

Part 1 基本知识

基本方法：

1. 社会化推荐：让好友给自己推荐物品
2. 基于内容的推荐：根据一些确定的信息进行推荐，比如已知喜欢的演员或者导演，直接推荐这个演员/导演的电影
3. 基于协同过滤：找兴趣相似的一群用户，把他们喜欢的物品作为推荐项，或者与自己的历史相似的物品，直接作为推荐项。

实验方法：

1. 离线实验（训练集，测试集）
优点：不需要有对实际系统的控制权，不需要用户参与实验，速度快，可以测试大量算法
缺点：无法计算商业上关心的指标，离线实验的指标与商业指标存在差距。
2. 用户调查（需要一些真实的用户，让他们在需要测试的推荐系统上完成一定的任务，并让他们回答一些问题）
优点：更真实，可以获得一些独特的指标，例如惊喜度，相对在线实验风险较低
缺点：成本高，选择时需要保证较为真实的分布，双盲实验。
3. 在线实验（AB 测试：按一定的规则将用户随机分成几组，并对不同组的用户采用不同的算法，然后对比不同组的评测指标）
优点：公平地获得不同算法实际在线时的性能指标
缺点：周期长（因此一般只在线测试前两步中表现比较好的算法）
系统设计比较复杂，因为前端到后端每个层可能都需要做 AB 测试，因此切分流量比较关键，不同层之间的流量不应该有重合。
一般是 离线实验（离线指标优）->用户调查（保证用户满意度）->在线实验

评价指标：

1. 用户满意度（询问用户是否满意，或是用点击率，停留时间等指标进行衡量）
2. 预测准确度（离线评测指标）
 - a) 评分预测

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

均方根误差

$$\text{MAE} = \frac{\sum_{u,v \in T} |r_{uv} - \hat{r}_{uv}|}{|T|}$$

平均绝对误差

- b) TopN 推荐(给用户一个个性化的推荐列表，一般通过准确率或者召回率进行衡量)

$$\text{Recall} = \frac{\sum_{u,v} |R(u) \cap T(u)|}{\sum_{u,v} |T(u)|} \quad \text{Precision} = \frac{\sum_{u,v} |R(u) \cap T(u)|}{\sum_{u,v} |R(u)|}$$

recall 衡量预测准确的行为占真实发生的行为的比例，precision 衡量预测准确的行为在整个预测列表中的比例。

3. 覆盖率（描述一个推荐系统对物品的长尾的发掘能力）

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

- a) 粗略的公式 (随机推荐可以达到很高的覆盖率但是会损失满意度)

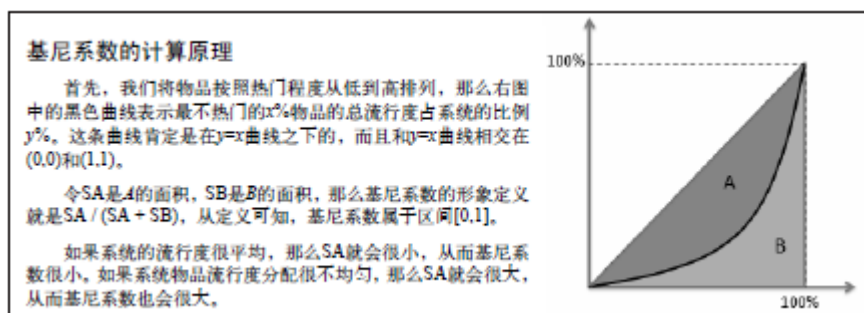
$$H = -\sum_{i=1}^n p(i) \log p(i)$$

- b) 信息熵 $p(i)$ 是物品 i 的流行度除以所有物品流行度之和

$$G = \frac{1}{n-1} \sum_{j=1}^n (2j-n-1)p(i_j)$$

- c) 基尼系数 $p(i_j)$ 是按照物品流行度 $p()$ 从小到大排序的物品列

表中第 j 个物品(这个公式感觉有点不是很懂,但是想表达的意思是下面这样)



4. 多样性

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

- a) 用户 u 推荐列表的多样性 $s(i,j)$ 是物品的相似度

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

- b) 推荐系统整体的多样性 所有用户多样性的均值

5. 新颖性 (推荐给用户那些他们以前没有听说过的物品, 注意这里可能还会有用户在别的网站上交互过的历史, 因为准确的新颖性评估只能通过用户调查)

6. 惊喜度:

定义惊喜度首先要定义推荐结果和用户历史上喜欢的物品的相似度, 其次需要定义用户对结果的满意度。

7. 信任度:

用户对系统产生信任可以增加用户与推荐系统的交互。

这个只能通过问卷调查得出。

增加的方法有两个 (1) 增加透明度 (推荐解释) (2) 使用用户的社交网络信息

8. 实时性:

- a) 实时更新列表来满足用户新的行为变化(离线的可能会一天计算一次然后才更新推荐列表) 可通过推荐列表的变化速率进行评测

- b) 能即使将新加入系统的物品推荐给用户

9. 健壮性 (防止作弊和反作弊斗争, 如行为注入攻击: 自己申请大量账号来实施行为)

- a) 通过模拟攻击来进行评测。

- b) 设计系统使使用代价比较高的用户行为进行评估。

- c) 使用数据前进行攻击检测，清理数据。

表1-3 获取各种评测指标的途径

	离线实验	问卷调查	在线实验
用户满意度	×	✓	○
预测准确度	✓	✓	×
覆盖率	✓	✓	✓
多样性	○	✓	○
新颖性	○	✓	○
惊喜度	×	✓	×

Part 2 早期算法/模型

用户行为的相关属性：

- 显性反馈行为（明确表明对物品喜好的行为，例如评分）
- 隐性反馈行为（不能明确反映用户喜好的，例如页面浏览行为）

表2-1 显性反馈数据和隐性反馈数据的比较

	显性反馈数据	隐性反馈数据
用户兴趣	明确	不明确
数量	较少	庞大
存储	数据库	分布式文件系统
实时读取	实时	有延迟
正负反馈	都有	只有正反馈

表2-2 各代表网站中显性反馈数据和隐性反馈数据的例子

	显性反馈	隐性反馈
视频网站	用户对视频的评分	用户观看视频的日志、浏览视频页面的日志
电子商务网站	用户对商品的评分	购买日志、浏览日志
门户网站	用户对新闻的评分	阅读新闻的日志
音乐网站	用户对音乐/歌手/专辑的评分	听歌的日志

表2-3 用户行为的统一表示

user id	产生行为的用户的唯一标识
item id	产生行为的对象的唯一标识
behavior type	行为的种类（比如是购买还是浏览）
context	产生行为的上下文，包括时间和地点等
behavior weight	行为的权重（如果是观看视频的行为，那么这个权重可以是观看时长；如果是打分行为，这个权重可以是分数）
behavior content	行为的内容（如果是评论行为，那么就是评论的文本；如果是打标签的行为，就是标签）

数据集的种类：

- 无上下文信息的隐形反馈数据集(用户 ID 和物品 ID) Book-Crossing
- 无上下文信息的显性反馈数据集(用户 ID 和物品 ID，评分)
- 有上下文信息的隐形反馈数据集(用户 ID 和物品 ID，时间戳) Lastfm
- 有上下文信息的显性反馈数据集(用户 ID 和物品 ID，评分，时间戳) Netflix Prize

1. 基于用户的协同过滤算法

UserCF

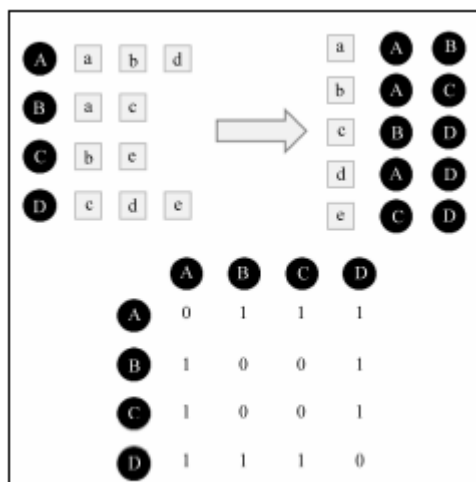


图2-7 物品-用户倒排表

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)| |N(v)|}}$$

$$p(u, i) = \sum_{v \in S(u, K) \cap N(i)} w_{uv} r_{vi}$$

首先根据交互记录算出用户两两的相似度，然后计算用户对物品的感兴趣程度时，找出和这个物品交互过的用户中较为相似的用户，用这个用户对于物品的兴趣乘以两用户的相似度

User-IIF（考虑到两个用户对冷门物品采取同样的行为更能说明他们兴趣的相似度）

$$w_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{\log 1 + |N(i)|}}{\sqrt{|N(u)| |N(v)|}}$$

惩罚了热门物品对于用户相似度的影响

2. 基于物品的协同过滤算法

ItemCF

- (1) 计算物品之间的相似度。
- (2) 根据物品的相似度和用户的历史行为给用户生成推荐列表。

$$w_{ij} = \frac{|N(i) \cap N(j)|}{\sqrt{|N(i)| |N(j)|}}$$

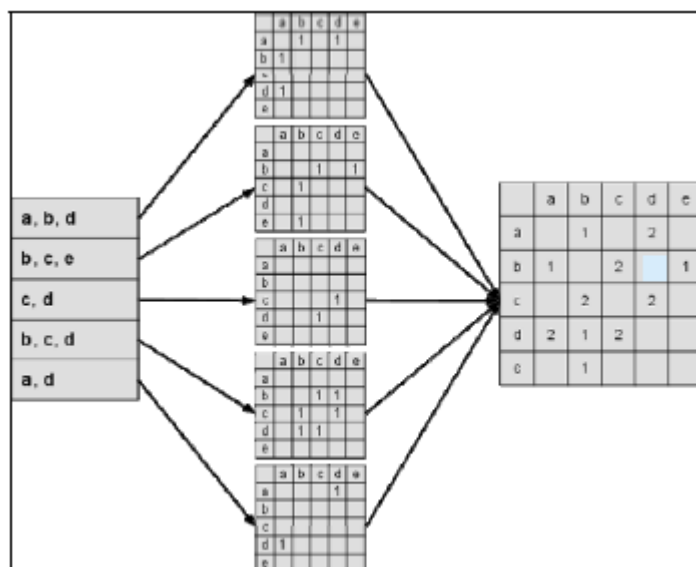


图2-11 一个计算物品相似度的简单例子

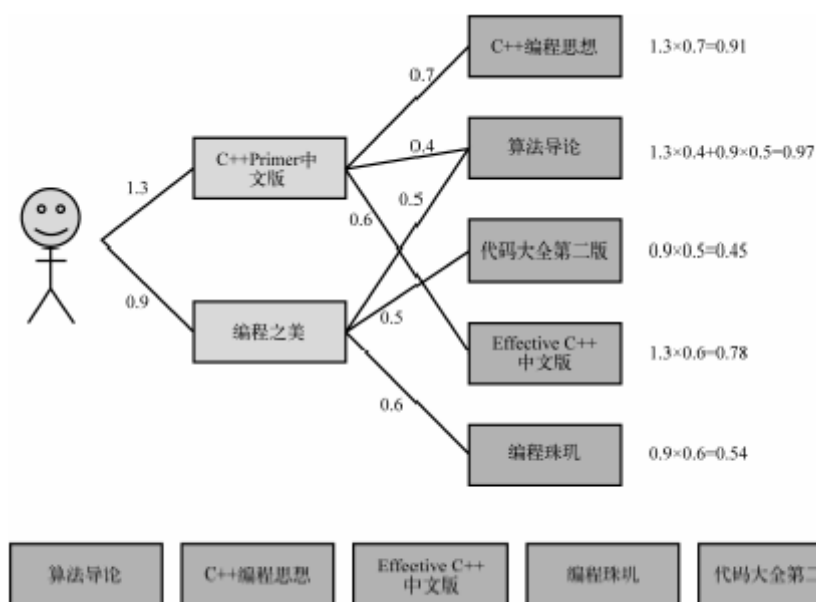


图2-12 一个简单的基于物品推荐的例子

ItemCF-IUF (认为活跃用户对于物品相似度的贡献要小于不活跃用户)

$$w_{ij} = \frac{\sum_{u \in N(i) \cap N(j)} \frac{1}{\log 1 + |N(u)|}}{\sqrt{|N(i)| |N(j)|}}$$

问题：两个不同领域的最热门物品之间往往具有比较高的相似度，这时候不能只依靠用户行为的数据来解决问题，还应该引入物品的内容数据（但这不是协同过滤的讨论范畴）

3. 隐语义模型 (LFM latent factor model)

先对物品进行分类，然后确定用户感兴趣的类别，然后在这个类里面挑选他可能喜欢的物品

问题：1.各种用户的意见 2.分类的粒度 3.一个物品可能属于多个类 4.多维度的分类 5.一个物品在某一个分类中的权重
是一种基于机器学习的方法

4. 基于图的模型

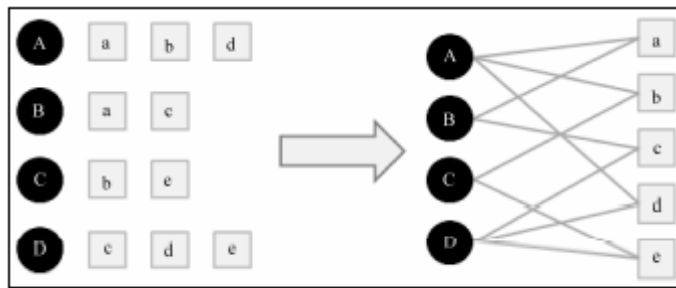


图2-18 用户物品二分图模型

相关性高的一对顶点有下列特征：

- 两个顶点之间有很多路径相连，连接两个顶点的路径长度比较短，连接两个顶点之间的路径不会经过出度比较大的顶点。
- 有一些基于以上特征的方法：personalRank（随机游走，到一个节点时以一定的概率决定是否继续游走，这样多次随机游走之后，每个节点访问到的概率会收敛，从而反映物品权重）

Part 3 冷启动问题

类型：用户冷启动 物品冷启动 系统冷启动

方法：非个性化的推荐（热门榜） 注册信息 社交网络账号登录从而利用社交信息
登录时对物品进行反馈 参考新加入物品的内容信息 专家知识

用户冷启动：

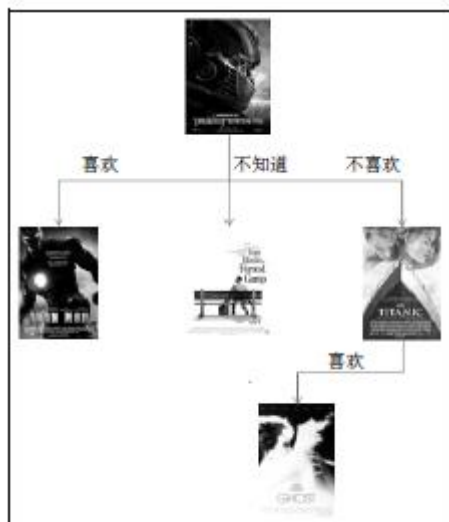
1. 利用用户注册信息

- 人口统计学信息
- 用户兴趣描述
- 从其他网站导入的用户站外行为数据

2. 选择合适的物品启动用户的兴趣

如何选择启动物品的集合呢？

根据对物品评分的方差来确定某个商品对于用户的区分度，用最高区分度的进行分类，之后对于每一类继续做相同的事情，直到最后的叶子节点，这时候对用户的兴趣已经有了比较清楚的了解。



截取自IMDB网站，图中相关内容的著作权归原作者所有人

物品冷启动：

1. 物品的内容信息

- 可以用**向量空间模型**进行表示

- 由于向量空间模型忽略了用户行为，流行度等等很多属性，只考虑了物体本身的特征，很多情况下效果远差与 item-based，但是如果用户选择强烈地受到某个属性的影响，协同过滤算法由于数据稀疏无法完全从用户行为中统计出这个特征，反而不如利用了先验信息的内容过滤算法，向量空间模型在内容数据丰富的时候可以获得比较好的效果

- **话题模型**：如何建立文章，话题和关键词的关系是话题模型研究的重点，代表模型有 LDA，是一种生成模型，对文档产生的过程进行建模，首先确定文章要讨论的话题，然后考虑这些话题要用什么词来描述，最终用词写一篇文章。那么基于这样的假设，LDA 做的就是给了一篇文章然后确定这篇文章有哪些主题，哪些词属于什么主题，这样可以实现词的聚类，找到每个词的相关词。

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

图3-12 通过LDA对词进行聚类的结果

- 还可以发挥**专家的作用**，请专家来为商品的内容进行标注。(Jinni 的电影基因系统)

Part 4 利用上下文信息

1. 物品的生存周期与系统的时效性：

- 物品平均在线天数（维基百科 VS 纽约时报）
- 相隔 T 天系统物品流行度向量的平均相似度
- 推荐的实时性：考虑用户的近期行为和长期行为

2. 时间多样性：

如果用户没有行为如何保证给用户的推荐结果具有时间多样性呢？

- 在生成推荐结果时加入一定的随机性
- 对前几天的推荐结果进行适当降权
- 每天给用户使用不同的推荐算法

3. 时间上下文推荐算法：

- 最近最热门
- 时间上下文相关的 itemCF 算法

物品相似度：相隔很短时间内喜欢的物品具有更高的相似度

$$\text{sim}(i, j) = \frac{\sum_{u \in N(i) \cap N(j)} f(|t_{ui} - t_{uj}|)}{\sqrt{|N(i)| |N(j)|}}$$

引入了一项时间相关的衰减项

在线推荐：用户近期的行为相比于用户很久以前的行为更能体现用户现在的兴趣。

$$p(u, i) = \sum_{j \in N(u) \cap S(i, K)} \text{sim}(i, j) \frac{1}{1 + \beta |t_0 - t_{ij}|}$$

时间上下文相关的 userCF 算法

1 **用户兴趣相似度**：如果两个用户同时喜欢相同的物品，则他们应该有更大的兴趣相似度

$$w_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{1 + \alpha |t_{ui} - t_{vi}|}}{\sqrt{|N(u)| |N(v)|}}$$

2 **相似兴趣用户的最近行为**：应该给用户推荐和他兴趣相似的用户最近喜欢的物品。

$$p(u, i) = \sum_{v \in S(u, K)} w_{uv} r_{vi} \frac{1}{1 + \alpha (t_0 - t_{vi})}$$

c) 时间段图模型：

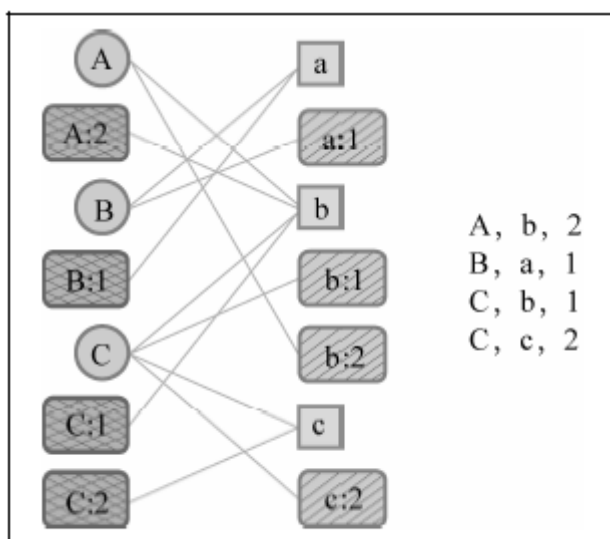


图5-8 时间段图模型示例

由于 personRank 在全图进行迭代计算，所以时间复杂度比较高，这里采用路径融合算法。首先提取出两个顶点之间长度小于一个阈值的所有路径，然后根据每条路径经过的顶点给每条路径赋予一定的权重，最后将两个顶点之间所有的路径权重之和作为两个顶点的关联度。

4. **地点上下文信息**：兴趣本地化，活动本地化

（用户，用户位置，物品，评分）LARS-U 从根节点出发，用每个中间节点上的数据训练一个推荐模型，从而为该用户生成推荐列表，然后把这一系列推荐列表加权

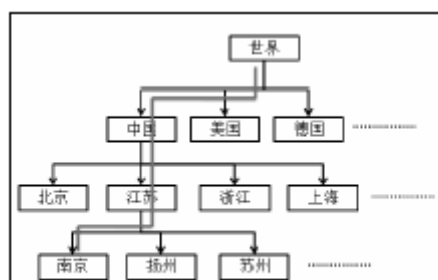


图5-16 一个简单的利用用户位置信息进行推荐的例子

(用户, 物品, 物品位置, 评分) 先忽略物品的位置信息, 用 itemCF 算出用户对物品的兴趣, 然后再减去一个 penalty (位置惩罚项), 如果当前物品与用户之前评分所有物品计算的平均距离相差越大, 则权重越低。

$\text{RecScore}(u, i) = P(u, i) - \text{TravelPenalty}(u, i)$ 后者可用欧式距离或是交通网络数据衡量