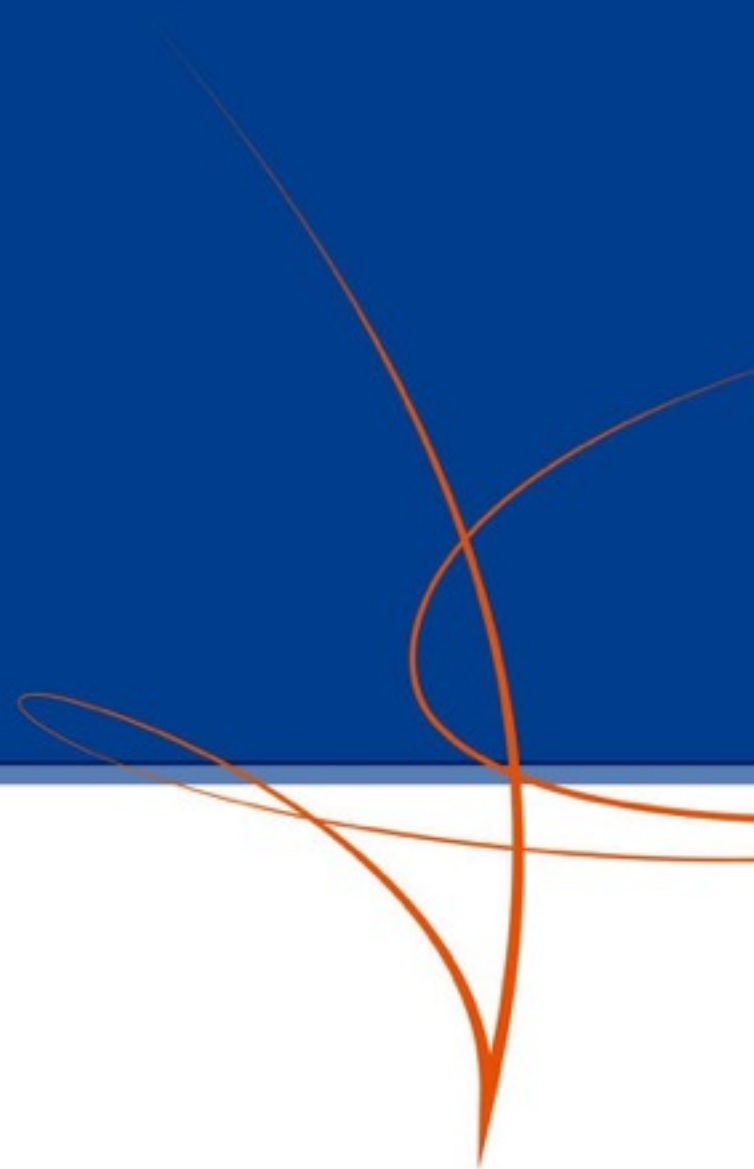


推荐系统



■ 推荐系统与评估

1. 推荐系统广泛应用
2. 推荐系统需求
3. 推荐系统结构与评估

■ 推荐算法串讲

1. 基于内容推荐 (content-based algorithms)
2. 协同过滤 (neighborhood-based algorithms)


TV Shows

Genres ▾







MONEY HEIST

Thieves take hostages and euros as a criminal mastermind manipulates the police.

▶ Play+ My Listⓘ More Info



Popular on Netflix





□ 一种数学定义：

- 设 C 为全体用户集合
- 设 S 为全部商品/推荐内容集合
- 设 u 是评判把 s_i 推荐 c_i 的好坏评判函数
- 推荐是对于 $c \in C$ ，找到 $s \in S$ ，使得 u 最大，即

$$\forall c \in C, s'_c = \operatorname{argmax}_{s \in S} (u(c, s))$$

- 部分场景下是Top N推荐

□ 通俗地说，推荐系统需要：

■ 根据用户的：

- a) 历史行为
- b) 社交关系
- c) 兴趣点
- d) 所处上下文环境
- e) ...

去判断用户的当前需求/感兴趣的item

□ 互联网大爆炸 => 信息过载

■ 一个人一天内

- 会看到20mb左右的文字信息
- 会听到600mb左右声音信息
- 每秒看到2mb左右图像信息

■ 每天有10w左右的新闻报道

■ 每秒钟优酷土豆爱奇艺搜狐腾讯B站会多出时长几百小时的视频

■ 淘宝京东亚马逊当当一天上架上百w商品

■ ...

The Economist, November 2006

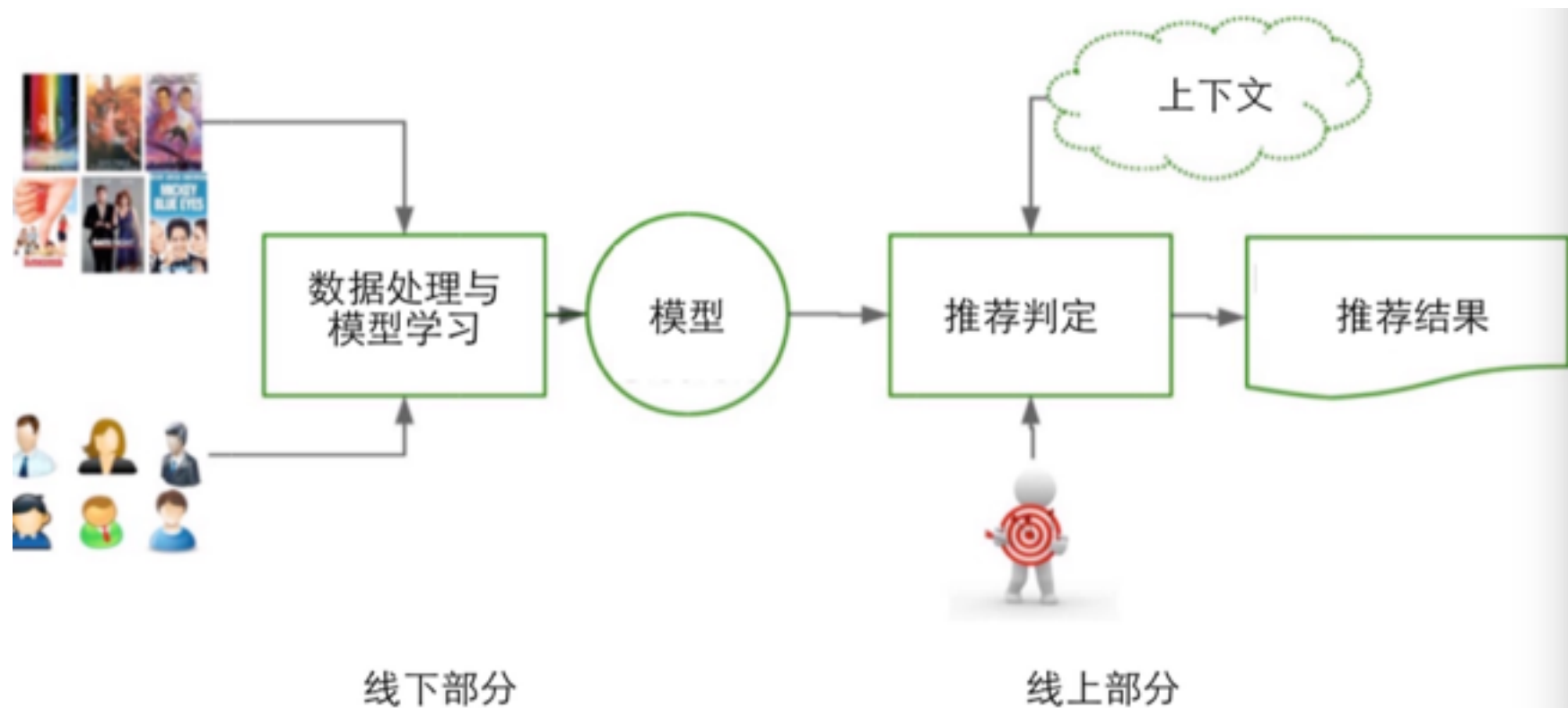
- 寻求解决信息过载的思路
- 思路变更
 - 分类导航页 => 雅虎
 - 搜索引擎 => 谷歌，必应，度娘
- 但是，人总是期望计算机尽量多地服务
 - 我们不愿意去想搜索词
 - 希望系统自动挖掘自己的兴趣点
 - 希望系统能给我们惊喜
- 今日头条，虾米音乐，电商猜你喜欢，豆瓣...

□ 商家需要推荐系统吗？

- Netflix每年2/3的观看电影from推荐
- Google news推荐系统能带来额外38%的点击
- 亚马逊每年35%的销售额都来源于它的推荐
- 头条半数以上新闻和广告点击来源于推荐
- 京东一年推荐和广告带来几亿的营收
- ...

□ 对用户而言：

- 找到好玩的东西
- 帮助决策
- 发现新鲜事物
- ...



□ 准确度:

① 打分系统

设 r_{ui} 为用户 u 对物品 i 的实际评分, \hat{r}_{ui} 为预测分
则有如下误差判定标准:

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

$$MAE = \frac{\sum_{u,i \in T} |r_{ui} - \hat{r}_{ui}|}{|T|}$$

□ 准确度:

② Top N推荐

设 $R(u)$ 为根据训练建立的模型在测试集上的推荐,
 $T(u)$ 为测试集上用户的选择。

准确率 vs 召回率

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|}$$

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}$$

□ 覆盖率:

■ 表示对物品长尾的发掘能力（推荐系统希望消除马太效应）

$$Coverage = \frac{|\cup_{u \in U} R(u)|}{|I|}$$

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

□ 多样性:

- 优秀的推荐系统能保证推荐结果列表中物品的丰富性（两两之间的差异性）。
- 设 $s(i,j)$ 表示物品 i 和 j 之间的相似度，多样性表示如下：

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

$$Diversity = \frac{1}{|U|} \sum_{u \in U} Diversity(R(u))$$

□ 基于内容的推荐

- 基于用户喜欢的物品的属性/内容进行推荐
- 需要分析内容，无需考虑用户与用户之间的关联
- 通常使用在文本相关产品上进行推荐
- 物品通过内容(比如关键词)关联：
 - 电影题材：爱情/探险/动作/喜剧/悬疑
 - 标志特征：黄晓明/王宝强...
 - 年代：1995, 2016...
 - 关键词
- 基于比对物品内容进行推荐

□ 基于内容的推荐

■ 对于每个要推荐的内容，我们需要建立一份资料

➤ 比如词 k_j 在文件 d_j 中的权重 w_{ij}

➤ 常用的方法比如TF-IDF

■ 需要对用户也建立一份资料：

➤ 比如说定义一个权重向量 (w_{c1}, \dots, w_{ck})

➤ 其中 w_{ci} 表示第 k_i 个词对用户 c 的重要度

■ 计算匹配度

➤ 比如用余弦距离公式

$$u(c, s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2} = \frac{\sum_{i=1}^K w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^K w_{i,c}^2} \sqrt{\sum_{i=1}^K w_{i,s}^2}}$$

□ 小例子

■ 基于书名进行书推荐 (只是帮助理解, 不一定有实际意义)

- 用户对《Building data mining applications for CRM》这本书感兴趣
- 从以下书中进行推荐

Building data mining applications for CRM

Accelerating Customer Relationships: Using CRM and Relationship Technologies

Mastering Data Mining: The Art and Science of Customer Relationship Management

Data Mining Your Website

Introduction to marketing

Consumer behavior

marketing research, a handbook

Customer knowledge manag

□ 基于标题内容相近度推荐

- 计算书名向量的相似度，推荐Top N接近的(这里n=3)。

- 结果如下：

rank 1: Data Mining Your Website

rank 2: Accelerating Custom Relationships: Using CRM ...

rank 3: Mastering Data Mining: The Art and Science...

其余未推荐...

□ Neighborhood-based algorithm

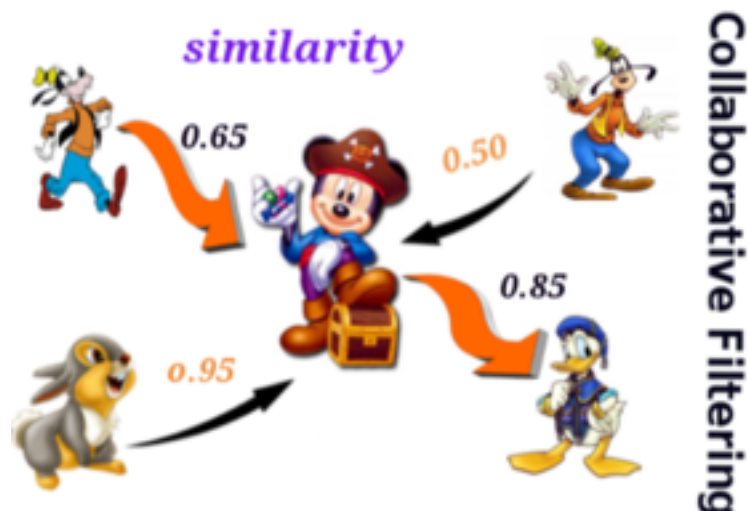
■ 协同过滤是一种基于“近邻”的推荐算法

■ 根据用户在物品上的行为找到物品或者用户的“近邻”



□ 基于用户的协同过滤 (*user-based CF*)

- 基于用户有共同行为的物品，计算用户相似度
- 找到“近邻”，对近邻在**新**物品的评价(打分)加权推荐



□ 基于物品的协同过滤 (*item-based CF*)

- 对于有相同用户交互的物品，计算物品相似度
- 找到物品“近邻”，进行推荐



□ 相似度/距离定义

■ 欧氏距离

$$\text{dist}(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

■ Jaccard相似度

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

■ 余弦相似度

$$\cos(\theta) = \frac{a^T b}{|a| \cdot |b|}$$

■ Pearson相似度

$$\frac{\sum_{i=1}^n (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i=1}^n (X_i - \mu_X)^2} \sqrt{\sum_{i=1}^n (Y_i - \mu_Y)^2}}$$

□ 相似度/距离定义

■ 欧氏距离

$$\text{dist}(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

■ Jaccard相似度

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

■ 余弦相似度

$$\cos(\theta) = \frac{a^T b}{|a| \cdot |b|}$$

■ Pearson相似度

$$\frac{\sum_{i=1}^n (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i=1}^n (X_i - \mu_X)^2} \sqrt{\sum_{i=1}^n (Y_i - \mu_Y)^2}}$$

□ 基于物品的协同过滤

- 一个用户序列 $u_i, i=1\dots n$, 一个物品序列 $p_j, j=1\dots m$
- $n \times m$ 得分矩阵 v , 每个元素 v_{ij} 表示用户 i 对物品 j 的打分
- 计算物品 i 和物品 j 之间的相似度/距离

$$S(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2} = \frac{\sum_{i=1}^n (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i=1}^n (X_i - \mu_X)^2} \sqrt{\sum_{i=1}^n (Y_i - \mu_Y)^2}}$$

- 选取 **Top K** 推荐或者加权预测得分

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij} similarity of items i and j
 r_{xj} rating of user u on item j
 $N(i;x)$ set items rated by x similar to i

□ 基于物品的协同过滤

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



- estimate rating of movie 1 by user 5

□ 基于物品的协同过滤

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

□ 基于物品的协同过滤

		users											
		1	2	3	4	5	6	7	8	9	10	11	12
movies	1	1		3		2.6	5			5		4	
	2			5	4			4			2	1	3
	<u>3</u>	2	4		1	2		3		4	3	5	
	4		2	4		5			4			2	
	5			4	3	4	2					2	5
	<u>6</u>	1		3		3			2			4	

Predict by taking weighted average:

$$r_{1.5} = (0.41 \cdot 2 + 0.59 \cdot 3) / (0.41 + 0.59) = 2.6$$

$$r_{ix} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{jx}}{\sum s_{ij}}$$

□ 基于用户的协同过滤

- 一个用户序列 $u_i, i=1\dots n$, 一个物品序列 $p_j, j=1\dots m$
- $n \times m$ 得分矩阵 v , 每个元素 v_{ij} 表示用户 i 对物品 j 的打分
- 计算用户相似度 (距离)

$$u_{ik} = \frac{\sum_j (v_{ij} - v_i)(v_{kj} - v_k)}{\sqrt{\sum_j (v_{ij} - v_i)^2 \sum_j (v_{kj} - v_k)^2}} \quad \text{or} \quad \cos(u_i, u_j) = \frac{\sum_{k=1}^m v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^m v_{ik}^2 \sum_{k=1}^m v_{jk}^2}}$$

■ 预测得分

$$v_{ij}^* = K \sum_{v_{ki} \neq ?} u_{jk} v_{kj} \quad \text{or} \quad v_{ij}^* = v_i + K \sum_{v_{ki} \neq ?} u_{jk} (v_{kj} - v_k)$$

□ *User-based CF vs Item-based CF*

	UserCF	ItemCF
性能	适用于用户较少的场合，如果用户很多，计算用户相似度矩阵代价很大	适用于物品数明显小于用户数的场合，如果物品很多（网页），计算物品相似度矩阵代价很大
领域	时效性较强，用户个性化兴趣不太明显的领域	长尾物品丰富，用户个性化需求强烈的领域
实时性	用户有新行为，不一定造成推荐结果的立即变化	用户有新行为，一定会导致推荐结果的实时变化
冷启动	<p>在新用户对很少的物品产生行为后，不能立即对他进行个性化推荐，因为用户相似度表是每隔一段时间离线计算的</p> <p>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对它产生行为的用户兴趣相似的其他用户</p>	<p>新用户只要对一个物品产生行为，就可以给他推荐和该物品相关的其他物品</p> <p>但没有办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</p>
推荐理由	很难提供令用户信服的推荐解释	利用用户的历史行为给用户做推荐解释，可以令用户比较信服