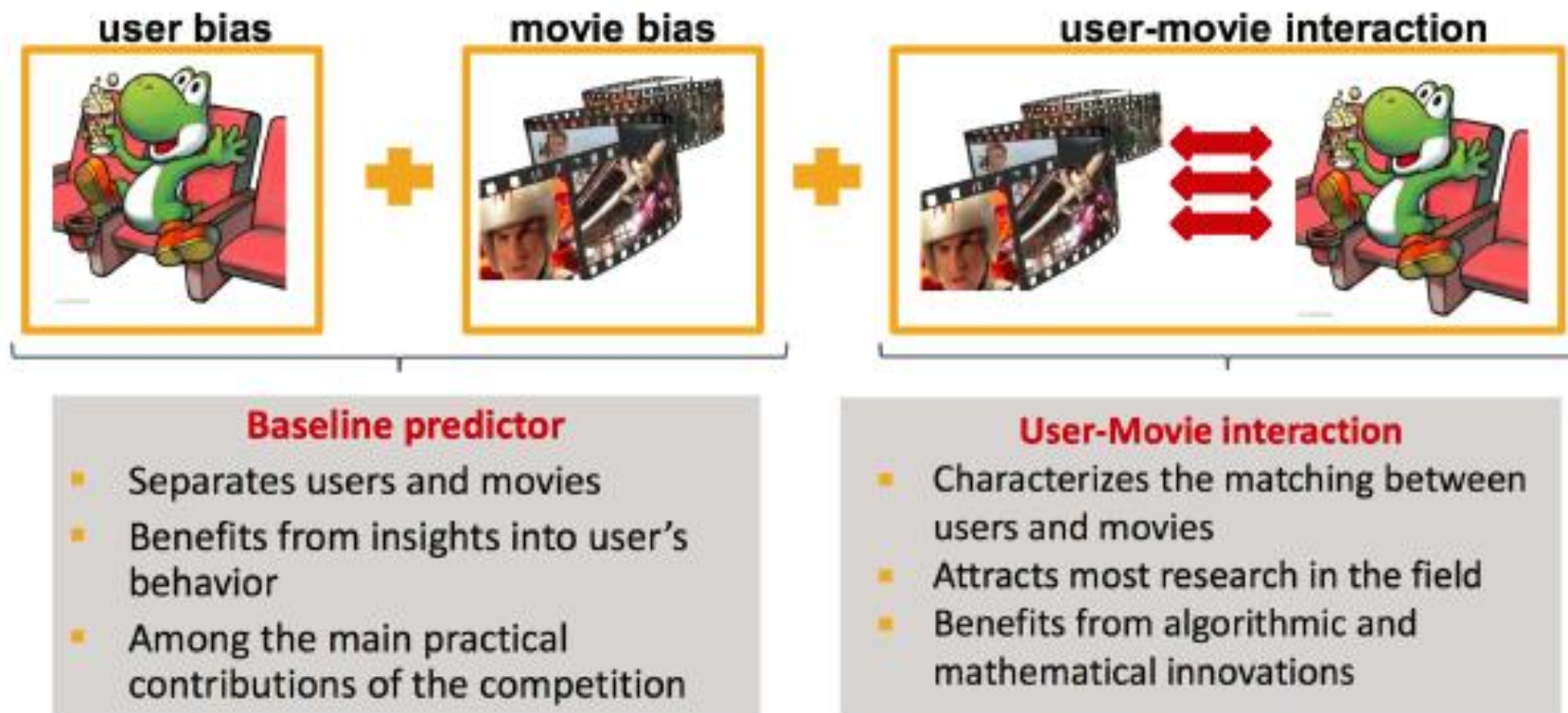
✓ 评估标准：

✎ 覆盖率：
$$\text{Coverage} = \frac{|\bigcup_{u \in U} R(u)|}{|I|}$$

$$H = -\sum_{i=1}^n p(i) \log p(i)$$

✎ 多样性：
$$\text{Diversity} = 1 - \frac{\sum_{i,j \in R(u), i \neq j} s(i,j)}{\frac{1}{2}|R(u)|(|R(u)|-1)}$$

# 推荐系统



- $\mu$  = overall mean rating
- $b_x$  = bias of user  $x$
- $b_i$  = bias of movie  $i$

# 推荐系统

$$r_{xi} = \underbrace{\mu}_{\text{Overall mean rating}} + \underbrace{b_x}_{\text{Bias for user } x} + \underbrace{b_i}_{\text{Bias for movie } i} + \underbrace{q_i \cdot p_x}_{\text{User-Movie interaction}}$$

## ■ Example:

- Mean rating:  $\mu = 3.7$
- You are a critical reviewer: your ratings are 1 star lower than the mean:  $b_x = -1$
- Star Wars gets a mean rating of 0.5 higher than average movie:  $b_i = +0.5$
- Predicted rating for you on Star Wars:  
 $= 3.7 - 1 + 0.5 = 3.2$

# 推荐系统

---

